Thesis for the Degree of Doctor of Philosophy

Intrusion tolerant Trust-based Privacy-assured Security Solution for Wireless Sensor Networks

Riaz Ahmed Shaikh

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

August, 2009

Thesis for the Degree of Doctor of Philosophy

Intrusion tolerant Trust-based Privacy-assured Security Solution for Wireless Sensor Networks

By

Riaz Ahmed Shaikh

Supervised By

Prof. Young-Jae Song, Ph.D.

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

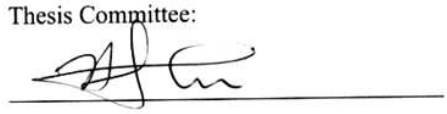
August, 2009

Intrusion tolerant Trust-based Privacy-assured Security Solution for Wireless Sensor Networks

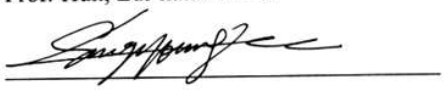
Riaz Ahmed Shaikh

Submitted to
the Faculty of the Graduate School of Computer Engineering
in Partial Fulfillment of the Requirements
for the Degree of
Ph.D.

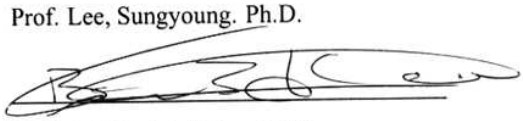
Thesis Committee:



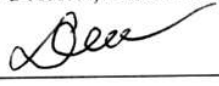
Prof. Huh, Eui-nam. Ph.D.



Prof. Lee, Sungyoung. Ph.D.



Prof. J. d'Auriol, Brian. Ph.D.



Prof. Lee, Heejo. Ph.D.

Prof. Song, Young-Jae. Ph.D.

Dedicated To

my beloved wife Sonia & caring brother Tariq Shaikh

Acknowledgments

Allah, the Compassionate, the Merciful

"Read in the name of your Lord Who created. He created man from a clot. Read and your Lord is Most Honorable, Who taught (to write) with the pen Taught man what he knew not. No indeed! Man surely transgresses, because he sees himself free from want. Surely to your Lord is the return."

(QURAN, 096:001-008)

First of all I am thankful to *Allah* for providing me intelligence and strength to carry out the demanding research for this thesis. Without this help I could not have finished my thesis work in its entirety in due time.

I would like to thanks my advisor Prof. Young-Jae Song and co-advisors Prof. Sungyoung Lee, Prof. Brian J. d'Auriol, Prof. Heejo Lee, and Prof. Eui-nam Huh who helped me in my thesis by providing insightful suggestions, comments, critics, and constant encouragement. These have greatly helped me to improve the quality and presentation of this thesis.

I also thank the following faculty members at KHU for their guidance and support in completion of this thesis. I am thankful to Prof. Young-Koo Lee, Prof. Muhammad Asmat Ullah Khan, and Prof. M. Kaykobad. I am also thankful for the support of uSec Research group for providing an engaging research environment. I am especially

thankful to Mr. Hassan Jameel for his sincere support and help throughout my research work.

I would like to extend my thanks to all of my friends and colleagues at KHU, especially, Umar, Ali, Shariyar, Faraz, Bilal, Uzair, Raazi, Shoaib, Obaid, Tahir, Adil, Mirza, Asad, Ozair, Le Xuan Hung, Truc, Weiwei, Donghai, Xiaoling, Lenin, Giang, Diep and others for their friendship and help to overcome the frustrations and difficulties throughout my thesis research. I would also like to thanks our Lab assistant Ms. Seoungae Kim for taking care of administrative tasks.

Finally, I would like to thankfully acknowledge the financial support of the Korean Government for providing me *IITA scholarship* to cover living expenses and Kyung Hee University for providing me *President scholarship* to cover tuition fees.

Riaz Ahmed Shaikh

August, 2009

Abstract

In Wireless Sensor Networks (WSNs), researchers have so far focuses on the individual aspects (cryptography, privacy or trust) of security that are capable of providing protection against specific types of attacks. However, efforts on achieving completeness via a composite and integrated security solution are lacking. Many cryptographic-based security solutions have been proposed, but surprisingly less importance is given to privacy and trust issues of WSNs. Later two aspects contribute in increasing the degree of completeness and reliability of a security solution. This thesis aims to achieve *more completeness and reliability in a security* solution for WSNs by addressing the requirements of high level security, energy, memory and communication overhead efficiency. The primary contribution is the schematic development of a unified, resource-efficient framework, called *intrusion tolerant trust-based privacy-assured security framework*. This framework is based on the cooperative interactions among individually proposed, mutually-complementary trust, privacy, and security solutions.

Existing trust management schemes of WSNs suffer from various limitations such as they do not meet the resource constraint requirements of the WSNs; and more specifically, for the large-scale WSNs. Also, they suffer from higher cost associated with trust evaluation specially of distant nodes. Thus, a new lightweight Group-based Trust Management Scheme (GTMS) is proposed for WSNs. It evaluates the trust of a group of

sensor nodes in contrast to traditional trust management schemes that always focus on trust values of individual nodes. This approach gives us the benefit of requiring less memory to store trust records at each sensor node. Also, theoretical as well as simulation results show that this scheme demands less memory, energy and communication overheads as compared with the current state-of-the-art trust management schemes. Furthermore, GTMS also detect and prevent malicious, selfish and faulty nodes.

Existing privacy schemes of WSNs only provide partial network level privacy. Full network level privacy spectrum comprises of identity, route, location, and data privacy. Providing full network level privacy is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g. energy, memory and computation power), sensor network (e.g. mobility, and topology) and QoS issues (e.g. packet reach-ability, and trustworthiness). Thus, two new identity, route and location privacy algorithms (IRL and r-IRL) and data privacy (DPriv) mechanism are proposed that addresses these problems. A new IRL privacy algorithm ensures the anonymity of source node's identity and location from the adversary. It also gives assurance that the packets will reach their destination by passing through only trusted intermediate nodes. A new r-IRL privacy algorithm has the ability to forward packets from multiple secure paths to increase the packet reach-ability. A new Data Privacy (DPriv) mechanism provides data secrecy and packet authentication *in the presence of identity anonymity*. Results show that these solutions are lightweight and provides protection against an adversary who is capable of performing privacy disclosure attacks e.g. eavesdropping and hop-by-hop trace backing.

Finally, a new Lightweight Security (LSec) solution is proposed that provides authentication, and authorization of sensor nodes. Also, LSec comprises of a simple secure key exchange mechanism that helps to provide data confidentiality. This security solution is memory efficient and introduces less communication overhead.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Complete security solution mainly comprises of three aspects: cryptography, privacy and trust. Security solution based on cryptography are mainly used to provide protection against security threats such as fabrication and modification of messages, unauthorized access, etc. For this purpose, assorted security mechanisms such as authentication, confidentiality, and message integrity are used. Additionally, these security mechanisms rely highly on a secure key exchange mechanism.

Security solution based on the trust features, such as reputation are mainly used to provide corresponding access control based on the judgment of the quality of nodes and their services [1]. Also, trust is used to provide complete reliable routing paths which are free from any malicious, selfish and faulty nodes [2, 3]. Therefore, incorporation of trust management features with cryptographic-based security mechanisms help in increasing robustness and reliability of the overall security solution.

Security solution based on the privacy features, such as route anonymity of the data packets, identity anonymity of nodes and their locations are mainly used to provide protection against security threats such as traffic analysis and eavesdropping. Additionally, these privacy features could also be used to provide protection against security threats such as camouflage [1, 4]. Therefore, the incorporation of these privacy features with cryptographic-based security mechanisms add to the degree of completeness of a secu-

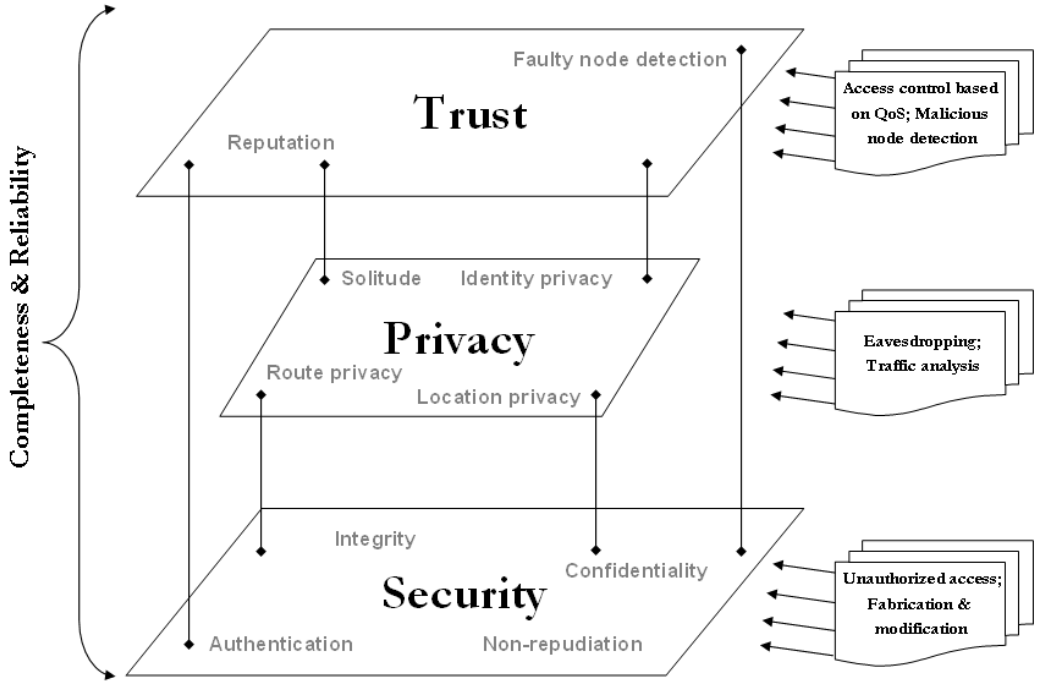


Figure 1.1: Complete security solution perspective

rity solution.

1.1 Motivation

With the emergence of ubiquitous computing, the need of ensuring single composite and integrated security solution is gaining more importance then ever before. In ubiquitous computing, computation is integrated within the environment in an invisible manner that enables people to interact with the computers anywhere, any time, and in any form [5]. One of the most important underlying technology used for ubiquitous computing is Wireless Sensor Networks (WSNs).

In WSNs, researchers have so far focuses on the individual aspects of security that are

capable of providing protection against specific types of attacks as shown in Figure 1.1. However, efforts on achieving completeness via a composite and integrated solution are lacking. That is ultimate necessary to attain because of its wide applicability in various sensitive applications, such as health-care, military, habitat monitoring etc. For example, in battlefield application scenario, "the location of a soldier should not be exposed if he initiates a broadcast query" [6]. In the meantime, query must be transferred to the destination in an encrypted manner via only trusted en-route nodes. Similarly, in habitat monitoring application scenarios, such as Great Duck Island [7] or Save-the-panda application [8], where large number of sensor nodes are deployed to observe the vast habitat for ducks and pandas respectively. In these scenarios, an adversary can try to capture the panda or duck by back-tracing the routing path until it reaches the source sensor nodes. Therefore, in order to prevent adversary from back-tracing, the route, location and data privacy mechanisms must be enforced. In the current era, it is unrealistic to assume that an attacker can perform single type of attack (like privacy disclosure attacks etc.) at a time. Now a day's, attacker is much more powerful and is capable of launching multi-dimension attacks at various layers simultaneously. Therefore, single composite and integrated security solution is needed to achieve some degree of completeness and reliability.

Current research so far intensively focuses on the cryptographic-based security aspects of WSNs. Many security solutions have been proposed such as SPINS [9], TinySec [10], and LEAP [11] etc., but surprisingly less focus has been given on privacy and trust issues of WSNs. As we mentioned earlier, cryptographic-based security solutions alone are not adequate. However, incorporating privacy and trust features in a security solution is not an easy task. This is due to the fact that the WSNs generally consist of large number of tiny sized sensor nodes. And they operate in highly resource-

constraint environment in which nodes have limited memory, energy, and computation power. Thus, lightweight privacy and trust management schemes are needed to ensure completeness and reliability in the security solution of WSNs.

1.2 Problems and Issues

Current state-of-the-art trust management schemes [12, 13, 14, 15, 16, 17] of wireless sensor networks suffer from various limitations, such as they do not meet the resource constraint requirements of the WSNs; and more specifically, for the large-scale WSNs. Also, these schemes suffer from higher cost associated with trust evaluation specially of distant nodes. Furthermore, existing schemes have some other limitations such as dependence on specific routing scheme, like the PLUS [13] scheme works on top of the PLUS_R routing scheme; dependence on specific platform, like the ATRM [14] requires an agent-based platform; and unrealistic assumptions, like the ATRM [14] assumes that agents are resilient against any security threats, etc. Therefore, these works are not well suited for realistic WSN applications. Thus, a lightweight trust management scheme is needed to address these issues.

Existing privacy schemes such as [6, 8, 18, 19, 20, 21] that have specifically been proposed for WSNs only provide partial network level privacy. Providing a full network level privacy spectrum that comprises of identity, route, location, and data privacy is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g. energy, memory and computation power), sensor network (e.g. mobility, and topology) and QoS issues (e.g. packet reach-ability, and trustworthiness). Thus, an energy-efficient privacy solution is needed to address these issues.

Cryptographic-based security schemes requires secure key management scheme. In general, there are three types of key management schemes [22, 23]: Trusted Server

scheme, self enforcing scheme and key pre-distribution scheme. Trusted server schemes relies on a trusted base station, that is responsible for establishing the key agreement between two communicating nodes as described in [24]. However, this approach is energy expensive; it requires extra routing overhead in the sense that each node need to communicate with base station several times [23]. Second approach is self enforcing that use public key cryptography for communication between sensor nodes. However, the problem with the traditional public keys cryptography schemes [25, 26] is that they require complex and intensive computations which is not possible to perform by sensor node having limited computation power. Some researchers [27, 1] use Elliptic curve cryptography as an alternative to traditional public key systems but still not perfect for sensor networks. Third approach is key pre-distribution that is based on symmetric key cryptography, in which limited number of keys are stored on each sensor node prior to their deployment. However, the degree of resiliency of node capture is dependent on the pre-distribution scheme [23]. Thus, a lightweight security solution is needed to address these issues.

1.3 Contributions

In this thesis, *more completeness and reliability in a security* solution is achieved for WSNs by addressing the requirements of high level security, energy, memory and communication overhead efficiency. The primary contribution is the schematic development of a unified, resource-efficient framework, called *intrusion tolerant trust-based privacy-assured security framework* as shown in Figure 1.2. This framework theoretically proves some degree of completeness and reliability in a security solution. This framework comprises of individually proposed, mutually complementary trust, privacy and security components that are discussed below.

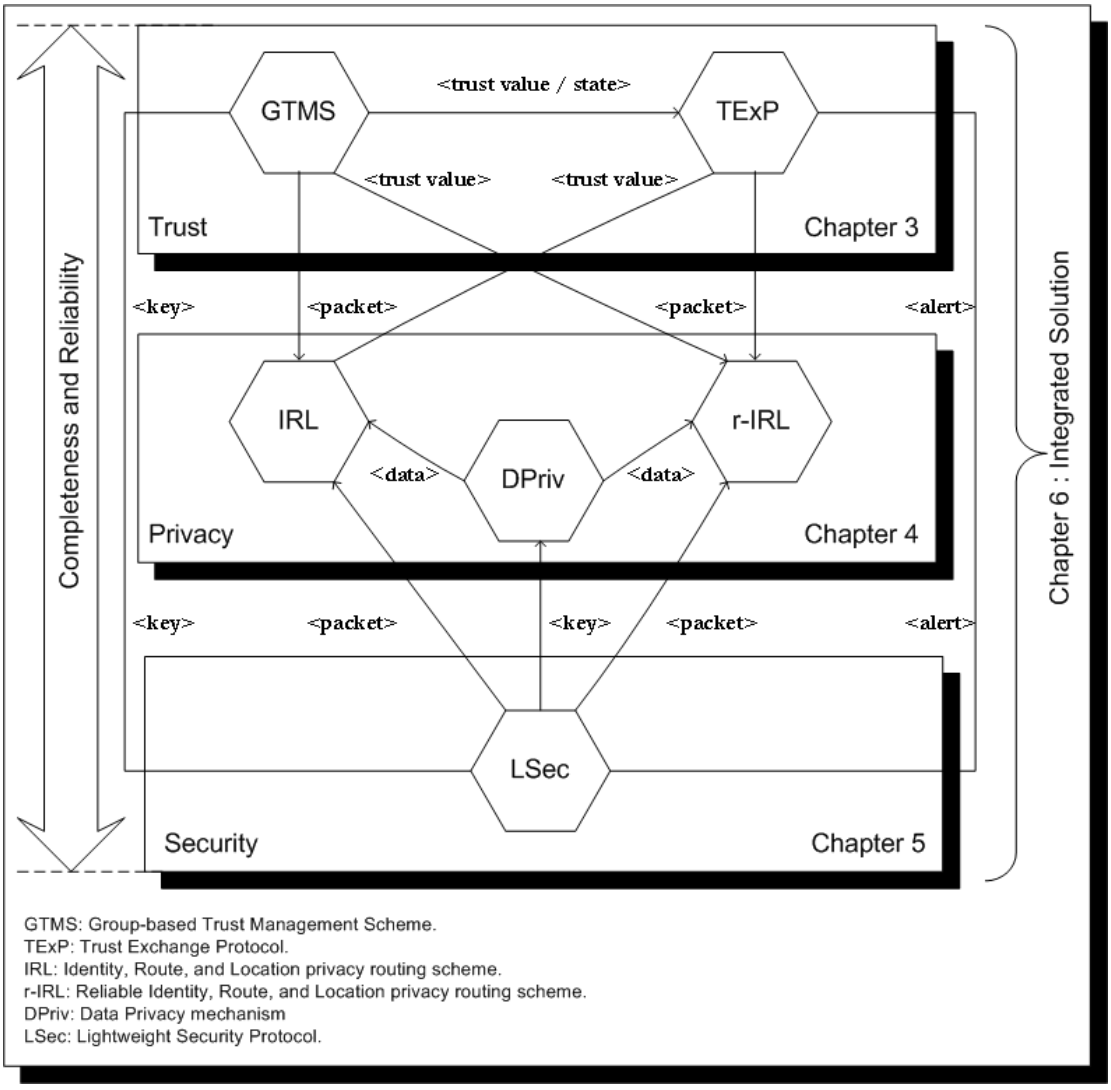


Figure 1.2: Intrusion tolerant trust-based privacy-aware security framework

Trust management: The problem of establishing and managing trust in wireless sensor networks is addressed and have the following contributions. A new lightweight Group-based Trust Management Scheme (GTMS) is proposed for WSNs. The GTMS consists of the three unique features:

1. GTMS evaluates the trust of a group of sensor nodes in contrast to traditional trust management schemes that always focus on trust values of individual nodes. This approach gives us the benefit of requiring less memory to store trust records at each sensor node in the network.

2. GTMS works on two topologies: intra-group topology where distributed trust management approach is used and inter-group topology where centralized trust management approach is adopted. This methodology helps to drastically reduce the cost associated with trust evaluation of distant nodes.

3. GTMS not only provides a mechanism to detect malicious nodes, but also provides some degree of prevention mechanism.

These and other specific features (e.g., independent of any specific routing scheme and platform etc.) collectively make the GTMS a new, lightweight, flexible, and robust solution that can be used in any clustered WSNs.

Privacy: The problem of achieving network level privacy in wireless sensor networks is addressed and have the following contributions.

- A new Identity, Route and Location (IRL) privacy algorithm is proposed that ensures the anonymity of source node's identity and location from the adversary. It also gives assurance that the packets will reach their destination by passing through only trusted intermediate nodes.

- A new reliable Identity, Route and Location (r-IRL) privacy algorithm is proposed, which is the extension of proposed IRL algorithm. This algorithm has the ability to forward packets from multiple secure paths to increase the packet reach-ability.
- A new Data Privacy (DPriv) mechanism is proposed, which is unique in the sense that it provides data secrecy and packet authentication *in the presence of identity anonymity*.

These solutions collectively provide protection against various privacy disclosure attacks such as eavesdropping and hop-by-hop trace-back attacks. Also, these solutions are light-weight and hence consume modest memory and energy.

Security: The problem of developing an energy-efficient security solution is addressed and have the following contributions.

- A new Lightweight Security (LSec) protocol is proposed that provides authentication, and authorization of sensor nodes.
- A simple secure key exchange mechanism is proposed that helps to provide data confidentiality.

This security solution is memory efficient and introduces less communication overhead.

1.4 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 presents the taxonomy of trust, privacy and security. Also, it contains detailed critical analysis of the current state-of-the-art research work.

Chapter 3 presents a lightweight group-based trust management scheme for wireless sensor networks. This chapter describes the trust evaluation and management mecha-

nism and then presents theoretical as well as simulation-based analysis and evaluation of proposed scheme.

Chapter 4 presents two new identity, route and location privacy algorithms and data privacy mechanism. This chapter further describes the protection mechanism provided by proposed privacy schemes against various privacy disclosure attacks such as eavesdropping and hop-by-hop trace-back attacks. Also, this chapter presents the simulation-based analysis and evaluation.

Chapter 5 presents a lightweight security protocol for wireless senor networks. This chapter describes the procedures for authentication, authorization, and key exchange. It then evaluates and discusses the performance of the proposed protocol using simulation.

Chapter 6 proposes a unified, resource-efficient framework, called *intrusion tolerant trust-based privacy-assured security framework* for wireless sensor networks. This chapter contains the interface description of proposed privacy, security and trust components. Also, this chapter provides theoretical analysis of the complete solution from the perspective of memory consumption and communication overhead.

Conclusions are drawn in Chapter 7. This chapter also discusses possible future research directions.

Chapter 2

Related Work

2.1 Introduction

Until recently, researchers have focused on the cryptographic-based security issues more intensively than the privacy and trust issues. However, without the incorporation of trust and privacy features, cryptographic-based security mechanisms are not capable of singlehandedly providing robustness, reliability and completeness in a security solution. The soft relationship between privacy, cryptographic-based security, and trust is shown in Figure 2.1, which illustrate the related aspects of these terms with other commonly found terms of the security domain. For example, confidentiality is a mutual feature of cryptographic-based security and privacy aspects. In order to provide confidentiality, cipher algorithms (such as AES, DES etc.) are used to prevent disclosure of information from any unauthorized entity. Similarly, an intrusion detection system may need a trust management feature such as reputation, as well as cryptographic-based security feature, such as integrity checking, to detect any malicious nodes. Also, solitude, which is used to isolate a node from the network either willingly or forcefully, is a mutual feature of trust and privacy aspects.

In this chapter, I present generic and flexible taxonomies of privacy and trust. I also give detailed critical analyses of the state-of-the-art research, in the field of privacy and

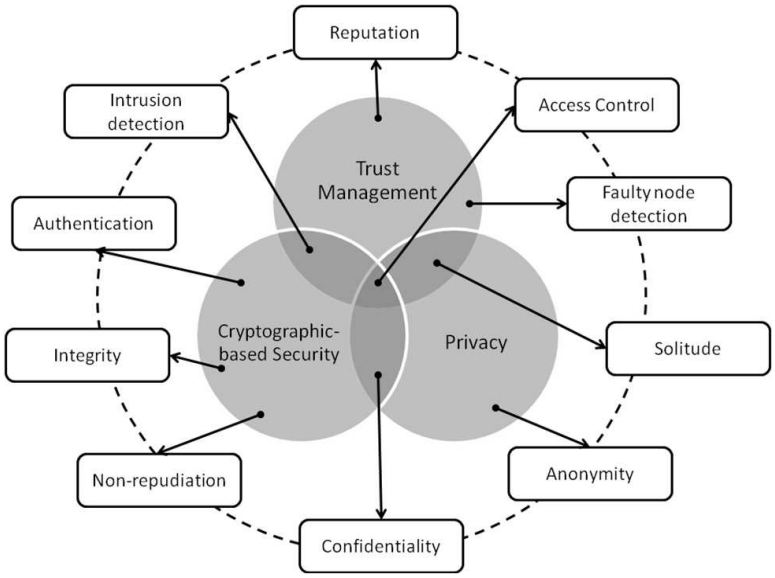


Figure 2.1: Relationship between privacy, trust and cryptographic-based security

trust that is currently not available in the literature.

2.2 Trust

2.2.1 Taxonomy of Trust

Trust management schemes are classified into three categories: centralized, distributed and hybrid as shown in Figure 2.2.

Centralized trust management (CTM) schemes (e.g. [28, 29] consist of a single globally trusted server that determines the trust values of every node in the network. This gives the benefit of lesser computational overhead at the sensor node because most of the trust calculation is performed at centralized trusted server that has no constraints of computational power and memory. This approach however has the drawbacks of a single point of failure, which makes it least reliable. Also, it suppresses the underlying fact

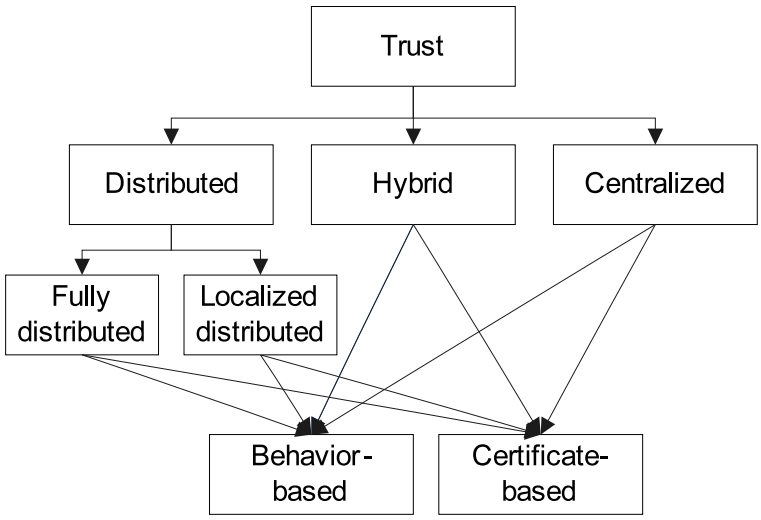


Figure 2.2: Taxonomy of trust

that different nodes may have different trust values about a particular given node [30]. For large scale sensor networks, centralized trust schemes are not suitable because the total routing cost for the exchange of trust values of a sensor node with the base station is quite energy expensive, especially when the base station is located far from the node. Therefore centralized approach introduces large communication overhead in the sensor network.

Distributed trust management (DTM) schemes (e.g. [12, 14]) also do not work well for large-scale sensor networks. In the distributed approach, every node locally calculates the trust values of all other nodes in the network that increases the computational cost. Also each node needs to maintain an up-to-date record about the trust values of the entire network in the form of a table. Size of the table is directly proportional to size of the network which results in large memory consumption. Each sensor node maintains its own trust record that gives the benefit of less communication overhead because node does not need to contact with some centralized server. Distributed approach is more reli-

able than the centralized one because it has no single point of failure. In wireless sensor network domain, some researchers use restricted DTM approach, in which sensor nodes only maintains the trust value about its neighboring nodes only e.g. [12]. We refer that approach as a localized DTM approach and the earlier one as a fully DTM approach, e.g. [14]. The major drawback of the localized DTM approach is that it introduces delay and dependency whenever any node wants to evaluate trust of distant nodes. This is due to the fact that trust is established "dynamically at runtime using the chain of trust relationships between neighboring nodes" [12].

Hybrid trust management (HTM) schemes (e.g. [31, 32]) contain the properties of both centralized as well as distributed trust management approaches. The main objective of this approach is to reduce the cost associated with trust evaluation as compared to distributed approaches. This scheme is used with clustering schemes, in which cluster-head acts as a central server for the whole cluster. This approach is more reliable than the centralized one but less reliable than the distributed one. Each node needs to maintain the record of only member nodes, which gives the benefit of less memory consumption than the distributed approach. For intra-cluster communication, nodes need to contact the cluster head. It introduces more communication overhead in the network as compared to the distributed one.

The advantages and disadvantages of all three approaches are summarized in Table 2.1. All these three trust management approaches are further classified into two categories [33]: certificate-based trust management approach and behavior-based trust management approach. In the certificate-based trust management approach, trust is mainly based on the provision of a valid certificate assigned to a target node by a centralized certification authority or by other trusted issuer. In the behavior-based trust management approach, an entity calculates the trust values by continuous direct or indirect monitoring

of other nodes.

Table 2.2 gives the classification of proposed trust management schemes of wireless sensor networks based on our proposed trust taxonomy. These schemes are discussed in more comprehensive manner in next section.

2.2.2 State-of-the-art Research

Research on trust management schemes for wireless sensor networks is in its infancy state. Few schemes have been proposed that are discussed below in chronological order.

RFSN: Ganeriwal et al. [12, 34] have proposed Reputation based framework for sensor network (RFSN), where each sensor node maintains the reputation for neighboring nodes. On the basis of that reputation trust values are calculated. The RFSN scheme follows the localized distributed approach and borrows some design features from several existing works in the literature. It uses Bayesian formulation for representing reputation of a node. The RFSN scheme assumes that the node would have enough interactions with the neighbors so that the reputation (beta distribution) can reach to a stationary state. If the mobility is at a higher rate, reputation information will not stabilize and it may degrade its performance. Therefore, this kind of architecture is most suitable for stationary networks as compared to the mobile networks. In the RFSN scheme, nodes are classified into two categories: cooperative and not cooperative. Trust formulation approach of RFSN scheme can not cope with uncertainty situations [17]. Also, in their scheme no node is allowed to disseminate bad reputation information. It is resilient against badmouthing [35] and ballot stuffing attacks [12] but at the cost of system efficiency, as nodes cannot share bad experiences with each other.

ATRM: Boukerch et al. [36, 14] have proposed an Agent based Trust and Reputation Management (ATRM) scheme for wireless sensor networks. ATRM is based on

Table 2.1: Advantages and disadvantages of trust management approaches

Centralized	• Least computational overhead. • Least memory usage.	• Least reliable (single point of failure). • Most communication overhead.
Distributed	• Most Reliable (no single point of failure). • Scalable.	• Most computational overhead. • Most memory usage.
Hybrid	• Less communication overhead than centralized. • Less memory consumption than distributed. • Less computational overhead than distributed. • More reliable and scalable than centralized.	• Large computational overhead then centralized. • Large memory requirement than centralized. • Less scalable and reliable than distributed.

Table 2.2: Application of trust taxonomy

		Certificate-based	Behavior-based
Centralized		-	-
Hybrid		-	GTMS [31]
		Aivaloglou et al. [33]	
Distributed	Fully	ATRM [14]	-
	Localized	-	PLUS [13], RFSN [12], T-RGR [16]

a clustered wireless sensor networks and calculates trust in a fully distributed manner. Every sensor node holds a local mobile agent that is responsible for administrating trust and reputation of hosting node. ATRM assumes that there is a trusted authority which is responsible for generating and launching mobile agents. It also assumes that mobile agents are resilient against malicious nodes that try to steal or modify information carried by the agent. We feel that in many applications this assumption is not realistic. The major advantage of the ATRM scheme is that they use mobile agents for trust calculation which reduces the bandwidth consumption and time delay.

The ATRM scheme work in two phases: 1) Network Initialization phase and 2) Service offering phase. In the first phase, the Agent Launcher (AL) distributes the mobile agents called Trust and Reputation Assessor (TRA) to each node. As long as node has local TRA, it is in service offering phase, in which it is ready to provide trust and reputation management services. This phase is composed of four sub-services: r-certificate acquisition, t-instrument issuance, r-certificate issuance, and trust management routine.

- The r-certificate acquisition is pre-transaction service whose objective is to find out the reputation value of the other node. This will be performed by the exchange of certificate request (CertReq) and reply (CertRep) messages. At the end of this

service node will decide whether it should start transaction or not.

- The t-instrument issuance is a post transaction service whose objective is to evaluate trust value based on the recent context. This will be performed by the exchange of t-Instrument issuance (InstrIssument) and acknowledgment (ACK) messages.

- The r-Certificate Issuance service is executed periodically by replica TRAs based on the t-Instruments of their hosts. Since t- t-Instruments are context-specific, therefore in this process single reputation value is calculated based on all context's value.

- The trust management routine is also periodically carried out by every replica TRA to maintain the evaluation table on its hosting node. In each run, this routine will eliminate the any record from the table that is older then specific threshold time.

PLUS: Yao et al. [13] have proposed Parameterized and Localized trUst management Scheme (PLUS) for sensor networks security. The authors adopt a localized distributed approach and trust is calculated based on either direct observations or indirect observations. Trust calculation mechanism involves the combination of six parameters: 1) Ordering, 2) Integrity checking, 3) Confidentiality checking, 4) Responsibility checking, 5) Positivity checking and 5) Cooperative checking. Involvement of so many parameters makes this scheme less generic and complex. For example in 'ordering', node checks whether the packet forwarded by node i is really coming from the base station or not. For this purpose, they assume that all the important control packets generated by the base station must contain hashed sequence number (HSN). Based on that HSN it performs checking. If the check is passed then the trust value of the forwarding node will increase. Involvement of the HSN in control packets introduces two problems: 1) it increases the size of the packet that results in higher consumption of the transmission and

reception power, 2) it increases the computational cost at the sensor node because sensor node needs to verify the control packet that contains the HSN. Also, in 'Positivity' checking case, judge node monitors the suspected node i whether the node has participated in the exchange of opinions as well as whether it has sent report measurement to the base station with an appropriate frequency. This parameter forces the sensor nodes to remain in promiscuous mode all the time, which results in large energy consumption. All the six parameters are multiplied with different weight values. The mechanism for deriving those weight values is not defined in their paper. In the PLUS scheme, node is classified into four categories: 1) Distrust (untrustworthy), 2) Minimal (low trust), 3) Average (common trustworthy), and 4) Good (trustworthy). However the mechanism of computing boundaries of four trust levels is missing.

T-RGR: Liu et al.[16] have proposed a very simple trust management scheme for Resilient Geographic Routing (T-RGR) scheme. Their trust algorithm works in a localized distributed manner, in which each node monitors the behavior of the one-hop neighbors. If neighboring node successfully forwards the packet it will increase the trust value by a constant parameter, δt, and if it drops the packet then the source node will decrease its trust value by another constant parameter, Δt. If the trust value of a particular node is greater than the predefined threshold value, then the node will be considered as a trusted node, otherwise it will be un-trusted. In their paper, the authors do not mention the mechanism to calculate those three constant parameters that make their scheme non-adaptive. The main advantage of their scheme is that it is not only simple and easy to implement but it also consume less memory and energy. The main problem in their scheme is that each node only relies on its direct monitoring for the calculation of a trust value. This makes their scheme vulnerable to collaborative attacks.

FTSN: Aivaloglou et al. [33] have proposed Flexible Trust establishment Framework

for Sensor Networks (FTSN[1]) but it is still in initial phases. The unique thing about the FTSN is that it combines the features of certificate-based and behavior-based trust establishment approaches. Some subset of nodes in the network perform certificate-based trust evaluation and some subset of nodes, called supervision nodes in the network, perform behavior-based trust evaluation. A certificate validation is performed locally and is distributed before the deployment of the sensor nodes in the field. These certificates are signed by offline trust management authorities. Since this scheme is based on pre-deployment knowledge, so it is suitable for static sensor network environment. Nodes are either classified into trusted or un-trusted. Support of un-certain evidence is not available in this framework.

Table 6 gives qualitative comparison of the proposed schemes based on number of different parameters as discussed below:

- Trust-based on direct observations: represents the trust value that is calculated based on the personal interaction experience with other nodes and/or via monitoring of nodes which reside inside its radio range.

- Trust-based on indirect observations: represents the value that is obtained from the recommendations of the peer nodes.

- Trust levels: Depending on the scope and functionality, various trust management schemes provide support for different trust levels. Minimum, we can classify the nodes into two categories: trusted and un-trusted.

- Dependency on routing scheme: There are various routing schemes that have been proposed for wireless sensor networks. If a proposed trust management scheme is

[1]This is our defined term.

Table 2.3: Comparative features of trust management schemes

	RFSN	ATRM	PLUS	T-RGR	FTSN
Trust-based on direct observations	Yes	Yes	Yes	Yes	Yes
Trust-based on indirect observations	Yes	No	Yes	No	Yes
Trust levels	2	-	4	2	2
Dependency on routing scheme	Any	Any clustered based RS	PLUS_R	Any geographic based RS	Any

independent of any specific routing strategy then that scheme is considered to be a generic scheme.

2.3 Privacy

2.3.1 Taxonomy of privacy

Privacy generally refers to "ability to control the dissemination of information about oneself" [37]. In the wireless sensor network domain, so far privacy is mainly focused from anonymity [38, 19, 8] and/or secrecy perspective [39, 10, 11, 40]. However, only these two dimensions are not capable of providing complete privacy. In real life, we observe that complete privacy is gained through three independent but interrelated ways: *anonymity*: when an individual's true identity remains unidentified; *secrecy*: when an individual or a group's information remains protected from disclosure; and *solitude*: when one needs a temporal isolation in which an individual can not serve any request [41].

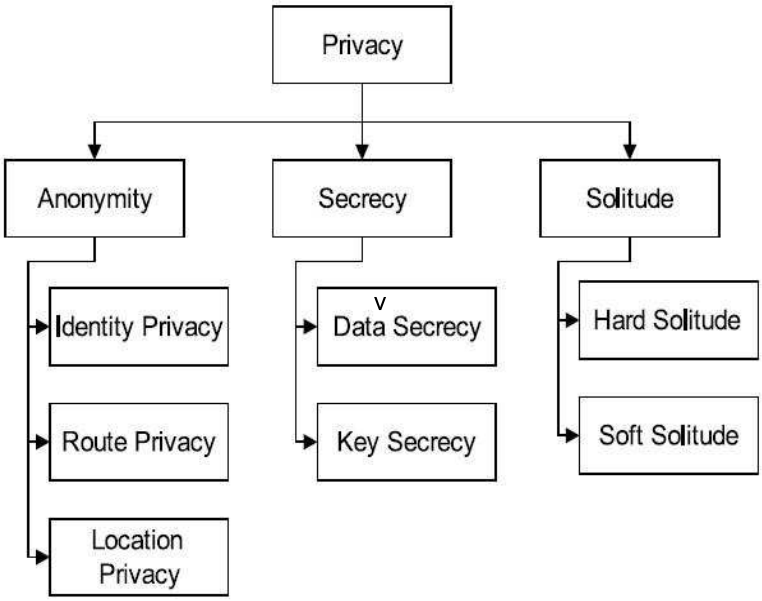


Figure 2.3: Taxonomy of privacy

Therefore, in order to achieve full privacy, we need to ensure that all these aspects: anonymity, secrecy, and solitude should be addressed. These three elements are further divided into sub categories as shown in Figure 2.3.

Anonymity provides three types of privacy protections, identity privacy, route privacy and location privacy [42].

- Identity privacy: no node can get any information about the source and destination nodes. Only the source and destination nodes can identify each other. Also, the source and destination nodes have no information about the real identities of the intermediate forwarding nodes.

- Route privacy: no node can predict the information about the complete path (from source to destination) of the packet. Also, a mobile adversary can not get any clue to trace back the source node either from the contents and/or directional informa-

tion of the captured packet(s).

- Location Privacy: no node can get to know any information about the location (either in terms of physical distance or number of hops) of the sender node except the source, its immediate neighbors and the destination.

Secrecy generally refers to the practice of hiding some information. Information is classified into two categories: One is the secrecy of actual sensed data forwarded by a sensor node to the specific destination and the other is key secrecy that is required to cipher data.

Solitude refers to the condition that a node goes into the state of isolation for a specific period of time. During that interval the node cannot fulfill jobs or is not able to provide services to the other nodes such as packet forwarding etc. We have categorized solitude into two types. Soft solitude means that a node goes into the solitude state by its own wish. Hard solitude means that a node is forced into the state of isolation.

Table 2.4 gives the classification of proposed privacy schemes (e.g. SAS & CAS [19], PFR [8], PSR [18], SIGF [20], CEM [21], GROW [6] of wireless sensor networks based on our proposed taxonomy. These schemes are discussed comprehensively in next section.

2.3.2 State-of-the-art Research

Current research so far sees privacy either from secrecy perspective or from anonymity perspective. As we mentioned earlier, full privacy consist of three elements, secrecy, anonymity and solitude. Unfortunately, no solution, in the wireless sensor network domain, can guarantee the triumph of all these three elements in a single solution. In this section, we presents the critical analysis of current state-of-the-art research work done

Table 2.4: Application of privacy taxonomy

		SAS & CAS	PFR	PSR	SIGF	CEM	GROW
Anonymity	Identity	Yes	No	No	No	No	No
	Route	Depending on routing scheme	Yes	Yes	Yes	Depending on routing scheme	Yes
	Location	No	Yes	Yes	Yes	Yes	Yes
Secrecy	Data	Yes	NA	NA	Yes	NA	NA
	Key	Yes	NA	NA	Yes	NA	NA
Solitude	Soft	No	No	No	No	No	No
	Hard	No	No	No	No	No	No

so far in the field of privacy in wireless sensor networks.

2.3.2.1 Anonymity Schemes

In wireless sensor network domain, some applications demand anonymity, for example, a panda-hunter application [8]; in which Save-The-Panda organization has deployed the sensor nodes to observe the vast habitat for pandas. Whenever any sensor node detect some panda it will make observations e.g. activity, location etc and periodically forward those to the sink node via some multi-path routing strategy. In this scenario, hunter can try to capture the pandas by back-tracing the routing path until it reaches the source sensor nodes. Therefore, in order to prevent hunter from back-tracing, the route and location anonymity mechanisms must be enforced. Similarly, in a battlefield application scenario, "the location of a soldier should not be exposed if he initiates a broadcast query" [6].

Traditionally, number of various anonymity schemes have been proposed such as

DC-Network [43], Crowds [44], Onion Routing [45], Hordes [46], ARM [47] etc. Most common approach used in these schemes is the employment of cover traffic. Cover traffic represents the dummy packets that are transmitted along with the original packets to the different destinations. Beside with the cover traffic some schemes uses pseudonyms for assigning identities to the nodes. The objective of using cover traffic is to make attacker clueless about the original packet and its destination. This kind of approach is even not suitable for traditional wired networks that cause large amount of traffic overhead. Also these schemes need high computational cost that is required for encryption and decryption of not only of original packets but also of dummy packets too. These common problems make traditional anonymity schemes unsuitable for wireless sensor networks that operate in highly resource constraints environment.

PFR: Ozturk et. al. [8] proposed phantom routing scheme for the wireless sensor networks which helps in preventing the location of a source node from the attacker. In this scheme each message reaches at the destination in two phases: 1) walking phase in which message is unicasted in random fashion within first h_{walk} hops. 2) After that message is flooded using the baseline flooding technique. In the first phase, authors have introduced a bias in random selection that makes it directed walk from pure random walk. The purpose of this approach is to minimize the chances of creating routing loops. However, this approach may incur delays. For example, because of directed walk, message may always move away to the base station. Thus, this approach is suitable for the applications that are not much time sensitive. The main advantage of this scheme is that the source location privacy protection improves as the network size and intensity increases because of the high path diversity. On the other hand, if the network size increases, the flooding phase consumes more energy, which in turn reduces life time of the network.

PSR: Kamat et al. [18] proposed Phantom Single-path Routing (PSR) scheme that works in the similar fashion as original phantom routing scheme [8]. They refer earlier one phantom-flood routing (PFR) scheme. The major difference between these two schemes is that after walking phase the packet will be forwarded to the destination via single path routing strategy such as shortest path routing mechanism. This scheme consumes less energy and requires slightly higher memory as compared to the phantom-flood routing scheme. The major limitation of this scheme is that it only provides protection against weaker adversary model.

SAS & CAS: Misra and Xue [19] proposed two schemes for establishing anonymity in the clustered wireless sensor networks. One is called Simple Anonymity Scheme (SAS) and other is called Cryptographic Anonymity Scheme (CAS). Both schemes are based on various assumptions such as sensor nodes are similar, immobile, consist of unique identities, and share pair wise symmetric keys. The SAS scheme uses dynamic pseudonyms instead of a true identity during communications. Each sensor node needs to store a given range of pseudonyms that are non-contiguous. Therefore, the SAS scheme is memory inefficient. On the other hand, the CAS scheme uses keyed hash functions to generate pseudonyms. That makes it memory efficient as compared to the SAS but it requires more computation power.

SIGF: Wood et al. [20] have proposed a configurable secure routing protocol family called Secure Implicit Geographic Forwarding (SIGF) for wireless sensor networks. The SIGF scheme is based on Implicit Geographic Forwarding (IGF) protocol [48], in which, a packet is forwarded to the node that lie within the region of 60°sextant, centered on the direct line from the source to the destination. This approach reduces the path diversity because of which only limited route anonymity is achieved. If we increase the forwarding area from 60° sextant up to 360° then in that case the objective of using

the IGF protocol will be lost. The SIGF protocol is mainly proposed by keeping security in mind. That is why some of the privacy aspects have not been covered such as identity privacy. Also, this protocol is unable to provide data secrecy in the presence of identity anonymity. Another, drawback of this protocol is that, when there is no trusted node within a forwarding area, it will forward packet to the un-trusted node. So, the reliability of a path is affected.

GROW: Xi et al. [6] proposed a Greedy Random Walk (GROW) scheme for preserving location of the source node. This scheme works in two phases. In a first phase, the sink node will set up a path through random walk with a node that acts as a receptor. Then the source node will forward the packet(s) towards the receptor in a random walk manner. Once the packet(s) reaches at the receptor, it will forward the packet(s) to the sink node through the pre-established path. Here receptor is acting a central point between the sink and the source node for every communication session. A criterion of selecting a trustworthy receptor is essential that is not defined.

CEM: Ouyang et al. [21] proposed a Cyclic Entrapment Method (CEM) to minimize the chance of an adversary to find out the location of the source node. In the CEM, when the message is sent by the source node to the base station, it activates the predefined loop(s) along the path. An activation node will generate the fake message and forward it towards the loop and original message is forwarded to the base station via specific routing protocol such as shortest path. Energy consumption in the CEM scheme is mainly dependent upon the number of loops in the path and their size.

Table 2.5 gives the summary of proposed privacy preserving schemes e.g. PFR [8], PSR [18], SAS & CAS [19], SIGF [20], CEM [21], and GROW [6].

Table 2.5: Summary of privacy preserving schemes of WSNs

	PFR	PSR	SAS & CAS	SIGF	CEM	GROW
Required information for routing	ID of destination	Routing table (e.g. Destination ID, # of hops etc.)	Depending on a routing scheme	Own, destination, & neighborhood locations	Depending on a routing scheme	Routing table (e.g. Destination ID, Receptor ID etc.)
Transmission Mechanism	1^{st} phase: Point-to-point; 2^{nd} phase: Broadcast	Point-to-point	Depending on a routing scheme	Point-to-point	Point-to-point	Point-to-point
Decision place for forwarding	1^{st} phase: Transmitter; 2^{nd} phase: Receiver	Transmitter	Depending on a routing scheme	Transmitter	Transmitter	Transmitter
Criteria for forwarding packet to next hop	1^{st} phase: random; 2^{nd} phase: flooding	1^{st} phase: random; 2^{nd} phase: shortest in terms of hops	Depending on a routing scheme	Randomly select any trusted node lies in forwarding region	Depending on a routing scheme	1^{st} phase: random; 2^{nd} phase: Pre-defined path

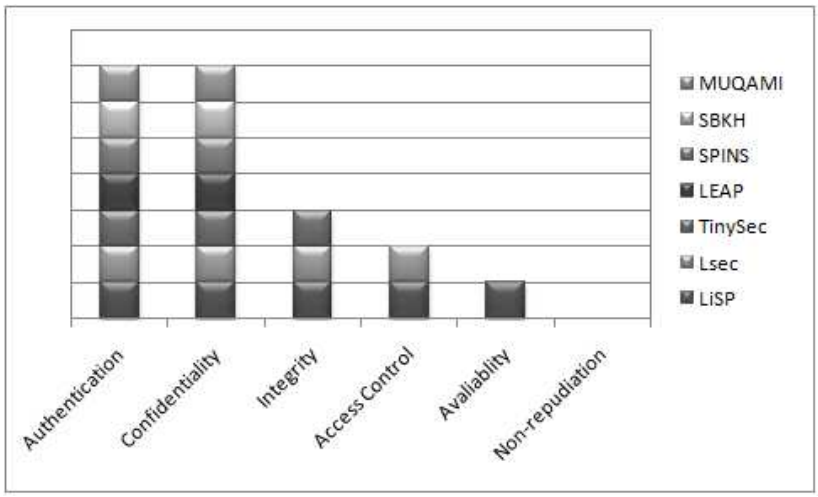


Figure 2.4: Comparison of security protocols

2.3.2.2 Secrecy

Secrecy is generally used to hide the contents of the message from unauthorized access, but it is not used to hide the source and destination identity. Overall secrecy is achieved through the combination of different security services such as authentication and confidentiality. Additionally, these security services highly rely on a secure key exchange mechanism. Quite recently many security solutions have been proposed such as SPINS [39], LEAP [11], TinySec [49], LiSP [40], SBKH [50], MUQAMI [51] etc., which provide various security services such as authentication, confidentiality, message integrity etc. High level qualitative comparison of these schemes is shown in Figure 2.4. This figure illustrate that the authentication, confidentiality, and integrity are well accommodated. However others (access control, availability, and non-repudiation) are not.

SPINS: Perrig et al., [39] have proposed security protocols suite called SPINS for wireless sensor networks. SPINS consist of two building blocks SNEP and uTESLA. SNEP provides data confidentiality, two party data authentication and data freshness

where as uTESLA provides authenticated broadcast for severely resource constraint environment. For data confidentiality they use symmetric encryption mechanism in which secret key called master key is shared between sensor node and base station. SNEP uses one time encryption key that produces from the unique master key. SNEP uses MAC function for two party authentications and checking data integrity. SPINS is based on binary security model, which means, either it provides maximum security or it does not provide any security. Usage of source routing scheme in SPINS makes the network vulnerable to traffic analysis [52].

LEAP: Zhu et al. [11] have proposed security mechanisms: Localized Encryption and Authentication protocol (LEAP), and a key management protocol for large scale distributed wireless sensor networks. In order to meet different security requirements LEAP provides the support of four types of keys for each sensor node: 1) each node shared a unique secret key with base station, 2) pairwise key shared between each pair of neighboring nodes, 3) cluster key shared with multiple neighboring nodes, and 4) a group key that is shared by all the nodes in the network. If a node has d neighbors, it needs to store one individual key, d pairwise keys, d cluster keys and one group key. Authors have employed uTESLA [39] protocol for broadcast authentication. However, in order to add more security such as inter-node authentication, authors have used hop-by-hop authentication strategy in which each node must authenticate the packet before forwarding it to the next hop. For this purpose, each node need to store one-way key chain of length L, and most recent authenticated key of each neighbor. Therefore, each node need to store total $3d + 2 + L$ keys.

TinySec: Karlof et al. [49] have proposed TinySec architecture for wireless sensor networks. TinySec is the first fully implemented link layer cryptography-based security protocol that provides authentication, integrity and confidentiality by adding less than

10% of energy, latency and bandwidth overhead. TinySec architecture comprises of two modes: 1) Authenticated encryption (TinySec-AE) mode, in which TinySec encrypts the payload (data) and authenticate the packet with a MAC. 2) Authentication only (TinySec-AH) mode, in which TinySec authenticates the entire packet with the MAC. TinySec protocol is tightly coupled with Berkeley TinyOS and can not be use for general sensor network model [53].

LiSP: Park and Shin [40] have proposed Lightweight Security protocol (LiSP) that's makes a trade off between security and resource consumption through efficient re-keying mechanism. This re-keying mechanism has number of features such as: efficient key broadcasting, which does not require any retransmissions or acknowledgments; implicit authentication of new keys without incurring any additional overhead; seamless key refreshments; detection and recovery of lost keys. LiSP protocol does not have any control packets or any type of retransmission that makes it energy efficient and secure against DoS attacks. LiSP achieves authentication, confidentiality, data integrity, access control and availability. In LiSP, each node need to save atleast eight keys therefore it is memory efficient. Also, computation cost of LiSP is very low because on average it needs to compute less then three hash computation.

SBKH: Michell and Srinivasan [50] have proposed lightweight security protocol called State Based Key Hop (SBKH) for low power devices such as sensor nodes. SBKH achieves authentication, confidentiality and integrity. In this protocol two communicating nodes share common knowledge about RC-4 states. These states are used to generate cipher streams. These states remain same for the pre-defined duration known only to two communicating nodes and will reinitialized only when base key changes. This approach gives the benefit of providing less computation overhead as compared to the traditional WEP and WPA 1.0 security solutions where RC-4 states are reinitialized for

every packet. However, security strength of this scheme is mainly depended on a stronger key management and distribution scheme.

MUQAMI: Raazi et al. [51], have proposed a key management scheme for clustered sensor networks called MUQAMI. In MUQAMI, responsibility of key management is divided among a small fraction of nodes within a cluster. Also, during the normal network operation, this responsibility can be transferred from one node to another with minimal overhead. This eradicates any single point of failure in the network. Also, this scheme is highly scalable and it eradicates all the inter-cluster communication. Lastly, it does not require all nodes to participate in key management, which reduces the security overhead substantially. This scheme is mainly designed for large-scale sensor networks. This scheme is more susceptible to collusion attacks [54] than other schemes such as LEAP+ [55]. Its parameters should be chosen carefully in order to avoid collusion attacks.

2.3.2.3 Solitude

As we mentioned earlier, so far the concept of solitude is not used for achieving privacy in the wireless sensor networks. The concept of solitude could be applied in different ways. For example, soft solitude is achieved whenever any node does not want to participate in communication due to any reason such as to preserve energy etc, then that node will broadcast message to all its neighboring nodes. That message contains the information like do not send packets to requester till specific period of time Δt. Once this message is received by the member nodes they will not consider that node for the purpose of forwarding a packet and virtually consider that node as an un-trusted node. After passage of time interval Δt, node's state will reset to original (trusted or un-trusted) state. In order to provide protection against spoofing, receiving node will first perform

Angle of Arrival (AoA) and single strength check [56], which will ensure that the packet was sent by the legitimate source node. Many other AoA based localization techniques have been specifically proposed for sensor networks such as [57, 58]. Any one of them could be used. Pseudo code of a Soft Solitude Algorithm (SSA) is given in 2.3.2.3.

Algorithm 1 SSA

1: Receive Packet Pkt;
2: Get NID = GetNodeID(Pkt);
3: **if** $checkAoA(Pkt) = true$ **then**
4: Set timer Δt;
5: Set $state = NID_{state}$;
6: **while** $\Delta t = true$ **do**
7: $NID_{state}\ remain\ untrusted$;
8: **end while**
9: $NID_{state} = state$;
10: **else**
11: Detect $spoof_{pkt}$.
12: **end if**

Hard Solitude could also be achieved with the help of trust values. If any node is considered to be un-trusted based on its trust value, that node will not be able to participate in a communication for a given period of time. For example, some intrusion detection techniques [59, 60] proposed for ad-hoc networks have the ability to gradually isolate the node(s) in case the node(s) are found to be malicious or un-trusted. However, those schemes require continuous monitoring and collection of information about intrusions at various places that increases overhead, and make them unsuitable for WSNs [61].

2.4 Summary

Current research so far focuses on the cryptographic-based security issues of WSNs. Although many survey papers are available in the security domain of WSNs, but we did not find any work in the literature which discusses the privacy and trust issues of WSNs in detail. In this chapter, I have given critical analysis of the current state-of-the-art research work done so far in the field of privacy and trust of WSN domain. This chapter also presents generic and flexible taxonomies of privacy and trust that are based on my own research experience with WSNs.

Chapter 3
Group-based Trust Management Component

3.1 Introduction

Trust in general is the level of confidence in a person or a thing. Various engineering models such as security, usability, reliability, availability, safety, and privacy models incorporate some limited aspects of trust with different meanings [62]. For example, in sensor network security, trust is a level of assurance about a key's authenticity that would be provided by some centralized trusted body to the sensor node [63, 64]. In wireless ad hoc and sensor network reliability, trust is used as a measure of node's competence in providing required service [65, 66, 31, 15]. In general, establishing trust in a network gives many benefits such as:

1. Trust solves the problem of providing corresponding access control based on judging the quality of sensor nodes and their services. This problem can not be solved through traditional security mechanisms [1].

2. Trust solves the problem of providing reliable routing paths that do not contain any malicious, selfish or faulty node(s) [2, 3].

3. Trust makes the traditional security services more robust and reliable by ensuring that all the communicating nodes are trusted during authentication, authorization

or key management [67].

For Wireless Sensor Networks (WSNs), we visualize that trust management is a co-operative business rather than an individual task due to the use of clustering schemes such as LEACH [68], PEGASIS [69], TEEN [70], and HEED [71] in real world scenarios. Moreover, sensor nodes can also be deployed in the form of groups [72] which are willing to collaborate with each other in order to process, aggregate and forward collected data [73]. This highlights the fact that these clustering schemes and group deployments enable sensor nodes to fulfill their responsibilities in a cooperative manner rather than individually. Therefore, establishing and managing trust in a cooperative manner in clustering environment provides many advantages. Such as, within the cluster, it helps in the selection of trusted cluster head by the member nodes. Similarly, the cluster head will be able to detect faulty or malicious node(s). In case of multi-hop clustering [71, 74], it helps to select trusted en-route nodes through which a node can send data to the cluster head. During inter-cluster communication, trust management helps to select trusted en-route gateway nodes or other trusted cluster heads through which sender node will forward data to the base station.

A number of trust management schemes have been proposed for peer-to-peer networks [75, 76, 77], and ad-hoc networks [30, 66, 78]. To the best of our knowledge, very few comprehensive trust management schemes (e.g. RFSN [12], ATRM [14] and PLUS [13]) have been proposed for sensor networks. Although, there are some other works available in the literature e.g. [31, 15, 16, 17], etc., that discuss trust but not in much detail. Within such comprehensive works, only ATRM [14] scheme is specifically developed for the clustered WSNs. However, this and other schemes, suffer from various limitations such as these schemes do not meet the resource constraint requirements of the WSNs; and more specifically, for the large-scale WSNs. Also, these schemes suffer

from higher cost associated with trust evaluation specially of distant nodes. Furthermore, existing schemes have some other limitations such as dependence on specific routing scheme, like the PLUS scheme works on the top of the PLUS_R routing scheme; dependence on specific platform, like the ATRM scheme requires an agent-based platform; and unrealistic assumptions, like the ATRM assumes that agents are resilient against any security threats, etc. Therefore, these works are not well suited for realistic WSN applications. Thus, a lightweight secure trust management scheme is needed to address these issues.

In this work, a new lightweight Group-based Trust Management Scheme (GTMS) is proposed for clustered WSNs. The GTMS consists of three unique features such as:

- GTMS evaluates the trust of a group of sensor nodes in contrast to traditional trust management schemes that always focus on trust values of individual nodes. This approach gives us the benefit of requiring less memory to store trust records at each sensor node in the network.

- GTMS works on two topologies: intra-group topology where distributed trust management approach is used and inter-group topology where centralized trust management approach is adopted. This methodology helps to drastically reduce the cost associated with trust evaluation of distant nodes.

- GTMS not only provides a mechanism to detect malicious nodes, but also provides some degree of prevention mechanism.

These and other specific features (e.g., independent of any specific routing scheme and platform etc.) collectively make the GTMS a new, lightweight, flexible, and robust solution that can be used in any clustered WSNs.

The rest of the chapter is organized as follows. Section 3.2 contains definitions, description on representation of trust value and assumptions. Section 3.3 proposes trust modeling and evaluation mechanism of the GTMS scheme. Section 3.4 and 3.5 provide theoretical and simulation-based analysis and evaluation of the GTMS scheme respectively. Section 3.6 concludes the chapter.

3.2 Definitions, Representation, and Assumptions

3.2.1 Definitions

The Group-based Trust Management Scheme (GTMS) calculates the trust value based on direct or indirect observations. Direct observations represent the number of successful and unsuccessful interactions and indirect observations represent the recommendations of trusted peers about a specific node. Here, interaction means the cooperation of two nodes. For example, a sender will consider an interaction as successful if the sender receives an assurance that the packet is successfully received by the neighbor node and that node has forwarded the packet towards the destination in an unaltered fashion.

- The first requirement, i.e., successful reception, is achieved on reception of the link layer acknowledgment (ACK). IEEE 802.11 is a standard link layer protocol which keeps packets in its cache until the sender receives an ACK. Whenever the receiver node successfully received the packet, it will send back an ACK to the sender. If the sender node did not receive the ACK during a predefined threshold time then it will retransmit that packet.

- The second requirement, i.e., forwarding of the packet, is achieved by using enhanced passive acknowledgment (PACK) by overhearing the transmission of a next

hop on the route, since they are within the radio range [3].

If the sender node does not overhear the retransmission of the packet within a threshold time from its neighboring node or the overheard packet is found to be illegally fabricated (by comparing the payload that is attached to the packet), then the sender node will consider that interaction as an unsuccessful one. For example, black hole attack is straight forwardly detected when malicious node drops the incoming packets and keeps sending self-generated packets [79]). Similarly, sink hole attack [80] that is advance version of the black hole attack is also easily detectable with the help of looking at the passive acknowledgment. Likewise, affects of selective forwarding attack [81] or gray-hole attack [82] could also be eliminate with the aid of above mentioned approach.

If the number of unsuccessful interactions increases, the sender node decreases the trust value of that neighboring node and may consider it as a faulty or malicious node.

3.2.2 Representation of trust value

Generally, a trust value is considered to be a numerical quantity lying between 0 to 1 (inclusive) as suggested earlier in [66, 30, 83] or between -1 to 1 (inclusive) as described in [65] on a real number line. In this work, i use trust value as an integer in the interval between 0 and 100 (inclusive). However other ranges, for example base 2 ranges, could be used as well. Although presenting the trust values as a real number or integer may not play an important role in traditional networks, but for sensor nodes (SNs) this issue is of critical importance due to limited memory, and transmission, reception power. This change will give us benefits such as: Representation of trust value [0,100] as an un-signed integer (1 byte) saves 75% of memory space as compared to trust values represented as a real number (4 bytes). Less number of bits need to be transmitted during the exchange of trust values between SNs. This gives us the benefit of less consumption

of transmission and reception power.

3.2.3 Assumptions

I assume that the sensor network consists of large number of SNs that are deployed in an open or hostile environment. I also assume that all SNs have unique identities as it is also assumed in [12, 14]. In some of the sensor network models, nodes do not have unique identities like IP in traditional networks. However, in order to uniquely identify the SNs and perform communication in those environments, class-based addressing scheme [84, 85, 86] is used, in which a node is identified by a triplet <location, node type, node subtype>. I also, assume that SNs are organized into clusters with the help of any proposed clustering scheme such as [68, 70]. These clustering schemes are used in a real world scenarios (e.g. habitat monitoring application, such as, James Reserve [87], Great Duck Island [7] etc.) for efficient network organization [88]. I also assume that the base station is a central command authority. It has no resource constraint problem, and furthermore it can not be compromised by an attacker. In order to provide protection of trust values from traffic analysis or fabrication during transfer from one node to another, I assume a secure communication channel, which can be established with the help of any key management scheme [9, 11, 89, 10].

3.3 Group-based Trust Management Scheme

The proposed trust model works with two topologies. One is the intra-group topology where distributed trust management is used. The other is inter-group topology where centralized trust management approach is employed. For the intra-group network, each sensor that is a member of the group, calculates individual trust values for all group

members. Based on the trust values, a node assigns one of the three possible states: 1) trusted, 2) un-trusted or 3) un-certain to other member nodes. This three-state solution is chosen for mathematical simplicity and is found to provide appropriate granularity to cover the situation. After that, each node forwards the trust state of all the group member nodes to the CH. Then, centralized trust management takes over. Based on the trust states of all group members, a CH detects the malicious node(s) and forwards a report to the base station. On request, each CH also sends trust values of other CHs to the base station. Once this information reaches the base station, it assigns one of the three possible states to the whole group. On request, the base station will forward the current state of a specific group to the CHs.

My group based trust model works in three phases: 1) Trust calculation at the node level, 2) Trust calculation at the cluster-head level, and 3) Trust calculation at the base station level.

3.3.1 Trust Calculation at the Node Level

At the node level, a trust value is calculated using either time-based past interaction or peer recommendations. Whenever a node x wants to communicate with node y, it first checks whether x has any past experience of communication with y during a specific time interval or not. If yes, then node x makes a decision based on past interaction experience, and if not, then node x moves for the peer recommendation method.

3.3.1.1 Time-based Past Interactions Evaluation

Trust calculation at each node measures the confidence in node reliability. Here the network traffic conditions such as congestion, delay etc., should not affect the trust attached to a node; this means that the trust calculation should not emphasize the timing

information of each interaction too rigidly. Therefore, I introduce a sliding time window concept which takes relative time into consideration and reduces the effects of network conditions on overall trust calculation. If real-time communication is a requirement, as is the case in most real-world applications, this timing window concept does not provide any hindrance when it comes to real-time delivery of packets. The communication protocol in such applications is always accompanied with time-stamps, and thus any node which delays the delivery of packets by taking advantage of the sliding timing window will be detected straightforwardly.

The timing window (Δt) is used to measure the number of successful and unsuccessful interactions. It consist of several time units. The interactions that occur in each time unit within the timing window are recorded. After a unit of time elapses, the window slides one time unit to the right, thereby dropping the interactions done during the first unit. Thus, as time progresses, the window forgets the experiences of one unit but adds the experiences of the newer time unit. The window length could be made shorter or longer based on network analysis scenarios. A sample scenario of the GTMS time window scheme is illustrated in Figure 3.1. The time window Δt consists of five units. During the first unit of Δt_1, the number of successful and unsuccessful interactions is 4 and 2 respectively, and during the whole Δt_1 interval, the number of successful and unsuccessful interactions is 29 and 15 respectively. After the passage of 1st unit, the new time interval Δt_2, drops the interaction values which took place during the very first unit of Δt_1 ($S = 4, U = 2$) and only consider the values of last 4 units of Δt_1 plus values of one recent unit added on the right ($S = 6, U = 2$).

With this time window information, the time-based past interaction trust value ($T_{x,y}$)

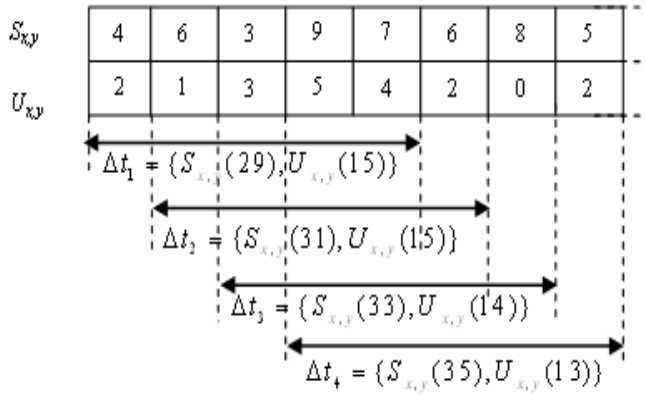


Figure 3.1: Sliding time window scheme of GTMS

of node y at node x that lies between 0 and 100, is defined as;

$$\begin{aligned} T_{x,y} &= \left[100\left(\frac{S_{x,y}}{S_{x,y}+U_{x,y}}\right)\left(1-\frac{1}{S_{x,y}+1}\right)\right] \\ &= \left[\frac{100(S_{x,y})^2}{(S_{x,y}+U_{x,y})(S_{x,y}+1)}\right] \end{aligned} \tag{3.1}$$

where [.] is the nearest integer function, $S_{x,y}$ is the total number of successful interactions of node x with y during time Δt, $U_{x,y}$ is the total number of unsuccessful interactions of node x with y during time Δt. The expression $\left(1-\frac{1}{S_{x,y}+1}\right)$ in the above approaches 1 rapidly with an increase in the number of successful interactions. I choose this function instead of a linear function since such a function would approach very slowly to 1 with the increase in successful interactions; hence it would take considerably longer time for a node to increase its trust value for another node. In order to balance this increase in the trust value with the increasing number of unsuccessful interactions, we multiply the expression with the factor $\left(\frac{S_{x,y}}{S_{x,y}+U_{x,y}}\right)$, which indicates the percentage of successful interactions among the total interactions. Thus, this equation has an inbuilt capability of diminishing the effects of a few wrong declarations of interactions that may be caused by any network traffic problems.

Figure 3.2 shows the behavior of time-based past interactions trust values against

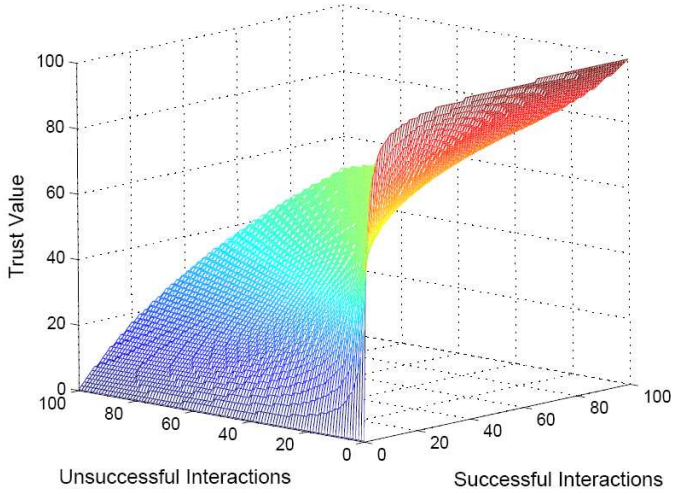


Figure 3.2: Time-based past interactions evaluation

successful and unsuccessful interactions. When we do not get even a single successful interaction, the trust value remains 0. With an increase in successful interactions, the trust value increases, but stays humble if the number of unsuccessful interactions is also considerably high. For example, with 60 unsuccessful and 50 successful interactions, the trust value is 45.

After calculating trust value, a node will quantize trust into three states as follows:

$$Mp(T_{x,y}) = \left\{ \begin{array}{ll} \text{trusted} & 100 - f \leq T_{x,y} \leq 100 \\ \text{uncertain} & 50 - g \leq T_{x,y} < 100 - f \\ \text{untrusted} & 0 \leq T_{x,y} < 50 - g \end{array} \right\} \tag{3.2}$$

where, f represents half of the average values of all trusted nodes and g represents one-third of the average values of all untrusted nodes. The usage of half and one-third of average values in evaluation directly affects the resiliency of a node, which is discussed

in section 3.4.1. Both f and g are calculated as follows:

$$f_{j+1} = \begin{cases} \left[\frac{1}{2}\left(\frac{\sum_{i \in R_x} T_{x,i}}{|R_x|}\right)\right] & 0 < |R_x| \leq n-1 \\ f_j & |R_x| = 0 \end{cases} \tag{3.3}$$

$$g_{j+1} = \begin{cases} \left[\frac{1}{3}\left(\frac{\sum_{i \in M_x} T_{x,i}}{|M_x|}\right)\right] & 0 < |M_x| \leq n-1 \\ g_j & |M_x| = 0 \end{cases} \tag{3.4}$$

where [.] is the nearest integer function, R_x represents the set of trustful nodes for node x, M_x the set of un-trustful nodes for node x, and n is the total number of nodes that contains trustful, un-trustful and uncertain nodes. At startup, the trust values of all nodes are 50 which is an uncertain state. Initially f and g are equal to 25 and 17 respectively, although other values could also be used by keeping the following constraint intact: $f_i - g_i \geq 1$, which is necessary for keeping the uncertain zone between a trusted and un-trusted zone. The values of f and g are adaptive. During the steady-state operation, these values can change with every passing unit of time which creates dynamic trust boundaries as shown in Figure 3.3. At any stage, when $|R_x|$ or $|M_x|$ becomes zero then the value of f_{j+1} or g_{j+1} remains the same as the previous values (f_j and g_j). The nodes whose values are above $100 - f$ will be declared as trustful nodes (Eq. 3.2), and nodes whose values are lower than $50 - g$ will be consider as untrusted nodes (Eq. 3.2). After each passage of time, Δt, nodes will recalculate the values of f and g. This trust calculation procedure will continue in this fashion.

3.3.1.2 Peer Recommendations Evaluation

Let a group be composed of n uniquely identified nodes. Furthermore, each node maintains a trust value for all other nodes. Whenever a node requires peer recommendation it will send a request to all member nodes except for the un-trusted ones. Let us assume

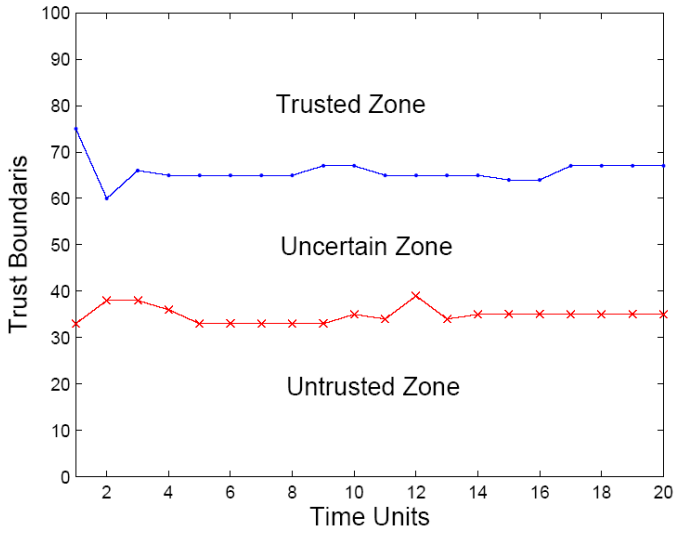


Figure 3.3: Adaptive trust boundaries creation

that j nodes are trusted or uncertain in a group. Then node x calculates the trust value of node y as follows:

$$T_{x,y} = \left[\frac{\sum_{i \in \mathrm{R}_x \cup \mathrm{C}_x} T_{x,i} * T_{i,y}}{100 * j} \right] ; j = |\mathrm{R}_x \cup \mathrm{C}_x| \leq n - 2 \tag{3.5}$$

where, [.] is the nearest integer function, $T_{x,i}$ is the trust value of the recommender, and $T_{i,y}$ is the trust value of node y sent by node i. Here, $T_{x,i}$ is acting as a weighted value of the recommender that is multiplied with the trust value $T_{i,y}$, sent by recommender, such that the trust value of node y should not increase beyond the trust value between node x and the recommender node i.

3.3.2 Trust Calculation at the Cluster-Head Level

Here I assume that the CH is the SN that has higher computational power and memory as compared to other SNs.

3.3.2.1 Trust State calculation of Own Group

In order to calculate the global trust value of nodes in a group, CH asks the nodes for their trust states of other members in the group. I use the trust states instead of the exact trust values due to two reasons. First, the communication overhead would be less as only a simple state is to be forwarded to the CH. Secondly, the trust boundaries of an individual node vary from other nodes. A particular trust value might be in a trusted zone for one node whereas it may only correspond to the uncertain zone for another node. Hence the calculation of the global trust state of nodes in a group would be more feasible and efficient if we only calculate it using the trust states.

Let us suppose there are $n + 1$ nodes in the group including the CH. The CH will periodically broadcast the request packet within the group. In response, all group member nodes forward their trust states, s, of other member nodes to the CH. The variable, s, can take three possible states: trusted, un-certain and un-trusted. The CH will maintain these trust states in a matrix form, as shown below:

$$TM_{ch} = \begin{bmatrix} s_{ch,1} & s_{1,ch} & \cdots & s_{n,1} \\ s_{ch,2} & s_{1,2} & \cdots & s_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ s_{ch,n} & s_{1,n} & \cdots & s_{n,n-1} \end{bmatrix}$$

where, TM_{ch} represents the trust state matrix of cluster-head ch and $s_{ch,1}$ represents the state of node 1 at cluster-head ch. The CH assigns a global trust state to a node based on the relative difference in trust states for that node. I emulate this relative difference through a standard normal distribution. Therefore, the CH will define a random variable

X such that:

$$X\left(s_{i,j}\right) = \begin{cases} 2 & \text{when} \quad s_{i,j} = \text{trusted} \\ 1 & \text{when} \quad s_{i,j} = \text{un-certain} \\ 0 & \text{when} \quad s_{i,j} = \text{un-trusted} \end{cases} \tag{3.6}$$

Assuming this to be a uniform random variable, I define the sum of m such random variables as S_m. The behavior of S_m will be that of a normal variable due to the central-limit theorem [90]. The expected value of this random variable is m and the standard deviation is $\sqrt{m/3}$. The CH defines the following standard normal random variable for a node j:

$$Z_j = \frac{\sqrt{3}\left(X\left(s_{ch,j}\right) + \sum_{i=1,i\neq j}^{m} X\left(s_{i,j}\right) - m\right)}{\sqrt{m}} \tag{3.7}$$

If $Z_j \in [-1, 1]$ then the node j is termed as un-certain, else if $Z_j > 1$, it is called trusted. If $Z_j < -1$, it is labeled as un-trusted.

3.3.2.2 Trust Calculation of Other Groups

During group-to-group communication, the CH maintains the record of past interactions of another group in the same manner as individual nodes keep record of other nodes. Trust values of a group is calculated on the basis of either past interaction or information passed on by the base station. Here I am not considering peer recommendations from other groups in order to save communication cost. Let us suppose CH i wants to calculate the trust value ($T_{i,j}$) of another cluster j. Then it can be calculated by using either time-based past interaction ($PI_{i,j}$) evaluation or by getting recommendation from the base station ($BR_{i,j}$) as shown below.

$$T_{i,j} = \left\{ \begin{array}{cc} \left[\frac{100(S_{i,j})^2}{(S_{i,j}+U_{i,j})(S_{i,j}+1)}\right] & \text{if } PI_{i,j} \neq \varphi \\ BR_{i,j} & \text{if } PI_{i,j} = \varphi \end{array} \right\} \tag{3.8}$$

If the cluster head does not have any record of past interactions within the time window, i.e., $PI_{i,j} = \varphi$, it requests the base station for the trust value.

3.3.3 Trust Calculation at Base Station Level

The base station (BS) also maintains the record of past interactions with CHs in the same manner as individual nodes do, as shown below.

$$T_{BS,ch_i} = \left[\frac{100\left(S_{BS,ch_i}\right)^2}{\left(S_{BS,ch_i} + U_{BS,ch_i}\right)\left(S_{BS,ch_i} + 1\right)} \right] \tag{3.9}$$

where, [.] is the nearest integer function, $S_{BS,ch}$ is the total number of successful interactions of BS with CH during time Δt, $U_{BS,ch}$ is the total number of unsuccessful interactions of BS with CH during time Δt.

Let us suppose there are $|G|$ groups in the network. BS periodically multicasts request packets to the CHs. On request, the CHs forward their trust vectors, related to the recommendations of other groups based upon past interactions, to BS as shown below: $\overrightarrow{T}_{ch} = (T_{ch,1}, T_{ch,2}, \ldots, T_{ch,|G|-1})$

On reception of trust vectors from all the CHs, the base station will calculate the trust value of each group in a manner shown below:

$$T_{BS,G_1} = \left[\frac{\sum_{i=1}^{|G|-1} \left(T_{BS,ch_i}\right)\left(T_{G_i,G_1}\right)}{|G|-1} \right], \ldots, T_{BS,G_m} = \left[\frac{\sum_{i=1}^{|G|-1} \left(T_{BS,ch_i}\right)\left(T_{G_i,G_{|G|}}\right)}{|G|-1} \right] \tag{3.10}$$

where, $T_{BS,ch}$ is the trust value of the CH i at the base station, $T_{Gi,G1}$ is the trust value of group G_1 at group G_i and $|G|$ represents the total number of groups in the network.

3.4 Theoretical Analysis and Evaluation

3.4.1 Security Resilience Analysis

In this section, the resiliency analysis of GTMS protocol against attacks on trust management is presented. Nodes are broadly categorize into two types: good ones and bad ones. Here, assumption is that good nodes interact successfully most of the time and submit true recommendations. On the other hand, bad nodes try to do as many unsuccessful interactions as possible and send false recommendations about good nodes. Clearly, this concept of good and bad nodes is relative. A node might be a good node in the view of one node whereas it might be bad for another. In the following, we define this concept more rigorously, capture the behavior of bad nodes and model how they might try to get unfair advantage in our trust model. Then we prove our protocol's resilience against such bad behaviors. This analysis can be applied straightaway to higher level groups in a modular way.

We begin with the notion of bad behavior and unfair advantage. Both these attributes define a malicious node. The goal of a malicious node while interacting with other nodes, is to do as many unsuccessful interactions as possible while keeping the following objectives intact:

- obtain a higher trust value for itself than the actual calculated trust value; more importantly, to get into the trusted zone when its rightful place is in the uncertain or un-trusted zone,
- decrease the trust value of a good node if possible,
- increase the trust value of a collaborating bad node if possible.

After defining a malicious node's objectives in this way, we can prove that our trust

management scheme at the node level is resilient against malicious nodes if it can stop the malicious nodes from fulfilling their objectives. Apparently, it is hard to come up with a scheme that can totally stop such behavior. However, if we can quantify the limits of such nodes, we can have a certain amount of assurance for our system. This assurance ensures that a *smart* node, which tries to minimize the number of successful interactions with other nodes while still being in the trusted zone, cannot accomplish its goals but within certain limits. More precisely, the *smart* node has to maintain the number of successful interactions higher or equal to the number of unsuccessful interactions, as will be explained in the following.

3.4.1.1 Resilience Analysis at Node level

In this section, we test the resilience of proposed trust model against malicious nodes. In what follows, we describe the interaction between nodes within a generic group G in the sensor network. Let R_i, C_i and M_i denote the set of trusted, uncertain and un-trusted nodes for a node i. We begin with a definition of a malicious node:

Definition 3.4.1. *A SN m is said to be bad for a node i if it has interacted with i at least once and $U_{j,m} \geq S_{j,m}$.*

Definition 3.4.2. *A bad node m for a node i is said to have deceived i if $s_{i,m} = trusted$.*

Definition 3.4.3. *A Trust Management Scheme is said to be resilient against deception by a bad node at the node level if no bad node can deceive another node.*

Claim 1: GTMS is resilient against deception by a bad node at the node level.

Proof. Suppose to the contrary that there exists a bad node m for a node i that successfully deceived i. Then according to the definition: $U_{i,m} \geq S_{i,m}$ and $s_{i,m} = trusted$. There are three cases:

Case 1: $S_{i,m} \geq 1$. This means that Node m has interacted with node i within the time window Δt. Let a denote the real number $U_{i,m}/S_{i,m}$. So, $a \geq 1$. Now since $s_{i,m} = trusted$, therefore at the time of the last interaction the trust calculation was done using the past interaction evaluation. Assume first that $R_i \neq \varphi$, Then:

$$100 - \frac{\sum_{k \in \mathrm{R}_i} T_{i,k}}{2|\mathrm{R}_i|} < T_{i,m}$$

Since i has previously interacted with node m within the time window in the past, we have:

$$T_{i,m} = 100 \left(\frac{S_{i,m}}{S_{i,m}+U_{i,m}}\right)\left(1 - \frac{1}{S_{i,m}+1}\right) = \frac{100}{a+1} - \frac{100}{(a+1)(S_{i,m}+1)}$$

This implies that:

$$\begin{aligned} 100 - \frac{\sum_{k \in \mathrm{R}_i} T_{i,k}}{2|\mathrm{R}_i|} &< \frac{100}{a+1} - \frac{100}{(a+1)(S_{i,m}+1)} \\ \Rightarrow 100\left(1 - \frac{1}{a+1} + \frac{1}{(a+1)(S_{i,m}+1)}\right) &< \frac{\sum_{k \in \mathrm{R}_i} T_{i,k}}{2|\mathrm{R}_i|} \leq \frac{100|\mathrm{R}_i|}{2|\mathrm{R}_i|} \end{aligned}$$

The last inequality is true since all the $T_{i,k}$'s are within the trusted zone. We get:

$$\frac{1}{2} < \frac{1}{a+1} - \frac{1}{(a+1)(S_{i,m}+1)}$$

Since $a \geq 1$, this gives us: $\frac{1}{(S_{i,m}+1)} < 0$, which is obviously impossible. If $R_i = \varphi$, then we have,

$$75 < T_{i,m} = \frac{100}{a+1} - \frac{100}{(a+1)(S_{i,m}+1)}$$

which again leads to the contradiction: $\frac{1}{(S_{i,m}+1)} < 0$.

Case 2: $S_{i,m} = 0$. We now consider $U_{i,m} \geq 1$. Let t_1 denote the first of these unsuccessful interactions within the time window Δt. For the 2^{nd} interaction request within the time window Δt, i must have calculated the trust value for m as:

$$T_{i,m} = 100 \left(\frac{S_{i,m}}{S_{i,m}+U_{i,m}}\right)\left(1 - \frac{1}{S_{i,m}+1}\right) = 100\left(\frac{0}{0+1}\right)\left(1 - \frac{1}{0+1}\right) = 0$$

But this is a contradiction, since the lower bound for the Trusted zone is always higher than 0. This proves the claim.

Case 3: $S_{i,m} = 0$, $U_{i,m} = 0$. This means that node m has no interaction with node i at all within the time window Δt. In that case, node m will rely on the recommendation of trusted peers. □

Definition 3.4.4. *A SN m is said to be really bad for a node i if it has interacted with i at least once and $U_{i,m} \geq 2S_{i,m}$.*

Definition 3.4.5. *A really bad node m for a node i is said to have deceived i if $s_{j,m} = uncertain$.*

Definition 3.4.6. *A Trust Management Scheme is said to be resilient against deception by a really bad node at the node level if no really bad node can deceive another node.*

Claim 2: GTMS is resilient against deception by a really bad node at the node level.

Proof. Suppose to the contrary that there exists a really bad node m for a node i that deceived i. Then according to the definition: $U_{i,m} \geq 2S_{i,m}$ and $s_{i,m} = uncertain$. We consider the three separate cases:

Case 1: $S_{i,m} \geq 1$. This means that Node m has interacted with node i within the time window Δt. Let a denote the real number $U_{i,m}/2S_{i,m}$. So, $a \geq 1$. Now since $s_{i,m} = uncertain$, therefore at the time of the last interaction the trust calculation was done using the past interaction evaluation. First, assume that $M_i \neq \varphi$, then:

$$50 - \frac{\sum_{k \in \mathrm{M}_i} T_{i,k}}{3|\mathrm{M}_i|} < T_{i,m}$$

Since i has previously interacted with node m within the time window in the past, we have:

$$T_{i,m} = 100 \left(\frac{S_{i,m}}{S_{i,m}+U_{i,m}} \right) \left(1 - \frac{1}{S_{i,m}+1} \right) = \frac{100}{2a+1} - \frac{100}{(2a+1)(S_{i,m}+1)}$$

This implies that:

$$50 - \frac{\sum_{k \in \mathrm{M}_i} T_{i,k}}{3|\mathrm{M}_i|} < \frac{100}{2a+1} - \frac{100}{(2a+1)(S_{i,m}+1)}$$

$$\Rightarrow 50\left(1 - \frac{2}{2a+1} + \frac{2}{(2a+1)(S_{i,m}+1)}\right) < \frac{\sum_{k \in \mathrm{M}_i} T_{i,k}}{3|\mathrm{M}_i|} \leq \frac{50|\mathrm{M}_i|}{3|\mathrm{M}_i|}$$

The last inequality is true since all the $T_{i,k}$'s are within the un-trusted zone. We get

$$\frac{1}{3} < \frac{1}{2a+1} - \frac{1}{(2a+1)(S_{i,m}+1)}$$

Since $a \geq 1$, this gives us: $\frac{1}{(S_{i,m}+1)} < 0$, which is again impossible. If $M_i = \varphi$, then we have,

$$\frac{100}{3} < T_{i,m} = \frac{100}{a+1} - \frac{100}{(a+1)(S_{i,m}+1)}$$

which again leads to the contradiction: $\frac{1}{(S_{i,m}+1)} < 0$.

Case 2: $S_{i,m} = 0$. We now consider $U_{i,m} \geq 1$. Let t_1 denote the first of these unsuccessful interactions within the time window Δt. For the second interaction request within the time window, i must have calculated the trust value for m as:

$$T_{i,m} = 100\left(\frac{S_{i,m}}{S_{i,m}+U_{i,m}}\right)\left(1 - \frac{1}{S_{i,m}+1}\right) = 100\left(\frac{0}{0+1}\right)\left(1 - \frac{1}{0+1}\right) = 0$$

But this is a contradiction, since the lower bound for the Uncertain zone is always higher than 0. This proves the claim. *Case 3:* $S_{i,m} = 0$, $U_{i,m} = 0$. Same as Case 3 of Claim 1. □

The above two claims are proved under the constraints that the trust value lies between 0 and 100. For a variable upper limit of trust value, the claims still hold. Let T_u be the variable denoting the upper limit of trust value. Notice that the formula for time based past interaction will change accordingly with the numeric value 100 replaced by T_u in Equation 3.1. Let us also give generic limits for the initial value of the function f as f_u, which in the above was fixed at 25, and for the initial value of uncertain zone as

R_u, which was previously fixed at 50. Assign a value of g_u to the initial value of g which is now fixed at 17. In both Claim 1 and Claim 2, cases 2 and 3 obviously still hold. For Case 1, it is not hard to see that the claims hold with certain restrictions on T_u, f_u, R_u and g_u. Let us first look at Case 1 of Claim 1: For $R_i \neq \varphi$, there are no constraints as T_u would cancel on both sides when replaced by the quantity 100 on both sides. For $R_i = \varphi$, we get:

$$T_u - f_u < T_u \left(\frac{1}{a+1} - \frac{1}{(a+1)(S_{i,m}+1)} \right) \Rightarrow 1 - \frac{f_u}{T_u} < \left(\frac{1}{a+1} - \frac{1}{(a+1)(S_{i,m}+1)} \right)$$

Carrying with the same argument as in the claim, we get that for the contradiction $\frac{1}{S_{i,m}+1} < 0$ to hold we should have that: $\frac{f_u}{T_u} < \frac{1}{2}$, i.e. $f_u < \frac{T_u}{2}$. In other words, f_u should be fixed at less than half the value of T_u.

Moving on to Case 1 of Claim 2, first suppose that $M_i \neq \varphi$. We have that:

$$R_u - T_u \left(\frac{1}{2a+1} - \frac{1}{(2a+1)(S_{i,m}+1)} \right) < \frac{R_u}{3}$$

Now for the contradiction $\frac{1}{S_{i,m}+1} < 0$ to hold with $a \geq 1$, after some algebraic manipulation we reach that: $R_u \geq \frac{T_u}{2}$. In other words, R_u should be at least half the value of T_u.

For $M_i = \varphi$, we have that:

$$R_u - g_u < T_u \left(\frac{1}{2a+1} - \frac{1}{(2a+1)(S_{i,m}+1)} \right)$$

Once again, since $a \geq 1$, we get after solving the inequalities that $\frac{1}{S_{i,m}+1} < 0$ will hold if the following condition is met: $g_u \leq R_u - \frac{T_u}{3}$. In other words, the upper limit of the untrusted zone should always be greater or equal to one-third the value of T_u.

By dishonest behavior, we mean a node providing false information about another node. Notice that this information might be a higher trust value or a lower trust value than the actual trust value. We assume that all good nodes for a particular node will

always remain honest whereas, bad nodes for a node might show dishonest behavior. A trust calculation method is said to be resilient against dishonest behavior if by simulating the bad and dishonest nodes in the algorithm by bad but honest nodes we get the same trust value.

Definition 3.4.7. *A set of bad nodes B_i for a node i is said to have successfully cheated i, if for a node j, the trust calculation algorithm 'A' for j*

$$\mathrm{A}\left(\left\{T'_{x,j} | x \in \mathrm{B}_i\right\}, \left\{T'_{y,j} | y \in \mathrm{B'}_i\right\}\right) \neq \mathrm{A}\left(\left\{T_{x,j} | x \in \mathrm{C}_i\right\}, \left\{T_{y,j} | y \in \mathrm{B'}_i\right\}\right)$$

Where C_i is a set in which every bad node in B_i is replaced by an honest but bad node.

Claim 3: GTMS is resilient against cheating at the node level.

Proof. The proof is straightforward. The only point in our protocol where we need the trust values from the other nodes while calculating the trust value of a node is during peer recommendation. However since we do not ask the recommendation from the bad nodes or the really bad nodes, therefore

$$\mathrm{A}\left(\left\{T'_{i,y} | y \in \mathrm{B'}_i\right\}\right) = \mathrm{A}\left(\left\{T'_{i,y} | y \in \mathrm{B'}_i\right\}\right)$$

As we assumed that the good nodes would always behave honestly. □

In the aforementioned text, we have attributed dishonest behavior (sending false recommendation values) to bad or really bad nodes for a particular node, say i. There might be nodes that are good nodes for i yet at the same time bad or really bad nodes for a node j. Whenever i wishes to find recommendations for j, these set of nodes might send false recommendations to i. Going further, we can even associate dishonest behavior to good nodes as well. If the number of such dishonest nodes is far less as compared to the honest ones, the effect of these false recommendations on the overall trust value as calculated by Equation 3.5 would be minimum. However, a collaboration of a greater

number of nodes will effect the trust value to a greater degree. This is true since Equation 3.5 has the form of a weighted average measure. Thus Equation 3.5 has a slight inbuilt capability of diminishing the effect of abnormal recommendations. As we will see in the next subsections, similar is true for trust calculation at the base station level.

There is another interesting way in which a collaboration of nodes might work together in achieving a malicious goal. Suppose we have nodes i, j and k. Node j is within i's radio range, while node k is not. k, however, is in the radio range of j. i sends a data packet to j which in turn sends the data packet to k. If k drops the packet, j should count that as an unsuccessful interaction. However, if j and k are collaborating, whereby j does not count it as an unsuccessful interaction, then there is no way that i would be able to detect it. Thus i might continue to send packets to j, which in turn would send them to k, only to be dropped by it. This, however, can be resolved if i sends its packets uniformly at random to all its trusted neighboring nodes turn by turn. This way, i will not send every packet to the two collaborating nodes and much of its packets will be forwarded successfully provided there is not a high percentage of collaborating nodes among its neighbors. This will prohibit the above mentioned scenario from reoccurring every time.

3.4.1.2 Resilience Analysis at Cluster Head level

At the CH, the trust value is calculated by getting the trust states of all nodes. At this stage of the protocol, we check the behavior of a collaboration of really bad nodes. We assume that in a group with $n+1$ nodes including the cluster head, the number of really bad nodes are less than or equal to $\lfloor n/2 \rfloor$. These really bad nodes are really bad for all other nodes in the group.

Definition 3.4.8. *A set of really bad nodes (mal) are said to be collaborating with each*

other if they provide false trust states of a particular node to the cluster head.

Definition 3.4.9. *A collaboration of really bad nodes is successful against a node $j \notin mal$, if the following conditions hold:*

1. $\forall i \notin mal, s_{i,j} = trusted$
2. $Z_j < -1$

Definition 3.4.10. *A collaboration of really bad nodes is successful internally for a node $m \in mal$, if the following conditions hold:*

1. $\forall i \notin mal, s_{i,m} = untrusted$
2. $Z_m > 1$

Claim 4: A set of really bad nodes cannot collaborate successfully against a node $j \notin mal$ and internally for a node $m \in mal$.

Proof. We have:

$$Z_j = \frac{\sqrt{3}\left(X\left(s_{ch,j}\right)+\sum_{i=1,i\neq j}^{n} X(s_{i,j})-n\right)}{\sqrt{n}}$$

Now, $\sum_{i\notin mal} X_{i,j} \geq 2\lfloor n/2 \rfloor \geq n$. Therefore,

$$Z_j \geq \frac{\sqrt{3}(n-n)}{\sqrt{n}} \geq 0$$

This shows that the cluster head will not label this node as an un-trusted node. For part 2, notice that $\sum_{i\in mal, i\neq m} X_{i,m} \leq 2\left(\lfloor n/2 \rfloor - 1\right) \leq n-2$ Since $\forall i \notin mal, s_{i,m} = untrusted$, therefore:

$$Z_m \leq \frac{\sqrt{3}(n-2-n)}{\sqrt{n}} \leq \frac{-2\sqrt{3}}{\sqrt{n}} < 0$$

This implies that bad nodes would never make it to the trusted zone at the cluster head.

□

Definition 3.4.11. *A group is said to be 'malicious' if during its course of interactions with the other group the majority of interactions are unsuccessful.*

We will denote a malicious group by G_m. Let G denote the set of nodes in a generic group inside the sensor network.

Definition 3.4.12. *A malicious group G_m is said to have successfully deceived a group G_j, if for all groups $G_i \in G - G_m$, $s_{G_i,G_m} = trusted$ and there exists at least one $G_j \in G - G_m$, such that: $U_{G_i,G_m} \geq S_{G_i,G_m}$ and at least one of U_{G_i,G_m} and S_{G_i,G_m} is non zero.*

Definition 3.4.13. *A Trust Management Scheme is said to be resilient against deception at group level if no group can successfully deceive another group.*

Claim 5: GTMS is resilient against deception at group level.

Proof. Similar to Claim 1. □

Definition 3.4.14. *A malicious group G_m is said to have partially deceived a group G_j, if for all groups $G_i \in G - G_m$, $s_{G_i,G_m} = uncertain$ and there exists at least one $G_j \in G - G_m$, such that: $U_{G_i,G_m} \geq 2S_{G_i,G_m}$ and at least one of U_{G_i,G_m} and S_{G_i,G_m} is non zero..*

Definition 3.4.15. *A Trust Management Scheme is said to be resilient against partial deception at group level if no group can partially deceive another group.*

Claim 6: GTMS is resilient against partial deception at group level.

Proof. Similar to Claim 2. □

3.4.1.3 Resilience Analysis at Base Station level

At the base station, the trust values of various groups are calculated. There can be three possible ways in which a particular group could cheat or try to get an unfair advantage. First, it might try to increase its own trust value even though it has not behaved well in the past. This cannot be done, as the base station asks other groups for there recommendations and its own past interaction records. Hence the group whose trust value is being calculated has no say in this computation. The second scenario deals with one or more group nodes collaborating to harm the trust calculation of a particular group by submitting low but false recommendations for that group. Finally, these collaborating nodes might try to enhance each others' trust values at base station by giving high but false recommendations to the base station. We assume that the only group that will show dishonest behavior is this set of really bad groups.

Definition 3.4.16. *A set of bad groups B_i for the base station is said to have successfully cheated, if for a group j, the trust calculation algorithm 'A' for j has the following property:*

$$\mathrm{A}\left(\left\{T'_{x,j} | x \in \mathrm{B}_i\right\}, \left\{T'_{y,j} | y \in \mathrm{B'}_i\right\}\right) \neq \mathrm{A}\left(\left\{T_{x,j} | x \in \mathrm{C}_i\right\}, \left\{T_{y,j} | y \in \mathrm{B'}_i\right\}\right)$$

Where C_i is the set obtained by replacing every bad and dishonest group in B_i with a bad but honest group.

Claim 7: GTMS is resilient against cheating at the base station.

Proof. The proof is straightforward. The only place in our protocol where we need the trust values from the other nodes, while calculating the trust value of a node is during peer recommendation. However, since the base station does not ask the recommendation from the bad groups, therefore:

$$\mathrm{A}\left(\left\{T'_{i,y} | y \in B'_i\right\}\right) = \mathrm{A}\left(\left\{T'_{i,y} | y \in B'_i\right\}\right)$$

□

3.4.2 Communication Overhead Analysis

We assume a worst case scenario, in which every member node wants to communicate with every other node in the group and every group wants to communicate with the rest of the groups in the network. Let us assume that the network consists of $|G|$ groups and the average size of groups is σ.

In the intra-group communication case, when node i wants to interact with node j, node i will send maximum $\sigma - 2$ peer recommendation requests. In response, node i will receive $\sigma - 2$ responses. If node i wants to interact with all the nodes in the group, the maximum communication overhead will be $2(\sigma - 1)(\sigma - 2)$. If all nodes want to communicate with each other, the maximum intra-group communication overhead (C_{intra}) of the GTMS scheme is: $2\sigma(\sigma - 1)(\sigma - 2)$.

In the inter-group communication case, when group i wants to interact with group j, it will send one peer recommendation request to the base station, at the maximum. So the communication overhead is two packets. If group i wants to communicate with all the groups then the maximum communication overhead will be $2|G| - 1$ packets. If all the groups want to communicate with each other, the maximum inter-group communication overhead (C_{inter}) of the GTMS scheme is: $2|G|(|G| - 1)$. Therefore the maximum communication overhead (C) introduced by the GTMS scheme in the network is:

$$\begin{aligned} C &= |G| \times C_{intra} + C_{inter} \\ C &= |G|\,[2\sigma(\sigma - 1)(\sigma - 2)] + 2|G|(|G| - 1) \\ C &= 2|G|\,[\sigma(\sigma - 1)(\sigma - 2) + (|G| - 1)] \end{aligned} \tag{3.11}$$

In general, communication overhead introduced by the GTMS scheme in the whole net-

Table 3.1: Communication overhead in worst case

	Communication overhead
GTMS	$2\|G\|\,[\sigma(\sigma-1)(\sigma-2)+(\|G\|-1)]$
RFSN	$2\|G\|\,[\sigma(\sigma-1)(\sigma-2)+(\|G\|-1)(\|G\|-2)]$
PLUS	$2\|G\|\,[\sigma(\sigma-1)^2+(\|G\|-1)^2]$
ATRM	$4\|G\|\,[\sigma(\sigma-1)+(\|G\|-1)]$

work is:

$$C = 2|G|\,[\sigma(\sigma-1)\rho+(|G|-1)] \tag{3.12}$$

where ρ represents the average number of recommender nodes in the group. Communication overhead of other schemes is shown in Table 3.1. More details about the RFSN, ATRM and PLUS schemes are given below.

In case of RFSN, when node i wants to interact with node j, it will send $n-2$ peer recommendation requests at the maximum. In response, node i will receive $n-2$ responses. If node i want to interact with all the nodes in the group then the maximum communication overhead will be $2(n-1)(n-2)$. If all the nodes want to communicate with each other, the maximum intra-group communication overhead (C_{intra}) will be: $2n(n-1)(n-2)$. When the CH of group i wants to interact with the CH of group j, it will send $|G|-2$ peer recommendation requests at the most. So the communication overhead will be: $2(|G|-2)$. If group i wants to communicate with all the groups then the maximum communication overhead will be: $2(|G|-1)(|G|-2)$. If all the groups want to communicate with each other, then the maximum inter-group communication overhead (C_{inter}) will be: $2|G|(|G|-1)(|G|-2)$. Therefore in the worst case, the maximum communication overhead (C) introduced by the RFSN scheme in the whole network is:

$$C = |G| \times C_{intra} + C_{inter}$$

$$C = |G|\,[2\sigma(\sigma-1)(\sigma-2)] + 2|G|(|G|-1)(|G|-2)$$
$$C = 2|G|\,[\sigma(\sigma-1)(\sigma-2) + (|G|-1)(|G|-2)]$$

where σ represents the average number of nodes in the group, and $|G|$ represents the total number of groups in the network.

In case of PLUS scheme, If node i wants to interact with another node j, then it will broadcast a request packet. In response, i will get $n-2$ responses. So the communication overhead will be: $1 + (n-2)$. If node i wants to communicate with all the nodes in the group, the communication overhead will be: $(n-1) + (n-1)(n-2)$. If all the nodes want to communicate with each other then the total intra-group communication overhead (C_{intra}) will be:

$$C_{intra} = n(n-1) + n(n-1)(n-2)$$
$$C_{intra} = n(n-1)^2$$

Each node in the group can also exchange anti-active protocol, whose communication cost is the same as getting recommendation from other nodes. So in the worst case, the total intra-group communication overhead (C_{intra}) will be: $2n(n-1)^2$. If group i wants to interact with another group j, then group i will broadcast a request packet. In response, it will get no more than $|G|-2$ responses. So the communication overhead will be: $1 + (|G|-2)$. If group i wants to communicate with all the groups then maximum communication overhead will be: $(|G|-1) + (|G|-1)(|G|-2)$. If all the groups want to communicate with each other the total inter-group communication overhead (C_{inter}) will be:

$$C_{inter} = |G|(|G|-1) + |G|(|G|-1)(|G|-2)$$
$$C_{inter} = |G|(|G|-1)[1 + (|G|-2)]$$
$$C_{inter} = |G|(|G|-1)^2$$

If we add the communication overhead of anti-active protocol then the maximum communication overhead for inter-group (C_{inter}) will be: $2|G|(|G|-1)^2$. Therefore, in the worst case, the maximum communication overhead (C) introduced by the PLUS scheme in the whole network is:

$$C = |G|C_{intra} + C_{inter}$$
$$C = |G|2\sigma(\sigma-1)^2 + 2|G|(|G|-1)^2$$
$$C = |G|2\sigma(\sigma-1)^2 + 2|G|(|G|-1)^2$$
$$C = 2|G|\left[\sigma(\sigma-1)^2 + (|G|-1)^2\right]$$

where σ represents the average number of nodes in the group.

In case of ATRM scheme, each node needs to exchange 4 packets in order to compute the trust. If a node i wants to communicate with all the nodes in the group then the communication overhead will be: $4(n-1)$. If all the nodes want to communicate with each other then the total communication overhead (C_{intra}) will be: $4n(n-1)$. Similarly if all groups want to communicate with each other, the inter-group communication (C_{inter}) will be: $4|G|(|G|-1)$. Therefore in the worst case, the maximum communication overhead (C) introduced by the ATRM scheme in the whole network is:

$$C = |G|C_{intra} + C_{inter}$$
$$C = |G|4\sigma(\sigma-1) + 4|G|(|G|-1)$$
$$C = |G|4\sigma(\sigma-1) + 4|G|(|G|-1)$$
$$C = 4|G|\left[\sigma(\sigma-1) + (|G|-1)\right]$$

where σ represents the average number of nodes in the group.

3.4.2.1 Comparison

Figure 3.4 shows the communication overhead of various trust management schemes for a large scale WSN (10000 nodes) having equal size of clusters. It shows that as the

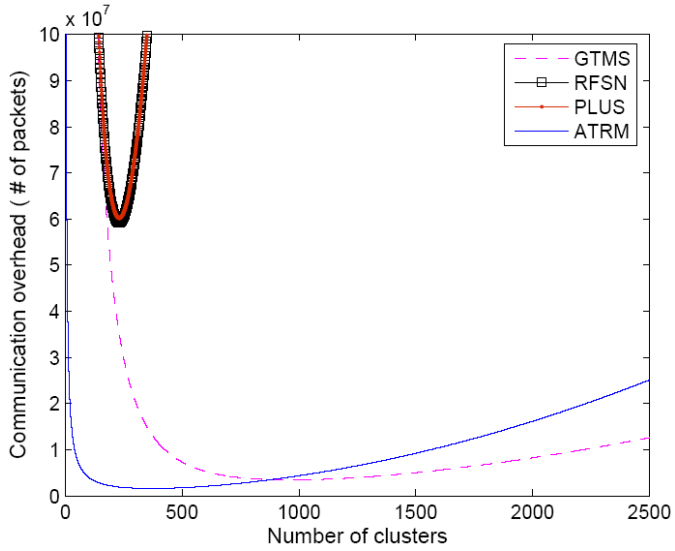


Figure 3.4: Communication overhead: Number of nodes=10000

number of cluster increases in the network the GTMS introduces less communication overhead as compared to the other schemes. Also, it indicates that GTMS is suitable for large scale WSNs having small size of clusters. The important thing that we need to note here about the ATRM scheme is that, it shows the result of just one transaction of each node. For example, when node i wants to communicate with node j they first exchange four packets. Once the transaction is completed and node i wants to initiate another transaction with j then the trust will be computed again. So the communication overhead of the ATRM scheme will increase with the factor of four with every transaction. Whereas for the case of GTMS scheme, after completion of first transaction, when node i wants to start another transaction with j, no extra communication overhead will occur. Because node i will calculate the trust based on the history of past transaction(s).

Table 3.2: Trust database at sensor node

Node ID	Past interactions based on time window						Peer recomm.	Trust value
	$S_{x,y}$			$U_{x,y}$				
	t_1	$\cdots$	t_n	t_1	$\cdots$	t_n		
2 bytes	2 bytes	$\cdots$	2 bytes	2 bytes	$\cdots$	2 bytes	1 byte	1 byte

3.4.3 Memory Consumption Analysis

One of the critical constraints of SNs is less availability of memory. For example, MICA2 SN has 128 Kbytes program flash memory, 512 Kbytes measurement flash, and 4 Kbytes EEPROM [91]. Our Group based trust management scheme does conform to this low-memory demand as discussed below.

3.4.3.1 Memory Requirement of GTMS at Node level

Each node maintains a small trust database as shown in Table 3.2. The size of each record is $4 + 4\Delta t$ bytes where Δt represents the size of the time window. Therefore, memory requirement for GTMS at each SN is $(n-1)(4+4\Delta t)$ bytes, where n represents the number of nodes in a group. The size of the trust table depends upon the size of the cluster and the length of time window.

3.4.3.2 Memory Requirement of GTMS at Cluster Head level

Each CH maintains two tables; one is similar to an individual SN's trust table and in the second, CH maintains the trust values of other groups as shown in Table 3.3. The size of each record is $4 + 4\Delta t$ bytes. Therefore the total size of Table 3.3 is $(|G| - 1)(4 + 4\Delta t)$ bytes, where $|G|$ represents the number of groups in the network. The total memory

Table 3.3: Group trust database at cluster-head

Node ID	Past interactions with other groups based on time window						Peer recomm. from BS	Trust value
	$S_{x,y}$			$U_{x,y}$				
	t_1	$\cdots$	t_n	t_1	$\cdots$	t_n		
2 bytes	2 bytes	$\cdots$	2 bytes	2 bytes	$\cdots$	2 bytes	1 byte	1 byte

Table 3.4: Memory requirement of trust management schemes

	Sensor node	cluster head
GTMS	$(n-1)(4+4\Delta t)$	$(\|G\|+\sigma-2)(4+4\Delta t)$
RFSN	$33(n-1)$	$33(G+\sigma-2)$
PLUS	$32.375(n-1)+28$	$32.375(G+\sigma-2)+28$
ATRM	$30n+8(k-1)$	$30(G+\sigma)+2(4k-19)$

space required at the CH for both tables is $(|G| + \sigma - 2)(4 + 4\Delta t)$ bytes. Here σ represents the average number of nodes in the group.

Memory requirement of other schemes is given in Table 3.4, in which n represents the number of nodes in the group, N represents the total number of nodes in the network, and k represents the number of the context. Details about how the memory requirements of the RFSN, ATRM and PLUS schemes are calculated are given below.

In case of the RFSN scheme, each SN also needs to store two tables: Reputation Module Matrix (RMM) and RFSN Monitor. RMM consist of 8 parameters: Context (4 bytes), Aging period (4 bytes), Aging weight (4 bytes), Integration weight(4 bytes), Size (1 byte), Alpha(4 bytes), Beta(4 bytes), and Node ID(2 bytes). So the size of one record in RMM is 27 bytes. RFSN Monitor maintains 2 parameters: Node ID(2 bytes) and Data

readings (4 bytes). So the size of one record of RFSN monitor is 6 bytes. Thus the total memory required by the RFSN scheme at SN is:

$$M_{SN} = \text{size(RMM)} + \text{size(Monitor)}$$
$$M_{SN} = 27(n-1) + 6(n-1)$$
$$M_{SN} = 33(n-1)$$

Here n represents the number of nodes in the neighborhood. Let us assume that every CH also maintains the trust value of other CHs in the same manner as nodes maintain trust value of other member nodes. Then, memory requirement at the CH in RFSN scheme is:

$$M_{CH} = 33(|G| - 1) + 33(\sigma - 1)$$
$$M_{CH} = 33(|G| + \sigma - 2)$$

where σ represents the average number of nodes in the group, and $|G|$ represents the total number of groups in the network.

In case of the ATRM scheme, each SN stores two tables: Trust evaluation table (Tab_{eval}) and t_Instrument table (Tab_{instr}). The Tab_{eval} table consists of 4 parameters: Node ID(2 bytes), Trust Context (4 bytes), Evaluation (4 bytes) and Time stamp (4 bytes). So the size of each record is 14 bytes. The Tab_{instr} table consists of 5 parameters: Node ID(2 bytes) , Trust context 4 bytes), INSTR (4 bytes), Time stamp (4 bytes), and ACK (2 bytes). So the size of each record for Tab_{instr} table is 16 bytes. Each SN also stores the r_certificate (r_{cert}) in a memory. The size of certificate varies with respect to the number of available contexts. The r_certificate is defined as: $RC = E_{AK}(R, H(R))$, where $R = ID_i, T, ((r_1, C_1), (r_2, C_2), ...(r_k, C_k))$. Here ID represents the identity of the node (2 bytes), T represents the time stamp (4 bytes), r_1 (4 bytes) represents the reputation of node i under context c_1 (4 bytes). So the size of R is $6 + 8k$. If we assume

the MD5 hash function (16 bytes) then the total size of r_{cert} is $22 + 8k$. Thus, the total memory required at the SN is:

$$M_{SN} = \text{size}(\text{Tab}_{\text{eval}}) + \text{size}(\text{Tab}_{\text{instr}}) + \text{size}(\text{r\_cert})$$
$$M_{SN} = 14(n-1) + 16(n-1) + (22 + 8k)$$
$$M_{SN} = 30(n-1) + 22 + 8k$$
$$M_{SN} = 30n + 8(k-1)$$

where n represents the number of nodes in the network. Let us assume that every CH also maintains the trust value of other CHs in the same manner as nodes maintain the trust value of other member nodes. CH maintains a single r_cert that is used for inter and intra communication. That is why the size of certificate will be added once. Thus in this case, memory requirement at the CH is:

$$M_{CH} = 30(|G| - 1) + 30(\sigma - 1) + 22 + 8k$$
$$M_{CH} = 30(|G| + \sigma - 2) + 22 + 8k$$
$$M_{CH} = 30(|G| + \sigma) + 2(4k - 19)$$

where σ represents the average number of nodes in the group, and $|G|$ represents the total number of groups in the network.

In the case of the PLUS scheme, each SN needs to store two tables and seven constant context parameters. First table consists of node ID (2 bytes), personal reference parameters (T_{or}(1 bit), T_{ai}(1 bit), T_{ce}(1 bit), T_{po}(4 bytes), T_{re}(4 bytes), T_{co}(4 bytes)), peer recommendation value T_i(4 bytes), and final calculated trust value (4 bytes). So the size of one record of first table is 22.375 bytes. In the 2^{nd} table, node needs to store information about node id (2 bytes), number of requests send (2 bytes), number of reply received (2 bytes), number of packets actually forwarded (2 bytes), and number of packets supposed to be forwarded (2 bytes). So the size of one record for 2^{nd} table is 10

bytes. Each context parameters ($W_{cp}, W_{po}, W_{re}, W_{oo}, W_{av}, W_{pr}, W_r$) is represented by 4 bytes. So the total size required to store context parameters is 28 bytes. Thus, the total memory required at the SN is:

$$M_{SN} = \text{size(table1)} + \text{size(table2)} + \text{contextParameters}$$
$$M_{SN} = 22.375(n-1) + 10(n-1) + 28$$
$$M_{SN} = 32.375(n-1) + 28$$

where n is the number of nodes in the neighborhood. Let us assume that every CH also maintains the trust value of other CHs in the same manner as nodes maintain trust values of other member nodes. Then in this case, memory requirement at the CH is:

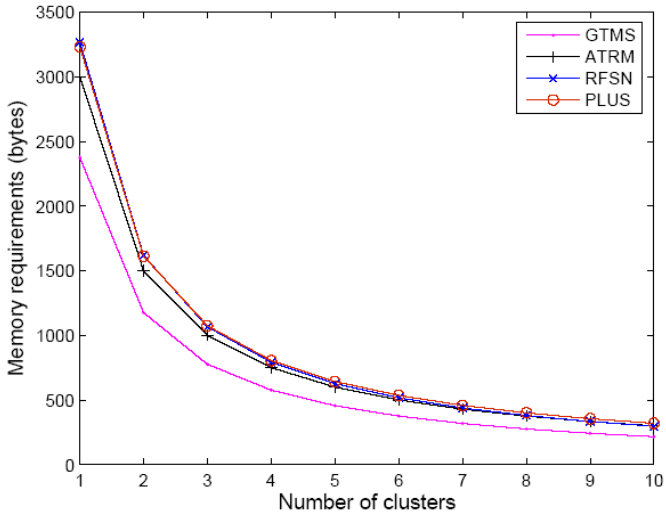
$$M_{CH} = 32.375(|G| - 1) + 32.375(\sigma - 1) + 28$$
$$M_{CH} = 32.375(|G| + \sigma - 2) + 28$$

where σ represents the average number of nodes in the group, and $|G|$ represents the total number of groups in the network.

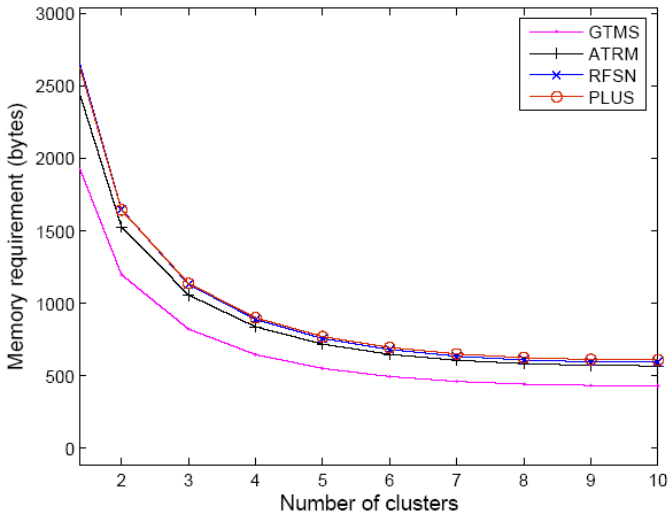
3.4.3.3 Comparison

In the simulation we assumed that all clusters are of equal size. We set the time window Δt equal to 5; So the size of trust record is 24 bytes. We have compared our scheme with RFSN [12], ATRM [14] and PLUS [13] schemes for the same clustering topology.

Results in Figure 4.5 are for 100 SNs. This graph shows that GTMS at SNs and CHs consumes less memory as compared to the ATRM, PLUS, and RFSN schemes. Memory consumption of GTMS at the CH depends upon the number of clusters in the network. As the number of clusters increases the memory consumption requirement also increases linearly at the CH. For example, if the network consists of 100 clusters with an average size of 20 nodes, then at the CH, GTMS consumes 2832 bytes of memory. This shows that GTMS can be used for large scale sensor networks.



(a) At sensor node



(b) At cluster head

Figure 3.5: Memory requirement: N=100 & $\Delta t = 5$ units.

3.4.4 Energy Consumption Analysis

In order to calculate the energy consumption, we must have the information about the number of bits transmitted and received during trust evaluation phase between different nodes. The size of packet is mainly dependent on the size of payload. Header and tailer fields of a packet generally remains constant. Therefore we have ignore those during theoretical analysis. Payload description of the GTMS and other schemes is given below.

The GTMS scheme is comprises of four pairs of request and response packets as shown in Table 3.5.

Pair 1: used for Peer Recommendation. Whenever a node x needs recommendation from node y about z, it sends a request packet (iTReq) of size 2 bytes to node y. In response, node y send a response packet (iTRep) of size 3 bytes to node x. The iTRep contains the trust value of z.

Pair 2: used for the transfer of trust vector from node to CH. After a periodic interval, the CH j broadcast a request (iVReq) packet inside the group. In response all nodes that belongs the cluster j send back a response packet (iVRep) of size $1 + 2.25v$ bytes, where $v \leq n - 1$ represents the length of the trust vector and n represents the total number of nodes in the cluster or group.

Pair 3: used for getting recommendation from BS by CH. Whenever a CH j need a recommendation from the BS about another cluster k, it send a request packet (oTReq) to the BS. In response, the BS send a response packet (oTRep) to the CH j that contain the trust value of CH k. The size of the response packet is 3 bytes.

Pair 4: used for the transfer of trust vectors from CH to BS. After every periodic interval, the BS multicast a request packet (oVReq) to all CHs in the network. In response, all CHs send back a response packet (oVRep) of size $1 + 3v$ bytes, where $v \leq |G|$ represents the length of the trust vector and $|G|$ represents the total number of groups.

Table 3.5: Packets of GTMS scheme

		Type	Payload	Size (payload)
packets move inside cluster	Pair 1: for peer recommendation	iTReq (SN-SN)	ID of evaluating node (2 bytes)	2 bytes
		iTRep (SN-SN)	ID of evaluating node (2 bytes), trust value (1 byte)	3 bytes
	Pair 2: for transfer of trust vector	iVReq (CH-SN)	Nil	-
		iVRep (SN-CH)	Vector length v(1 byte), ID (2 bytes) and trust state (1 bit) of v member nodes	1+2.25v bytes
packets move outside cluster	Pair 3: for peer recommendation	oTReq (CH-BS)	ID of evaluating node (2 bytes)	2
		oTRep (BS-CH)	ID of evaluating node (2 bytes), trust value (1 byte)	3 bytes
	Pair 4: for transfer of trust vector	oVReq (BS-CH)	Nil	-
		oVRep (CH-BS)	Vector length v(1 byte), ID (2 bytes) and trust value (1 byte) of other clusters	1+3v bytes

Table 3.6: Packets of RFSN scheme

Type	Payload	Size of payload
Req	ID of evaluating node (2 bytes)	2 bytes
Rep	ID of evaluating node(2 bytes), trust value(4 bytes)	6 bytes

In case of the RFSN scheme [12, 92], whenever a node needs recommendation value of the other node it will send a request packet (Req) to trusted nodes of the neighborhood. This request packet contain the identity of the evaluating node. In response to the Req packet, trusted neighborhood nodes send back reply messages (Rep) to the requester. This reply packet contain the identity of the evaluating node and its trust value. Packet description of the RFSN scheme is shown in Table 3.6.

In case of the PLUS scheme [13], whenever a node needs recommendation about another node, it will broadcast a request packet ($EReq$) to its neighbors. This packet contain the identity of the evaluating node. In response all the nodes (except the node whose is going to be evaluated) send back a response packet ($ERep$) to the requester. Once all the response packets are received, the requester will calculate the final trust value. If the node find any misbehavior about the evaluated node, then the node will broadcast a exchange information packet ($EInf$) to its neighbors. This packet contain information about identity of the node and error code. Based on the trust policy, the neighboring nodes sends out its opinion: exchangeAck ($EAck$) packet in case if they agree with the sender, otherwise neighbors will reply with exchageArgue ($EArg$) packet. Packet description of the PLUS scheme is shown in Table 3.7.

For the energy consumption analysis, we assume first order radio model, in which the energy expanded to transfer a k-bit packet to a distance d, and to receive that packet,

Table 3.7: Packets of PLUS scheme

Type	Payload	Size of payload
EReq	ID of evaluating node (2 bytes)	2 bytes
ERep	ID of evaluating node(2 bytes), trust value(4 bytes)	6 bytes
EInf	ID of evaluating node(2 bytes), Error code(2 bytes)	4 bytes
EAck	ID of evaluating node (2 bytes)	2 bytes
EArg	ID of evaluating node (2 bytes), trust value(4 bytes)	6 bytes

as suggested by H.O. Tan and I. Korpeoglu in [93] is:

$$\begin{aligned} E_{Tx}(k,d) &= kE_{elec} + kd^2E_{amp} \\ E_{Rx}(k) &= kE_{elec} \end{aligned} \tag{3.13}$$

Here, E_{elec} is the energy dissipation of the radio in order to run the transmitter and receiver circuitry and is equal to $50nJ/bit$. The E_{amp} is the transmit amplifier that is equal to $100pJ/bit/m^2$.

We have performed the theoretical energy consumption analyses and evaluation of various trust management schemes in different scenarios.

3.4.4.1 Scenario 1

When a SN needs a recommendation about other nodes, it will send a request packet to its peers. In the case of the GTMS scheme, the requester will send request to all the the nodes except the un-trustful ones. Assume that out of n nodes, j nodes are trusted and

uncertain. Then, the total energy consumed at the requester end is,

$$E = j\left[E_{Tx}(k,d) + E_{Rx}(k')\right] \tag{3.14}$$

where, $0 < j \leq n-2$, and n is the number of nodes in the group. For peer recommendation, the payload size of a request packet is 2 bytes, thus $k = 16$ bits. The payload size of a response packet is 3 bytes, thus $k' = 24$ bits. So the total energy consumed at the requester end is:

$$\begin{aligned} E &= j\left[E_{Tx}(16,d) + E_{Rx}(24)\right] \\ E &= j\left[16(E_{elec} + d^2 E_{amp}) + (24E_{elec})\right] \end{aligned} \tag{3.15}$$

Also for the GTMS, the energy consumed at the responder end is:

$$\begin{aligned} E &= E_{Rx}(16) + E_{Tx}(24,d) \\ E &= 16E_{elec} + 24(E_{elec} + d^2 E_{amp}) \end{aligned} \tag{3.16}$$

Energy consumption during peer recommendation of other schemes is shown in Table 3.8.

In the case of the RFSN scheme, the energy consumption at the requester end is:

$$E = t \times \left[E_{Tx}(16,d) + E_{Rx}(48)\right] \tag{3.17}$$

where t represents the number of trusted node in the cluster ($0 < t \leq n-2$), 16 and 48 represents the size of the request and response packets of RFSN scheme respectively. Also for the RFSN, the energy consumed at the responder end is:

$$\begin{aligned} E &= E_{Rx}(16) + E_{Tx}(48,d) \\ E &= 16E_{elec} + 48(E_{elec} + d^2 E_{amp}) \end{aligned} \tag{3.18}$$

Table 3.8: Peer recommendation of sensor nodes within a cluster

	GTMS	RFSN	PLUS
Number of request packets forwarded	$j \leq n-2$	$t \leq n-2$	1
Number of response packets received	$j \leq n-2$	$t \leq n-2$	$n-2$
Size of request packet (payload only)	16 bits	16 bits	16 bits
Size of response packet (payload only)	24 bits	48 bits	48 bits
Energy consumption at requester	$j[E_{T_x}(16,d)+E_{R_x}(24)]$	$t[E_{T_x}(16,d)+E_{R_x}(48)]$	$E_{T_x}(16,d)+(n-2)E_{R_x}(48)$
Energy consumption at responder	$E_{T_x}(24,d)+E_{R_x}(16)$	$E_{T_x}(48,d)+E_{R_x}(16)$	$E_{T_x}(48,d)+E_{R_x}(16)$

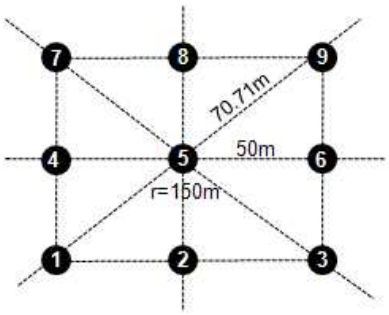


Figure 3.6: Sample group scenario

In the case of the PLUS scheme, the minimum energy consumption at the requester end is:

$$\begin{aligned} E &= E_{Tx}(16, d) + (n-2)E_{Rx}(48) \\ E &= 16(E_{elec} + d^2 E_{amp}) + (n-2)(48E_{elec}) \end{aligned} \tag{3.19}$$

Here 16 and 48 represents the size of the request and response packets of the PLUS scheme respectively. Also for the PLUS, the energy consumed at the responder end is:

$$\begin{aligned} E &= E_{Rx}(16) + E_{Tx}(48, d) \\ E &= 16E_{elec} + 48(E_{elec} + d^2 E_{amp}) \end{aligned} \tag{3.20}$$

In order to compare the energy consumption during peer recommendation scenario within the a cluster, we have assumed that a single group consists of nine nodes arranged in a grid fashion as shown in Figure 3.6. For this small topology, we have taken two scenarios. In the first scenario we have only two requesters getting recommendation from one available trusted node, and in second scenario, two requesters are getting recommendation from all the available trusted nodes (excluding the one who is going to be evaluated) by the requester. First scenario shows the minimum energy consumption analysis and second scenario shows the maximum energy consumption analysis of the group.

Figure 3.7(a) shows the minimum energy consumption analysis (first scenario), which shows that GTMS consume less energy as compared to the PLUS scheme. Also, GTMS consume approximately same amount of energy as RFSN scheme. Figure 3.7(b) illustrates the maximum energy consumption analysis (second scenario), which shows that the GTMS scheme overall consume less energy in a group then the PLUS scheme at the cost of slightly more energy consumption at the requester ends. Also, as compared to the RFSN scheme, GTMS scheme consume less energy at the responder (recommender) ends and approximately same energy at the requester ends.

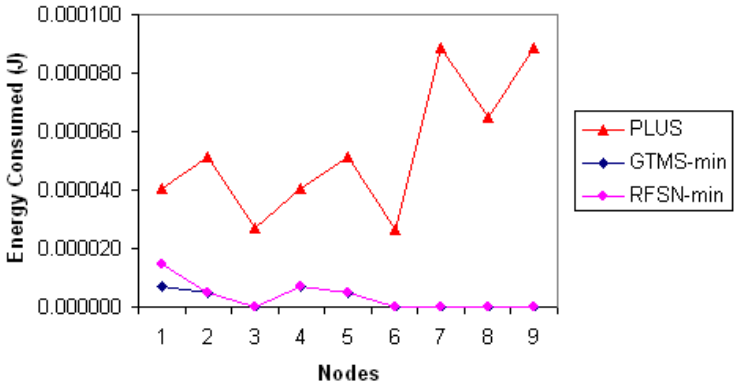
3.4.4.2 Scenario 2

In case of the GTMS scheme, when ever a cluster head need a recommendation value about another group then the cluster head will send a request packet to the base station, in response base station will send back trust value of other group. Therefore, tin case of the GTMS scheme, the total energy consumed at the cluster head will be;

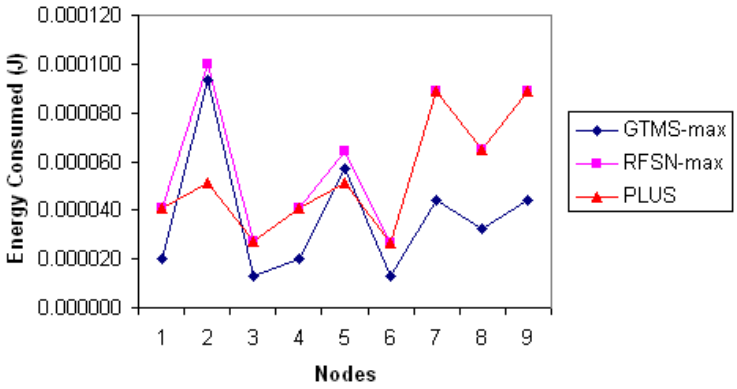
$$\begin{aligned} E &= E_{Tx}(16, d) + E_{Rx}(24) \\ E &= 16(E_{elec} + E_{amp} \times d^2) + 24E_{elec} \end{aligned} \tag{3.21}$$

where 16 bits represents the size of the request packet and 24 bits represents the size of the response packet. In this case responder is base station which usually does not have any resource constraints. Therefore, we can ignore the energy consumption analysis at the base station.

In case of the RFSN scheme, when ever a cluster head need a recommendation value about another group then the cluster head will send a request packets to its neighboring cluster heads. In response neighboring cluster heads will send back trust value of other group. Therefore, in case of the RFSN scheme, the total energy consumed at the



(a) Minimum energy consumption with 2 requesters (2 need recom. about 3 from 1, and 5 needs recom. about 6 from 4)



(b) Maximum energy consumption with 2 requesters (2 need recom. about 3, & 5 need recom. about 6 from all other nodes)

Figure 3.7: Energy consumption during peer recommendation scenario of sensor nodes

requester cluster head will be;

$$\begin{aligned} E &= \sum_{j=0}^{r} E_{Tx}(16, d) + \sum_{j=0}^{q} E_{Rx}(48) \\ E &= \sum_{j=0}^{r} (16(E_{elec} + E_{amp} \times d^2)) + \sum_{j=0}^{q} (48E_{elec}) \\ where, q &\leq r; \end{aligned} \tag{3.22}$$

where r represents the number of request packets and q represents the number of response packets. The size of request packet is 16 bits and the size of response packet is 48 bits. The total energy consumed at the responder cluster head will be:

$$E = 16E_{elec} + 48(E_{elec} + E_{amp} \times d^2) \tag{3.23}$$

In case of the PLUS scheme, when ever a cluster head need a recommendation value about another group then the cluster head will broadcast request packet to its neighboring cluster heads. In response, all neighboring cluster heads will send back trust value of the required group. Therefore, in case of the RFSN scheme, the total energy consumed at the requester cluster head will be;

$$\begin{aligned} E &= E_{Tx}(16, d) + \sum_{j=0}^{q} E_{Rx}(48) \\ E &= 16(E_{elec} + E_{amp} \times d^2) + \sum_{j=0}^{q} (48E_{elec}) \end{aligned} \tag{3.24}$$

where q represents the number of response packets. The size of request packet is 16 bits and the size of response packet is 48 bits. The total energy consumed at the responder cluster head will be:

$$E = 16E_{elec} + 48(E_{elec} + E_{amp} \times d^2) \tag{3.25}$$

Summary of energy consumption during peer recommendation of cluster heads is shown in Table 3.9. Here m represents the total number of neighboring cluster heads.

Table 3.9: Peer recommendation of cluster heads

	GTMS	RFSN	PLUS
Number of request packets forwarded	1	$r \leq m-1$	1
Number of response packets received	1	$q \leq r$	$q \leq m-1$
Size of request packet (payload only)	16 bits	16 bits	16 bits
Size of response packet (payload only)	24 bits	48 bits	48 bits
Energy consumption at requester	$E_{T_x}(16,d) + E_{R_x}(24)$	$\sum_{j=0}^{r} E_{T_x}(16,d) + \sum_{j=0}^{q} E_{R_x}(48)$	$E_{T_x}(16,d) + \sum_{j=0}^{q} E_{R_x}(48)$
Energy consumption at responder	-	$E_{T_x}(48,d) + E_{R_x}(16)$	$E_{T_x}(48,d) + E_{R_x}(16)$

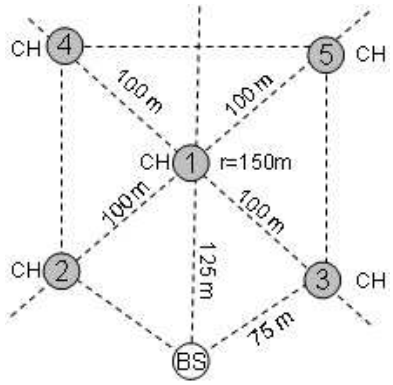


Figure 3.8: Cluster scenario

In order to compare the energy consumption during peer recommendation scenario between clusters, we have assumed 5 clusters and one base station in the network as shown in Figure 3.8. In this scenario CH_1 needs recommendation about CH_2 and CH_3 needs recommendation about CH_4.

Figure 3.9 clearly shows that the GTMS consumes less energy as compared with the RFSN and PLUS schemes. This is because, in GTMS cluster head only need recommendation from the base station. Whereas, in RFSN and PLUS schemes cluster head need recommendation from its neighboring cluster heads. This figure also illustrates that at the requester ends (CH_1 and CH_3) PLUS scheme consume less energy, because request packet is broadcast to all its neighboring cluster heads. Whereas, in case of the RFSN scheme, the request packet is unicasted to all trusted neighboring cluster heads.

Scenario 3 and 4 are only applicable to the GTMS scheme. Therefore, we have compared the GTMS scheme with the generic Distributed Trust Management Scheme (DTMS) in which each node maintains a one-to-one trust relationship with each other.

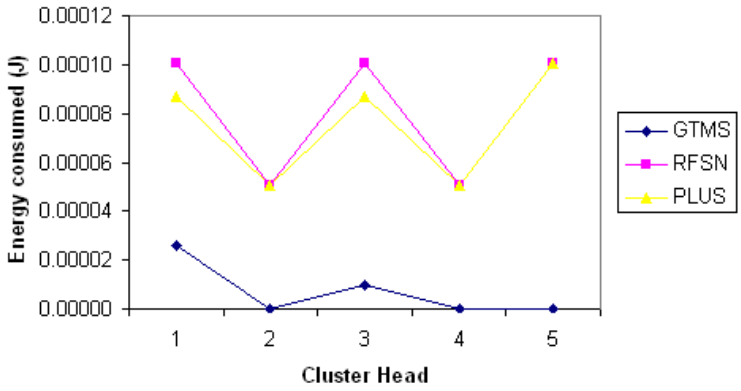


Figure 3.9: Peer recommendation: 1 needs recom. for 2 & 3 needs recom. for 4.

3.4.4.3 Scenario 3

Whenever a sensor node gets request to send trust vector from the cluster head, it will send $n-1$ bytes of trust vector data to the cluster head. Here n is the number of nodes in the cluster. At the requester end, the total energy consumed during this phase is the sum of the energy consumed during sending of the request packet (E_{T_x}) plus energy consumed during receiving of the response packet (E_{R_x}) from all member nodes, as given below:

$$E = E_{T_x}(k, d) + \sum_{j=0}^{r} E_{R_x}(k') \tag{3.26}$$

$$E = k \times (E_{elec} + E_{amp} \times d^2) + \sum_{j=0}^{r} E_{elec} \times k' \tag{3.27}$$

Here k is the length of the request packet, k' is the length of the response packet and r represents the number of responses received by the requester. Payload of the request packet does not contains any additional information and can be identified by the *type* field present in the header of the packet. As we have already mentioned in earlier discussion that the size of header remains constant for all protocols. Therefore, we can assume that

size of the request packet is 1 and, the size of the response packet (k') is $8 + 18v$ bits [See Table 3.5]. Then the total energy consumed at the requester end will be;

$$E = 1 \times (E_{elec} + E_{amp} \times d^2) \ + \sum_{j=0}^{r} E_{elec} \times (8 + 18v) \tag{3.28}$$

In the case of the GTMS, $r \leq n - 1$ and $v \leq n - 1$, where n is the number of nodes in the group, where as in the case of the DTMS $r \leq N - 1$ and $v \leq N - 1$, where N is the number of nodes in the network.

At the responder end, the total energy consumed during this phase is the sum of energy consumed during receiving of the request packet (E_{R_x}) plus energy consumed during transfer of the response packet (E_{T_x}) as given below:

$$E = E_{elec} \times k + k' \times (E_{elec} + E_{amp} \times d^2) \tag{3.29}$$

Then the total energy consumed at the responder end will be;

$$E = E_{elec} \times 1 + (8 + 18v) \times (E_{elec} + E_{amp} \times d^2) \tag{3.30}$$

In the case of the GTMS, $v \leq n - 1$ where n is the number of nodes in the group and in the case of the DTMS, $v \leq N - 1$, where N is the number of nodes in the network.

Comparison of energy consumption from the requester and responder point of view is shown in Figure 3.10. In a simulation, the requester and responder reside at the distance of 150 meters from each other. Initially for 100 nodes in the sensor network, we assumed only one cluster. In this case, energy consumption of the GTMS and DTMS at the requester and responder ends remains same. But as we increase the number of clusters in the network, the GTMS shows lower energy consumption as compared with the DTMS. For example, for the case of five clusters in the network comprises of 100 nodes, at the requester end, the GTMS scheme consumed 26.47 times less energy as compared with the DTMS. For the same case at the responder end, the GTMS scheme consumed 5.11

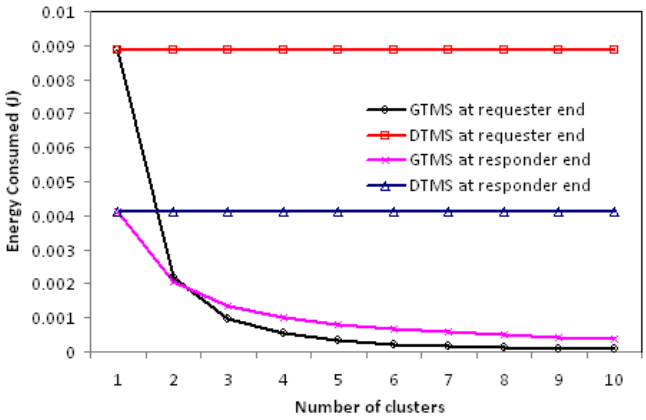


Figure 3.10: Energy Consumption of SN: N=100, d=150

times less energy as compared with the DTMS. This significant energy saving is only because the size of trust vector is depended on the size of the cluster. As we increase the number of clusters in the network, the average number of nodes in the cluster will decrease. If the numbers of nodes in the cluster become small then the size of trust vector will also reduce, which will take less transmission and reception power during transfer from a node to the cluster head.

3.4.4.4 Scenario 4

Whenever a base station needs a trust vector from the cluster heads it will send the request packet to all the cluster heads. In response all cluster heads will send the response packet to the base station. Since, the base station does not have any resource constraint problem, therefore, we have focused only on the energy consumption of the cluster heads. The total energy consumed at the responder (cluster head) end is:

$$E = E_{elec} \times 1 + [(8 + 24v) \times (E_{elec} + E_{amp} \times d^2)] \tag{3.31}$$

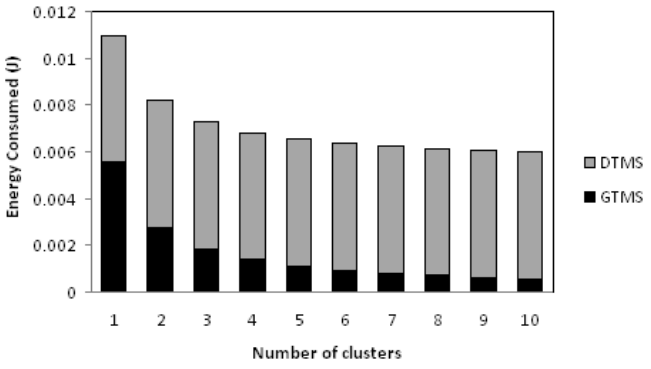


Figure 3.11: Energy Consumption of CH: N=100, d=150

In the case of the GTMS $v \leq |G| - 1$, where $|G|$ is the number of groups in the network. In the case of the DTMS $v \leq N - 1$, where N is the number of nodes in the network.

Comparison of both the schemes is shown in Figure 3.11. For the scenario of 100 nodes comprises of 10 equal size clusters, the GTMS consumed approximately 10.64 times less transmission and reception power as compared with the DTMS.

Discussion: The GTMS scheme is invariant of any specific radio technology. The energy consumption analysis presented above, is just a single application of first order radio model proposed by H. O. Tan and I. Korpeoglu in [93].

The GTMS scheme never consume more energy than the DTMS scheme as shown in Table 3.10. In a worst case scenario, when the number of nodes in a cluster is equal to the number of nodes in the network, than the energy consumption of both schemes remain same. In other cases, the GTMS scheme always consume less energy than the DTMS scheme.

Table 3.10: Summary: GTMS vs DTMS

Scenario	Node	Equation	Scaling factor
Scn-1	CH	$E_{T_x}(k,d) + rE_{R_x}(k')$	$r_{GTMS} \leq r_{DTMS}\,; k'_{GTMS} \leq k'_{DTMS}$
	SN	$E_{R_x}(k) + E_{T_x}(k',d)$	$k'_{GTMS} \leq k'_{DTMS}$
Scn-2	SN	$j[E_{T_x}(k,d) + E_{R_x}(k')]$	$j_{GTMS} \leq j_{DTMS}$
Scn-3	CH	$E_{T_x}(k,d) + E_{R_x}(k')$	-
Scn-4	CH	$E_{R_x}(k) + E_{T_x}(k',d)$	$k'_{GTMS} \leq k'_{DTMS}$

Table 3.11: Sensor network's specifications

Network size	No. of clusters	Terrain
144 nodes	16	$600m \times 600m$
225 nodes	25	$800m \times 800m$
324 nodes	36	$1000m \times 1000m$

3.5 Simulation-based Analysis and Evaluation

3.5.1 Simulation Environment

We have performed simulation using Sensor Network Simulator and Emulator (SENSE) [94]. We have deployed three different sized sensor networks consisting of 144, 225 and 324 sensor nodes. More details about these networks are available in Table 3.11. Nodes are static and are organized in a grid fashion. First, second and third network is comprised of 16, 25 and 36 clusters respectively. These numbers are chosen to make all clusters in equal size of nine nodes. Each network comprises of one base station that is located at the middle of the corresponding terrain. In all three networks,

Source ID	Dest. ID	Protocol ID	Type	Payload	Send Time
2 bytes	2 bytes	1 byte	1 byte	variable	4 bytes

Figure 3.12: TExP Protocol

we used free space wireless channel, IEEE 802.11 MAC protocol, and a simplified version of DSR routing protocol (without route repairing). At the application layer I have developed my own generic and simple Trust Exchange Protocol (TExP) as shown in Figure 3.12. This protocol consists of six fields:

1. SourceID: contains the identity of the source node.
2. DestID: contains the identity of the destination node.
3. Protocol ID: represents the identity of the trust management protocol e.g. GTMS, RFSN, etc.
4. Type: is used to identify the type of the packet such as request packet, response packet, acknowledgment packet etc.
5. Payload: field is of variable size containing the data specific to the type and protocol, such as trust value, identity of evaluating node etc.
6. SendT: contains the sending time of the packet.

The objective of the TExP protocol is to exchange the trust values between communicating nodes in an efficient manner. Sensor node architecture based on SENSE [94] is shown in Figure 6.2, which shows the interactions between GTMS, TExP and other components. The rest of the specifications of a sensor node are defined in Table 3.12.

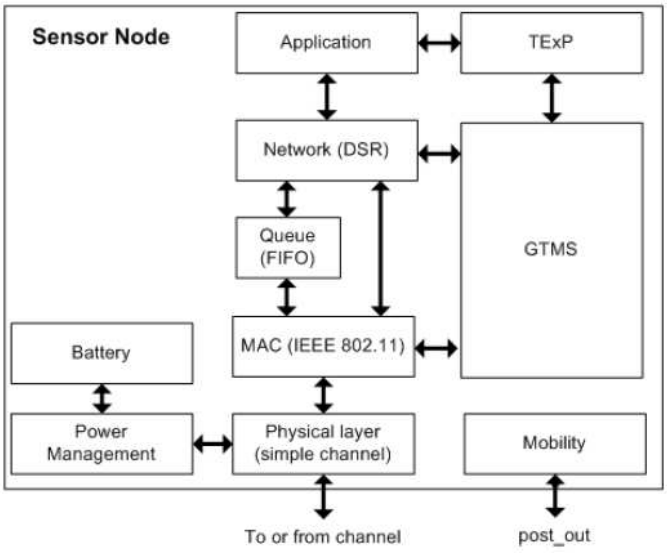


Figure 3.13: Sensor node architecture

Table 3.12: Sensor node's specifications

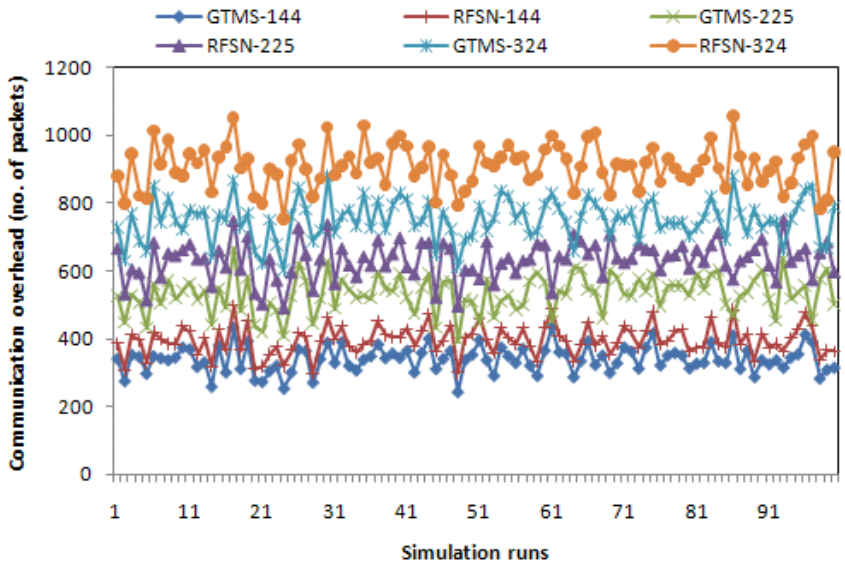
Initial battery of each sensor node	$1 \times 10^6 J$
Power consumption for transmission	$1.6W$
Power consumption for reception	$1.2W$
Power consumption in idle state	$1.15W$
Transmission power of the antenna	0.0280
Transmission and Reception gain	1.0
Carrier sense threshold	$3.652e^{-10}W$
Receive power threshold	$1.559e^{-11}W$

3.5.2 Comparison

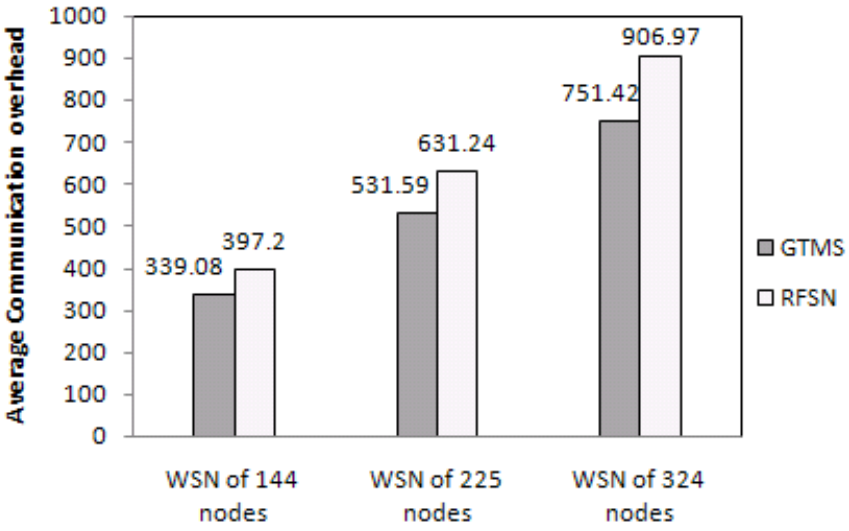
For the purpose of comparison, we have implemented a peer recommendation scenario. During simulation, in each cluster, random number of source nodes are selected which perform peer recommendation with the other nodes. Also, each cluster head will perform peer recommendation with neighboring cluster heads only. In the simulation we have only compared our proposed GTMS scheme with the RFSN scheme because both are independent of any specific routing scheme and platform. We did not implement the ATRM scheme because it requires some specific agent-based platform. Also, we did not implement the PLUS scheme because it works on the top of its own defined routing protocol.

Communication overhead for the three different networks is shown in Figure 3.14, which confirms our conclusions from the theoretical analysis. Figure 3.14(a) shows that the GTMS scheme introduces less communication overhead as compared to the RFSN scheme, and this pattern (overhead difference) approximately remains same for all 100 simulation runs. Therefore, we conclude that the 100 simulations runs can give us reliable results. Figure 3.14(b) shows that, as the network size increases, the communication overhead difference between the GTMS and RFSN schemes also increases. It shows that the GTMS would introduce 14.6%, 15.7% and 17.1% less communication overhead as compared to the RFSN scheme for the network of 144, 225 and 324 nodes respectively.

Communication overhead also effects the energy consumption of the sensor nodes. That effect is visible in Figure 3.15, which show that the GTMS also consume less energy as compared with the RFSN scheme.



(a) Communication overhead



(b) Average communication overhead

Figure 3.14: Average communication overhead analysis (100 simulations)

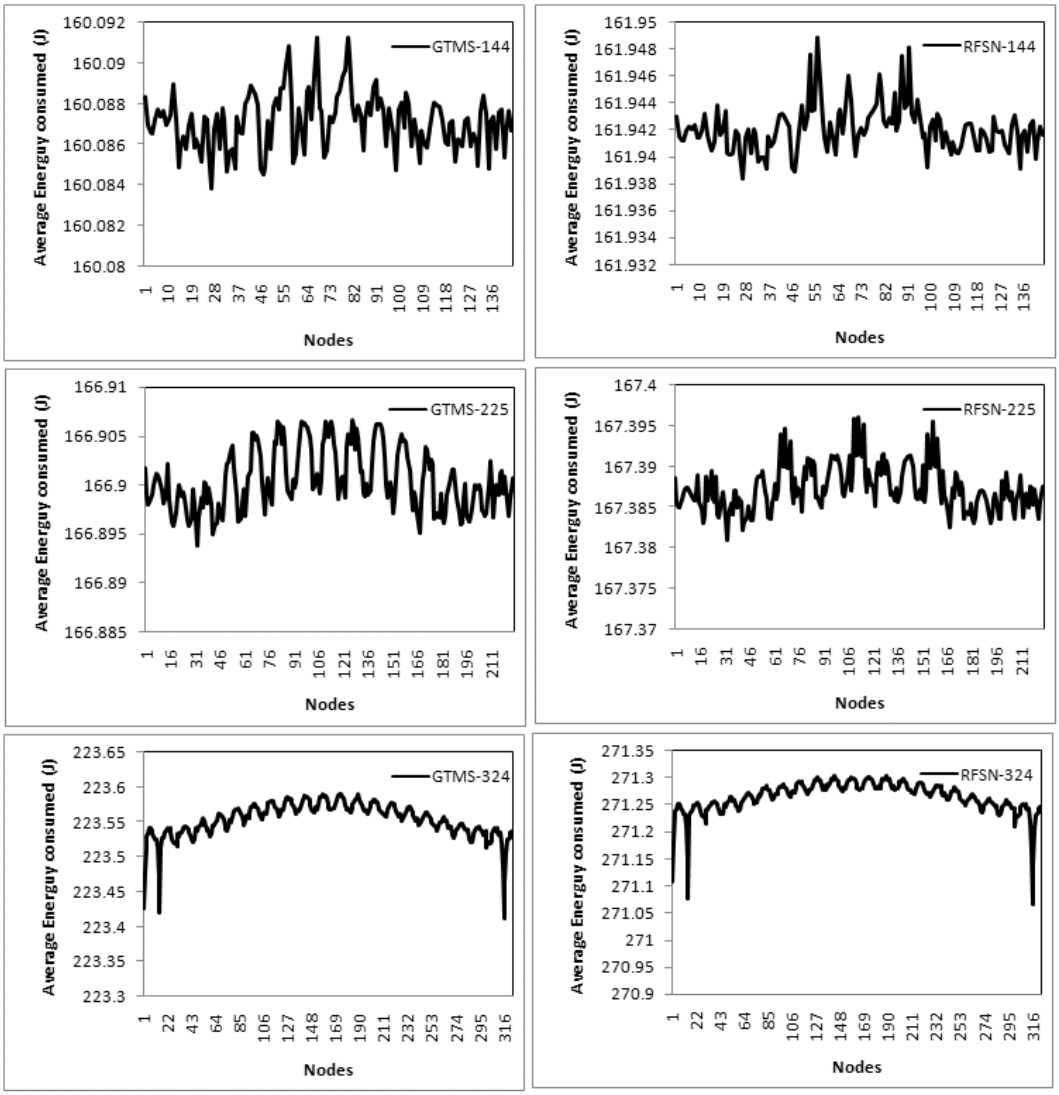


Figure 3.15: Average energy consumption at each node (100 simulations)

3.6 Summary

With the emergence of widespread use of WSNs, the need of a proper trust management scheme is strongly felt. In this work, we have proposed a robust lightweight group-based trust management scheme (GTMS) for clustered WSNs. GTMS uses a hybrid trust management approach, which reduces the cost of trust evaluation. Theoretical and simulation-based results showed that our scheme is memory efficient, and consumes less communication overhead. We also proved that the GTMS is intrusion tolerant and provides protection against malicious, selfish and faulty nodes.

Chapter 4

Network Level Privacy Component

4.1 Introduction

With the emergence of Wireless Sensor Networks (WSNs), the need of ensuring privacy is also gaining importance. Privacy can be categorized into two types: user level privacy and network level privacy. User level privacy is mainly assured by an authorization together with a confidentiality mechanism. Whereas, network level privacy is mainly assured by employing anonymity together with a confidentiality mechanism.

Networks are comprised of three dynamic entities: nodes, routes and packets. Based on these dynamic entities, full network level privacy has often been categorized into four sub-categories:

1. Sender node identity privacy: no intermediate nodes can get any information about who is sending the packets except the source, its immediate neighbors and the destination,

2. Sender node location privacy: no intermediate nodes have any information about the location (in terms of physical distance or number of hops) about the sender node except the source, its immediate neighbors and the destination,

3. Route privacy: no nodes can predict the information about the complete path (from

source to destination). Also, a mobile adversary gets no clue to trace back the source node either from the contents and/or directional information of the captured packet(s), and

4. Data packet privacy: no nodes can be able to see the information inside in a payload of the data packet except the source and the destination.

This chapter focuses on these four aspects. However, since the destination node is usually the sink node or the base station that is known to all the nodes in the network (For example, monitoring-based WSNs [6]), there is no need to consider the identity and location privacy of the destination node.

Existing privacy schemes such as [6, 8, 18, 95, 20, 21, 96] that have specifically been proposed for WSNs only provide partial network level privacy. Providing a full network level privacy spectrum is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g. energy, memory and computation power), sensor network (e.g. mobility, and topology) and QoS issues (e.g. packet reach-ability, and trustworthiness).

In order to achieve this goal, I incorporate basic design features from related research fields such as geographic routing and cryptographic systems. To my knowledge, I propose the first full network level privacy solution for WSNs. My contribution lies in following features.

- A new Identity, Route and Location (IRL) privacy algorithm is proposed that ensures the anonymity of source node's identity and location from the adversary. It also gives assurance that the packets will reach their destination by passing through only trusted intermediate nodes.

- A new reliable Identity, Route and Location (r-IRL) privacy algorithm is proposed, which is the extension of our proposed IRL algorithm. This algorithm has the

ability to forward packets from multiple secure paths to increase the packet reachability.

- A new data privacy mechanism is proposed, which is unique in the sense that it provides data secrecy and packet authentication *in the presence of identity anonymity*.

Our solutions collectively provides protection against various privacy disclosure attacks such as eavesdropping and hop-by-hop trace-back attacks. Also, our solutions are lightweight and hence consumes modest memory and energy.

The rest of this chapter is organized as follows: Section 4.2 articulates the network model, assumptions and adversary model. Section 4.3 describes the proposed privacy schemes, Section 4.4 consists of analysis and evaluation, and Section 4.5 concludes the chapter.

4.2 Network, Assumptions and Adversary Model

4.2.1 Network Model

A WSN is composed of large number of tiny sized resource-constraint sensor nodes that are densely deployed in an environment. Whenever end users require information about any event related to some object(s), they send a query to the sensor network via the base station. And the base station propagates that query to the entire network or to a specific region of the network. In response to that query, sensor nodes send back required information to the base station. A typical wireless sensor network scenario is shown in Figure 4.1. Links are bidirectional. Also, sensor nodes uses IEEE 802.11 standard link layer protocol, which keeps packets in its cache until the sender receives

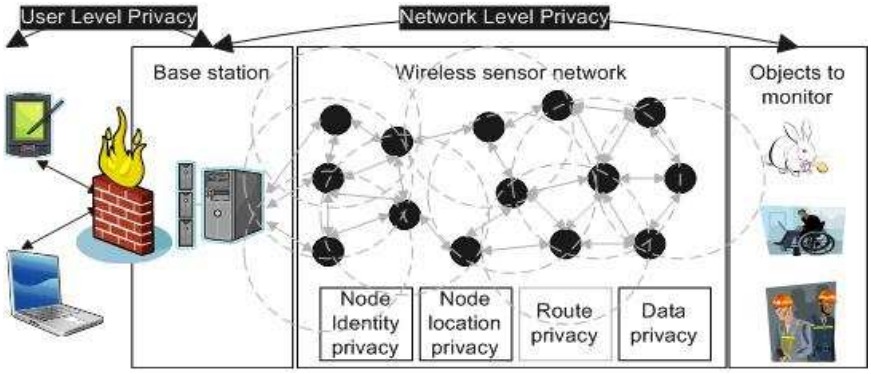


Figure 4.1: Typical WSN scenario

an ACK. Whenever a receiver (next hop) node successfully receives the packet it will send back an ACK to the sender. If the sender node does not receives an ACK during predefined threshold time, then the sender node will retransmit that packet.

4.2.2 Assumptions

For reason of scalability, it is assumed that no sensor node needs to know the global network topology, except, that it must know the geographical location of its own, its neighboring nodes and the base station.

It is also assumed that each sensor node in the network can share a unique secret key with the base station [9, 10]. These keys are periodically updated. It is also assumed that the public key of the base station is known to all the nodes in the network. Sensor nodes do not require their own public and private keys, because computation cost of a public and private keys at the sensor nodes is very high.

It is also assumed that sensor nodes are capable of performing encryption and decryption of the data by using any cipher algorithm such as DES, AES etc. This provides an additional layer of security.

4.2.3 Adversary Model

I have assumed that an adversary can mostly perform passive attacks (like eavesdropping [97]), since such attacks helps to conceal the adversary's presence in the network. Nevertheless the adversary is also capable of performing some active attacks like fabrication and packet drop attacks. I also assumed that the adversary is both device-rich and resource-rich [18]. These characteristics are defined below.

- Device-rich: the adversary is equipped with devices like antenna and spectrum analyzers, so that the adversary can measure the angle of arrival of the packet and received signal strength. These devices will help the adversary to find out the immediate sender of the packet and move to that node. This kind of hop-by-hop trace back mechanism will be carry out by the adversary until the actual sender node is reached.

- Resource-rich: the adversary has no resource constraint problems of computation power, memory and energy.

It is also assumed that the adversary has some basic domain knowledge like the range of identities assigned to the sensor nodes, the public key of the base station and information about the cipher algorithms used in the network.

4.3 Proposed Scheme

4.3.1 Concepts and Definitions

The first notion used in our algorithms is that of direction. The physical location of the base station is the reference point for each sensor node. Based on this reference point,

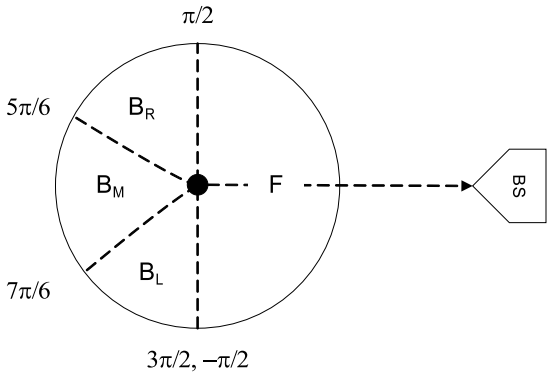


Figure 4.2: Neighbor node classification

each node classifies its neighboring nodes into four categories: 1) forward neighboring nodes (F), 2) right side backward neighboring nodes (B_r), 3) left side backward neighboring nodes (B_l), and 4) middle backward neighboring nodes (B_m). The objective of this categorization is to provide more path diversity as discussed in Section 4.3.2. A node x classifies its neighboring node y in following fashion:

$$C_{x,y} = \begin{cases} F & \frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ B_r & \frac{\pi}{2} < \theta \leq \frac{5\pi}{6} \\ B_m & \frac{5\pi}{6} < \theta \leq \frac{7\pi}{6} \\ B_l & \frac{7\pi}{6} < \theta < \frac{3\pi}{2} \end{cases} \tag{4.1}$$

where θ is the angle between the node x and its neighboring node y with respect to the line joining node x and the base station as shown in Figure 4.2.

The second notion used in my algorithms is that of trust. The definition of a trust here is based on my previous concept [See Chapter 3] that is defined as:

$$T_{x,y} = \left[100 \left(\frac{S_{x,y}}{S_{x,y} + U_{x,y}} \right) \left(1 - \frac{1}{S_{x,y} + 1} \right) \right] \tag{4.2}$$

where [.] is the nearest integer function, $S_{x,y}$ is the total number of successful interactions of node x with y and $U_{x,y}$ is the total number of unsuccessful interactions of node

x with y during last session.

I have used both these notions (direction and trust) in order to select reliable (non-malicious and non-faulty) secure paths for achieving robust route privacy. Direction information will help to forward packet to the destination in a timely manner and trust will help to forward the packets via reliable nodes.

4.3.2 Identity, Route, and Location Privacy (IRL)

My proposed identity, route and location privacy scheme works in two phases. The first is neighbor node state initialization phase, and the second is routing phase.

Route Privacy: In initialization phase, Let the node i have m neighboring nodes; out of which, t nodes are trusted. So, $0 \leq t \leq m$ and $M(t) = M(t_F) \cup M(t_{B_r}) \cup M(t_{B_l}) \cup M(t_{B_m})$. Here $M(t_F)$, $M(t_{B_r})$, $M(t_{B_l})$, and $M(t_{B_m})$ represent the set of trusted nodes that are in the forward, right backward, left backward, and middle backward directions, respectively. These neighbor sets ($M(t_F)$, $M(t_{B_r})$, $M(t_{B_l})$, and $M(t_{B_m})$) are initialized and updated whenever a change occur in neighborhood. For example, the entrance of a new node, change of a trust value, etc.

Whenever a node receives packet from the application layer (for the purpose of forwarding to the other node), the routing phase (Algorithm 2 for source node and Algorithm 3 for intermediate node) of IRL algorithm is called.

Whenever a source node (Algorithm 2) wants to forwards the packet, it will first check the availability of the trusted neighboring nodes in its forward direction set $M(t_F)$ (Line 2). If trusted nodes exists then it will randomly select one node as a next hop (Line 3) from the set $M(t_F)$ and forward the packet towards it (Lines 13:21). If there is no trusted node that exists in its forward direction, then the source node will check the availability of a trusted node in the right ($M(t_{B_r})$) and left ($M(t_{B_l})$) backward sets.

Algorithm 2 IRL - Routing at Source Node

$prev_{hop} \leftarrow \emptyset$; $next_{hop} \leftarrow \emptyset$;
if $M(t_F) \neq \emptyset$ **then**
    $next_{hop}(k)$ = Rand($M(t_F)$);
else
    if $M(t_{B_r}) \cup M(t_{B_l}) \neq \emptyset$ **then**
        $next_{hop}(k)$ = Rand($M(t_{B_r}) \cup M(t_{B_l})$);
    else if $M(t_{B_m}) \neq \emptyset$ **then**
        $next_{hop}(k)$ = Rand($M(t_{B_m})$);
    else
        Drop packet and Exit;
    end if
end if
Set $prev_{hop} = my_{id}$;
Form pkt $p = \{prev_{hop}, next_{hop}, seqID, \text{payload}\}$;
Create Signature and save in buffer;
Forward packet to $next_{hop}$;
Set timer $\Delta t = \frac{D}{d_{next_{hop}}} \times p_t$;
while $\Delta t = true$ **do**
    Signature remains in buffer;
end while
Signature removed from buffer;

If the trusted nodes are available then the source node will randomly select one node as a next hop (Line 3) from these sets and forward the packet towards it (Lines 13:21). If the trusted node also does not exist in these sets, then the source node will randomly select (Line 8) one trusted node from the backward middle set ($M(t_{B_m})$) and forward the packet towards it (Lines 13:21). If there are no trusted nodes available in all of the sets then the packet will be dropped (Line 9:10).

When an intermediate node (Algorithm 3) receives the packet (either from the source node or from another en-route node), it will first check whether the packet is new or old (Line 3). If it is new, then the node will first check the availability of the trusted node from the forward direction set (M_F) excluding the $prev_{hop}$ node if it belongs to forward set (Line 13). If trusted nodes exists in the forward set then the node will randomly select any one trusted node as a next hop (Line 14) and forward the packet towards it (Line 45). If there is no trusted node available in the forward direction, then it will check to which set the sender of the packet belongs to. For example, If the packet, forwarded by a node, belongs to the right backward set (Line 16), then it will first check whether the left or middle backward sets contain any trusted nodes or not (Lines 17:18). If yes, it will randomly select one node from those sets (Line 19) and forward the packet towards it (Line 45). If there is no trusted node in those two sets then the node will randomly select a trusted node from the right backward set ($M(t_{B_r})$) excluding the one from where the node received the current packet (Lines 20:21) and forward the packet towards it (Line 45). Similar operations will be performed, if the packet, forwarded by a node, belongs to the left (Lines 25:33) and middle backward or forward (Lines 34:43) sets. An example IRL routing scenario is shown in Figure 4.3.

This routing strategy, may result in the creation of a cycle (loop). However, due to the randomness in the selection of the next-hop and the presence of the different four

Algorithm 3 IRL - Routing at Intermediate Node

1: $next_{hop} \leftarrow \emptyset$;
2: $M_{temp} = \emptyset$
3: **if** Signature of new packet already exists in buffer **then**
4: $M_{temp} = \{M_{temp}\} + LasttimePrev_{hop}$
5: $M_{temp} = \{M_{temp}\} + LasttimeNext_{hop}$
6: Set $counter$ = $timesRecevied Before$ + 1;
7: Remove signature from buffer;
8: **if** $counter$ = 3 **then**
9: Drop packet and exit;
10: **end if**
11: **end if**
12: $M_{temp} = \{M_{temp}\} + prev_{hop}$
13: **if** $(M(t_F) - \{M(t_F) \bigcap M_{temp}\}) \neq \emptyset$ **then**
14: $next_{hop}(k)$ = Rand$(M(t_F) - \{M(t_F) \bigcap M_{temp}\})$;
15: **else**
16: **if** packet came from B_r **then**
17: M_{temp1} = $M(t_{B_l}) \cup M(t_{B_m})$
18: **if** $M_{temp1} \neq \emptyset$ **then**
19: $next_{hop}(k)$ = Rand(M_{temp1});
20: **else if** $M(t_{B_r}) \neq \emptyset$ **then**
21: $next_{hop}(k)$ = Rand$(M(t_{B_r}) - \{M(t_{B_r}) \bigcap M_{temp}\})$;
22: **else**
23: Drop packet and Exit;
24: **end if**
25: **else if** packet came from B_l **then**
26: M_{temp2} = $M(t_{B_r}) \cup M(t_{B_m})$
27: **if** $M_{temp2} \neq \emptyset$ **then**
28: $next_{hop}(k)$ = Rand$(M_{temp2} - \{M_{temp2} \bigcap M_{temp}\})$;
29: **else if** $M(t_{B_l}) \neq \emptyset$ **then**
30: $next_{hop}(k)$ = Rand$(M(t_{B_l}) - \{M(t_{B_l}) \bigcap M_{temp}\})$;
31: **else**
32: Drop packet and Exit;
33: **end if**
34: **else**
35: M_{temp3} = $M(t_{B_r}) \cup M(t_{B_l})$
36: **if** $M_{temp3} \neq \emptyset$ **then**
37: $next_{hop}(k)$ = Rand$(M_{temp3} - \{M_{temp3} \bigcap M_{temp}\})$;
38: **else if** $M(t_{B_m}) \neq \emptyset$ **then**
39: $next_{hop}(k)$ = Rand$(M(t_{B_m}) - \{M(t_{B_m}) \bigcap M_{temp}\})$;
40: **else**
41: Drop packet and Exit;
42: **end if**
43: **end if**
44: **end if**
45: Rest is same as Algorithm 2 from lines 13:21;

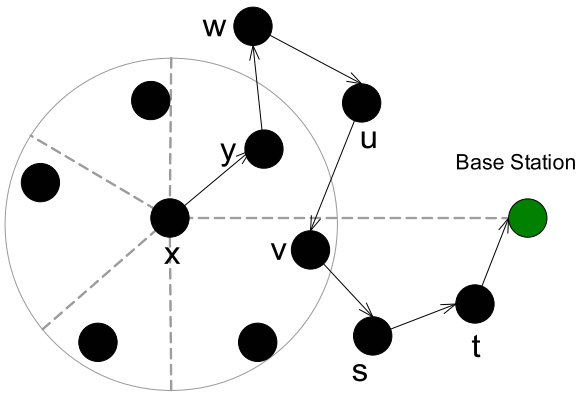


Figure 4.3: Sample routing scenario of IRL scheme

direction sets, the probability of creation of any cycle is very low. Nevertheless, in order to fully avoid the occurrence of the cycles, each node (prior to forwarding of a packet) will save the signature of the packet in the buffer for the Δt time, that is:

$$\Delta t = 2\left(\frac{D}{d} \times p_t\right) \tag{4.3}$$

where D is the distance between the forwarding node and the base station, d is the distance between the forwarding node and the next hop, and p_t is the propagation transfer time between the forwarding node and the next hop. This signature consists of two fields: 1) sequence number of the packet, and 2) the payload. The potential of the signature to compare and identify the same packet is detailed n the later section. Corresponding to this signature, three more fields are also stored in the buffer: 1) previous hop identity, 2) next hop identity where the packet is forwarded, and 3) counter, that tells how many times the same packet is received by the node. This information will later be used to get rid of any cycle. The size of the buffer is mainly dependent on the network traffic conditions. However, it is expected to be low due of fact that, the sensor nodes sent data either in periodic intervals or upon the occurrence of a some event.

If the node received the packet whose signature exists in the buffer (Algorithm 3,

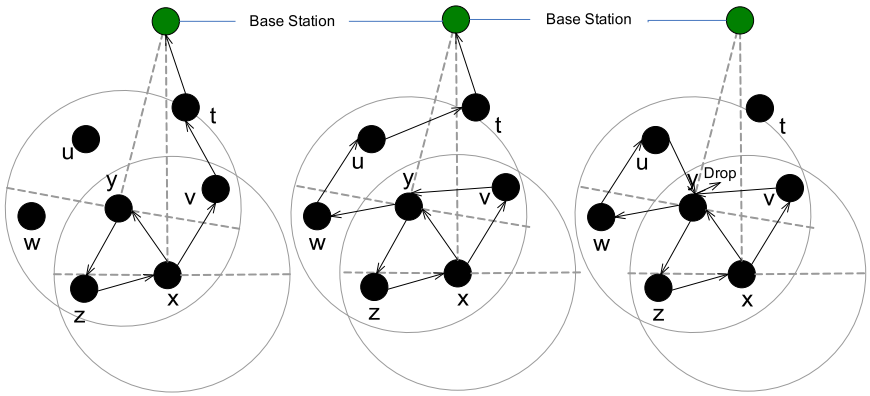


Figure 4.4: Three sample cycle detection and prevention scenarios

Lines 3:11), then including the previous hop node (Line 12), two other nodes will also be excluded from the selection of the next hop process: 1) the node from where last time the packet was received (Line 4) and 2) the node where last time the packet was forwarded (line 5). If the same packet is received three times by the same node (Line 8) then the packet will be dropped (Line 9). Three sample scenarios of the loop creation, detection and prevention are shown in Figure 4.4. Creation of loops and traversing of the packets in the backward direction is not a completely negative effect. Rather it provides positive effects in terms of strengthening the route and source location privacy. Because these effects will helps to increase the safety period [8], that is the time for an adversary to reach at the source node.

Identity Privacy: Whenever a node receives the packet p from the source node or en-route node then the receiving node will replace the previous hop's identity $prev_{hop}$ contained in the packet with its own (Algorithm 2, Line 13). After that, the node will get the next forwarding node $next_{hop}$ (as described earlier) and update the header of the packet $p = \{prev_{hop}, next_{hop}, \text{payload}\}$ (Line 14). After modification of the two header fields, the node will forward the packet (Line 16). In this way, all the intermediate

forwarding nodes replace the source and next hop's identity contained in the packet p. This process will go on until the packet reaches the base station.

Location Privacy: The neighboring nodes which are in each other's radio range can easily approximate the location of each other by measuring the received signal strength and the angle of arrival [56]. If the adversary is within the range of the source node, then adversary can easily estimate the location of the source. Once the packet has crossed the radio range of the original source node, then becomes very difficult for an attacker to estimate the location of the node either in terms of the physical distance or in terms of the number of hops of an original source node. The main reason for this is that the path selection is random and packets are forwarded by only trusted nodes which only contain the information of the last and the next hop.

4.3.3 Reliable Identity, Route, and Location Privacy (r-IRL)

It is also possible that some applications require more reliability in terms of packet reachability; and the packet could be dropped due to either network congestion or due to malicious behavior of an en-route node. Thus, in order to achieve more reliability, the packet should be forwarded from multiple paths simultaneously, which will give trustworthiness in the sense that at least the packet should reach the base station by any one of the paths, although, this may increase some communication overhead.

My reliable IRL (r-IRL) algorithm is the extended version of my proposed IRL algorithm in which I introduce one more parameter, reliability r. The source node i will multi-cast a packet to all r randomly selected neighboring trusted nodes that are in the forward direction. If there are no adequate trusted nodes present in the forward direction then it will select the remaining trusted nodes from the backward direction. The rest of the mechanism of the r-IRL algorithm is the same as the IRL algorithm.

4.3.4 Data Privacy

The payload contains identity of the source node (ID_x) and the actual data (d). Identity is encrypted with the public key (k_{bs}) of the base station and data is encrypted with the secret key ($k_{x,bs}$) shared between the sender node and the BS. Both are appended with the payload as shown below:

$$\text{payload} = [E(ID_x, k_{bs}), E(d, k_{x,bs})]$$

If we assume that the adversary knows the range of identities assigned to the sensor nodes, public key of the base station and information about cipher algorithm used in the network. Then, an adversary can successfully obtain the identity of the source by performing simple brute-force search attack [98] by comparing the pattern of encrypted identity with a known range of identities. Therefore in order to provide protection against brute-force search attack, I append a random number (R_n) (equivalent to the size of identity) with the identity of a node and then perform encryption. Now the payload is:

$$\text{payload} = [E(ID_x||R_n, k_{bs}), E(d, k_{x,bs})]$$

where $||$ is the append operation.

This approach provides several benefits as follows. Firstly, data secrecy is achieved in the presence of identity anonymity. This feature is not available in earlier proposed privacy schemes. Secondly, the base station will not only able to get the identity of actual source node but also it provides message authentication.

4.4 Analysis and Evaluation

4.4.1 Security Resiliency Analysis

Suppose we have an adversary $\mathcal{A}$ whose wish is to defeat my privacy protocols and guess the original source node. We will distinguish between two kinds of nodes. A source node is the node which is the original sender of a packet q and a forwarding node is a node which forwards a packet to another node until it reaches the destination. Hence the source node is also a forwarding node. The adversary's goal is to find out the source node. This analysis assumes that we are using IRL algorithm including our proposed data privacy mechanism. So if the adversary sees a packet, it will trivially know the identity of the last forwarding node (which could possibly be the sender node).

We will deal with separate cases. Case 1 is when the adversary is close to the base station and can eavesdrop on any packet received by the base station. Case 2 deals with the case when the adversary can see any packet within the radio range of a particular node. Case 3 extends this into two or more nodes.

An adversary will try to solve the following problem: Given a packet q and a subset of nodes N', find out the sender node s. In other words the algorithm for the adversary takes two inputs and outputs a node s'; Namely $\mathcal{A}(q, N') = s'$. If $s' = s$, the adversary wins and is successful in defeating our protocol. We have to find: $P(\mathcal{A}(q, N') = s)$ that is the probability of an adversary to find out the sender node.

Notations and definitions: Denote a generic node by m. The set of neighbors of m is denoted by N_m, which also includes m itself. The number of forward and backward nodes of m is denoted by m_f and m_b respectively. If a node a is a backward node of m, then we denote it as $a \rightarrow m$. We say that a node a is in the backward set of node m, if $a \rightarrow a_1 \rightarrow \ldots a_r \rightarrow m$, for some nodes $a_1, \ldots a_r$ where $r \geq 0$. For compact notation

we will denote this as $a \rightarrow^r m$, if the IDs of the intermediate nodes are not significant. We will also use the notation $\rightarrow^r m$ to denote a generic node, who is r links (hops) away from m. Define the backward set C_m of m as $C_m = \{a | a \rightarrow^r m, r \geq 0\}$, that is the set of all the possible nodes such that they have a forward link to m. Denote the base station as B. It will also be seen as another node. Let the total number of nodes in the network excluding the base station be N. We will use the term "adversary is in possession of a node" to indicate that the adversary can passively listen to any communication within the radio range of that node.

Claim 1: Suppose A is in possession of B. Let B_b be the number of backward nodes of the base station (nodes one hop away from the base station). Then for any packet q received by B:

$$P\left(A\left(q, N\right) = s\right) = \frac{B_b + 1}{N} \tag{4.4}$$

Proof. The adversary can always know the ID of the last forwarding node. Let B_b be the number of backward nodes to the base station. The packet could only have come from one of the nodes in $N_B - \{B\}$ (which only contains backward nodes to B). Since the nodes are just a hop away from the BS, so they will not send the packet to another node. Hence for large N we have:

$$\begin{aligned} P\left(A\left(q, N\right) = s\right) &= P\left(A\left(q, N\right) = s | s \in N_B - \{B\}\right) \times P\left(s \in N_B - \{B\}\right) + \\ &\quad P\left(A\left(q, N\right) = s | s \notin N_B - \{B\}\right) P\left(s \notin N_B - \{B\}\right) \\ &= 1 \cdot \frac{B_b}{N} + \frac{1}{N - B_b - 1}\left(1 - \frac{B_b}{N}\right) \\ &\approx \frac{B_b}{N} + \frac{1}{N - B_b}\left(1 - \frac{B_b}{N}\right) = \frac{B_b + 1}{N} \end{aligned}$$

□

Now let us assume that A is in possession of a node m in the network. Let us exclude the possibility that a packet will be sent backwards during its course to the base station,

since the probability of it happening is very small. Furthermore even if we consider it, it will decrease the probability of success of the adversary since there would be more possible nodes. Thus in this scenario our result would be like an upper bound on the adversary's limitations.

Claim 2: Suppose A is in possession of a node m. Let $c = |C_{\rightarrow^2 m}|$ be denote the number of backward nodes in backward set $C_{\rightarrow^2 m}$ of some node $\rightarrow^2 m$. Then,

$$P\left(A\left(q,N\right)=s\right)=\frac{m_f+m_b+1}{N}+\frac{1}{c+1}\left(1-\frac{m_f+m_b+1}{N}\right) \tag{4.5}$$

Proof. Since the adversary is in possession of a node m, it knows its backward and forward nodes. Furthermore, if any of these nodes including the node m itself is the sender of a packet q, then the adversary will know. This is true since the adversary can see all the incoming packets to the node m and to its neighbor nodes (the forward and the backward nodes). Thus it can see if the payload of q is not equal to the payload of any q' being received by these nodes in a given interval of time. If this is the case, then the adversary will know the sender.

Now if none of the nodes in N_m are the senders, then the packet was forwarded by a node i which is two hops away from m. The adversary knows the ID of that node through the packet q. Thus the adversary makes a list of all the possible backward nodes in the backward set of i. Let that number be denoted by c. Notice that the node i could also be the possible sender. Hence the total number of possible senders would be $c+1$. We have:

$$\begin{aligned}P\left(A\left(q,N\right)=s\right)&=P\left(A\left(q,N\right)=s|s\in N_m\right)P\left(s\in N_m\right)+\\&\quad P\left(A\left(q,N\right)=s|s\notin N_m\right)P\left(s\notin N_m\right)\\&=\frac{m_f+m_b+1}{N}+\frac{1}{c+1}\left(1-\frac{m_f+m_b+1}{N}\right)\end{aligned}$$

□

Now, suppose the adversary is in possession of two nodes at the same time m_1 and m_2. We can safely assume that $N_{m_1} \cap N_{m_2} = \varphi$, since it would be more advantageous to the adversary to cover nodes with non overlapping radio ranges. The adversary will always know whenever any node in N_{m_1} or N_{m_2} is the sender of a packet. How about the case when they are not the senders? There could be two possible cases: without loss of generality, first assume that $m_2 \in C_{m_1}$. If the packet q was received by some node in N_{m_1} and was received byk some node in N_{m_2} before, then the adversary had already checked it when the packet was sent to a node in N_{m_1}. Thus the adversary need only check packets received in N_{m_1} which were not received by N_{m_2}. In this case, the sender cannot be in N_{m_2}. In any case, the adversary has to find out the backward sets of $\rightarrow^2 m_1$ or $\rightarrow^2 m_2$, depending on where the packet was received. Since the network traffic is uniformly distributed, therefore the probability of a packet being received at the two sets is the same. In case $m_2 \notin C_{m_1}$, then the adversary has no real advantage except that it can see packets at two disjoint locations in the network. Thus we only state the case when $m_2 \in C_{m_1}$. We have the following result:

Claim 3: Suppose the adversary is in possession of two nodes m_1 and m_2. Assume further that $m_2 \in C_{m_1}$. Let $c_1 = |C_{\rightarrow^2 m_1}|$ and $c_2 = |C_{\rightarrow^2 m_2}|$ then:

$$P\left(A\left(q,N\right)=s\right) = \frac{\left|N_{m_1}\right|+\left|N_{m_2}\right|}{N} + \frac{1}{2}\left(\frac{1}{c_1+1-\left|N_{m_2}\right|}+\frac{1}{c_2+1}\right)\left(1-\frac{\left|N_{m_1}\right|+\left|N_{m_2}\right|}{N}\right) \tag{4.6}$$

In general, we have:

Claim 4: Let us assume that A is in possession of k nodes $m_k \rightarrow^{r_1} \cdots \rightarrow^{r_{k-2}} m_2 \rightarrow^{r_{k-1}} m_1$ and let m_f and m_b denote the average number of forward and backward nodes averaged over all the k nodes. Let $t = m_f + m_b + 1$. Let for $1 \leq i \leq k$, $c_i = |C_{\rightarrow^2 m_i}|$, then:

$$P\left(A\left(q,N\right)=s\right) = \frac{kt}{N} + \frac{1}{k}\left(\frac{1}{c_1+1-(k-1)t}+\frac{1}{c_2+1-(k-2)t}\cdots+\frac{1}{c_k+1}\right)\left(1-\frac{kt}{N}\right) \tag{4.7}$$

Table 4.1: Neighbor list table at sensor node

Neighbor nodeID (Integer)	Direction	Successful interactions (Integer)	Unsuccessful interactions (Integer)	Trust state (Boolean)
1	F (00)	10	4	trusted (true)
2	B_R (01)	2	8	untrusted (false)
⋮	⋮	⋮	⋮	⋮
M	B_L(11)	5	0	trusted (true)

Observations: The probability is lowest when the adversary is actually at the base station. If the adversary has more nodes in possession, the probability increases linearly, with more success rate when the nodes are actually connected. This also shows that if a packet originates from any node which does not have a backward node, the adversary will always know the sender.

4.4.2 Memory Consumption Analysis

Each sensor node needs to maintain one table that contains the list of neighboring nodes, their direction and their trust states as shown in Table 4.1. Node identity can be represent in two bytes [10, 99], four sets of directions can be easily represent in 2 bits, number of successful and unsuccessful interactions can be represent in two bytes each and trust state can be represent as a boolean variable (1 bit). Therefore the size of each record is 51 bits. If we assume that the node has M neighboring nodes then the total size of the table will be $51 \times M$ bits.

Table 4.2: Memory requirement in bits

PFR [8]	(16+1)M bits
PSR [18]	(16+16+1)M bits
SAS [95]	K(4M+2N)+16M bits
CAS [95]	K(6+7M)+16M bits
IRL / r-IRL	$51 \times M$ bits

Table 4.2 shows the memory requirement of various privacy schemes, in which M represents the neighborhood size, K represents pseudonym space and N is the total number of nodes in the network. In the Phantom Flood Routing (PFR) [8] scheme, each sensor node needs to maintain the list of neighbor nodes and these neighbor nodes are divided into two sets. Here I assume that identity of a node is represent by two bytes, and set is distinguish by a single bit. So the total memory required by each node in the PFR scheme is (16+1)M bits. In the Phantom Single-path Routing (PSR) [18] scheme, each node maintain the list of neighbor nodes, hop count (2 bytes), and set identification (1 bit). Therefore, the total memory required by each node in the PSR scheme is (16+16+1)M bits. In the SAS scheme, each node needs K(4M+2N)+16M bits of memory. Here M represents the neighborhood size, K represents pseudonym space and N is the total number of nodes in the network. For the CAS scheme, each node requires K(6+7M)+16M bits of memory. For more details about the SAS and CAS schemes, please refer paper [95].

Let us assume that the sensor node has ten neighbor nodes, then the total memory required by the sensor node by the PFR, PSR, IRL, CAS and SAS is 21.25, 41.25, 63.75, 628 and 1940 bytes respectively as shown in Figure 4.5.

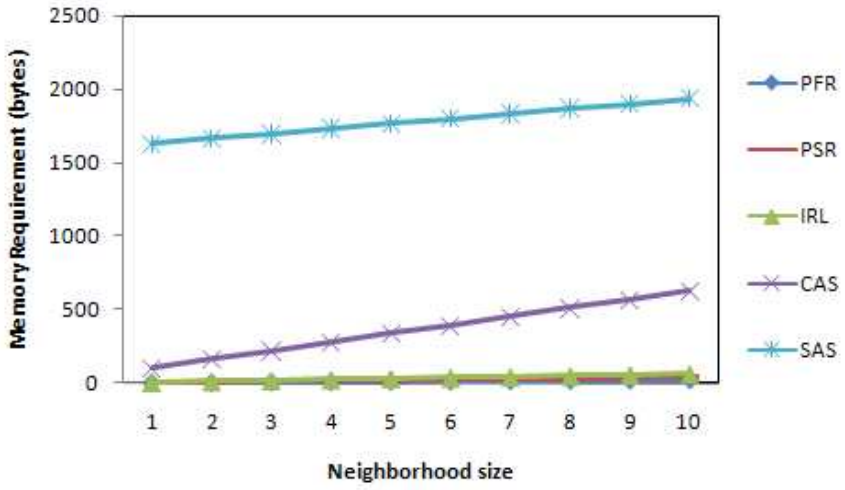


Figure 4.5: Memory consumption analysis: N= 100; K=8 bytes

4.4.3 Energy Consumption Analysis

I have implemented IRL and r-IRL routing schemes on Sensor Network Simulator and Emulator (SENSE) [94]. At the application layer I used constant bit rate component (CBR) that generate constant traffic during simulation between randomly selected source node(s) and the base station. For the simplicity, I assumed that both sensor nodes and the base station are static. Other simulation parameters are given in Table 4.3.

I have compared my proposed IRL and r-IRL algorithms with the four variations of phantom routing schemes [8, 18] that are:

1. Phantom single path routing scheme with hop-based approach (PSR-hop).
2. Phantom single path routing scheme with sector-based approach (PSR-sec).
3. Phantom flood routing scheme with hop-based approach (PFR-hop).
4. Phantom flood routing scheme with sector-based approach (PFR-sec).

Table 4.3: Simulation parameters

Network specific	Number of nodes	300
	organization (grid fashion)	15x20
	Distance b/w nodes	50 units
	Mobility of nodes	zero
Node specific	Sensor node's Initial battery	1x10^6J
	Power consumption for trans.	1.6W
	Power consumption for recv.	1.2 W
	Idle power consumption	1.15W
	Carrier sense threshold	3.65e-10W
	Receive power threshold	1.55e-11W
	Frequency	9.14e8
	Trans. & Recv. antenna gain	1.0
Protocol & Application specific	Application	CBR
	Reliability param. r for r-IRL	3
	h_{walk} param. for PFR & PSR	10

I did not compared my schemes with the SAS and CAS [95] schemes because the authors did not propose any routing strategy.

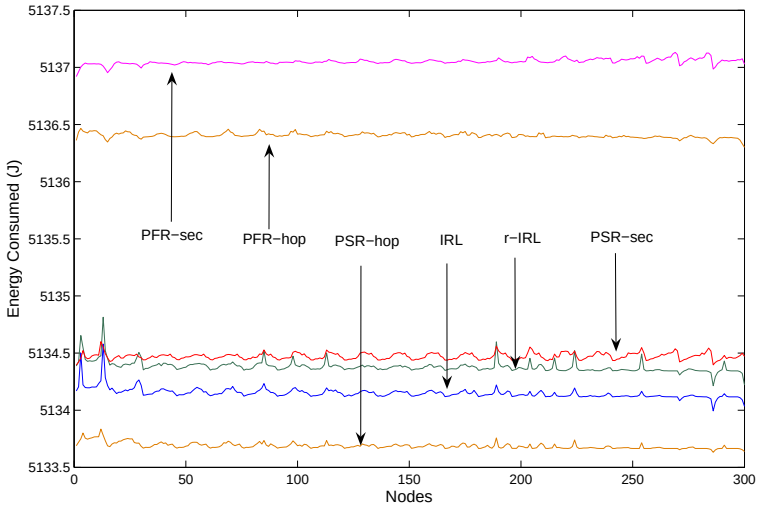
The energy consumption analysis with different scenarios are shown in Figure 4.6. For the r-IRL scheme I select $r = 3$, which means a single packet will reach to the

destination via three different routes simultaneously. For phantom routing schemes, I select parameter h_{walk}=10 (as recommended in [8]). Figure 4.6 clearly indicates that, the IRL and r-IRL schemes consume less energy as compared to the PSR-sec, PFR-hop and PFR-sec schemes but slightly consume higher energy as compared to the PSR-hop scheme. This is due to the fact that the IRL and r-IRL algorithms provides more path diversity and packets some times took longer paths.

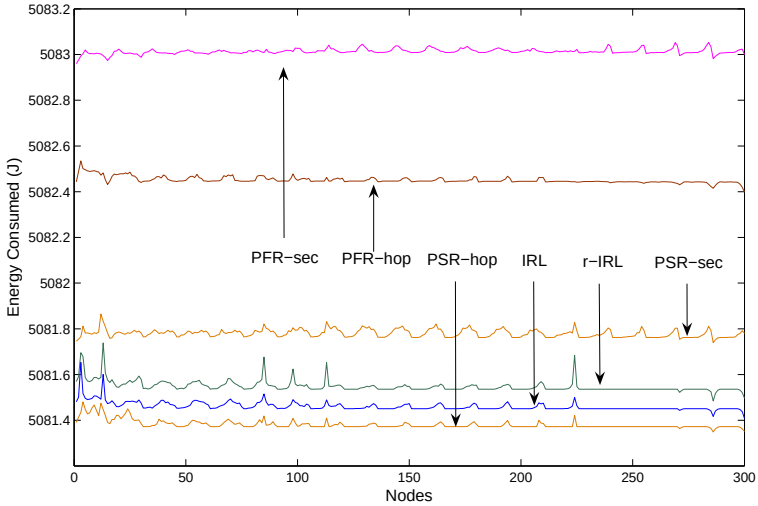
4.4.4 Path Diversity Analysis

Longer paths incur delay while shorter paths leads towards a limited or weak route privacy. In order to analyze this behavior, I have organized 300 sensor nodes in a 10 by 30 grid manner. The rest of simulation parameters are given in Table 4.3. In the simulation, a single source node (ID: 224) generates 100 data packets for the base station. Figure 4.7 shows the path diversity (in terms of path length) of the IRL, PSR-hop and PSR-sec schemes. The average path taken by the PSR-hop, IRL and PSR-sec is 22.12, 36.81 and 38.17, respectively. It indicates that the IRL scheme incurs more delay as compared with the PSR-hop scheme and less delay as compared with the PSR-sec scheme. This figure also indicates that the IRL scheme has more path variation as compared with the other schemes, which creates more difficulties for the adversary to trace back the source from the captured packets.

Figure 4.7 also shows that some packets took longer paths in the IRL scheme as compared with others. This is due to the fact that the source or en-route node did not find any trusted node in its forward direction, so the packet is relayed back in the backward direction. If we assume that each node has p probability to be trusted and all probabilities are independent of each other, then the total probability P_b for a node i to relay the packet



(a) Source nodes:5



(b) Source nodes:10

Figure 4.6: Energy consumption analysis: Simulation time:5000

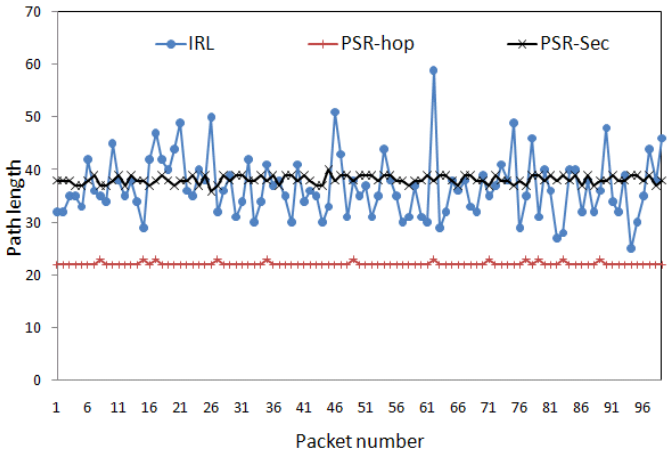


Figure 4.7: Path diversity of privacy schemes

in the backward direction is:

$$P_b(i) = \prod_{k=1}^{m_f} (1 - p_k) \tag{4.8}$$

where m_f represents the number of nodes in the forward direction. Figure 4.8 shows the result of 100 simulation runs in which I have assumed that each node has equal probability to be trusted and un-trusted. It shows that, as the neighborhood size increases, the probability of the packet to move in the backward direction decreases sharply.

4.4.5 Discussion

From the memory, energy and path diversity analysis, I observe that my solutions are not a very optimal solutions especially with respect to the PSR-hop scheme. However, at a modest cost of memory and energy, it provides full network level privacy as compared with the other existing schemes. This cost is justifiable because I have additionally achieved trustworthiness and reliability (in terms of packet reach-ability). With this level of resource consumption, our solutions can easily be used on real sensor nodes, for

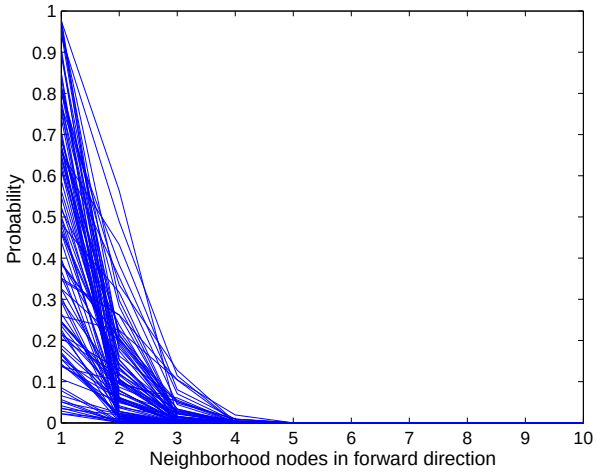


Figure 4.8: Probability of a packet to move in the backward direction

example, MICA2 sensor node has ATMega 128L micro-controller (8 MHz @ 8 MIPS), 512 Kbytes measurement (serial) flash, and 4 Kbytes EEPROM [91].

4.5 Summary

Existing privacy schemes of WSNs only provides partial network level privacy. Providing full network level privacy is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g. energy, and memory), sensor network (e.g. mobility, and topology) and QoS issues (e.g. packet reach-ability, and timeliness). Therefore, I proposed first full network level privacy solution that is composed of two new identity, route and location privacy algorithms and data privacy mechanism. At the modest cost of energy and memory, my solutions additionally provides trustworthiness and reliability. I also proved analytically that my solutions provides protection against an adversary who is capable of performing privacy disclosure attacks, e.g. hop-by-hop trace backing.

Chapter 5
Lightweight Security Component

5.1 Introduction

Wireless networks are relatively more vulnerable to security attacks than wired networks due to the broadcast nature of communication [81]. In order to implement security mechanism in sensor networks, we need to ensure that communication overhead is less and consumes less computation power. With these constraints it is impractical to use traditional security algorithms and mechanisms meant for powerful workstations.

Sensor networks are vulnerable to a variety of security threats, such as DoS, eavesdropping, message replay, message modification, malicious code, etc. In order to secure sensor networks against these attacks, we need to implement message confidentiality, authentication, message integrity, intrusion detection and some other security mechanisms. Encrypting communication between sensor nodes can partially solve the problems, but it requires a robust key exchange and distribution scheme.

In general, there are three types of key management schemes [22, 23]: Trusted Server scheme, self enforcing scheme and key pre-distribution scheme. Trusted server schemes relies on a trusted base station, that is responsible for establishing the key agreement between two communicating nodes as described in [24]. It uses symmetric key cryptography for data encryption. The main advantages of this scheme are, it is memory

efficient, nodes only need to store single secret key and it is resilient to node capture. But the drawback of this scheme is that it is energy expensive, it requires extra routing overhead in the sense that each node need to communicate with base station several times [23]. Self enforcing schemes use public key cryptography for communication between sensor nodes. This scheme is perfectly resilient against node capture and it is fully scalable and memory efficient. But the problem with the traditional public keys cryptography schemes such as DSA [25] or RSA [26] is the fact that they require complex and intensive computations which is not possible to perform by sensor node having limited computation power. Some researchers [27, 1] use Elliptic curve cryptography as an alternative to traditional public key systems but still not perfect for sensor networks. Third scheme is key pre-distribution scheme based on symmetric key cryptography, in which limited number of keys are stored on each sensor node prior to their deployment. This scheme is easy to implement and does not introduce any additional routing overhead for key exchange. The degree of resiliency of node capture is dependent on the pre-distribution scheme [23].

Quite recently some security solutions have been proposed in [9, 11, 10, 40] especially for WSNs but each suffers from various limitations, such as higher memory and power consumptions that are discussed in Section 5.4. Keeping all these factors in mind, I propose a lightweight security protocol (LSec) for WSNs. LSec combines the features of trusted server scheme and self enforcing security schemes. My main contribution is the designing and implementation of LSec that provides

- Authentication and authorization of sensor node.
- Simple secure key exchange scheme.
- Confidentiality of data.

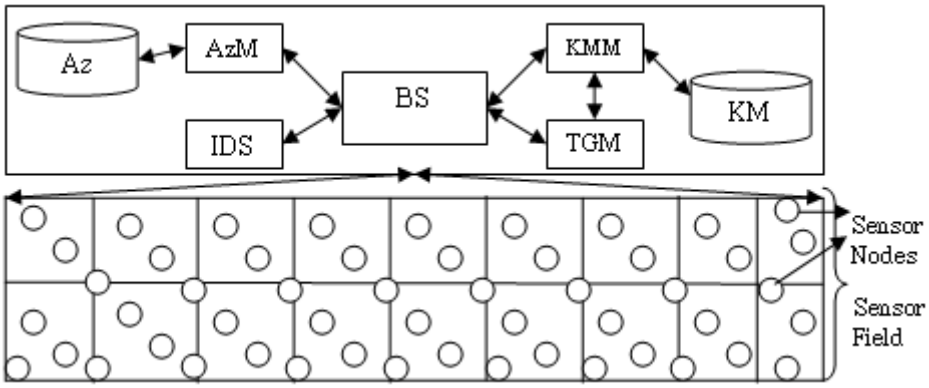


Figure 5.1: LSec system architecture

5.2 Lightweight Security Protocol (LSec)

The basic objective of the LSec is to provide lightweight security solution for WSNs, where all nodes can communicate with each other. LSec can support both static and mobile environment, which may contain single and multiple Base Stations (BS). Basic system architecture is shown in Figure 5.1. LSec uses both symmetric and asymmetric schemes for providing secure communication in WSNs.

Key Management Module (KMM) is used to store public and shared secret key of each node with the BS to the database. Token Generator Module (TGM) is used to generate the tokens for the requesters, which will be further used by the other communicating party for the authentication of the requester node. Authorization Module (AzM) is used to check whether a particular node is allowed to communicate with other node or not. Basic assumptions and rules of LSec are given below.

5.2.1 Assumptions

1. Base Station (BS) is the trusted party and it will never be compromised. Compromising the BS can render the entire sensor network useless, and it is the only point from where sensor node can communicate with external networks.

2. Only BS knows the Public keys (Pk) of all the sensor nodes in the network. Communicating nodes will know each other's public key during the time of connection establishment.

5.2.2 Rules

- Asymmetric scheme will only be used for sharing ephemeral secret key between communicating nodes.

- For every session, new random secret key will be used.

- Data will be encrypted by using symmetric schemes, because these schemes are considered to be executed three to four times faster than asymmetric schemes [63].

5.2.3 LSec Packet Format

LSec packet format is shown in Table 5.1. Currently, LSec uses seven types of packets, '*Request*', '*Response*', '*Init*', '*Ack*', '*Data*', '*Update Group Key*' and '*Alert*' packet. All seven packets are distinguished by '*type*' field in the LSec packet. ID_{src} field contains the identity of sending node. Encrypted portion contains the information depending upon the type of the packet, as shown in Table 5.1.

The distribution of bits to different fields (as shown in Table 5.2), introduces some upper limits, such as, size of source address is of 2 bytes, it means LSec works only

Table 5.1: LSec: *Type* field description

Type	ID_{src}	Encrypted Portion
Request	Any (sensor node)	EK_{A-BS} (Intended-ID_{dest} , N)
Response	BS	$EK_{A-BS}(R-type$, Intended-$ID_{dest}, N, Pk, token\|R)$
Init	Any (sensor node)	$EK_B^+(N, Pk$, token)
Ack	Any (sensor node)	$EK_A^+(N, sk)$
Data	Any (sensor node)	EK_{sk} (data)
UpdateGroupKey	Any CH	EK_G(GroupID, new Key), MAC
Alert	Any CH	EK_{CH-BS} (Alert-type), MAC

EK_{A-BS} = Encrypt with the secret key shared between node A and BS

EK_A^+ = Encrypt with the public key of node A

EK_B^+ = Encrypt with the public key of node B

EK_{sk} = Encrypt with the shared secret key

EK_G = Encrypt with group key

EK_{CH-BS} = Encrypt with the secret key shared between cluster head and BS

$R-type$ = Response type (positive or negative response)

R = Reason of negative acknowledgment

Intended-ID_{dest} = ID of intended destination

Pk = public key

ID_{src} = ID of source node

N = Nonce (Unique random number)

MAC = Message Authentication Code

CH = Cluster head

Table 5.2: Distribution of bits to different fields of LSec

Field	Size	Field	Size
Type	4 bits	Public and Private key	128 bits
ID_{src}, ID_{dest}	16 bits	Secret key	64 bits
Nonce (N)	23 bits	token	4 bytes
R-type	1 bit	data	30 bytes

in the environment where number of sensor nodes are not exceeding 2^{16}. Length of a Nonce (unique random number) field is of 3 bytes, so LSec can allow maximum of 2^{24} connections at a time. The length of public key and private key is of exactly 128 bits and the length of secret key is of exactly 64 bits. Only stream cipher encryption algorithms are allowed to use because of a fixed length size of packets. MAC is of 64 bits.

5.2.4 Procedure

LSec works in three phases, authentication and authorization phase, key distribution phase, and data transmission phase. Authentication and authorization is performed during the exchange of "*Request*" and "*Response*" packets by using symmetric scheme. Key distribution phase involves sharing of random secret key in a secure manner by using asymmetric scheme. In this phase "*INIT*" and "*ACK*" packets will be exchanged. Data transmission phase involves transmission of data packet in an encrypted manner.

Let us suppose node A wants to communicate with the node B. It will first send a request packet to the BS for receiving token and public key of a node B. The request packet is encrypted with the secret key shared between the node A and BS. The BS first checks in the database via AzM that whether a node A has rights to establish connection with a node B or not. If yes, it generates the token which will be further used by the

node B for the authentication of a node A. That token is encrypted with the secret key shared between node B and BS, so that node A will not able to decrypt token. The BS will send back a response packet that contains token, public key of node B and nonce (Unique Random Number) that was there in request packet. Nonce will ensure node A that packet came from genuine BS. When node A gets the positive response from the BS, it sends the *INIT* packet to the node B that contains nonce, its own public key and token generated by the BS. The whole *INIT* packet is encrypted with the public key of the node B. When node B gets *INIT* packet it first checks token, if it is correct, it will generate the secret key and send it back to the node A in an encrypted manner. When node A gets the *ACK* packet, it deletes the public key of a node B from its memory and sends data to the node B by using new session secret key. When data transmission is complete, both nodes delete that session key. For group communication, each node uses the group secret key for data transmission in a secure manner. Cluster head will update this key after periodic interval.

5.3 Simulation and Performance Analysis

I have tested LSec protocol on Sensor Network Simulator and Emulator (SENSE) [94]. In sensor node I introduce the middleware between application layer and network layer as shown in Figure 5.2.

The middleware uses LSec for the enforcement of security in the sensor network. At application layer I use constant bit rate component (CBR) that generates constant traffic during simulation between two communicating sensor nodes. For the demonstration and performance evaluation of LSec, CBR is run with and without LSec. I randomly deploy 100 sensor nodes plus one Base station (BS) in 1000 by 1000 terrain. Basic simulation parameters employed are described in Table 5.3.

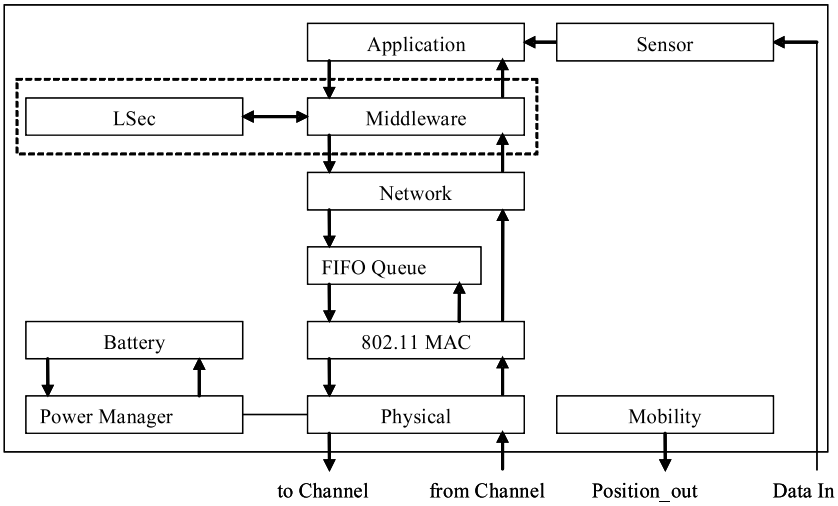


Figure 5.2: Sensor node architecture

Table 5.3: Simulation Parameters

Terrain	1000x1000
Total Number of Nodes	101 (including BS)
Initial battery of each sensor node	$1 \times 10^6 J$
Power consumption for transmission	$1.6W$
Power consumption for reception	$1.2W$
Idle power consumption	$1.15W$
Carrier sense threshold	$3.652e^{-10}W$
Receive power threshold	$1.559e^{-11}W$
Frequency	$9.14e^8$
Transmitting and Receiving antenna gain	1.0

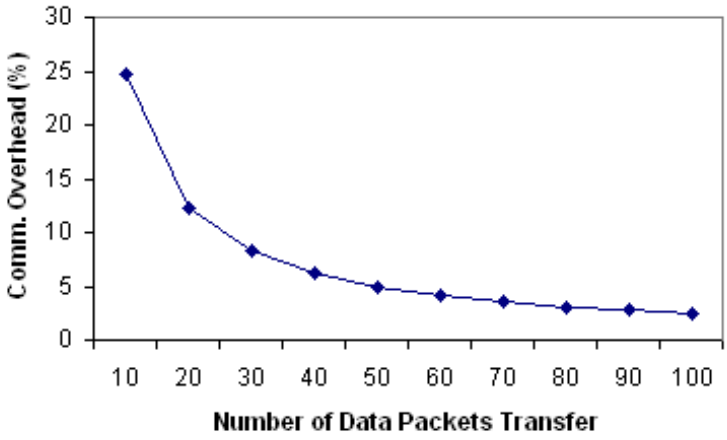


Figure 5.3: Communication overhead of LSec: Data packet size = 30 bytes

5.3.1 Communication Overhead Analysis

In simulation scenario, application sends data packets of size 30 bytes in a periodic interval. The overall communication overhead of LSec for one to one communication decreases with the increase in transfer of number of the data packets as shown in Figure 5.3. Communication Overhead ($CO\%$) is calculated as

$$CO(\%) = (\frac{Nc * 74.125}{\sum_{i=1}^{n} N_i^P * 30}) * 100 \tag{5.1}$$

Where as Nc is the total number of connections. N_i^P is the number of packets transferred by node i. I multiplied 74.125 bytes to Nc because for every connection LSec exchange four control packets (*Request*, *Response*, *Init*, and *Ack*) during the authentication, authorization and key exchange phase whose cumulative size is 74.125 bytes. The size of each data packet is 30 bytes.

5.3.2 Power Computation Analysis

Power computation primarily depends upon the kind of symmetric and asymmetric scheme. If we assume that computation power required for symmetric encryption and decryption scheme is CSE and CSD respectively and computation power of asymmetric encryption and decryption scheme as CAE and CAD respectively, then the total power consumption required by single node during first two phases is

$$PowerComputation = (CSE + CSD) + (CAE + CAD) \tag{5.2}$$

Computation power required by a single node during data transmission phase is calculate as,

$$PowerComputation = (TNSP \times CSE) + (TNRP \times CSD) \tag{5.3}$$

Where TNSP is the total number of data packets sent and TNRP is the total number of data packets received.

5.3.3 Memory Consumption Analysis

Every sensor node needs to store only six keys, three of them are permanent and three are ephemerals. Permanent keys consist of one public key (self), one private key and one public key of the BS. Ephemerals keys consist of group key, public key of other node and session secret key. In order to save these keys only 72 bytes are needed. Details are given in Table 5.4. This approach will make sensor network memory efficient.

5.3.4 Energy Consumption Analysis

The main source of energy consumption at the sensor node is its transmission and reception cost. I used SENSE [94] that consumes energy in four different modes: TRANSMIT, RECIEVE, IDLE, and SLEEP. Energy consumption rate of each mode is given

Table 5.4: Memory requirement of LSec

S/No.	Keys	Size (in bytes)
Permanent Keys		
1	Public key of node	16
2	Private key of node	16
3	Shared secret key b/w Node & BS	8
Ephemeral Keys		
4	Group Key	8
5	Public key of other node	16
6	Session key	8
Total memory required		72 bytes

in Table 5.3. For each connection, LSec exchanges four control packets (*Request*, *Response*, *Init*, and *Ack*) of cumulative size 74.125 bytes required for authentication, authorization and key exchange mechanism. That is an acceptable trade-off between energy and security. Simulation result of energy consumption is shown in Figure 5.4.

5.3.5 Resilience Against Node Compromise

Single node compromised will not expose the whole communication in network. Only the communication links that are established with compromised node will expose the network. Let us suppose N_{cn} is the set of nodes that establish connections and N_{cp} is the set of compromised nodes. Then $N_{cn} \bigcap N_{cp}$ will gives us the set of nodes that are compromised as well as connected. Then maximum number of connections can be exposed only if all compromised nodes are connected to un-compromised nodes. On the other hand, minimum numbers of links can be exposed only if all compromised nodes

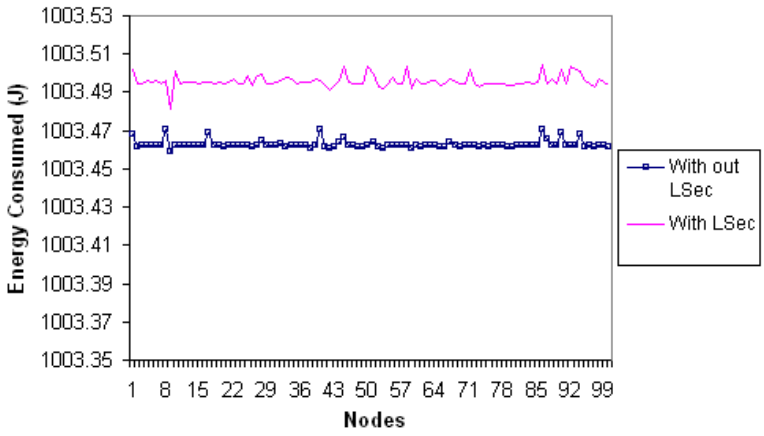


Figure 5.4: Energy consumption of LSec

are connected with each other.

$$Max : N_{cn} \bigcap N_{cp} \tag{5.4}$$

$$Min : \begin{pmatrix} \frac{N_{cn} \bigcap N_{cp}}{2} & foreven \\ (\frac{N_{cn} \bigcap N_{cp}+1}{2}) & forodd \end{pmatrix} \tag{5.5}$$

If we assume that sensor network consists of 1000 nodes and total 500 connections established between pair of nodes then the total links that can be minimum and maximum compromised is shown in Figure 5.5.

5.4 Comparison of LSec with other security solutions

Comparison of all above discussed schemes with LSec is given in Table 5.5. I provided comparison from the perspective of memory requirement, transmission cost, and some other basic security parameters such as authentication, authorization, confidentiality, etc. Data integrity is generally handled at link layer with the help of some hashing schemes,

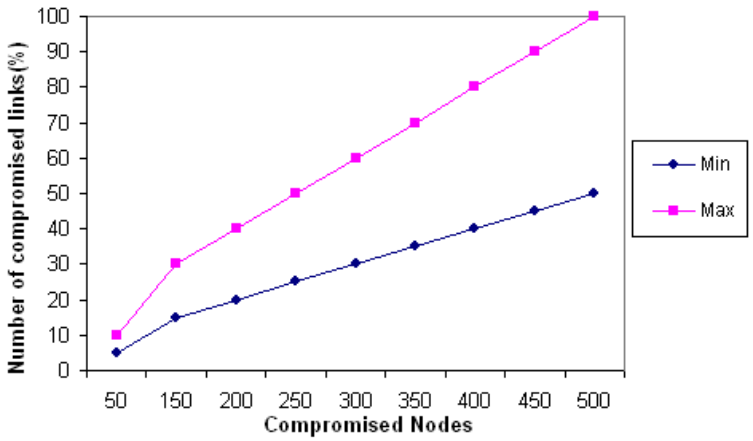


Figure 5.5: Percentage of compromised links: N=1000, Connections=500

Table 5.5: Comparison of LSec with other security solutions

	SPINS [9]	TinySec [10]	LiSP [40]	LSec
Memory requirement with respect to storage of keys	3	Depended on KMS[1]	8	6
Transmission cost during key exchange (bytes)	-	Depended on KMS	12.6 × TNN[2]	74.125×TNC[3]
Transmission cost during data transmission	20%	10%	>20%	8.33%
Authentication support	Yes	Yes	Yes	Yes
Authorization support	No	No	Yes	Yes
Data integrity support	Yes	Yes	Yes	No
Confidentiality support	Yes	Yes	Yes	Yes
Availability support	No	No	Yes	No

[1]KMS: Key Management Scheme

[2]KNN: Total Number of Nodes

[3]KNC: Total Number of Connections

such as MD5, and SHA1 etc or by CRC schemes and availability is normally handled at physical layer. LSec lies between network and application layer, that is why it does not provide explicit data integrity and availability support.

5.5 Summary

In this chapter, Lightweight security protocol (LSec) for WSNs is proposed, which provides authentication and authorization of sensor node. It also provides simple secure key exchange scheme and confidentiality of data. LSec is highly scalable and memory efficient. It uses 6 keys, which takes only 72 bytes of memory storage. It introduces 74.125 bytes of transmission and reception cost per connection. It has the advantage of simple secure defense mechanism against compromised nodes.

Chapter 6

Integrated Solution

6.1 Introduction

A new unified intrusion tolerant trust-based privacy-assured security framework is shown in Figure 6.1. This framework is built on top of the proposed trust [Chap. 3], privacy [Chap. 4] and security [Chap. 5] components that closely interact with one another. In the trust component, the GTMS is responsible for calculating the trust values [Chap. 3, Sec. 3.3.1] of sensor nodes. With the help of generic trust exchange communication protocol (TExP) [Chap. 3, Sec. 3.5.1], the GTMS module will exchange trust values with other nodes. These trust values are further used by the proposed routing schemes (such as IRL [Chap. 4, Sec. 4.3.2] and r-IRL [Chap. 4, Sec. 3.4.1]) that help selecting reliable and secure paths. In case of malicious node(s) detection [Chap. 3, Sec. 4.3.2], the GTMS will send alert message to the security protocol (LSec) [Chap. 5, Sec. 5.2] that will then take further protection steps, such as deletion of a shared secret key, termination of any on going session with the malicious node and alert other member nodes. The LSec protocol is used to generate shared secret keys [Chap. 5, Sec. 5.2.4] for communication between nodes. The secret keys are used by the different modules (like DPriv module of privacy component [Chap. 4, Sec. 4.3.4]) of the framework to exchange information in an encrypted manner.

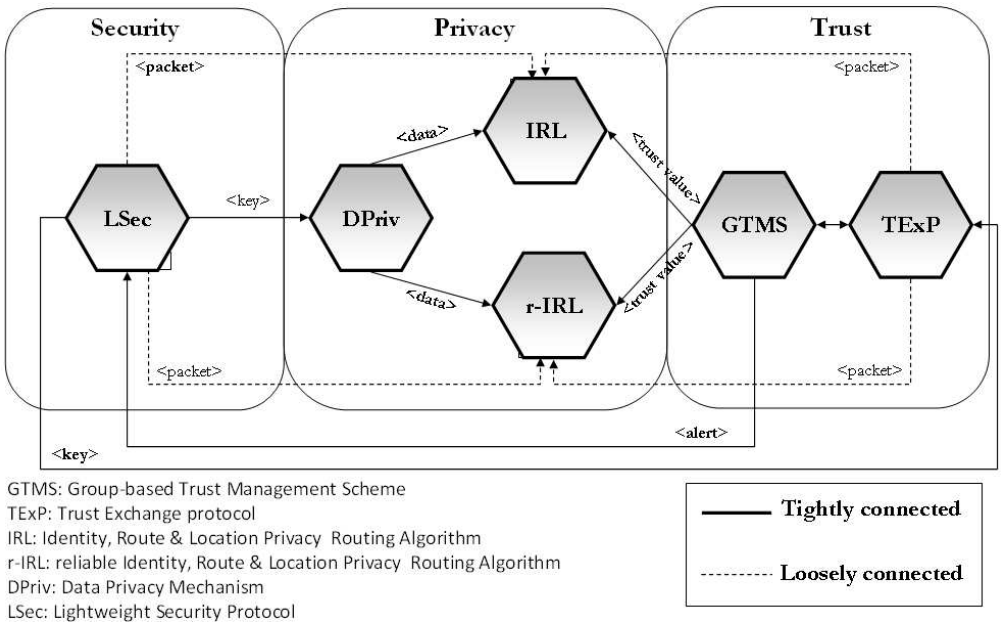
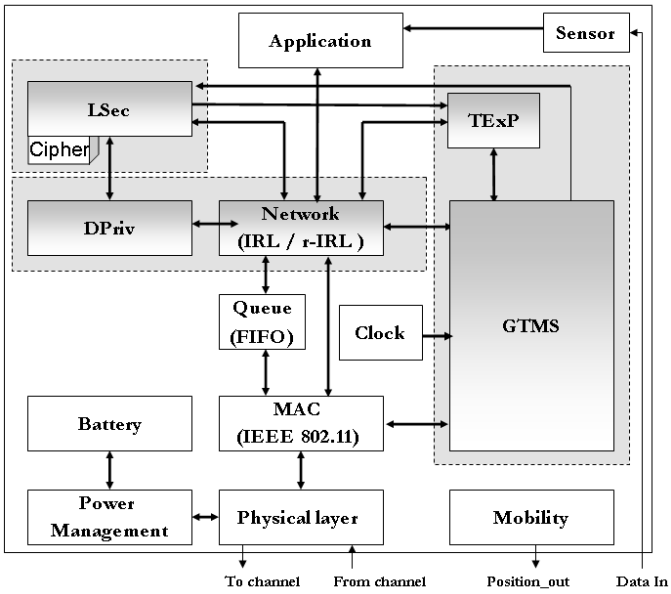


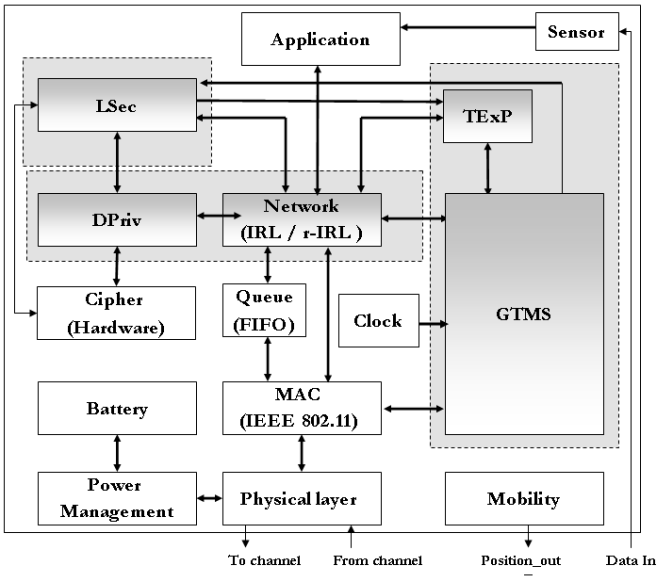
Figure 6.1: Intrusion tolerant trust-based privacy-aware security framework

6.2 Schematic Layout of Complete System

Schematic layout of the complete system based on SENSE [94], is shown in Figure 6.2 which shows the integration of all the components on a single sensor node. This figure shows the completeness of my research is available on a single node. Figure 6.2(a) represents the schematic layout of proposed solution for the sensor nodes where encryption facility is available as a software. However, in order to strengthen the security, many vendors provides the support of hardware level encryption. For example, AES encryption module is available on the Chipcon CC2420 transceiver chip that is used in Crossbow MICAz and MoteIV's TmoteSKY [100]. Proposed solution could also be used in such sensor nodes as shown in Figure 6.2(b).



(a) Encryption module available as a software



(b) Encryption module available as a hardware

Figure 6.2: Schematic layout of the system

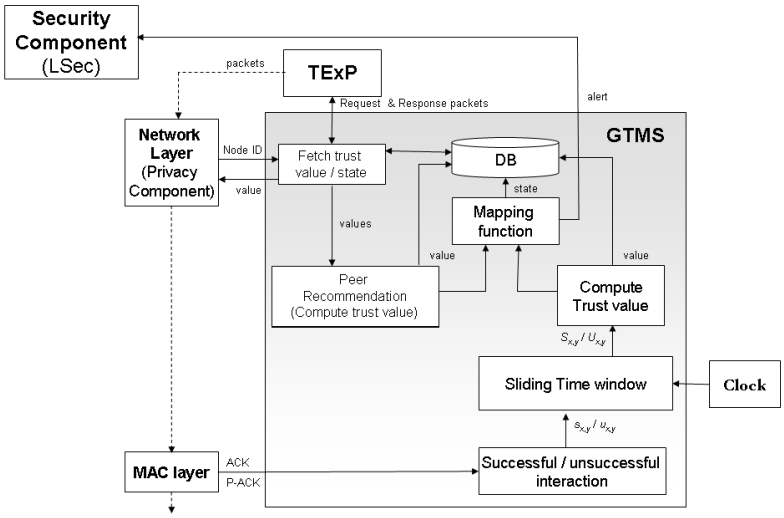


Figure 6.3: Interfaces of trust component

6.3 Interfaces of Trust Component

Proposed GTMS module has four external interfaces as shown in Figure 6.3.

1. *MAC interface:* From the MAC layer, GTMS component receives link layer acknowledgment (ACK) and enhanced passive acknowledgment (P-ACK) for transfer of each packet [Chap. 3, Sec. 3.2.1]. Based on these two information, the GTMS module considers an interaction as a successful or an unsuccessful one [Chap. 3, Sec. 3.2.1]. This information will be further recorded in the the sliding time window [Chap. 3, Sec. 3.3.1]. With this time window information, the time-based past interaction trust value of the other node is calculated [Chap. 3, Sec. 3.3.1, Equation 3.1].

2. *Network interface:* Whenever a routing protocol (e.g IRL [Chap. 4, Sec. 4.3.2] or r-IRL [Chap. 4, Sec. 4.3.3]) needs to select trusted next hop node for the purpose of forwarding packets, it first interacts with the GTMS module. During the

initialization phase [Chap. 4, Sec. 4.3.2], IRL and r-IRL protocols provide node identities to the GTMS module. GTMS module tells IRL and r-IRL protocols that which neighboring nodes are trusted. Based on this information, the routing protocol makes routing decisions [Chap. 4, Sec. 4.3.2].

3. *Exchange interface:* Whenever GTMS module needs recommendations [Chap. 3, Sec. 3.3.1] from other nodes, it sends request packets via generic Trust Exchange Protocol (TExP) [Chap. 3, Sec. 3.5.1]. Based on the recommendation received via TExP protocol, it computes trust value [Chap. 3, Sec. 3.3.1, Equation 3.5].

4. *Alert interface:* Whenever GTMS module detects some malicious node [Chap. 3, Sec. 4.3.2], it will send alert message to the security component.

6.4 Interfaces of Privacy Component

Privacy component is mainly used to generate routing packets. This component ensures the anonymity of a source node's identity and location from an adversary. It also takes care of route anonymity of data packets and data privacy. This privacy component has four external interfaces as shown in Figure 6.4.

1. *Application interface:* Firstly, it is connected to the application layer, from where it receives data packets for forwarding [Chap. 4, Sec. 4.3.2].

2. *Trust interface:* Secondly, it is connected with the trust component [Chap. 3], from where it receives trust values [Chap. 3, Sec. 3.3.1] of the neighboring nodes. These trust values are further used to make reliable routing decisions [Chap. 4, Sec. 4.3.2].

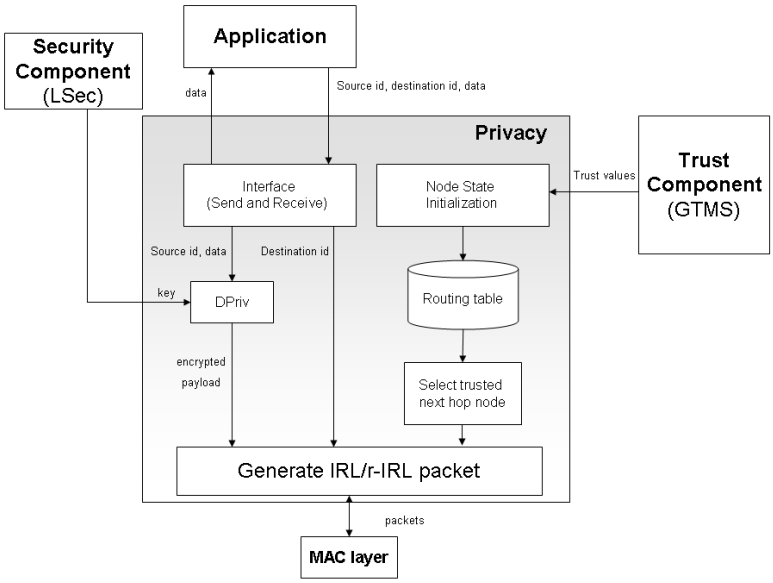


Figure 6.4: Interfaces of privacy component

3. *Security interface:* Thirdly, it is connected to the security component [Chap. 5] from where it receives secret key [Chap. 5, Sec. 5.2.4], that is used to perform encryption of the data packets.

4. *MAC interface:* Lastly, it is connected to the MAC layer, through which it sends and receives packets.

6.5 Interfaces of Security Component

Security component is mainly used to generate secret temporal session key. This privacy component has four external interfaces as shown in Figure 6.5.

1. *Key generation interface:* It is mainly used for the authorization and generation of secret session keys. Through this interface, security component sends and receives

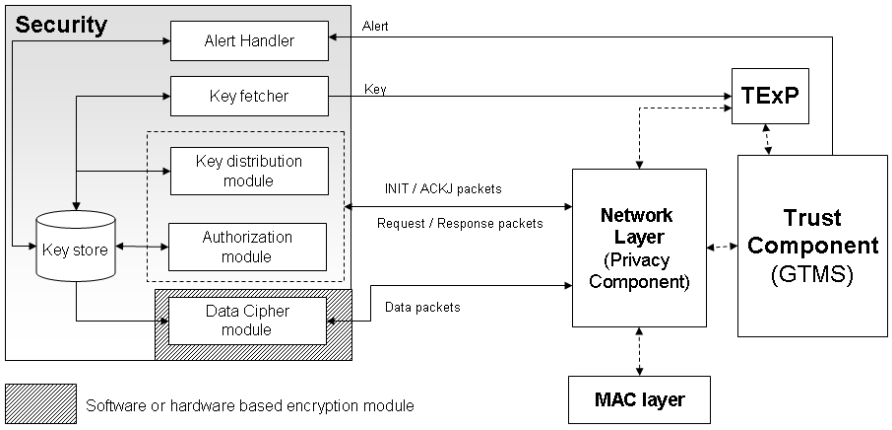


Figure 6.5: Interfaces of security component

INIT, *ACK*, *Request* and *Response* packets via network layer [Chap. 5, Sec. 5.2.4]. In proposed solution, it sends these packets via IRL [Chap. 4, Sec. 4.3.2] or r-IRL [Chap. 4, Sec. 4.3.3] privacy component.

2. *Cipher interface:* It is used to perform encryption and decryption of the data packets. In proposed solution, it is connected with the DPriv module [Chap. 4, Sec. 4.3.4] of the privacy component.

3. *Alert handler interface:* It is mainly used to receive alert messages. On reception of an alert message, security component will terminate earlier key and generate new one if required. In proposed solution, it receive alert messages from the trust component [Chap. 5].

4. *Key provider interface:* it is used to provide secret keys [Chap. 5, Sec. 5.2.4] to other components. In proposed solution, it provides secret key to the TExP module [Chap. 3, Sec. 3.5.1] of trust component.

6.6 Theoretical Analysis and Evaluation

6.6.1 Memory Consumption Analysis

At each sensor node, trust component needs $(n-1)(4+4\Delta t)$ memory space to store trust records. Here n represent the total number of nodes in the group and Δt represents the size of time window. For privacy component, each sensor node needs $6.375n$ memory space and security component requires 72 bytes of memory to store keys. Therefore, memory requirement of complete solution at each sensor node is:

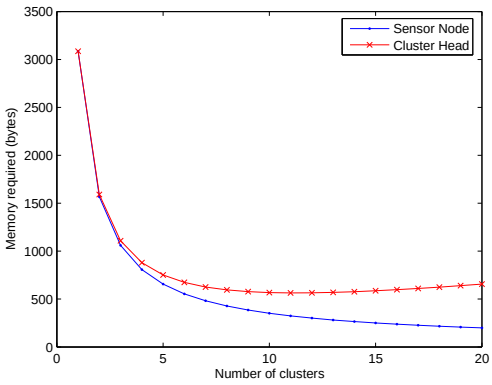
$$M_{SN} = (n-1)(4+4\Delta t) + 6.375n + 72 \tag{6.1}$$

This equation shows that the memory space at each sensor node mainly depended on the size of the cluster and the length of time window. As i have mentioned earlier in discussion that the window length could be made shorter or longer based on the network analysis scenarios. Let us assume that the size of cluster as a one parameter, then based on the Equation 6.1, the window length could be calculate as following.

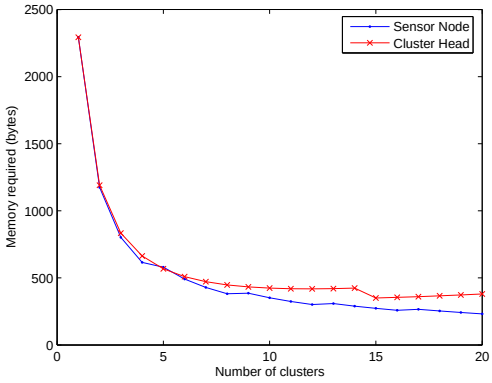
$$\Delta t = \left\lceil \left| \frac{-6.375n - 72}{4\,(n-1)} - 1 \right| \right\rceil \tag{6.2}$$

At each sensor node, trust component needs $(|G| + \sigma - 2)\,(4 + 4\Delta t)$ memory space to store trust records. Here $|G|$ represent the total number of groups / clusters in the networks, and σ represents the average size of the cluster. For privacy and security components consume same amount of memory as at sensor node. Therefore, memory requirement of complete solution at each cluster head is:

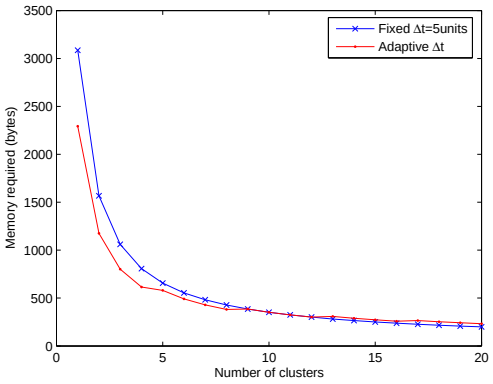
$$M_{CH} = (|G| + \sigma - 2)\,(4 + 4\Delta t) + 6.375\sigma + 72 \tag{6.3}$$



(a) Fixed Δt = 5 units



(b) Adaptive Δt



(c) Fixed vs Adaptive Δt at SN

Figure 6.6: Memory requirement of complete solution: N=100

Similarly, based on the Equation 6.3, the window length could be calculate as following.

$$\Delta t = \left\lceil \left| \frac{-6.375\sigma - 72}{4\left(|G| + \sigma - 2\right)} - 1 \right| \right\rceil \tag{6.4}$$

Figure 6.6 shows the affects of a size of cluster and window length on memory consumption at the sensor node and cluster head. Figures 6.6(a) and 6.6(b) show that as the size of cluster decreases the memory requirement at the sensor node and cluster head also decreases. Also, Figure 6.6(c) shows that the adaptive time window length is more efficient in terms of memory consumption as compared with the fixed time window length.

6.6.2 Communication Overhead Analysis

First I assume a worst case scenario, in which every member node wants to communicate with every other node in the group and every group wants to communicate with the rest of the groups in the network. In order to calculate the trust value, each node perform peer recommendation before start of any communication. Additionally, peer recommendation will be performed in a secure manner. Let us assume that the network consist of $|G|$ groups and the average size of groups is σ.

In the intra-group communication case, when node i wants to interact with the node j, node i will send maximum $\sigma - 2$ peer recommendation requests [Chap. 3, Sec. 3.4.2]. Since peer recommendation are performed in a secure manner, therefore four additional control packets will be forwarded to generate session key [Chap. 5, Sec. 5.3.4]. Therefore, maximum communication overhead for sending peer recommendation request will be $4(\sigma-2)$. In response of peer recommendation requests, node i will maximum receive $\sigma-2$ responses. If node i wants to interact with all the nodes in the group, the maximum communication overhead will be $5(\sigma - 1)(\sigma - 2)$. If all nodes want to communicate

with each other, the maximum intra-group communication overhead ($C_{w-intra}$) of the complete solution is:

$$C_{w-intra} = 5\sigma(\sigma-1)(\sigma-2) \tag{6.5}$$

In the inter-group communication case, when group i wants to interact with group j, it will send one peer recommendation request to the base station, at the maximum. Since cluster head already shared secret key with the base station, therefore security component will not introduce any additional overhead [Chap. 5, Sec. 5.2.2]. So the communication overhead for each request is 2 packets. If group i wants to communicate with all the groups then the maximum communication overhead will be $2|G|-1$ packets. If all the groups want to communicate with each other, the maximum inter-group communication overhead (C_{w_inter}) is:

$$C_{w-inter} = 2|G|(|G|-1) \tag{6.6}$$

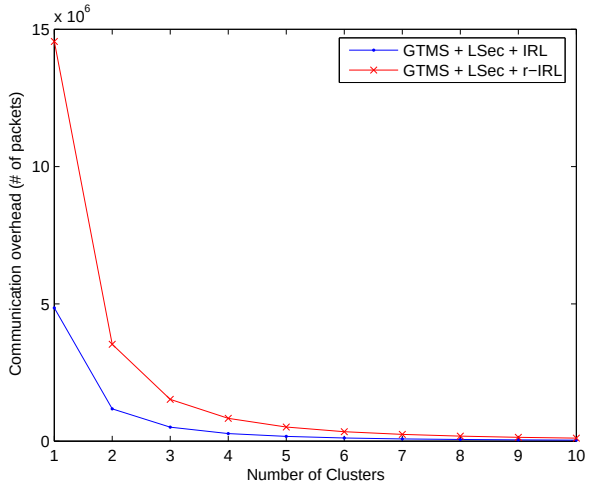
Therefore, in the worst case, the maximum communication overhead C_{worst} introduce by the complete solution in the network is:

$$\begin{aligned} C_{worst} &= |G| \times C_{\mathrm{w-intra}} + C_{\mathrm{w-inter}} \\ C_{worst} &= |G| \times [5\sigma\,(\sigma-1)\,(\sigma-2)] + 2\,|G|\,(|G|-1) \\ C_{worst} &= |G| \times [5\sigma\,(\sigma-1)\,(\sigma-2) + 2\,(|G|-1)] \end{aligned} \tag{6.7}$$

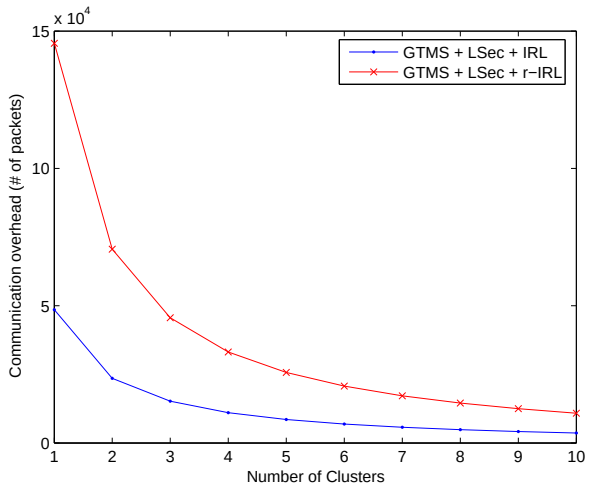
On average, communication overhead C_{avg} introduce by the complete solution in the network is:

$$\begin{aligned} C_{avg} &= |G| \times \tfrac{C_{\mathrm{w-intra}}}{\sigma} + \tfrac{C_{\mathrm{w-inter}}}{|G|} \\ C_{avg} &= |G| \times \left[\tfrac{5\sigma(\sigma-1)(\sigma-2)}{\sigma}\right] + \tfrac{2|G|(|G|-1)}{|G|} \\ C_{avg} &= 5\,|G|\,(\sigma-1)\,(\sigma-2) + 2\,(|G|-1) \end{aligned} \tag{6.8}$$

In the best case, no peer recommendation will be performed by each node in the network. Nodes will make decision based on direct observations. Before start of each session, four control packets are exchanged between communicating nodes.



(a) Worst case



(b) Average case

Figure 6.7: Communication overhead of complete solution: N=100, r=3

Table 6.1: Communication overhead of complete solution

Cases	GTMS + LSec + IRL	GTMS + LSec + rIRL
Worst	$\|G\|\left[5\sigma\left(\sigma-1\right)\left(\sigma-2\right)+2\left(\|G\|-1\right)\right]$	$r\,\|G\|\left[5\sigma\left(\sigma-1\right)\left(\sigma-2\right)+2\left(\|G\|-1\right)\right]$
Average	$5\,\|G\|\left(\sigma-1\right)\left(\sigma-2\right)+2\left(\|G\|-1\right)$	$r\left[5\,\|G\|\left(\sigma-1\right)\left(\sigma-2\right)+2\left(\|G\|-1\right)\right]$

For the routing purpose, if we assume r-IRL routing scheme [Chap. 4, Sec. 4.3.3], then communication overhead will increase with the factor of r. Summary of communication overhead for the two different cases is given in Table 6.1. Figure 6.7 shows the comparison of the two possible combinations of proposed solution in the worst and average case scenarios.

6.7 Summary

This chapter contains the integration details of a proposed unified intrusion tolerant trust-based privacy-assured security framework. It also contains a brief description about the interfaces of each component (trust, privacy and security). This description is helpful in understanding the interactions between proposed components. This chapter also provides theoretical analysis and evaluation of the complete solution from the perspective of memory consumption and communication overhead.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

This thesis aims to achieve more *completeness and reliability in a security* solution for wireless sensor networks by addressing the requirements of high level security, energy, memory and communication overhead efficiency. The primary contribution is schematic development of the unified, resource-efficient framework, called intrusion tolerant trust-based privacy-assured security framework. My contributions are rehashed as follows.

A new lightweight Group-based Trust Management Scheme (GTMS) is proposed for WSNs. The GTMS consists of the three unique features:

1. GTMS evaluates the trust of a group of sensor nodes in contrast to traditional trust management schemes that always focus on trust values of individual nodes. This approach gives us the benefit of requiring less memory to store trust records at each sensor node in the network.

2. GTMS works on two topologies: intra-group topology where distributed trust management approach is used and inter-group topology where centralized trust management approach is adopted. This methodology helps to drastically reduce the cost associated with trust evaluation of distant nodes.

3. GTMS not only provides a mechanism to detect malicious nodes, but also provides some degree of prevention mechanism.

These and other specific features (e.g., independent of any specific routing scheme and platform etc.) collectively make the GTMS a new, lightweight, flexible, and robust solution that can be used in any clustered WSNs.

The problem of achieving network level privacy in wireless sensor networks is addressed and have the following contributions.

- A new Identity, Route and Location (IRL) privacy algorithm is proposed that ensures the anonymity of source node's identity and location from the adversary. It also gives assurance that the packets will reach their destination by passing through only trusted intermediate nodes.

- A new reliable Identity, Route and Location (r-IRL) privacy algorithm is proposed, which is the extension of proposed IRL algorithm. This algorithm has the ability to forward packets from multiple secure paths to increase the packet reach-ability.

- A new Data Privacy (DPriv) mechanism is proposed, which is unique in the sense that it provides data secrecy and packet authentication *in the presence of identity anonymity*.

These solutions collectively provides protection against various privacy disclosure attacks such as eavesdropping and hop-by-hop trace-back attacks. Also, these solutions are light-weight and hence consumes modest memory and energy.

Finally, the problem of developing an energy-efficient security solution is addressed and have the following contributions.

- A new Lightweight Security (LSec) protocol is proposed that provides authentication, and authorization of sensor nodes.

- A simple secure key exchange mechanism is proposed that helps to provide data confidentiality.

This security solution is memory efficient and introduces less communication overhead.

7.2 Future Directions

This thesis provides theoretical analysis and evaluation of the complete solution. However, simulation and real implementation is needed to observe the overall energy consumption of the proposed solution.

The idea of achieving some degree of completeness in the security solution could be extended beyond the wireless sensor networks to the other domains, such as wireless mesh networks etc. However, each domain has its own unique attributes e.g., routing, deployment etc. Therefore, some effort is required to tune various proposed components to make it applicable on the other domains.

In many application scenarios [101, 95], sensor nodes identities should remain hidden for achieving identity anonymity. So, the challenging problem is: how to establish and maintain trust between communicating nodes in an identity anonymous environment? This motivates future work.

In general, privacy is a dynamic need at every level in wireless sensor networks. Applications, nodes and communication packets require different levels of privacy throughout their operation. Thus, privacy cannot be maintained at the same level all the time and an effective privacy scheme should efficiently cater for the dynamic privacy needs at all levels in wireless sensor networks. Hence, privacy should be tackled in a flexible and adaptive manner. Here flexible means that the scheme should support variable levels of privacy and adaptive means that with respect to time and demand, the solution should

automatically adjust the required level of privacy. Therefore, more work is needed to achieve this kind of flexibility and adaptability.

Bibliography

[1] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. Chapter 16: Wireless sensor network security: A survey. In Yang Xiao, editor, *Security in Distributed, Grid, and Pervasive Computing*, pages 367–410. CRC Press, 2006.

[2] Zhaoyu Liu, Anthony W. Joy, and Robert A. Thompson. A dynamic trust model for mobile ad hoc networks. In *Proc. of the 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems*, pages 80–85, Suzhou, China, May 2004.

[3] Sonja Buchegger and Jean-Yves Le Boudec. Self-policing mobile ad hoc networks by reputation systems. *IEEE Comm. Magazine*, 43(7):101–107, July 2005.

[4] A. Czarlinska and D. Kundur. Distributed actuation attacks in wireless sensor networks: implications and countermeasures. In *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS 2006)*, page 10, 2006.

[5] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.

[6] Yong Xi, Loren Schwiebert, and Weisong Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *Proc. of Parallel. and Distributed Processing Symposium (IPDPS 2006)*, Rhodes Island, Greece, Apr 2006.

[7] Habitat monitoring on Great Duck Island (Maine, USA) `http://www.greatduckisland.net/`, 2002.

[8] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proc. of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks*, pages 88–93, DC, USA, Oct 2004.

[9] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, 2002.

[10] Chris Karlof, Naveen Sastry, and David Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proc. of the 2nd Int. Conf. on Embedded networked sensor systems*, pages 162–175, Baltimore, MD, USA, November 2004.

[11] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proc. of the 10th ACM Conf. on Computer and Comm. security*, pages 62–72, NY, USA, 2003.

[12] Saurabh Ganeriwal and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. In *Proc. of ACM Security for Ad-hoc and Sensor Networks*, October 2004.

[13] Zhiying Yao, Daeyoung Kim, and Yoonmee Doh. PLUS: Parameterized and localized trust management scheme for sensor networks security. In *Proc. of the 3rd IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems*, pages 437–446, Vancouver, Canada, October 2006.

[14] Azzedine Boukerche, Xu Li, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Comm.*, 30:2413–2427, September 2007.

[15] Mohammad Momani, Subhash Challa, and Khalid Aboura. Modelling trust in wireless sensor networks from the sensor reliability prospective. In Tarek Sobh et al., editor, *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, pages 317–321. Springer, 2007.

[16] Ke Liu, Nael Abu-Ghazaleh, and Kyoung-Don Kang. Location verification and trust management for resilient geographic routing. *J. of Parallel and Distributed Computing*, 67(2):215–228, 2007.

[17] Haiguang Chen, Huafeng Wu, Xi Zhou, and Chuanshan Gao. Reputation-based trust in wireless sensor networks. In *Proc. of Int. Conf. on Multimedia and Ubiquitous Engineering*, pages 603–607, Korea, April 2007.

[18] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Proc. of the 25th IEEE Int. conf. on Distributed Computing Systems*, pages 599–608, Columbus, Ohio, USA, Jun 2005.

[19] S. Misra and G. Xue. Efficient anonymity schemes for clustered wireless sensor networks. *Int. J. of Sensor Networks*, 1(1/2):50–63, 2006.

[20] A. D. Wood, L. Fang, J. A. Stankovic, and T. He. SIGF: a family of configurable, secure routing protocols for wireless sensor networks. In *Proc. of the 4th ACM*

workshop on Security of ad hoc and sensor networks, pages 35–48, Alexandria, Virginia, USA, 2006.

[21] Yi Ouyang, Zhengyi Le, Guanling Chen, James Ford, and Fillia Makedon. Entrapping adversaries for source protection in sensor networks. In *Proc. of the 2006 Int. Sym. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*, pages 23–34, Buffalo-NY, June 2006.

[22] Wenliang Du, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *Proc. of INFOCOM 2004*, pages 586–597, Hong Kong, China, Mar 2004.

[23] Lydia Ray. Active security mechanisms for wireless sensor networks and energy optimization for passive security routing. In *PhD Dissertation*, Dep. of Computer Science, Louisiana State University, Aug 2005.

[24] J. Kohl and B. Clifford Neuman. The kerberos network authentication service (v5). In *RFC 1510*, Sep 1993.

[25] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transaction on Information Theory*, 22:644–654, 1976.

[26] R. L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public key cryptosystem. *Communication of ACM*, 21(2):120–126, 1978.

[27] Erik-Oliver Bla and Martina Zitterbart. Towards acceptable public-key encryption in sensor networks. In *proc. of 2nd International Workshop on Ubiquitous Computing*, pages 88–93, Miami, USA, 2005.

[28] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems: Facilitating trust in internet interactions. *Comm. of the ACM*, 43(12):45–48, 2000.

[29] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytics. The keynote trust management system. In *RFC2704*, 1999.

[30] George Theodorakopoulos and John S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE J. on Selected Areas in Comm.*, 24(2):318–328, February 2006.

[31] Riaz Ahmed Shaikh, Hassan Jameel, Sungyoung Lee, Saeed Rajput, and Young Jae Song. Trust management problem in distributed wireless sensor networks. In *Proc. of 12th IEEE Int. Conf. on Embedded Real Time Computing Systems and its Applications*, pages 411–414, Sydney, Australia, August 2006.

[32] K. Krishna and A. bin Maarof. A hybrid trust management model for mas based trading society. *The Int. Arab Journal of Information Technology*, 1:60–68, July 2003.

[33] E. Aivaloglou, S. Gritzalis, and C. Skianis. Towards a flexible trust establishment framework for sensor networks. *Telecommunication System*, 35:207–213, 2007.

[34] S. Ganeriwal, L.K. Balzano, and M.B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sensor Networks*, 4(3):1–37, 2008.

[35] Y.L. Sun, Z. Han, and K.J.R. Liu. Defense of Trust Management Vulnerabilities in Distributed Networks. *IEEE Communications Magazine*, 46(2):112–119, 2008.

[36] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *Proc. of IEEE GLOBECOM 2005*, pages 1857–1861, St. Louis, MO, USA, 28 Nov.-2 Dec. 2005.

[37] R. J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Publisher Wiley, 2001.

[38] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones. On providing anonymity in wireless sensor networks. In *Proc. of the 10th Int. conf. on Parallel and Distributed Systems*, pages 411–418, Calafornia, USA, July 2004.

[39] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. SPINS: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, 2002.

[40] Taejoon Park and Kang G. Shin. LiSP: A lightweight security protocol for wireless sensor networks. *Trans. on Embedded Computing Sys.*, 3(3):634–660, 2004.

[41] DeCew and Judith. Privacy. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy (Fall 2006 Edition)*. 2006.

[42] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng. Anonymous secure routing in mobile ad-hoc networks. In *Proc. of the 29th IEEE Int. conf. on Local Computer Networks*, pages 102–108, Tampa, USA, 2004.

[43] D. L. Chaum. The dinning cryptographers problem: unconditional sender and recipient untraceability. *Journal of Cryptography*, 1(1):65–75, 1988.

[44] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. on Information and System Security*, 1(1):66–92, 1998.

[45] M. G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE J. on Selected Areas in Communications*, 6(4):482–494, 1998.

[46] C. Shields and B. N. Levine. A protocol for anonymous communication over the internet. In *Proc. of the 27th ACM conf. on Computer and communications security*, pages 33–42, Athens, Greece, November 2000.

[47] S. Seys and B. Preneel. ARM: Anonymous routing protocol for mobile ad hoc networks. In *Proc. of the 20th Int. conf. on Advanced Information Networking and Applications*, pages 33–37, Vienna Austria, April 2006.

[48] B. Blum, T. He, S. Son, and J. Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. *Tech. Rep. CS-2003-11, Dept. of Comp. Sci. University of Virginia, USA*, 2003.

[49] C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proc. of the 2nd Int. Conf. on Embedded networked sensor systems*, pages 162–175, Baltimore, MD, USA, Nov. 2004.

[50] S. Michell and K. Srinivasan. State based key hop protocol: a lightweight security protocol for wireless networks. In *Proc. of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 112–118, 2004.

[51] S.M.K. Raazi, A.M. Khan, F.I. Khan, S.Y. Lee, and Y-J. Song. MUQAMI: A locally distributed key management scheme for clustered sensor networks. In S. Etalle and S. Marsh, editors, *Int. Federation for Infor. Proc.*, pages 333–348. Boston Springer, 2007.

[52] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston. Security for sensor networks. In *CADIP Research Symposium*, 2002.

[53] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Communications of ACM*, 47(6):53–57, 2004.

[54] T. Moore. A collusion attack on pairwise key predistribution schemes for distributed sensor networks. In *Proc. of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pages 251–255, 2006.

[55] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(4):500–528, 2006.

[56] A. Durresi, V. Paruchuri, M. Durresi, and L. Barolli. Anonymous routing for mobile wireless ad hoc networks. *Int. J. of Distributed Sensor Networks*, 3:105–117, 2007.

[57] Nasipuri Asis and Kai Li. A directionality based location discovery scheme for wireless sensor networks. In *proc. of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02)*, pages 105–111, New York, NY, USA, 2002. ACM.

[58] R. Peng and M.L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *Proc. of the Third Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '06)*, pages 374 – 382, 2006.

[59] P. Michiardi and R. Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. of the 6th IFIP conf. on communications and multimedia security*, pages 107–121, Portoroz, Slovenia, 2002.

[60] S. Buchegger and J. Le Boudec. Performance analysis of the CONFIDANT protocol. In *Proc. of the 13th ACM Symp. on Mobile Ad Hoc Networking and Computing*, pages 226–236, Lausanne, Switzerland, June 2002.

[61] V. S. Bhuse. Lightweight intrusion detection: A second line of defense for unguarded wireless sensor networks. *PhD thesis, Dept. of Comp. Sci., Western Michigan University*, 2007.

[62] Lance J. Hoffman, Kim Lawson-Jenkins, and Jeremy Blum. Trust beyond security: An expanded trust model. *Comm. of the ACM*, 49(7):95–101, July 2006.

[63] Elaine Shi and Adrian Perrig. Designing secure sensor networks. *IEEE Wireless Comm.*, 11(6):38–43, 2004.

[64] H. S. Ng, M. L. Sim, and C. M. Tan. Security issues of wireless sensor networks in healthcare applications. *BT Technology Journal*, 24(2):138–144, April 2006.

[65] Asad Amir Pirzada and Chris McDonald. Establishing trust in pure ad-hoc networks. In *Proc. of 27th Australasian Computer Science Conf.*, pages 47–54, Dunedin, New Zealand, January 2004.

[66] Yan Lindsay Sun, Wei Yu, Zhu Han, and K. J. Ray Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE J. on Selected Areas in Comm.*, 24(2):305–317, February 2006.

[67] Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Comm. Surveys & Tutorials*, 3(4), 2000.

[68] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Comm.*, 1(4):660–670, October 2002.

[69] S Lindsey, CS Raghavendra, and S Raghavendra. PEGASIS- power-efficient gathering in sensor information systems. In *Proc. of IEEE Aerospace Conference, vol 3*, pages 1125–1130, 2002.

[70] Arati Manjeshwar and Dharma P. Agrawal. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proc. of 15th Int. Parallel and Distributed Processing Symp. Workshops*, pages 2009–2015, San Francisco, USA, April 2001.

[71] Ossama Younis and Sonia Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *IEEE Trans. on Mobile Computing*, 3(4):366–379, October 2004.

[72] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Trans. on Dependable and Secure Computing*, 3(1):62–77, 2006.

[73] Mohamed Shehab, Elisa Bertino, and Arif Ghafoor. Efficient hierarchical key generation and key diffusion for sensor networks. In *Proc. of the 2nd Annual IEEE Conf. on Sensor and Ad Hoc Comm. and Networks*, pages 197–213, California, USA, September 2005.

[74] Seema Bandyopadhyay and Edward J. Coyle. Minimizing communication costs in hierarchically-clustered networks of wireless sensors. *Computer Networks*, 44(1):1–16, 2004.

[75] Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for peer-to-peer networks. In *Proc. of the 13th Int. workshop on Network and operating systems support for digital audio and video*, pages 144–152, Monterey, CA, USA, June 2003.

[76] David Ingram. An evidence based architecture for efficient, attack-resistant computational trust dissemination in peer-to-peer networks. In *Proc. of 3rd Int. Conf. on Trust Management*, volume 3477 of *LNCS*, pages 273–288, Paris, May 2005. Springer-Verlag.

[77] L. Xiong and L. Liu. Peer trust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[78] Sonja Buchegger and Jean-Yves Le Boudec. A robust reputation system for peer-to-peer and mobile ad-hoc networks. In *Proc. of P2PEcon*, Harvard University, Cambridge MA,USA, June 2004.

[79] Sumit Gupta. Automatic Detection of DOS Routing Attacks in Wireless Sensor Networks. *MS thesis, Dept. of Comp. Sci., University of Houston*, 2006.

[80] X. Du, M. Guizani, Y. Xiao, and H.H. Chen. Two tier secure routing protocol for heterogeneous sensor networks. *IEEE Transactions on Wireless Communications*, 6(9):3395–3401, 2007.

[81] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (WSNA'03)*, pages 113–127, Anchorage, Alaska, USA, May 2003.

[82] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and Trust-based Systems for Ad Hoc and Sensor Networks. *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*, 2006.

[83] Hassan Jameel, Le Xuan Hung, Umar Kalim, Ali Sajjad, Sungyoung Lee, and Young-Koo Lee. A trust model for ubiquitous systems based on vectors of trust values. In *Proc. of 3rd IEEE Int. workshop on Security in Storage*, pages 674–679, California, USA, December 2005.

[84] Anna Hac. *Wireless Sensor network Designs*. John Wiley & Sons, Ltd., 2003.

[85] R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. of Wireless Communications and Networking Conference*, pages 350 – 355, California, USA, 2002.

[86] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communication Magazine*, 43(3):8–13, 2005.

[87] Environmental Sensing Array at James Reserve Using Crossbow's MICA2 motes `http://www.xbow.com/General_info/Info_pdf_files/Xbow_Newsletter.pdf`, 2004.

[88] Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20(3):20–25, 2006.

[89] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *IEEE Symp. on Security and Privacy*, pages 197–213, California, USA, May 2003.

[90] Henk Tijms. *Understanding Probability: Chance Rules in Everyday Life*. Cambridge University Press, Cambridge, 2004.

[91] Crossbow Inc., Wireless sensor networks, MICA2 Series url-http://www.xbow.com, 2008.

[92] Saurabh Ganeriwal, Laura K. Balzano, and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transaction on Sensor Networks*, 4(3):1–37, 2008.

[93] Huseyin Ozgur Tan and Ibrahim Korpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *ACM SIGMOD Record*, 32(4):66–71, December 2003.

[94] B. K. Szymanski, SENSE: Sensor network simulator and emulator. `http://www.ita.cs.rpi.edu/sense/index.html`, 2008.

[95] Satyajayant Misra and Guoliang Xue. Efficient anonymity schemes for clustered wireless sensor networks. *Int. J. of Sensor Networks*, 1(1/2):50–63, 2006.

[96] Haodong Wang, Bo Sheng, and Qun Li. Privacy-aware routing in sensor networks. *Computer Networks*, 53(9):1512–1529, 2009.

[97] Sergio Armenia, Giacomo Morabito, and Sergio Palazzo. Analysis of location privacy/energy efficiency tradeoffs in wireless sensor networks. In *IFIP-NETWORKING 2007, LNCS 4479*, pages 215–226, Atlanta, Georgia, USA, May 2007.

[98] C. P. Pfleeger and S. L. Pfleeger. *Security in Computing*. Prentice Hall, 4 edition, 2006.

[99] Riaz A. Shaikh, Sungyoung Lee, M. A. U. Khan, and Young Jae Song. LSec: Lightweight security protocol for distributed wireless sensor network. In *11th IFIP Int. Conf. on Personal Wireless Comm., LNCS 4217*, volume 4217 of *LNCS*, pages 367–377, Albacete, Spain, September 2006. Springer-Verlag.

[100] M. Healy, T. Newe, and E. Lewis. Analysis of Hardware Encryption Versus Software Encryption on Wireless Sensor Network Motes. *Smart Sensors and Sensing Technology, Lecture Notes in Electrical Engineering, vol. 20*, pages 3–14, 2008.

[101] S. Olariu, Q. Xu, M. Eltoweissy, A. Wadaa, and A. Y. Zomaya. Protecting the communication structure in sensor networks. *Int. J. of Distributed Sensor Networks*, 1:187–203, 2005.

Publications

Patents

1. Sungyoung Lee, Young-Koo Lee, **Riaz Ahmed Shaikh**, "Method of trust management in wireless sensor networks", US-Patent Application number: 12/178,722, July 24, 2008

2. Sungyoung Lee, Young-Koo Lee, **Riaz Ahmed Shaikh**, "Method for identity, route, and location anonymity in wireless sensor networks", Korean Patent Application number: 10-2007-0133041, December 18, 2007

Book Chapters

1. **Riaz Ahmed Shaikh**, Brian J. d'Auriol, Heejo Lee and Sungyoung Lee, "Privacy and Trust Management Schemes of Wireless Sensor Networks: A Survey", *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice*, Hai Jin and Wenbin Jiang (Eds.), IGI Global (Publisher).

Journals

1. **Riaz Ahmed Shaikh**, Hassan Jameel, Brian J. d'Auriol, Sungyoung Lee, Young-Jae Song and Heejo Lee, "Group-based Trust Management Scheme for Clustered

Wireless Sensor Networks", *IEEE Transaction on Parallel and distributed Systems*, IEEE Computer Society, (in press).

2. **Riaz Ahmed Shaikh**, Young-Koo Lee, and Sungyoung Lee, "An Extended Energy Consumption Analysis of Reputation-based Trust Management Schemes of Wireless Sensor Networks", *Journal of Networks*, Academy Publishers (in press).

3. **Riaz Ahmed Shaikh**, Hassan Jameel, Brian J. d'Auriol, Sungyoung Lee, Young-Jae Song and Heejo Lee, "Network Level Privacy for Wireless Sensor Networks (Extended Version)", *Submitted for Publication in Journal*.

4. **Riaz Ahmed Shaikh**, Sungyoung Lee, M. A. U. Khan and Young Jae Song, "LSec: Lightweight Security Protocol for Distributed Wireless Sensor Network", *Lecture Notes in Computer Science*, vol. 4217, P. Cuenca and L. Orozco-Barbosa (Eds.), Springer, Sep 2006, pp. 367-377.

Conferences

1. M. Shoaib Siddiqui, **Riaz A. Shaikh**, C. S. Hong, "Trust-based Anonymity Framework for Wireless Mesh Networks", in proc. *of the 11th International Conference on Advanced Communication Technology*, Korea, Feb 2009, pp. 1638-1642.

2. M. Shoaib Siddiqui, **Riaz A. Shaikh**, C. S. Hong. "QoS Control in Service Delivery in IMS", in proc. *of the 11th International Conference on Advanced Communication Technology*, Korea, Feb 2009, pp. 157-160.

3. **Riaz A. Shaikh**, Young-Koo Lee, Sungyoung Lee, "Energy Consumption Analysis of Reputation-based Trust Management Schemes of Wireless Sensor Networks", in proc. *of the 3rd International Conference on Ubiquitous Information Management and Communication*, Suwon, Korea, Jan 2009, pp. 602-606.

4. L. Hung, **Riaz A. Shaikh**, Hassan J., S.M.K. Raazi, Y. Weiwei, N. Canh, P. Truc, S. Lee, H. Lee, Y. Son, and M. Fernandes, "Activity Oriented Access Control for Ubiquitous Environments", in proc. *of the 6th Annual IEEE Consumer Communications & Networking Conference (CCNC 2009)*, Las Vegas Jan, 2009, pp. 1-5.

5. L. Hung, Hassan J., **Riaz A. Shaikh**, S.M.K. Raazi, Y. Weiwei, N. Canh, P. Truc, S. Lee, H. Lee, Y. Son, and M. Fernandes, "Activity-based Security Scheme for Ubiquitous Environments", in proc. *of 27th IEEE International Performance Computing and Communications Conference* , USA, Dec 2008, pp. 475-481.

6. **Riaz A. Shaikh**, Hassan Jameel, Brian J. d'Auriol, Sungyoung Lee, Young-Jae Song and Heejo Lee, "Trusting Anomaly and Intrusion Claims for Cooperative Distributed Intrusion Detection Schemes of Wireless Sensor Networks", in proc. *of the 2008 International Symposium on Trust Computing (TrustCom 2008)*, Hunan, China, Nov 2008, pp. 2038-2043.

7. Hassan Jameel, **Riaz A. Shaikh**, Le Xuan Hung, Yuan WeiWei, Syed Muhammad Khaliq-ur-rehman, Raazi, Ngo Trong Canh, Sungyoung Lee, Heejo Lee, Yuseung Son and Miguel Fernandes, "Image-Feature based Human Identification Protocols on Limited Display Devices", In proc. of *the 9th International Workshop on Information Security Applications (WISA 2008)*, Jeju, korea, Sep 2008, pp. 211-224.

8. **Riaz A. Shaikh**, Hassan Jameel, Brian J. d'Auriol, Sungyoung Lee, Young-Jae Song and Heejo Lee, "Network Level Privacy for Wireless Sensor Networks", *in proc. of 4th International Conference on Information Assurance and Security (IAS 2008)*, Naples, Italy, Sep 2008, pp. 261-266.

9. Pho Duc Giang, Le Xuan Hung, **Riaz A. Shaikh**, Sungyoung Lee, Young-Koo Lee and Heejo Lee, "A Trust-Based Approach to Control Privacy Exposure in

Ubiquitous Computing Environments", in proc. *of IEEE International Conference on Pervasive Services (ICPS 2007)*, Istanbul, Turkey, Jul 2007, pp. 149-152.

10. Brian J. d'Auriol, Jie Yang, Xiaoling Wu, Hui Xu, Yu Niu, Jin Wang, **Riaz A. Shaikh**, Min Meng, Sungyoung Lee, and Young-Koo Lee, "A Research Framework Model to Guide Both Broad and Focused Research into Ubiquitous Sensor Networks", in proc. *of the 2007 International Conference on Wireless Networks (ICWN'07)*, Las Vegas, Nevada, USA, Jun 2007, pp. 468-473.

11. Hassan Jameel, **Riaz A. Shaikh**, Sungyoung Lee and Heejo Lee, "Human Identification through Image Evaluation using Secret Predicates", *RSA Conference 2007, Cryptographers' Track*, LNCS vol. 4377, USA, Feb 2007, pp. 67-84.

12. **Riaz A. Shaikh**, Hassan Jameel, Sungyoung Lee, Young Jae Song, and Saeed Rajput, "Trust Management Problem in Distributed Wireless Sensor Networks", in proc. *of the 12th IEEE International Conference on Embedded Real Time Computing Systems and its Applications*, Sydney, Australia, Aug 2006, pp. 411-415.

13. **Riaz A. Shaikh**, Sungyoung Lee, Young Jae Song, and Yonil Zhung, "Securing Distributed Wireless Sensor Networks: Issues and Guidelines", in proc. *of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2006) - vol. 2 - Workshops*, Taiwan, Jun 2006, pp. 226-231.

14. Xiaoling Wu, Hoon Heo, **Riaz A. Shaikh**, Jinsung Cho, Oksam Chae, and Sungyoung Lee, "Individual Contour Extraction for Robust Wide Area Target Tracking in Visual Sensor Networks", in proc. *of the 9th IEEE International Symposium on Object and component-oriented Real-time distributed Computing (ISORC 06)*, Gyeongju, Korea, Apr 2006, pp. 179-185.

List of Abbreviations

ATRM	Agent based Trust and Reputation Management
CAS	Cryptographic Anonymity Scheme
CEM	Cyclic Entrapment Method
GROW	Greedy Random Walk
GTMS	Group-based Trust Management Scheme
LEAP	Localized Encryption and Authentication Protocol
LiSP	Lightweight Security Protocol
MAC	Message Authentication Code
PFR	Phantom-Flood Routing
PSR	Phantom Single-path Routing
PLUS	Parameterized and Localized trUst management Scheme
RFSN	Reputation-based Framework for Sensor Networks
SAS	Simple Anonymity Scheme
SBKH	State Based Key Hop
SENSE	Sensor Network Simulator and Emulator
SIGF	Secure Implicit Geographic Forwarding
T-RGR	Trust-Resilient Geographic Routing
WSN	Wireless Sensor Networks