FALSE DATA DETECTION IN WIRELESS SENSOR NETWORKS

Thesis

Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

ALOK KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575 025

NOVEMBER, 2019

DECLARATION

by the Ph.D. Research Scholar

I hereby declare that the Research Thesis entitled **FALSE DATA DETECTION IN WIRELESS SENSOR NETWORKS** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfilment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Computer Science and Engineering Department** is a bonafide report of the research work carried out by me. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Alok Kumar

Register No. 155087-CS15F01

Department of Computer Science and Engineering

Place: NITK, Surathkal.

Date: 4^{th} November, 2019

CERTIFICATE

This is to certify that the Research Thesis entitled **FALSE DATA DETECTION IN WIRELESS SENSOR NETWORKS** submitted by **ALOK KUMAR**, (Register Number: 155087-CS15F01) as the record of the research work carried out by him, is accepted as the Research Thesis submission in partial fulfilment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. Alwyn Roshan Pais
Research Guide

(Signature with Date and Seal)

Chairman - DRPC
(Signature with Date and Seal)

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to all the people who supported and inspired me during the tenure of my Ph.D. study. First and foremost I would like to thank my supervisor and Head of the Department, Dr. Alwyn Roshan Pais for giving me the guidance, encouragement and counsel throughout my research. Without his invaluable advice and assistance, it would not have been possible for me to complete this thesis. Beyond the research work, I have learned a lot from him that, I am sure, will be useful in the later stages of my career and life.

I would like to express my sincere thanks and gratitude to the members of my research progress assessment committee, Dr. P. Santhi Thilagam and Dr. Pushparaj Shetty D. for their positive criticism, insightful feedback and constructive suggestions throughout my research work. I also extend my sincere thanks to the entire faculty of the Department of Computer Science and Engineering for their support. Nevertheless, I am also grateful to the technical and administrative staff of the department for their timely help and co-operation. Further, I would like to express my sincere thanks to NITK for providing the necessary infrastructure and facilities required for the successful completion of this research work.

Special thanks to Apurva Kittur, who constantly supported me during the tenure of my Ph.D. I warmly thank my lab-mates Srinivas, Nikhil, Ajnas, Sivakumar Sir, Somesha Sir and Zubair for their encouragement and for the good times we spent in the department. Further, I want to thank my fellow batch-mates, seniors and juniors for their support.

I thank my fellow roommates Rajesh, Biswajit and Pramod who made my stay at NITK a memorable one.

This thesis would not have been possible without the exceptional support of my

family. I am indebted to my parents and family members for their unconditional love, inspiration and support over the years.

Finally, I would like to thank all of them whose names are not mentioned here but have helped me in some way to accomplish the work.

Alok Kumar

ABSTRACT

En-Route filtering is a method to detect and filter false reports in Wireless Sensor Networks (WSNs). The radio capabilities of sensor nodes are very limited. Thus the reports have to be forwarded through intermediate nodes to be collected at a central facility. In En-Route filtering, the intermediate nodes do an authenticity check of all the reports before they are forwarded to next hop. In recent times, many En-Route filtering schemes have been proposed. Each of these schemes use different cryptographic methods to filter false reports from the WSNs. However, the majority of these techniques can handle only limited compromised nodes or either needs node localization or statically configured routes for sending reports. Furthermore, the majority of En-Route filtering techniques are vulnerable to various Denial of Service (DoS) attacks.

Though, the contemporary techniques proposed in the field of En-Route filtering have evolved with the time, but still, the majority of them are prone to selective forwarding and report disruption attacks. This research work focuses on handling the problems and limitations of En-Route filtering to device new techniques which are resilient to various DoS attacks. We in our work will try to reduce communication overhead and reduce the effect of various DoS attacks (Report Disruption Attack and Selective Forwarding Attack) in WSNs.

The basic idea of En-Route filtering is checking of reports by intermediate nodes. This helps to decrease the processing and checking overhead of sink and thus false reports can be removed from the network within some nodes from the origin, saving energy and bandwidth. In this approach, each report is attached to Message Authentication Codes (MACs) or signatures. Whenever these reports are being forwarded over the network, intermediate nodes can authenticate these MACs or signatures and if any fault is found, reports are dropped. For creation and verification of MACs in the network, sensor nodes exchange secret keys with other sensor nodes in the network. Thus, this research work mainly focuses on proposing new key pre-distribution schemes and

then to extend the proposed key pre-distribution schemes to propose new En-Route filtering schemes.

In this thesis, secure key pre-distribution mechanisms are studied. The first study is based on improvements in combinatorial design based key pre-distribution mechanism. We developed three combinatorial design based key pre-distribution schemes which improved the resiliency of the network against compromised sensor nodes without alarmingly increasing the key storage overhead in the network.

Second study is devoted to propose a new hybrid key pre-distribution scheme which uses both pair-wise keys and combinatorial design based keys. This helped to ensure high resiliency against compromised sensor nodes in the network while maintaining very low key storage overhead when compared to existing schemes.

The last study focused on extending the proposed key pre-distribution schemes to propose novel En-Route filtering schemes. Use of combinatorial design based keys provided a deterministic mechanism for verification of forwarded reports. Thus, the filtering efficiency of the proposed schemes is excellent. For the proposed schemes, a novel report endorsement and verification mechanism is also proposed for robust data authentication and availability in the network. This helped to provide better tolerance against Report Disruption Attack and Selective Forwarding Attack in WSNs.

With thorough analysis and simulation results, we have claimed that the network performances of our key pre-distribution and En-Route filtering schemes are much better as compared to those for the existing schemes.

Keywords: Wireless Sensor Networks (WSNs), False Data Detection, En-Route filtering, Compromised Nodes, Combinatorial design, Pair-Wise keys, Key Pre-distribution, Secure Communication.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Expansion
WSNs	Wireless Sensor Networks
DoS	Denial of Service
MACs	Message Authentication Codes
SCBKE	Symmetric Cryptography Based Key Exchange
ASCBKE	Asymmetric Cryptography Based Key Exchange
CD-KPD	Combinatorial Design based Key Pre-Distribution
CD-RKPD	Combinatorial Design based Reduced Key Pre-Distribution
CD-PKPD	Combinatorial Design based Partial Key Pre-Distribution
CD-HKPD	Combinatorial Design based Hybrid Key Pre-Distribution
CD-EFS	Combinatorial Design based En-Route Filtering Scheme
CD-PEFS	Combinatorial Design based Partial En-Route Filtering Scheme
ECC	Elliptic Curve Cryptography
CoS	Center-of-Stimulus
CH	Cluster Head
LSSS	Linear Secret Sharing Scheme
CAA	Context-Aware Architecture
DCNs	Data Collection Nodes
MAP	Message Authentication Polynomials
EAB	En-Route Authentication Bitmap
LBKs	Location Based Keys
IPKs	Immediate Pairwise Keys
MPKs	Multi-Hop Pairwise Keys
MOLS	Mutually Orthogonal Latin Squares

Abbreviations	**Expansion**
BIBD	Balanced Incomplete Block Design
GPS	Global Positioning System

CHAPTER 1

Introduction

The fast developments in micro-electro mechanical systems and integrated electronics devices have given birth to low-cost Wireless Sensor Networks (WSNs) (Akkaya and Younis 2005; Akyildiz et al. 2002). These networks consist of numerous small nodes which have sensing, computing and communication capabilities (Chi and Cho 2006). These networks have been well adopted as a ubiquitous approach by emerging applications like military tracking, environmental monitoring, surveillance, etc. (He et al. 2010; Szewczyk et al. 2004). The main function of these networks is to detect events of interest and forward the data to sink using multi-hop wireless paths.

Majority of WSNs are placed in hostile and unattended environments, thus security mechanisms are necessary to ensure proper working of these networks. Present WSNs are expected to contain thousands of sensor nodes, thus it is impractical to protect and monitor each sensor node individually. Moreover, it is costly and unrealistic to provide tamper-resistant hardware to each node. Thus, sensor nodes can be easily compromised and these compromised sensor nodes can cause severe security threats (Chan and Perrig 2003).

Attacker can compromise multiple sensor nodes using node replication attacks (Xing and Cheng 2010), code injection attacks (Younan et al. 2011), etc.. These compromised nodes can be used to get the node's cryptographic keys (Przydatek et al. 2003; Zhu et al. 2004) which can be used to control them and can also be used to intercept the data trans-

mitted from these nodes. An attacker can use compromised nodes to inject bogus data traffic into the network, where compromised nodes pretend to have detected an event of interest (Yang et al. 2011). This can cause sink to estimate and take wrong system states (Liu et al. 2011; Cárdenas et al. 2011). Such attacks can damage the network's function and can lead to failure of mission-critical feedbacks. Such attacks waste scarce network resources like energy and bandwidth and also cause network congestion. These attacks can hamper basic security requirements of the network (Shi and Perrig 2004; Karlof and Wagner 2003). The solution to the above problems is the collaborative endorsement of reports and En-Route verification/filtering of reports as they are forwarded towards the sink.

1.1 Wireless Sensor Networks

This section gives a brief introduction of WSNs, where these are deployed, data security requirements in WSNs and various attacks possible on WSNs. Each of these is explained below:

1.1.1 WSN Deployment Areas

There are many scenarios where WSNs are deployed and where data is collected from sensors and analyzed to take appropriate actions, for example:

- In a military application (Hussain et al. 2009), where soldiers have to cross the battlefield, sensor nodes deployed in that area can detect and tell the presence of the enemy. Soldiers can use this information to decide their position strategically on the battlefield.
- In traffic application (Wenjie et al. 2005), where sensors detect traffic jam or an accident on the road, sends this information in all the directions to alert all the traffic coming to that spot.
- In forest fire alarm application (Son et al. 2006), sensors sense and inform authorities about the fire and exact location of fire so that required measures can be

taken.

- In health care applications (Alemdar and Ersoy 2010), where sensors are deployed to observe and monitor patients for health care reasons. Vital signs and conditions of patients along with their GPS location can be useful in case of emergency.
- In manufacturing process control (Akhondi et al. 2010), where sensors are deployed to monitor the condition or for observing the manufacturing process on assembly lines.
- In homeland Security (Potyrailo et al. 2012), where sensors are deployed in public buildings, airports, bridges, subways etc. to recognize and give feedback for any intruder.

The list of applications which uses WSN as a backbone is never ending, but all these scenarios require reports to be sent quickly so that appropriate measures can be taken as soon as possible. WSNs are prone to different attacks as they have limited tamper resistance. Hence there is a need to define proper data security requirements for the WSNs.

1.1.2 WSN Data Security Requirements

The basic data security requirements in WSNs are mainly the same as traditional networks, which are data confidentiality, authenticity and availability (Shi and Perrig 2004; Vogt 2004). Data should be genuine, should be accessible to only authorized entities and should be available at all the time are the basic security requirements for WSNs and traditional networks.

- **Data Confidentiality**- In WSNs, sensors are generally required to send sensed data to sink for reporting an event. As the communication range of the sensor nodes is limited, the reports are to be forwarded through the network by intermediate nodes. The data confidentiality in WSNs states that confidentiality of

reports should be maintained until the sending node is compromised even in the presence of other compromised nodes in the network.

- **Data Authenticity**- Data authenticity in WSNs requires only authenticated nodes to be allowed to create and send reports to the sink. Reports generated in WSNs are generally sensitive and critical. Thus, it is important to assure data authenticity in the network.

- **Data Availability**- Data availability in WSNs states that data in the network should be available at all the time. The sink should be able to query the sensors and receive data from sensors at all the time. Moreover, no genuine reports should be dropped when reports are being forwarded by intermediate nodes. The size of WSN can be very big and protecting each sensor node is not possible thus we assume that some compromised nodes are part of the network. Hence, it is very important to prevent or reduce the interference of compromised nodes to ensure the data availability in the network. The deployed security infrastructure should be robust to tackle compromised nodes. Security designs should be able to prevent, detect and recover from various DoS attacks which can hamper data availability in WSNs.

There are many attacks which can be performed over WSNs to hamper any of the above security requirements.

1.1.3 Attacks on WSNs and their characteristics

Different types of attacks are possible over WSNs, some of them are discussed below:

- **Report Fabrication Attack**- In this attack (Yang et al. 2005) adversary compromises sensor nodes and uses them to inject false reports in the network. Such false reports can deceive the sink to take wrong decisions and cause false alarms. This also leads to increased network congestion and resource wastage.

- **Report Disruption Attack**- In this attack (Yang et al. 2005) adversary inten-

tionally submits corrupted Message Authentication Codes (MACs), signatures or wrong data when the reports are being generated. This can disrupt report generation and data filtering by intermediate nodes.

- **Selective Forwarding Attack**- In normal Denial of Service (DoS) attack compromised nodes refuse to forward messages and simply drops them. But this approach can cause neighbors to catch this behavior. In a more effective attack, an adversary can selectively forward packets, where some packets can be forwarded and some can be dropped. This type of attack is called selective forwarding attack (Yu and Xiao 2006).

- **Spoofing Attacks**- Majority of report delivery and routing techniques rely on acknowledgments. Due to the broadcast nature of acknowledgments, the adversary can spoof acknowledgments for overheard packets and broadcast these acknowledgments to neighbors. The major goal of spoofing attacks (Karlof and Wagner 2003) includes misleading genuine nodes to change the path to reach the sink, etc..

- **Wormhole and Sinkhole Attacks**- In this attack (Hu et al. 2003) adversary tunnel the messages received at one end to other end using low latency link. An adversary could make nodes which are multiple hops away from the base station to believe that base station is only one or two hops away via wormhole. This will create a sinkhole on the other side from where all the traffic can be forwarded to the base station, thus getting all the traffic from surrounding nodes.

- **Sybil Attack**- In this attack (Karlof and Wagner 2003) adversary creates multiple fake identities in the network. Using this attack, an attacker can be at more than one place at a time by presenting many identities of itself in the network. This can significantly reduce the effectiveness of fault tolerance of schemes and can also disturb geographic routing protocols.

- **Eavesdropping and Altering Attacks**- Adversary in this attack can passively be present in the network and listen to passing traffic. This is not a big concern if

there is a robust security protocol deployed, but this monitoring can lead to any of the other attacks discussed above.

- **Hello Flood**- Many routing protocols need to broadcast HELLO packets to show their presence to neighbors. In this attack (Hamid et al. 2006) adversary with enough transmission power can convince nodes which are far from it to believe that adversary is its neighbor and can exchange secure information with them.

- **Sensors Relocation attacks**- In this attack, adversary can physically relocate nodes from their original location. So when an event happens in the range of new location, these nodes will generate wrong reports using their original location and keying material.

Attacks discussed above can hamper the normal working of the network and because of receiving spoofed reports from compromised nodes sink can take wrong decisions. To stop the adversary from performing any of the above attacks and to detect any attack done, proper authentication of reports is needed. For robust authentication of data, En-Route verification/filtering of reports can be adopted in WSNs.

1.2 En-Route filtering

As the name suggests En-Route filtering is checking and filtering of reports En-Route from origin to sink. The basic idea of En-Route filtering is checking of reports by intermediate nodes. This helps to decrease the processing and checking overhead of sink and thus false reports can be removed from the network within some nodes from the origin, saving energy and bandwidth. In this approach, each report is attached to MACs or signatures. Whenever these reports are being forwarded over the network, intermediate nodes can authenticate these MACs or signatures and if any fault is found, reports are dropped.

The existing En-Route filtering techniques have mainly 3 phases (Figure 1.1) - Key exchange phase, En-Route filtering phase and Sink verification phase. In the Key exchange phase, nodes exchange keys with intermediate nodes on the forwarding path to the sink. In En-Route filtering phase, intermediate nodes filter and forward the reports

toward the sink. In sink verification phase, sink act as a final goalkeeper for the whole network where it collects and verifies all the reports. Many techniques (Ye et al. 2005a; Sun et al. 2009; Sun and Wu 2011; Yang et al. 2005; Yu and Li 2009; Yang and Lu 2004; Zhang et al. 2006; etc..) have been proposed for key exchange phase which can be grouped into two major categories- Symmetric Cryptography Based Key Exchange (SCBKE) and Asymmetric Cryptography Based Key Exchange (ASCBKE). Majority of SCBKE techniques uses MACs derived from symmetric keys shared between multiple nodes. Each legitimate report should have certain minimum valid MACs. On the other hand, ASCBKE techniques use signatures which can be verified by intermediate nodes and sink. These techniques do not require any pre-shared keys and these mainly use Elliptic Curve Cryptography (ECC) (Hankerson et al. 2006) and Shamir's threshold cryptography (Shamir 1979) to generate signatures.

But because of the use of En-Route filtering, adversaries can also launch DoS attacks

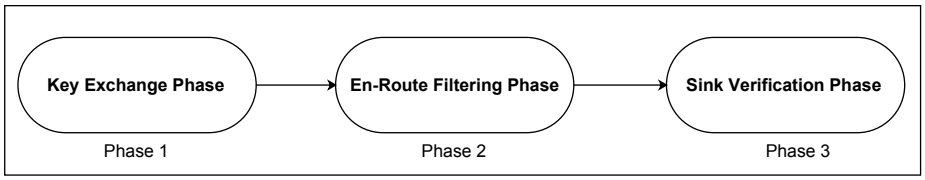


Figure 1.1: Three phases of En-Route filtering

(Zargar et al. 2013) against legitimate data using selective forwarding attack (Yu and Xiao 2006) and report disruption attack (Bauer et al. 2007). In selective forwarding attack, compromised nodes selectively drop legitimate reports. In report disruption attack, compromised nodes can intentionally contaminate the information which is needed to create a report, these reports will be filtered out by the sink or intermediate nodes using En-Route filtering.

1.2.1 Example

In this subsection, we will discuss a typical WSN and how En-Route filtering provides data authentication and false data detection in WSNs. Figure 1.2 shows a typical WSN, where the whole network has nine sensor nodes and a sink. Further, there are three virtual cells each having three sensor nodes. At the time of report gener-

ation, sensor nodes in each cell collaborate to prepare a report. As radio capabilities of sensor nodes are limited, reports are forwarded towards sink through intermediate nodes. For example, a report is created in the Cell 1 (collaboration between $Node_1(N_1), Node_2(N_2), Node_3(N_3)$), which is then forwarded to sink through intermediate nodes ($Node_4(N_4), Node_9(N_9)$). Data forwarding/authentication works on the application layer of the networking, where the network administrator cannot predict the intermediate hops for a particular report. So, the path discussed in the above example is decided by the underlaid routing protocol.

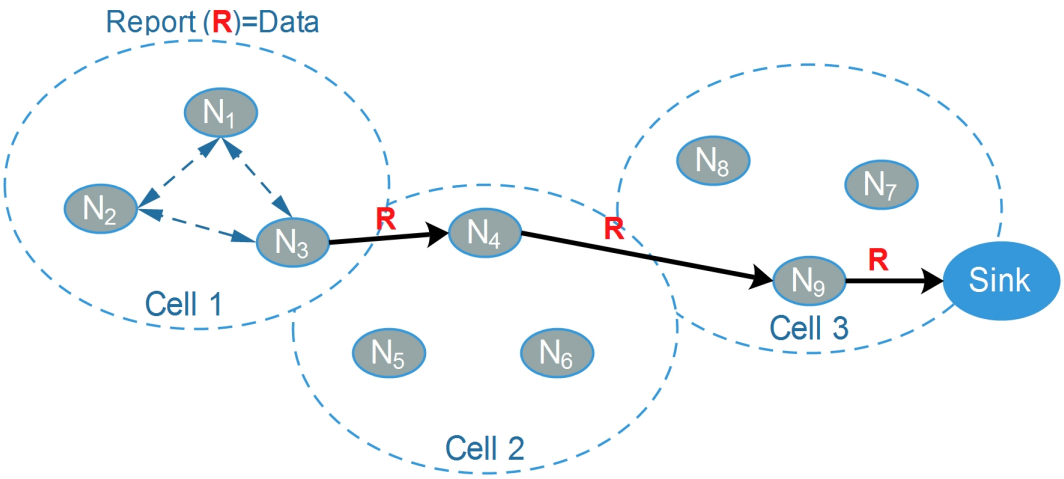


Figure 1.2: Example for Hop by Hop data transmission in WSNs

Now, En-Route filtering provides a method to verify each report which is being forwarded in the network. This helps in removal of false reports En-Route before they can reach the sink. As discussed earlier, this can be done by attaching MACs/signatures with the reports at the time of report generation. Thus, these MACs/Signatures can be verified by intermediate sensor nodes to check the authenticity of each report. If we consider same example of data forwarding as discussed in Figure 1.2, we can assign secret keys to the sensor nodes, which can be used to generate MACs and further be used for report verification. Specifically, secret key assignment looks like: N_1= $\{Key_{14}, Key_{19}\}$, N_2= $\{Key_{24}, Key_{29}\}$, N_3= $\{Key_{34}, Key_{39}\}$, N_4= $\{Key_{41}, Key_{42}, Key_{43}\}$, N_9= $\{Key_{91}, Key_{92}, Key_{93}\}$. Now at the report generation phase, $Node_1(N_1)$, $Node_2(N_2)$, $Node_3(N_3)$ collaboratively generate the report and MACs using secret keys (refer Figure 1.3). Thus, intermediate sensor nodes ($Node_4(N_4), Node_9(N_9)$) can ver-

ify the report by verifying the MACs associated with the report.

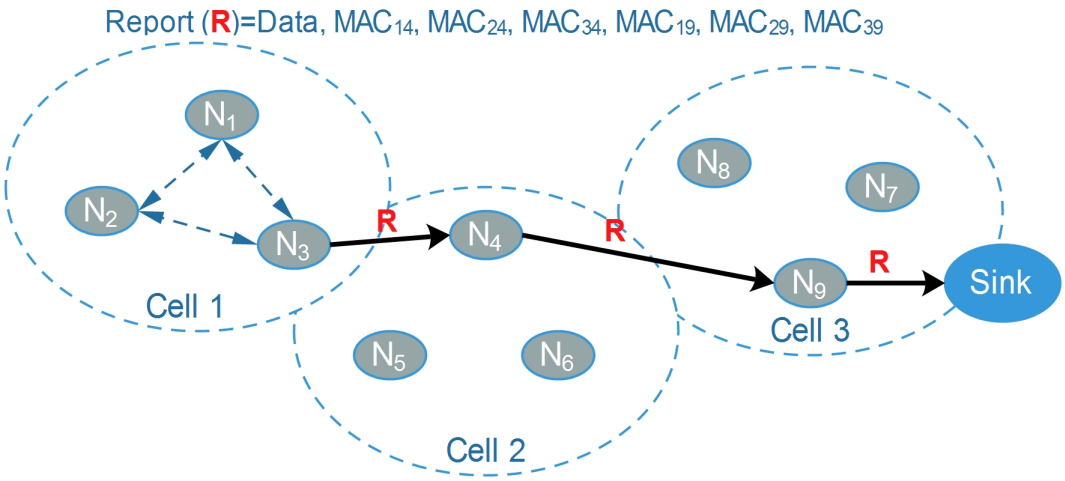


Figure 1.3: Example for En-Route data authentication and filtering in WSNs

But as discussed earlier, we cannot predict the data forwarding path for any report. Thus, to ensure En-Route filtering in the whole network, secret keys are to be assigned in the whole network. Specifically, for a network of n sensor nodes, each sensor node should store $n - 1$ shared secret keys. Further, each report has to carry $(n - 1) * (m)$ MACs, where m is the number of sensor nodes in each cell. Thus, the storage overhead and communication overhead with this naive En-Route filtering method is very high. So, the main aim of this research work is to reduce this storage and communication overhead associated with En-Route filtering of false reports in WSNs.

1.3 Motivation

WSNs comprise of a large number of sensor nodes which are very limited in computational and memory resources. The major use of these sensor nodes is to sense the environment where they are deployed, it could be temperature, humidity, pressure, fire, or movement, etc.. Because of these sensing capabilities sensor nodes are deployed in hostile environments like military monitoring, industrial sensing, environmental monitoring etc. for sensing and tracking purposes.

When a WSN is deployed, the sensor nodes sense the environment and send this data to sink (data collection node). The radio capabilities of each sensing node are very limited, so the sensed data is to be forwarded through intermediate nodes to reach the

sink. Sensing nodes deployed in hostile and unattended environments can be easily compromised, which can hamper the overall security of the network. These compromised nodes can be used to inject false or bogus reports in the network, which will unnecessarily increase the network traffic and can also cause the sink to take wrong decisions or raise false alarms. These compromised nodes can also be used to launch various DoS attacks, which can jeopardize the normal working of the network. Therefore, it is very important to drop the false reports from the network as soon as possible to limit the effect of an attack on the network. To drop the false reports as soon as possible many En-Route filtering techniques have been proposed. But because of use of En-Route filtering, adversaries can also launch DoS attacks against legitimate data using selective forwarding and report disruption attacks.

So need of the hour is to design filtering schemes to filter the false reports and reduce the effect of false reports on WSNs. But because of the existence of many attack vectors and because of inherited constraints of sensor nodes, it becomes very difficult to design schemes which can ensure confidentiality, authenticity and availability in the network.

1.4 Objectives

Though the contemporary techniques proposed in the field of En-Route filtering have evolved with the time, still the majority of them are prone to selective forwarding and report disruption attacks. In recent times, some asymmetric techniques have also been proposed which do not require any pre-shared keys in the network and which provides high detection rate and good resiliency against compromised nodes. But still, the implementation of asymmetric techniques is questionable on WSNs where sensor nodes have limited computation power. Majority of symmetric techniques require the exchange of keys which results in high communication overhead. The main goal of this research work is to invent the features to cope with different limitations of existing En-Route filtering techniques. This research work mainly focuses on proposing new key pre-distribution schemes and then to extend the proposed key pre-distribution schemes to propose new En-Route filtering schemes.

The first objective of this work is to develop new key pre-distribution schemes for

WSNs. Key pre-distribution can be done in many ways, easiest of all is to assign a single secret key to all the sensor nodes. But, the security of the whole network can be compromised instantly if the adversary can capture this secret key. A more practical approach for key pre-distribution is the assignment of unique pair-wise keys to each link between sensor nodes in the network. The resiliency of such setup is very high, as any compromised pair-wise key does not affect the remaining network. But, pair-wise keys based setup has huge key storage overhead, as each sensor node needs to maintain shared secret keys with all the other sensor nodes in the network. Combinatorial design (Anderson 1990) based key pre-distribution is like a middle ground, where we compromise resiliency of the network for saving storage overhead. Such design includes the assignment of a set of keys to all the sensor nodes in such a way that any given pair of key-sets have some shared keys. Thus in this work, our focus is to propose a novel combinatorial design based key pre-distribution schemes.

The second objective of this work is to extend the proposed key pre-distribution schemes to propose novel En-Route filtering schemes. Existing En-Route filtering schemes have no tolerance for selective forwarding attacks or report disruption attacks. Further, existing literature is not able to find a practical deterministic approach for En-Route filtering, thus filtering efficiency of existing schemes is poor and associated key storage overhead is high. Hence, the main aim of this work is to propose deterministic En-Route filtering methods which do not require sending reports through a fixed pre-defined path. Deterministic nature of the proposed methods ensure excellent filtering efficiency and use of combinatorial design based keys promise low associated key storage overhead. Further, this work focuses on proposing new data authentication/verification methods to ensure high resiliency against report disruption attacks and selective forwarding attacks.

To summarize, the primary objectives of this research work are as follows :-

1. To develop new efficient key pre-distribution schemes for WSNs.

2. To develop new symmetric cryptographic En-Route filtering methods which are more resilient to selective forwarding and report disruption attacks.

3. To develop new methods and techniques to decrease the communication overhead and key storage overhead in the En-Route filtering.

1.5 Contributions

This section includes the contributions of the research work done in the direction to find new solutions for existing problems in En-Route filtering of false data in WSNs. Major contributions of the research work are as follows :-

An extensive survey of existing En-Route filtering schemes has been done. The main aim of this survey (Kumar and Pais 2017) was to describe the major En-Route filtering techniques, to analyze these techniques on various parameters including security and to outline main unresolved research challenges in En-Route filtering of false data in WSNs.

In the direction to device a new key pre-distribution scheme, Combinatorial Design is adopted for this research work. Combinatorial design based key pre-distribution includes assignment of a set of keys to all the sensor nodes in such a way that, any given pair of key-sets has some shared keys. This way of the key assignment is an ideal balance between key storage overhead and resiliency against compromised sensor nodes in the network. For proposing a new scheme, a new combinatorial design is adopted to create key-sets and these key-sets are assigned to sensor nodes in such a way that the proposed scheme **(CD-KPD)** (Kumar and Pais 2018c) provides better resiliency than existing schemes. Further, a new shared key discovery algorithm is also proposed to secure the shared key discovery in the network.

To reduce key storage overhead in the previously proposed scheme (CD-KPD), two new modified schemes are proposed. In the first modified scheme **(CD-RKPD)** (Kumar and Pais 2018c), inter-cell communication for each cell is limited within its *Lee sphere* (Blackburn et al. 2008) region. So, now each cluster head requires less number of keys because of limited communication in the network. But, this affects the connectivity of the overall network. In the second modified scheme **(CD-PKPD)** (Kumar et al. 2019), key pre-distribution is limited to the partial network. In the proposed scheme, each cell has three cluster heads and key assignment between all the cluster heads was limited to

exactly $3/4^{th}$ of the total cluster heads. As the total number of cluster heads to whom each cluster head can communicate decreases, the number of keys stored by each cluster head decreases. But for keys assignment, the selection of $3/4^{th}$ of total cluster heads for each cluster head was done in such a way that the network still maintained proper connectivity and very good resiliency against compromised nodes in the network.

A new hybrid scheme **(CD-HKPD)** (Kumar and Pais 2018d) is proposed which uses both pair-wise keys and combinatorial design based keys to take advantages of both the worlds (pair-wise keys and combinatorial design based keys). In the proposed scheme, combinatorial design based keys are used to secure intra-cell communication, which helped to maintain low key storage overhead in the network. For inter-cell communication, each cell maintained multiple associations with all the other cells within the communication range, and these associations are secured with pair-wise keys. This helped to ensure high resiliency against compromised sensor nodes in the network while maintaining very low key storage overhead.

For the second part of the research, previously proposed scheme (CD-KPD) is extended to propose a new En-Route filtering scheme **(CD-EFS)** (Kumar and Pais 2018b; Kumar and Pais 2018a). For the same, a novel beam model is proposed to identify the upstream and downstream region of each cell to reduce the key storage overhead in the network. Finally, a novel report endorsement and verification technique is also proposed for robust data authentication and availability in the network. Continuing the second part of our research, CD-PKPD is extended to propose a partial key pre-distribution based En-Route filtering scheme **(CD-PEFS)**. Further, as the key assignment in CD-PKPD was limited to only $3/4^{th}$ of the network, suitable changes were made in the report endorsement and verification technique.

1.6 Thesis Organization

The remaining thesis is organized as follows: Chapter 2 provides the literature review of existing En-Route filtering schemes where we mention the advantages and disadvantages associated with all the discussed schemes. Additionally, this chapter provides unresolved problems in existing En-Route filtering schemes and various research direc-

tions to solve the same.

Chapter 3 presents three novel key pre-distribution schemes based on combinatorial design. Initially, we propose a combinatorial design based key pre-distribution scheme (CD-KPD). We then modify CD-KPD to propose a combinatorial design based reduced key pre-distribution scheme (CD-RKPD) and a combinatorial design based partial key pre-distribution scheme (CD-PKPD) to reduce the key storage overhead in the network. Chapter 4 presents a novel hybrid key pre-distribution scheme based on Combinatorial design keys and Pair-wise keys.

Chapter 5 presents a novel deterministic combinatorial design based En-Route filtering scheme (CD-EFS) by extending CD-KPD. In Chapter 6, CD-PKPD is extended to propose a partial key pre-distribution based En-Route filtering scheme (CD-PEFS).

Chapter 7 summarizes the contribution of this thesis and discusses future research directions.

CHAPTER 2

Literature Review

A substantial amount of work has been done to propose techniques and architectures to provide data authentication, including identification and removal of false data from the network. This chapter summarizes the literature survey for En-Route filtering and key pre-distribution techniques. This chapter can be broadly divided into two subparts. In the first part of the chapter, we provide a literature survey for En-Route filtering techniques. We also mention the advantages and disadvantages associated with all the discussed techniques. Further, we provide basic and security analysis of all the discussed techniques. Finally, we provide a critical and in-depth analysis of unresolved research challenges in data authentication in WSNs.

In the second part of the chapter we discuss various key pre-distribution schemes. Specifically, we discuss key pre-distribution schemes based on combinatorial design and key pre-distribution schemes which uses deployment knowledge. Due to the scope of the proposed schemes in the next few chapters, the survey of key pre-distribution is limited to these two methods only.

2.1 En-Route filtering techniques

All En-Route filtering based techniques can be classified in two ways (refer Figure 2.1), based on Cryptography and based on the probability of filtering. Classification based on Cryptography mainly has two sub-divisions, Symmetric Cryptography based

En-Route filtering techniques and Asymmetric Cryptography based En-Route filtering techniques. On the other hand, classification based on the probability of filtering can be sub-divided into three groups namely, Deterministic, Probabilistic and Hybrid.

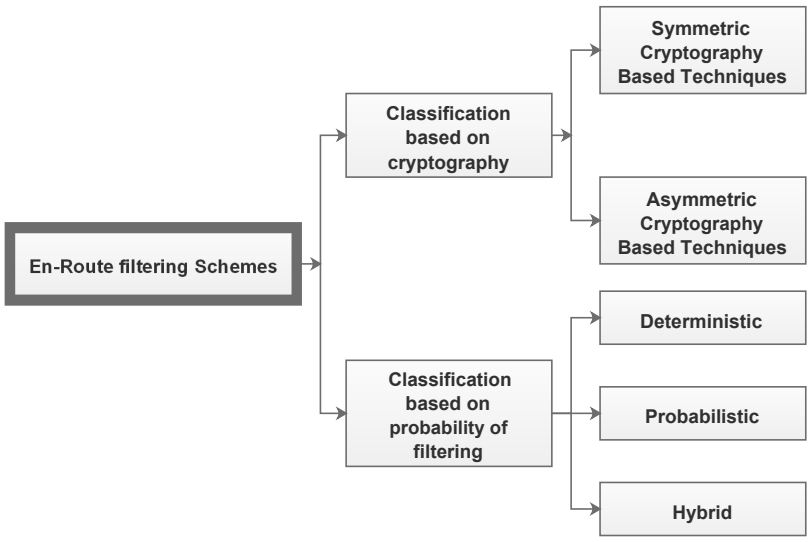


Figure 2.1: En-Route filtering techniques

Classification based on Cryptography:

Symmetric Cryptography based techniques have attracted huge attention and because of which large amount of work has been done in recent times in this area. Most of the techniques in this group provide authentication using MACs derived from symmetric keys. In these, all reports are to be attached with MACs and each legitimate report should carry certain minimum valid MACs. Each forwarding node checks the report using these MACs. Asymmetric Cryptography based techniques mainly rely on signatures, which can be checked by forwarding nodes and sink. These techniques do not require any pre-shared keys which is their major advantage over Symmetric techniques. Majority of these techniques use Elliptic Curve Cryptography (ECC) (Hankerson et al. 2006) and Shamir's threshold cryptography (Shamir 1979) to filter false reports. The secret required to generate the signature is shared among N neighbors so that no single node can create the signature and at least T nodes should collaborate to generate signature and report, where T ($T \leq N$) is predefined.

Classification based on the probability of filtering:

All the En-Route filtering techniques can also be classified based on the probability of filtering of false data. Based on the probability of filtering, techniques can be either Deterministic, Probabilistic or Hybrid. In Probabilistic techniques, sensor nodes are randomly selected in the network and secret keys are assigned to them. In Deterministic techniques, deterministic methods and processes are used to select and assign keys to fixed sensor nodes. Finally, in Hybrid techniques both probabilistic and deterministic methods are adopted for key exchange in the network.

Detailed information on the classification and En-Route filtering techniques is provided in Section 2.2 and Section 2.3.

2.2 Classification based on Cryptography

En-Route filtering based techniques use collaborative report generation, where each report has to be endorsed by a minimum T sensors nodes, where T is predefined. Moreover, each report is checked by the intermediate node to filter out false reports as soon as possible. Key assignment in the network for En-Route filtering is done mainly by two methods, Symmetric Cryptography based methods and Asymmetric Cryptography based methods. Each of these is explained in next subsections.

2.2.1 Symmetric Cryptography based techniques

A large number of techniques have been proposed which uses Symmetric Cryptography in En-Route filtering for authentication and filtering of false data from the network. Figure 2.2 provides a list of symmetric cryptography based techniques used in en-route filtering. Different Symmetric Cryptographic based techniques are explained below:

A) Global Key Pool Partition based En-Route filtering techniques

The basic idea behind the global key pool based techniques is sharing of common keys between few sensor nodes in the network, thus when the report is being forwarded by intermediate nodes, they can be checked by using those shared common keys. For the

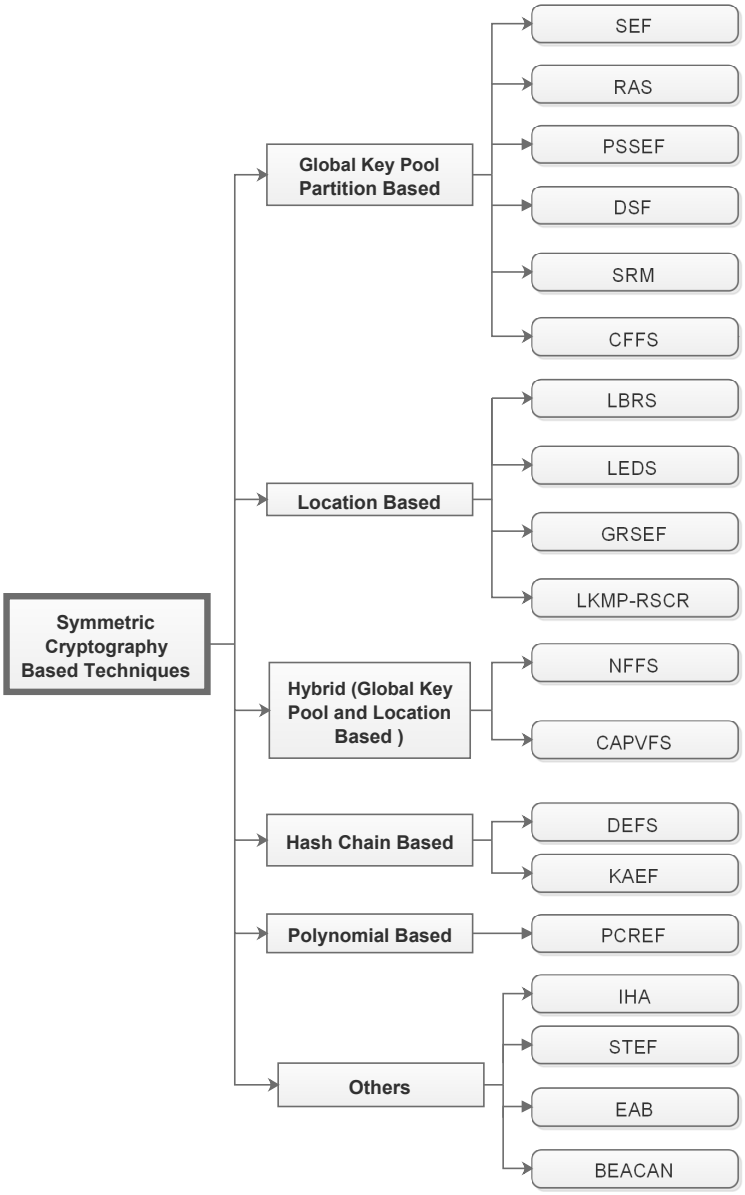


Figure 2.2: Symmetric Cryptography based techniques

same, there is a pre-generated global key pool $\{K_i|0 \leq i \leq N-1\}$ of N keys, each having a unique key index. This global key pool is further divided into n non-overlapping partitions $\{N_i|0 \leq i \leq n-1\}$, where each partition has m keys. At the time of keys assignment, each node randomly selects a partition and chooses some keys from it. Various Global Key Pool Partition based techniques are discussed below:

a) *Statistical En-Route filtering of Injected False Data* **(SEF)** : In this technique (Ye et al. 2005a) keys are assigned to sensor nodes using a global key pool. Whenever an

event occurs, multiple nodes simultaneously detect and create a report for the event. The report is like $\{L, t, E\}$, where L is the location of the event, t is detection time and E is the event. A Center-of-Stimulus (CoS) node is elected, which creates the final report. Moreover, the detecting nodes also create $MAC_M = MAC(K_i, L||t||E)$ where $MAC(a, b)$ computes MAC of message b using key a and $||$ denotes concatenation. CoS collects all the MACs and send them with the final report. These multiple MACs are used to check whether the report is genuine or not. All reports with forged MACs or insufficient MACs are dropped.

Each node is assigned keys from a single partition, this makes sure that a node can only create a single MAC. Thus, an adversary has to forge multiple MACs to completely create a bogus report. Each node selects a partition probabilistically thus common keys are shared with a certain probability between many nodes. So, forwarding nodes can check the correctness of reports probabilistically, thus dropping the forged reports enroute to the sink. Sink acts as a final goalkeeper for the whole network. Whenever it receives a report, it checks all the MACs attached to the report, as it has the global key pool.

b) *A Robust Authentication Scheme for Filtering False Data* **(RAS)** : In this technique (Hu et al. 2007) each node uses its secret keys assigned from a global key pool and dynamic authentication key tokens generated from a one-way hash chain to endorse the reports. This two-way endorsement make it very difficult for compromised nodes to mimic the behavior of normal nodes.

The global key pool and initial assignment of keys to each node are the same. Beside these keys, a unique authentication key is also shared between nodes and sink. Each node maintains a one-way hash chain of keys which is produced using the authentication key. Each report is endorsed by multiple sensors, but before the report can be sent and endorsed, a special *HELLO* packet is to be forwarded in the network. *HELLO* packet is of form $H = (TID, \{u_1, u_2, ..., u_t\}, \{a_0^{u_1}, a_0^{u_2}, ..., a_0^{u_t}\})$ where *TID* is used to identify filtering task, $\{u_1, u_2, ..., u_t\}$ are the node ids and $\{a_0^{u_1}, a_0^{u_2}, ..., a_0^{u_t}\}$ are the initial tokens. Cluster head creates this *HELLO* packet by getting information from other

nodes and forwards it to sink. Each forwarding node will record the node ID list and the tokens.

The report is endorsed after the cluster head broadcasts the report in the neighborhood and all the participating nodes agree to the report. Each participating node randomly selects one key, K_i, from all its keys to create a MAC for the event e, where MAC is $M = MAC(K_i, e)$. This MAC is used to encrypt the up-to-date authentication token, like $(a_0^u)_M$. Then, the node sends this encrypted token with the key index to the cluster head which collects all the endorsements and generates a final report. The final report is of the form $R = \{e, TID, i_1, (a_0^{u_1})_{M_{i_1}}, i_2, (a_0^{u_2})_{M_{i_2}}, ..., i_t, (a_0^{u_t})_{M_{i_t}}\}$, where e is the event, TID is the filtering task, $\{i_1, i_2..\}$ are the key indexes and $\{(a_0^{u_1})_{M_{i_1}}, (a_0^{u_2})_{M_{i_2}}..\}$ are the encrypted authentication tokens. This report is then sent toward the sink. Each forwarding node has a certain probability of posing the key used by nodes to generate the endorsements because of the partition overlapping between the nodes at the time of key assignment. If the forwarding node has a key to generate any endorsement, it verifies the particular authentication token to check the authenticity of the report. Finally, the base station checks all the authentication tokens and verifies the report.

c) *A Path Selection Method for Improving the detection power of SEF* **(PSSEF)** : In this technique (Sun et al. 2009) each sensor node evaluates the detection probability of forged reports through all incoming paths and selects the most secure path for data transmission. The basic structure of the technique is similar to SEF, where key exchange is based on a global key pool. To establish routing paths control messages are flooded in the network after the deployment of the nodes. In this technique, control messages are used for establishing routing path which has an array of bits to mark the partition ID of visited nodes. After control message flooding in the network, each node evaluates the incoming paths by marked bits in the array. If all the bits in the array are set, then a particular path can detect all the false reports. Further, if only a few bits are set in the array, then the majority of the reports will not be verified by intermediate nodes through a particular path.

d) *A Double Key-Sharing Based False Data Filtering Scheme* **(DSF)** : Sun and Wu (2011) proposed a double sharing based false data filtering technique (DSF) which assigns secret keys to the sensor nodes in two ways namely, associated key sharing and random key sharing. The associated keys are used for false data filtering within a few hops and random keys are used to provide data authenticity. Keys assigned from the global key pool are called *random keys*. A cluster head (CH) is selected by all participating nodes which creates a Hello message. This Hello message is of the form $\{y, CH, S_1, S_2, ...S_y\}$ where y is a counter, CH is the chosen cluster head and $S_1, S_2, ...S_y$ represents in-cluster nodes. This Hello message is forwarded toward the sink. Each forwarding node which receives the Hello message removes the last node's information from the Hello message, saves the information for further use, decrease the counter by 1 and add its information to the report. This process is repeated until the counter becomes 0. When counter becomes 0, the last forwarding node generates an ACK (Acknowledgment) message which includes the node ID. This acknowledgment is sent in reverse direction and finally, it reaches CH. Each node saves the node ID of its associated node using ACK message. These associated nodes exchange pairwise keys which are called *associated keys*.

In case of an event, CH broadcasts its readings to all the neighbors and each node create two MACs if they agree with the readings. The first MAC is generated using any one of the random key and another MAC using the associated key. The CH collects all the MACs and uses bloom filter (Bloom 1970) to convert all the MACs into smaller size strings to save overhead. The final report look like $R = \{C|e|R_1, R_2, ..., R_t|A_1, A_2, ..., A_t |F_1|F_2\}$, where C is the counter, e is the event, $\{R_1, R_2, ..\}, \{A_1, A_2, ..\}$ are the key indexes of the random and associative keys and F_1, F_2 are the compressed MACs. The final report with compressed MACs is forwarded using intermediate nodes. Each forwarding node can verify the report by generating MACs, if they have the associated keys which were used in the generation of MACs. As the sink has complete knowledge of all the keys, it serves as a final checker of the reports.

e) *A secure routing method for detecting false reports and wormhole attacks in wire-*

less sensor networks **(SRM)** : Choi et al. (2013) proposed a SEF based technique which uses secure routing method for detecting false data and worm hole attacks in the WSNs. False data is filtered out by En-Route filtering as proposed in SEF. Further, the proposed technique uses ACK messages for the detection of worm holes in the network. Key assignment and report generation is the same as proposed in SEF and so is the En-Route filtering. In En-Route filtering phase, if some intermediate node drops a report, it sends a drop message to the next intermediate node. The next intermediate node after receiving a drop message sends an ACK message to the initial sender of the packet. If any node which is expecting ACK or a drop message, do not receive the message, can assume the presence of a worm hole in the network. So these drop and ACK messages are used for wormhole detection in the network. But the proposed technique does not improve any limitations of SEF and thus suffers from low resiliency.

f) *A Cluster-Based False Data Filtering Scheme in Wireless Sensor Networks* **(CFFS)** : Liu et al. (2014) proposed a cluster-based false data detection technique which groups the sensor nodes into clusters and all the clusters form a tree-like structure using the cluster heads. There are 5 phases in CFFS, first is the pre-deployment phase. In this phase, the technique calculates the value of a new parameter called burden. The value of burden depends on the distance of a particular cluster head from the sink and the number of paths going through that particular cluster head. This parameter is calculated by each cluster head and is used in further phases. After burden calculation, each cluster head constructs a tree rooted at the sink. This can also be termed as path discovery for the technique. Next phase is distributed key assignment, where each cluster head uses burden value calculated in the previous phase to assign a different number of keys to all the upstream nodes. This is followed by report generation phase, where in case of an event, participating nodes create and endorse the reports with MACs. Cluster head creates the final report and forwards the report toward the sink. Next phase is En-Route filtering where upstream nodes filter out false reports by generating the MACs using the keys exchanged in the previous phase and matching it with MACs present in the report. Each upstream node also checks whether participating nodes for report generation are

from a single cluster or not, this helps to detect collaborative false data injection attack in the network. The final phase is sink verification where sink verifies the report by verifying all the MACs present in the report.

Discussion- Table 2.1 gives the summary of techniques which assign keys based on global key pool partition. The table lists the pros and cons associated with each technique followed by the contributions of each. In all these techniques sensor nodes choose a particular key pool partition and choose some keys from it. At the time of report endorsement, keys from at least T different partitions are needed. So, if an adversary gets the secret keys from T partitions, it can successively endorse fake report as genuine. This is called the T-threshold limitation. The majority of techniques which uses global key pool partition for key assignment are prone to T-threshold limitation. But, as the keys are assigned from a global key pool and no further message exchange is needed for exchanging keys, scalability of the network is high and such techniques can easily adapt to dynamic networks.

B) Location Based En-Route filtering techniques

In these techniques, secret keys are bound to sensor nodes based on their geographical location. This helps to overcome the T-threshold limitation. Techniques using this approach are explained below:

a) *Toward Resilient Security in WSNs* **(LBRS)** : Yang et al. (2005) proposed a location-based approach for secret keys assignment. Secret keys are assigned to nodes based on the geographical location of nodes. Initially, each cell is assigned a master secret K^I and location information of its center, which is (X_i, Y_i) such that $\{X_j = X_0 + j.C, Y_k = Y_0 + k.C; j, k = 0, \pm1, \pm2, ...\}$, where (X_0, Y_0) is the sink location. Each node uses the cell's center location information and uses the localization scheme (Tsudik 1992) to derive its keys. This location-based binding of keys limits the scope of misuse of keys by compromised nodes.

Table 2.1: Pros and Cons of Global Key Pool Partition based En-Route filtering techniques

Technique	Pros	Cons	Contribution
	Global Key Pool Partition Based		
SEF (Ye et al. 2005a)	- Maintenance is very low. - Independent of the data dissemination protocol. - Works well on networks with dynamic topology.	- Has a T-threshold limitation. - Prone to selective forwarding attack and report disruption attack.	It uses grouping based shared key mechanism. It also used the bloom filter to reduce the communication overhead.
RAS (Hu et al. 2007)	- Double authentication is very effective against compromised nodes. - Can adapt to dynamic networks.	- Large overhead of sending *HELLO* packet. - Prone to selective forwarding and report disruption attacks.	The technique uses double authentication (secret keys and one-way hash chain) to endorse each report.
PSSEF (Sun et al. 2009)	- Take into account all the available path and choose the best one. - More effective than SEF technique.	- Huge time and communication overhead. - Prone to selective forwarding and report disruption attack.	The technique evaluates all the available paths from a node to base station and choose the best one for forwarding report.
DSF (Sun and Wu 2011)	- Double checking of reports using two MAC. - Bloom filter is used to reduce overhead.	- Not feasible for dynamic networks. - Prone to selective forwarding and report disruption attacks.	The technique gave a double sharing based false data filtering mechanism.
SRM (Choi et al. 2013)	- Independent of data dissemination protocol. - Can detect false reports and wormholes in the network.	- Has T-threshold limitation. - Prone to selective forwarding and report disruption attacks.	The technique has proposed a SEF based technique which using secure routing method for detecting false data and worm hole attacks in the network.
CFFS (Liu et al. 2014)	- Can detect collaborative false data injection attacks.	- Is prone to T-threshold limitation, selective forwarding and report disruption attacks. - Cannot adapt to dynamic networks.	The technique has given a cluster based false data detection mechanism which groups the sensor nodes into clusters and all the cluster form a tree-like structure using cluster heads.

Initially, the terrain is divided into virtual grids where each grid or cell is bounded with multiple keys having unique indexes. These keys are termed as *location-binding keys*. These keys are assigned to each node in a cell based on their location which are derived using a secure one-way function $H(.)$ (Royer and Toh 1999) i.e. $K_{X_i,Y_j} = H_{K'}(X_i||Y_j)$. Each node is assigned two types of keys, namely *sensing cell keys* and *verifiable cell keys*. Sensing cell keys are used for validating reports for events detected in the same area by neighboring nodes. Verifiable cell keys are used by intermediate verifiable cells to verify the reports coming from the cells with whom they shared keys.

Each report to be termed legitimate should contain M distinct MACs which are generated by sensing nodes using their keys which are bound to the event's cell. Upon event happening, all the nodes agree on event description, including the location. They

independently generate the MAC and broadcasts it. The final report with the MACs is sent towards the sink.

Intermediate nodes upon receiving the reports checks for M distinct MACs in the report. If M distinct MACs are not found, the report is dropped. Further, they find the event location and checks whether it is one of the remote cells whose key they have. If intermediate nodes have any such key, they compute the MAC and compares it with the report's MAC. If any mismatch happens, the report is dropped, else it is forwarded. The sink acts as the final checker. As sink has all the location binding keys, it can compute all the MACs and check all the endorsements carried by the report.

b) *Providing Location Aware End-to-end Data Security* **(LEDS) :** The technique (Ren et al. 2008) proposed a location-based end to end security framework where secret keys are assigned to the nodes based on their geographical location. It follows an interleaved hop by hop filtering and forwarding technique and uses a cell-based report generation scheme. Two major components of the technique are key management and end to end data security.

Key management framework- In this, each sensor node is assigned three different keys based on its location. First, a unique *secret key*, shared between the sensor node and the sink, which is used by the sink for authentication of reports. The *secret key* is derived using $K_s = H(K_M^I|u|I_u)$, where K_M^I is the master secret key, u is the node's ID and I_u is the cell location where node is present. Second, the *cell keys*, shared among all sensor nodes in a particular cell which are used for providing data confidentiality. The *cell keys* are derived using $K_c = H(K_M^I|I_u)$, where K_M^I is the master secret key and I_u is the cell location where node is present. Third, a set of *authentication keys*, shared between report-authentication cells which provides cell to cell authentication of reports. The *authentication keys* are derived using $K_a = H(K_M^{II}|(x_c, y_c)|(x_1, y_1))$, where K_M^{II} is the master secret key, (x_c, y_c) is the location of authentication cells and (x_1, y_1) is node's cell location.

End to End data security- In this, each report is encrypted with corresponding keys of the event cell. As these keys are only shared between nodes of the cell and sink, it

provides needed data confidentiality. Each report is also checked by other nodes in the forwarding route, and finally, the report is checked by the sink. Each report is divided into unique shares individually generated by each participating node. A set of MACs are also calculated and attached with the report which makes the technique robust to report disruption attack. The technique makes sure that each report is forwarded and verified simultaneously through multiple nodes by sending the same report by multiple paths to the sink. This one-to-many data forwarding and filtering approach makes the technique robust to selective forwarding attack.

c) *Grouping Based Resilient Statistical En-Route filtering* **(GRSEF) :** This technique (Yu and Li 2009) groups all the nodes without any overlap and proposes a location-aware approach which derives the secret keys based on the division of terrain along multiple axes. At the time of deployment, each sensor node is assigned a global key K_g, size of terrain, the shape of group, reference point (x_0, y_0), key-sharing probability q, parameters T and list of division parameter triples.

During the initial phase, each sensor node joins a group by running a grouping algorithm. There are T groups covering all the nodes. The parameters for multiple axis division are $\{(j, Q_j, W_j)|x \in 1, ..., T, -\pi \leq Q_j \leq \pi, Q_1 < Q_2 < ... < Q_T, Q_T - Q_1 < \pi\}$, where j is axis number, Q_j is the angle between x-axis and j-th axis and W_j is the division width along j-th axis. The terrain is divided into multiple partitions along j-th axis using $l = (L_j/W_j)$, where L_j is the length of projection of terrain on j-th axis. Based on (j, Q_j, W_j) and reference point, node calculates the partition co-ordinates. Nodes derive group master key K_{g_j} using $K_{g_j} = H_{K_g}(j)$, where K_g is the global master key and $H(*)$ is a secure one-way hash function. Each node uses the partition co-ordinates and group master key to derive the *location aware authentication keys*.

In case of an event, all the detecting nodes organize in a cluster and they all reach on an agreement including the location of the event. Each detecting node generates a MAC using its *authentication key* bound to its group and the partition, where the event is located at a particular axis. The MACs along with the group number are sent to the cluster head by all the participating nodes. The cluster head collects at least T MACs,

add them to the final report and send it to the sink.

When any forwarding node receives the report, it computes the partition coordinates of the event based on the axis and can know whether it owns the authentication key for that partition. Using this authentication key, it can verify that particular MAC in the report and can drop the report if any mismatch happens. When a report reaches the sink, it checks all the MACs, as it has the global key which was used to generate all the partition binding keys along each axis.

d) *The Optimum Design of Location-Dependent Key Management Protocol for a WSN With a Random Selected Cell Reporter* **(LKMP-RSCR) :** Fakhrey et al. (2016) proposed a new location dependent key management protocol. They suggested the use of randomly selected cell reporters for the signature generation which is used to endorse reports. This provides a more secure and robust endorsement mechanism. In the technique, each report to be true should have unique individual data, MACs generated by authentication nodes and cell reporter's signature. For unique individual data, the authors proposed the use of (e, n) threshold linear secret sharing scheme (LSSS) (Shamir 1979). This ensures that at least n sensor nodes participate in creating a report. The MACs for the report are derived using authentication keys shared between authentication nodes and the base station. The third endorsement is signature, which includes a signature from a randomly chosen fixed number of cell reporters. In En-Route filtering phase, MACs in the report are used by intermediate nodes to filter out false reports. Finally, the base station checks the signatures and MACs in the report to check the authenticity of the reports.

Discussion- Table 2.2 gives the summary of techniques which assign keys based on geographical location of nodes. The table lists the pros and cons associated with each technique followed by the contributions of each. All these techniques bind secret keys to the sensors using their geographical location, they also share some keys with their forwarding nodes. These keys are used to check the authenticity of reports while forwarding them towards the sink. But as the keys are shared between few forwarding

Table 2.2: Pros and Cons of Location Based En-Route filtering techniques

Technique	Pros	Cons	Contribution
		Location Based	
LBRS (Yang et al. 2005)	- Resilient to increase of compromised nodes. - Compromised nodes cannot move in network easily. - Compromised keys cannot be used for global usage.	- Cannot be applied on dynamic networks and requires node's localization. - Prone to selective forwarding and report disruption attack.	Technique used the location-binding key generation to overcome T-threshold limitation.
LEDS (Ren et al. 2008)	- Can adapt to large number of compromised nodes. - Robust to selective forwarding and report disruption attacks.	- Uses specific data dissemination route for data forwarding and requires node localization. - Cannot be applied to dynamic networks.	Technique used location aware key to provide end to end security and to avoid T-threshold limitation.
GRSEF (Yu and Li 2009)	- Multiple axis division avoids T-threshold limitation. - Can work with mobile sink and special data dissemination protocols.	- Cannot be used in dynamic networks. - Prone to selective forwarding and report disruption attacks. - Requires node localization.	In place of redundant groups, GRSEF divide all the sensor into T groups to provide resilience against T-threshold limitation.
LKMP-RSCR (Fakhrey et al. 2016)	- Security parameters are renewed in every small time interval. - Resilient to Sybil and Selective forwarding attacks.	- Report carries encrypted data, thus authentication nodes cannot verify the data in reports.	Technique has proposed a double authentication method where each report is endorsed by neighboring nodes of a cell and by a fixed number of cell reporters.

nodes and that too only in the direction of the sink, these techniques cannot adapt to dynamic networks. Majority of these techniques are also prone to selective forwarding and report disruption attacks.

C) Hybrid En-Route filtering techniques

Hybrid techniques uses both global key pool and location information to bind secret keys to the sensor nodes. The techniques using this method of key binding are explained below:

a) *Defending Collaborative False Data Injection Attacks* **(NFFS) :** The technique (Wang et al. 2014) has given a geographical information based technique which uses the position of nodes for false data detection, which authors further modifies to propose NFFS which uses neighbors information for false data detection.

In this technique, each node acquire some keys from a global key pool, same as other global key pool based techniques (Ye et al. 2005a; Hu et al. 2007; Sun et al.

2009). Each node also acquire its location $L_i = (X_i, Y_i)$ using any location algorithm (Rabaey and Langendoen 2002). Using multi-cast algorithm (Wang et al. 2014), each node exchange its information (S_i, L_i, U_i) with all the intermediate nodes, where S_i is the node's index, L_i is node's location and U_i denotes the key partition index. Each node stores information of other nodes with probability $P = (c/N)$, where c is the number of intermediate nodes and N is the total number of nodes deployed in the network. In case of an event, CoS finds the location of stimulus L_s. The CoS broadcasts its reading to all the detecting nodes. All the detecting nodes in the same area generate a MAC $M = K_i(e)$, where K_i is one of the authentication keys stored in the node and e is the event. The CoS finally generate the report by including all MACs and other parameters and forwards it to sink. The final report is of the form $\{e|L_s|i_1, i_2, ..., i_x|M_{i_1}, M_{i_2}, .., M_{i_x}$ $|j_1, j_2, .., j_x|L_{j_1}, L_{j_2}, .., L_{j_x}\}$, where e is the event, L_s is the location of event, $\{i_1, i_2, ..., i_x\}$ are the key index, $\{M_{i_1}, M_{i_2}, .., M_{i_x}\}$ are the MACs for the event, $\{j_1, j_2, .., j_x\}$ denotes the node indexes and $\{L_{j_1}, L_{j_2}, .., L_{j_x}\}$ denotes the location of the participating nodes. Each forwarding node in the path to sink can verify the correctness of MACs using the keys and the location information, which was exchanged in the initial phase of nodes deployment. But, a lot of energy is wasted by each node to be aware of its location. To overcome this, the author proposed NFFS which uses relative positioning (Patwari et al. 2003) of sensor nodes to filter false data.

In NFFS, each node selects a distinct key from a key pool. After this, each node broadcast its ID and key. CoS generate a *hello* packet which includes all the information of the neighbors including keys and it forwards *hello* packet to the sink. The *hello* packet is of the form $\{S_i|S_{a_1}, S_{a_2}, ..., S_{a_j}|K_i|K_{a_1}, K_{a_2}, .., K_{a_j}\}$, where S_i, K_i are the CoS node index and key respectively, $\{S_{a_1}, S_{a_2}, ..., S_{a_j}\}$ are the indexes of all the intermediate nodes and $\{K_{a_1}, K_{a_2}, .., K_{a_j}\}$ denote the keys of intermediate nodes. Each intermediate node selects a node S_{a_x} from the *hello* packet and elects itself as its authentication node based on the probability $P = (h/H)$, where h denotes the number of hops between CoS and sink and H denotes the number of hops between S_{a_x} and sink. If S_{a_x} is elected as authentication node, intermediate node stores S_{a_x} information and removes S_{a_x} entry from the *hello* packet and adds its own entry in it. This modified *hello* packet is then

forwarded to the next intermediate node and this process is repeated until sink receives the *hello* packet. This *hello* packet makes a proper selection of authentication nodes for all the sensor nodes in the network. Report generation includes generation of MACs by each neighbor and then forwarding of final report including the MACs by CoS. All the authentication nodes can verify the relative location of all the participating nodes and correctness of the MACs.

b) *Context-Aware Architecture for Probabilistic Voting-based Filtering Scheme in Sensor Networks* **(CAPVFS) :** Nam and Cho (2016) proposed a Context-Aware Architecture (CAA) for filtering of false data in wireless sensor network. The proposed technique can be integrated with any existing En-Route filtering technique to provide more robust false data detection and identification of compromised nodes in the network. The proposed technique includes three architectures such as- any existing En-Route filtering technique, Comm-Arch and CAA. For the first architecture, the authors have used the PVFS (Li and Wu 2006) technique which is probabilistic voting based En-Route filtering technique. In Comm-Arch, there are Data Collection Nodes (DCNs) spread across the network. These DCNs collect the sensed data from the sensor nodes and sends it to CAA. CAA transforms the data into spatio-temporal data. Further, CAA analyzes the data against normal behavior of network with the knowledge of different attacks possible over the network. This analysis helps to identify compromised nodes which violate CAA parameters. So, the normal En-Route filtering is carried forward by the first architecture and further analysis of reports is done by the third architecture. This provides a double analysis of data in the network.

Discussion- Table 2.3 gives the summary of techniques which assigns keys based on global key pool partition and geographical location of nodes. The table lists the pros and cons associated with these techniques followed by the contributions of all. These techniques use relative position of nodes to assign keys, which is more energy efficient than other location-based techniques. These techniques also share keys with intermediate nodes at a fixed path, so reports are to be sent through fixed paths.

Table 2.3: Pros and Cons of Hybrid En-Route filtering techniques

Technique	Pros	Cons	Contribution
	Hybrid(Global key pool partition and Location Based)		
NFFS (Wang et al. 2014)	- Used the relative position of nodes which is more energy efficient as compared to location-based techniques. - Do not have T-threshold limitation.	- Cannot be applied on dynamic networks. - Data can only be checked on a fixed path to the sink. - Prone to selective forwarding and report disruption attacks.	Used the relative position of nodes which is more energy efficient as compared to location-based techniques.
CAPVFS (Nam and Cho 2016)	- Compromised nodes can be identified and removed from the network. - Existing attacks can be identified very easily. - High resiliency against compromised nodes.	- Prone to selective forwarding attack. - Energy is wasted by CAA for data analysis and attacks detection. - Cannot adapt to dynamic networks.	Technique proposed a context based architecture for analysis of sensor data. This added layer with En-Route filtering provides robust and secure network even in presence of compromised nodes.

D) Hash Chain based En-Route filtering techniques

In hash chain based techniques, each sensor node uses a hash chain of authentication keys for endorsement of reports. For authentication of reports, each node disseminates its present key to the intermediate nodes. Hash chain based techniques are explained below:

a) *A Dynamic En-Route filtering Scheme* **(DEFS) :** This technique (Yu and Guan 2010) proposed a dynamic En-Route filtering mechanism where each node maintains a hash chain of authentication keys which are used to authenticate reports. In case of an event, the cluster head collects the sensed data from all the cells, aggregate them and forward them to the base station. Moreover, each report also contain MACs which are generated using authentication keys.

Each node is assigned a distinct seed key $K_m^{v_i}$ at the time of deployment. Each node uses this distinct seed key to generate a sequence of *auth-keys* using a hash function H. Thus each node maintains a hash chain of keys, i.e. $K_{m-1}^{v_i} = H(K_m^{v_i})$, $K_{m-2}^{v_i} = H(K_{m-1}^{v_i}) = h^2(K_m^{v_i})$, ..., $K_1^{v_i} = H^{m-1}(K_m^{v_i})$, where $K_m^{v_i}$ is the seed key for a particular node v_i and m denote the length of hash chain. The keys of the hash chain are used in reverse direction to its generation sequence. Before forwarding the report,

auth-key of all the participating nodes are disseminated by cluster head. The reports are sent in rounds and in each new round nodes use a new *auth-key* to authenticate the reports. For verification by forwarding nodes, participating node discloses their *auth-keys* by the end of each round. After receiving auth-keys, all the forwarding nodes verify the report and instructs its next hop to either drop or forward the report based on results.

Specifically, the technique has three phases: *Key pre-distribution phase*, in which each node is assigned a distinct seed key which is used to maintain a hash chain of *auth-keys*. *Key dissemination phase*, in which cluster head disseminates each node's first *auth-key* to forwarding nodes. *Report forwarding phase*, in this all the forwarding nodes use the *auth-key* shared in the previous phase to verify the report. If the report is verified, the forwarding nodes disclose the *auth-key* to next-hop node.

b) *An En-Route Scheme of Filtering False Data* **(KAEF) :** This technique (Yuan et al. (2008)) has given an En-Route filtering mechanism based on one-way key chain authentication which uses a one-way key chain for endorsement and verification of reports. At the time of deployment, each sensor node is assigned a one-way key chain (Lamport 1981) which contains m+1 keys, denoted by $\{K_i^0, K_i^1, ..., K_i^t\}$. This key chain is maintained using a one-way hash function. The keys of the chain are used in reverse direction to its generation sequence. To secure the communication between any two sensors, the technique uses LEAP (Zhu et al. 2006) which guarantees secure communication link. After deployment, nodes are grouped in clusters and a cluster head is elected for each cluster. In case of an event, sensing nodes send the reading R to the cluster head. Cluster head creates the final report and broadcasts it to all the sensing nodes. Each sensing node checks the report and generates a MAC denoted by $\{MAC(K^0, R)\}$ using an unused key K^0. After getting at least t endorsements, cluster head creates the final report containing data and MACs and forwards it to the sink. The report is of the form $\{R, i, i_1, ..., i_{t+1}, M_{i_1}^{p_{i_1}}, M_{i_2}^{p_{i_2}}, ..., M_{i_{t+1}}^{p_{i_{t+1}}}\}$, where R is the data, $\{i, i_1, ..., i_{t+1}\}$ are the ids of nodes participating in report generation and $\{M_{i_1}^{p_{i_1}}, M_{i_2}^{p_{i_2}}, ..., M_{i_{t+1}}^{p_{i_{t+1}}}\}$ are the corresponding MACs. But before sending report, an initialization message is sent along the path to sink which is of form $\{i, i_1, i_2, ..., i_N, K_{i_1}^0, K_{i_2}^0, ..., K_{i_N}^0\}$, where $\{i, i_1, ..., i_N\}$ are the IDs

of all the nodes in cluster i and $\{K^0_{i_1}, K^0_{i_2}, ..., K^0_{i_N}\}$ are the unused key. Each intermediate node randomly stores some values from the initialization message and forwards the message to the next hop. So, now when the report is forwarded, each forwarding node verifies the report depending on the probability of containing the authentication information shared with endorsing nodes. The sink acts as the final goalkeeper who can verify the report with probability one since it contains the authentication information for all the endorsing nodes.

Discussion- Table 2.4 gives the summary of techniques which assign keys based on the hash chain. The table lists the pros and cons associated with each technique followed by the contributions of each technique. All the techniques explained above use a hash chain of secret keys for endorsement. Moreover, each key has to be disclosed to forwarding nodes for authentication of these reports. Since each time a new key is used for endorsement, these techniques provide good resiliency against adversaries. But, this adds a huge overhead of sharing keys with intermediate node.

E) Polynomial based En-Route filtering techniques

These techniques use polynomials at the place of MACs for endorsement and authentication of the reports. Each node stores few polynomials which are assigned at the time of deployment. A technique using this method is explained below:

a) *A Novel En-Route filtering Scheme Against False Data Injection Attacks* **(PCREF) :** This technique (Yang et al. 2015) uses polynomials in place of MACs to endorse and verify the reports. The technique divides the WSN into clusters and each cluster has nodes which can monitor the same components. At the time of deployment, each node is assigned $\{K_x, f(x, y, z), T, H(*)\}$, where K_x is the master key, $f(x, y, z)$ is the primitive polynomial element, T is the threshold and $H(*)$ is a hash function. Each node computes the authentication-polynomial for the cluster, i.e. $auth^u_i(y, z) = \alpha f_i(u, y, z)$, where u is the sensing node ID and $\alpha \in \{2, 2^2, 2^3, 2^4\}$. The value of α is chosen ran-

Table 2.4: Pros and Cons of Hash Based En-Route filtering techniques

Technique	Pros	Cons	Contribution
		Hash Chain Based	
DEFS (Yu and Guan 2010)	- Is independent of data dissemination protocol. - Can be applied on dynamic networks. - Robust to selective forwarding and report disruption attacks.	- Has T-threshold limitation. - Include large number of extra control messages. - Difficult to implement in energy efficient networks.	Technique uses a dynamic En-Route filtering mechanism which periodically update the authentication keys. Technique also uses Hill Climbing key dissemination approach to improve filtering efficiency.
KAEF (Yuan et al. 2008)	- As one key is only used once in each node, the security is really good.	- As each node only has at most $m+1$ keys, the network has to be re-initiated after all the keys have been used by nodes.	The technique gives an En-Route filtering technique based on one-way key chain authentication.

domly by each node. After this, each node calculates the verification-polynomial. Each node probabilistically computes and store some verification-polynomials from all the clusters. The verification-polynomials are computed by using formula (Yang et al. 2015) $verf_j^u(x,z) = \beta f_j(x,u,z)$, where $\beta \in \{2^5, 2^6, 2^7, 2^8\}$. The use of α and β makes En-Route filtering more efficient. The different clusters use different primitive-polynomials. Thus, each cluster generates different authentication-polynomials and check-polynomials. Authentication-polynomials are used to endorse the reports in a cluster and check-polynomials are used to verify the reports in other clusters. Each node also generates cluster keys using $K_{C_i} = F(K_C, CH_i)$, where K_C is the master key, CH_i is the cluster head ID and $F(*)$ is the cluster key generation function. These cluster keys are used to encrypt the report E, on which further hash function $H(*)$ is applied to compute z, i.e. $z = H((E)_{KC_i})$.

At the time of report generation, each node creates the authentication-polynomial MAP = $auth_i^u(y,z) = \alpha f_i(u, y, H((E)_{KC_i}))$, which is sent to the cluster head. Cluster head collects the sensed data and MAPs from all the sensing nodes to create the final report. The final report is of form $\{((E)_{KC_i})|C_i|u_1, ..., u_T|C_i^{j_1}, ..., C_i^{j_T}|auth_i^{u_1}(y, H((E)_{K_{C_i}})), ..., auth_i^{u_T}(y, H((E)_{K_{C_i}}))|T\}$, where C_i is the cluster ID, T is the time stamp and other parameters are same as discussed above. This final report is forwarded toward the sink. Intermediate nodes upon receiving the report checks for T distinct MAPs, the time stamp and verifies the MAPs according to the stored check-polynomials. Any error in checking phase results in the dropping of the report. The sink acts as a final goalkeeper

as it has all the check polynomials for participating nodes in report generation.

Discussion- Table 2.5 gives the summary of the technique which assigns the secret keys based on primitive polynomials. The table lists the pros and cons associated with the technique followed by the contributions of it. In the above technique, all the nodes are divided into the cluster and each cluster is assigned a primitive polynomial and using this polynomial all the sensor nodes in that cluster derive their authentication-polynomial and verification-polynomials. Because of this, it is prone to T-threshold limitation. Moreover, the technique cannot defend against report disruption and selective forwarding attacks.

F) Other En-Route filtering techniques (Miscellaneous)

These techniques use miscellaneous methods for key assignment, report endorsement and use En-Route filtering at the back-end to filter out false reports. Techniques using miscellaneous methods are explained below:

a) *An Interleaved Hop-by-Hop authentication scheme* **(IHA) :** This technique (Zhu et al. 2004) has given an interleaved hop-by-hop authentication mechanism. During initialization, sink loads every node with IDs and necessary keys which are needed to establish pairwise keys with other sensor nodes. At the time of deployment, each sensor node exchange secret keys with all its neighbors.

When an event has to be reported, minimum $T + 1$ nodes are required which generates the report collaboratively. Each participating node generates two MACs, one using pairwise key shared with upper associated nodes and other using the key shared with the base station. All the MACs are collected by the cluster head and are attached to the report. The final report is then forwarded to the base station.

Every forwarding node after getting the report, verifies the MAC generated by its lower association node, if MAC matches, it replaces this MAC with the new one calculated using pairwise key shared with its upper association node. When the base station

Table 2.5: Pros and Cons of Polynomial Based En-Route filtering techniques

Technique	Pros	Cons	Contribution
		Polynomial Based	
PCREF (Yang et al. 2015)	- No need of node localization. - Do not rely on fixed routes for data transmission.	- Prone to selective forwarding and report disruption attacks. - Has T-threshold limitation.	The technique is first of its kind which have used polynomials in place of MACs to endorse and verify reports.

receives the report, it can check whether the MACs retained in the report are correct or not using the key shared with the participating nodes.

b) *A secure ticket based En-Route filtering scheme* **(STEF) :** Kraub et al. (2007) proposed a light weight one-way function based technique which gave a ticket concept, where messages are forwarded only if they have valid tickets. This concept is implemented using a query-response communication between nodes. At the time of deployment, each node is assigned a unique key shared with the sink. After the deployment, each node obtains its location using any localization scheme and also establishes a pair wise key with its neighbors using existing techniques. To receive the data, the sink generates a query and a ticket. The query message is of the form $\{QID, c^1, Q, (c)_{K_{CH}}\}$, where QID is query identifier, c is the ticket and c^1 is the hash value of ticket which is generated using $c^1 = h(c)$, Q is the query and $(c)_{K_{CH}}$ is the ticket c encrypted using key K_{CH} shared between the sink and that particular cluster head. This is forwarded to a chosen cluster head. Each intermediate nodes store the QID and c^1 for further uses.

After receiving the query, the cluster head generates the report and broadcasts the report to neighbors. Each neighbor generates a MAC for the report if they agree with the report being sent by the cluster head. Cluster head collects all the MACs and compresses the MACs using bitwise XOR operation, i.e. $CMAC = MAC(R, K_{S_1}) \oplus MAC(R, K_{S_2}) \oplus .. \oplus MAC(R, K_{S_T})$. The final report is formed which is of the form $\{QID|R|c|CMAC|(ID_1, ID_2, ..., ID_T)\}$, where c is the decrypted ticket from the query, $CMAC$ is the compressed MAC and $(ID_1, ID_2, ..., ID_T)$ is the ID of participating nodes. The response is forwarded using the same path in reverse direction. Each intermediate node authenticates the report using immediate authentication schemes, e.g., RPT or LEA (Luk et al. 2006). Each intermediate node checks whether each report has appro-

priate QID and verifies the value of c by calculating the hash of it and comparing it with stored c^1 value. Sink performs the final verification of the report.

c) *Filtering False Messages En-Route* **(EAB) :** This technique (Chen and Lei 2010) has used bloom filter techniques (Bloom 1970) to build an authentication block, which is called *En-Route Authentication Bitmap* (EAB). The technique uses bloom filter for hop-by-hop authentication to create a set of MACs for different forwarding nodes. The source node initially generates signature $sign$ on the message M which is used to provide end to end authentication between source and sink. Source node creates the report of the form $\{S|T|M|sign\}$, where S is the source node ID, T is the destination node ID. For providing En-Route filtering through intermediate nodes, source creates authentication bitmap. Initially, bitmap has m-bits which are all unmarked (set to 0). For the creation of bitmap, source node generates MACs for all the intermediate nodes using $M = MAC(K_{SR}, M)$, where K_{SR} is the key shared between source and the intermediate node. Bitmap is generated using $bitmap \xleftarrow{Bloom} (H, k)$, where H denotes random hash functions and k represents the MACs. This, in turn, will change specific bits of the bitmap to 1, where the number of bits changed and which specific bit is changed purely depends on the different MACs which are fed as input.

Finally, the report is forwarded toward the sink. Each intermediate node checks the report by verifying the EAB included in the report. Each intermediate node checks whether the number of bits marked in the EAB is in the range $\{\lfloor mp - \delta \rfloor, \lceil mp + \delta \rceil\}$, where m is the total length of the bitmap, p is the probability that specific bit will be 1 and δ is a constant. This checking helps to stop the adversary from making all the bits of EAB as 1 to avoid detection. Also, each intermediate node creates $\{EAB^1 = H_i(MAC_{S,R_j}(M))\}$ using the hash function, key shared between source and present node and message. This EAB^1 is compared with original EAB from the report, where bits marked in EAB^1 should also be marked in the original EAB to validate report. If any mismatch happens, the report is dropped. Sink does not need to verify the EAB and it only verifies the $sign$ in the report. Authentication Bitmap is different from message authentication code (MAC) which allows only some specific forwarding

nodes to verify the legitimacy of the report. Moreover, the size of EAB is much less than MAC which saves bandwidth in transmission. With the increase in receivers and more checking in forwarding nodes, detection of false data increases considerably in EAB.

d) *A Bandwidth-Efficient Cooperative Authentication Scheme* **(BEACAN) :** This technique (Lu et al. 2012) is based on co-operative bit-compressed authentication method which provides a bandwidth efficient authentication mechanism. The technique uses Co-operative Neighbor X Router (CNR) based filtering mechanism where all the neighbor nodes co-operatively endorse the report with MACs, so that all the neighbors, intermediate nodes and finally sink can verify the report. The MAC is of the form $\{M = MAC(m, k, n) = (H(m||k)mod2^n)\}$, where m is the message, k is the key used and n is an adjustable parameter. In this technique, all MACs form a matrix of size N x M, where N is neighboring nodes and M are all the forwarding nodes with sink at last. This type of MAC matrix helps in early filtering of false reports. Initially, all nodes share private keys with sink. After deployment, the technique deploys TinyECC based non-interactive key pair (Liu and Ning 2008), so that full bipartite key graph can be made between all the neighboring and forwarding nodes. Using all these keys cluster head creates a MAC matrix which makes filtering more effective.

Discussion- Table 2.6 gives the summary of techniques which uses different methods to assign keys to nodes. The table lists the pros and cons associated with each technique followed by the contributions of each technique. All the above techniques used different approaches to solve the problem of false data detection in WSNs, but none of them can stop DoS attack from happening in the network. Majority of them are prone to selective forwarding and report disruption attacks.

2.2.2 Asymmetric Cryptography based techniques

Asymmetric Cryptography based techniques mainly use signatures in place of MACs for authentication of reports. They mainly use Shamir's threshold cryptography (Shamir

Table 2.6: Pros and Cons of Other En-Route filtering techniques

Technique	Pros	Cons	Contribution
		Others	
IHA (Zhu et al. 2004)	- Key storage overhead is low. - The technique can do association maintenance if any node fails.	- Has a T-threshold limitation and Relies on specific data dissemination route and thus can't be applied in mobile sinks. - Prone to selective forwarding and report disruption attacks.	The technique gave a deterministic En-Route filtering mechanism which utilizes node association to establish authentication keys.
STEF (Kraub et al. 2007)	- As the response are only based on tickets, the technique is resistant from DoS attacks. - Compromised nodes are limited to their vicinity.	- Prone to selective forwarding and report disruption attacks. - Request and reply comes through same route, so in case of route failure the report will be lost.	The technique is based on ticket concept and report are only forwarded only if they contain valid tickets.
EAB (Chen and Lei 2010)	- Reduces communication overhead and have high filtering efficiency.	- Has a precondition that source know the route and all authentication keys of all forwarding nodes. - Prone to selective forwarding and report disruption attacks.	The technique used the bloom filter Bloom (1970) technique for hop-by-hop authentication to create a set of MACs for multiple receivers.
BEACAN (Lu et al. 2012)	- Technique uses MAC matrix thus filtering is more effective - Technique is resistant to selective forwarding attack.	- Technique have high communication overhead. - Technique is prone to report disruption attack. - Technique requires node to know path to the sink.	The technique uses cooperative bit-compressed authentication method which significantly improves filtering capabilities of the technique.

1979) and ECC (Hankerson et al. 2006) for creating and sharing a secret between sensors. Figure 2.3 shows different Asymmetric Cryptography based techniques which are further explained below:

a) *Commutative Cipher Based En-Route filtering* **(CCEF) :** The technique (Yang and Lu 2004) proposed a commutative cipher based En-Route filtering technique. In this technique, each sensor node is assigned a unique ID and a secret key which is shared with the sink. After the deployment, each node sends their location to the sink. Sink uses the unique key and the location of nodes for the authentication of the reports and initiates the query by installing per session security state in sensor nodes. For each session, the sink prepares two keys K_S, K_W such that $\{C(R, K_W) = C^{-1}(R, K_S)\}$, where C and C^{-1} are the encryption and decryption of cipher, K_S is the *session key* and K_W is the *witness key*. The *session key* is used for endorsement of reports, whereas *witness keys* are used to verify the authenticity of the reports. Moreover, $\{C(R, K_S) \neq C^{-1}(R, K_W)\}$

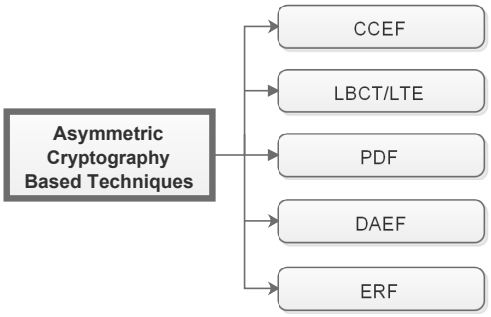


Figure 2.3: Asymmetric Cryptography based techniques

as C and C^{-1} are not commutative. Finally, the sink creates a query of the form $\{QID, CH, (K_S)_{K_{CH}}, K_W\}$, where QID is the query ID, CH is the cluster head ID, $(K_S)_{K_{CH}}$ is session key encrypted using secret key K_{CH} of cluster head. This query is forwarded towards the chosen CH. Each intermediate node stores $\{QID, k_W\}$ for verification of reports.

Cluster head after receiving query packet, decrypts the session key $\{K_S\}$ and broadcasts the query and the report R to neighboring nodes. The participating nodes create the MACs for report, and sends them to the cluster head. The cluster head compresses all the MACs by performing XOR function, i.e. $\{CMAC = MAC(R, K_1) \oplus MAC(R, K_2) \oplus, .., \oplus MAC(R, K_N)\}$, where $K_1, K_2, ..., K_N$ are the secret keys of corresponding neighboring nodes. Additionally, the cluster head also generate MAC using session key K_S, i.e. $\{MAC_{K_S} = (R, K_S)\}$. Cluster head creates the final report which is of form $\{QID|R|CMAC|MAC_{K_S}|\ (ID_1, ID_2, .., ID_N)\}$, where $(ID_1, ID_2, .., ID_N)$ are the IDs of participating nodes. This final report is disseminated toward the sink where each intermediate node verifies the truthfulness of the report by checking the MAC_{K_S} and stored witness keys K_W. This can be done easily by $\{C(MAC_{K_S}, K_W) = C(C(R, K_S), K_W) = C(C(R, K_W), K_S) = C(C^{-1}(R, K_S), K_S) = R\}$. The base station finally checks the report by checking $CMAC$ from the report and acts as a final goalkeeper. Sink also refreshes the session state after every fixed interval of time to handle the compromised nodes.

b) *Location-Based Compromise-Tolerant Security Mechanism* **(LBCT/LTE) :** The technique (Zhang et al. 2006) has used the Cryptographic concept called pairing, and has

proposed *Location Based Keys* (LBKs) by binding private keys with IDs and geographical location of nodes. The authors have also given an LBK based neighborhood authentication mechanism to decrease the effect of compromised nodes in the network.

At the time of deployment, each sensor node is loaded with system parameters and its ID-based key ID_A. After this, localization of the nodes is done, by which each node generates its LBK as $LK_A = KH(ID_A||l_A)$, where K is the master secret key, ID_A is ID-based key for the node and l_A is the location of the node. All the neighbors do a 3-way handshake for neighborhood authentication. These pairwise keys shared between neighbors are called *Immediate Pairwise Keys* (IPKs), i.e. $K_{A,B} = K_{B,A}$. IPKs are used to encrypt and authenticate the messages which are exchanged between neighbors. In addition to IPKs, *Multi-hop Pairwise Keys* (MPKs) are also generated between nodes which are multi-hop away from each other. To generate MPK between two multi hop away nodes, both nodes send a message to each other, i.e. node 1 sends to node 2 $\{ID_1, l_1, n_1H(ID_1||l_1)\}$ and node 2 sends to node 1 $\{ID_2, l_2, n_2H(ID_2||l_2)\}$, where n_1, n_2 are random private numbers. Node 1 calculates $K_{2,1} = \hat{e}\ \{LK_1, n_1H(ID_2||l_2) + n_2H(ID_1||l_1)\}$ and node 2 calculates $K_{1,2} = \hat{e}\ \{LK_2, n_2H(ID_1||l_1) + n_1H(ID_2||l_2)\}$. Using these MPKs, nodes can derive various shared session keys for their use. To filter out bogus reports the technique adopted threshold endorsement method (Wang et al. 2008) where each report has to be endorsed by at least T nodes to be considered authentic. Moreover, each node also attaches its location information to the report which further improves the efficiency of filtering false reports.

c) *Public Key Based Approach* **(PDF) :** The authors have used Shamir's threshold cryptography (Shamir 1979) and ECC (Hankerson et al. 2006) and have devised a public key based false data filtering technique (Wang and Li 2010). In the technique, every report is attached with a digital signature which is signed using a private key of the system. Thus, any forwarding node with system's public key can verify the correctness of report. However, a single sensor node cannot be trusted with the private key. This problem is solved using Shamir's secret sharing, where the secret is distributed to various sensors.

Each sensor maintains a unique share of the secret and any t nodes can collaborate to construct the secret key. The two above techniques (Shamir's secret sharing and ECC) are combined to generate threshold signature, where ECC (Hankerson et al. 2006) is used for signature generation and Shamir's secret sharing technique (Shamir 1979) is used to pass the parameters to nodes secretly which are needed for signature generation.

d) *Authentication and En-Route Data filtering* **(DAEF) :** This technique (Yu and He 2013) uses efficient ID-based signature algorithm to generate signature shares and uses verifiable secret sharing technique to distribute secrets in the network. The technique is similar to PDF which generates ECC-based signatures. After the deployment, the head of each cluster distributes a secret share to all the nodes. Each sensor node maintains a unique share of the secret and any t nodes can collaborate to construct the secret key. In case of an event, the cluster head prepares the report and broadcast the report in the group. If neighboring nodes agree with the report, they all together endorse the report by creating the signature. A compromised node can also be detected if many nodes find and claim that a particular node has sent corrupted partial secret. The final report is broad-casted to sink using multiple streams for multi-path routing. The forwarding nodes verify the reports depending on the probability of the presence of shared secret with that cluster. Finally, the sink can verify the report and act as a final goalkeeper.

e) *An Energy-Aware Routing and Filtering Node Selection in CCEF to Extend Network Lifetime in WSN* **(ERF):** Shahzad and Cho (2017) proposed a CCEF based En-Route filtering technique which uses energy-aware routing and specific filtering node selection. This helps to extend the network lifetime, as the proposed technique balances energy within all the sensor nodes. The technique is almost similar to CCEF with some subtle changes. The first phase in the technique is key dissemination where keys are disseminated to only some intermediate nodes using a probabilistic method. Path creation is the next phase where distance and energy in all sensor nodes are considered to create a path from the base station to the selected cluster head. After the path creation, the base station sends a query message to the chosen cluster head. When the cluster head

receives a query, all the neighboring nodes participate in report generation and generate the endorsements for the report. The cluster head creates the final report which contains all the MACs from the participating nodes and this report is forwarded toward the base station. Intermediate nodes which were chosen in the first phase verifies the MACs using the property of commutative ciphers. Finally, the base station verifies the report.

Discussion- Table 2.7 gives the summary of techniques which uses Asymmetric Cryptography based keys. The table lists the pros and cons associated with each technique followed by the contributions of each. Asymmetric Cryptography based techniques are generally more resilient to adversaries as each node only have part of secret and at least t nodes have to join together to create the signature. But because of it, some Asymmetric Cryptography techniques are prone to T-threshold limitation. Generally, signature generation also increases the communication and processing overhead in the network which is a big problem in energy deficient WSNs.

2.3 Classification based on the probability of filtering

All the En-Route filtering techniques can also be classified on the basis of probability of sharing secret keys. These techniques can be grouped in three sub groups namely, Probabilistic, Deterministic and Hybrid (refer Figure 2.4). In Probabilistic techniques, keys are randomly selected and distributed to senor nodes. In Deterministic techniques, deterministic methods and processes are used to select and assign keys to the sensor nodes. Finally, in Hybrid techniques, both probabilistic and deterministic methods are adopted for key exchange between the sensor nodes. All the subgroups are discussed in following subsections.

2.3.1 Probabilistic Methods

In this group, all the sensor nodes exchange keys with randomly selected sensors in the network with a fixed probability. Thus, each intermediate node has a certain probability of possessing keys which were used to generate the reports. If in any case, the intermediate node does not have keys for verification of the report, it just forwards the report to

Table 2.7: Pros and Cons of Asymmetric Cryptography Based techniques

Technique	Pros	Cons	Contribution
CCEF (Yang and Lu 2004)	- Installs the security state in nodes on demand.	- Relies on special data dissemination protocol. - Has low En-Route filtering efficiency and is prone to selective forwarding and report disruption attacks.	Technique uses commutative cipher to filter out false data and avoid T-threshold.
LBCT/LTE (Zhang et al. 2006)	- Has dissent tolerance against report disruption and selective forwarding technique. - The technique is highly scalable.	- Has a T-threshold limitation. - The bilinear pairing is very expansive. - High communication overhead for 3-way handshake.	Technique has used identity based Cryptography which make the nodes to derive public keys using its public information.
PDF (Wang and Li 2010)	- No requirement of pre shared keys. - Have more security resilience as compared to Symmetric Cryptography key based techniques.	- Has a T-threshold limitation. - Prone to report disruption and selective forwarding attacks.	Technique relies on signature approaches based on Shamir's threshold cryptography and ECC for verification and endorsement of reports.
DAEF (Yu and He 2013)	- No requirement of pre shared keys. - Resilient to selective forwarding and report disruption attacks.	- Takes time to get stable and have high communication overhead. - Very difficult to implement in dynamic networks.	Technique has used ID-based signature algorithm and verifiable secret sharing Cryptography for generation and distribution of the signature.
ERF (Shahzad and Cho 2017)	- Keys are to be disseminated to only few randomly chosen intermediate nodes.	- Relies on special data dissemination protocol. - Is prone to selective forwarding and report disruption attacks.	Technique uses energy-aware routing and specific filtering node selection to extend network life of CCEF based network.

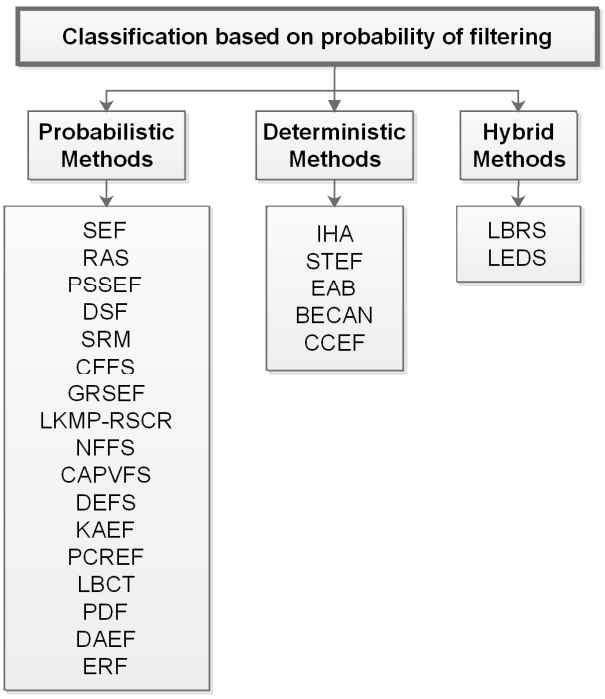


Figure 2.4: Grouping based on the probability of filtering

the next intermediate node without checking. So, in such techniques, all the reports are checked and filtered probabilistically based on the key sharing.

In SEF (Ye et al. 2005a), SRM (Choi et al. 2013) and PSSEF (Sun et al. 2009), intermediate nodes only checks whether the report has minimum N distinct MACs and checks the MAC whose key it has to filter the false reports. CFFS (Liu et al. 2014) also checks whether all the MACs in the report are from the same cluster or not. PDF (Wang and Li 2010) and DAEF (Yu and He 2013) uses basic signature technique to filter the false reports from the networks. In PCREF (Yang et al. 2015), filtering is done based on the check polynomials. LBCT (Zhang et al. 2006) uses location information and MACs in the report to filter the false reports from the network. In GRSEF (Yu and Li 2009) sensor nodes use T-grouping method and the technique divides the terrain into multiple axes to exchange keys which is much better than LBCT (Zhang et al. 2006) for filtering false reports. This method of using location information is more efficient than SEF (Ye et al. 2005a) and PSSEF (Sun et al. 2009). To improve basic filtering, DSF (Sun and Wu 2011) was proposed where each report carries two types of MACs. Thus, each intermediate node checks both types of MAC for authentication. RAS (Hu et al. 2007) is a much-improved technique where intermediate nodes check all endorsements and the authentication tokens to filter any false reports which make the technique more effective than other probabilistic techniques. In NFFS (Wang et al. 2014), each intermediate node verifies the MACs, IDs and location of each participating nodes to authenticate each report which makes the technique more effective against compromised nodes in the network. DEFS (Yu and Guan 2010) and KAEF (Yuan et al. 2008) are hash chain based techniques where the cluster head disseminates the keys to intermediate nodes for verification. These techniques are good against compromised nodes but have high communication overhead. ERF (Shahzad and Cho 2017) is a probabilistic CCEF where the intermediate nodes are chosen probabilistically for reports authentication.

2.3.2 Deterministic Methods

In this group, all the sensor nodes exchange keys with fixed sensor nodes. These sensor nodes form the path to reach the sink. Thus, each intermediate node has a probability of

1 to check and filter the reports. Moreover, each intermediate node does not have keys for all the endorsements in the report. Thus, each intermediate node only verifies a part of the report.

In IHA (Zhu et al. 2004), all the nodes exchange keys based on a HELLO packet traversal in the network. Thus, when a cluster head has to send a report, it has to be sent by the same path of the HELLO packet to reach the sink. So, IHA (Zhu et al. 2004) shows very less resiliency against compromised or faulty nodes in the network. STEF (Kraub et al. 2007) and CCEF (Yang and Lu 2004) are ticket based techniques where sink creates a different ticket for each cluster head and sends those tickets to a particular cluster head. After receiving the ticket, cluster head attaches the report with the ticket through the same route back to sink where the ticket is checked by each intermediate node. EAB (Chen and Lei 2010) on the other hand uses authentication bitmap over multiple paths to sink. But still, the report has to travel same chosen path for being authenticated or else intermediate nodes will not be able to check the authenticity of reports. BECAN (Lu et al. 2012) creates a matrix of MACs to be sent with the report, where matrix include the list of intermediate nodes from origin to sink. Thus, the cluster head fixes the path for a report to reach the sink.

2.3.3 Hybrid Methods

Techniques in this group use both probabilistic and deterministic approaches to intelligently exchange key with few intermediate nodes. Normally such techniques are designed to take advantages of both probabilistic and deterministic approaches. In the majority of such techniques, deterministic approaches are used initially to narrow down the whole network to find the part of the network where a report could travel to reach the sink. Then, probabilistic approaches are applied to this narrowed part of the network to probabilistically exchange keys with fixed intermediate nodes.

LBRS (Yang et al. 2005) and LEDS (Ren et al. 2008) are two hybrid techniques which use both deterministic and probabilistic methods for filtering. Both techniques initially use deterministic methods. LBRS (Yang et al. 2005) finds an arc of upstream cells whose reports a cell can get as an intermediate nodes whereas, in LEDS (Ren

Table 2.8: Pros and Cons of different classifications based on the probability of filtering

Technique	Pros	Cons
Probabilistic	- Reports can choose different path to reach the sink. - Can adapt to dynamic networks.	- Key Storage overhead is high. - Filtering of report is probabilistic. - Scalability is low.
Deterministic	- Key exchange and storage overhead is low. - False reports are dropped much earlier as compared to probabilistic techniques. - Scalability is much better than probabilistic techniques.	- Cannot adapt to dynamic networks. - Reports has to be forwarded through pre-defined paths. - Limited resiliency against compromised or faulty nodes.
Hybrid	- Resiliency against compromised nodes is very high in these techniques. - Reports can travel through multiple paths to reach sink. - Scalability is high.	- Techniques require more time to setup. - Techniques require more time to adapt to dynamic networks.

et al. 2008), each node only finds few adjacent upstream cells whose report it would have to forward. After this, probabilistic methods are adopted where probabilistically some nodes from the upstream region exchange keys with intermediate nodes. Overall this mechanism of using both deterministic and probabilistic methods is very effective against compromised nodes and promises high filtering efficiency. But, such approaches have high communication overhead and takes more time to get stable in dynamic networks.

Discussion- In classification based on the probability of filtering, the majority of techniques are probabilistic. It is just because probabilistic techniques are very easy to setup and reports are not bound to travel through the same path. Deterministic techniques, on the other hand, are more efficient in storage with limited communication overhead. Table 2.8 gives pros and cons associated with all three groups.

2.4 Basic Analysis

The purpose of this chapter is to provide a detailed research review of data En-Route filtering techniques in WSNs. Detailed analysis of different En-Route filtering techniques is provided in this section. First, we discuss all the parameters followed by the analysis of techniques against the defined parameters.

Parameters : Various parameters for the analysis of previously discussed techniques are as follows:

- **Scalability**- This parameter decides how much scalable the technique is after the initial setup of the network and how many changes are needed if the network is to be extended further.
- **Resiliency**- This parameter decides how much resilient the technique is against compromised nodes in the network and how it behaves if the number of compromised nodes increases in the network.
- **Processing overhead**- This parameter is used to get the amount of processing and calculations needed by each node for the generation and authentication of reports.
- **Communication overhead**- This parameter takes into account the message overhead at the time of network setup and endorsement/authentication of reports. This parameter is further divided into two subparts, where we discuss communication overhead at the time of setup and communication overhead after the deployment.
- **Storage overhead**- This parameter is used to tell how much storage is required by sensor nodes for the proper working of the technique.
- **Time to setup**- This parameter tells the initial time requirements for the technique setup.

2.4.1 Symmetric Cryptography based techniques

In this section, we will discuss how the Symmetric Cryptography based En-Route filtering techniques performs against the above discussed parameters.

A) Global key pool partition based En-Route filtering techniques

SEF (Ye et al. 2005a) has no deterministic secret shared between nodes. Thus, the

scalability of the technique is high. The technique is prone to the many attacks, thus resiliency of the technique is poor. The technique has low processing, communication and storage overhead as keys are assigned before the deployment. Moreover, this technique uses bloom filter (Bloom 1970) mechanism which further decreases the communication overhead. But still, the overall power consumption is high. Initial WSN setup time is very less as nodes do not need any message exchange for key sharing.

RAS (Hu et al. 2007) provides a double authentication technique, thus is more resilient to compromised nodes as compared to SEF. RAS maintains a hash function and processes encryption/decryption in each step, thus processing load is more than SEF. After deployment, the communication overhead is high in RAS as a *HELLO* packet is to be sent each time a report is to be forwarded.

PSSEF (Sun et al. 2009) is a SEF based technique where each node evaluates the best path to send the report, thus resiliency is higher than SEF. But, this increases the processing load and communication overhead in PSSEF. Storage overhead in PSSEF is more than SEF as each node has to maintain filtering information for each path.

DSF (Sun and Wu 2011) performs double sharing of keys, one from a global key pool and another using Hello packet, thus the scalability is lower than SEF and RAS. But as DSF uses double key sharing, resiliency is more than SEF and PSSEF. Process overhead in DSF is more than SEF as two types of MACs are to be generated and checked. Communication overhead of DSF is high than SEF and RAS as Hello packet based key sharing includes message exchange from each node. Storage overhead for DSF is much higher than SEF, RAS and PSSEF.

SRM (Choi et al. 2013) is a SEF based technique, thus values of all the parameters are the same as SEF. CFFS (Liu et al. 2014) assign keys to the upstream nodes based on the tree-like path to sink. Thus, scalability is much lesser than SEF, RAS and PSSEF. Communication overhead in CFFS is much higher at time of setup than SEF, RAS, PSSEF and DSF as each node have to participate in path generation to reach the sink.

B) Location based En-Route filtering techniques

LBRS (Yang et al. 2005) is a location-based technique, thus scalability of the technique

is moderate as only a few keys are to be exchanged between chosen associated nodes. The technique is free from T-threshold limitation, but is prone to selective forwarding and report disruption attack. Thus, resiliency is moderate. No much processing is to be done by each node for authentication of reports, thus processing overhead is low. Communication overhead at the time of deployment is moderate as associated keys are to be exchanged between chosen nodes. Communication overhead and storage overhead is low after deployment. Power utilization of this technique is moderate. It requires more time for setup, in which most of the time is used for node localization and in this period many attacks can be done on the network.

LEDS (Ren et al. 2008) is also a location-based technique where each node stores three types of keys, thus scalability is similar to LBRS and resiliency is much better than LBRS. Each node has to store three types of keys, thus storage overhead is more than LBRS.

GRSEF (Yu and Li 2009) is another location-based technique which binds the keys based on the division of terrain along multiple axes. So, the scalability of GRSEF is similar to LEDS and LBRS. Each node has to derive its location-based on multiple axes and on that basis secret keys are assigned, thus processing overhead in GRSEF is more than LEDS and LBRS. Further, multiple axes based key derivation in GRSEF incurs high communication overhead than LEDS and LBRS. Storage overhead is high in GRSEF as each node has to store many parameters and also has to derive some on the go.

In LKMP-RSCR (Fakhrey et al. 2016), at the time of deployment, each node needs to send its neighbor list to the base station, thus scalability of LKMP-RSCR is much lower than LEDS, LBRS and GRSEF. But, as each report is endorsed by neighboring nodes and cell reporters, the resiliency of technique is higher than LEDS, LBRS and GRSEF. The communication overhead is similar to LEDS, but the storage overhead is much lower than LEDS. Time for setup is high as each node needs to send its neighbor list to the base station.

C) Hybrid En-Route filtering techniques

NFFS (Wang et al. 2014) is a hybrid technique which uses both global key pool and geographical location for key assignment and thus scalability is very low, but on the other hand resiliency from compromised nodes is high. There is huge processing and communication overhead during deployment due to the exchange of keys based on the relative position of the nodes. Communication overhead after deployment is low. Each node has to store different types of information, thus storage overhead is moderate. The overall power utilization of technique is low and time for setup is high.

CAPVFS (Nam and Cho 2016) is also a hybrid technique, thus scalability is similar to NFFS. Resiliency for CAPVFS is very high as compared to NFFS because the data is filtered by En-Route filtering and is analyzed by CAA for attack vectors. But, this leads to high process overhead in CAPVFS.

D) Hash based En-Route filtering techniques

DEFS (Yu and Guan 2010) is a hash chain based technique which uses a single seed key to create the hash chain of keys, thus making scalability of the system moderate. The processing load is high, but the communication overhead at the time of setup is low. Each intermediate node also has to communicate with neighboring nodes to know current seed key, thus communication overhead is moderate even after deployment. A lot of storage is also needed to store the hash chain and current seed key for many nodes. But, resiliency for the technique is good. DEFS takes more time to set up as each node has to forward its initial seed to forwarding nodes.

KAEF (Yuan et al. 2008) is also a hash chain based technique. Thus, the scalability, resiliency and process overhead of KAEF is similar to DEFS and communication overhead is more than DEFS at the time of deployment.

E) Polynomial based technique

PCREF (Yang et al. 2015) is a polynomial based technique where each node is provided with authentication polynomial and check polynomial, because of which technique is highly scalable. The processing load is high for the system as each node has to derive its keys from assigned polynomials. Communication overhead at the time of setup is

low which remains the same even after the setup. Storage overhead for the technique is moderate, so is the power utilization. But, the resiliency of the technique is good. Overall time for setup is also moderate.

F) Other En-Route filtering techniques

IHA (Zhu et al. 2004) requires each node to exchange pair wise keys with other nodes, thus scalability of the technique is low. Communication overhead is high at the time of setup. The processing load is moderate as each node has to generate two MAC's. Communication overhead after the setup is also moderate as each report has to carry two types of MAC's. The resiliency of technique is good as it uses two MAC's at the place of one.

STEF (Kraub et al. 2007) is a query-response based technique wherein the nodes only have to exchange keys within the cell. Thus, scalability is moderate. Communication overhead at the time of setup is moderate, and the processing load is low. Communication overhead after deployment is low as responses are only sent in reply for a request. Storage overhead is also low, and power utilization is low. The time required for setup is also low.

EAB (Chen and Lei 2010) is a bloom filter (Bloom 1970) based technique which creates authentication bitmap for En-Route filtering. Thus, scalability of this technique is low, the processing load is high and communication overhead at the time of setup is moderate. Communication overhead after setup is high as a single report is forwarded to multiple receivers. Storage overhead is also high as each node has to store keys of all the forwarding nodes.

BEACAN (Lu et al. 2012) creates a matrix of MACs, thus the scalability is low. Processing load and communication overhead are also high as each report needs to include MAC matrix. Communication overhead at the time of setup is also high as each node will have to exchange keys with all forwarding nodes; this also increases the storage overhead in BECAN. But, the resiliency of the technique is good. Overall power utilization of the technique is low. But, it requires more time for network setup.

2.4.2 Asymmetric Cryptography based techniques

In this section, we will discuss how the Asymmetric Cryptography based En-Route filtering techniques perform against the above discussed parameters.

CCEF (Yang and Lu 2004) is a commutative cipher based technique which only requires secret keys exchanged with sink based on its location, thus scalability is high. But, as there is no interaction between other nodes, the technique has poor resiliency. Processing load and communication overhead after the setup are moderate as two MAC's are generated and checked by intermediate nodes, which moreover uses expansive public-key operations. Storage overhead is low as no extra keys are to be stored by each node. Overall power utilization is high for the technique and time required for setup is low.

LBCT/LTE (Zhang et al. 2006) uses a Cryptographic concept called *paring* where each node has to derive location-based keys with immediate and multi-hop nodes using a 3-way handshake. Thus, scalability is much lower than CCEF. Processing overhead and communication overhead at the time of setup is more than CFFS. But, the resiliency of LBCT/LTE is much better than CCEF.

PDF (Wang and Li 2010) on the other hand uses Shamir's threshold cryptography (Shamir 1979) and ECC (Hankerson et al. 2006) in the technique for derivation and sharing of secrets. Thus, the scalability is lower than CCEF and is similar to LBCT/LTE. Processing overhead and communication overhead at the time of setup is similar to LBCT/LTE. Communication overhead after the setup in PDF is moderate as each report requires combining of a secret for getting the private key.

DAEF (Yu and He 2013) is an improved PDF technique, thus the scalability, processing overhead and communication overhead is similar to PDF. But, the resiliency of DAEF is better than PDF. Communication overhead after the setup is the same as PDF. Storage overhead in DAEF is higher than PDF as it has to store more information as compared to PDF. ERF (Shahzad and Cho 2017) is a CCEF based technique, thus values of all the parameters are the same as CCEF.

2.4.3 Summary

Global key pool based techniques are overall very flexible in term of scalability, but their resiliency is poor, and these cannot handle compromised nodes in the network. To solve the problem, location-based techniques came into existence where keys were assigned based on location, promising high resiliency. But, this increased the overall power utilization of techniques as power is wasted to find the location of nodes (either relative or actual positioning). Hash-based techniques, on the other hand, gave good resiliency against compromised nodes as each node uses a key only once, thus making it very difficult for an adversary to know the keys in use and insert false packets, but these readily changing keys increases the communication overhead and processing overhead for the nodes. PCREF used polynomials in place of MACs which showed another way to solve the problem of filtering of false data, but at higher processing, communication and storage overhead. Asymmetric Cryptography based techniques are generally very difficult to scale up as part of the secret has to be saved in the node at the time of deployment. But, the resiliency of these techniques is much better than Symmetric Cryptography based techniques. The processing load is generally very high due to the creation and checking of certificates and signatures. Communication overhead is also a big problem in these techniques. These techniques are generally energy-hungry solutions which can be difficult to implement over battery-powered sensor nodes. Analysis of all the techniques is given in Table 2.9.

2.5 Security Analysis

In this section, we carry out the security analysis of all the techniques discussed. First, we discuss all the security limitations followed by the security analysis of the techniques.

Security Limitations : Various security limitations of previously discussed techniques are as follows:

- **T-Threshold limitation**- This limitation refers to the case where compromising more then T key pool partitions breaks the security of the network, making it very

Table 2.9: Basic Analysis of En-Route filtering techniques

Technique	Scalability	Resiliency	Process Overhead	Communication Overhead		Storage Overhead	Time for Setup
				At Setup	After Setup		
SEF (Ye et al. 2005a)	High	Poor	Low	Low	Low	Low	Low
RAS (Hu et al. 2007)	High	Good	Moderate	Low	High	Moderate	Low
PSSEF (Sun et al. 2009)	High	Poor	High	High	Low	Moderate	High
DSF (Sun and Wu 2011)	Low	Moderate	Moderate	High	Low	High	High
SRM (Choi et al. 2013)	High	Poor	Low	Low	Low	Low	Low
CFFS (Liu et al. 2014)	Low	Poor	High	High	Low	Low	Moderate
LBRS (Yang et al. 2005)	Moderate	Moderate	Low	Moderate	Low	Low	Moderate
LEDS (Ren et al. 2008)	Moderate	Good	Low	Moderate	Low	Moderate	Moderate
GRSEF (Yu and Li 2009)	Moderate	Good	Moderate	Moderate	Low	High	Moderate
LKMP-RSCR (Fakhrey et al. 2016)	Low	Good	Low	Moderate	Low	Low	High
NFFS (Wang et al. 2014)	Low	Good	High	High	Low	Moderate	High
CAPVFS (Nam and Cho 2016)	Low	Very Good	High	Moderate	Moderate	Low	Moderate
DEFS (Yu and Guan 2010)	Moderate	Good	High	Low	Moderate	High	High
KAEF (Yuan et al. 2008)	Moderate	Good	High	Moderate	Moderate	High	Moderate
PCREF (Yang et al. 2015)	High	Good	High	Low	Low	Moderate	Moderate
IHA (Zhu et al. 2004)	Low	Good	Moderate	High	Moderate	Moderate	Moderate
STEF (Kraub et al. 2007)	Moderate	Moderate	Low	Moderate	Very Low	Low	Low
EAB (Chen and Lei 2010)	Low	Moderate	High	Moderate	High	High	Moderate
BEACAN (Lu et al. 2012)	Low	Good	High	High	High	High	High
CCEF (Yang and Lu 2004)	High	Poor	Moderate	Low	Moderate	Low	Low
LBCT (Zhang et al. 2006)	Low	Good	High	High	Low	Moderate	High
PDF (Wang and Li 2010)	Low	Moderate	High	High	Moderate	Moderate	Moderate
DAEF (Yu and He 2013)	Low	Good	High	High	Moderate	High	Moderate
ERF (Shahzad and Cho 2017)	High	Poor	Moderate	Low	Moderate	Low	Low

easy for the attacker to gain access and inject false data in the network.

- **Key Sharing limitation**- This limitation explains the need for sharing secret keys between multiple nodes and nodes with sink to generate and authenticate reports.
- **Location information limitation**- This limitation refers to the need for GPS capabilities by nodes to locate themselves.
- **Static Network limitation**- This limitation refers to the need for static network and sink for working of the technique. If the network changes, the technique requires reconfiguration.
- **Prone to Selective forwarding**- This attack refers to the situation in which compromised nodes can drop legitimate reports. This attack is also called Path-based DoS (PDoS) attack.
- **Prone to Report Disruption**- This attack refers to disruption of report generation due to submission of corrupted information (Data, MACs, Signatures) from the

compromised nodes. This is also called False-Endorsement based DoS (FEDoS) attack.

2.5.1 Symmetric Cryptography based techniques

In this section, we will discuss how the Symmetric Cryptography based En-Route filtering techniques perform against the above discussed security limitations.

A) Global key pool partition based En-Route filtering techniques

SEF (Ye et al. 2005a) have low filtering probability and thus false reports can travel many intermediate hops before being detected and dropped. It is prone to T-threshold limitation and key-sharing limitation, but it does not require any location information for setup. This technique is not prone to static network limitation and can adapt to change in network and sink. Further, it is independent of data dissemination protocols. SEF is prone to selective forwarding and report disruption attacks.

RAS (Hu et al. 2007) uses double authentication, thus is not prone to T-threshold limitation. PSSEF (Sun et al. 2009) is a SEF based technique, thus all the security parameters have the same values as SEF. DSF (Sun and Wu 2011) uses double key sharing for authentication, thus is not prone to T-threshold limitation. SRM (Choi et al. 2013) is a SEF based technique, thus all the security parameters have the same values as SEF. CFFS (Liu et al. 2014) also have the same values as of SEF.

B) Location based En-Route filtering techniques

LBRS (Yang et al. 2005) uses location-based keys, thus it is free from T-threshold limitation where keys are bound with the location, but is prone to key sharing limitation. Moreover, each node requires location information for setup. The routing protocol uses a beam model, thus it is prone to static network limitation. The technique is also prone to selective forwarding and report disruption attacks.

LEDS (Ren et al. 2008) is not prone to report disruption attack as each report is forwarded through multiple paths. Other parameter values are the same as of LBRS. GRSEF (Yu and Li 2009) have the same parameter values as of LBRS. LKMP-RSCR

(Fakhrey et al. 2016) uses double authentication, thus is not prone to T-threshold limitation. Moreover, the technique is resilient to selective forwarding attack.

C) Hybrid En-Route filtering techniques

NFFS (Wang et al. 2014) uses both global key pool and location-based keys, thus it is free from T-threshold limitation. But, it is having key sharing limitation and requires location information for deployment. Moreover, this technique cannot adapt to dynamic networks and is prone to selective forwarding and report disruption attack. CAPVFS (Nam and Cho 2016) have all the parameter values similar to NFFS.

D) Hash based En-Route filtering techniques

DEFS (Yu and Guan 2010) does not use location binding for the keys and use a single seed key to create a hash chain of keys. Thus, it has T-threshold limitation and key sharing limitation. The technique can adapt to dynamic networks, but is prone to selective forwarding and report disruption attack. KAEF (Yuan et al. 2008) is not prone to T-threshold limitation. All the other parameter values are similar to DFFS.

E) Polynomial based En-Route filtering techniques

PCREF (Yang et al. 2015) uses primitive polynomials, thus do not have T-threshold limitation. But, it still has key sharing limitation. The technique does not require any location information for setup and can adapt to dynamic networks. But, it is prone to selective forwarding and reports disruption attacks.

F) Other En-Route filtering techniques

IHA (Zhu et al. 2004) uses MAC for authentication of reports, but the keys used are assigned without any fixed pattern, thus technique is prone to T-threshold limitation and key sharing limitation. It does not require location information for setup. But, the technique requires the association to be established between other nodes and thus technique cannot adapt to dynamic networks and cannot be used in the case where protocols (Karp and Kung 2000), (Yu et al. 2001) need to be deployed. Moreover, the technique

is also prone to selective forwarding and report disruption attacks.

STEF (Kraub et al. 2007) only share keys with neighboring nodes, thus it does not have T-threshold limitation, but it has key sharing limitation. It also requires location information for setup and is also limited to static networks. The technique is also prone to selective forwarding and reports disruption attacks.

EAB (Chen and Lei 2010) is not prone to T-threshold limitation, but it has key sharing limitation. It does not require location information for setup, but it cannot adapt to dynamic networks. The technique is resilient to selective forwarding attack, but is prone to report disruption attack.

BEACAN (Lu et al. 2012) creates a matrix of MAC with the keys shared with all the nodes. Thus, it does not have T-threshold limitation, but it has key sharing limitation. The technique does not require location information for setup, but it can adapt to dynamic networks. The technique uses multiple paths to send the same report, thus is resilient to selective forwarding attack, but is prone to report disruption attack.

2.5.2 Asymmetric Cryptography based techniques

In this section, we will discuss how the Asymmetric Cryptography based En-Route filtering techniques perform against the above discussed security limitations.

CCEF (Yang and Lu 2004) uses commutative cipher and location information for key generation, thus it does not have T-threshold limitation. But, it has key sharing limitation and it also requires location information for setup. The technique cannot adapt to dynamic networks, but can adapt to the moving sink. The technique is prone to selective forwarding and reports disruption attacks. It can filter only reports by malicious nodes which do not have the session key, but cannot tackle false reports from compromised cluster-heads.

LBCT/LTE (Zhang et al. 2006) uses location-based keys. Thus, it does not has T-threshold limitation but requires location information for setup. It does not have any key sharing limitation and can adapt to dynamic networks, but the time required to get stable is very high. The technique is not prone to selective forwarding and report dis-

ruption attacks. PDF (Wang and Li 2010) have T-threshold limitation, but do not have key sharing limitation. It does not require location information and does not have static network limitation. The technique is prone to selective forwarding and reports disruption attacks. DAEF (Yu and He 2013) uses Shamir's threshold technique (Shamir 1979) and ECC (Hankerson et al. 2006) and a secret random number for endorsement and authentication of the reports. Thus, it does not have T-threshold limitation and it also does not have key sharing limitation. It does not require location information for setup, but it cannot adapt to dynamic networks because of the shared random number within nodes. The technique is not prone to selective forwarding and report disruption attacks.

2.5.3 Summary

Majority of the global key pool partition based techniques have T-threshold limitation where a technique can fail if an adversary can compromise T nodes. This problem is solved by location-based techniques using location-aware key generation. But, the majority of these techniques use node localization and node association which are only practical where the network is static. Moreover, a large amount of energy is wasted in maintaining and checking location awareness in the nodes. Hash-based and polynomial based techniques had key sharing limitation. Majority of the techniques are prone to selective forwarding and report disruption attacks. Asymmetric Cryptography based techniques use signatures for endorsement and these signatures are generated by combining the secret of at least T nodes in the cell. Thus, if not used properly can led to T-threshold limitation. Majority of these do not have key sharing limitation, but the majority of them are limited to static network and cannot adapt to dynamic networks. All the results are tabulated in Table 2.10.

2.6 Key pre-distribution schemes

Key pre-distribution is a method to assign secrets keys to all the sensor nodes at the time of stationing. Thus, using assigned keys sensor nodes can communicate with others securely. Key pre-distribution can be done in many ways including, random pairwise keys, grid-based pre-distribution, group-based pre-distribution, based on combinato-

Table 2.10: Security Limitations in En-Route filtering techniques

Technique	T-threshold limitation	Key Sharing limitation	Location Information Limitation	Static Network Limitation	Selective Forwarding Attack	Report Disruption Attack
SEF (Ye et al. 2005a)	Yes	Yes	No	No	Yes	Yes
RAS (Hu et al. 2007)	No	Yes	No	No	Yes	Yes
PSSEF (Sun et al. 2009)	Yes	Yes	No	No	Yes	Yes
DSF (Sun and Wu 2011)	No	Yes	No	No	Yes	Yes
SRM (Choi et al. 2013)	Yes	Yes	No	No	Yes	Yes
CFFS (Liu et al. 2014)	Yes	Yes	No	Yes	Yes	Yes
LBRS (Yang et al. 2005)	No	Yes	Yes	Yes	Yes	Yes
LEDS (Ren et al. 2008)	No	Yes	Yes	Yes	No	Yes
GRSEF (Yu and Li 2009)	No	Yes	Yes	No	Yes	Yes
LKMP-RSCR (Fakhrey et al. 2016)	No	Yes	Yes	Yes	No	Yes
NFFS (Wang et al. 2014)	No	Yes	Yes	No	Yes	Yes
CAPVFS (Nam and Cho 2016)	No	Yes	Yes	No	Yes	No(Limited)
DEFS (Yu and Guan 2010)	Yes	Yes	No	No	Yes	Yes
KAEF (Yuan et al. 2008)	No	Yes	No	No	Yes	Yes
PCREF (Yang et al. 2015)	No	Yes	No	No	Yes	Yes
IHA (Zhu et al. 2004)	Yes	Yes	No	Yes	Yes	Yes
STEF (Kraub et al. 2007)	No	Yes	Yes	Yes	Yes	Yes
EAB (Chen and Lei 2010)	No	Yes	Yes	Yes	Yes	Yes
BEACAN (Lu et al. 2012)	No	Yes	No	No	No	Yes
CCEF (Yang and Lu 2004)	No	Yes	Yes	Yes	Yes	Yes
LBCT (Zhang et al. 2006)	No	No	Yes	No(Limited)	No	No
PDF (Wang and Li 2010)	Yes	No	No	No	Yes	Yes
DAEF (Yu and He 2013)	No	No	No	Yes	No	No
ERF (Shahzad and Cho 2017)	No	Yes	Yes	Yes	Yes	Yes

rial design and using deployment knowledge. But, due to the scope of the proposed schemes in the next few chapters, we discuss key pre-distribution based on combinatorial design and using deployment knowledge in our study and which is explained in the next subsections.

2.6.1 Combinatorial Design based key pre-distribution schemes

Combinatorial design based key pre-distribution is like a middle ground, where we compromise resiliency of the network for saving storage overhead. Such a design includes assignment of a set of keys to all the sensor nodes in such a way that any given pair of key-sets have some shared keys.

Çamtepe and Yener (2007) were first to use the combinatorial design for key pre-distribution in WSNs. Further, the authors provided a mapping of combinatorial design based key sharing to the sensor networks. This presented a foundation stone for the research in this area. This resulted in many new schemes based on different combina-

torial designs being proposed for key pre-distribution in present times. We now discuss all such schemes briefly:

a) *Camtepe and Yener's Scheme*: As already discussed Çamtepe and Yener (2007) were the first to adopt the combinatorial design for key pre-distribution in WSNs. Authors adopted generalized quadrangles and projective planes for proposing key pre-distribution scheme. According to the adopted method, each cell has $k^2 + k + 1$ sensor nodes, and each sensor node is assigned $k + 1$ keys, where k is a prime. If in case the number of sensor nodes in a cell is n which is not of the form $k^2 + k + 1$ for any prime k, then we choose the smallest prime value k such that $n <= k^2 + k + 1$ and assign keys to n sensor nodes. So first, the number of sensor nodes (n) in a given cell is decided and then the prime value k. If P_i denotes the keys assigned in a cell C_i and all the cells use a different key pool, then $P_i \cap P_{i'} = \emptyset$, for all $i \neq i'$. Thus, nodes compromised in a particular cell do not affect other cells.

Further, Çamtepe and Yener (2007) constructed the symmetric design using mutually orthogonal latin squares. The symmetric design construction starts with the mutually Orthogonal Latin Squares (MOLS) generation, which is used to construct affine planes. These affine planes are used to construct projective planes (Çamtepe and Yener 2007, Section 3.1). The scheme did not provide any algorithm for shared key discovery. The scheme suggested using methods proposed in (Eschenauer and Gligor 2002; Chan et al. 2003) to identify secret key between the sensor nodes. In schemes (Eschenauer and Gligor 2002; Chan et al. 2003) each sensor node broadcasts their key identifiers for shared key discovery. This is followed by a reply from all the sensor nodes to the broadcasting node with their key identifiers, resulting in the identification of common key with each neighboring node.

b) *Lee and Stinson's Scheme* : Formal method of using combinatorial design for key pre-distribution was given by Lee and Stinson (2005). They proposed the concept of common intersection designs (Lee and Stinson 2006) for key pre-distribution. They adopted block graphs for key pre-distribution to sensor nodes in the network, as by the

adopted design each pair of sensor nodes is connected by a maximum two hops. Key pre-distribution in the proposed scheme is based on transversal designs, where all the sensor nodes are indexed by (a, b, c) where $a, b, c \in GF(k)$. By these indexes, keys are assigned to all the sensor nodes. For shared key discovery, sensor nodes broadcast their indexes in the cell which can be used with shared key discovery algorithm to identify the shared key. Any sensor node can use the broadcasted index and its index with a shared key algorithm to identify the shared key. According to the adopted scheme, any pair of sensor nodes share either zero or a single common key.

c) *Chakrabarti-Maitra-Roy's Scheme* : Chakrabarti et al. (2006) proposed to merge the multiple blocks in combinatorial design to propose a novel hybrid key pre-distribution scheme. They adopted the same method of block generation using transversal design as discussed by Lee and Stinson (2005) and they merged them before being assigned to the sensor nodes. Merging of multiple blocks increased key storage overhead in the proposed scheme, but the proposed scheme was more resilient to compromised sensor nodes in the network. Further, merging of blocks also increases the probability of sharing keys within sensor nodes, improving overall connectivity in the network.

d) *Ruj and Roy's Scheme* : Ruj and Roy (2009) proposed a grid-group based key pre-distribution scheme for WSNs. For the proposed scheme authors adopted combinatorial design. For the key assignment in the network, the authors used transversal design. They used a heterogeneous network having two types of sensor nodes, such as ordinary sensor nodes and agents. Within a particular cell, ordinary sensor nodes can communicate directly. For communication across the cells, agents are used. In the proposed scheme, the number of agents in each cell is always three. Key pre-distribution in the proposed scheme is based on transversal designs, where all the sensor nodes are indexed by (a, b, c) where $a, b, c \in GF(k)$. Based on the indexes, keys are assigned to all the sensor nodes and agents, where multiple agents share either one, two or three keys. For shared key discovery, sensor node broadcasts their indexes in the cell. Any other sensor node can use the broadcasted indexes and its index with the shared key algorithm

to identify the shared keys. Key pre-distribution and shared key discovery in the proposed scheme is much similar to the one discussed by Lee and Stinson (2005). Further, authors provide a detailed security analysis of the proposed scheme including resiliency.

e) *Mitra et al.'s Scheme* : Mitra et al. (2012) proposed a key pre-distribution scheme using combinatorial design where sensor nodes are virtually placed at intersection points in a grid. In the virtual grid, keys along the rows are assigned using a projective plane which ensures direct communication between any two sensor nodes in the same row. Further, keys are also assigned along all the columns which ensure secure communication between immediate columns. Thus, the proposed key pre-distribution ensures that all the sensors nodes are connected with a maximum of three hops.

f) *Bag and Roy's Scheme* : Bag and Roy (2013) proposed a combinatorial design based key pre-distribution scheme which adopted Blom's scheme (Blom 1984). The proposed scheme is then mapped to a grid-group deployment of sensor nodes. In the proposed scheme, sensor nodes in the same group can communicate directly with each other and for inter-cell communication special types of nodes (supernodes) are used. Each cell has one super node and scheme assumes supernode can be compromised only if all other nodes in the cell are already compromised. In the proposed scheme, the author adopts a symmetric design for creating blocks for all the sensor nodes in the network. Initially, a public matrix is created. Further, based on a predefined security parameter c and public matrix the private matrices are computed for all the sensor nodes in the network. Now at the time of deployment, each sensor nodes is assigned a row from the public matrix, a row from the private matrix and its position. In need of communication, two sensor nodes can easily communicate using the common key which can be identified using the sensor node's position and public/private matrices.

g) *Bag's Scheme* : Bag (2015) proposed a new key pre-distribution scheme based on a combinatorial design for grid group deployment of sensor networks. The author proposed a heterogeneous scheme, where there are two types of sensor nodes in the net-

work, namely agents and sensor nodes. His scheme had multiple agents in each cell, opposite to a fixed number of agents in scheme Ruj and Roy (2009). This promised better performance than the base technique, but resulted in added over head of an increased number of agents in the network.

2.6.2 Deployment Knowledge based key pre-distribution schemes

Aerial scattering is one of the methods used for deployment of sensor nodes in the region. This method is adopted in areas where physical deployment is not possible, either could be because of difficult terrain or security reasons. For the method, sensor nodes are grouped together and then are dropped sequentially through helicopters/planes. Because of which sensor nodes which are in the same group are likely to be deployed close to each other. In literature, several schemes are proposed which exploit use of deployment knowledge in proposing key pre-distribution schemes (Liu et al. 2005; Du et al. 2006; Huang et al. 2004; Simonova et al. 2006). In such schemes, sensor nodes are deployed in some fixed pattern. Then, this pattern is used for assigning keys to all the sensor nodes in the network. We now discuss all such schemes briefly :

a) *Liu and Ning's Scheme* : Liu and Ning (Liu et al. 2005; Liu and Ning 2005) were one of the first authors who used deployment knowledge for proposing key pre-distribution schemes. Their first scheme (Liu et al. 2005) modified the pairwise key pre-distribution method to propose closest pairwise scheme. For key pre-distribution in the scheme, a setup server establishes keys (pairwise) between all the neighboring nodes. When new nodes are added in the network, the same method is adopted by the setup server to assign keys to the new nodes. Further, a polynomial based key sharing mechanism was presented by Blundo et al. (1992), where nodes used polynomial evaluation to retrieve the pair-wise keys. This method of key assignment was inherited by Liu and Ning (2005) to assign keys to all the sensor nodes in the network. In the proposed scheme, the deployment region is divided into equi-sized cells, where each cell is assigned a bivariate polynomial. At the time of deployment, setup server assigns the polynomials of the home cell and four immediate neighboring cells to each sensor node. So, for

key establishment, polynomials are broadcasted in the network to identify the common polynomials.

b) *Du et al.'s Scheme* : A new method of key pre-distribution was formulated by Blom (1984) which used symmetric matrices. For the scheme, two matrices were maintained, a public matrix and a private matrix. Sensor nodes can use the private matrix's row with the public matrix to identify the shared secret keys. Using multiple key spaces, Du et al. (2005) proposed a multi-space blom scheme. Further, the authors used deployment knowledge to propose an improved scheme (Du et al. 2006). The scheme proposed a grid-group based deployment of sensor nodes, where sensor nodes are deployed in groups. Sensor nodes of each group are deployed on a single deployment point in such a way that probability distribution function for all the sensor nodes is the same.

c) *Huang and Medhi's scheme* : Huang and Medhi (2007) and Huang et al. (2004) adopted multiple space blom filter (Blom 1984) and location knowledge of sensor nodes to introduce a new key pre-distribution scheme. The proposed scheme adopts the same deployment method as given by Du et al. (2006) which is more secure from selective forwarding and random node capture attacks. In the scheme, the deployment region is cleft into regions/zones, where a group of sensor nodes is deployed in zone. Further, according to the adopted scheme, each sensor node is assigned keys from two key spaces, such that in any case no more than a fixed number of sensors select same keys from the key space.

d) *Simonova-Ling-Wang's Scheme* : Simonova et al. (2006) discussed two pre-distribution schemes, one for homogeneous networks and other for heterogeneous networks. In both the schemes, the deployment region is broken into grids and sensor nodes are deployed in each grid, as done in Du et al. (2006). In both the schemes, two key pools are maintained namely, deployment key pool and original key pool. Each grid maintains a unique original key pool, where original key pools for different grids are disjoint.

Sensor nodes in each grid are assigned keys based on the original key pool. Then, the multiple grids are grouped to create bigger cells to form a deployment key pool. For key pre-distribution authors stated that any scheme could be adopted, but their scheme used the same transversal design as adopted by Lee and Stinson (2005) for assignment of keys to sensor nodes.

2.7 Unresolved Problems / Research Directions

The existing surveyed techniques have evolved from time to time, but have not raised the bar for attackers. Some techniques showed promising results, but they inherited some limitations which are still unresolved. In this section, a brief description of open research problems in En-Route filtering is given. Various open research problems in the area of En-Route filtering are:

- **Selective forwarding Attack**- Majority of the techniques which we have discussed so far are not able to mitigate selective forwarding attack from the WSN's. Moreover, the techniques which are resistant to selective forwarding attack sends the same reports from multiple paths. This increases the communication overhead in the network. Some techniques have shown the direction of using Watchdog (Marti et al. 2000) to prevent selective forwarding attack in WSNs, which help to identify misbehaving nodes in the network. But, the watchdog is also limited in working and have many limitations.

- **Report Disruption Attack**- No efficient techniques have been proposed for the mitigation of report disruption attack in WSNs. Majority of the solutions include checking of data or MACs provided by each participating node which can be spoofed easily by the adversary. Use of watchdog (Marti et al. 2000) was also encouraged by many techniques to avoid report disruption attack where neighboring nodes can decide whether any node is misbehaving or not. But, implementation and effectiveness of watchdog are limited which further increases the communication overhead and radio on overhead in the sensor networks.

- **Scalability**- Majority of the techniques are not scalable after deployment, as the

majority of authentication done in the networks relies on the key exchanged at the time of deployment with other neighboring and forwarding nodes. But, if we try to add the nodes after the deployment, it is very difficult to ensure keys are exchanged safely with other nodes.

- **Communication Overhead**- Majority of the techniques are having moderate to high communication overhead which is a big problem in resource constraint WSNs. Main reasons for communication overhead is key exchange or node localization or sending of bulky reports in the network.

- **Static Network Limitation**- Majority of the techniques cannot adapt to dynamic networks. The techniques fail if the network or sink is dynamic. Moreover, none of the techniques have discussed the presence of multiple sinks in the network. Moreover, the techniques which can adapt to the dynamic network requires huge time for node localization and path reconstruction to the sink.

- **Identification and Removal of compromised nodes in network**- Majority of the networks only filter out false data. But, no efficient techniques are available where compromised nodes can be identified and can be quarantined or removed from the network.

The majority of existing en-route filtering schemes are prone to selective forwarding and report disruption attacks. In recent times, some asymmetric techniques have also been proposed which do not require any pre-shared keys in the network and which provides high detection rate and good resiliency against compromised nodes. But still, the implementation of asymmetric techniques is questionable on WSNs where sensor nodes have limited computation power. Majority of symmetric techniques require the exchange of keys which results in high communication overhead. The main goal of this research work is to invent the features to cope with different limitations of existing En-Route filtering techniques. This research work mainly focuses on proposing new key pre-distribution schemes and then to extend the proposed key pre-distribution schemes to propose new En-Route filtering schemes. We propose new en-route filtering schemes which ensure excellent filtering efficiency and low associated key storage overhead.

Further, this work focuses on proposing new data authentication/verification methods to ensure high resiliency against report disruption attacks and selective forwarding attacks. Finally, reducing the communication overhead associated with key exchange and en-route filtering is also one of the main objectives of this research work.

2.8 Concluding Remarks

In this chapter, we provided an in-depth survey of En-Route filtering and key pre-distribution techniques in WSNs. We discussed the research done in the last decade, with major boom and directions taken by various researchers. We discussed many En-Route filtering techniques, explained their general working and basic architecture, followed by comparative analysis of various design choices. This led to the discussion of open and unresolved issues in En-Route filtering in WSNs, which so far have received much less attention from the majority of researchers. Further, we also discussed combinatorial design based key pre-distribution schemes and other schemes which uses deployment knowledge for key pre-distribution in the network.

CHAPTER 3

Combinatorial Design Based Key Pre-Distribution

In this chapter, we present three novel key pre-distribution schemes based on combinatorial design. Initially, we propose a combinatorial design based key pre-distribution scheme (CD-KPD). CD-KPD assumes a network which is further divided into cells of equal size, as proposed by Liu et al. (2005). Each cell has normal sensor nodes and Cluster Heads (CHs). Each cell uses different key pools for key assignment to ensure direct intra-cell communication in the network. For inter-cell communication CHs are used. For keys assignment, we used combinatorial design, which is used to obtain key-sets by choosing keys from a given key pool. We also propose a new shared key discovery algorithm which provides a safe and efficient way to find out shared keys. In each cell, we have three CHs, as proposed by Ruj and Roy (2009) and three CHs are enough to ensure needed security in the network.

We modify CD-KPD to propose a combinatorial design based reduced key pre-distribution (CD-RKPD) where cells communicate only within given *Lee sphere* region (Blackburn et al. 2008). Because of which sensor nodes of a particular cell can now only communicate with sensor nodes present in other cells which are within *Lee sphere* region of that particular cell. For this, we remove all the extra keys stored in each CH. This helps in reduction of overall keys storage overhead and the total keys exposed when a particular CH is compromised.

We further modify CD-KPD to propose a combinatorial design based partial key

pre-distribution (CD-PKPD) where key sets are only assigned to $3/4^{th}$ of the CHs of each type. Specifically, to reduce the key-storage overhead in CD-KPD, we reduce the number of links maintained between cells. This can be achieved by assigning key-sets to only limited CHs of each type. Selection of CHs of each type for key-set assignment is done in such a way that at-least one link is maintained between all the cells. Thus, maintaining desired connectivity in the network.

We analyzed CD-KPD, CD-RKPD and CD-PKPD against node compromise. We considered the resiliency in terms of links broken and cells disconnected when certain nodes and CHs are compromised in the network. We were able to achieve high resiliency against schemes Simonova et al. (2006), Huang et al. (2004), Ruj and Roy (2009), Bag (2015) and Mitra et al. (2012).

Structure of the remaining chapter is as follows: Section 3.1 provides the preliminaries and notations used in this chapter. In Section 3.2, we present our key pre-distribution schemes. Section 3.3 and 3.4 provides an in-depth analysis of our schemes. Finally, the concluding remarks are given in Section 3.5. This chapter is based on the articles Kumar and Pais (2018c) and Kumar et al. (2019).

3.1 Preliminaries

3.1.1 Combinatorial Design

A set system (Anderson 1990) is a 2-tuple *(X,A)*, where *X* is a cluster of elements and *A* is set of subsets of *X*. This set of subsets is also known as *blocks*. A Balanced Incomplete Block Design (BIBD) is represented by *(v,b,r,k,λ)*, where *v* is total number of elements in *X* and *b* is total number of blocks. Such design satisfies following properties-

- Each element of *X* occurs in *r* blocks,
- All blocks have exactly *k* elements,
- Each pair of element of *X* is present in exactly λ blocks.

A BIBD is called symmetric design or Symmetric BIBD when *v*=*b*. It can also be shown that in a Symmetric BIBD *k*=*r* (Anderson 1990).

A difference set $(v, k, \lambda)(mod\ v)$ is a set $D = \{d_1, d_2, .., d_k\}$, where d_k represents distinct elements of Z_v, such that each element d, where $d \neq 0$ can be expressed in the form $d = d_i - d_j(mod\ v)$ in exactly λ ways (Anderson 1990). Then the blocks for symmetric design (v, k, λ) can be easily obtained by $D, D+1, D+2, D+3, ..., D+(v-1)(mod\ v)$ (Anderson 1990). For example, to generate $(7, 3, 1)$ Symmetric design, difference set {1,2,4} can be used. All the resulting blocks will be : {1,2,4},{2,3,5}, {3,4,6}, {4,5,7}, {5,6,1}, {6,7,2}, {7,1,3}.

A multiplier (q) (Liu and Ning 2005) of a given difference set (D) for (v, k, λ) in an Abelian group $(G, +)$ satisfies following properties-

- (q) is a prime number such that $gcd(q, v) = 1$,
- $q > \lambda$ such that $k - \lambda \equiv 0(mod\ q)$.

3.1.2 Bloom Filter

Bloom filter (Bloom 1970) is a popular data structure used to verify membership i.e. chosen an element, Bloom filter can identify the presence of that element in a predefined set. Bloom filter requires a set $T = \{t_1, t_2, t_3..., t_x\}$, a string of size f-bits and s independent hash functions $(H_1, H_2, ...H_s)$. Each item (t_i) with a hash function (H_i) creates a hash value and maps it in the range $\{0, 1, 2....f-1\}$ uniformly, where $\{0, 1, 2....f-1\}$ represents bits in a f-bit string. Every bit of the f-bit string is preset to 0 at the initialization stage. $\forall t_i \in T$, hash values are generated using all hash functions (H_i) to set the corresponding values as 1 in the f-bit string. If x and s represents the size of set T and total hash functions respectively, then finally total xs bits are set in the f-bit string (Figure 3.1).

To verify membership of a particular element t' in the given set T, hash values are generated using all the hash functions. These hash values are compared with the exact values in the f-bit string. If all the values are already marked 1, t' is considered to belong to T and even if one of the values is 0 then item t' is definitely not in T.

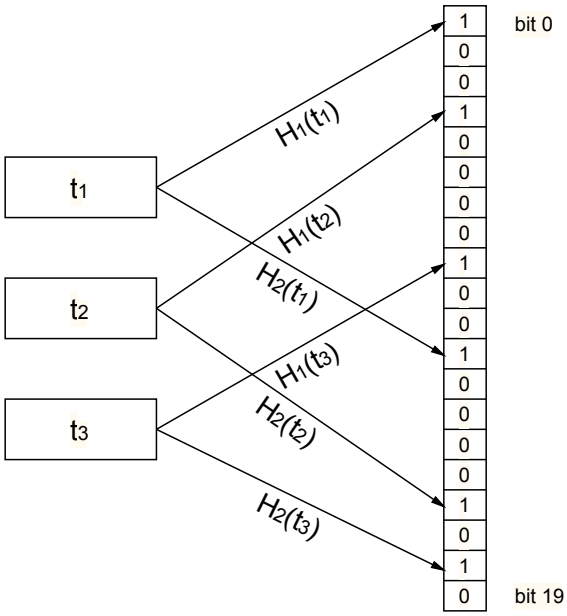


Figure 3.1: Bloom Filter which have 3 sets, using a string of 20 bits and total hash functions are 2

3.1.3 Lee Sphere Region

If we consider a network which is cleft into cells (Figure 3.2), a *Lee Sphere* (Blackburn et al. 2008) for a chosen Lee distance (ρ), comprise of all the neighboring cells that are at-most ρ distance from a chosen cell. To calculate the distance between any two cells the sum of horizontal and vertical distance (*Manhattan distance* (Black 2006)) can be calculated. Figure 3.2 shows *Lee sphere* region (highlighted region) of a chosen cell.

Table 3.1 represents the notations used in this chapter.

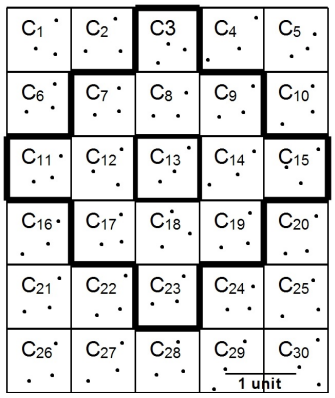


Figure 3.2: Deployment of 30 cells. Each cell has three cluster heads represented as dots in each cell. Lee distance is 2. Highlighted region shows cells which are within *Lee sphere* of C_{13}

Table 3.1: Notations

P	Total sensor nodes in the network.
C_i	i^{th} cell in the network.
N	Total cells in network.
n	Nodes in a particular cell.
$k+1$	Keys assigned to each node.
P_i	Key-set assigned to sensor nodes in a cell.
ρ	*Lee Sphere* region.
CH_{ix}	x^{th} cluster head in i^{th} cell.
$e+1$	Total keys assigned to cluster heads in CD-KPD.
f(Min.)	Minimum number of keys stored by any cluster head in CD-RKPD.
f(Max.)	Maximum number of keys stored by any cluster head in CD-RKPD.
f(Avg.)	Average number of keys stored by cluster heads in CD-RKPD.
$k'+1$	Total keys assigned to cluster heads in CD-PKPD.
$Ll(K)$	Fraction of intralinks broken when K sensor nodes are compromised.
$Lg(S)$	Fraction of interlinks broken in CD-KPD when S cluster heads are compromised.
$Lg'(S)$	Fraction of interlinks broken in CD-RKPD when S cluster heads are compromised.
$L_g(S)$	Fraction of interlinks broken in CD-PKPD when S cluster heads are compromised.
$Dl(K)$	Fraction of nodes disconnected when K sensor nodes are compromised.
$Dg(S)$	Fraction of cells disconnected in CD-KPD when S cluster heads are compromised.
$Dg'(S)$	Fraction of cells disconnected in CD-RKPD when S cluster heads are compromised.
$D_g(S)$	Fraction of cells disconnected in CD-PKPD when S cluster heads are compromised.
$GF(k)$	Galois field of k elements.

3.2 Key pre-distribution schemes (CD-KPD, CD-RKPD and CD-PKPD)

In our work, a distributed sensor network is assumed which has P sensor nodes. We cleft the network into the same sized cells, similar to scheme Liu and Ning (2005). All the sensors are consistently distributed in the network. Sensor nodes directly communicate with each other within the cells and to secure this communication sensor nodes require shared secret keys. For inter-cell communication, a special type of sensor nodes are used, known as *Cluster Heads* (CHs).

In the proposed scheme CD-KPD, inter-cell communication is possible between any two cells, providing connectivity of 1 in the network. We further modify CD-KPD to propose CD-RKPD where CHs only communicate within the given radio frequency range. To consider the communication range of a CH in CD-RKPD, we use $Lee\ sphere$ region, where a particular CH only communicates with CHs which are within its $Lee\ sphere$ region. Thus, all the CHs in a given $Lee\ sphere$ region must have pair-wise keys to have a secure communication. In the proposed schemes (CD-KPD, CD-RKPD and CD-PKPD), each cell has three CHs ensure secure inter-cell communi-

cation. Further, sensor nodes can use any particular CH for inter-cell communication, providing much needed fault tolerance against failures and attacks.

3.2.1 Key pre-distribution in a cell

Key pre-distribution in a cell is the same for all the three proposed schemes CD-KPD, CD-RKPD and CD-PKPD. In our schemes, each cell uses a different key pool for key assignment and each sensor node is assigned a key-set of fixed size at the time of deployment. Multiple key pre-distribution schemes have been proposed in literature like, deterministic (Lee and Stinson 2005; Ruj and Roy 2009), randomized (Eschenauer and Gligor 2002; Chan et al. 2003), and hybrid (Chakrabarti et al. 2006). We use a deterministic scheme which ensures that each sensor nodes has a common key shared with other sensor nodes in a given cell, to ensure direct communication. This can be done by creating key-sets (blocks) [Section 3.1.1] and then assigning the key-sets to all the sensor nodes. The common keys between the key-sets are used to secure the communication between the sensor nodes. Probabilistic and hybrid schemes cannot guarantee this direct communication. In the proposed scheme, each cell has $k^2 + k + 1$ sensor nodes and each sensor node is assigned $k + 1$ keys, where k is a prime. If, the number of sensor nodes in a cell is n which is not of the form $k^2 + k + 1$ for any prime k, then we choose the smallest prime value k such that $n <= k^2 + k + 1$ and assign keys to n sensor nodes. So first, the number of sensor nodes (n) in a given cell is decided and then the prime value k. If all the keys assigned in a cell C_i is denoted by P_i and all the cells use different key pools for each cell. So $P_i \cap P_{i'} = \emptyset$, for all $i \neq i'$. Thus, nodes compromised in a particular cell do not affect other cells.

We in our scheme use *Difference Method* or *Difference Sets* (Anderson 1990) to construct the Symmetric design [Section 3.1.1]. Construction algorithm for blocks generation is given in Algorithm 3.1. First step of algorithm is to find a multiplier (Stinson 2007) of given difference set (D) for (v, k, λ) in an $Abelian\ group$ $(G, +)$. This multiplier can be used to find a translate of D. The multiplier is used to find the orbits of Z_v, where Z_v represents $Abelian\ group$ $(Z_v, +)$ for a given (v, k, λ). Orbits are the cycle decomposition of Z_v where orbits are subsets of elements that form cycles. Multiple or-

bits can be used in the union to get the desired difference set of fixed size. Next step is to find all the blocks using the difference set. After obtaining all the blocks, each block can be randomly assigned to sensor nodes in a cell. In the algorithm, steps 1-3 takes $O(k^2) = O(n)$ time. Steps 4-9 are used to generate all the blocks using the difference set derived in previous steps. Steps 4-9 takes $O(k^3) = O(n^{1.5})$ time.

Algorithm 3.1: Blocks generation using symmetric design

Input: Symmetric design (v, k, λ) where $\lambda = 1$
Output: $k^2 + k + 1$ blocks of keys, each block has $k + 1$ keys and any two blocks have one shared key

1 Find Multiplier (a) for difference set.
2 Compute all the orbits by mapping $x \longmapsto ax \; mod \; v$.
3 Find $difference \; set \; \{d_1, d_2, ..., d_{k+1}\}$ of $(k + 1)$ length using the orbits.
4 **for** $j \leftarrow 1 \; to \; (k^2 + k + 1)$ **do**
5 $\quad Block_j = \{d_1, d_2, ..., d_{k+1}\}$
6 $\quad$ **for** $i \leftarrow 1 \; to \; (k + 1)$ **do**
7 $\quad\quad \{d_i = (d_i + 1) mod (k^2 + k + 1)\}$
8 $\quad$ **end**
9 $\quad$ end for
10 **end**
11 end for

3.2.2 Shared key discovery in a cell

Here in our scheme for shared key discovery, we use Bloom filter (Bloom 1970). The Bloom filter is previously explained in Section 3.1.2. For our scheme, set T represents the $k + 1$ keys assigned to each sensor node. Further, each node at the time of deployment is assigned a fixed number of hash functions. Each node uses these hash functions over the $k + 1$ keys to set particular bits in f-bit strings. Construction algorithm is given in Algorithm 3.2.

This f-bit string is then broadcasted to all the nodes in the cell and each node can use its key-set with hash functions to find the common key. In the whole set of keys, a common key will have all the corresponding hash values set to 1 in f-bit string. Algorithm 3.3 discusses the share key discovery phase. The verification of the hash function for shared key discovery can be done in $O(k) = O(\sqrt{n})$ time. The only information needed for shared key discovery is broadcasted f-bit string by all the nodes. Thus,

Algorithm 3.2: Creation of f-bit string for shared key discovery

Input: $(k + 1)$ keys stored in each sensor node denoted by k_j
f-bit string with all bits set to 0.
Output: f-bit string with $(k + 1)s$ bits set to 1

```
Each node have independent hash functions (H_1, H_2, ....H_s).
for j ← 1 to (k + 1) do
    for i ← 1 to s do
        x = H_i(k_j)
        f-bit string(x)=1
    end
    end for
end
end for
```

communication overhead is $O(f)$ bits which is much better than schemes Eschenauer and Gligor (2002), Chan et al. (2003). But over head of the proposed scheme is much more than schemes Ruj and Roy (2009), Bag (2015).

For key pre-distribution in schemes Ruj and Roy (2009), Bag (2015), all the nodes in a cell are indexed by (a, b, c) where $a, b, c \in GF(k)$. Based on these identifiers keys are assigned to all the sensor nodes. At the time of shared keys discovery, sensor nodes broadcast their identifiers which are used with share key discovery algorithm to identify the shared key. Any particular node can use its identifier and broadcasted identifier with a shared key algorithm to identify the shared key. Since all the identifiers are broadcasted in the network, the adversary can obtain all the identifiers of the sensor nodes. The adversary can easily use any two identifiers with the shared key algorithm to identify the shared key between two sensor nodes. But in our proposed CD-KPD scheme, for shared key identification, we use Bloom filter (Bloom 1970). Thus, even if the adversary obtains all the f-bit strings broadcasted in the cell, he cannot obtain the shared keys. So, even if our proposed shared key discovery phase has much more overhead than other schemes, it provides more secure shared key discovery between any two sensor nodes.

3.2.3 Key pre-distribution in cluster heads for CD-KPD

In the proposed scheme CD-KPD, each cell is denoted by C_i. Each cell has three CHs denoted by CH_{ix} where i represents cell number and x represents three CHs

Algorithm 3.3: Identification of shared key using f-bit string

Input: Broadcasted f-bit string
Output: Shared key K_j, if any shared key exists

```
1 Each node have independent hash functions (H_1, H_2, ....H_s).
2 for j ← 1 to (k + 1) do
3     for i ← 1 to s do
4         x_i = H_i(k_j)
5     end
6     end for
        if ∀x : f-bit string(x) == 1 then
            Shared key is k_j
        else
            It is not the shared key
        end if
7 end
8 end for
```

i.e. $CH_{i1}, CH_{i2}, CH_{i3}$. To facilitate secure communication, each CH requires pair-wise keys shared within its cell so that sensor nodes in the same cell can communicate with CHs. This is looked after by *key pre-distribution in cell* phase. In addition to these keys, each CH also needs pair-wise keys with CHs in other neighboring cells for inter-communication between cells. We have three types of key-sets which are derived using different key pools i.e. $Type$ 1, $Type$ 2, $Type$ 3 keys. Further, all three types of key-sets are equally assigned to the CHs. Each CH in a particular cell is assigned key-set of a different type. If total cells in the network are N, then there are $3N$ CHs in total and precisely N CHs will have $Type$ 1, N CHs will have $Type$ 2 and N CHs will have $Type$ 3 keys. For assignment of a particular type of keys to N CHs, we choose e such that $N <= e^2 + e + 1$ where e is prime and assign $e + 1$ keys to all the CHs of a particular type. Using Algorithm 3.1 we can generate all the $e^2 + e + 1$ key sets each having $e + 1$ keys and each pair of key-set have one key in common. These sets are then randomly assigned to a particular type of CHs in the whole network. This process of key distribution is repeated three times to assign keys to three different types of CHs from three different key pools. Because of which any cell can communicate with any other cell in the network, providing connectivity of 1 in the whole network. Any two cells share exactly three keys and each of the shared keys belong to a different type.

In the proposed CD-KPD scheme, keys are shared in the whole network and thus

CH compromised in any part of the network affects the whole remaining network. In practical scenarios, a particular cell communicates with only a few other cells in the network. To improve the performance of CD-KPD to work with limited distances some subtle changes are needed.

3.2.4 Shared key discovery in cluster heads

Shared key discovery in CHs is same as discussed in Section 3.2.2. The method of shared key discovery using Bloom filter ensures needed security in the network.

3.2.5 Combinatorial design based reduced key pre-distribution scheme (CD-RKPD)

To reduce the key storage overhead in CD-KPD some changes were needed. Normally, radio capabilities of CHs are limited and thus each CH can only communicate with limited other CHs. These few CHs normally represent a *Lee sphere* region (Section 3.1.3) around a particular CH which are within its communication range. So, a particular CH only needs secret keys shared with CHs of the same type which are within its *Lee sphere* region. This can be achieved easily by minor changes in CD-KPD where common keys were spread across the network to propose a new scheme called CD-RKPD. In CD-KPD, each CH has $e + 1$ keys and has shared keys with all other CHs of the same type in the whole network. In CD-RKPD, firstly each CH creates its *Lee sphere* region ρ and figures out all the CHs which have the same type of keys and are within its *Lee sphere* region. In the shared key discovery phase, each cell only accepts f-bit string from the cells which are within its *Lee sphere* region and identify the shared keys. After identification of all the shared keys, each CH will be knowing set of keys (f) which will be used for communication with CHs within its *Lee sphere* region and set of keys $q = ((e+1)-f)$ which will never be used. These never used keys (q) in a particular CH are the keys that are shared with CHs of the same type which are not in its *Lee sphere* region and these keys will never be required by a particular CH. So these never used keys (q) can be removed from each CH. Finally, each CH is left with only those keys (f) which are shared with CHs of the same type in its *Lee sphere* region. These changes in the scheme helped to further decrease the key storage over-

head in CHs and also helped to improve the resiliency against compromised CHs in the network.

3.2.6 Combinatorial design based partial key pre-distribution scheme (CD-PKPD)

Another way to reduce the key storage overhead in CD-KPD can be assigning keys in the partial network. In CD-KPD, each cell has three CHs of three different types namely, $Type1, Type2$ and $Type3$. We use different key pools to derive key-sets for each type of CHs. This algorithm is repeated three times to assign key-sets to all the CHs of different types. If all the CHs in the network are assigned key-sets then according to CD-KPD, each CH will store $e + 1$ keys where $N \leq e^2 + e + 1$ and each cell maintain three links with all other cells in the network. To further reduce the key-storage overhead in CD-KPD, we can reduce the number of links maintained between the cells. This can be achieved by assigning key-sets to only limited CHs of each type. Selection of CHs of each type for key-set assignment is done in such a way that at-least one link is maintained between all the cells, thus maintaining desired connectivity in the network.

To ensure that at-least one link exists between every pair of cells we assign key-sets to exactly $3/4^{th}$ of total CHs of each type. Selection of $3/4^{th}$ CHs of each type is done as given in Table 3.2. We create a random list L_1, which represents cell ids of all the cells in the network. Using this random list we generate three sets namely, set 1, set 2 and set 3 each of size $3/4^{th}$ of total CHs. Now, these sets can be individually chosen by each type of CHs to assign key-sets to CHs. Cell ids in each set represents CH of particular cells which will be assigned key-sets. Thus, CHs which are assigned key-sets, can only communicate securely with each other.

Table 3.2: Generation of sets

$L_1 = \{a_1, \cdots, a_N\}$
$set\ 1 = \{a_1, \cdots, a_{(3/4)N}\}$
$set\ 2 = \{a_{(3/4+1)N}, \cdots, a_N, a_1, \cdots, a_{(1/2)N}\}$
$set\ 3 = \{a_{(1/2+1)N}, \cdots, a_N, a_1, \cdots, a_{(1/4)N}\}$

Experimentally, we found that limiting communication between $3/4^{th}$ of total CHs

reduces the number of links maintained between cells in the network. But still, at-least 85% of cells have either 2 or 3 links in the network. But if we try to further reduce the number of CHs from $3/4^{th}$ of total CHs, then connectivity in the network cannot be guaranteed in the network, which can disconnect the cells from the network.

Now, key-sets are generated only for $3/4^{th}$ CHs using Algorithm 3.1, such that $k'^2 + k' + 1 \geq (3/4)N$, where k' is smallest prime number. These key-sets of size $k' + 1$ are assigned to CHs of a particular type whose ids are in that corresponding set. This eventually helps in reducing key storage overhead in CHs while maintaining desired connectivity in the network.

3.3 Analysis of CD-KPD and CD-RKPD

When sensor nodes are compromised, an adversary can get hold of all the keys stored in them. Thus, these compromised keys cannot be used further. Our proposed schemes have multiple nodes sharing the same key. So, all the links which were communicating using the compromised keys will now be unsafe for communication. In such circumstances, communication needs to be carried through alternative paths. In some cases, a particular node can also be disconnected from the network, where all its keys are compromised and thus it will not be able to communicate safely. In this section, we find out the resiliency of both proposed schemes. The over all resiliency of the network can be found using two parameters, first is links disrupted when some sensor nodes are compromised in the network and second is node disconnection from the network. These two parameters can be mathematically expressed using formula's-

$$L(p) = \frac{Links\ broken\ when\ p\ nodes\ are\ compromised}{Total\ number\ of\ links\ in\ network} \tag{3.1}$$

$$D(p) = \frac{Nodes\ disconnected\ when\ p\ nodes\ are\ compromised}{Total\ number\ of\ nodes\ in\ network} \tag{3.2}$$

Both these parameters will be discussed in detail in next subsections. But firstly, we will discuss the false positive for the Bloom Filter (Section 3.1.2).

The proposed scheme is evaluated in a custom built simulator in Python language. The simulation includes deployment and assignment of combinatorial design based se-

cret keys to sensor nodes/CHs in the network. For simulating compromised sensor nodes/CHs, we randomly select fixed number of sensor nodes/CHs and we assume that all the information stored in these nodes is obtained by an adversary. For obtaining simulation results, we analyze the percentage of links affected/nodes disconnected when a fixed number of sensor nodes/CHs are compromised in the network. All the experiments, like choosing a fixed number of compromised sensor nodes/CHs in the network is repeated 50 times. This is mainly done to remove any ambiguity in the results.

3.3.1 False Positive for Bloom Filter

In rare cases more than one hash functions over given set of elements can map to same bit in the given f-bit string. So false positive in Bloom filter occurs when an particular element t' is not in set T, but its hash values are set in given f-bit string. If hash functions used for the bloom filter gives uniformly random values, probability P of particular bit in string to be 0 can be found using Equation 3.3. In the equation, xs represents total bits set to 1 in the f-bit string.

$$P = (1 - (\frac{1}{f}))^{xs} \tag{3.3}$$

Thus probability for false positive can be derived using Equation 3.4. This false positive is negligible and can be skipped if considerably big f-bit string is used as input for bloom filter.

$$P = \{1 - \{1 - (\frac{1}{f})^{xs}\}\}^{s} \tag{3.4}$$

3.3.2 Estimation of resiliency $(L(p))$

In proposed schemes, both normal sensor nodes and CHs are prone to be compromised. Normal sensor nodes share keys only within cells. Thus, if K normal nodes are compromised, only links inside cells will be affected which can be termed as *Local Resiliency (Ll(K))*. For communication outside the cell, we use CHs which share keys with other CHs. Thus, if S CHs are compromised, links connecting the cells will be broken which can be termed as $Global\ Resiliency(Lg(S))$.

a) Local Resiliency ($Ll(K)$)

As both our proposed schemes (CD-KPD and CD-RKPD) uses the same method to assign keys within the cell, local resiliency for both the schemes will be the same. The proposed pre-distribution scheme ensures that each pair for key-sets have one common secret key and to fulfill this each key is assigned to exactly $k+1$ sensor nodes. So, if any key K' is compromised by an adversary, then $k(k+1)/2$ links are disrupted. Each sensor node has $k+1$ keys and if a particular sensor node is compromised, then all its keys are compromised and total links disrupted can be given by $k(k+1)^2/2$. Thus finally, if K sensor nodes are compromised in the whole network, a total of $K\{k(k+1)^2/2\}$ links are broken. This is the maximum number for the links disrupted in the network because multiple nodes compromised from a particular cell will be sharing some keys and thus total individual keys compromised will be less and so will be the links broken between sensor nodes. Finally, local resiliency of the network can be expressed using Equation 3.5,

$$Ll(K) = \frac{K\{k(k+1)^2/2\}}{N(k^2+k+1)(k^2+k)/2} \tag{3.5}$$

where $N(k^2+k+1)$ represents total sensor nodes in the network. The formula can further be simplified as Equation 3.6.

$$Ll(K) = \frac{K(k+1)}{N(k^2+k+1)} \tag{3.6}$$

Table 3.3 gives theoretical and experimental values of local resiliency for the proposed schemes (CD-KPD and CD-RKPD). So, lower the value of $Ll(K)$, higher is the resiliency of the scheme. From the table it is evident that the experimental values almost matches the theoretical values for local resiliency in both the proposed schemes. Further, the value of $Ll(K)$ always remain low even with the increased network size and increased compromised sensor nodes in the network. Thus, we can conclude that the proposed schemes provide high resiliency against a large number of compromised sensor nodes (K) in the network. We can also observe that the proposed schemes (CD-KPD and CD-RKPD) performs equally well for different network sizes.

Table 3.3: Theoretical and Experimental values of Local Resiliency $Ll(K)$ for CD-KPD and CD-RKPD

N	n	$k+1$	K	$Ll(K)$ **Experimental**	$Ll(K)$ **Theoretical**
400	307	18	500	0.0707	0.0732
400	307	18	600	0.0843	0.0879
625	381	20	1000	0.0807	0.0839
900	553	24	1200	0.0563	0.0578
1225	993	32	2000	0.0512	0.0526
1600	1407	38	2200	0.0364	0.0371

b) Global Resiliency $(Lg(S))$ for CD-KPD

To secure the inter-cell communication in CD-KPD, each CH has shared secret keys with other CHs. There are three CHs in each cell and different types of keys are assigned to each CHs. So, for inter-cell communication, sensor nodes can use any one of three CHs to create a secure link for communication. CD-KPD makes sure that each cell is connected with all other cells in network individually by $Type$ 1, $Type$ 2 and $Type$ 3 keys. There are two types of links we need to discuss to calculate global resiliency of the CD-KPD. First, are the $primary\ links$ which represents cell to cell connection in the network and second are the $secondary\ links$ which represents CH to CH connection. For example, if we take Cell 1 and Cell 2, there is one primary link between $Cell$ 1 and $Cell$ 2 and there are 3 secondary links between $Cell$ 1 and $Cell$ 2 namely CH_{11} and CH_{21}, CH_{12} and CH_{22}, CH_{13} and CH_{23}. If the total number of cells present in the network is N, then the number of primary links present in the whole network can be given by the formula $N(N-1)/2$. The number of secondary links is three times the primary links. Since cells are connected by multiple keys, a link between cells is broken only if all the shared keys are compromised. In other words, if all the secondary links between two cells are broken, then only the primary link between them is broken. Moreover, if any secondary link is broken between cells because of node compromise, other secondary links can be used for secure communication. So, for estimation of global resiliency of the proposed scheme, we only consider broken primary links when S CHs are compromised in the network.

Each CH has only one type of keys, either $Type$ 1 or $Type$ 2 or $Type$ 3. So, if a particular CH is compromised, it results in keys compromised of only one type. As we

have adopted a combinatorial design for key sharing, each CH is assigned $e+1$ keys and each CH shares one key with all the CHs of the same type. To fulfill this design, each key is repeated in $e + 1$ CHs. Thus, if a particular key is compromised, then $e(e + 1)/2$ secondary links are broken. So, if a CH is compromised, all the $e + 1$ keys stored in it are also compromised and these keys cannot be used for further communication. Thus, total secondary links broken because of one CH compromise are $e(e + 1)^2/2$. Suppose S CHs are compromised in the network, there can be many cases to be considered. We discuss all these cases with the number of secondary and primary links broken in each case:

1. All the compromised CHs have single type of keys either $Type$ 1 or $Type$ 2 or $Type$ 3. Then, the number of secondary links broken can be given by Equation 3.7,

$$Broken\ Secondary\ Links = \frac{Se(e + 1)^2}{2} \tag{3.7}$$

   where S is the total number of compromised CHs. Since compromised CHs have keys of only one type, other two types of keys are not compromised and thus none of the primary links are broken in the whole network.

2. All the compromised CHs have two type of keys i.e. either ($Type$ 1 and $Type$ 2) or ($Type$ 2 and $Type$ 3) or ($Type$ 1 and $Type$ 3). Then, the number of secondary links broken will be only for two types of keys. If total CHs compromised are S and compromised CHs having one type of keys are X then total secondary links broken can be given by Equation 3.8.

$$Broken\ Secondary\ Links = \left(\frac{Xe(e + 1)^2}{2}\right) + \left(\frac{(S - X)e(e + 1)^2}{2}\right) \tag{3.8}$$

   Since, compromised CHs have keys of only two type, other one type of keys are not compromised and thus none of the primary links are broken in the whole network.

3. All the compromised CHs have keys of all the $Type$ 1, $Type$ 2 and $Type$ 3. Then, secondary links will be broken in all the three types of keys and in this

case only some primary links will be broken. In other words, adversary needs to compromise all three types of keys to break some primary links. If, the total CHs compromised are S and compromised CHs of $Type$ 1 are X and $Type$ 2 are Y then total secondary links broken can be given by Equation 3.9.

$$Broken\ Secondary\ Links = \left(\frac{Xe(e+1)^2}{2}\right)+\left(\frac{Ye(e+1)^2}{2}\right)+\left(\frac{(S-(X+Y))e(e+1)^2}{2}\right) \tag{3.9}$$

This is the only case where secondary links are broken from all the three types of keys. Thus, in this case only some primary links will be broken.

If total number of secondary links broken of $Type$ 1, $Type$ 2 and $Type$ 3 are represented by T_1, T_2 and T_3 respectively. Then, total primary interlinks broken can be given by $T_1 \cap T_2 \cap T_3$. Moreover, the fraction of primary links broken depends on the overlapping of all three types of secondary links broken in the network. But, the total number of primary links broken can not be calculated deterministically. How many distinct keys and which keys of what type will be compromised is uncertain, thus we cannot predict which secondary links will be broken. Moreover, the fraction of primary links broken depends on the intersection of secondary links broken and this cannot be predicted in advance. Table 3.4 gives experimental values of global resiliency for CD-KPD. So, lower the value of $Lg(S)$, higher is the resiliency of the scheme. In the table, we can observe that the value of $Lg(S)$ always remain low even with the increased network size and increased compromised sensor nodes in the network. Thus, we can conclude that CD-KPD has very high resiliency against compromised nodes in the network. We further can observe the proposed scheme performs equally well with different network sizes and increased compromised CHs.

c) Global Resiliency $(Lg'(S))$ for CD-RKPD

For reducing the key storage overhead in CD-KPD, some changes were made to propose CD-RKPD. These changes included limiting the communication of CHs within their $Lee\ sphere$ region. These changes were incorporated by removal of unused keys from CHs which helped to reduce the key storage overhead and increased the resiliency

Table 3.4: Experimental values of Global Resiliency $Lg(S)$ for CD-KPD

N	$e+1$	S	$Lg(S)$ **Experimental**
289	18	20	0.0386
361	20	20	0.0294
529	24	25	0.0305
841	30	30	0.0252
961	32	35	0.0319
1369	38	40	0.0271
2209	48	45	0.0188
2809	54	50	0.0186

of the scheme. Table 3.5 gives the reduced number of secret keys stored by CHs in CD-RKPD in different network sizes. We can observe from the table that the key storage overhead significantly decreases in CD-RKPD when compared with CD-KPD, i.e. values of f(Min.), f(Max.) and f(Avg.) is always less than $e+1$. This indirectly boosts the resiliency of the network, as now each CH has fewer keys, so compromised CHs will expose fewer keys, resulting in less number of secondary links broken. This eventually helps in reducing the primary links broken in the whole system.

Table 3.5: Difference in number of keys stored by cluster heads in CD-KPD and CD-RKPD.

N	$e+1$	ρ	f(Min.)	f(Max.)	f(Avg.)
289	18	3	7	17	13
361	20	4	9	19	16
529	24	5	15	24	21
841	30	6	18	30	28
961	32	7	21	31	30
1369	38	7	24	38	35
2209	48	8	30	48	45

Table 3.6 gives the reduced number of primary links broken for variable network sizes when few CHs are compromised in the network. We can observe in Table 3.6 that we were able to considerably decrease the number of broken primary links in the whole network. But, this reduction in broken primary links decreases with the increase in *Lee sphere* region, this is because of the increased number of cells in the communication range. With the increase of CHs in communication range, the reduction in keys stored in each CH drops. Thus, now each CH has more keys and this results in more secondary links broken, which eventually results in more broken primary links.

But, when we limit the communication within *Lee sphere* region, the number of

Table 3.6: Difference in number of compromised Primary links in CD-KPD and CD-RKPD.

N	S	$e+1$	Links Broken (**CD-KPD**)	ρ	f(Avg.)	Links Broken (**CD-RKPD**)
289	10	18	206	4	15	99
361	15	20	837	5	18	469
529	20	24	2191	5	21	926
841	20	30	3294	7	28	2289
961	25	32	6901	7	30	4752
1369	40	38	25395	7	35	17801
2209	50	48	46005	9	46	36993

primary links in the network decreases considerably. Each cell is now only connected to $2\rho(\rho + 1)$ primary links. Thus accordingly, total primary links in the network are $2N\rho(\rho + 1)$. As the total number of primary links decreases considerably, the overall resiliency of entire system reduces.

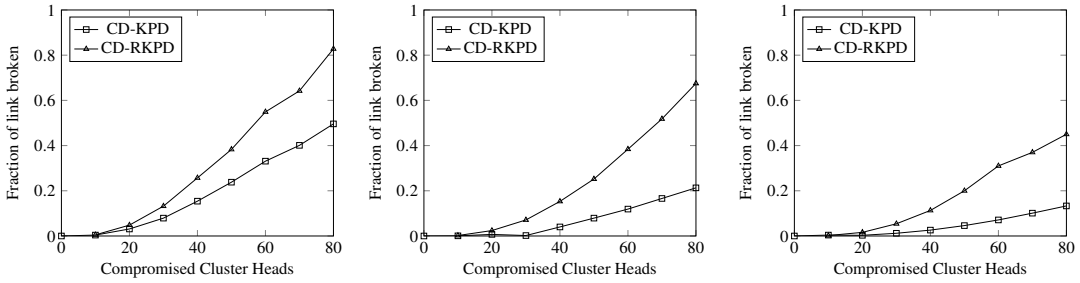
Table 3.7 gives the experimental values for global resiliency in CD-RKPD. We can observe in the table that resiliency in CD-RKPD has reduced considerably when compared with results of CD-KPD. These results are quite obvious, as limiting the communication within $Lee\ sphere$ region results in reduction of total primary links in the whole network. Thus, resiliency reduced accordingly. But, with the proposed modifications we were able to reduce the number of broken primary links in the network. Figure 3.3 gives a graphical comparison of CD-KPD and CD-RKPD with different network sizes. So, lesser the fraction of links broken, higher is the resiliency of the scheme. In the figure, we can observe that both the proposed schemes (CD-KPD and CD-RKPD) have very low fraction of links broken even when large number of cluster heads are compromised in the network. Specifically, fraction of links broken in CD-KPD and CD-RKPD are 24% and 38% respectively, when 50 CHs are randomly compromised, total cell are 289 and $\rho = 4$. Thus, we can conclude that the proposed schemes provide high resiliency against compromised CHs in the network.

3.3.3 Estimation of nodes disconnected $(D(p))$

When an adversary compromises a sensor node, all its keys are also compromised and these keys cannot be used in secure communication. So, sensor nodes have to use alternative paths and keys which are not compromised for secure communication. There

Table 3.7: Experimental values of Global Resiliency $Lg'(S)$ for CD-RKPD

N	ρ	f(Avg.)	S	$Lg'(S)$ **Experimental**
289	4	15	10	0.0085
361	5	18	15	0.0216
529	5	21	20	0.0291
841	7	28	20	0.0243
961	7	30	25	0.0441
1369	7	35	40	0.1160
2209	9	46	50	0.0930



(a) (b) (c)

Figure 3.3: Fraction of links broken when S cluster heads are compromised for CD-KPD and CD-RKPD. (a) Total number of cells are 289 and $\rho = 4$, (b) Total number of cells are 841 and $\rho = 7$, (c) Total number of cells are 1369 and $\rho = 7$.

could be a case where all the keys stored in a non compromised nodes are compromised. This can happen when sufficient nodes sharing keys with a particular node get compromised. In such a case, the non compromised node cannot securely communicate with any other node, as all its keys are compromised. Such type of nodes is termed as disconnected nodes. This parameter was initially proposed in scheme Ruj and Roy (2009). We are using the same parameter for analyzing our proposed schemes. This parameter also has two variations $Dl(K)$ and $Dg(S)$. First is the fraction of sensor nodes disconnected when K sensor nodes are compromised, denoted by $Dl(K)$. Second is the fraction of cells disconnected when S CHs are compromised, denoted by $Dg(S)$. Firstly, we will discuss local nodes disconnected $Dl(K)$ when K sensor nodes are compromised. As each sensor node is assigned $k + 1$ keys, to disconnect a particular sensor node all its $k + 1$ keys should be compromised. To do so minimum of $k + 1$ nodes have to be compromised, as each node share at most one key with other nodes. If, total K nodes are compromised, then on an average K/N nodes are compromised in a single cell. So, to disconnect a sensor node in a cell, the value of K should be such that it

fulfills $K/N > k + 1$, or more precisely the value of K should be $K > N(k + 1)$ to disconnect one sensor node. For example, in a network of 529 cells, if each cell has 529 sensor nodes, then each sensor node is assigned 24 keys. So, to disconnect a sensor node at least 12696 nodes should be compromised. So, we can see that the number of sensor nodes needed to be compromised for disconnecting a single sensor node is quite large.

Now we discuss cells disconnected $Dg(S)$ when S CHs are compromised. A cell is disconnected if all the keys of 3 CHs in a cell are compromised. So to disconnect one cell from the network, the adversary needs to compromise sufficient CHs of each type so that all the keys of 3 CHs in a particular cell can be compromised. To disconnect a particular CH, you need to compromise at least $e + 1$ CHs of the same type. If, total S CHs are compromised then on an average $S/3$ CHs are compromised of the same type. So, to disconnect a CH from network $S/3 > e + 1$ should hold, or more precisely S should always be $S > 3(e + 1)$ to disconnect one CH. And to disconnect all three CHs of a particular cell, compromised CHs (S) should be at least $S > 9(e + 1)$. This is least bound for disconnecting a particular cell from the network. In practical scenarios, this value will be much higher than discussed above. This is because for previous calculations we took the best case where each compromised CH has a unique shared key with a non compromised CH which is to be disconnected. But in general, each CH shares the same key with many CHs.

For experimental calculation of the cell disconnected, we consider two cases, first for CD-KPD denoted by $Dg(S)$ and second for CD-RKPD denoted by $D'g(S)$. The experimental values for cell disconnected for CD-KPD and CD-RKPD are given in Table 3.8. So, lesser the values of $Dg(S)$ and $D'g(S)$, lower are the disconnections in the network. We can observe from the table that the values of $D'g(S)$ are much better than $Dg(S)$. Thus, we can conclude that CD-RKPD results in fewer disconnected cells than CD-KPD when the same number of CHs are compromised in both the cases. Never the less, both the proposed schemes (CD-KPD and CD-RKPD) have very low cell disconnections even with increased network size and increased compromised CHs. Figure 3.4 gives a graphical comparison of CD-KPD and CD-RKPD with different net-

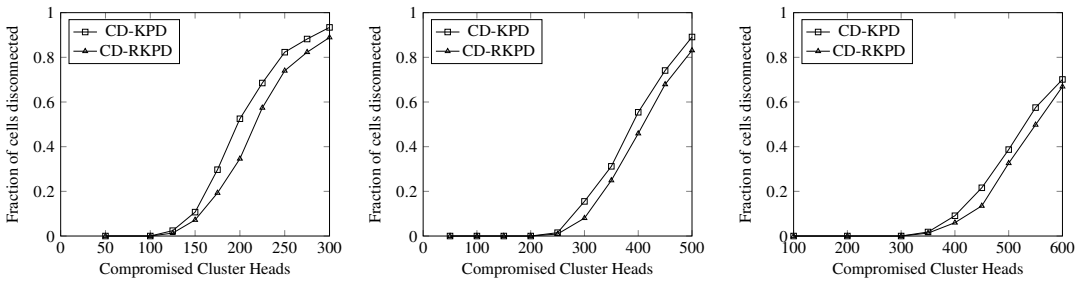


Figure 3.4: Fraction of cells disconnected when S cluster heads are compromised for CD-KPD and CD-RKPD. (a) Total number of cells are 289 and $\rho = 4$, (b) Total number of cells are 841 and $\rho = 7$, (c) Total number of cells are 1369 and $\rho = 7$.

work sizes. So, lesser the fraction of cell disconnected, higher is the resiliency of the scheme. We can observe that CD-RKPD has lower fraction of cell disconnections than CD-KPD. Specifically, fraction of cells disconnected in CD-KPD and CD-RKPD are 10% and 7% respectively, when 150 CHs are randomly compromised, total cell are 289 and $\rho = 4$. Thus, we can conclude that both the proposed schemes have very low cell disconnections in presence of compromised CHs in the network.

Table 3.8: Experimental values of Nodes Disconnected $Dg(S)$ and $D'g(S)$ for CD-KPD and CD-RKPD.

N	S	$e+1$	$Dg(S)$	ρ	f(Avg.)	$D'g(S)$
289	125	18	0.0242	4	15	0.0138
361	150	20	0.0304	5	18	0.0166
529	200	24	0.0378	5	21	0.0245
841	250	30	0.0154	7	28	0.0095
961	300	32	0.0478	7	30	0.0187
1369	350	38	0.0189	7	35	0.0131
2209	400	48	0.0004	9	46	0

3.3.4 Comparison with existing schemes

Our proposed schemes (CD-KPD and CD-RKPD) have several advantages over existing schemes. In our schemes we have 3 CHs in each cell, thus compromising of single CH in a particular cell does not disconnect a cell from the network. Moreover, the adversary needs to compromise all the 3 CHs in a particular cell to disconnect a cell from a network. In our scheme for the assignment of keys in a particular cell, we use different key pools. Thus, the compromised node in a particular cell does not affect the remaining

network. Each cell shares three keys with other cells, providing high resiliency against compromised CHs. The resiliency of our schemes is much higher than the majority of existing schemes. Our schemes are based on combinatorial design, which does not require deployment knowledge for key exchange. In our scheme, we need deployment knowledge only for CD-RKPD where we try to limit the communication of cells within their *Lee sphere* region. Moreover, here we can use relative positioning of cells, saving a lot of energy when compared with existing schemes which require the geographical location of sensor nodes. Comparative analysis of resiliency of our schemes with other schemes is presented in Figure 3.5. So, lower are the fraction of links broken, higher is the resiliency of the scheme. We can notice that our schemes have lower link broken and thus higher resiliency than most of the schemes. Bag and Roy (2013) assumes that super nodes cannot be captured until all other sensor nodes in a particular cell are not compromised, thus performs very well in simulated results. But this assumption is superficial for actual WSNs. We in our schemes take into account the equal probability of CHs and sensor nodes being compromised. Further, Liu and Ning (2005) also performs much better than the proposed schemes because of the use of pair-wise keys in the network. Other than the schemes Bag and Roy (2013), Liu and Ning (2005), the proposed schemes (CD-KPD and CD-RKPD) provide significantly better resiliency when compared with other schemes.

3.4 Analysis of CD-PKPD

When a sensor node is compromised by an adversary, it obtains all the keys stored in it. Since we are using symmetric design, the same keys are used for communication by multiple nodes. Thus, all those links which use the compromised keys can no longer be used for secure communication. Also, sometimes a node can be disconnected if all its keys are compromised thereby, making it impossible to securely communicate with other nodes and vice-versa. In this section, we will measure the resiliency of the presented scheme. We will use the ratio of links broken $L(p)$ and the ratio of nodes disconnected $D(p)$ as a measure of resiliency, when p sensor nodes/CHs compromised in the network.

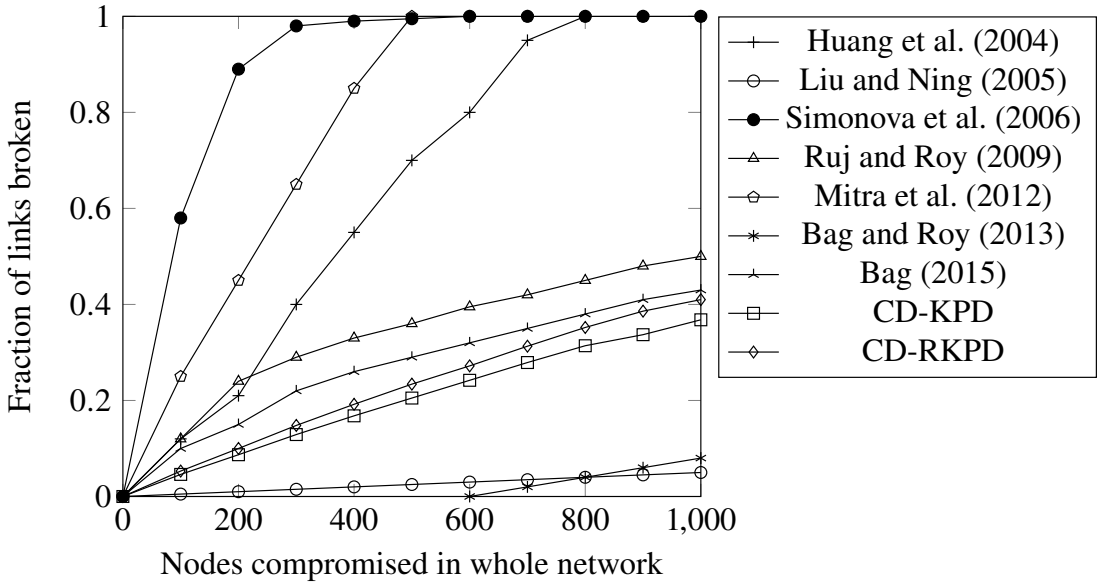


Figure 3.5: Comparison of Simonova et al. (2006) ($p = 11, k = 16$, $m = 4$ and $P = 12100$), Huang et al. (2004) ($k = 200, \tau = 3$, $\omega = 27$ and $P = 10000$), Liu and Ning (2005) ($L = 1, m = 60$, $k = 200$ and $P = 10000$), Ruj and Roy (2009) ($k = 12$ and $P = 16093$), Bag (2015) ($q = 13$ and $P = 16055$), Bag and Roy (2013) ($p = 11$, $c = 4$ and $P = 16093$), Mitra et al. (2012) ($p = 15$ and $P = 10032$), CD-KPD ($k = 12$ and $P = 16093$) and CD-RKPD ($k = 12, \rho = 4$ and $P = 16093$) (P = total sensor nodes in a WSN).

3.4.1 Estimation of resiliency ($L(p)$)

In proposed schemes, both normal sensor node and CHs are prone to be compromised. Normal sensor nodes share keys only within cells. Thus, if K normal nodes are compromised, only links inside cells will be affected which can be termed as *Local Resiliency ($L_l(K)$)*. For communication outside the cell, we use CHs which share keys with other CHs. Thus, if S CHs are compromised, links connecting cells will be broken which can be termed as $Global\ Resiliency(L_g(S))$.

a) Estimation of local resiliency ($Ll(K)$)

Local resiliency $L_l(K)$ measures the number of intra-links (i.e., links between the nodes within a cell) broken to the total number of intra-links in a cell when a node is compromised. Our proposed scheme for key-sets generation uses *difference sets* which is the same as discussed for CD-KPD. Thus, the local resiliency for the proposed scheme (CD-PKPD) is the same as discussed in Section 3.3.2.

b) Estimation of Global resiliency ($L_g(S)$)

Global resiliency $L_g(S)$ is a measure of the number of inter-links (i.e., links between nodes in different cells) broken to the total inter-links present in the network. There are two types of links to be considered for inter-cell communication namely, primary links and secondary links. A primary link represents the connection between cells whereas a secondary link denotes the connection between CHs. There can be multiple secondary links (maximum 3 in our case) between two cells, but there is only one primary link between them. When a secondary link is broken between two CHs, communication can be carried on using other secondary links. But, when all the secondary links are broken between two cells, the primary link between them is broken and nodes in one cell can no longer communicate with nodes in another cell. The total number of primary links are $N(N+1)/2$, where the total number of cells in the network is represented by N. For our scheme, we have either one, two or three secondary links between CHs in different cells. To measure global resiliency in the proposed scheme, we will consider the count of primary links affected/broken when S CHs are compromised in the network.

The CHs can be of three types namely $Type\ 1$, $Type\ 2$ and $Type\ 3$. Each CH stores $(k'+1)$ keys and each key is present in $(k'+1)$ CHs of the same type. Thus, the number of secondary links broken when a particular key is compromised are $k'(k'+1)/2$. Now if a node is compromised, all its $(k'+1)$ keys are compromised. Thus, the number of secondary links broken is $k'(k'+1)^2/2$. Let us assume *'S'* CHs are compromised and *'c'* be the number of CHs that are not used in the set generation, where $0 \leq c \leq n/4$. Since there can be one, two or three secondary links between two cells, different cases can be possible when *'S'* CHs are compromised in the network. Now, we will discuss all the different cases along with the number of primary and secondary links broken in each case:

1. All the compromised CHs are of a single type: The number of secondary links broken in such case can be given by Equation 3.10.

$$Secondary\ links\ broken = (S-c)k'(k'+1)^2/2 \tag{3.10}$$

Since, the compromised CHs are only of a particular type, the number of primary links broken is proportional to the cells which maintain only a single link for communication with other cells.

2. Compromised CHs are of two types: If *'X'* compromised CHs are of one type and out of 'X', *'x'* are not used for communication, then the number of secondary links broken in such case can be given as :

$$Secondary\ links\ broken = \{(X-x)k'(k'+1)^2/2\}+\{((S-X)-(c-x))k'(k'+1)^2/2\} \tag{3.11}$$

In this case, the number of primary links broken is proportional to the cells which maintain either one or two links for communication with other cells.

3. Compromised CHs are of three types: If compromised CHs of $Type\ 1$ and $Type\ 2$ are *'X'* and *'Y'* respectively and *'x'* and *'y'* are the nodes of $Type\ 1$ and $Type\ 2$ that are not used for communication, then total secondary links broken in such case can be given by Equation 3.12. In this case, the number of primary links broken is proportional to all the cells in the network.

$$\begin{aligned} Secondary\ links\ broken = \{(X-x)k'(k'+1)^2/2\} + \{(Y-y)k'(k'+1)^2/2\}+ \\ \{((S-(X-Y))-(c-(x+y)))k'(k'+1)^2/2\} \end{aligned} \tag{3.12}$$

In the proposed scheme a cell can have one, two or three links with any other cell. Each cell can use any one of these links for inter-cell communication. Thus, it is very difficult to determine deterministically how many primary links are broken when few CHs are compromised in the network. Experimental results for global resiliency are given in Table 3.9. So, lower the value of $L_g(S)$, higher is the resiliency of the scheme. In the table, we can observe that the value of $L_g(S)$ always remain low even with the increased network size and increased compromised sensor nodes in the network. Thus, we can conclude that CD-KPD has very high resiliency against compromised nodes in the network. The results in Table 3.9 are acquired by randomly choosing S over

Table 3.9: Experimental values of Global Resiliency $L_g(S)$ for CD-PKPD

N	k'**(Expected)**	k'**(Experimental)**	S	$L_g(S)$ **(Experimental)**
120	11	11	5	0.0559
150	13	11	5	0.0585
250	17	17	10	0.0784
350	19	17	15	0.0969
550	23	23	20	0.0968
850	29	29	25	0.0952
950	31	29	30	0.1196
1350	37	37	35	0.1142

100 iterations. The number of keys required if all the CHs were assigned key-sets is represented by k'(Expected). k'(Experimental) represents the number of keys required in the proposed scheme. We can observe from the Table 3.9 that the proposed scheme is highly resilient to compromised CHs while maintaining low key storage overhead in CHs. Figure 3.6 gives the graphical results of the proposed scheme with different number of cells in the network. So, lower are the fraction of links broken, higher is the resiliency of the scheme. We can observe that the proposed scheme has very low fraction of links broken even in variable network sizes. Moreover, the fraction of links broken reduces considerably with increased network size. Specifically, fraction of links broken in CD-PKPD are 31%, when 50 CHs are randomly compromised, total cell are 289 and $\rho = 4$. Thus, we can conclude that the proposed scheme provides high resiliency against compromised cluster heads in the network.

3.4.2 Estimation of nodes disconnected ($D(p)$)

Nodes disconnected $D(p)$ has two variations $Dl(K)$ and $D_g(S)$. First is the fraction of sensor nodes disconnected when K sensor nodes are compromised, denoted by $Dl(K)$. Second is the fraction of cells disconnected when S CHs are compromised, denoted by $D_g(S)$. Our proposed scheme has the key assignment in a cell same as of CD-KPD. Thus the sensor nodes disconnected for the proposed scheme is the same as discussed in Section 3.3.3.

Now, we will discuss the cell disconnections in the proposed scheme. A cell is disconnected when keys stored in all its CHs are revealed. CHs which are assigned keys

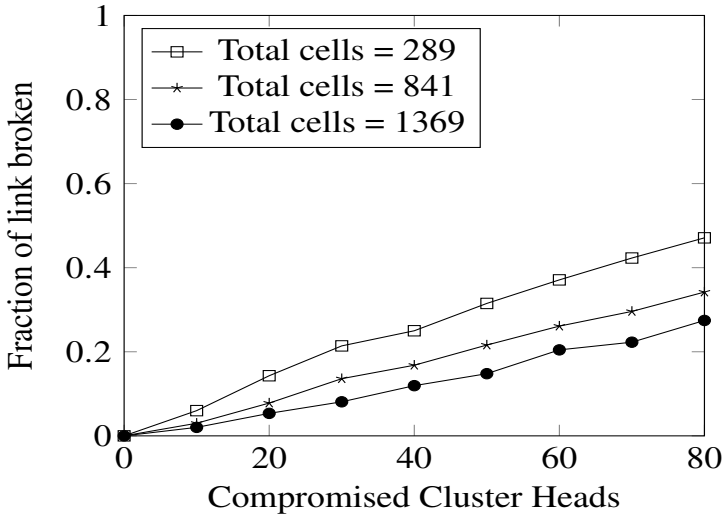


Figure 3.6: Graphical results of Global Resiliency for CD-PKPD

has $(k'+1)$ keys and each key is shared with $(k'+1)$ other CHs of the same type. Thus, to compromise the whole key-set of a CH, at-least $(k'+1)$ CHs of the same type need to be compromised. Therefore, if *'S'* CHs are compromised in the entire sensor network then on average *'S/3'* CHs of a particular type are compromised. Therefore, at-least *'S/3'*$\leq k'+1$ CHs must be compromised to disconnect a cell. In the proposed scheme we assign key-sets to only $3/4^{th}$ of the total CHs. Further, in the proposed scheme each cell has either one, two or three links with other cells and it can use any one among them for communication. Therefore, the fraction of CHs which needs to be compromised increases in the proposed scheme to disconnect cells in the network.

Experimental results for nodes disconnected in the proposed scheme are given in Table 3.10. The values in Table 3.10 are acquired by randomly selecting S over 100 iterations. So, lower the value of $D_g(S)$, lower will be the cell disconnections in the network. From the table, we can observe that the proposed scheme has very low $D_g(S)$ values even with increased network size and increased compromised cluster heads. Thus, it is evident that the proposed scheme is highly resilient to cell disconnections. Further, we can observe that a large number of CHs needs to be compromised to disconnect a few cells in the network.

Table 3.10: Experimental values of Cell Disconnected $D_g(S)$ for CD-PKPD

N	k'**(Expected)**	k'**(Experimental)**	S	$D_g(S)$ **(Experimental)**
120	11	11	75	0.0416
150	13	11	100	0.1000
250	17	17	125	0.0240
350	19	17	150	0.0314
550	23	23	200	0.0105
850	29	29	250	0.0117
950	31	29	300	0.0105
1350	37	37	350	0.0192

3.4.3 Comparison with existing schemes

Our proposed scheme CD-PKPD has several benefits over similar schemes. For the presented scheme we have three CHs in each cell, thus compromising of single CH in a particular cell does not disconnect a cell from the network. Moreover, an adversary needs to break all the links in a particular cell to disconnect a cell from a network. In our scheme for the assignment of keys in a particular cell, we use different key pools. Thus, compromised node in a particular cell does not affect the remaining network. Each cell shares either one, two or three keys with other cells, providing high resiliency against compromised CHs. The resiliency of the presented scheme is significantly higher than the majority of similar existing schemes. The proposed scheme is based on combinatorial design, which does not require deployment knowledge for key exchange, saving a lot of energy when compared with existing schemes which require the geographical location of sensor nodes. Finally, the proposed scheme facilitate more secure shared key discovery phase when compared with existing schemes. Comparative analysis of resiliency of our scheme with other schemes is presented in Figure 3.7. So, lower is the fraction of links broken, higher is the resiliency of the scheme. In the figure we can observe that the proposed scheme has lower links broken when compared to majority of existing schemes. Thus, the proposed scheme has higher resiliency than majority of the existing schemes. Further, we can observe that the proposed scheme provides better resiliency than CD-RKPD but provides marginally less resiliency than CD-KPD. Thus, all three proposed schemes (CD-KPD, CD-RKPD and CD-PKPD) provides high resiliency against compromised sensor nodes in the network.

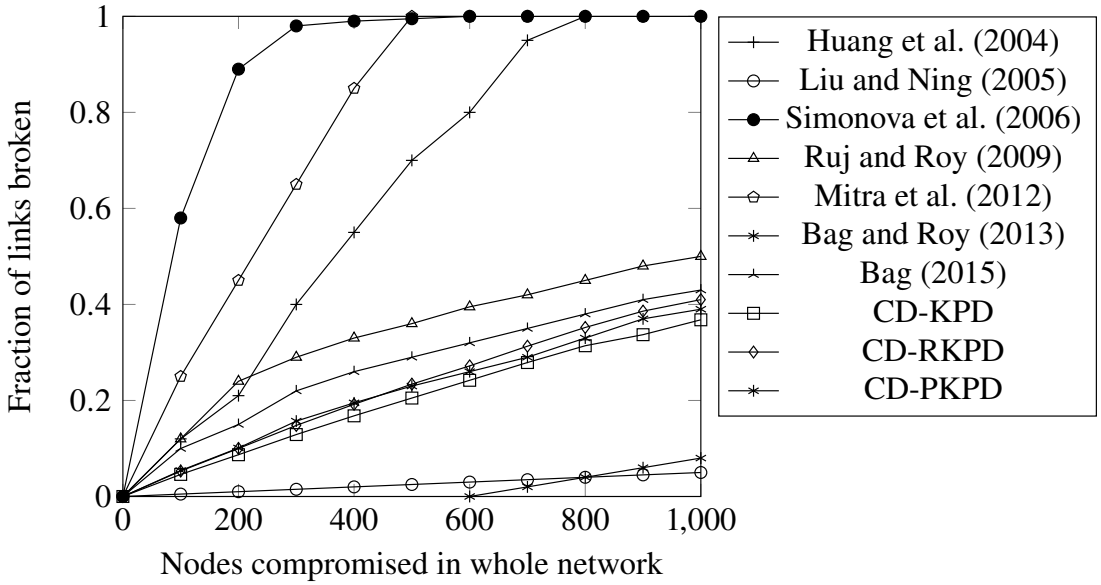


Figure 3.7: Comparison of Simonova et al. (2006) ($p = 11, k = 16$, $m = 4$ and $P = 12100$), Huang et al. (2004) ($k = 200, \tau = 3$, $\omega = 27$ and $P = 10000$), Liu and Ning (2005) ($L = 1, m = 60$, $k = 200$ and $P = 10000$), Ruj and Roy (2009) ($k = 12$ and $P = 16093$), Bag (2015) ($q = 13$ and $P = 16055$), Bag and Roy (2013) ($p = 11$, $c = 4$ and $P = 16093$), Mitra et al. (2012) ($p = 15$ and $P = 10032$), CD-KPD ($k = 12$ and $P = 16093$), CD-RKPD ($k = 12, \rho = 4$ and $P = 16093$) and CD-PKPD $\{k = k' = 12$ and $P = 16093\}$ (P = total sensor nodes in a WSN).

3.5 Concluding remarks

In this chapter, we proposed three novel combinatorial design based key pre-distribution schemes (CD-KPD, CD-RKPD, and CD-PKPD). We observed that all three proposed schemes provide better resiliency in the network when compared with the majority of the existing schemes. In the next chapter, we propose a hybrid scheme for key pre-distribution which uses both combinatorial design based keys and pair-wise keys.

CHAPTER 4

Hybrid Key Pre-Distribution

In this chapter, we propose a novel hybrid key pre-distribution scheme (CD-HKPD) based on Combinatorial design keys and Pair-wise keys. For the proposed scheme (CD-HKPD), the deployment zone is cleft into equal-sized cells. We use the combinatorial design based keys to secure intra-cell communication, which helps to maintain low key storage overhead in the network. For inter-cell communication, each cell maintains multiple associations with all the other cells within the communication range, and these associations are secured with pair-wise keys. This helps to ensure high resiliency against compromised sensor nodes in the network.

In this chapter, we introduce a new deterministic hybrid key pre-distribution scheme for the homogeneous network. In recent times many pair-wise keys based schemes (Liu et al. 2005; Liu and Ning 2005) and combinatorial design based schemes (Ruj and Roy 2009; Bag 2015; Bag and Roy 2013; Mitra et al. 2012) have been introduced, but all of them have their own associated drawbacks. We in our scheme use both pair-wise keys and combinatorial design based keys to propose a novel hybrid key pre-distribution scheme. Our scheme take advantages of both the worlds (pair-wise keys and combinatorial design based keys) but does not inherit the disadvantages of both. This helped us in obtaining much higher resiliency than Ruj and Roy (2009), Bag (2015), Bag and Roy (2013) and Mitra et al. (2012) and very less storage overhead than Liu et al. (2005), Liu and Ning (2005) and Simonova et al. (2006).

The structure of the remaining chapter is as follows: In Section 4.1, we propose our key pre-distribution scheme. Section 4.2 provides an in-depth analysis of our scheme. Section 4.3 provides a comparison of other existing schemes with our scheme. Finally, concluding remarks are given in Section 4.4. This chapter is based on the article Kumar and Pais (2018c).

Notations

Table 4.1 presents the notations used in this chapter.

Table 4.1: Notations

N	Total number of cells in network.
n	Number of nodes in a particular cell.
$k+1$	Number of keys assigned to each node.
ρ	*Lee sphere* region.
P_i	Set of keys assigned in a particular cell.
C_i	i^{th} cell in the network.
K	Sensor nodes compromised in the network including Heads.
K'	Heads compromised in the network.
r_i	Heads compromised in cell C_i.
K_i	Sensor nodes compromised in cell C_i.
(x_c, y_c)	Center location of a particular cell.
SP	Security Parameter.

4.1 Hybrid key pre-distribution scheme

We now propose the key pre-distribution scheme followed by the shared key discovery in the network. For the proposed scheme (CD-HKPD), we presume sensor nodes are evenly distributed in the network and the whole network is further split into identical-sized cells. The total number of cells in the network is N.

4.1.1 Outline

In the proposed scheme (CD-HKPD), all the sensor nodes in a cell can directly communicate with each other. Further, sensor nodes of a particular cell can also communicate with sensor nodes in other cells which are within its communication range. For considering the communication range of sensor nodes, we use $Lee\ Sphere$ region (Section 3.1.3), where sensor nodes of a particular cell only communicate with sensor nodes of other cells which are within it's *Lee sphere* region. At the time of deployment, a fixed number of sensor nodes (known as $cell\ identifiers$) in each cell are assigned ρ and

(x_c, y_c), where ρ is the chosen *Lee sphere* region and (x_c, y_c) is center of the cell where these sensor nodes are deployed. After the deployment, $cell\ identifiers$ in particular cell C_i collaborate with other $cell\ identifiers$ in neighboring cells to identify cells which are within its $Lee\ Sphere$ region (refer Figure 3.2). We observe that only the center of each cell is used by $cell\ identifiers$ for calculations of $Lee\ Sphere$ region. Thus, the actual deployment location of $cell\ identifiers$ in each cell do not affect the calculations of $Lee\ Sphere$ region until $cell\ identifiers$ are deployed in correct cells. So, the proposed scheme (CD-HKPD) is more tolerant to errors in the deployment knowledge when compared with grid-based scheme like Simonova et al. (2006), Ruj and Roy (2009), Bag (2015), Bag and Roy (2013) and Mitra et al. (2012).

Communication in the whole network is secured by secret keys. For securing intra-cell communication, we use combinatorial design based keys. For securing inter-cell communication we use pair-wise keys. Both of these are discussed in next subsections.

4.1.2 Key pre-distribution for intra-cell communication

In this section, we discuss the key pre-distribution inside the cells to ensure secure intra-cell communication. For the proposed scheme (CD-HKPD), each cell has n sensor nodes which can directly communicate with each other. Each sensor node is allocated a set of keys chosen from a key pool also known as key-sets. Two sensor nodes share common secret keys in their key-sets to ensure secure communication. For a symmetric design, each cell has $k^2 + k + 1$ sensor nodes, and each sensor node is allocated $k + 1$ keys, where k is prime. If in any case, the number of sensor nodes (n) is not of the form $k^2 + k + 1$ for any prime number k, then we opt for smallest prime number k which satisfies $n <= k^2 + k + 1$. The procedure for key set generation and assignment is the same as previously discussed in Section 3.2.1.

4.1.3 Key pre-distribution for inter-cell communication

For inter-cell communication in the proposed scheme (CD-HKPD) we use pair-wise keys. Each cell maintains multiple associations with all the other cells which are within its communication range. The associations can be used by any sensor node to communi-

cate with the sensor nodes in other cells. These associations are secured with pair-wise keys. Sensor nodes for creating these associations are randomly selected from all the sensor nodes in the cell and are called heads. Moreover, a sensor node can be associated with a maximum of one sensor node present in another cell. The number of associations between any two cells is fixed and can be termed as a security parameter (SP) for the scheme. $Cell\ identifiers$ collaborate to identify cells which are within the *Lee sphere* region. Further, in a particular cell, $cell\ identifiers$ and sensor nodes collaborate to create the associations with all the cells within the *Lee sphere* region.

A sensor node n_i is a head for the cell C_i if it has a key with some sensor node n_j present in cell C_j where cell C_i and C_j are within communicating range. $Cell\ identifiers$ of both cells C_i and C_j collaborate together to identify whether C_i and C_j are within *Lee sphere* region or not. If the total number of sensor nodes present in a cell is n and ρ represents given *Lee sphere* region, then the maximum value of SP for the network can be given by $n >= SP(2\rho(\rho + 1))$. The total heads chosen in a particular cell will be $SP(2\rho(\rho + 1))$, and each head will be storing just one extra key than other non-head sensor nodes in a particular cell. All the keys used in the whole network are unique. Construction algorithm for the same is given in Algorithm 4.1 which takes $O(m)$ time, where m is the total number of cells in communication range.

Algorithm 4.1: Key assignment to heads in the cell

Input: Cell (C_i), Security Parameter (SP) and *Lee sphere* region (ρ)
Output: SP associations each with all the cells within *Lee sphere* region (ρ) for cell C_i

```
1  Identification of neighboring cells within Lee sphere region of cell (C_i) by
     cell identifiers nodes.
2  List of cells (m) within Lee sphere region of cell C_i .
3  for j ← 1 to m do
4      if C_i has not done association with cell C_j in previous steps then
5          for i ← 1 to SP do
6              Randomlly select node k_i ∈ C_i such that node k_i is not a head
7              Randomlly select node k_j ∈ C_j such that node k_j is not a head
8              Assign key to k_i and k_j
9          end
10         end for
11     end
12 end
13 end for
```

We maintain multiple associations within any two cells in communication range, where at the time of communication any one association is selected randomly. Thus, the job of communication between any two cells is equally divided between all the associations. This ensures that no particular association has to overwork, culminating in almost equal utilization of battery power in heads.

4.1.4 Shared key discovery in the network

Shared key discovery inside a particular cell for intra-cell communication takes place using Bloom filter (Bloom 1970), as we used in Section 3.2.2.

In the proposed scheme (CD-HKPD), we create multiple associations between all the cells within the communication range. These associations are secured with pairwise keys. Thus, shared key discovery is not required in heads to identify the shared secret keys.

4.2 Analysis

Now we inspect the security aspects of the proposed scheme (CD-HKPD). We perform analysis of the proposed scheme (CD-HKPD) in terms of well-known measures, i.e., $E(s)$ and $V(s)$ (Same as discussed in Section 3.3), where $E(s)$ is the resiliency measure and $V(s)$ is the disconnection measure. These are the most widely used and standard measures for analyzing any key pre-distribution scheme.

4.2.1 Estimation of $E(s)$

When s sensor nodes are compromised in the network, $E(s)$ can be defined as the ratio of total links affected to the total number of links in the network. Mathematically, $E(s) = \frac{links\ effected}{total\ link}$, when s sensor nodes are compromised. Here the term "effected" implies that link cannot be used in further communication. Let the total number of sensor nodes compromised randomly are K, out of these K sensor nodes number of heads compromised are K'. We first study local resiliency $El(K)$ (fraction of intra-links effected when K sensor nodes are compromised), then we study global resiliency $Eg(K')$ (fractions of inter-links effected when K' heads are compromised) and finally we study

$Eo(K)$ (fraction of links (intra-links and inter-links) effected when K sensor nodes are compromised in the network).

a) Estimation of Local resiliency $El(K)$

The proposed key pre-distribution scheme makes sure that each sensor node shares a key with all the other sensor nodes in a particular cell. To ensure this, key-sets are generated using *difference sets* which is the same as discussed for CD-KPD (Section 3.2.1). Thus, the local resiliency for the proposed scheme (CD-HKPD) is the same as discussed in Section 3.3.2.

b) Estimation of Global resiliency $Eg(K')$

In the proposed scheme (CD-HKPD), sensor node of a particular cell can also communicate with sensor nodes present in other cells which are within its communication range. The number of cells within the communication range of a particular cell can be given by $2\rho(\rho+1)$, where ρ is $Lee\ sphere$ region. The inter-cell communication has to be done through multiple associations maintained with all the cells in the communication range. These associations are assigned pair-wise keys to provide end to end secure communication. All the pair-wise keys used in the whole network are unique. Thus, an association will be secure until one of its end points is compromised. Each cell has an equal number of associations with all the cells in its communication range and is denoted by SP (security parameter). Thus, two cells can communicate securely until all these SP associations are effected by compromised nodes. If we assume total heads compromised in a particular cell are r, then r associations will be broken and in the worst case total r/SP inter-links will be broken. Thus, global resiliency for a cell can be given by $Eg(r) \leq \frac{\frac{r}{SP}}{2\rho(\rho+1)}$. The global resiliency of the entire network can be given by $Eg(K') \leq \sum_{i=0}^{N} \frac{r_i}{2\rho(\rho+1)SP}$. This can further be simplified to get $Eg(K') \leq \frac{\sum_{i=0}^{N} r_i}{2N\rho(\rho+1)SP}$. Now, if total number of nodes compromised in a cell C_i are K_i and total sensor nodes in each cell are n, then the probability (P_{k_i}) that r_i heads are compromised in cell C_i when

K_i nodes are captured is given by Equation 4.1.

$$P_{k_i} = \frac{\binom{2\rho(\rho+1)SP}{r_i}\binom{n-(2\rho(\rho+1)SP)}{K_i-r_i}}{\binom{n}{K_i}} \tag{4.1}$$

Accordingly, the expected number of heads compromised in cell C_i, when K_i nodes are compromised can be calculated by Equation 4.2, where $EXP()$ represents expectation operator.

$$EXP(r_i) = \sum_{i=0}^{2\rho(\rho+1)SP} r_i \frac{\binom{2\rho(\rho+1)SP}{r_i}\binom{n-(2\rho(\rho+1)SP)}{K_i-r_i}}{\binom{n}{K_i}} \tag{4.2}$$

Equation 4.2 can further be modified to get Equation 4.3.

$$EXP(r_i) = \sum_{i=1}^{2\rho(\rho+1)SP} 2\rho(\rho+1))SP \frac{\binom{2\rho(\rho+1)SP-1}{r_i-1}\binom{n-(2\rho(\rho+1)SP)}{K_i-r_i}}{\binom{n}{K_i}} \tag{4.3}$$

Finally, we can derive Equation 4.4.

$$EXP(r_i) = 2\rho(\rho+1))SP \sum_{i=1}^{2\rho(\rho+1)SP} \frac{\binom{2\rho(\rho+1)SP-1}{r_i-1}\binom{n-(2\rho(\rho+1)SP)}{K_i-r_i}}{\binom{n}{K_i}} \tag{4.4}$$

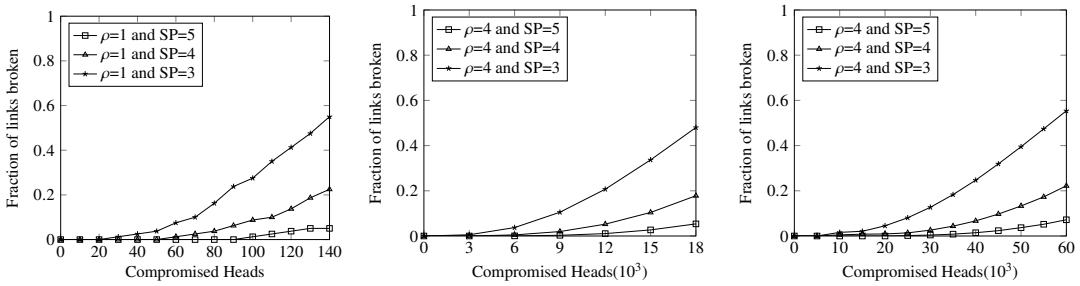
The value of $EXP(r_i)$ from Equation 4.4 can be assigned to the global resiliency to get Equation 4.5.

$$Eg(EXP(K')) \leq \frac{1}{N} \sum_{C_i=0}^{N} \sum_{r_i=1}^{2\rho(\rho+1)SP} \frac{\binom{2\rho(\rho+1)SP-1}{r_i-1}\binom{n-(2\rho(\rho+1)SP)}{K_i-r_i}}{\binom{n}{K_i}} \tag{4.5}$$

The experimental results for $Eg(K')$ are given in Table 4.2. These results are obtained by choosing K' randomly from the network over 100 iterations. So, lower the value of $Eg(K')$, higher is the resiliency of the scheme. In the table we can observe that the proposed scheme has very low $Eg(K')$ values even with increased network sizes and increased compromised cluster heads. Thus, the proposed scheme provides very high resiliency against compromised cluster heads. Further, the proposed scheme (CD-HKPD) is equally efficient for sparse and dense networks. Figure 4.1 provides the performance of the proposed scheme (CD-HKPD) with certain values of parameters.

Table 4.2: Experimental values of $Eg(K')$ for the proposed scheme (CD-HKPD)

n	N	ρ	k	SP	K'	$Eg(K')$ Experimental
8	25	1	2	3	20	0.050
25	25	1	5	5	150	0.075
49	49	1	7	5	300	0.0714
289	289	4	17	5	10000	0.0281
361	361	5	19	5	20000	0.0057
529	529	5	23	5	40000	0.0146
841	841	6	29	5	60000	0.0026
961	961	7	31	5	80000	0.0016



(a) (b) (c)

Figure 4.1: Global resiliency of the proposed scheme (CD-HKPD) when K' heads are compromised. (a) $N = 25$ and $n = 25$, (b) $N = 289$ and $n = 289$, (c) $N = 529$ and $n = 529$.

So, lower the fraction of links broken, higher is the resiliency of the scheme. In the figure we can observe that very less fraction of links are broken even when a large number of cluster heads are compromised in the network. Specifically, fraction of links broken in CD-HKPD are 8%, when 100 CHs are randomly compromised, $N = 289, n = 289$, $\rho = 4$ and SP = 4. Thus, it is evident that the proposed scheme provides high resiliency against compromised cluster heads. Further, in the figure we can observe that resiliency of the proposed scheme (CD-HKPD) increases if we increase the security parameter (SP).

c) Estimation of Overall resiliency $Eo(K)$

Now we will study the overall resiliency of the proposed scheme (CD-HKPD). Firstly we discuss the cases in which links are effected in the network when some sensor nodes are compromised in the network. The cases are as follows:

1. Intra-links disrupted because of compromised nodes in a cell (local resiliency).

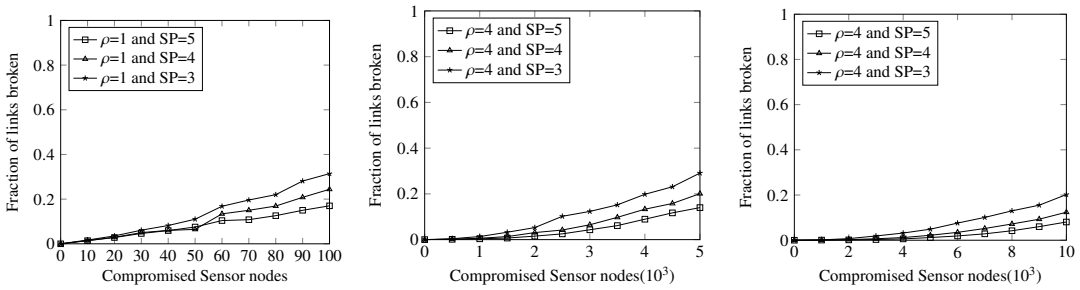
2. Inter-links disrupted because of compromised heads in a cell (global resiliency).

3. Inter-links disrupted because of compromised keys in the key-set of any head.

4. Inter-links disrupted because of compromised keys in the key-set of head present in another cell with whom particular cell maintains the association.

The first and second cases are the scenarios we discussed in local and global resiliency respectively. Third and fourth cases make the study of overall resiliency very important. Each head has two types of keys, one key-set for ensuring secure communication within its cell and other is the single key used to secure the association. In global resiliency, we studied the effect of any head compromise, but we did not study what happens if some keys from the key-set of a particular head are compromised. So, in both third and fourth cases if some of the keys from the key-set are compromised in any pair of heads which maintain an association, then that association cannot secure inter-cell communication between all the sensor nodes in two cells. Thus, effecting overall resiliency of the network.

Theoretical bound for $Eo(K)$ is very difficult to estimate because it depends on multiple parameters including position and type of sensor nodes being compromised. Moreover, all the cases (1-4) which effect the links (inter-links and intra-links) in the network are inter-dependent. Thus, the effect of any particular case cannot be quantified. Finally, the effect of third and fourth cases cannot be predicted, because we cannot predict how many keys from the key-set of a particular head will be compromised at any point of time. We give experimental results for the overall resiliency of the network in Table 4.3. These results are obtained by choosing K randomly from the network over 100 iterations. So, lower the value of $Eo(K')$, higher is the resiliency of the scheme. In the table we can observe that the proposed scheme has very low $Eo(K')$ values even with increased network size and increased compromised cluster heads. Thus, the proposed scheme provides very high overall resiliency against compromised cluster heads. Further, the proposed scheme (CD-HKPD) is equally efficient for sparse and dense networks. Figure 4.2 provides the results of $Eo(K)$ for different network sizes. So, lower the fraction of links broken, higher is the resiliency of the scheme. In the figure we



(a) (b) (c)

Figure 4.2: Overall resiliency ($Eo(K)$) of the proposed scheme (CD-HKPD) when K sensor nodes are compromised. (a) $N = 25$ and $n = 25$, (b) $N = 289$ and $n = 289$, (c) $N = 529$ and $n = 529$.

can observe that very less fraction of links are broken even when a large number of cluster heads are compromised in the network. Specifically, fraction of links broken in CD-HKPD are 6%, when 50 CHs are randomly compromised, $N = 25, n = 25$, $\rho = 1$ and SP=4. Thus, it is evident that the proposed scheme provides high overall resiliency against compromised cluster heads. Finally, in the figure we can observe that the overall resiliency of the proposed scheme (CD-HKPD) increases if we increase the security parameter (SP).

Table 4.3: Experimental values of $Eo(K)$ for the proposed scheme (CD-HKPD)

n	**N**	ρ	**k**	**SP**	K	$Eo(K)$ **Experimental**
8	25	1	2	3	10	0.0611
25	25	1	5	5	100	0.170
49	49	1	7	5	200	0.0651
289	289	4	17	5	3000	0.0467
361	361	4	19	5	4000	0.0326
529	529	5	23	5	5000	0.0120
841	841	6	29	5	8000	0.0049
961	961	7	31	5	10000	0.0054

4.2.2 Estimation of V(s)

When a sensor node is compromised by an adversary, keys stored in the sensor nodes are revealed. In some cases all the keys allocated to a non-compromised node can also be revealed, this happens when multiple sensor nodes sharing keys with this non-compromised node are compromised. Now, this node cannot communicate with other nodes in the network. Thus, it is disconnected from the network. When s sensor nodes

are compromised, $V(s)$ can be defined as the ratio of total nodes disconnected to the total number of nodes in the network. Mathematically, $V(s) = \frac{nodes\ disconnected}{total\ nodes}$, when s sensor nodes are compromised. This parameter was formulated by Ruj and Roy (2009). We are using the same parameter to analyze our scheme. We first study nodes disconnected $Vl(K)$ (fraction of nodes disconnected when K sensor nodes are compromised), then we study cells disconnected $Vg(K')$ (fractions of cells disconnected when K' heads are compromised), and finally we study $Vo(K)$ (fraction of total disconnections (nodes and cells) when K sensor nodes are compromised).

a) Estimation of Nodes disconnected $Vl(K)$

The proposed key pre-distribution scheme makes sure that each node shares a key with all the sensor nodes in a particular cell. To ensure this, key-sets are generated using *difference sets* which is the same as discussed for CD-KPD (Section 3.2.1). Thus, the nodes disconnected for the proposed scheme (CD-HKPD) is the same as discussed in Section 3.3.3.

b) Estimation of Cells disconnected $Vg(K')$

In the proposed scheme (CD-HKPD), inter-cell communication takes place through multiple associations. To disconnect a cell from the network, all its associations should be broken. If the number of compromised heads in each cell are same and in each cell r_i heads are compromised, then non-compromised associations in a particular cell C_i will be $(2\rho(\rho+1)SP - r_i)$. To disconnect cell C_i from the network, these non-compromised associations should be effected from neighboring cells. If, the number of non-effected associations between neighboring cell C_j and C_i are N_j, then the probability $(P_{r_j}^{N_j})$ that these N_j heads will be effected because of r_j heads compromised in cell C_j is given by Equation 4.6.

$$P_{r_j}^{N_j} = \frac{\binom{N_j}{N_j}\binom{2\rho(\rho+1)SP-N_j}{r_j-N_j}}{\binom{2\rho(\rho+1)SP}{r_j}} \tag{4.6}$$

This can further be simplified to get Equation 4.7.

$$P_{r_j}^{N_j} = \frac{\binom{2\rho(\rho+1)SP-N_j}{r_j-N_j}}{\binom{2\rho(\rho+1)SP}{r_j}} \tag{4.7}$$

Further probability to affect all the non-compromised heads of cell C_i from all the neighboring cells can be given by Equation 4.8.

$$P(r_i) = \prod_{i=1}^{2\rho(\rho+1))} \frac{\binom{2\rho(\rho+1)SP-N_i}{r_i-N_i}}{\binom{2\rho(\rho+1)SP}{r_i}} \tag{4.8}$$

This is the probability of cell C_i to be disconnected from the network when r_i heads were compromised in C_i and all neighboring cells of C_i. Finally $Vg(K') = EXP(P(r_i))$, where EXP() represents expectation operator. The performance of the proposed scheme (CD-HKPD) in terms of $Vg(K')$ with different parameter values is laid out in Table 4.4. These results are obtained by choosing K' randomly in the network over 100 iterations. So, lower the value of $Vg(K')$, lower will be the cell disconnections in the scheme. In the figure we can observe that proposed scheme maintains very low $Vg(K')$ values even with increased network size and increased compromised cluster heads in the network. Thus, it is evident that the proposed scheme ensures very low cell disconnections in the network and it is practically impossible to disconnect a cell from the network.

Table 4.4: Experimental values of Cells Disconnected $Vg(K')$ for the proposed scheme (CD-HKPD)

n	**N**	ρ	SP	**k**	K'	$Vg(K')$
8	25	1	2	3	75	0.12
25	25	1	4	5	200	0.16
25	25	1	5	5	200	0.04
49	49	1	4	7	400	0.122
49	49	1	5	7	400	0.0204
289	289	4	4	17	35000	0.0588
289	289	4	5	17	35000	0.0034
529	529	5	4	23	105000	0.0207
529	529	5	5	23	105000	0.0094
841	841	6	4	29	140000	0
961	961	7	4	31	175000	0

c) Estimation of Overall disconnections $Vo(K)$

Finally, we study overall disconnections in the network when a fixed number of sensor nodes are compromised in the network. For the study, we take into account all the sensor nodes and cells in the network. In overall resiliency (Section 4.2.1), we noticed many new cases which effect inter-links between the cells. Those cases also apply in the study of nodes and cells disconnected in the network. We take all the cases into consideration, where cells can be compromised because of compromised associations from neighboring cells and because of all keys compromised in the key-set of a particular head.

Similar to Overall resiliency $Eo(K)$, the theoretical bound for $Vo(K)$ is very difficult to estimate because it depends on multiple parameters including position and type of sensor nodes being compromised. We provide the experimental results for overall disconnections including nodes disconnected and cells disconnected when K sensor nodes are compromised in the network. The number of sensor nodes disconnected in the network is always the same as local sensor nodes disconnected $Vl(K)$. The effect of cases (3 and 4) from $Eo(K)$ is only on cells disconnected in the network. Thus, for calculating experimental results for $Vo(K)$ we only consider total cells disconnected in the network. Table 4.5 provides the experimental results for the same. These results are obtained by choosing K randomly from the network over 100 iterations. So, lower the value of $Vl(K)$, lower are the overall disconnections in the scheme. In the figure we can observe that proposed scheme maintains very low $Vl(K)$ values even with increased network size and increased compromised cluster heads in the network. We can observe from the table that only four cells on average are disconnected from the network ($N = 361, n = 361, \rho = 4, SP = 4$) when around 45000 sensor nodes are compromised in the network. Moreover, the fraction of cells disconnected also decreases with an increase in SP. Thus, it is evident that the proposed scheme ensures very low overall disconnections in the network.

4.3 Comparison with existing schemes

In this section, we present a comparative analysis of the proposed scheme (CD-HKPD) with existing schemes in terms of communication overhead, storage overhead, and re-

Table 4.5: Experimental values of Overall Disconnections $Vo(K)$ for the proposed scheme (CD-HKPD)

n	**N**	ρ	SP	**k**	K	$Vo(K)$
8	25	1	2	3	75	0.08
25	25	1	4	5	200	0.08
25	25	1	5	5	200	0.04
49	49	1	4	7	800	0.102
49	49	1	5	7	800	0.0408
289	289	4	4	17	15000	0.0138
289	289	4	5	17	15000	0
529	529	5	4	23	45000	0.0094
529	529	5	5	23	45000	0.0018
841	841	6	4	29	100000	0.0011
841	841	6	5	29	100000	0
961	961	7	4	31	150000	0.0343
961	961	7	5	31	150000	0.0184

siliency. Table 4.6 gives a detailed analysis of the same. Table 4.7 provides the key storage overhead of all the existing schemes and the proposed scheme (CD-HKPD). Figure 4.3 provides a comparison of resiliency for existing schemes with the proposed scheme (CD-HKPD).

Liu et al. (2005), Liu and Ning (2005) introduced a key pre-distribution scheme for

Table 4.6: Comparison of existing schemes with the proposed scheme (CD-HKPD)

Schemes	**Types of Keys**	**Deployment Type**	**Network Type**	**Storage Overhead**	**Resiliency**
Liu et al. (2005), Liu and Ning (2005) (LN)	Pair-wise	Cell based	Homogeneous	Very High	Very High
Huang et al. (2004) (HMMH)	Key Pool	Grid-Cell	Homogeneous	Very High	Very Low
Simonova et al. (2006) (SLW)	Combinatorial Design	Grid-Cell	Homogeneous /Heterogeneous	High	Very Low
Ruj and Roy (2009) (RR)	Combinatorial Design	Grid-Cell	Heterogeneous	Low	Moderate
Bag (2015) (SB)	Combinatorial Design	Grid-Cell	Heterogeneous	Low	Moderate
Bag and Roy (2013) (BR)	Combinatorial Design	Grid-Cell	Heterogeneous	Low	Very High
Mitra et al. (2012) (MMD)	Combinatorial Design	Grid	Homogeneous	Very Low	Very Low
proposed scheme(CD-HKPD)	Hybrid	Cell	Homogeneous	Very Low	Very High

group-based deployment in a homogeneous network. The scheme used pair-wise keys in each group. Thus, the storage overhead was very high. If, each cell has n sensor nodes, then the number of keys allocated to each sensor node is $O(n^2)$. But, in the proposed scheme (CD-HKPD) we used the combinatorial design for key assignment inside

the cells for intra-cell communication. Thus, the maximum number of keys allocated to any sensor node in the proposed scheme (CD-HKPD) is $O(\sqrt{n})$.

Huang et al. (2004) adopted multiple space blom scheme to propose a new key pre-distribution scheme for the homogeneous network. In their scheme, sensor nodes in a particular cell can do intra-cell communication with probability > 0.5. Our scheme ensures that each sensor node can communicate with all the sensor nodes in the same cell with a probability of 1. Thus, our scheme reduces the overhead and delay for communication within cells.

Based on transversal design (Stinson 2007), Simonova et al. (2006) proposed a new key pre-distribution scheme for heterogeneous network. There are two types of sensor nodes in the network namely, weak nodes and strong nodes. Weak nodes in the same cell can communicate directly with each other, and strong nodes are used for inter-cell communication. In the scheme Simonova et al. (2006), the number of strong nodes is dependent on the size of the network. But in the proposed scheme (CD-HKPD) number of heads can be fixed in advance and only depends on security parameter (SP) and *Lee sphere* region. Moreover, the resiliency of the proposed scheme (CD-HKPD) is much higher than scheme Simonova et al. (2006).

Ruj and Roy (2009) proposed a new key pre-distribution scheme for heterogeneous network based on Çamtepe and Yener (2007). In the scheme, there are two types of sensor nodes namely, sensor nodes and agents. Any two cells in the network communicate using agents, where multiple agents share either one, two or three keys. Thus, if any agent is compromised, many inter-links are affected in the network. But in the proposed scheme (CD-HKPD), we adopt pair-wise keys for inter-cell communication, thus compromising of any head in the network does not affect other links. So, the proposed scheme (CD-HKPD) has very high resiliency when compared with the scheme Ruj and Roy (2009).

Bag (2015) proposed a key pre-distribution scheme much similar to scheme Ruj and Roy (2009). In the scheme, each cell had the variable number of agents depending on sensor node density and network size. Thus, their scheme has huge number of agents for inter-cell communication and as keys stored in each agent are assigned using com-

Table 4.7: Key Storage Overhead in different schemes

Schemes	Keys in each Sensor Node	Keys in each Head	Connectivity
Liu et al. (2005), Liu and Ning (2005) (LN)	121	126	0.92
Huang et al. (2004) (HMMH)	68	68	0.52
Simonova et al. (2006) (SLW)	20	40	0.80
Ruj and Roy (2009) (RR)	12	24	1
Bag (2015) (SB)	12	21	1
Bag and Roy (2013) (BR)	12	24	1
Mitra et al. (2012) (MMD)	15	15	1
proposed scheme (CD-HKPD)	12	13	1

(1) Parameters for LN scheme are $\gamma = 121$ and $\mu = 5$, (2) Parameters for HMMH scheme are $\tau = 2$, $\omega = 7$, and $n_z = 100$, Parameters for SLW scheme are $p = 11$, $k = 16$ and $m = 4$, (4) Parameters for RR scheme are $p = 11$ and $k = 12$, (5) Parameters for SB scheme are $p = 11$ and $q = 11$, (6) Parameters for MMD scheme are $r = 121$ and $p = 11$, (7) Parameters for the proposed scheme (CD-HKPD) are $\rho = 3$, $SP = 4$. The total number of sensor nodes for LN, MMD is 14641, for HMMH is 10000, for RR, BR, proposed scheme (CD-HKPD) is 16093, for SLW is 12100 and for SB is 16055.

binatorial design, the number of keys stored in an agent is very high. Our proposed scheme (CD-HKPD) has a fixed number of heads in each cell based on chosen security parameter (SP) and *Lee sphere* region. Moreover, we used pair-wise keys to create associations for inter-cell communication, where each head is only associated with one head from another cell. Thus, only one key is assigned to each head for inter-cell communication which is much lower than scheme Bag (2015).

Mitra et al. (2012) proposed a new combinatorial design based key pre-distribution scheme. Authors used projective planes and pair-wise connectivity to assign keys to each sensor nodes. Thus, key storage overhead is much lower. But the network used in the scheme is not divided into cells. Thus, resiliency of the scheme is very poor. Our scheme has very high resiliency against compromised nodes with minimal key storage overhead.

Bag and Roy (2013) proposed another combinatorial design based key pre-distribution scheme for heterogeneous networks. The scheme has only one super node in each cell which is responsible for inter-cell communication. So, if any super node gets compromised, a particular cell will be disconnected from the network. But, in the proposed scheme (CD-HKPD), we maintain multiple associations with each neighboring cell, thus compromising of even multiple heads has minimal effect on the whole network.

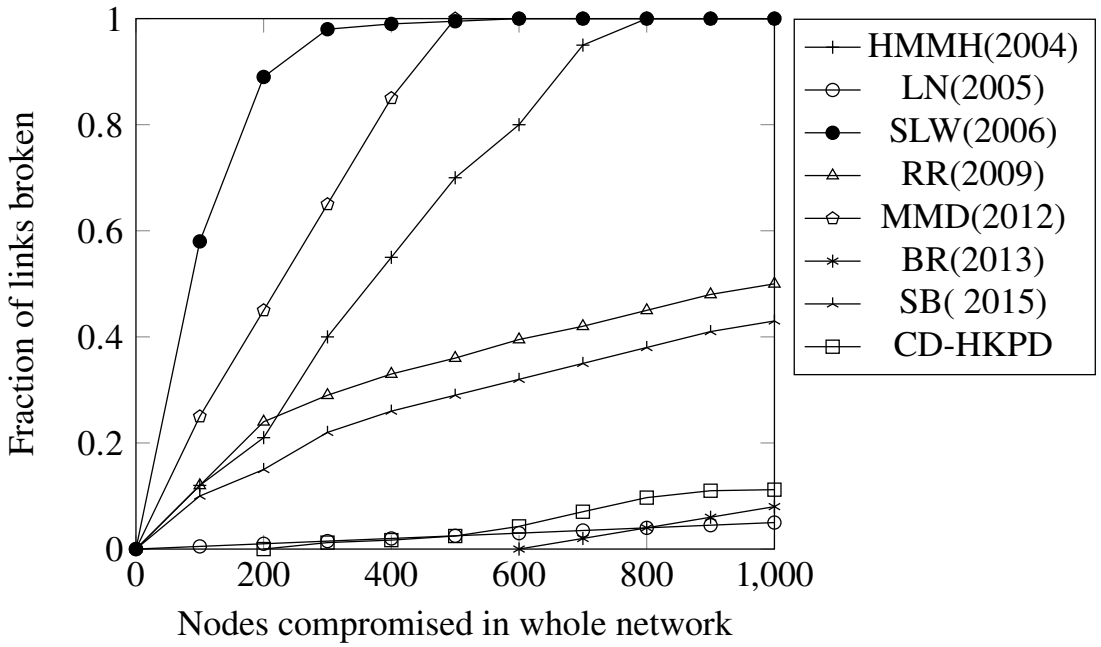


Figure 4.3: Comparison of Simonova et al. (2006) (SLW), Huang et al. (2004) (HMMH), Liu et al. (2005), Liu and Ning (2005) (LN) , Ruj and Roy (2009) (RR), Bag (2015) (SB), Bag and Roy (2013) (BR), Mitra et al. (2012) (MMD) and the proposed scheme (CD-HKPD). (1) Parameters for SLW scheme are $p = 11, k = 16$ and $m = 4$, (2) Parameters for HMMH scheme are $k = 200, \tau = 3$ and $\omega = 27$, (3) Parameters for LN scheme are $L = 1, m = 60$ and $k = 200$, (4) Parameters for RR scheme are $k = 12$, (5) Parameters for SB scheme are $q = 13$, (6) Parameters for BR scheme are $p = 11$ and $c = 4$, (7) Parameters for MMD scheme are $p = 15$, (8) Parameters for the proposed scheme (CD-HKPD) are $\rho = 3$, $SP = 4$. The total number of sensor nodes for SLW is 12100, for HMMH, DDHV, LN is 10000, for MMD is 10032, for RR, BR, proposed scheme (CD-HKPD) is 16093, for SB is 16055.

Moreover, the scheme Bag and Roy (2013) presumes that super nodes can only be compromised when all other sensor nodes have been compromised in a particular cell. But for actual WSNs this assumption is superficial. In our scheme, we take equal probability for sensor nodes and heads being compromised by an adversary.

Table 4.7 provides the storage overhead of existing schemes, and from the table, we can observe that the proposed scheme (CD-HKPD) has least storage overhead when compared with existing schemes. But this reduction in storage overhead does not affect the resiliency of the whole system. Figure 4.3 gives the comparison of the resiliency of several schemes with the proposed scheme (CD-HKPD). So, lower is the fraction of

links broken, higher is the resiliency of the scheme. In the figure we can observe that the proposed scheme has very low fraction of links broken when compared with majority existing schemes. Thus, the proposed scheme (CD-HKPD) provides better resiliency than the majority of combinatorial design based key pre-distribution schemes.

4.4 Concluding Remarks

In this chapter, we proposed a novel hybrid key pre-distribution scheme based on combinatorial design and pair-wise keys. We observed that the proposed scheme (CD-HKPD) provides significantly better resiliency from compromised nodes in the network when compared with the majority of the existing schemes. Further, the proposed scheme (CD-HKPD) is much better than CD-KPD, CD-RKPD, and CD-PKPD (discussed in the previous chapter).

In the next chapters we extend the proposed key pre-distribution schemes to propose novel En-Route filtering schemes for WSNs. In the typical En-Route filtering schemes, we use pair-wise keys for data authentication between sensor nodes and cluster heads/sink. For data endorsement/verification of the forwarded reports, we assign combinatorial design based keys to cluster heads, which ensures low key storage overhead and very effective En-Route filtering of false reports. This way of the key assignment is almost reverse to what was proposed in CD-HKPD, where we assigned combinatorial design based keys to sensor nodes to secure intra-cell communication in the network and pair-wise keys to few sensor nodes in each cell to secure inter-cell communication. This was mainly done keeping in mind one-to-one communication between sensor nodes in the network for CD-HKPD. On the other hand, in the En-Route filtering scenario, where we collect data from the sensor nodes at a sink, we have many-to-one communication in the network. Thus, we adopted pair-wise keys for intra-cell data collection/authentication and combinatorial design based keys for inter-cell data authentication. So, in the next chapters, we extend CD-KPD and CD-PKPD to propose novel En-Route filtering schemes.

CHAPTER 5

Deterministic En-Route Filtering Of False Reports

Wireless sensor networks are an easy target for report fabrication attack, where compromised sensor nodes can be used by an adversary to flood the network with bogus/false reports. En-Route filtering is a mechanism where intermediate forwarding nodes identify and drop false reports while they are being forwarded towards the sink. Most of the existing En-Route filtering schemes are probabilistic, where sensor nodes in each cell share secret keys with a fixed probability with intermediate nodes. Thus, forwarded reports are verified probabilistically by intermediate nodes, because of which false reports can travel several hops before being dropped. Few deterministic En-Route filtering schemes have also been proposed in the literature, but all such schemes require a source to send the reports through a fixed path to reach the sink.

In this chapter, we propose a novel deterministic combinatorial design based En-Route filtering scheme **(CD-EFS)**. Different from existing deterministic schemes, the proposed design does not require sending reports through a fixed pre-defined path. Further, because of the deterministic nature of the proposed scheme (CD-EFS), filtering efficiency of the proposed scheme (CD-EFS) is excellent. For the proposed scheme (CD-EFS) we assume a distributed WSN where the deployment network is partitioned into equi-sized cells. Further, we have two different types of sensor nodes in each cell namely, Cluster Heads (CHs) and ordinary sensor nodes. In the proposed scheme (CD-EFS), we assign pair-wise keys to the sensor nodes to securely communicate with

CHs/sink. For En-Route data endorsement/verification, we assign keys to the CHs based on a combinatorial design. Thus, all the CHs can communicate with each other without alarmingly increasing the key storage overhead in the network. In the proposed scheme (CD-EFS), reports are forwarded and verified only by CHs. This helps in reducing the energy requirements in sensor nodes while maintaining desired security in the network. This also helps in reducing the effect of selective forwarding attack in the network. In the proposed scheme (CD-EFS), similar to CD-KPD (Section 3.2), each cell has three CHs and all the three CHs participate in the report generation. So, three copies of each report with different endorsements are forwarded in the network, which considerably improves data authenticity in the proposed scheme (CD-EFS). We further define a novel beam model for each cell which identifies all the upstream and downstream cells. Based on the upstream and downstream cells, keys stored in each CH are further reduced. This helps in reducing the overall key storage overhead in the network.

Major contributions of the chapter are as follows:

- We have proposed a novel deterministic En-Route filtering scheme based on a combinatorial design for WSNs.
- We have proposed a novel beam model to identify the upstream and downstream region of each cell to reduce the key storage overhead.
- We have proposed a novel report endorsement and verification technique for more effective En-Route filtering of the false reports.

The remaining chapter is organized as follows: Section 5.1 provides the associated system and threat model for the proposed scheme (CD-EFS). The proposed scheme (CD-EFS) is discussed in Section 5.2. Security analysis of the proposed scheme (CD-EFS) is provided in Section 5.3, followed by performance evaluation in Section 5.4. Finally, we conclude our chapter in Section 5.5. This chapter is based on articles Kumar and Pais (2018a) and Kumar and Pais (2018b).

Notations

For convenience, we discuss all the notations (refer Table 5.1) used in this chapter.

Table 5.1: Notations

N	Total number of sensor nodes in the network.
C	Total number of cells in the network.
n	Number of sensor nodes in a particular cell.
C_c	c^{th} cell in the network.
CH_c^i	i^{th} Cluster Head in c^{th} cell.
(x_c, y_c)	Center location of a particular cell.
x_{loc}	Geographical location of sensor node x.
$k+1$	Number of keys assigned to each sensor node.
K_m, P^1, P^2, P^3	4 master secret keys.
K_x and P_c^i	Secondary secret keys calculated by sensor node x in cell c.
M	Event report.
M_x	Unique secret share for the report M calculated by sensor node x.
M_{encr}	Encrypted report.
P	Large prime number.
H	Hash function.
T	Number of endorsements to be included with each report.
t	Minimum number of correct endorsements required in each report to validate it.

5.1 System Model and Threat Model

System Model

- We assume a distributed sensor network with N sensor nodes for the proposed scheme (CD-EFS) which is partitioned into equi-sized cells. Each cell has n sensor nodes in the network. Further, there is a sink in the network which verifies/collects all the reports.

- Each cell has two types of sensor nodes namely, CHs and ordinary sensor nodes.

- For the proposed scheme (CD-EFS), we have three CHs in each cell similar to Ruj and Roy (2009). Further, we have three types of key-sets namely, $Type\ 1$, $Type\ 2$ and $Type\ 3$ generated from different key pools. These key-sets are assigned to all the CHs such that different types of key-sets are assigned to all three CHs in each cell. Thus, each CH can communicate with CHs of the same type in the whole network.

- CHs can obtain their geographical location via any localization scheme (He et al. 2003; Patwari et al. 2005) or using in-built GPS (Misra and Enge 2006). The proposed scheme (CD-EFS) can tolerate location errors as only centers of the home cell are used for the scheme.

- Each event in the network is detected by multiple sensor nodes. All the sensor nodes which detect the event, generate the report and send it to the CHs. CHs then forward the final report to the sink via multi-hop path.
- Reports verification and report forwarding are only done by CHs in the network.
- All the sensor nodes and CHs have unique IDs.

Threat Model

We assume CHs and sensor nodes can be compromised by an adversary. When an adversary compromises a sensor node/CH, all the information stored in it is revealed to the adversary. Using the obtained information, an adversary can inject false packets and drop/alter genuine packets. However, an adversary cannot compromise the sink.

5.2 Proposed scheme (CD-EFS)

In this section, we explain the process of deployment of sensor nodes in the network, including sensor node initialization, key assignment and key exchange. Further, we discuss report generation and En-Route filtering in the network.

Deployment

For the key assignment in sensor nodes and CHs, network administrator prepares four master secret keys (K_m, P^1, P^2, P^3). Administrator further chooses a large prime number (P), hash function $H(.)$, and the parameters T, t. The parameters T, t were defined in Shamir (1979), where T represents the number of secret shares to be included in each report and t represents the minimum number of correct secret shares required in each report to validate it.

5.2.1 Initialization of Sensor Nodes

Each sensor node in a particular cell is assigned K_m, P^1, P^2, P^3, P, $H(.)$, (x_c, y_c), and x_{loc}, where (x_c, y_c) is the center location of a particular cell (C_c) and x_{loc} is location of each sensor node x. Each sensor node x computes the secret key K_x using K_m and x_{loc} as $K_x = H(K_m|x_{loc})$, where | represents a concatenation operation. This key is used by

x to communicate securely with the sink. To secure communication with CHs in cell C_c, each sensor node x uses other 3 master secret keys to generate secret keys (P_c^1, P_c^2, P_c^3) as $P_c^i = H(P^i|(x_c, y_c))$. So, P_c^1 can be used by sensor node x to securely communicate with CH_c^1. After computation of keys in a sensor node, all the master secret keys are removed by it. The secret keys generation process is done by each sensor node in the network.

5.2.2 Initialization of Cluster Heads

Each CH is assigned one master key P^i, such that P^1, P^2, P^3 are individually assigned to three CHs in a particular cell. CHs can use this master key to securely communicate with sensor nodes in the cell. In addition to the master key, CHs are also assigned T and (x_c, y_c). For enabling En-Route filtering in the network, CHs are assigned keys based on a combinatorial design. In the proposed scheme (CD-EFS), each type of CHs is assigned keys from different key pools, limiting communication of CHs among the same type. If total number of cells in the network are C, then there are C CHs of type $Type\ 1$, $Type\ 2\ and\ Type\ 3$. CHs of a particular type are assigned keys based on same symmetric design explained in CD-KPD (refer Section 3.2).

Symmetric design in key-set generation ensures that any pair of key-set shares few keys. So assignment of a key-set to a CH ensures that it can verify all the reports from other CHs of the same type, that too without any shared key discovery. Moreover, the adoption of symmetric design for key-set generation helps in providing a deterministic way of sharing secret keys in the network while maintaining marginal key storage overhead. At the time of report generation, each CH creates and appends $k + 1$ MACs to the final report. When this report is forwarded through the network, each intermediate CH of the same type checks the authenticity of the report by verifying the MACs attached with the report.

When the reports are forwarded from an event cell towards the sink, they follow a narrow beam like path to reach the sink, i.e. each report is forwarded only through a limited part of the network. Thus, we do not need all $k + 1$ MACs with each report. In the proposed scheme (CD-EFS) to reduce the number of MACs required with

each report and to reduce the keys stored in each CH, $Cell - Upstream$ region and $Cell - Downstream$ region for each cell is defined. Figure 5.1 shows $Cell - Upstream$ region and $Cell - Downstream$ region for a cell C_i. In a normal network, CHs of cell C_i have to verify reports only from its Upstream region and any report being sent by cell C_i is only verified by Downstream region. Thus, CHs in cell C_i need keys shared only within $Cell - Upstream$ and $Cell - Downstream$ region. For shared key discovery in $Cell - Upstream$ and $Cell - Downstream$ region, we create $report\ verification$ and $report\ endorsement$ key lists. Both key lists construction is discussed in the next subsection. This helps in reducing the number of MACs included with each report and reduces the number of keys stored in each CH.

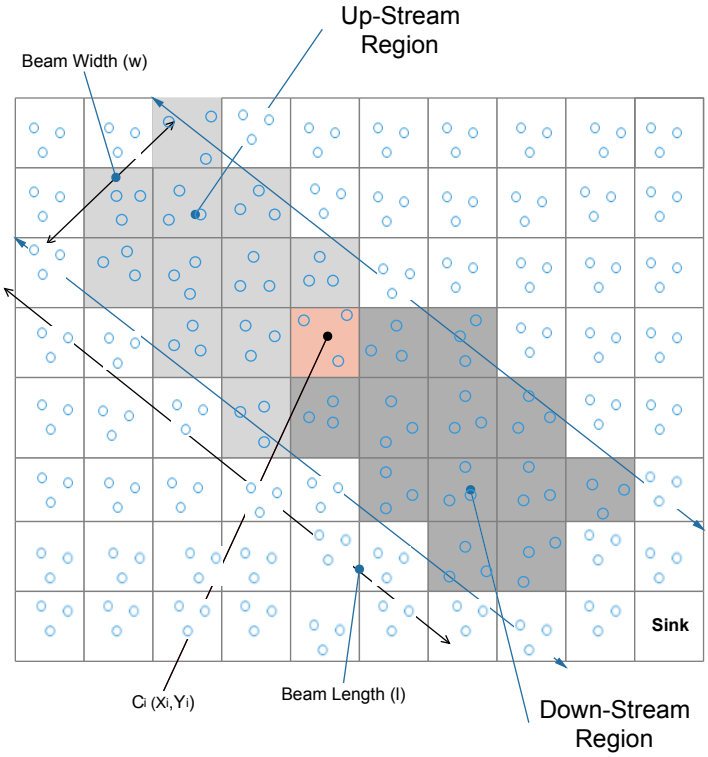


Figure 5.1: Beam model implementation in the network.

5.2.3 Creation of Report Verification and Report Endorsement key list

$Cell - Upstream$ region and $Cell - Downstream$ region for a particular cell are determined by the cell's and sink's location in the network. Both regions are represented by a parallel beam in the direction of a cell from the sink. Further, the beam width

(w) and beam length (l) are variables (refer Figure 5.1) and can be chosen by the network administrator accordingly. Specifically, $Upstream$ and $Downstream$ regions for a particular cell represents a rectangular area around it and all the cells which are covered in this area are identified by the cell using simple geometry. All these calculations can be done by any CH of each cell to identify other cells in both the upstream and downstream region. This information is then forwarded to other CHs in the same cell. Upstream region for a particular CH_c^i covers all CHs whose reports it can forward and verify. Further, the downstream region for a CH_c^i covers all CHs who can verify the reports sent by it. Thus, CH_c^i creates two key lists namely, $report\ verification$ key list and $report\ endorsement$ key list. Both the key lists are created by each CH by identifying common secret keys shared with other CHs in upstream and downstream region respectively.

For the creation of key lists, each CH_c^i creates a message containing key indices of all the keys stored in it. This message is then broadcasted in the network. When a particular CH_j^i receives this broadcasted message, CH_j^i checks whether CH_c^i is in its upstream region or downstream region. If true and CH_c^i and CH_j^i are of the same type, CH_j^i identifies the shared key using key indices from the message and it appends this secret key in either report verification key list or report endorsement key list accordingly. This process of broadcasting the message is recapitulated by all the CHs in the network. So, the communication overhead for both the key lists generation is $O(m)$, where m is the length of key indices for all the keys stored in any CH. The identification of shared key takes only $O(1)$ time. After the process of shared key discovery and key list generation, initial key-set assigned to each CH can be deleted by all the CHs, as now only $Report\ verification$ key list and $Report\ endorsement$ key list are used by CHs for report endorsement and report verification.

This process of creating two different key lists in each CH helps to reduce the keys stored in each CH and also helps to reduce the number of MACs to be sent with each report.

5.2.4 Report Generation

When an event happens, any $2T$ sensor nodes in a particular cell agrees on the event report using the technique given in Ye et al. (2005b). A typical report M contains information about the type of event, location, time of an event, etc.. After the agreement between sensor nodes for the report, each participating sensor node x creates a unique share M_x for the report with the help of predefined threshold (t, T) $LSSS$ (Ren et al. 2008). Precisely, M_x is derived by evaluating the polynomial (Equation 5.1) over $GF(P)$, where $GF(.)$ is a finite $Galois\ field$ (Dickson 2003), P and t are pre-assigned parameters, K_x represents secret exchanged between x and sink, full partition of M is denoted by p_i where i belongs to 0 to $t-1$.

$$M_x = \sum_{0 \leqslant i < t} p_i K_x mod(P) \tag{5.1}$$

This polynomial evaluation is done by all the participating sensor nodes using their secret key shared with the sink. As M_x is uniquely generated by each sensor node, it can be used by the sink as an endorsement. Further, node x encrypts the original report M using other three secret keys $P_x^1,\ P_x^2,\ P_x^3$ as $M_{encr}^i = E_{P_x^i}(M)$. Finally, sensor node s sends the tuple $\{M_x,\ x_{id},\ M_{encr}^i\}$ to each CH in the home cell by attaching appropriate M_{encr}^i.

All the three CHs in the event cell collects all the $2T$ tuples from the participating sensor nodes. Initially, the freshness of all the secret shares included in the tuples is verified. CHs also check whether the participating sensor nodes are from the home cell or not. Now, all three CHs coordinate with each other to choose T tuples from $2T$ tuples such that each CH chooses at-least 50% different tuples when compared with the other two CHs. Selection of different tuples by each CH helps to improve the data authenticity of the proposed scheme (CD-EFS) and because of which the proposed scheme (CD-EFS) is more resilient to report disruption attack. Further details for the same are given in Section 5.3. After choosing T tuples, each CH decrypts M_{encr}^i from each tuple to check whether M sent by all nodes is the same or not. Further, CHs co-ordinate to find whether at-least two CHs have got all correct

M. Next, each CH create MACs for the report M using all the keys from the report endorsement key list. Finally, each CH prepares the final report of the form $\{M_1, M_2.., ID_1, ID_2.., M, MAC_1, MAC_2..MAC_{k_i}, ID_{k_1}, ID_{k_2}..ID_{k_i}\}$, where MAC_{k_i} are the MACs generated using k_i keys from the endorsement list and ID_{k_i} are the key indices. The final report is then forwarded by each CH towards the sink through the same type of CHs. In the proposed scheme (CD-EFS) each cell has three CHs. Thus, each event results in three different copies of the same report.

5.2.5 En-Route filtering and Sink verification

When any intermediate CH receives a report, it checks for the common key used for creating any MAC in the report. If no such key is found by the CH in its $report\ verification$ key list, the report is dropped immediately. Otherwise the CH generates the MAC using the common secret key and compares it with the MAC included in the report. If both the MAC matches, the report is assumed to be correct, else it is dropped. Further, all three CHs in the forwarding cell co-ordinate with each other to identify whether at-least two copies of the report are found to be correct or not, if yes, then only correct copies of the report are forwarded to next hop, if not, all the copies of the report are dropped immediately.

Sink, on the other hand, performs *2-way authentication* for verifying each report. Sink starts the verification process of the report if it receives at-least two copies of the same report from two different types of CHs. Initially, sink verifies the freshness of all the M_x included in the report and checks whether all the participating sensor nodes are from the same cell or not. Next, the sink verifies all the MACs included with the report. If all the MACs are found to be correct, sink tries to recover M from M_x. This can be done by recovering M from any t correct M_x. More specifically, sink picks any t shares out of T shares and try to solves the t-variable linear equation (Equation 5.1) to get p_i, where $i = [0, t-1]$ and thus obtains M. If M is meaningful, the recovery is successful, otherwise sink tries other combinations of t shares to recover M. In the proposed scheme (CD-EFS), sink receives at-least two copies of each report, where each copy contains at-least 50% different M_x from other. Thus, in the worst case where sink only

receives two copies of a report, until no more than $((3/2)T - t)$ invalid M_x are present, the sink can always recover the original report.

5.3 Security Analysis of the Proposed scheme (CD-EFS)

In this section, we illustrate the security strengths of the proposed scheme (CD-EFS) in terms of data authenticity, expected filtering of false reports and data availability. But before that, we describe the simulation settings.

The proposed scheme (CD-EFS) is evaluated in a custom built simulator in Python language. The simulation parameters for the proposed scheme (CD-EFS) are given in Table 5.2. Further, there is a sink positioned in the center of the network. Typical parameter values for PCREF (Yang et al. 2015), LEDS (Ren et al. 2008) and SEF (Ye et al. 2005a) are also same as discussed in Table 5.2. Further for PCREF and SEF, polynomial sharing probability or keys sharing probability q is set to 0.2, which is a typical value in both the schemes for simulation results. For the simulations, compromised sensor nodes and CHs are chosen randomly in the network.

In terms of simulating various attacks we randomly select the sensor nodes and

Table 5.2: Simulation Parameters

Parameters	**Values**
Number of Sensor nodes	10005
Total number of Cells	667
Total sensor nodes in each cell	15
Cell Size	$50X50m^2$
Cluster Heads in each cell	3
Communication range of Sensor nodes	$25m$
Communication range of Cluster Heads	$50m$
Beam Width (w)	$250m$
Beam Length (l)	$500m$
(T, t)	5,4
Size of Report	36 Bytes
Size of MAC	4 Bytes
Energy Consumed to generate MAC	$15\mu J$(Per Byte)
Energy Consumed to verify MAC	$75\mu J$
Energy Consumed to Transmit/Receive	$16.25/12.5\mu J$(Per Byte)

cluster heads which are compromised. After the identification of compromised sensor nodes we simulate a report generation and forwarding phase. In this phase each cell in the network tries to create a report and sends the report towards the sink through intermediate CHs.

In terms of simulating report disruption attack in the network, all the compromised sensor nodes provide with dummy data/MACs at report generation phase. As report

generation in each cell is done by mutual collaboration between few sensor nodes, in simulation we try to identify cells where compromised sensor nodes can deny this report generation by sending false data/MACs. Specifically, we count the number of cells which can successfully generate correct reports by neglecting false data/MACs from the compromised sensor nodes when a given number of sensor nodes are compromised in the network.

In terms of simulating selective forwarding attack in the network, all compromised intermediate CHs purposely drop the incoming reports in place of forwarding them to the next hop. As report forwarding/ verification is done hop by hop in direction of sink, in simulation we try to identify how many genuine reports will be dropped by intermediate compromised CHs. Specifically, we count the number of cells whose correct reports are received by the sink when a given number of CHs are compromised in the network.

All the experiments, like choosing a fixed number of compromised sensor nodes/CHs in the network is repeated 50 times. This is mainly done to remove any ambiguity in the results.

5.3.1 Data authenticity

In the proposed scheme (CD-EFS), at the time of report generation, each CH chooses T tuples from $2T$ tuples in such a way that each CH has 50% different tuples. Further, atleast two CHs should agree with the tuple values and only then a report is generated and forwarded from the event cell. So, an adversary can inject a bogus/false report which can successfully by-pass En-Route filtering and sink verification only if:

1. Adversary is able to compromise at-least $(3/2)T$ sensor nodes in a particular cell. If total number of sensor nodes compromised in the network are X, then the probability that an adversary can successfully inject bogus report is given by

$$P_{Auth}(X) = \sum_{z=(3/2)T}^{n} \frac{\binom{n}{z}\binom{N-n}{X-z}}{\binom{N}{X}} \tag{5.2}$$

2. Adversary is able to compromise at-least one CH and T sensor nodes in a partic-

ular cell. If total number of sensor nodes compromised in the network are X, out of which x are CHs, then the probability that an adversary can successfully inject bogus report is given by

$$P_{Auth}(X) = \left(\sum_{z=T}^{n} \frac{\binom{n}{z}\binom{N-n}{(X-x)-z}}{\binom{N}{X-x}} \right) . \left(\sum_{i=1}^{3} \frac{\binom{3}{i}\binom{3C-3}{x-i}}{\binom{3C}{x}} \right) \tag{5.3}$$

3. Adversary is able to compromise at-least two CHs and T sensor nodes in a particular cell. If the number of sensor nodes compromised in the network are X, out of which x are CHs, then the probability that an adversary can successfully inject bogus report is given by

$$P_{Auth}(X) = \left(\sum_{z=T}^{n} \frac{\binom{n}{z}\binom{N-n}{(X-x)-z}}{\binom{N}{X-x}} \right) . \left(\sum_{i=2}^{3} \frac{\binom{3}{i}\binom{3C-3}{x-i}}{\binom{3C}{x}} \right) \tag{5.4}$$

The cases discussed above are the worst case scenarios where an adversary is able to inject bogus report from a particular cell, but in such scenario also remaining network is still un-compromised. The proposed scheme (CD-EFS) is better than schemes like LEDS (Ren et al. 2008) and PCREF (Yang et al. 2015) where adversary only requires t and T compromised sensor nodes in any particular cell respectively to inject bogus reports. Further, the proposed scheme (CD-EFS) is a major improvement over schemes such as IHA (Zhu et al. 2004), SEF (Ye et al. 2005a), and LBRS (Yang et al. 2005), in which single compromised sensor node can result in multiple gains.

Figure 5.2 provides the ratio of compromised cells vs total compromised sensor nodes in the network. So, lesser the number of compromised cells, better is the data authenticity of the scheme. In the figure we can observe that if the total number of compromised nodes in the network are 2000 (20% of total sensor nodes in the network), percentage of Secure Cells, Affected Cells and Compromised Cells in the network is 49%, 47% and 4% respectively. Overall, even when large number of sensor nodes/CHs are compromised in the network, only few cells are totally compromised.

Figure 5.3 presents the comparison among proposed scheme (CD-EFS), SEF, LEDS and PCREF for the probability of introducing bogus reports in the network when few

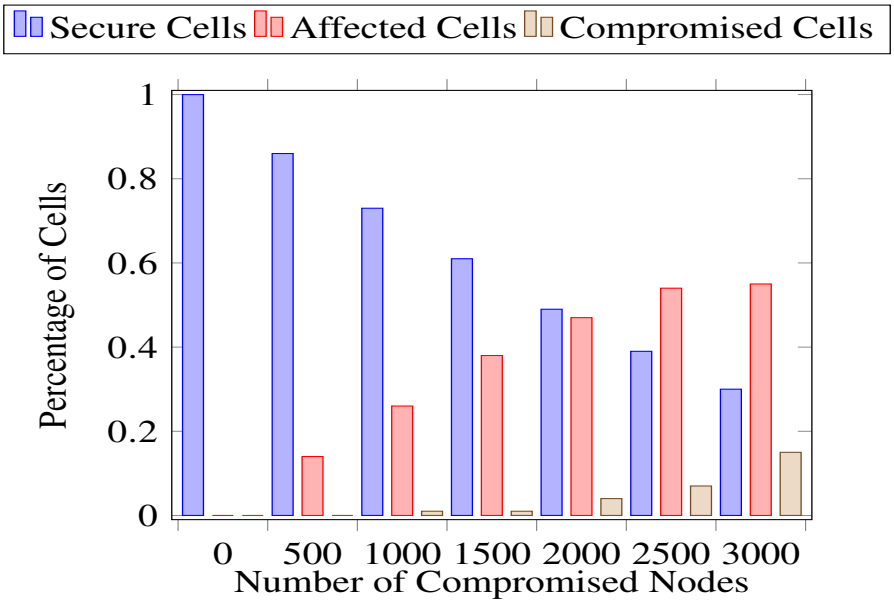


Figure 5.2: Resiliency vs Compromised Nodes in the proposed scheme (CD-EFS)

sensor nodes are compromised in the network. So, lesser the probability of introducing bogus reports, better is the data authenticity of the scheme. In the figure we can observe that SEF performs worst in the compared schemes, where probability of introducing bogus report becomes one when only 2000 sensor nodes are compromised in the network. LEDS and PCREF performs much better than SEF, where probability of introducing bogus report becomes one when 5000 sensor nodes are compromised in the network. The proposed scheme (CD-EFS), on the other hand, has 61% probability of introducing bogus reports when 5000 sensor nodes are compromised in the network. Finally, in the figure we can observe that the probability of introducing bogus reports in the proposed scheme (CD-EFS) is always less than SEF, PCREF and LEDS. Thus, the proposed scheme (CD-EFS) provides much higher data authenticity when compared with SEF, LEDS and PCREF.

5.3.2 Expected Filtering of Bogus Reports

The proposed scheme (CD-EFS) provides a deterministic En-Route filtering mechanism where reports are verified by all the intermediate hops. In the proposed scheme (CD-EFS), we assign combinatorial design based keys to CHs. So, if a CH is compromised, all its secret keys are exposed, affecting the other remaining network too. In the network

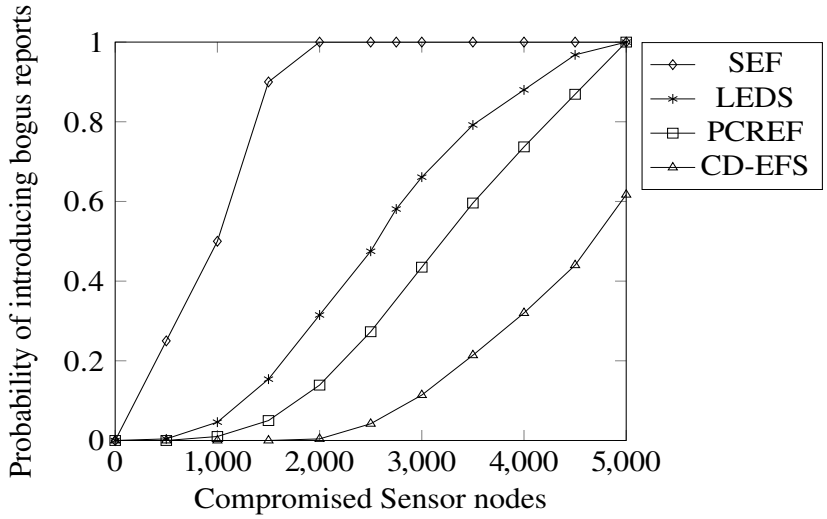


Figure 5.3: Data Authenticity in LEDS, PCREF, SEF and the proposed scheme (CD-EFS)

because of the compromised CHs, at any point of time, a particular CH_c^i can have its y keys exposed out of total Y keys. So, if an adversary wants to create a false report on behalf of CH_c^i, it has to forge other $(Y - y)$ MACs for successful report generation. To ensure this false report is dropped before it reaches the sink, one intermediate non-compromised CH is enough which has a key which was used to create any one of the $(Y - y)$ forged MACs.

In the network, if x sensor nodes are compromised and out of which X CHs are compromised, then the probability of filtering a false report generated from CH_c^i can be given by

$$P(X) = \sum_{x=0}^{H} (1 - P_{CH}^x) \frac{(Y - y)}{Y} \tag{5.5}$$

where P_{CH}^x is the probability of any particular CH_x^i being compromised and H represents total hops between CH_c^i and the sink. Further, P_{CH}^x can be given as

$$P_{CH}^x = \frac{\binom{C}{X/3}\binom{3C-C}{X-(X/3)}}{\binom{3C}{X}} \tag{5.6}$$

where C is total CHs of a particular type and $X/3$ represents compromised CHs of a particular type. But in the proposed scheme (CD-EFS), 3 copies of each report with different MACs are forwarded toward the sink. So, to completely drop a false report

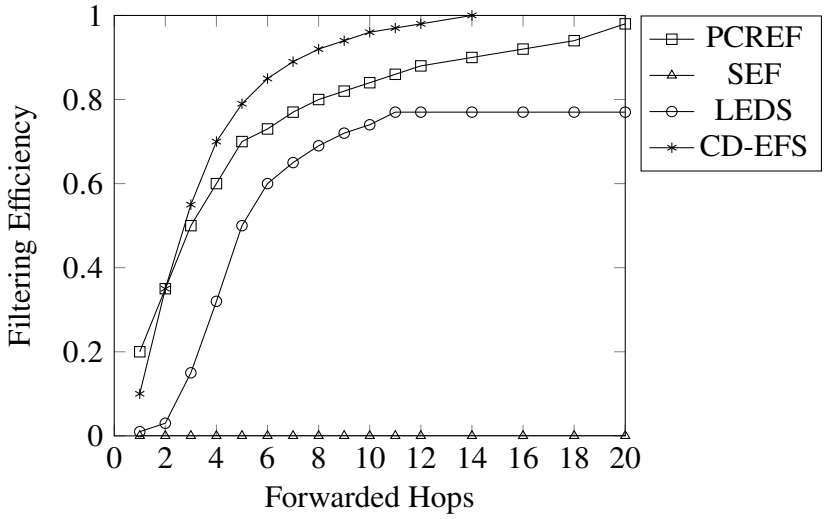


Figure 5.4: Filtering Efficiency vs Forwarded Hops in PCREF, LEDS, SEF and the proposed scheme (CD-EFS)

from the network, at-least two copies of the same report must be dropped. Thus, the probability of completely dropping a false report is given by Equation 5.7.

$$P_{filtering} = \binom{3}{2} P(X)^2 \tag{5.7}$$

Figure 5.4 provides the comparison of filtering efficiency vs hops traveled in PCREF, LEDS, SEF and the proposed scheme (CD-EFS). So, higher the filtering efficiency, better is the scheme. In the figure we can observe that SEF, LEDS and PCREF have filtering efficiency of 0%, 77% and 90%, respectively, when we fix the forwarded hops for the reports to 14. But on the other hand, the proposed scheme (CD-EFS) has 100% filtering efficiency when we fix the forwarded hops for the reports to 14. Thus, from the figure it is evident that the proposed scheme (CD-EFS) promises high filtering efficiency in the least number of hops. The filtering efficiencies of the schemes such as IHA (Zhu et al. 2004), LBRS (Yang et al. 2005) are always poorer than PCREF as explained in Yang et al. (2015), thus the proposed scheme (CD-EFS) also has better filtering efficiencies than IHA, LBRS.

Experimental results of expected filtering position of the false report in the proposed scheme (CD-EFS), LEDS (Ren et al. 2008) and PCREF (Yang et al. 2015) are given in

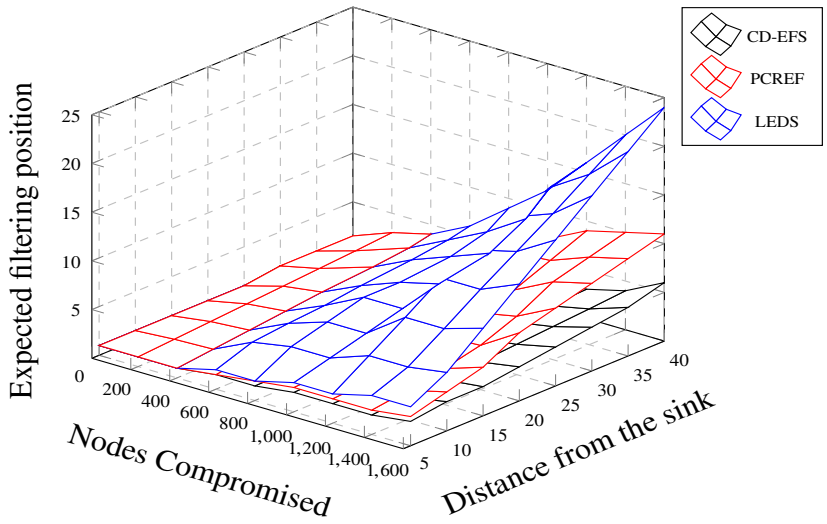


Figure 5.5: Average forwarded hops for false reports in LEDS, PCREF and the proposed scheme (CD-EFS).

Figure 5.5. So, lesser the expected filtering position, better is the filtering efficiency of the scheme. In the figure, we can observe that the proposed scheme (CD-EFS) filters the false reports in much fewer hops when compared with LEDS and PCREF. Moreover, in the figure, we can observe that the proposed scheme (CD-EFS) can filter false reports in 6 hops on average, which is a big improvement from 24 hops in LEDS and 11 hops in PCREF, when total nodes compromised in the network are 1600 and distance from the sink is 40. Finally, in the figure we can observe that the expected filtering position in the proposed scheme (CD-EFS) is always less than PCREF and SEF. Thus, the proposed scheme has better filtering efficiency than PCREF and SEF.

5.3.3 Data Availability

Data availability in WSNs can severely be affected by two types of attacks namely, *Report Disruption* attack and *Selective Forwarding* attack. In report disruption attack, compromised sensor nodes can intentionally send wrong tuples to CHs or compromised CHs can attach wrong MACs to the final report. Thus, correct reports are either dropped by intermediate CHs or by the sink because of wrong data in the report. On the contrary, compromised intermediate CHs can drop all the reports passing through them; this is termed as selective forwarding attack. Effect of both attacks on the pro-

posed scheme (CD-EFS) is discussed below:

a) Report Disruption attack

In the proposed scheme (CD-EFS), participating sensor nodes create the shares for the report and CHs generate MACs for the report. Thus, both compromised sensor nodes and compromised CHs can participate in report disruption attack. Compromised participating sensor nodes can send wrong M_x with the tuple to the CH, so that sink cannot recover the original report. Further, compromised CHs can attach wrong MACs with the reports, thus such reports are either dropped by intermediate CHs or sink. In the proposed scheme (CD-EFS), each participating sensor node only contributes one share of the report follows (t,T) threshold LSSS (Ren et al. 2008), thus sink can always recover the correct report if it gets at-least t correct tuples. Thus, correct reports are received and recovered by sink until:

- No CH is compromised and less then $2T - t$ participating sensor nodes are compromised in any cell. In the proposed scheme (CD-EFS), $2T$ sensor nodes participate in report generation and if no CHs are compromised in the event cell, the sink will receive $2T$ tuples. Thus, the sink can use any t correct tuples from $2T$ tuples to recover the correct report. Given the number of sensor nodes compromised in the network is X, the security strength of the proposed scheme (CD-EFS) in such a case can be given as:

$$P_{Avail}(X) = \sum_{z=0}^{2T-t} \frac{\binom{2T}{z}\binom{N-2T}{X-z}}{\binom{N}{X}} \tag{5.8}$$

- One CH is compromised and less than $(3/2)T - t$ sensor nodes are compromised in any cell. In this case the sink will receive only two copies of the report because the third copy of the report will be dropped either by intermediate CHs or by sink due to in-correct MACs. Thus, the sink will only receive two copies of the report with a total $(3/2)T$ tuples and sink can use any t correct tuples from them to recover the correct report. Given the number of sensor nodes compromised in the network are X out of which x CHs are compromised, security strength of the

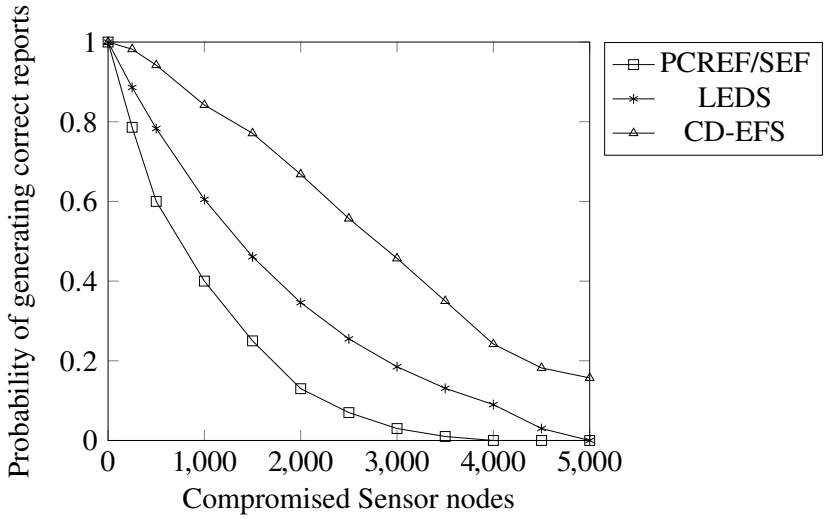


Figure 5.6: Data Availability under Report Disruption attack in LEDS, PCREF, SEF and the proposed scheme (CD-EFS).

proposed scheme (CD-EFS) in such case can be given as:

$$P_{Avail}(X) = \frac{\binom{3}{1}\binom{3C-3}{x-1}}{\binom{3C}{x}}\left(\sum_{z=0}^{(3/2)T-t}\frac{\binom{(3/2)T}{z}\binom{N-(3/2)T}{(X-x)-z}}{\binom{N}{X-x}}\right) \tag{5.9}$$

So, the above discussed cases formalize the probability of receiving/recovering correct reports by sink. In the above cases, we discussed that there is high probability of recovering the original reports in the proposed scheme even in presence of multiple compromised sensor nodes/CHs in the network. Thus, the proposed scheme is more resilient than LEDS (Ren et al. 2008), where sink can recover the correct report only if compromised participating sensor nodes are less then $T - t$. Further, the proposed scheme (CD-EFS) is better than PCREF (Yang et al. 2015), SEF (Ye et al. 2005a), IHA (Zhu et al. 2004) and LBRS (Yang et al. 2005) where sink can recover the correct report only if all participating sensor nodes are non-compromised.

Figure 5.6 provides the probability of generating correct reports in the proposed scheme (CD-EFS) with SEF, LEDS and PCREF. So, greater the probability of generating correct reports, better is the data availability in the scheme. Specifically, in the figure we can observe that the probability of generating correct reports in PCREF/SEF, LEDS and the proposed scheme (CD-EFS) is 3%, 19% and 50%, respectively, when 3000

sensor nodes are compromised in the network. Finally, in the figure we can observe that the proposed scheme always has higher probability of generating correct reports when compared with PCREF, SEF and LEDS. Thus, we can conclude that the proposed scheme (CD-EFS) provides much better data availability than SEF, LEDS and PCREF.

b) Selective Forwarding attack

In the proposed scheme (CD-EFS), three copies of the same report are forwarded with different MACs from the event cell. Further, the sink can accept the reports from the event cell if it receives at-least two copies of the same report through different types of CHs. So, to purposely drop a report and limit the sink from obtaining the report, at-least two copies of the same report should be dropped before they reach the sink. To drop two copies of the same report, at-least two CHs of different types must be compromised on the path from event cell to the sink. Precisely, if a report originating from the event cell has to travel H hops to reach the sink and in the network x sensor nodes are compromised and out of which X are CHs, then the probability that at-least one intermediate CH of a particular type is compromised can be given by Equation 5.10.

$$P_{com}(X) = \sum_{z=1}^{H} \frac{\binom{H}{z}\binom{C-H}{(X/3)-z}}{\binom{C}{X/3}} \tag{5.10}$$

Further, the dropping probability of a correct report in the proposed scheme (CD-EFS) can be given by Equation 5.11.

$$P_{select}(X) = \left\{\binom{3}{2}(P_{com}(X))^2(1 - P_{com}(X))\right\} + (P_{com}(X))^3 \tag{5.11}$$

Figure 5.7 presents the experimental comparison of the dropping probability of genuine reports in the proposed scheme (CD-EFS), SEF (Ye et al. 2005a), PCREF (Yang et al. 2015) and LEDS (Ren et al. 2008). So, lesser the dropping probability of genuine reports, better is the data availability in the scheme. Existing schemes such as SEF, PCREF, LBRS, IHA do-not adopt any preventive measures for selective forwarding attack because of which these schemes are highly prone to selective forwarding attack. In the figure we can observe that SEF/PCREF has highest probability of dropping gen-

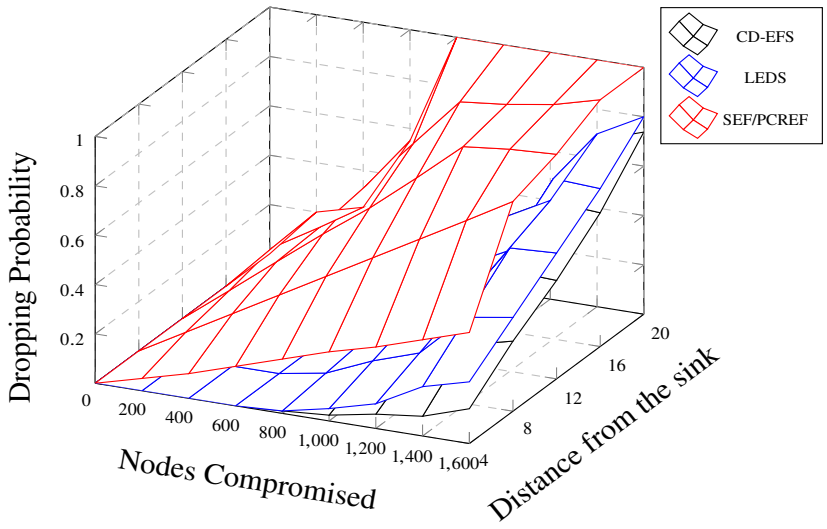


Figure 5.7: Data Availability under Selective Forwarding attack for SEF/PCREF, LEDS and the proposed scheme (CD-EFS).

uine reports because of the selective forwarding attack. In LEDS at the time of report forwarding, each report is broadcasted to all the sensor nodes in an intermediate cell and because of which LEDS is highly resilient to selective forwarding attack. In figure we can observe that LEDS lowers the dropping probability of genuine reports. Further, we can observe from the Figure 5.7 that the proposed scheme (CD-EFS) outperforms LEDS even without broadcasting reports in intermediate cells. Finally, in the figure we can observe that the proposed scheme always has lower probability of dropping genuine reports when compared with PCREF, SEF and LEDS. Thus, we can conclude that the proposed scheme (CD-EFS) provides much better data availability than SEF, LEDS and PCREF.

5.4 Performance Evaluation of the Proposed scheme (CD-EFS)

Now, we discuss the associated storage overhead, computation and communication overhead, and energy requirements for the proposed scheme (CD-EFS).

5.4.1 Key Storage Overhead

Each sensor node in the proposed scheme (CD-EFS) stores 4 secret keys, three for communicating with CHs in the same cell and one for communicating with the sink.

On the other hand, CHs are assigned keys based on a combinatorial design where each CH is assigned $k + 1$ keys. Then, we deploy a beam model (Section 5.2.2) to further decrease the number of keys stored in CHs by limiting secret keys shared only within the upstream and downstream region of a particular CH. In the proposed scheme, if $C = 121$, then an average of 9 keys are stored in each CH. In LEDS (Ren et al. 2008), each sensor node is assigned $\{(T+1)(T+2)/2\}+5$ keys. Thus, the total number of keys stored by sensor nodes/ CHs in LEDS is 26, when $T = 5$. In PCREF (Yang et al. 2015), each sensor node is assigned large number of polynomials and secret keys which can be given by $(64.n_k + 16.n_c)$, where n_k is total keys and n_c is the number of coefficients. Thus, a large number of polynomials and secret keys are stored by each sensor node and CH in PCREF. So, the proposed scheme (CD-EFS) has less storage overhead both in sensor nodes and CHs when compared with LEDS and PCREF.

5.4.2 Communication and Computation Overhead

In any en-route filtering scheme, Communication overhead is mainly because of initial key exchange and because of MACs which are included with the reports. In the proposed scheme (CD-EFS), we deploy combinatorial design based secret keys, which makes sure that any two CHs always share a secret key. Further, for shared key discovery, only a single message is sent by each CH in the network which is much efficient than 3-way handshake used in existing schemes like LBRS (Yang et al. 2005), LEDS (Ren et al. 2008), IHA (Zhu et al. 2004). For En-Route filtering in the proposed scheme (CD-EFS), each report is attached with multiple MACs. Each CH has to create MACs with the keys present in its *Report Endorsement* key list. For example, if total cells in the network are 121, then on an average each CH has 7 keys in the *Report Endorsement* key list. Thus, on an average 7 MACs are attached with each report. In such a case, the communication overhead for the proposed scheme (CD-EFS) is 28 bytes, considering the size of MAC as 4 bytes. In LEDS, $T + 1$ MACs are included with each report and thus total communication overhead is 24 bytes when $T = 5$. In PCREF (Yang et al. 2015) for a network of the same size, communication overhead is 40 bytes (explained in Yang et al. (2015)). Thus, we can observe that the communica-

tion overhead of the proposed scheme (CD-EFS) is under considerable limits.

In the proposed scheme (CD-EFS), computation overhead for pair-wise secret keys assignment is very limited because of the use of combinatorial design based keys. Thus, in the proposed scheme (CD-EFS), the computation overhead is mainly due to MACs generation/verification. In the proposed scheme (CD-EFS) sensor nodes only submit the initial report to the CHs and CHs create and attach the MACs with the report. Moreover, the reports are forwarded only through CHs. Thus, MAC verification is done only by CHs. This helps in saving a lot of computation overhead from energy deprived sensor nodes.

5.4.3 Energy Requirements

False reports in the network leads to extra energy consumption for forwarding such reports. Further, compromised sensor nodes can intentionally drop legitimate reports, leading to further energy wastage for sending same reports again. In this subsection, we identify the energy requirements for the proposed scheme (CD-EFS). If H represents average hops from source to sink, e represents energy required by any intermediate sensor node for receiving and sending the report to next hop and P represents the length of the report, then the energy requirements for a report where no En-Route filtering is implemented can be given by Equation 5.12.

$$E_{without} = H.P.e \tag{5.12}$$

In this case all the reports (correct or forged) will travel H hops. When the proposed En-Route filtering scheme is implemented, correct reports travel all H hops and false reports are dropped in maximum h hops. Thus, energy requirement in such case can be given by,

$$E_{with\ En-Route} = \{H(1-Z) + hZ\}.(P + L_{MCs} + L_{IDs}).e \tag{5.13}$$

where Z is the percentage of false reports and L_{MCs}, L_{ID} represents the length of MACs and key indices respectively. This is the energy requirement for a report without selective forwarding attack. If we take into account the effect of selective forwarding attack,

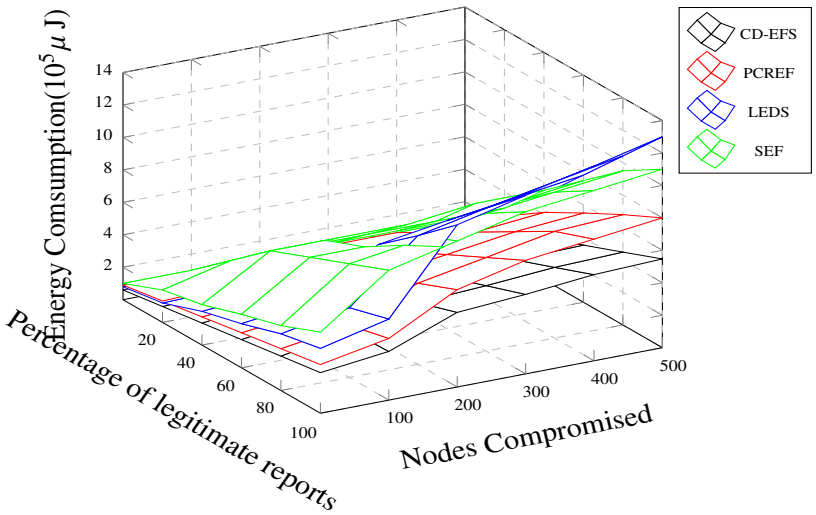


Figure 5.8: Energy Comparison for LEDS, PCREF, SEF and the proposed scheme (CD-EFS).

few correct reports can also be dropped before reaching the sink and such reports need to be sent again by the source. Thus finally, energy requirements for the proposed scheme (CD-EFS) can be given by,

$$E_{with\ En-Route} = \{H(1-Z) + h'(1-Z)P_{select}(X) + hZ\}.(P + L_{MCs} + L_{IDs}).e \tag{5.14}$$

where $P_{select}(X)$ represents the probability of dropping a report when X CHs are compromised in the network and h' represents average hops each correct report travels before being dropped by intermediate compromised CH. Moreover, energy consumption for computation is much less than communication. Thus, we only consider the energy consumed by communication of reports while calculating energy requirements for the network.

Experimental results for energy requirements for the whole network in the proposed scheme (CD-EFS), SEF (Ye et al. 2005a), LEDS (Ren et al. 2008) and PCREF (Yang et al. 2015) are given in Figure 5.8. So, lesser the energy consumption, better is the scheme. In the figure, we can observe that LEDS consumes maximum energy among the four compared schemes mainly because of poor filtering efficiency and because of the broadcasting nature of the forwarded reports. SEF performs better than LEDS be-

cause it do-not broadcast reports in each cell. PCREF performs better than LEDS and SEF because of improved filtering efficiency. But energy requirements in PCREF rises significantly with increased compromised sensor nodes. This is because of no preventive measures adopted by PCREF to restrict selective forwarding attack in the network, thus many correct reports can be dropped intentionally by intermediate sensor nodes and such reports are needed to be sent again by the source. The proposed scheme (CD-EFS), on the other hand, has very high filtering efficiency and is highly resilient to report disruption and selective forwarding attacks, resulting in very low energy requirements in the network. Finally, in the figure we can observe that the proposed scheme always has lower energy consumption when compared with PCREF, SEF and LEDS.

5.5 Concluding Remarks

In this chapter, we proposed a novel deterministic en-route filtering scheme based on combinatorial design. In the proposed scheme, the secret keys to CHs are assigned based on combinatorial design. We proposed a novel beam model to further reduce key storage overhead in the network. We observed that the proposed scheme promised high filtering efficiency in the network. Further, we proposed novel report generation and novel en-route filtering/sink verification methods. In the proposed scheme, each cell had three CHs and report forwarding/verification is only done by CHs. This helped in reducing the effect of selective forwarding attack while maintaining desired security in the network. It further reduced the energy requirements of the network. In the proposed scheme, three copies of each report with different endorsements were forwarded by the event cell towards the sink. This considerably improved data authenticity in the proposed scheme.

CHAPTER 6

A Partial Key Pre-Distribution Based En-Route Filtering Of False Reports

Compromised sensor nodes can be used to inject false/bogus reports in Wireless Sensor Networks (WSNs). This can cause the sink to take wrong decisions. En-Route filtering is a method to detect and filter false reports from WSNs. Most of the existing En-Route filtering schemes use probabilistic approaches to filter false reports from the network, where filtering of the false reports is based on a fixed probability. Thus, false reports can travel multiple hops before being dropped. Furthermore, few deterministic En-Route filtering schemes have also been proposed in the literature, but they use predetermined paths to forward the reports.

In this chapter, we propose a new En-Route filtering scheme (CD-PEFS) based on partial key pre-distribution using combinatorial design. In the proposed scheme (CD-PEFS) we consider a distributed network consisting of equal-sized cells. Each cell has two types of sensor nodes namely, normal sensor nodes and Cluster Heads (CHs). Normal sensor nodes use pairwise keys for communication with CHs and the sink. Assignment of keys to normal sensor nodes is done based on their location, which ensures that the scheme is unaffected by T-threshold limitation (Kumar and Pais 2017). CHs on the other hand, are assigned combinatorial design based keys to ensure secure inter-cell communication in the network. Because of the deterministic nature of combinatorial design based keys, filtering efficiency in the proposed scheme (CD-PEFS) is very ex-

cellent. Moreover, the use of combinatorial design based keys in the proposed scheme (CD-PEFS) reduces the key storage overhead in the CHs. In the proposed scheme (CD-PEFS), multiple copies of the same report with different endorsements are forwarded in the network which helps to maintain very high data authenticity. In the proposed scheme (CD-PEFS), reports are forwarded and verified only by CHs. This helps in limiting the effect of selective forwarding attack in the network. Further, it also reduces the energy requirements in the network, while maintaining desired security in the network. Finally, the proposed scheme (CD-PEFS) does not require any shared key discovery phase which helps to maintain very low associated communication and computation overhead in the network.

The remaining chapter is organized as follows: Section 6.1 provides the associated system and threat model for the proposed scheme (CD-PEFS). Proposed scheme (CD-PEFS) is presented in Section 6.2 and the effect of partial key pre-distribution is discussed in Section 6.3. Security analysis of the proposed scheme (CD-PEFS) is provided in Section 6.4, followed by performance evaluation in Section 6.5. Finally, we conclude the chapter in Section 6.6.

Notations

For convenience, we discuss all the notations (refer Table 6.1) used in this chapter.

6.1 System and Threat Model

System Model

We consider a distributed WSN consisting of N sensor nodes and a sink which collects/verifies all the data in the network. The entire network is divided into cells and each cell has the same number of sensor nodes, say n. In the network, we have two types of sensor nodes namely, CHs and normal sensor nodes. CHs have higher computation, communication and storage capabilities than normal sensor nodes. Each cell has a unique cell-ID which is used for identification of cells in the network. Moreover, each cell has three CHs, which are enough to maintain the desired connectivity and resiliency in the network. Further, CHs are of three different types (assigned keys using different key pools) namely, $Type$ 1, $Type$ 2, $Type$ 3 and one of each type of CH is placed in

Table 6.1: Notations

N	Number of sensor nodes in the network.
C	Total number of cells in the network.
n	Number of sensor nodes in a cell.
C_i	i^{th} cell.
T	Number of endorsements to be included with each report.
t	Minimum number of correct endorsements required to validate a report.
P	A large prime.
K_m,P_m^1, P_m^2, P_m^3	Four master secret keys.
H()	Hash Function.
x	Any sensor node.
x_{loc}	Location of sensor node x.
C_{loc}	Location of Cluster Head.
(x_c, y_c)	Location of center of home cell.
P_{sm}^i	Sub-master keys derived by a sensor node to communicate with i^{th} Cluster Head.
P_x^i	Secret key derived by sensor node x to communicate with i^{th} Cluster Head.
M	Event report.
M_x	Unique share of event report generated by sensor node x.
M_{encr}	Encrypted report.
$k+1$	Number of keys stored in Cluster Heads when all Cluster Heads are assigned keys.
$k'+1$	Number of keys stored in Cluster Heads when $3/4^{th}$ Cluster Heads of each type are assigned keys.
X_i	Secret key between i^{th} Cluster Head and sink.
MAC_i	Message authentication code generated using combinatorial design based keys.
MAC_{X_1}, MAC_{X_2}	Message authentication code generated using pair wise keys.

every cell. Each CH can communicate with CHs of the same type in the whole network and can communicate with CHs of other types in the same cell. Each event in the network is detected by multiple sensor nodes, which forward the event to the CHs. CHs create the final report and forward the final report to the sink via multi-hop path. Report forwarding and verification are done only by CHs in the network. CHs can find the location of sensor nodes within the same cell. This can be done using any localization scheme (He et al. 2003; Patwari et al. 2005). Finally, all sensor nodes and CHs have unique IDs.

Threat Model

We assume CHs and sensor nodes can be compromised by an adversary. When an adversary compromises a sensor node, all the information stored in the sensor node is revealed to the adversary. Using the obtained information an adversary can inject false reports and drop/alter genuine reports. However, the sink cannot be compromised by

an adversary.

6.2 Proposed scheme (CD-PEFS)

In this section, we discuss the proposed scheme (CD-PEFS) in depth including initialization of sensor nodes, initialization of CHs, report generation and En-Route filtering in the network.

Deployment - In the deployment phase, network controller selects the value of parameters T and t (Shamir 1979), where T denotes the number of endorsements required for generating a valid report and t denotes the number of correct endorsements required for successful validation of a report. The network controller also generates four master secret keys (K_m, P_m^1, P_m^2, P_m^3), a large prime number P and a hash function $H()$.

6.2.1 Initialization of Sensor Nodes

Network controller assigns four master secret keys (K_m, P_m^1, P_m^2, P_m^3) to each sensor nodes. Further, network controller assigns P, $H()$, location of sensor node (x_{loc}) and center location of the home cell (x_c, y_c) to each sensor node. Each node then follows Algorithm 6.1 to compute secret keys to secure the communication with the sink and all three CHs in the home cell.

Algorithm 6.1: Generation Of Secret Keys

Input: $K_m, P_m^i, x_{loc}, (x_c, y_c)$ where, $1 \leq i \leq 3$
Result: Secret Keys (K_x, P_{sm}^i, P_x^i)

```
for each sensor node x in network do
    K_x = H(K_m|x_loc) ;
    for i= 1 to 3 do
        P^i_sm = H(P^i_m|(x_c, y_c)) ;
        P^i_x = H(P^i_sm|x_loc);
    end
end
```

First, each sensor node x computes the secret key K_x by concatenating the master secret key K_m with its location (x_{loc}). Then, it computes a sub-master key (P_{sm}^i) which is derived by concatenating P_m^i with the home cell's location. Every node then computes

a secret key (P_x^i) using the sub-master secret key and its location. P_x^i secret key is used for securing the communication with respective CHs. Every sensor node in the network executes the above algorithm and deletes the master secret keys stored in them thereafter.

6.2.2 Initialization of Cluster Heads

During initialization, network controller allocates a master key K_m and one P_m^i (such that $Type\ 1, Type\ 2, Type\ 3$ CHs are assigned P_m^1, P_m^2, P_m^3 respectively) to all CHs in a particular cell. Further, CHs are also assigned T and (x_c, y_c). Each CH uses the master key K_m to obtain a unique secret key X_i as:

$$X_i = H(K_m \mid C_{loc}) \tag{6.1}$$

where C_{loc} is the location of the CH. This unique secret key is used by the CH to securely communicate with the sink. Further, each CH computes the sub-master secret key using P_m^i as:

$$P_{sm}^i = H(P_m^i | (x_c, y_c)) \tag{6.2}$$

where (x_c, y_c) is the location of home cell. This is used by the CH to communicate with all other sensor nodes in the home cell.

Further, for secure validation and authentication of the reports, each CH is assigned keys using combinatorial design which is the same as explained as in CD-KPD (refer Section 3.2.1).

6.2.3 Reduction of keys stored in Cluster Heads

For reduction in keys stored by CHs, we adopt the same method as proposed in CD-PKPD (Section 3.2.5). In the proposed method we assign key-sets to exactly $3/4^{th}$ of the total CHs of each type in the network.

6.2.4 Report Generation

On the occurrence of an event, any $2T$ sensor nodes that detect the event will agree on the report as explained in Ye et al. (2005b). After agreeing on the report, all participating sensor nodes derive their unique secret share M_x of the report M with the help of predefined threshold (t, T) LSSS (Ren et al. 2008) as given in Equation 6.3.

$$M_x = \sum_{0 \leq i \leq t} p_i X_x mod(P) \tag{6.3}$$

M_x is derived using a univariate polynomial of degree $(t-1)$ over finite field $GF(P)$ ($GF(.)$ is a finite $Galois\ field$ (Dickson 2003)) using X_x and p_i, where p_i is the full partition of M such that $0 \leq i < t$. Both t and P are preloaded parameters. Each share is uniquely generated by sensor node and therefore can be used by the sink as an endorsement. Next, each of the $2T$ participating sensor nodes will encrypt the report M using P_x^i as

$$M_{encr}^i = E_{P_x^i}(M) \tag{6.4}$$

Each sensor node sends the tuple $\{M_x, x, M_{encr}^i\}$ to the CHs where x is the ID of the sensor node. Now all three CHs coordinate with each other to choose T tuples from $2T$ tuples such that each CH chooses at-least 50% different tuples when compared with the other two CHs. CHs ensure this by sharing the IDs of participating sensor nodes. This constraint in the report generation helps in preserving the data authenticity when an adversary tries to fabricate a report.

Upon receiving the tuples from T sensor nodes satisfying above constraints, the CHs check for the freshness of the report, the location of the event and the location of the sensor nodes sending the tuple. All the sensor nodes involved in tuple generation must be from the same cell, else the CH drops the report. CH then decrypts M_{encr}^i and checks whether all the M it received are same or not. For the successful generation of reports, the M value must be agreed upon by at-least two CHs, i.e., at-least two CHs must receive the same M values after decryption. Initially, CH generates a Message Authentication Code (MAC) for the message M using the unique secret key it shares with the sink (X_i) as $MAC_{X_i}(M)$ (where i represents the IDs of CHs in a cell) and sends

it to other two CHs in the same cell. After receiving MAC from at-least one of the other CHs, each CH start preparing the final report if it is participating in report generation. A particular CH would participate in report generation only if it was assigned key-set at time of initialization of CHs. Then, participating CHs generate MAC (MAC_i) for the report M with all the $k'+1$ keys. Finally, CHs generate the final report of the form $\{M_1, M_2, \cdots, M_T, ID_1, ID_2 \cdots, ID_T, MAC_1, MAC_2 \cdots, MAC_{k'+1}, ID_{k_1}, ID_{k_2} \cdots, ID_{k_{k'+1}}, MAC_{X_1}, MAC_{X_2}\}$, where $\{M_1, M_2, \cdots, M_T\}$ represents the secret shares, $\{ID_1, ID_2 \cdots, ID_T\}$ represents the IDs of participating sensor nodes, $\{MAC_1, MAC_2 \cdots, MAC_{k'+1}\}$ represents the MACs generated using all the keys from the key-set, ID_{k_i} represents IDs of keys used for creating MACs and MAC_{X_1}, MAC_{X_2} represents MACs sent by other two CHs in the same cell.

Each event can result in either one, two or three copies of each report, depending on the number of CHs participating in the generation of the report. All the copies of the report are finally forwarded towards the sink through the same type of intermediate CHs.

6.2.5 Shared Key Discovery

The proposed scheme (CD-PEFS) uses a symmetric design for generation of key-sets which ensures that any pair of the key-sets has at-least one key in common. Moreover, the forwarding report consists of the key indices used by the CHs for generation of MACs. Therefore, the shared key between any two CHs can be obtained by comparing the key indices present in the report with those stored in the CHs. Thus, the proposed scheme (CD-PEFS) does not require any shared key discovery phase thereby reducing the overall communication overhead in the network.

6.2.6 En-Route Filtering and Sink Verification

In the En-Route filtering phase, each receiving/forwarding CH checks the authenticity of each report if it is assigned key-set at the time of initialization. If the CH does not store the key-set, then it will simply forward the report to next hop. Otherwise, it checks for the common key that has been used for generation of a MAC attached

with the report. If no common key is found, the report is dropped immediately. After identification of common key, CH verifies the MAC sent with the report by regenerating the MAC using the common key. If a mismatch is found, the report is dropped, else the report is forwarded to the next hop.

After receiving the report, the sink verifies the report. Sink performs a two-way authentication for each report it receives. Firstly, it checks the report for the freshness of shares (M_x) and whether all the participating sensor nodes are from the same cell or not. Then, sink verifies all the MACs (MAC_i) included with the report. If all the MACs (MAC_i) are found to be correct, sink further checks the correctness of MAC_{X_1}, MAC_{X_2}. If at-least one of the MACs (MAC_{X_1} or MAC_{X_2}) is found to be correct, sink tries to recover M from M_x. This can be done by recovering M from any t correct M_x included with the report. More specifically, sink picks any t shares out of all the shares included with the report, then sink solves a t-variable linear equation to get p_i, where $i = [0, t-1]$ in Equation 6.3 and thus obtains M. If M is meaningful, the recovery is successful, otherwise sink tries other combinations of t shares to recover M.

6.3 Effect of partial key pre-distribution

In this section, we will discuss the effect of partial key pre-distribution on network throughput, latency, network connectivity, key storage overhead and data filtering.

6.3.1 Network Connectivity

In the proposed scheme (CD-PEFS), we have three CHs in each cell. Thus, if total cells in the network are C, then there are a total of $3C$ CHs in the network. Further, these CHs create a backbone for data forwarding in the network. At the time of key pre-distribution, we assign keys to only $3/4^{th}$ of the total CHs of each type. This limits the En-Route filtering capability to only $3/4^{th}$ of the total CHs of each type. Thus, CHs who have not been assigned keys cannot participate in data authentication, but they still participate in report forwarding. So, partial key pre-distribution in the network reduces the filtering efficiency in the network, but it certainly does not effect the connectivity in the network. Further, if connectivity in the network is not effected, throughput and

latency in the network will also be un-effected. Effect of partial key pre-distribution on key storage overhead and filtering efficiency is discussed in next subsections. Thus, the partial key pre-distribution in the proposed scheme (CD-PEFS) do not effect connectivity, throughput and latency of the network.

6.3.2 Key Storage Overhead

In CD-KPD (refer Section 3.2.1), we assign combinatorial design based keys to the CHs, where $k + 1$ keys are assigned to each CH. But in the proposed scheme (CD-PEFS) we assign secret keys to only $3/4^{th}$ of the total CHs of each type. Thus, now $3/4C \leq k'^2 + k' + 1$ and in this case $k' + 1$ keys are assigned to the CHs. As the number of CHs for secret keys assignment is reduced in the proposed scheme (CD-PEFS), in all the network sizes $k' \leq k$.

Table 6.2 shows the comparison of keys stored by each CH in different network sizes. So, lesser the keys stored, lower is the associated key storage overhead in the scheme. In the table, we can observe that the number of keys stored by each CH reduces with the partial key pre-distribution. In the figure we can observe that the value of k and k' is 23 and 17 respectively, when total CHs in the network are 400. Further, it can be observed in the table that the partial key pre-distribution always results in less key storage overhead than the key pre-distribution done in the whole network, irrespective of the network size. Finally, the reduction of keys stored by each CH in partial key pre-distribution is more significant in larger network sizes.

Table 6.2: Key stored by Cluster Heads when key pre-distribution done in Partial ($(3/4)^{th}$ Cluster Heads) and Whole network

C	k(Whole Network)	k′($(3/4)^{th}$ Cluster Heads)
75	11	7
150	13	11
200	17	13
350	19	17
400	23	17
600	29	23
900	31	29
1000	37	29

6.3.3 Filtering Efficiency

In the proposed scheme (CD-PEFS) we have $3C$ CHs in the network from which we methodically select $3/4^{th}$ CHs to assign secret keys for En-Route filtering. As now keys are assigned to fewer CHs, the associations between CHs are reduced. This reduces the filtering efficiency of the network. But in the proposed scheme (CD-PEFS), we choose $3/4^{th}$ CHs of each type in such a manner that connectivity between CHs which perform data authentication can be maximized. This ensures high En-Route filtering efficiency in the network even if only $3/4^{th}$ of CHs are assigned secret keys.

Figure 6.1 shows the comparison of filtering efficiency vs forwarded hops for key pre-distribution in whole and partial ($3/4^{th}$ of CHs) network. So, higher the filtering efficiency, better is the scheme. In the figure, we can observe that filtering efficiency certainly reduces in the partial key pre-distribution. Specifically, filtering efficiency of false reports reaches 100% in maximum 14 hops when key pre-distribution is done in the whole network. But, the filtering efficiency reaches 100% in 16 hops when key pre-distribution is done in partial ($3/4^{th}$ of CHs) network. But still, the filtering efficiency in partial key pre-distribution is still significantly much better than existing schemes Ren et al. (2008) and Yang et al. (2015). Comparison of the filtering efficiency of the proposed scheme (CD-PEFS) with existing schemes is done in Section 6.4.

6.4 Analysis

In this section, we discuss the security analysis of the proposed scheme (CD-PEFS) concerning data authenticity, filtering efficiency and data availability. But before that, we describe the simulation settings.

The proposed scheme (CD-PEFS) is evaluated in a custom built simulator in Python language. The simulation parameters for the proposed scheme (CD-PEFS) are given in Table 6.3. Further, there is a sink positioned in the center of the network. Typical parameter values for PCREF (Yang et al. 2015), LEDS (Ren et al. 2008) and SEF (Ye et al. 2005a) are also same as discussed in Table 6.3. Further for PCREF and SEF, polynomial sharing probability or keys sharing probability q is set to 0.2, which is a typical value in both the schemes for simulation results. For the simulations, compromised

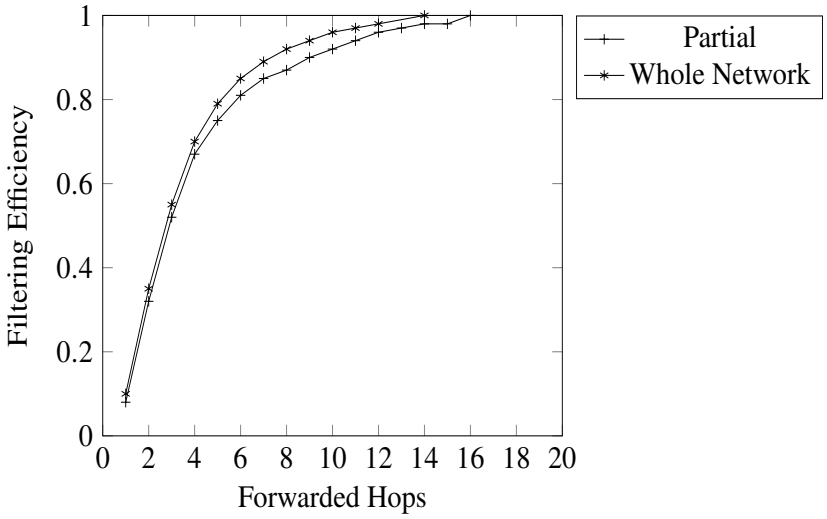


Figure 6.1: Filtering Efficiency vs Forwarded Hops when key pre-distribution done in Partial ($(3/4)^{th}$ Cluster Heads) and Whole network

sensor nodes and CHs are chosen randomly in the network.

In terms of simulating various attacks we randomly select the sensor nodes and

Table 6.3: Simulation Parameters

Parameters	**Values**
Number of Sensor nodes	10005
Total number of Cells	667
Total sensor nodes in each cell	15
Cell Size	$50X50m^2$
Cluster Heads in each cell	3
Communication range of Sensor nodes	$25m$
Communication range of Cluster Heads	$50m$
(T, t)	5,4
Size of Report	36 Bytes
Size of MAC	4 Bytes
Energy Consumed to generate MAC	$15\mu J$(Per Byte)
Energy Consumed to verify MAC	$75\mu J$
Energy Consumed to Transmit/Receive	$16.25/12.5\mu J$(Per Byte)

cluster heads which are compromised. After the identification of compromised sensor nodes we simulate a report generation and forwarding phase. In this phase each cell in the network tries to create a report and sends the report towards the sink through intermediate CHs.

In terms of simulating report disruption attack in the network, all the compromised sensor nodes provide with dummy data/MACs at report generation phase. As report generation in each cell is done by mutual collaboration between few sensor nodes, in simulation we try to identify cells where compromised sensor nodes can deny this report generation by sending false data/MACs. Specifically, we count the number of cells

which can successfully generate correct reports by neglecting false data/MACs from the compromised sensor nodes when a given number of sensor nodes are compromised in the network.

In terms of simulating selective forwarding attack in the network, all compromised intermediate CHs purposely drop the incoming reports in place of forwarding them to the next hop. As report forwarding/ verification is done hop by hop in direction of sink, in simulation we try to identify how many genuine reports will be dropped by intermediate compromised CHs. Specifically, we count the number of cells whose correct reports are received by the sink when a given number of CHs are compromised in the network.

All the experiments, like choosing a fixed number of compromised sensor nodes/CHs in the network is repeated 50 times. This is mainly done to remove any ambiguity in the results.

6.4.1 Data Authenticity

In the proposed scheme (CD-PEFS), CHs of the event cell co-ordinate with each other such that 50% of the sensor nodes participating in the generation of the tuples are different for each CH. A CH will only generate a report if it receives T correct tuples. Further, CH generates MAC_{x_i} to be sent to other CHs in the same cell. The sink will accept the report if and only if at-least one of the MAC_{x_i} is correct. Thus, an adversary can inject a false report in the network which can by-pass the En-Route filtering and sink verification successfully if and only if one of the following conditions is satisfied:

1. Adversary can compromise at-least (3/2)T sensor nodes in a particular cell. In this condition, an adversary can produce (3/2)T correct tuples. So in such scenario, remaining (1/2)T tuples needs to be forged. Now, if only one of the CHs in the generating cell participates in the report generation, then it must receive T correct tuples. Otherwise, it will reject the forged tuples and report will not be generated. Let the probability that only one CH participates in the report generation and it receives forged tuples is P_x. Then $(1 - P_x)$ denotes the probability that any one of the CHs receives T correct tuples. The adversary can successfully forge a report

only in this case. If the number of compromised sensor nodes in the network are X, then the probability of injecting a false report in the network by an adversary is given by

$$P_{inj}(X) = \sum_{i=(3/2)T}^{n} \frac{\binom{n}{i}\binom{N-n}{X-i}}{\binom{N}{X}} (1 - P_x) \tag{6.5}$$

2. Adversary can compromise at-least T sensor nodes and one CH in the event cell. Similar to the above case, here also the CHs that participate in report generation must receive all T correct tuples. Moreover, if there is only one CH that can participate in report generation, then it must not be compromised. Let Q_x denotes the probability that there is only one CH that can participate in report generation and it is compromised. If the number of compromised sensor nodes in the network is X, out of which c CHs are compromised, then the probability of injecting a false report is given by:

$$P_{inj}(X) = \left(\sum_{i=T}^{n} \frac{\binom{n}{i}\binom{N-n}{(X-c)-i}}{\binom{N}{X-c}} \right) . \left(\frac{\binom{3}{1}\binom{3C-3}{c-1}}{\binom{3C}{c}} \right) . (1 - P_x)\,(1 - Q_x) \tag{6.6}$$

where C is the total number of cells in the network and $3C$ is the total number of CHs in the network.

3. Adversary is able to compromise at-least T sensor nodes and more than one CHs in the event cell. If the number of compromised sensor nodes in the network are X out of which c CHs are compromised then the probability of injecting a false report is given by:

$$P_{inj}(X) = \left(\sum_{i=T}^{n} \frac{\binom{n}{i}\binom{N-n}{(X-c)-i}}{\binom{N}{X-c}} \right) . \left(\sum_{j=1}^{3} \frac{\binom{3}{i}\binom{3C-3}{c-j}}{\binom{3C}{c}} \right) . (1 - P_x) \tag{6.7}$$

The cases discussed above are the worst case scenario where an adversary is successful in injecting a false report from a particular cell. To inject bogus reports for an

event occurring in other cells, more sensor nodes and CHs have to be compromised. Therefore, the proposed scheme (CD-PEFS) is better than schemes like LEDS (Ren et al. 2008) and PCREF (Yang et al. 2015), where an adversary needs to compromise only t and T sensor nodes in a particular cell respectively. Further, schemes like SEF (Ye et al. 2005a), IHA (Zhu et al. 2004) and LBRS (Yang et al. 2005) require only a single compromised sensor node which can result in multiple gains.

The comparison of the probability of introducing bogus reports in the network when few sensor nodes are compromised in the proposed scheme (CD-PEFS), PCREF, LEDS and SEF are given in Figure 6.2. So, lesser the probability of introducing bogus reports, better is the data authenticity of the scheme. In the figure, we can observe that SEF performs worst in the compared schemes, where the probability of introducing a bogus report becomes one when only 2000 sensor nodes are compromised in the network. LEDS and PCREF perform much better than SEF, where the probability of introducing a bogus report becomes one when 5000 sensor nodes are compromised in the network. The proposed scheme (CD-PEFS) on the other hand has 61% probability of introducing bogus reports when 5000 sensor nodes are compromised in the network. Finally, in the figure, we can observe that the probability of introducing bogus reports in the proposed scheme (CD-PEFS) is always less than SEF, PCREF and LEDS. Thus, the proposed scheme (CD-PEFS) provides much higher data authenticity when compared with SEF, LEDS and PCREF. Further, in the figure, we can observe that the proposed scheme (CD-PEFS) provides the same data authenticity as provided by CD-EFS (Discussed in Chapter 5).

6.4.2 Filtering Efficiency

The proposed scheme (CD-PEFS) uses a deterministic combinatorial key pre-distribution method which ensures that any two CHs of the same type share a common key if both of them are allocated the key-sets at the time of initialization. When a CH is compromised, all the $k'+1$ keys stored in it are undermined. Thus, when few CHs are compromised in the network, few keys in an un-compromised CH are also revealed. If the total number of keys stored in an un-compromised CH CH_i be $k'+1$, out of which y keys are revealed

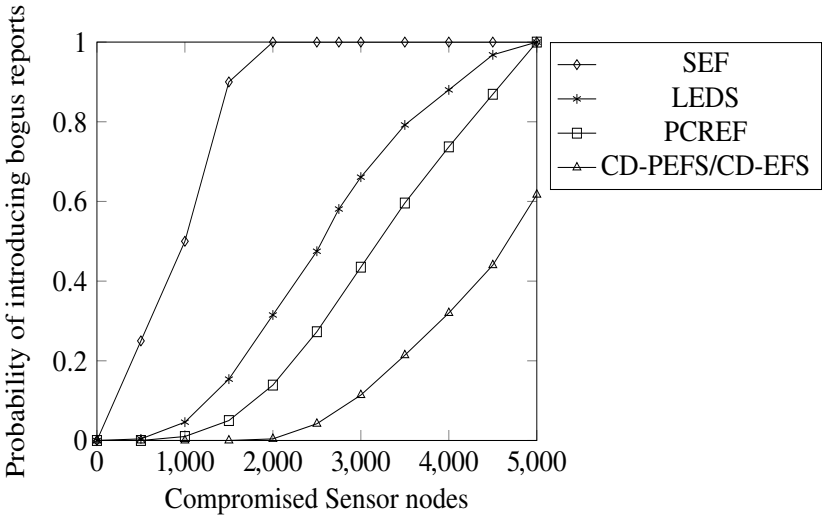


Figure 6.2: Data Authenticity in LEDS, PCREF, CD-EFS and the proposed scheme (CD-PEFS)

to an adversary. Then, to generate a false report on behalf of CH_i, an adversary needs to forge the remaining $(k' + 1 - y)$ MACs. Now, to filter this forged report from the network at-least one intermediate non-compromised CH which stores any one of the $(k' + 1 - y)$ non-revealed keys is sufficient.

If x sensor nodes are compromised in the network out of which X CHs are compromised and CH CH_i is H hops away from the sink, then the probability of filtering the false report generated on behalf of CH_i is given by,

$$P(X) = \sum_{i=0}^{H} (1 - P_{CH}^{x}) \binom{(3/4)C}{C} \frac{(k' + 1 - y)}{(k' + 1)} \tag{6.8}$$

where P_C^x is the probability that any CH is compromised. Further, P_C^x can be given by,

$$P_C^x = \frac{\binom{C}{X/3}\binom{3C-C}{X-X/3}}{\binom{3C}{X}} \tag{6.9}$$

where $3C$ is the total number of CHs in the network and $X/3$ is the number of CHs of a particular type that are compromised. At any given instance, the proposed scheme (CD-PEFS) can have either one, two or three copies of the same report depending on the number of CHs participating in report generation. Therefore, the probability of filtering

a false report from the network is given by,

1. When one copy of report is generated:

$$P_{filt} = P(X) \tag{6.10}$$

2. When two copies of report are generated:

$$P_{filt} = (P(X))^2 \tag{6.11}$$

3. When three copies of report are generated:

$$P_{filt} = (P(X))^3 \tag{6.12}$$

Figure 6.3 presents the experimental results of expected filtering position of the false report in the proposed scheme (CD-PEFS), CD-EFS (Discussed in Chapter 5), PCREF (Yang et al. 2015) and LEDS (Ren et al. 2008). So, lesser the expected filtering position, better is the filtering efficiency of the scheme. From the figure, we can observe that the proposed scheme (CD-PEFS) filters fabricated report in fewer hops than PCREF and LEDS. The main reason for this improvement in filtering efficiency is the adoption of a deterministic key pre-distribution method to assign keys to all the CHs. Further, we observe from the figure that the proposed scheme (CD-PEFS) can filter bogus reports from the network on average in 7 hops. This is a major improvement over schemes like LEDS, which requires 24 hops and PCREF which requires 11 hops on average to filter false reports, when total nodes compromised in the network are 1600 and distance from the sink is 40. Finally, in the figure, we can observe that the expected filtering position in the proposed scheme (CD-PEFS) is always less than PCREF and SEF. Thus, the proposed scheme has better filtering efficiency than PCREF and SEF. But the proposed scheme (CD-PEFS) has less filtering efficiency when compared to CD-EFS, this is mainly because of partial key pre-distribution in CD-PEFS.

Figure 6.4 provides filtering efficiency vs forwarded hops comparison between pro-

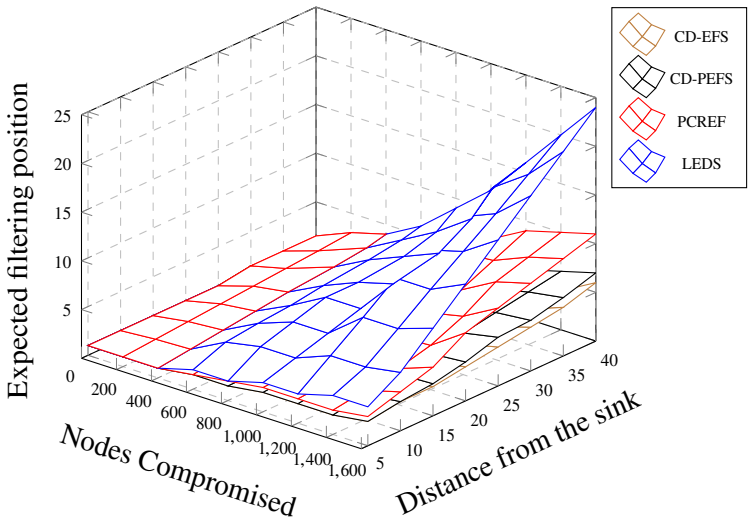


Figure 6.3: Average forwarded hops for false reports in LEDS, PCREF, CD-EFS and the proposed scheme (CD-PEFS).

posed scheme (CD-PEFS), CD-EFS, PCREF and LEDS. So, higher the filtering efficiency, better is the scheme. In the figure we can observe that LEDS and PCREF have filtering efficiency of 77% and 92%, respectively, when we fix the forwarded hops for the reports to 16. But on the other hand, the proposed scheme (CD-PEFS) has 100% filtering efficiency when we fix the forwarded hops to 16. Thus, it is evident that the proposed scheme (CD-PEFS) promises high filtering efficiency in very less number of hops.

6.4.3 Data Availability

Major threats to data availability in WSNs is because of report disruption attack and selective forwarding attack. In report disruption attack a compromised sensor node or CH hinders the report generation by sending wrong tuples or attaching wrong MACs to the report respectively. Due to which the genuine report is either dropped by intermediate CHs during En-Route filtering or by the sink during sink verification. On the other hand, in selective forwarding attack, a compromised intermediate CH can drop a genuine report passing through it. In the next section, we discuss the effect of both these attacks on the proposed scheme (CD-PEFS).

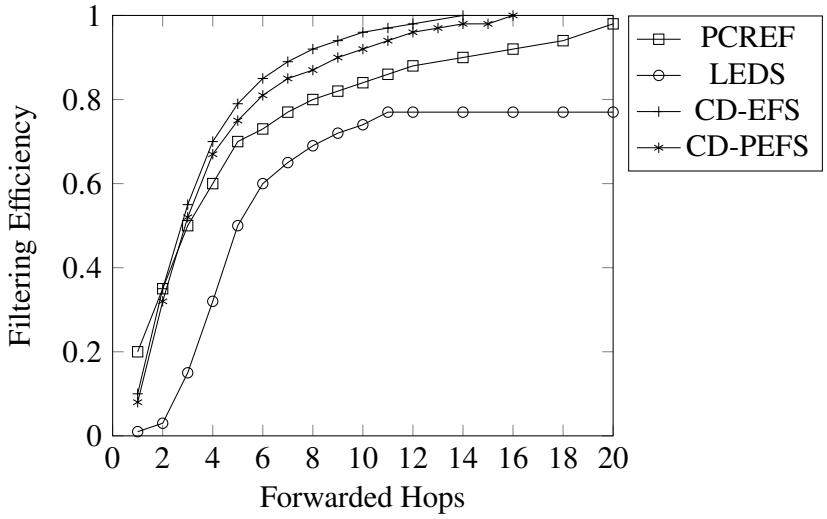


Figure 6.4: Filtering Efficiency vs Forwarded Hops in PCREF, LEDS and the proposed scheme (CD-PEFS)

a) Report Disruption Attack

The proposed scheme (CD-PEFS) follows (t, T) threshold LSSS (Ren et al. 2008) for generation of shares in the network, where each sensor node can produce at-most one share. Each participating CH contributes T shares, out of which the sink can use any t correct shares to obtain the correct message. Therefore, a sink can successfully recover a message if at-least t correct shares are present in the report. Moreover, all the participating CHs coordinate with each other to choose T tuples such that each CH chooses at-least 50% different tuples when compared with other CHs. Thus, the sink will be able to obtain the message successfully in the following cases:

1. **When no CH is compromised in the event cell** - The proposed scheme (CD-PEFS) can have either one, two or three reports generated by CHs in the event cell. Accordingly, there can be T, $(3/2)T$ or $2T$ unique shares received by the sink. Now in any of the three cases, the sink can successfully obtain the message if and only if it receives t correct tuples. Thus, data availability can be expressed as:

- *When one CH participates in report generation*: Data availability (P_{Avail}) in such a scenario can be expressed as,

$$P_{Avail}(X) = \sum_{z=0}^{T-t} \frac{\binom{T}{z}\binom{N-T}{X-z}}{\binom{N}{X}} (P'_x) \tag{6.13}$$

where, P'_x is the probability that only one CH participates in report generation, N is the total number of sensor nodes in the network and X is the number of sensor nodes/CHs compromised in the network. P'_x can be expressed as,

$$P'_x = \binom{3}{1}\binom{(3/4)C}{C}\binom{(1/4)C}{C}^2 \tag{6.14}$$

- *When two CHs participate in report generation*: Data availability (P_{Avail}) in such a scenario can be expressed as,

$$P_{Avail}(X) = \sum_{z=0}^{(3/2)T-t} \frac{\binom{(3/2)T}{z}\binom{N-(3/2)T}{X-z}}{\binom{N}{X}} (P''_x) \tag{6.15}$$

where, P''_x is the probability that two CHs participate in report generation. It can be expressed as,

$$P''_x = \binom{3}{2}\binom{(3/4)C}{C}^2\binom{(1/4)C}{C} \tag{6.16}$$

- *When three CHs participate in report generation*: Data availability (P_{Avail}) in such a scenario can be expressed as,

$$P_{Avail}(X) = \sum_{z=0}^{2T-t} \frac{\binom{2T}{z}\binom{N-2T}{X-z}}{\binom{N}{X}} (P'''_x) \tag{6.17}$$

where, P'''_x is the probability that all three CHs participates in report generation. It can be expressed as

$$P'''_x = \binom{(3/4)C}{C}^3 \tag{6.18}$$

2. **When one CH is compromised in the event cell** - If one CH is compromised in the event cell, then the sink can receive a maximum of two copies of the report. Now, depending on the number of CHs participating in report generation we can have the following cases:

- *When only one CH participates in report generation*: If single CH which participates in report generation is compromised then no report is generated by participating CH. If any one of the other two CH is compromised and less than $T - t$ participating nodes are compromised, a report can be successfully received by the sink. In such a case, the sink will receive T shares with at-least one correct MAC_{x_i}. The sink will use any of the t correct tuples to generate the message. Let the number of compromised sensor nodes in the network be x out of which X is the number of compromised CHs, data availability in such a scenario can be expressed as,

$$P_{Avail}(X) = \frac{\binom{3}{1}\binom{3C-3}{x-1}}{\binom{3C}{x}} \sum_{z=0}^{T-t} \frac{\binom{T}{z}\binom{N-T}{(X-x)-z}}{\binom{N}{X-x}}.(1 - Q_x) \tag{6.19}$$

  where, Q_x denotes the probability that there is only one CH that can participate in report generation and it is compromised and N is the total number of sensor nodes in the network.

- *When more than one CHs participate in report generation*: If one of the CHs which participated in report generation is compromised then the number of shares received by the sink is T. Otherwise, the sink will receive $(3/2)T$ shares. Data availability in both the cases can be expressed as,

$$P_{Avail}(X) = \frac{\binom{3}{1}\binom{3C-3}{x-1}}{\binom{3C}{x}} \sum_{z=0}^{T-t} \frac{\binom{T}{z}\binom{N-T}{(X-x)-z}}{\binom{N}{X-x}}.(P''_x) \tag{6.20}$$

$$P_{Avail}(X) = \frac{\binom{3}{1}\binom{3C-3}{x-1}}{\binom{3C}{x}} \sum_{z=0}^{(3/2)T-t} \frac{\binom{T}{z}\binom{N-(3/2)T}{(X-x)-z}}{\binom{N}{X-x}}.(P'''_x) \tag{6.21}$$

  where, P''_x represents (Eq. 6.16) probability that two CHs participates in report generation and P'''_x represents (Eq. 6.18) probability that three CHs participates in report generation.

So, the above discussed cases formalize the probability of receiving/recovering correct reports by the sink. In the above cases, we discussed that there is a high probability of recovering the original reports in the proposed scheme because of the forwarding of multiple copies of the same report in the network. Thus, the proposed scheme (CD-PEFS) is more resilient than LEDS (Ren et al. 2008) where sink can obtain the message only if the number of compromised nodes is less than $(T - t)$. Thus, the proposed scheme (CD-PEFS) is more resilient than LEDS. Further, other schemes like SEF (Ye et al. 2005a), IHA (Zhu et al. 2004), LBRS (Yang et al. 2005) and PCREF (Yang et al. 2015) requires all the secret shares to be correct to recover the original report. Hence, the proposed scheme (CD-PEFS) is more resilient than SEF, IHA, LBRS and PCREF.

Figure 6.5 provides the probability of generating correct reports in the proposed scheme (CD-PEFS) with CD-EFS (Discussed in Chapter 5), SEF, LEDS and PCREF. So, greater the probability of generating correct reports, better is the data availability in the scheme. Specifically, in the figure we can observe that the probability of generating correct reports in PCREF/SEF, LEDS and the proposed scheme (CD-PEFS) is 3%, 19% and 50%, respectively, when 3000 sensor nodes are compromised in the network. Moreover, in the figure, we can observe that the proposed scheme always has a higher probability of generating correct reports when compared with PCREF, SEF and LEDS. Thus, we can conclude that the proposed scheme (CD-PEFS) provides much better data availability than SEF, LEDS and PCREF. Further, the proposed scheme (CD-PEFS) and CD-EFS has an equal probability of generating correct reports. Thus, both the schemes CD-PEFS and CD-EFS ensures high resiliency against report disruption attack in the network.

b) Selective Forwarding Attack

In the proposed scheme (CD-PEFS) there can be one, two or three copies of the same report in the network. The sink can recover the original message if it receives at-least one copy of the report. Any compromised node in the route from source to sink might drop the report. But, if multiple copies of the same report are in the network, then more compromised sensor nodes will be required to drop all the copies of the same report. Thus, the probability that a report is dropped in the network will vary based upon the

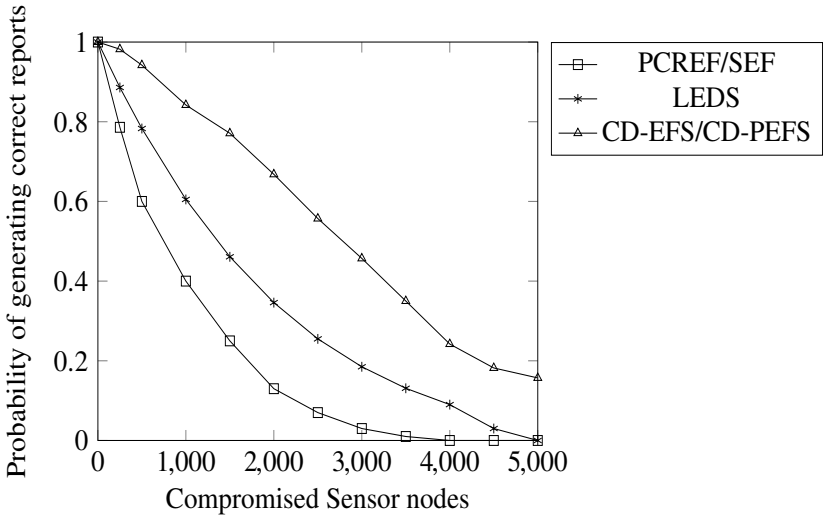


Figure 6.5: Data Availability under Report Disruption attack in LEDS, PCREF, SEF, CD-EFS and the proposed scheme (CD-PEFS).

number of copies of the report in the network.

If the sink is H hops away from the source and let x sensor nodes be compromised out of which X CHs are compromised. Then, the probability that at-least one CH of a particular type is compromised can be expressed as:

$$P_{com}(X) = \sum_{z=1}^{H} \frac{\binom{H}{z}\binom{C-H}{(X/3)-z}}{\binom{C}{X/3}} \tag{6.22}$$

Thus, the probability that a report is dropped is expressed as:

1. If one copy of the report exist in network

$$P_{drop}(X) = P_{com}(X) \tag{6.23}$$

2. If two copies of the report exist in network

$$P_{drop}(X) = (P_{com}(X))^2 \tag{6.24}$$

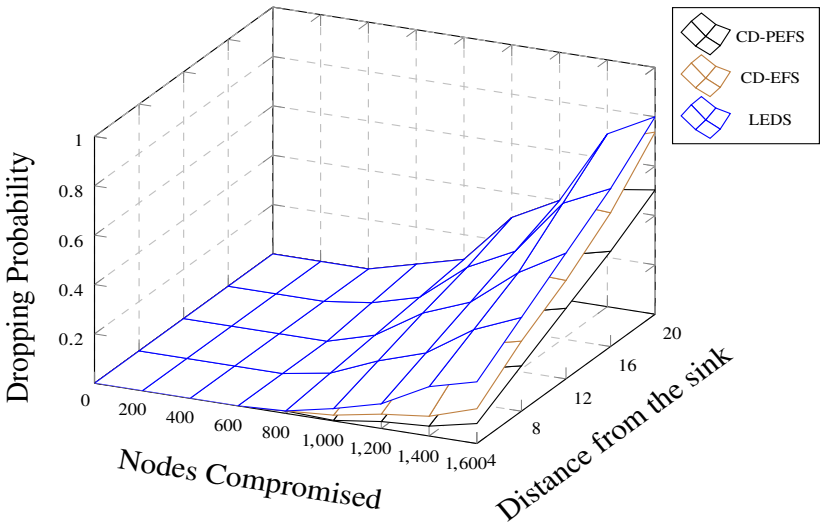


Figure 6.6: Data Availability under Selective Forwarding attack for the proposed scheme (CD-PEFS), CD-EFS and LEDS

3. If three copies of the report exist in network

$$P_{drop}(X) = (P_{com}(X))^3 \tag{6.25}$$

As discussed above, the proposed scheme (CD-PEFS) forwards multiple copies of the same report in the network. Thus, the proposed scheme is highly resilient to selective forwarding attack. LEDS (Ren et al. 2008) broadcasts a report to all sensor nodes in the report forwarding phase, due to which it is highly resilient to selective forwarding attack. Other schemes like SEF (Ye et al. 2005a), IHA (Zhu et al. 2004), LBRS (Yang et al. 2005) and PCREF (Yang et al. 2015) are highly vulnerable to selective forwarding attack. In the proposed scheme (CD-PEFS) only CHs participate in En-Route filtering due to which it becomes highly resilient to selective forwarding attack since compromised sensor nodes can not participate in dropping the report.

Figure 6.6 presents the experimental comparison of the dropping probability of genuine reports in the proposed scheme (CD-PEFS), CD-EFS (Discussed in Chapter 5) and LEDS (Ren et al. 2008). So, lesser the dropping probability of genuine reports, better is the data availability in the scheme. In LEDS at the time of report forwarding, each report is broadcasted to all the sensor nodes in an intermediate cell and because of which LEDS is highly resilient to selective forwarding attack. But other existing schemes such

as SEF (Ye et al. 2005a), PCREF (Yang et al. 2015), LBRS (Yang et al. 2005), IHA (Zhu et al. 2004) do-not adopt any preventive measures against selective forwarding attack because of which these schemes are highly prone to selective forwarding attack. We can observe from Figure 6.6 that the proposed scheme (CD-PEFS) outperforms LEDS even without broadcasting reports to intermediate cells. Finally, in the figure, we can observe that the proposed scheme (CD-PEFS) always has a lower probability of dropping genuine reports when compared with LEDS and CD-EFS. Thus, we can conclude that the proposed scheme (CD-EFS) provides much better data availability than LEDS and CD-EFS.

6.5 Performance Evaluation of the Proposed scheme (CD-PEFS)

In this section, we discuss the performance evaluation of the proposed scheme (CD-PEFS). We discuss the key storage overhead, computation overhead, communication overhead and energy requirements associated with the proposed scheme (CD-PEFS).

6.5.1 Key Storage Overhead

The proposed scheme (CD-PEFS) uses combinatorial design based keys in CHs. Thus, each CH stores only $k' + 1$ keys for En-Route filtering. In addition to these keys, each CH stores a secret key to communicate securely with the sink and one to communicate with other sensor nodes in the cell. Sensor nodes on the other hand store only four master secret keys, one for securing communication with the sink and remaining three to communicate securely with three CHs in the home cell. In the proposed scheme, if $C = 150$, then 13 keys are stored in $3/4^{th}$ of total CHs and remaining $1/4^{th}$ CHs have only 2 keys. In LEDS (Ren et al. 2008), each sensor node stores $(T+1)(T+2)/2+5$ keys. Thus, the total number of keys stored by sensor nodes/CHs in LEDS is 26, when $T = 5$. In PCREF (Yang et al. 2015), sensor nodes need to store large number of polynomials and secret keys which can be obtained from $(64.n_k + 16.n_c)$, where n_k is the total keys and n_c is the number of coefficients. Thus, the proposed scheme (CD-PEFS) has less key storage overhead when compared with existing schemes. The reduction in key storage overhead is mostly due to the symmetric key pre-distribution scheme employed

in the generation of keys for En-Route filtering in CHs.

6.5.2 Computation and Communication Overhead

In any En-Route filtering scheme, communication overhead is mostly due to the messages exchanged between sensor nodes for initial key exchange, shared key discovery and report forwarding from source to the sink. In the proposed scheme (CD-PEFS), the initial key exchange requires only one message to be sent to all sensor nodes by the network controller. The shared key discovery is not required as the keys for En-Route filtering are generated using symmetric design which ensures that there exists a common key between any two CHs. For En-Route filtering, CHs need to attach $k' + 1$ MACs and two MAC_{X_i} in the report. Thus, the communication overhead for En-Route filtering will be approximately equal to the cost of transferring $k' + 3$ MACs in the network. IHA (Zhu et al. 2004), LBRS (Yang et al. 2005) and LEDS (Ren et al. 2008) uses 3-way-handshake for shared key discovery. For En-Route filtering, LEDS attaches $T+1$ MACs in the final report and PCREF (Yang et al. 2015) has communication overhead of around 40 bytes (details in Yang et al. (2015)) for the same network configuration.

Computation overhead in any En-Route filtering schemes is due to the generation and verification of MACs. In the proposed scheme (CD-PEFS), sensor nodes that detect the event, send their tuples to the CHs and CHs generate the final report. Further, only CHs are involved in the report verification process. Sensor nodes do not participate in the verification of MACs. This saves a lot of computation overhead in sensor nodes.

6.6 Concluding Remarks

In this chapter, we proposed a new deterministic En-Route filtering scheme using combinatorial design. In the proposed scheme (CD-PEFS), key-sets are generated using the combinatorial design. To reduce the key storage overhead in the CHs, we assign key-sets to only $3/4^{th}$ of the total CHs. Due to the applied changes, the number of association which provide data authentication in the network reduces. The selection of CHs of each type for key-set assignment is done in such a way that connectivity between all the cells in the network is maintained. In the proposed scheme (CD-PEFS), we can

eliminate the shared key discovery phase, thus reducing communication and computational overhead. We performed a detailed analysis of the proposed scheme (CD-PEFS) and compared it with existing schemes. We observed that the proposed scheme (CD-PEFS) has high filtering efficiency while maintaining associated low overheads than most of the existing schemes. Moreover, the proposed scheme (CD-PEFS) is highly robust against DoS attacks like selective forwarding attacks and report disruption attack. Finally, the proposed scheme (CD-PEFS) has very low energy requirements when compared with other existing schemes.

CHAPTER 7

Conclusion and Future Work

En-Route filtering is a method to filter false reports from the WSNs. The basic idea of En-Route filtering is checking of the reports by the intermediate nodes. This helps to decrease the processing and checking overhead of sink and thus false reports can be removed from the network within few nodes from the origin, saving energy and bandwidth. In this approach, each report is attached to MACs or signatures. Whenever these reports are being forwarded over the network, intermediate nodes can authenticate these MACs or signatures and if any fault is found, reports are dropped. For creation and verification of MACs in the network, sensor nodes exchange secret keys with other sensor nodes in the network.

This research work was mainly focused on proposing novel and effective En-Route filtering schemes. This started with a thorough survey of existing En-Route filtering schemes. This led to the identification of design flaws including security issues in existing En-Route filtering schemes. Further, it helped to identify basic phases in En-Route filtering, i.e., Key Pre-Distribution phase and En-Route Filtering phase. Thus, the first half of the research work was focused on proposing new key pre-distribution schemes and the following half was dedicated on extending the proposed key pre-distribution schemes to propose new En-Route filtering schemes.

In the direction to device a new key pre-distribution scheme, we adopted Combinatorial Design. We adopted a new combinatorial design to create key-sets and we

assigned these key-sets to all the sensor nodes in such a way that the proposed scheme (**CD-KPD**) provided better resiliency than existing schemes. To reduce the key storage overhead in CD-KPD, two new schemes were proposed. In the first proposed scheme (**CD-RKPD**), inter-cell communication was limited to only *Lee sphere* region; thus now each sensor node required less number of keys. In the second proposed scheme (**CD-PKPD**), the key assignment was done in the partial network. In the proposed scheme, each cell has three cluster heads and key assignment between all the cluster heads was limited to exactly $3/4^{th}$ of the total cluster heads. As the total number of cluster heads to whom each cluster head can communicate decreases, the number of keys stored by each cluster head decreases. But for keys assignment, the selection of $3/4^{th}$ of total cluster heads for each cluster head was done in such a way that the network still maintained proper connectivity and very good resiliency against compromised nodes in the network. Finally, a novel hybrid key pre-distribution (**CD-HKPD**) was proposed which used both pair-wise keys and combinatorial design based keys. In the proposed scheme, combinatorial design based keys are used to secure intra-cell communication, which helped to maintain low key storage overhead in the network. For inter-cell communication, each cell maintained multiple associations with all the other cells within the communication range, and these associations are secured with pair-wise keys. This helped to ensure high resiliency against compromised sensor nodes in the network while maintaining very low key storage overhead.

For the second part of the research, CD-KPD was extended to propose a new Combinatorial design based En-Route filtering scheme (**CD-EFS**). Use of combinatorial design based keys provided a deterministic mechanism for verification of forwarded reports. Thus, the filtering efficiency of the proposed scheme (CD-KPD) was excellent. For the same, a novel beam model was proposed to reduce key storage overhead in the network. Further, CD-PKPD was extended to propose a new En-Route filtering scheme (**CD-PEFS**). Finally for both the proposed schemes (CD-EFS and CD-PEFS), a novel report endorsement and verification mechanism was proposed for robust data authentication and data availability in the network. This helped to provide better tolerance against report disruption and selective forwarding attacks in WSNs.

Future Scope

As the schemes proposed in this thesis are pioneering approaches for key pre-distribution and En-Route filtering in WSNs, there is significant scope for future research. Further research can be carried in the following directions:

- **Scalability**- All the proposed key pre-distribution and En-Route filtering schemes in this thesis are based on combinatorial design. We require the number of sensor nodes participating in the key assignment, for assigning them combinatorial design based keys. Thus, the size of the network has to be pre-determined for key assignment in such a scenario. So, all the proposed schemes in this thesis provide limited scalability. Thus, addressing scalability while using combinatorial design based keys would be interesting future work.

- **Identification and Removal of compromised nodes in the network**- All the proposed En-Route filtering schemes in the thesis are centered on filtering false reports from the network. But, no mechanisms were adopted to identify and quarantine/remove the compromised sensor nodes from the network. Therefore, devising a new mechanism for the identification and removal of compromised nodes in the network would be another interesting future research direction.

In Conclusion, this dissertation proposes novel En-Route filtering schemes based on combinatorial design. Initially, we proposed novel combinatorial design based key pre-distribution schemes. We have also proposed a hybrid key pre-distribution scheme which uses both pair-wise keys and combinatorial design based keys. Further, we extended the proposed key pre-distribution schemes to propose deterministic En-Route filtering schemes.

Bibliography

Akhondi, M. R., Talevski, A., Carlsen, S. and Petersen, S. (2010). "Applications of wireless sensor networks in the oil, gas and resources industries." In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, IEEE, 941–948.

Akkaya, K. and Younis, M. (2005). "A survey on routing protocols for wireless sensor networks." *Ad hoc networks*, 3(3), 325–349.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002). "A survey on sensor networks." *Communications magazine, IEEE*, 40(8), 102–114.

Alemdar, H. and Ersoy, C. (2010). "Wireless sensor networks for healthcare: A survey." *Computer Networks*, 54(15), 2688–2710.

Anderson, I. (1990). *Combinatorial designs: construction methods*, Ellis Horwood.

Bag, S. (2015). "A new key predistribution scheme for grid-group deployment of wireless sensor networks.." *Adhoc & Sensor Wireless Networks*, 27.

Bag, S. and Roy, B. (2013). "A new key predistribution scheme for general and grid-group deployment of wireless sensor networks." *EURASIP Journal on Wireless Communications and Networking*, 2013(1), 145.

Bauer, K., McCoy, D., Grunwald, D., Kohno, T. and Sicker, D. (2007). "Low-resource routing attacks against tor." In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, ACM, 11–20.

Black, P. E. (2006). "Manhattan distance." *Dictionary of Algorithms and Data Structures*, 18, 2012.

Blackburn, S. R., Etzion, T., Martin, K. M. and Paterson, M. B. (2008). "Efficient key predistribution for grid-based wireless sensor networks." In *International Conference on Information Theoretic Security*, Springer, 54–69.

Blom, R. (1984). "An optimal class of symmetric key generation systems." In *Workshop on the Theory and Application of of Cryptographic Techniques*, Springer, 335–338.

Bloom, B. H. (1970). "Space/time trade-offs in hash coding with allowable errors." *Communications of the ACM*, 13(7), 422–426.

Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U. and Yung, M. (1992). "Perfectly-secure key distribution for dynamic conferences." In *Annual International Cryptology Conference*, Springer, 471–486.

Çamtepe, S. A. and Yener, B. (2007). "Combinatorial design of key distribution mechanisms for wireless sensor networks." *IEEE/ACM Transactions on networking*, 15(2), 346–358.

Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y. and Sastry, S. (2011). "Attacks against process control systems: risk assessment, detection, and response." In *Proceedings of the 6th ACM symposium on information, computer and communications security*, ACM, 355–366.

Chakrabarti, D., Maitra, S. and Roy, B. (2006). "A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design." *International Journal of Information Security*, 5(2), 105–114.

Chan, H. and Perrig, A. (2003). "Security and privacy in sensor networks." *Computer*, 36(10), 103–105.

Chan, H., Perrig, A. and Song, D. (2003). "Random key predistribution schemes for sensor networks." In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, IEEE, 197–213.

Chen, Y.-S. and Lei, C.-L. (2010). "Filtering false messages en-route in wireless multi-hop networks." In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, IEEE, 1–6.

Chi, S. H. and Cho, T. H. (2006). "Fuzzy logic based propagation limiting method for message routing in wireless sensor networks." In *Computational Science and Its Applications-ICCSA 2006*, Springer, 58–67.

Choi, H. M., Nam, S. M., Cho, T. H. et al. (2013). "A secure routing method for detecting false reports and wormhole attacks in wireless sensor networks." *Wireless Sensor Network*, 5(03), 33.

Dickson, L. E. (2003). *Linear groups: With an exposition of the Galois field theory*, Courier Corporation.

Du, W., Deng, J., Han, Y. S. and Varshney, P. K. (2006). "A key predistribution scheme for sensor networks using deployment knowledge." *IEEE Transactions on dependable and secure computing*, 3(1), 62–77.

Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J. and Khalili, A. (2005). "A pairwise key predistribution scheme for wireless sensor networks." *ACM Transactions on Information and System Security (TISSEC)*, 8(2), 228–258.

Eschenauer, L. and Gligor, V. D. (2002). "A key-management scheme for distributed sensor networks." In *Proceedings of the 9th ACM conference on Computer and communications security*, ACM, 41–47.

Fakhrey, H., Tiwari, R., Johnston, M. and Al-Mathehaji, Y. A. (2016). "The optimum design of location-dependent key management protocol for a wsn with a random selected cell reporter." *IEEE Sensors Journal*, 16(19), 7217–7226.

Hamid, M. A., Mamun-Or-Rashid, M. and Hong, C. S. (2006). "Routing security in sensor network: Hello flood attack and defense." *IEEE ICNEWS*, 2–4.

Hankerson, D., Menezes, A. J. and Vanstone, S. (2006). *Guide to elliptic curve cryptography*, Springer Science & Business Media.

He, S., Chen, J., Sun, Y., Yau, D. K. and Yip, N. K. (2010). "On optimal information capture by energy-constrained mobile sensors." *Vehicular Technology, IEEE Transactions on*, 59(5), 2472–2484.

He, T., Huang, C., Blum, B. M., Stankovic, J. A. and Abdelzaher, T. (2003). "Range-free localization schemes for large scale sensor networks." In *Proceedings of the 9th annual international conference on Mobile computing and networking*, ACM, 81–95.

Hu, Y., Lin, Y., Liu, Y. and Zeng, W. (2007). "Ras: A robust authentication scheme for filtering false data in wireless sensor networks." In *Networks, 2007. ICON 2007. 15th IEEE International Conference on*, IEEE, 200–205.

Hu, Y.-C., Perrig, A. and Johnson, D. B. (2003). "Packet leashes: a defense against wormhole attacks in wireless networks." In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, IEEE, 1976–1986.

Huang, D. and Medhi, D. (2007). "Secure pairwise key establishment in large-scale sensor networks: An area partitioning and multigroup key predistribution approach." *ACM Transactions on Sensor Networks (TOSN)*, 3(3), 16.

Huang, D., Mehta, M., Medhi, D. and Harn, L. (2004). "Location-aware key management scheme for wireless sensor networks." In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, ACM, 29–42.

Hussain, M. A., Khan, P. and Sup, K. K. (2009). "Wsn research activities for military application." In *Proceedings of the 11th international conference on Advanced Communication Technology-Volume 1*, IEEE Press, 271–274.

Karlof, C. and Wagner, D. (2003). "Secure routing in wireless sensor networks: Attacks and countermeasures." *Ad hoc networks*, 1(2), 293–315.

Karp, B. and Kung, H.-T. (2000). "Gpsr: Greedy perimeter stateless routing for wireless networks." In *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 243–254.

Kraub, C., Schneider, M., Bayarou, K. and Eckert, C. (2007). "Stef: A secure ticket-based en-route filtering scheme for wireless sensor networks." In *Availability, reliability and security, 2007. ARES 2007. The second international conference on*, IEEE, 310–317.

Kumar, A., Bansal, N. and Pais, A. (2019). "A new key pre-distribution scheme based on combinatorial design for wireless sensor networks." *IET Communications*.

Kumar, A. and Pais, A. R. (2017). "En-route filtering techniques in wireless sensor networks: a survey." *Wireless Personal Communications*, 96(1), 697–739.

Kumar, A. and Pais, A. R. (2018a). "Deterministic en-route filtering of false reports: A combinatorial design based approach." *IEEE Access*, 6, 74494–74505.

Kumar, A. and Pais, A. R. (2018b). "A new combinatorial design based data en-route filtering scheme for wireless sensor networks." In *2018 Twenty Fourth National Conference on Communications (NCC)*, IEEE, 1–6.

Kumar, A. and Pais, A. R. (2018c). "A new combinatorial design based key pre-distribution scheme for wireless sensor networks." *Journal of Ambient Intelligence and Humanized Computing*, 1–16.

Kumar, A. and Pais, A. R. (2018d). "A new hybrid key pre-distribution scheme for wireless sensor networks." *Wireless Networks*, 25(3), 1185–1199.

Lamport, L. (1981). "Password authentication with insecure communication." *Communications of the ACM*, 24(11), 770–772.

Lee, J. and Stinson, D. R. (2005). "A combinatorial approach to key predistribution for distributed sensor networks." In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 2, IEEE, 1200–1205.

Lee, J. and Stinson, D. R. (2006). "Common intersection designs." *Journal of Combinatorial Designs*, 14(4), 251–269.

Li, F. and Wu, J. (2006). "A probabilistic voting-based filtering scheme in wireless sensor networks." In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, ACM, 27–32.

Liu, A. and Ning, P. (2008). "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks." In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, IEEE, 245–256.

Liu, D. and Ning, P. (2005). "Improving key predistribution with deployment knowledge in static sensor networks." *ACM Transactions on Sensor Networks (TOSN)*, 1(2), 204–239.

Liu, D., Ning, P. and Li, R. (2005). "Establishing pairwise keys in distributed sensor networks." *ACM Transactions on Information and System Security (TISSEC)*, 8(1), 41–77.

Liu, Y., Ning, P. and Reiter, M. K. (2011). "False data injection attacks against state estimation in electric power grids." *ACM Transactions on Information and System Security (TISSEC)*, 14(1), 13.

Liu, Z., Wang, J., Zhang, S., Liu, H. and Zhang, X. (2014). "A cluster-based false data filtering scheme in wireless sensor networks.." *Adhoc & Sensor Wireless Networks*, 23.

Lu, R., Lin, X., Zhu, H., Liang, X. and Shen, X. (2012). "Becan: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks." *Parallel and Distributed Systems, IEEE Transactions on*, 23(1), 32–43.

Luk, M., Perrig, A. and Whillock, B. (2006). "Seven cardinal properties of sensor network broadcast authentication." In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, ACM, 147–156.

Marti, S., Giuli, T. J., Lai, K. and Baker, M. (2000). "Mitigating routing misbehavior in mobile ad hoc networks." In *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 255–265.

Misra, P. and Enge, P. (2006). "Global positioning system: Signals, measurements and performance second edition." *Massachusetts: Ganga-Jamuna Press*.

Mitra, S., Mukhopadhyay, S. and Dutta, R. (2012). "A flexible deterministic approach to key pre-distribution in grid based wsns." In *International Conference on Ad Hoc Networks*, Springer, 164–179.

Nam, S. M. and Cho, T. H. (2016). "Context-aware architecture for probabilistic voting-based filtering scheme in sensor networks." *IEEE Transactions on Mobile Computing*.

Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O., Moses, R. L. and Correal, N. S. (2005). "Locating the nodes: cooperative localization in wireless sensor networks." *IEEE Signal processing magazine*, 22(4), 54–69.

Patwari, N., Hero III, A. O., Perkins, M., Correal, N. S. and O'dea, R. J. (2003). "Relative location estimation in wireless sensor networks." *Signal Processing, IEEE Transactions on*, 51(8), 2137–2148.

Potyrailo, R. A., Nagraj, N., Surman, C., Boudries, H., Lai, H., Slocik, J. M., Kelley-Loughnane, N. and Naik, R. R. (2012). "Wireless sensors and sensor networks for homeland security applications." *TrAC Trends in Analytical Chemistry*, 40, 133–145.

Przydatek, B., Song, D. and Perrig, A. (2003). "Sia: Secure information aggregation in sensor networks." In *Proceedings of the 1st international conference on Embedded networked sensor systems*, ACM, 255–265.

Rabaey, C. S. J. and Langendoen, K. (2002). "Robust positioning algorithms for distributed ad-hoc wireless sensor networks." In *USENIX technical annual conference*, 317–327.

Ren, K., Lou, W. and Zhang, Y. (2008). "Leds: Providing location-aware end-to-end data security in wireless sensor networks." *IEEE Transactions on Mobile Computing*, 7(5), 585–598.

Royer, E. M. and Toh, C.-K. (1999). "A review of current routing protocols for ad hoc mobile wireless networks." *IEEE personal communications*, 6(2), 46–55.

Ruj, S. and Roy, B. (2009). "Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks." *ACM Transactions on Sensor Networks (TOSN)*, 6(1), 4.

Shahzad, M. K. and Cho, T. H. (2017). "An energy-aware routing and filtering node (erf) selection in ccef to extend network lifetime in wsn." *IETE Journal of Research*, 1–13.

Shamir, A. (1979). "How to share a secret." *Communications of the ACM*, 22(11), 612–613.

Shi, E. and Perrig, A. (2004). "Designing secure sensor networks." *Wireless Communications, IEEE*, 11(6), 38–43.

Simonova, K., Ling, A. C. and Wang, X. S. (2006). "Location-aware key predistribution scheme for wide area wireless sensor networks." In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, ACM, 157–168.

Son, B., Her, Y.-s. and Kim, J.-G. (2006). "A design and implementation of forest-fires surveillance system based on wireless sensor networks for south korea mountains." *International Journal of Computer Science and Network Security (IJCSNS)*, 6(9), 124–130.

Stinson, D. R. (2007). *Combinatorial designs: constructions and analysis*, Springer Science & Business Media.

Sun, C. I., Lee, H. Y. and Cho, T. H. (2009). "A path selection method for improving the detection power of statistical filtering in sensor networks.." *J. Inf. Sci. Eng.*, 25(4), 1163–1175.

Sun, Q. and Wu, M. (2011). "A double key-sharing based false data filtering scheme in wireless sensor networks." In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, IEEE, 509–516.

Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J. and Culler, D. (2004). "An analysis of a large scale habitat monitoring application." In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ACM, 214–226.

Tsudik, G. (1992). "Message authentication with one-way hash functions." *ACM SIGCOMM Computer Communication Review*, 22(5), 29–38.

Vogt, H. (2004). "Exploring message authentication in sensor networks." In *Security in Ad-hoc and Sensor Networks*, Springer, 19–30.

Wang, H. and Li, Q. (2010). "Achieving robust message authentication in sensor networks: a public-key based approach." *Wireless Networks*, 16(4), 999–1009.

Wang, H., Sheng, B., Tan, C. C. and Li, Q. (2008). "Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control." In *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, IEEE, 11–18.

Wang, J., Liu, Z., Zhang, S. and Zhang, X. (2014). "Defending collaborative false data injection attacks in wireless sensor networks." *Information Sciences*, 254, 39–53.

Wenjie, C., Lifeng, C., Zhanglong, C. and Shiliang, T. (2005). "A realtime dynamic traffic control system based on wireless sensor network." In *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, IEEE, 258–264.

Xing, K. and Cheng, X. (2010). "From time domain to space domain: Detecting replica attacks in mobile ad hoc networks." In *INFOCOM, 2010 Proceedings IEEE*, IEEE, 1–9.

Yang, H. and Lu, S. (2004). "Commutative cipher based en-route filtering in wireless sensor networks." In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 2, IEEE, 1223–1227.

Yang, H., Ye, F., Yuan, Y., Lu, S. and Arbaugh, W. (2005). "Toward resilient security in wireless sensor networks." In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ACM, 34–45.

Yang, Q., Yang, J., Yu, W., Zhang, N. and Zhao, W. (2011). "On a hierarchical false data injection attack on power system state estimation." In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, IEEE, 1–5.

Yang, X., Lin, J., Yu, W., Moulema, P.-M., Fu, X. and Zhao, W. (2015). "A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems." *IEEE Transactions on Computers*, 64(1), 4–18.

Ye, F., Luo, H., Lu, S. and Zhang, L. (2005a). "Statistical en-route filtering of injected false data in sensor networks." *IEEE Journal on Selected Areas in Communications*, 23(4), 839–850.

Ye, F., Zhong, G., Lu, S. and Zhang, L. (2005b). "Gradient broadcast: A robust data delivery protocol for large scale sensor networks." *Wireless Networks*, 11(3), 285–298.

Younan, Y., Philippaerts, P., Piessens, F., Joosen, W., Lachmund, S. and Walter, T. (2011). "Filter-resistant code injection on arm." *Journal in computer virology*, 7(3), 173–188.

Yu, B. and Xiao, B. (2006). "Detecting selective forwarding attacks in wireless sensor networks." In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, IEEE, 8–pp.

Yu, H. and He, J. (2013). "Authentication and en-route data filtering for wireless sensor networks in the internet of things scenario." *International Journal of Grid and Distributed Computing*, 6(1), 1–12.

Yu, L. and Li, J. (2009). "Grouping-based resilient statistical en-route filtering for sensor networks." In *INFOCOM 2009, IEEE*, IEEE, 1782–1790.

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

Yu, Z. and Guan, Y. (2010). "A dynamic en-route filtering scheme for data reporting in wireless sensor networks." *IEEE/ACM transactions on networking*, 18(1), 150–163.

Yuan, T., Zhang, S., Zhong, Y. and Ma, J. (2008). "Kaef: An en-route scheme of filtering false data in wireless sensor networks." In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, IEEE, 193–200.

Zargar, S. T., Joshi, J. and Tipper, D. (2013). "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks." *Communications Surveys & Tutorials, IEEE*, 15(4), 2046–2069.

Zhang, Y., Liu, W., Lou, W. and Fang, Y. (2006). "Location-based compromise-tolerant security mechanisms for wireless sensor networks." *Selected Areas in Communications, IEEE Journal on*, 24(2), 247–260.

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

Zhu, S., Setia, S., Jajodia, S. and Ning, P. (2004). "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks." In *Security and privacy, 2004. Proceedings. 2004 IEEE symposium on*, IEEE, 259–271.

Publications

1. Kumar, A. and Pais, A. R. (2017). En-route filtering techniques in wireless sensor networks: a survey. *Wireless Personal Communications (Springer)*, 96(1):697–739. [**Published**]

2. Kumar, A. and Pais, A. R. (2018). A New Hybrid Key Pre-Distribution Scheme for Wireless Sensor Networks. *Wireless Networks (Springer)*, 25(3):1185–1199. [**Published**]

3. Kumar, A. and Pais, A. R. (2018). A new combinatorial design based key pre-distribution scheme for Wireless Sensor Networks. *Journal of Ambient Intelligence and Humanized Computing (Springer)* [**Published**]

4. Kumar, A. and Pais, A. R. (2018). Deterministic En-Route Filtering of False Data: A Combinatorial Design Based Approach. *IEEE Access (IEEE)*, 6:74494–74505. [**Published**]

5. Kumar, A. and Pais, A. R. (2019). A New Key Pre-Distribution Scheme Based on Combinatorial Design for Wireless Sensor Networks. *IET Communication (IET)* [**Published**]

6. Kumar, A. and Pais, A. R. (2019). A Novel Combinatorial Design Based En-Route Filtering Scheme for Wireless Sensor Networks. *Computers & Security (Elsevier)* [**Communicated**]

7. Kumar, A. and Pais, A. R. (2018). A New Combinatorial Design Based Data En-Route Filtering Scheme for Wireless Sensor Networks. 24^{th} *National Conference on Communication-2018 (NCC-2018)* [**Published**]