Data Security in Unattended Wireless Sensor Networks

by

Sasi Kiran Vepanjeri Lokanadha Reddy

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Master of Computer Science degree

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In traditional Wireless Sensor network's (WSN's), the sink is the only unconditionally trusted authority. If the sink is not connected to the nodes for a period of time then the network is considered as unattended. In Unattended Wireless Sensor Network (UWSN), a trusted mobile sink visits each node periodically to collect data. This network differs from the traditional multi hop wireless sensor networks where the nodes close to the sink deplete their power earlier than the other nodes. An UWSN can prolong the life time of the network by saving the battery of the nodes and also it can be deployed in environments where it is not practical for the sink to be online all the time. Saving data in the memory of the nodes for a long time causes security problems due to the lack of tamper-resistant hardware. Data collected by the nodes has to be secured until the next visit of the sink. Securing the data from an adversary in UWSN is a challenging task. We present two non-cryptographic algorithms (DS-PADV and DS-RADV) to ensure data survivability in mobile UWSN. The DS-PADV protects against proactive adversary which compromises nodes before identifying its target. DS-RADV makes the network secure against reactive adversary which compromises nodes after identifying the target. We also propose a data authentication scheme against a mobile adversary trying to modify the data. The proposed data authentication scheme uses inexpensive cryptographic primitives and few message exchanges. The proposed solutions are analyzed both mathematically and using simulations proving that the proposed solutions are better than the previous ones in terms of security and communication overhead.

Acknowledgements

First of all, words cannot express my gratitude to my advisor Prof. Amiya Nayak. This work would not have been possible without his incredible guidance. His technical advice and inspiration were the source of ideas in this work. His generous help, both spiritual and financial, enabled me to concentrate on my goal without other worries or fears. His energy and positive enthusiasm towards both work and life will influence the rest of my professional life. I just feel so lucky to have been one of his students. If there is an "ideal" advisor, it must be Prof. Amiya Nayak.

I must also thank Dr. Sushmita Ruj, one of my co-authors, for leading me to the right path and her consistent help and encouragement over the duration of my study. Her insightful comments and feedback have been extremely useful to me.

I am forever indebted to my parents for their unconditional love and for always putting the needs of their children above their own. They have been my greatest source of strength through all these years.

Contents

List of Tables

List of Figures

Acronyms

WSN	Wireless Sensor Network
UWSN	Unattended Wireless Sensor Network
DS-PADV	Data Survivability Against Proactive Adversary
DS-RADV	Data Survivability Against Reactive Adversary
CH	Cluster Head
MEMS	Micro-electronic Mechanical Systems
MAC	Message Authentication Code
ADV	Adversary
TRNG	True Random Number Generator
PRNG	Pseudo Random Number Generator
DISH	Distributed Self Healing
POSH	Proactive co-Operative Self-Healing
SBR	Sensor Based Restriction
DBR	Data Based Restriction
SDBR	Sensor and Data Based Restriction
DDHC	Dual Direction Hash Chain scheme
ECC	Elliptic Curve Cryptography
TRE	Timed Release Encryption
RJ	Random Jump

Chapter 1

Introduction

Recent advancements in micro-electronic mechanical systems (MEMS) technology paved way for the development of a large number of tiny, low cost and low power sensors that can communicate wirelessly [7]. Sensor network tends to be very application sensitive, they are deployed and utilized in various civilian and military settings. Real time applications include sensing, health monitoring, data acquisition in hazardous environments, habitat monitoring, traffic control, etc., [2, 3]. Each sensor is equipped with a sensing circuitry to measure the surrounding environment and transform the parameters into electrical signals. By processing these signals some properties about the objects located or the events happening around can be revealed. Typically a sensor is severely constrained in terms of energy reserves and computational capacity. Each sensor is battery powered, has a low memory, and a low communication module capable of short range wireless communication. Since sensors are used in many mission critical operations, security must pervade every aspect of the design of the wireless sensor network. Providing security is challenging in these networks due to the resource limitations of the sensor nodes A straight forward way of collecting the data from the nodes is to allow the nodes to forward the data to trusted authority

commonly known as sink or the collector through intermediate nodes as shown in Fig. 1.1. Sensor networks often have one or more sinks. The sinks are typically a gateway to another network, a storage centre, or an access point for human interface.

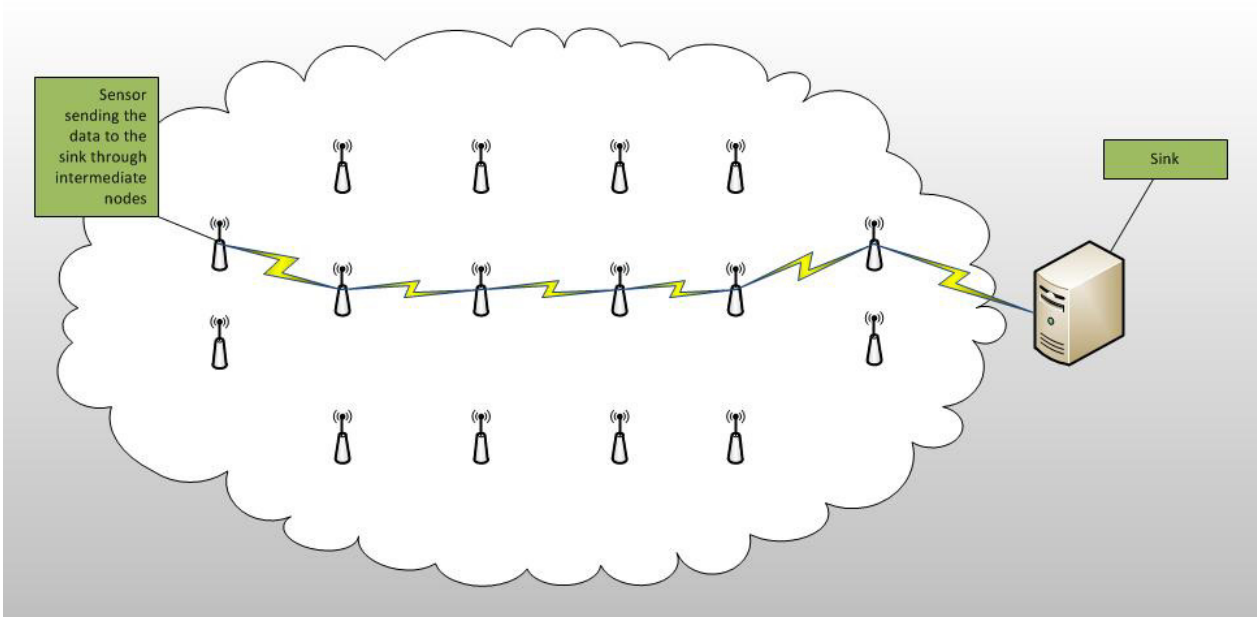

Figure 1.1: A typical WSN forwarding the data to the sink

1.1 Unattended Wireless Sensor Networks

Unattended Wireless Sensor Networks (UWSN) [4] (Fig. 1.2) is an hybrid type of WSN where data sensed by the sensors is not collected continuously by the sink. Data has to be secured by every node until the next visit of the mobile sink. This inability to communicate with sink might be for reasons such as: limited transmission ranges, power constraints or signal propagation problems [26]. The concept of UWSNs with a mobile sink looks realistic if we consider the environments where the sensing field is too far from the base station and sending data through intermediate nodes may result in weakening the security (e.g., intermediate nodes may modify the data)

or increase the energy consumption of the nodes close to the base station. In normal multi-hop Wireless Sensor Networks, power of the nodes placed near the sink will be depleted earlier than the other nodes. This is because all the nodes have to transmit the data to the sink through the nodes placed near the sink. An UWSN can be used to save the battery of these nodes and as a result increase the lifetime of the network Unattended environments as mentioned in [14, 4] include sensor networks for monitoring sound and vibration produced by troop movement, airborne sensor networks for tracking enemy aircrafts, LANdroids [25] which retain information until soldiers move close to the network, sensor networks for monitoring nuclear emissions, national parks for firearm discharge and illicit cultivation, etc.

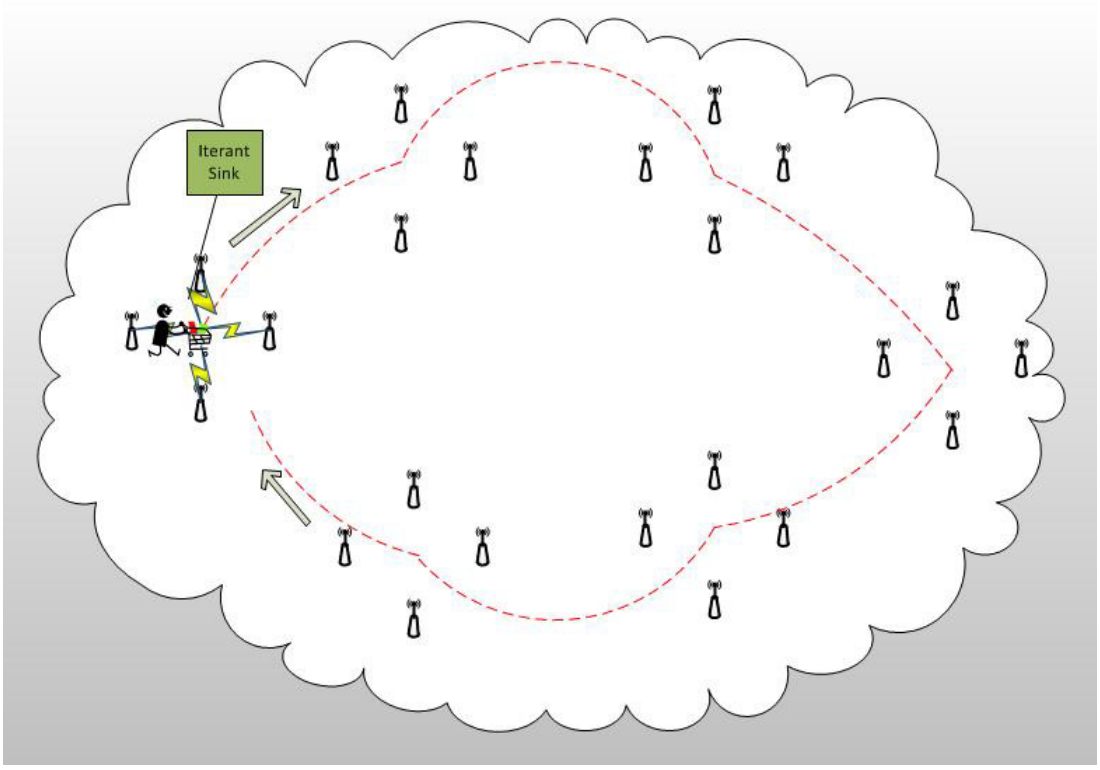

Figure 1.2: Unattended Wireless Sensor Network with Mobile Sink collecting the data

1.2 Motivation

In many real world applications, critical data is collected and stored in the unattended nodes in hostile environments. The data should be accumulated until the next visit of the sink. The unattended nature of the network and the lack of tamper resistant hardware increases the susceptibility of attacks over the data collected by the sensors [29]. Since the UWSN scenario is different from the traditional WSN's, defence solutions from WSN security literature are not suitable for coping with a mobile adversary in UWSN's. The sensors battery power is more limited compared to the battery power of the nodes in MANET's and hence the security protocols for MANETs are not effective for UWSNs.

Security needs should be taken into account to ensure data protection (also called *data survivability*) in these sensors at the time of design. Distributed security schemes are preferable over centralized schemes, because centralized schemes are prone to single point failure. Data security and data authentication is a major concern in UWSNs. Most cryptographic techniques provide data authenticity and integrity but do not ensure data survivability. This implies that if an adversary compromises a sensor and destroys the data contained therein, the data is lost forever. The other drawback of cryptographic schemes is that they are computationally expensive for resource-constrained sensor nodes. For these reasons, non-cryptographic techniques can be considered over cryptographic ones. In past few years, techniques for data authentication were proposed in [28, 29] and cryptographic techniques for data protection were proposed in [1, 4, 13, 17, 11]. All these schemes assume that the sensors are static between successive visits from the sink. However, this assumption is not practical in many real world applications and hence should be relaxed and allow nodes to move between two consecutive visits from the sink.

Another important concern in UWSN's is a mechanism is needed to ensure that the data received at the sink is authentic. The main goal of some of the adversaries is to inject fraudulent data into the information collected by the nodes and remain undetected. The mobile adversary is capable of compromising k out of n nodes in each round and it can also switch to different set of k nodes per round. Authentication schemes for UWSN against a mobile adversary presented in [28] and [29] guarantee good security but suffers from high communication cost relative to the level of security achieved.

The problems identified above motivates us to find the best possible for securing the data in UWSN's.

1.3 Contributions

1. We propose two non-cryptographic schemes (DS-PADV and DS-RADV) suitable for data protection against two types of adversaries with less communication and memory overhead with easy routing scheme. This work has been published in IEEE Globecom, 2012 [62].

2. We propose a scheme to achieve data authentication against the data modification attempts made by the mobile adversary. This work has been accepted in IEEE WCNC, 2013.

The proposed data survivability schemes protect data in UWSN against two types of mobile adversaries: proactive and reactive. Proactive adversary starts compromising the nodes before it identifies the target node. Reactive adversary remains inactive until it locates the target node where the data has to be erased or changed. Our first scheme protects against proactive adversaries and is known as *DS-PADV*, and the

second scheme protects against reactive adversaries and is known as *DS-RADV*. Proposed techniques make use of data replication. In DS-PADV, sensors on receiving the data forward it to one of its neighbours selected at random, and the neighbours in turn forward the data to its neighbours located in the opposite direction of the source node. Data is forwarded for a fixed number of hops, and a copy of data is saved in all the intermediate nodes. Similar approach is followed in DS-RADV but with less memory overhead. In these approaches, when the adversary tries to modify the data it can be easily detected by the sink as the same data will be present in some other nodes in the network. Thus, the sink can retrieve data from those nodes. A mobile node do not know the identity of the nodes where its replicated copies are stored, so there is no chance for an adversary to know the identities of the nodes which have stored the data. Routing is done based on the GRoVar protocol proposed in [6] wherein data is transmitted only to the neighbors in a specific knowledge range using beaconless routing techniques. We analyze our schemes mathematically and using simulations and compare with existing schemes like [1, 28, 29]. Our schemes increase the probability of data survival and have lower communication cost and memory overhead than the known techniques [1, 28, 29]. The communication cost is $O(tN\sqrt{N})$ for [28] and $O(N\sqrt{N})$ for [29], whereas for DS-PADV and DS-RADV, the costs are $O(Nh)$ and $O(tNh)$ respectively, where $h \ll N$ and t is an arbitrary constant.

We also introduce a collaborative authentication mechanism against a mobile adversary focusing on maximizing security while reducing the communication overhead. In the proposed data authentication scheme, we focus on implementing a mechanism through which sink is able to detect any kind of modification made by the adversary. Multiple message authentication codes (MAC) sent by the nodes are hashed at a single node (cluster head) and the hashed value is replicated to avoid the single point of failure. Routing in this scheme is done based on the GRoVar protocol proposed

in [6] wherein data is transmitted to the neighbours in a specific knowledge range using beacon less routing techniques. Through this scheme, data integrity can be achieved with little overhead which is less than the existing authentication schemes. For the adversary to successfully tamper the data of a single node, it should be able to break the MAC created by that node, break the hash value created by the cluster head and should be able to destroy the replicated copies of the hash value sent across the network. Proposed approach greatly increases the probability of detecting the modification attempts made by the adversary with a minimum overhead. Proposed scheme is analyzed and evaluated based on survival probability, communication costs and memory overhead. The communication cost is $O(t\sqrt{N})$ per node for the scheme in [28] and $O(\sqrt{N})$ per node for the scheme in [29], whereas for the proposed scheme, cost is $O(t\sqrt{N})$ per cluster (where N is size of the network and t is an arbitrary constant) which proves that the proposed scheme is better than the existing schemes.

1.4 Organization

This thesis is organized as follows:

- Chapter 2 is a background on the UWSN, as well as security research applied to the field. This chapter presents the work done in the fields related to this thesis. We present work that has been done on data authentication, data survivability, location privacy and secure data aggregation. We also present the challenges faced by UWSN to achieve the security goals.

- Chapter 3 present the proposed data survival schemes along with the evaluations and simulation results. We also explain the network and adversarial models for the proposed data survival schemes. Proposed schemes are compared with the

existing ones mathematically and using simulations.

- In Chapter 4, we introduce a different data authentication protocol that increases the probability of detecting the modifications made by the adversary on the data. Once again we compare the work with the existing data authentication schemes.

- Chapter 5 then provides conclusion and discusses possible ways in which the work in this paper can be improved and expanded upon.

Chapter 2

Background and Related Work

2.1 Wireless Sensor Network Security

Since the sensors are low cost and can be deployed easily in any environment, they have wide range of applications in both civil and military settings. Considering the nature of applications sensor are used in, sensors transmit mission critical information over the network and hence there is a need for security services like authentication, encryption, key management, etc., Because of the low cost of sensors, lack of tamper resistant hardware and meagre resources, security in sensor networks presents a number of formidable and unique challenges. A good amount of research has been done in recent years dealing with various aspects of sensor network security, such as data authentication/privacy, key management, secure routing and secure aggregation, as well as attack detection and mitigation. Most of the research done rely on the continuous presence of the sink which is not the case in UWSNs.

The concept of UWSNs with a mobile sink looks realistic if we consider the environments where the sensing field is too far from the base station and sending data through intermediate nodes may result in weakening the security (e.g., intermediate

nodes may modify the data) or increase the energy consumption of the nodes close to the base station thereby reducing the lifetime of the network (Fig. 2.1). In general, the security requirements in UWSNs include integrity, confidentiality, authenticity, and availability of generated data. Data security or confidentiality is one of the major security issues in UWSNs. Encryption is one straight way of achieving data security but it comes with computational cost. Public key cryptography solves most of the data security issues, and it is considered as computationally expensive in the past. So, for a long time research was focused on symmetric key cryptography until recently [57, 58, 59] when it is proved that the public key encryption is feasible on low cost sensors. According to the results in [60], wireless communication in sensors causes more energy loss than performing computations for encryption. Furthermore, the new generation of ultra-low-power micro controllers, such as the 16-bit Texas Instruments MSP430 [61] were used in sensors. They can execute the same number of instructions at less than half the power required by the 8-bit micro controllers used in some of the sensors. So, communication-efficient cryptographic schemes with less communication overhead will be more efficient in success of sensor applications. The rest of the chapter provides some background and related work on UWSNs.

2.2 Background

2.2.1 Applications of UWSNs

Applications of UWSNs as mentioned in [1, 8, 14, 19] include

1. Network of nuclear emission sensors deployed to monitor any potential nuclear activity.

2. Underground sensor network aimed at monitoring the troop movements.

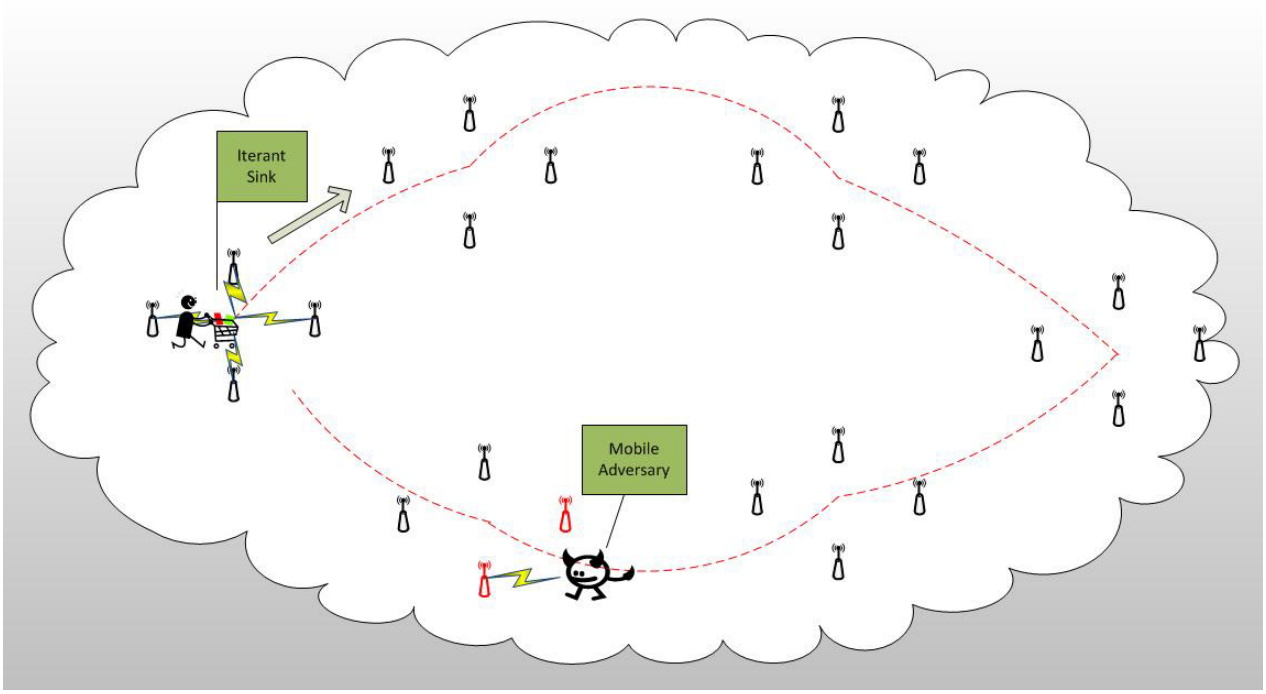

Figure 2.1: Unattended Wireless Sensor Network with Mobile Sink and adversary compromising nodes

3. Airborne sensor network tracking fluctuations in air turbulence and pressure to detect enemy aircrafts.

4. Monitoring system to detect poaching in a national park.

5. Submarine sensor network to track the movements of animals.

6. Monitoring underground or underwater oil pipelines to detect the leakage of oil.

7. A WSN mounted on a tree to monitor firearm discharge or illegal cutting of the trees.

8. A sensor network deployed in the international border to monitor illegal crossings.

9. US Advanced Research Projects Agency (DARPA), initiated a new research

program called LANdroids [25]: LANdroid nodes are deployed in hostile environments to collect critical information until the soldiers approach the network.

In general, UWSNs are needed in places where nodes are deployed in hostile or inaccessible environments. Sensors might be deployed in hostile environments where sink is not secure; if the sink gets compromised then potentially valuable data can be lost. In the environments where the number of sensors deployed is large, one sink might not be enough to cover the entire area. In some environments, sink might be physically present in the network but it needs to conserve its energy by switching itself off periodically [14]. The size of the deployment area, inaccessibility and difficulty for the sink to hide motivates the need for UWSNs.

2.2.2 Mobile Adversary

Initial investigation on mobile adversary was done in [46]. This adversary can compromise a fixed number of nodes in a round and migrate to a new set of nodes in the next round. Given enough rounds, the adversary can compromise all the nodes in the network for which the security of the network will be undermined. There is an entire branch of cryptography, called *proactive cryptography*, for developing cryptographic techniques to provide security against a mobile adversary [47, 48].

In UWSNs, adversaries (ADV) are the ones that try to modify or destroy the data. The adversaries are mostly classified into two types: proactive and reactive. Proactive adversary starts compromising the nodes before it identifies the target node. Reactive adversary remains inactive until it locates the target node where the data has to be erased or changed. Different flavours of these adversaries are discussed in literature [1, 14, 27, 21, 22]. Some of them are:

- Lazy: This type of adversary compromises a set of nodes at the beginning and

stay put. This attack strategy may not be sensible when the data is moved between nodes.

- Frantic: This adversary moves from one set of nodes to an other set nodes at the beginning of each interval provided these two sets does not have any common nodes.
- Smart: Smart adversary selects two sets of nodes and simply alters the control between these sets at each interval.
- Curious: It tries to acquire information about the data as much as possible or it can also focus on the specific data which is of high importance.
- Polluter: This adversary does not alter any existing data but injects fraudulent data into the network to mislead the sink.
- Search-and-Erase: Aims to erase the data before it reaches the sink. If the sink is programmed to tolerate some missing data then the adversary can remain stealth even after erasing the data.
- Search-and-Replace: If the sink is programmed to detect the missing data then in order to prevent the data from reaching the sink the ADV has to replace the target data with some concocted value.
- Eraser: It erases as much data as possible, so it can be easily detected.
- Curious: Aims to learn as much data as possible. It is read-only type adversary, it does not erase the data. This type of adversary might be interested in learning some specific data which might be of high importance.

Most of the adversaries remain in the stealth mode. In few cases, based on the goals of the adversaries they choose not to remain undetected. However, the adversary

tries to remain undetected all the time. Since the network is unattended most of the time, it is also possible for the adversary to physically destroy the sensors.

2.2.3 Security Goals

While dealing with security in UWSNs, we mainly focus on achieving some or all of the following security goals:

- Forward Security: The compromise of a secret in one round should not lead to the compromise of the secrets in the rounds preceding compromise. Forward security is critical in UWSNs so that even if the adversary can compromise the current key it is infeasible for it to generate the previous keys using the current key. The adversary cannot even forge the authentication tags for the data generated and authenticated before compromise.

- Backward Security: The compromise of a secret at any time should not lead to the compromise of the secrets to be used in the future. If an adversary obtains the current status of the node, it cannot decrypt the data generated and encrypted after compromise. Adversary should not able to forge the authentication tags for the data generated and authenticated after compromise.

- Data Confidentiality: Confidentiality ensures that the information is inaccessible to unauthorized users. In UWSNs, data in the sensors should be encrypted in a way that it can only be read by the sink. In some scenarios, data from the nodes will not be sent to the sink in a single hop. Sink visits a point for data collection which can be few hops away from the node and the data has to be sent through other nodes. In these cases the intermediate nodes should not be able to read the transmitted information. Data confidentiality also prevents the

read only adversary from reading the stored data in the compromised node's memory.

- Data Integrity: Data integrity protects against unauthorized alteration of the data. Data integrity can be achieved only if the network has the ability to detect the manipulations done to the data by unauthorized parties, i.e., insertion, substitution and deletion. Data integrity protects against the "search and replace" and "search and erase" types of adversaries.

- Data Authentication: Authentication applies to both nodes and data. It ensures the identity of the node with which it is communicating i.e., the two communicating parties can identify each other. Information delivered through the network should be authenticated with respect to the generation time, date of origin, origin etc., Data origin authentication also provides data integrity as the message modification can be detected.

2.2.4 Security Challenges

While designing security schemes suitable for a wireless sensor network, resource constraints should be considered. A sensor has limited battery life, computation power and memory. For all the security schemes, we need to consider the storage and communication overhead incurred by the schemes on the nodes. It is crucial for any security scheme to be both storage and communication efficient. In UWSNs since the nodes are deployed in hostile and unattended environments, (Fig. 2.2) they become an easy target for an adversary.

As mentioned in [18], the first question to be asked while designing a security scheme for UWSNs would be "How can a disconnected sensor network protect itself from an adversary ?" (Fig. 2.2). The nodes in the network should be able to prevent

the adversary from learning the sensed data even though the adversary can break into the node and learn all its secrets. In general, the greatest challenge in UWSN is to secure the data long enough until the arrival of the sink.

Tamper resistant hardware [10] can be used to avoid many attacks in the network but they add significant cost to the sensor hardware and hence not a practical approach. Current message authentication code schemes and signature schemes are either based on a secure hardware or a trusted online third party (sink); these techniques are not feasible in the case of UWSNs. So, the security schemes must be inexpensive (in terms of both hardware and computation), simple, and efficient.

The authentication schemes designed should hide the data origin, data content, and time of data collection [26]. Another challenge with the UWSNs would be finding a way to combine light weight cryptographic schemes involving node collaboration with bandwidth-conscious data migration methods [19].

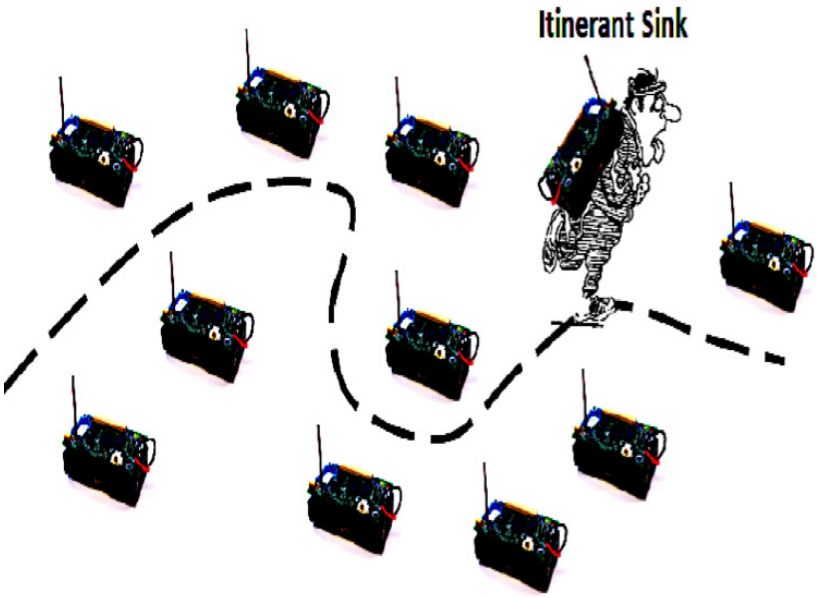

Figure 2.2: Unattended Wireless Sensor Network with Mobile Sink [29]

2.2.5 Possible Attacks on Nodes

To appreciate the challenge of securing a network, all the possible threats should be considered. Threats in UWSNs can be broadly classified into four categories: attacks on the data in the nodes, routing attacks, cloning attacks, and physical attacks. In normal WSN's since the sink is online for the whole time, it is feasible for the sink to detect most of the attacks in the network, but this line of defence is not possible in the case of UWSNs. Some possible attacks on the nodes are as follows [14]:

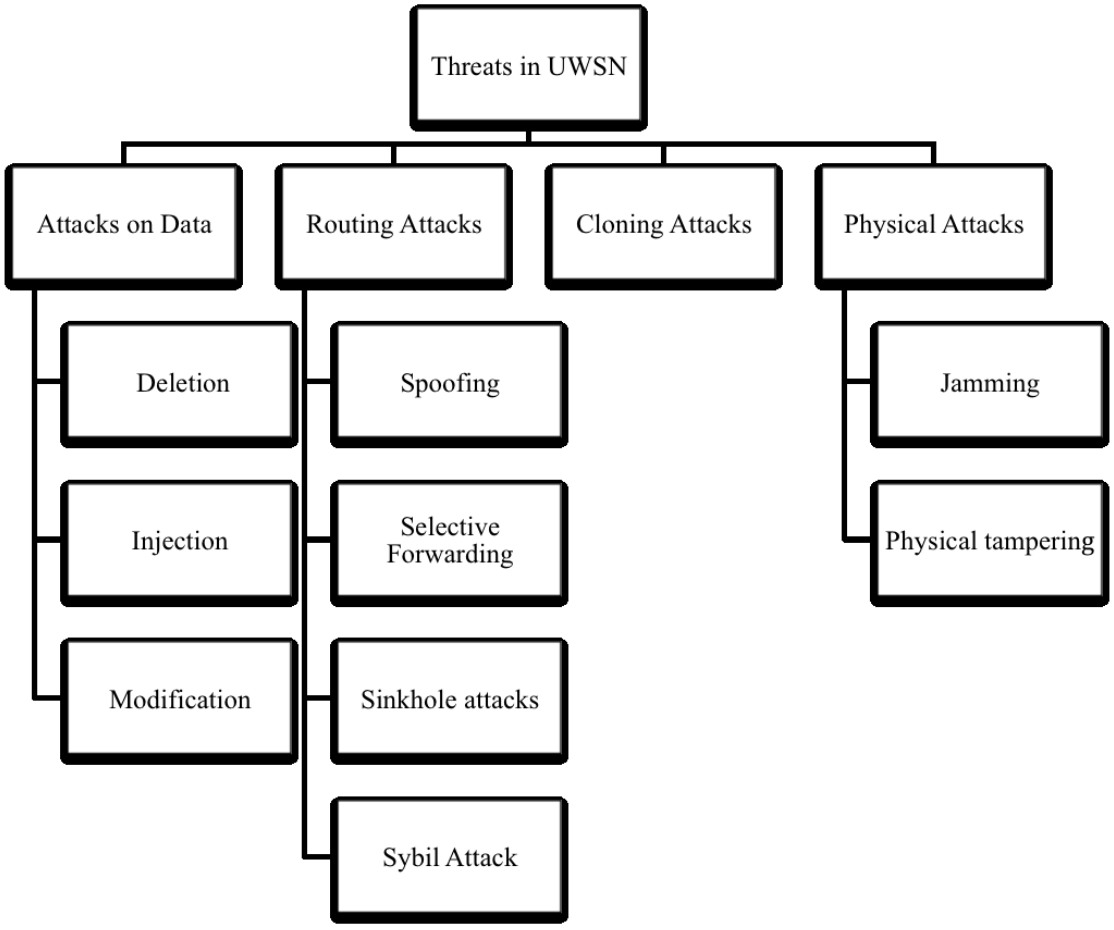

Figure 2.3: Threats in UWSNs

1. Attacks on Data: Since the network is unattended, the nodes have to save the data in their memory for a long time (i.e., until the next visit of the sink). This

inability to offload the data makes the data an easy target for the adversary. The adversary can delete a critical information or inject false measurement into the memory or modify the data.

2. Routing Attacks: UWSNs use multi-hop routing to transmit the data to the mobile sink (when the sink comes into the range of the node) or to send the data replicas and Message Authentication Codes (MAC's) of a node to the other nodes. All the data transfer is done wirelessly, and this incurs routing attacks by the adversary. Some possible routing attacks [49] as follows:

   (a) Spoofing: Spoofing is a direct attack against a routing protocol. The adversary can spoof, alter or replay routing information between the nodes. By doing this the adversary can create routing loops, extend or shorten service routes, generate false error messages and increase end-to-end latency.

   (b) Selective Forwarding: General assumption when a message is forwarded through multi-hops is that the nodes will faithfully forward the received messages. In selective forwarding attack, malicious node behaves like a black hole, refuses to forward all the messages and drops certain messages. This attack can remain undetected as the malicious node suppresses only critical messages and forwards the rest to limit the suspicion.

   (c) Sink Hole Attack: In the sink hole attack, all the traffic is attracted to a compromised node. This attack works by making the compromised node look attractive to the surrounding nodes. Sink hole attacks can enable other attacks like selective forwarding in the network.

   (d) Sybil Attack: In sybil attack, a node presents the identity of multiple nodes in the network. The sybil attack targets the fault tolerant schemes such as

multi-path routing, distributed storage, disparity, topology, replicas storage partitions. Sybil attack can be avoided by proper authentication and encryption techniques.

3. Cloning: Since the nodes in the network are deployed in hostile environments, they are vulnerable to capture and compromise. The attacker can copy the ID and data of the captured node into a new node and deploy it in the network. Node replication can severely disrupt the network performance. Data can be corrupted or misrouted. Node can send false reading to the sink. If the attacker can insert clones at specific segments of the network then it can gain access to a specific segment of the network.

4. Physical Attacks: In physical attacks, the attackers manually capture the nodes and reprogram them. The main advantage of this attack is quick and easy [7]. In UWSNs, the unattended and distributed nature of the network make them highly susceptible to the physical attacks. Unlike the other attacks mentioned above, physical attacks can destroy the nodes permanently and the losses occurred are irreversible. Two types of physical attacks are possible:

   (a) Jamming: In this attack the channel will be jammed with an interrupting signal. To jam a node or set of nodes in the network, a simple radio signal can be transmitted which interferes with the signals used by the other sensor nodes.

   (b) Physical Tampering: Sensor nodes are vulnerable to physical harm or tampering. Tamper resistant hardware can be used to avoid this kind of attacks but it is not practical use such an expensive hardware for the entire size of the network.

2.2.6 Network Assumptions

We assume that the UWSN consists of many homogenous low cost sensors distributed over a wide geographical area and one or more mobile sinks. To make the sensor networks economically viable, sensor must be inexpensive and as a result they have very limited resources in terms of computation, memory and energy. A sink is usually a powerful source which acts as a gateway for other network or as an access point. In normal sensor networks, sink is assumed to be online all the time to collect the data and monitor the network to detect the abnormal behaviour or attacks.

As explained before, since the network is "unattended", nodes are not under constant supervision. All the nodes have to store the data until the sink collects it. Sensors obtain measurements from the environment, data collection can be event driven (reacting to the changes in the environment) or scheduled (programmed to collect between time intervals). Time is divided into equal and fixed number of rounds and every node collects a data item in a round. It is assumed that the nodes have sufficient power and memory to store the data until the next visit of the sink. It is also assumed that each node is capable of performing cryptographic hashing, public key encryption and symmetric key encryption. The intervals between the sink visits can be short or long, depending on the nature of the network.

2.3 Related Work

2.3.1 Data Survivability in UWSNs

Data survival can be defined as the requirement that the data in the nodes should be protected while the base station is not available to send the data. A data item is said to be survived if and only if it is successfully sent to the base station with out

being modified by any adversary. In this context, the threat is that the adversary can compromise a set of nodes to destroy the critical data by modifying them or deleting them from the node's memory.

Pietro *et al.* in [1, 14] first identified the data survival problem in Unattended Wireless Sensor Networks (UWSNs). They suggested simple data survival defence strategies like DO-NOTHING, MOVE-ONCE and KEEP-MOVING against different types of adversaries. In DO-NOTHING, sensors do not do anything, they just wait for the sink to collect the data. In this case, the adversary can easily succeed by compromising the node and erasing the data. An alternative for this strategy is MOVE-ONCE where the adversary move the data to a randomly picked sensor. Here the data is moved only once and the sink collects the data from new location. A better alternative is KEEP-MOVING where every sensor continuously moves data to a randomly chosen sensor at every interval. This work also provides an analysis on effects of degree of replication and encryption on data survivability.

In [14, 19], some challenges and techniques against the attacks of mobile adversaries in UWSNs were explained. This work evaluates the effects of mobile adversaries against various defence strategies.

1. Search-and-Erase: For the search-and erase adversary, the data has to be erased before it reaches the sink. The nodes does not know the target data and hence all data should be protected equally. In DO-NOTHING strategy, nodes store the data locally and wait for the sink to collect the data. So, it is easy for a search and erase adversary to locate the data and erase it. In the MOVE-ONCE strategy, the data is moved to a randomly selected sensor and it is equally likely for any sensor to possess this data. So, in each round the adversary selects a set of sensors different from the set selected in the previous round. Assuming that

the adversary can compromise k nodes in each round, adversary visits every node once in n/k rounds. So, on an average adversary it takes $n/2k$ rounds for the adversary to find and delete the data. In the KEEP-MOVING strategy, since every node keeps moving its data, the best bet for the adversary is to switch between two independent sets of nodes (FRANTIC) and wait until one node among these sets receive the data. KEEP-MOVING strategy involves a lot of storage and communication costs than the other two strategies. MOVE-ONCE performs better than KEEP-MOVING for the first n/k rounds with less communication overhead. In the case of replication, the data survival strategy is very high compared to the moving strategies. This is because one copy among the r replicated copies is sufficient for the sink. Memory and communication overhead is high in the case of replication, the data survival strategies can be chosen based on the application and the available resources. Cryptography is also a good choice to prevent the adversary from erasing the data. By encrypting the data, the adversary cannot find the target data which forces it to make a guess to choose the target data. Forward and backward security should be provided with encryption. So, every node should do the key evolution and should have a source of randomness such as TRNG (TRNG produces theoretically independent values by extracting randomness from physical phenomena [19]. TRNG is both forward and backward secure, i.e., by learning a current state value it is not possible to compute the previous nor the following values) to perform randomized encryption. To increase the data survival chances, encryption can be combined with data migration strategies explained above. (In randomized encryption [9] a non-deterministic probabilistic algorithm is used for encrypting the data making it infeasible to determine the corresponding plain texts. In the public key randomized encryption, a random number is ac-

cepted as input along with the plain text to the encryption function. In the symmetric key randomized encryption, different keys can be used to achieve the randomness)

2. Curious: As mentioned earlier, curious adversary tries to read as much data as possible. Moving cannot be considered as a good defence strategy against a curious adversary, it is because if the adversary has the ability to read the data then regardless of the location the data is insecure. Randomized encryption [9] (public or symmetric) along with key evolution can be used to protect the data by providing both forward and backward security. Since the range of sensor readings (plain texts) is very small if the randomized encryption is not used then it is easy for the adversary to match the plain texts with the calculated cipher texts. A True Random Number Generator (TRNG) is needed to provide the randomness needed for the encryption for the reasons mentioned earlier. However, randomized encryption alone is not enough because if the adversary compromises the node, it knows the key and it can read the data no matter where the data is stored. So key evolution should be done to achieve the forward security. It can be achieved by periodically evolving the initial key shared with sink using a one way hash function. So, public key encryption with TRNG gives desired level of protection. It should also be noted that public key encryption results in high communication and storage costs that the symmetric encryption.

3. Eraser: For the DO-NOTHING strategy, the best strategy for the adversary is to move in a round robin fashion deleting k messages in every round. So, the nodes visited during the round one were the nodes which have not been visited for a long time. Since there will be a constant number of new measurements introduced every round, there will be a stable number of messages remaining in

the network. In the MOVE-ONCE strategy, the data moved will be deleted after n/k rounds. In the KEEP-MOVING strategy, the adversary cannot identify the set of sensors having a largest number of data items as the data will be moving constantly in the network. So, the adversary instead of following a round robin technique switches between two independent set of nodes. This way there is a high chance that the survival probability of KEEP-MOVING is less than the DO-NOTHING and MOVE-ONCE strategies. This is because if the data does no move to other sensors, adversary can only delete the items present in the storage of the compromised node but by moving the data, the adversary deletes the data stored in the node as well the data received from other nodes. Replication has greater effectiveness than the other survival strategies and the survival probability against the Eraser increases with the number of replicas. So, it can be concluded that the best survival strategy against the Eraser will be DO-NOTHING as the replication results in greater communication and memory over head.

4. Search-and-Replace: Search-and-Replace is similar to search-and-erase except that without erasing, the data must be replaced with an other authentic data so that the adversary can remain stealthy. As in the case of search-and-erase, migrating data with forward and backward secrecy is a very good defence strategy against search-and replace adversary nevertheless there are some other issues to be considered. Since the adversary is stealthy and focuses on replacing the data, there should be an authentication mechanism such as Message Authentication Codes (MAC) [23] to detect if the data has been changed. The MACs should be forward and backward secure and there are some forward authentication schemes discussed in literature[20, 24]. The forward security for encryption and

authentication are not similar, the key evolution process for authentication cannot be randomized. If the key evolution is randomized, it is impossible to verify the signature. So the key evolution method should be same with the nodes and the sink that verifies the signatures. A one way hash function can be used for the key evolution. Considering the unattended nature of the network, collaborative techniques are more recommended to develop authentication schemes.

2.3.1.1 Cryptographic Techniques for Data Survivability

Encryption can also be used to protect the data but it involves certain non-negligible costs such as key management and computation involved to perform the encryption [1]. Symmetric cryptography is considered as a good choice for sensors and public key cryptography is considered too expensive in terms of energy and computation costs. However, symmetric cryptography is not a good choice, unless key evolution is employed to achieve forward security and also public key cryptography is ineffective if it is not used with a random number [14]. Using encryption, nodes can hide the data such that the adversary cannot identify a specific data. Identity of the nodes that collected the data can also be hidden using encryption. However, encryption cannot be considered as an option if the goal of the adversary is to completely erase the data in the node. In case of public key encryption, sink has a public key which is known to all the nodes. Nodes after collecting the data, encrypts the data using the sink's public key. To decrypt this sink's decryption key should be used. In this case, it is difficult for the adversary to detect the target data. Adversary can try to detect the target data by encrypting a sample data using sink's public key, the target data can also be replaced by different data since the adversary has the knowledge of the public key. To avoid this a one-time random number can be used along with the public key to encrypt the data [11]. This way it is not feasible for the adversary to

distinguish the encrypted data and also re-create the same cipher text.

Two cryptographic protocols (super-encryption and re-encrytion) were proposed in [13] considering the security properties and efficiency of energy and storage. In these protocols data collected is encrypted and moved to a different node. By moving the data, communication and memory overheads caused by replication can be avoided. Super-encryption is a symmetric scheme where each node has its own key shared with sink. When a node (s_i) senses data, it uses its own key (k_i) to encrypt the data and move the encrypted data to a randomly selected node(s_j). The node receiving the data (s_j) applies another round of encryption using its own key (k_j). This node (s_j) either saves the data in its memory or move it some other random node where the same procedure is repeated again. If the number of rounds of encryption increases then the adversary has to make a great effort to find the location of the data and destroy data. Super encryption scheme cannot be effective against proactive adversaries as these adversaries have the ability to memorize many keys at a time. Re-encryption uses asymmetric model and this scheme is effective against the proactive adversaries. The goal of this approach is to convert the cipher texts of one key to another and this approach is costlier than the symmetric cryptographic scheme. In the re-encryption, a node (s_i) selects an other node randomly and sends the encrypted data (C_i) by encrypting it again with the public key (k_j) of the randomly selected node(s_j). So the node s_j receives the cipher text C_j. Since all the nodes can decrypt the data, data aggregation is feasible in this scheme.

A self healing protocol, called DISH (Distributed Self healing), was proposed in [18] which uses the key evolution and sensor cooperation to achieve both forward and backward security. DISH does not guarantee data security but gives a probabilistic tunable degree of secrecy depending on the power of the adversary (i.e., the number of nodes that can be captured by the adversary in a given time), number of messages

transferred between nodes and the time between successive visits of the sink. In this protocol, every node shares a key with the sink initially. In this work, nodes are categorized in three sets namely healthy, sick and occupied. A healthy node is the one that is not compromised by the adversary at present, so the secrets of the healthy node are not known to the adversary. Sick nodes were the nodes compromised by the adversary in the past i.e., the secrets of the sick nodes are known to the adversary and the occupied nodes were the ones that are currently compromised by the sink. Sick sensors use the random secret key from the healthy sensors to calculate the new round key. The random secret sent from the healthy sensors will be an input to the one way hash function along with the old key to calculate the new round key. In this protocol, a healthy sensor always remains healthy until it is directly compromised by the adversary and a sick sensor becomes healthy if it receives the secret from a healthy sensor. In DISH, the ratio of sick to healthy sensors stabilize within a few rounds.

In [17], a self healing cryptographic key distribution protocol, called POSH (Proactive co-Operative Self-Healing in Unattended Wireless Sensor Networks), was proposed to secure the nodes against a mobile adversary. The mobile adversary assumed in this work is capable of compromising k out of n nodes in each round. The main idea in POSH is that every sensor acts a source of randomness for its neighbouring sensors. If the key of a node is compromised then it can compute a new key unknown to the adversary by obtaining randomness from its neighbours whose key is not compromised by the adversary. All the nodes in the network are divided into three sets namely *Red*, *Green* and *Yellow*. Nodes in the red set are currently compromised by the adversary, so the secrets of these nodes are know to the adversary. Green sensors are those that have been compromised by the adversary or the ones that have regained their security using through the POSH scheme. Yellow nodes are the secure

nodes, secrets of these nodes are unknown to the adversary. In every round, each sensor produces t random values and sends each value to a randomly selected node. The nodes receive the random values and calculate a new key using the current key and the received random values. Since the keys are sent to randomly selected nodes, there is a possibility that a node does not receive any key or a node may receive keys from only green and red sensors. In this case, the new key computed cannot be secure. The calculated key can be considered secure only when at least one of the received keys is from a green sensor. This protocol does not guarantee the backward security to the nodes.

Pietro *et al.* in [11] proposed a self healing cryptographic protocol for mobile sensors which is both forward and backward secure. Forward security is achieved through one way cryptographic hash function and backward security is achieved using a random number generator. Here life time of a sensor is divided into three epochs (1) before compromise, (2) during compromise, and (3) after compromise. Sensors are categorized based on these epochs. The protocol uses randomized public key encryption [9] where one time random number is used for the encryption. As the protocol is self-healing, sensors themselves act as a source to provide randomness to other sensors. The random number for encryption is calculated from all the random values received from the node's immediate neighbours. The main idea in this protocol is if previously compromised sensor obtains randomness that is unknown to adversary then the sensor can regain security and its new secrets are unknown to the adversary. Here two mobility models are discussed for sensors: one is random jump where speed of the sensor is set such that sensor reaches the deployment area in one round, other is random waypoint where sensor covers a distance not greater than m in one round. Sensors use any of the above two mobility models to move to its new position. After reaching the new position, sensor broadcasts a random value based and its current

pseudo random number generator (PRNG) status to immediate neighbours. PRNG is an algorithm which takes a initial value called *seed* as input and produces a sequence of random values. The functions that are not computationally feasible to revert are used as PRNG's. PRNG's provides forward security but if the current state of the function is know then it is easy to compute the subsequent values, so PRNG does not provide backward security. Next, it uses all the values obtained from its neighbours along with its current secret to compute secret for next round. In this protocol, security of the network depends on the number of nodes that are not compromised in the network and also the density of the nodes in the field. If a node does not have any healthy nodes in its proximity during the key calculation for new round then the key calculated is not considered secure. This protocol involves overhead in terms of number of messages sent to calculate the key and it is not effective when the nodes are static.

In [26], authors provided some interesting results which proved that the level of security achieved by moving the data encrypted using public key cryptography is same as the level of security achieved by moving the data once and performing re-encryption. Reencryption of messages is done to achieve survivability. Message is first encrypted, stored and sent to a neighbour which is stored and reencrypted for further transmission. Reencryption follows a symmetric key approach. The authors believe that reencryption is better than encrypting once. However, this approach can be easily avoided and use a sink's public key for encryption. This eliminates the need for reencrypting data and thus saves computation overheads.

Rasheed *et al.* in [54] proposed two key pre-distribution schemes to establish a secure data transfer between the mobile sink and any sensor in the network. The first scheme is probabilistic generation key pre-distribution scheme [55]combined with polynomial pool-based key pre-distribution [56]. This scheme consists of three phases.

In the first phase (polynomial and generation key subsets assignments) polynomials and the keys are setup in the sensor nodes and the mobile sinks. In the next phase, mobile sink directly establishes key with the sensor nodes. This happens only if the mobile sink and the sensor node has common polynomial shares which were assigned initially. The last phase occurs only when the mobile sink fails to establish a secure link with the seor node. In this case, mobile sink has to discover at least one of the neighbours of the node which can establish a secure link with the sink so that the node can act as an intermediate node between the mobile sink and the node that needs to transfer the data. The second scheme proposed is Q-Composite Generation key Scheme Combined with the Polynomial pool- based scheme. This scheme is similar to the first one except that the generation Q keys overlap in this scheme. The mobile sink is initially loaded with a large number of generation keys so that it can find at least Q common generation keys with a node. By increasing the number of common generation keys, the resilience of the network against node capture can be increased. The proposed schemes in this work assigns too important role to the mobile sink. If the mobile sink is compromised or failed 90% of the keys cannot be used which results in very low probability of establishing security channels between the sink and the sensor nodes.

In [53], the authors proposed a key management scheme which uses the Blundo symmetric polynomial mechanism [50] to generate the keys between the nodes and to guard against the compromised nodes while updating the keys. This scheme uses the reverse hash chain for key updating and revocation, which reduces the privileges of the mobile sinks. The network assumed in this work contains three layers consisting of the mobile sinks, cluster heads and the sensor nodes. The nodes send their data to the cluster heads and the mobile sink collects the data from the cluster heads. The proposed scheme consists of three phases; key material pre-distribution, shared-key

generation and key update as well as mobile sink revocation. Base station initializes the mobile sinks, cluster heads and the nodes with Blundo symmetric polynomials and hash values (hash values are generated using reverse hash functions). Mobile sink and the cluster head will be initialized with a same Blundo polynomial to generate the keys between the mobile sink and the cluster head and a different Blundo polynomial will be initialized between the cluster heads and the nodes to generate the keys between cluster head and the nodes and also between the nodes. During the key establishment, the nodes exchange their hash values to calculate the keys, if the keys are not same then there should be at least one node failing to update its hash value and will be considered compromised and unsecured. The base station constantly updates the identities for the mobile sinks so that the cluster head can identify a legitimate mobile sink. This scheme prevents the compromised nodes from communicating with the normal static nodes.

Santos *et al.* in [52] proposed a data survival scheme based on secure multi party protocols and secret sharing. The proposed protocol works in two phases: (1) Initialization and sharing, (2) private key and share renewal. In the first phase, some shares of the secret are derived from the secret and distributed among the authorized nodes. The authorized nodes can collaborate to reconstruct the secret. The importance of using the secret sharing is that many nodes should collaborate and exchange data securely, so the trust is not in one node but many of them. Thus, the adversary should compromise a specific set of nodes to reconstruct the secret and break the system. The nodes collecting critical data can do this secret sharing, critical nodes can derive the secret into shares and distribute them to the other nodes. In the second phase, nodes change their private keys and renew their shares. In this scheme the task of the adversary is to compromise the nodes in a specific area to compromise all the nodes containing the shares of the key. To minimize the adversaries chance

of destroying the data, this scheme can be combined with the Pietros data survival strategies [14] MOVE-ONCE and KEEP-MOVING. The shares can be moved (not replicated) continuously to other sensors to increase the survival chances. In this scheme, the source node that initiates the secret sharing is considered safe which may not be practical in all the cases. If the node gets compromised before sensing the data, the system cannot be safe as the polynomial is also compromised.

Liu *et al.* in [51] proposed a asymmetric pre-distribution scheme with mobile sinks (mAKPS), this scheme facilitates the key distribution, protects the data in the sensors and restrict the privileges of the mobile sinks. In mAKPS, each sensor is pre-loaded with secret material and later when the nodes need to negotiate a secure session key, they get the public keying material from the mobile sink. Nodes who need to communicate request the public keying material from the sink when it comes to the vicinity of one of sensors that need to communicate. To reduce the memory overhead, a secrecy transfer model was proposed. Through the proposed secure transfer model, two random nodes can transfer secrets securely along the secure edge of the graph. However, the adversary should not be present until the secrecy transfer model is completed otherwise the undetected malicious nodes can degrade the security of the security of the secrecy transfer. In this work, it is assumed that the mobile sinks may get compromised like the other nodes in the network and hence privileges of the mobile sink are restricted. The following three techniques were suggested to restrict the privileges of mobile sinks:

1. Sensor Based Restriction (SBR): Mobile sink should be allowed to access only pre determined sensors along the path.

2. Data Based Restriction (DBR): In DBR, sensed data is classified into different types, and the mobile sink is allowed to collect only few types of data.

3. Sensor and data based Restriction (SDBR): In SDBR, SBR and DBR are combined. Mobile sink is allowed to access only few types of data in certain sensors.

Along with mAKPS, two security techniques were proposed to protect the data stored in the sensors: Dual Direction Hash Chain scheme (DDHC) and Key-cycle Lock scheme. DDHC scheme contains two one way hash functions, forward hash chain, and backward hash chain. These functions are of equal length and are pre-loaded in every sensor. In this scheme, lifetime of a sensor is divided into phases a mobile sink collects the data from the nodes during every phase. A phase counter is setup and it will be incremented at the end of every phase. The mobile sink that is authorized to access data from these sensors will be pre-loaded with the secret material and the public keying material required for mutual authentication using the forward and backward hash chains. The sink and the sensors verify the messages of each other and then the data is transferred from the nodes to the sink. Once the data is transferred to the mobile sink, the phase counter is incremented and the hash chains are updated. In the Key-cycle Lock scheme, public key encryption is used. The trusted authority has a long term public key know to all the sensors and the private key is secret and is kept to itself. All the sensors encrypt the collected data using the public key of the trusted authority and forward it to mobile sink when it comes to the vicinity. Since the adversary or the mobile sink has no knowledge of the primary key, the data cannot be decrypted and is secure.

2.3.1.2 Replication Techniques for Data Survivability

Replication is the process where sensors create multiple copies of sensed data before moving it to the other locations. Data survival chances increases with replication but the major drawback is the increase of storage and communication overhead. The main

goal of the replication is that sink will find at least one surviving copy among all the copies created by the node. There main factor to be considered for replication is the degree of replication i.e., number of copies to be created to ensure data survivability. Memory and communication overhead depends on the degree of replication.

Pietro *et al.* in [8] proposed data survivability scheme which uses replication. The choice of nodes for storing replicas is based on an epidemic model (SIS) [12]. SIS stands for Susceptible (S), Infected (I), Susceptible(S); in general, an individual that is susceptible to disease becomes infected with a certain probability and an infected individual becomes susceptible if it is cured (healed) of the infection. In this work, nodes are considered as individuals and the disease is considered as data. A node gets a replicated copy from other node then the node is said to be infected with certain probability and if the adversary compromises the node (healed) then it gets back to the susceptible state with certain probability. The basic idea of the model is when a node receives the data, using the replication approach it forwards the data to randomly selected nodes and the sink collects the data as soon as it comes into contact with one of the sensors possessing data. By selecting random nodes flooding can be avoided. In this model, if the replication rate is greater than the healing rate than data survivability is assured but the focus here is to choose minimum replication rate while preserving data survivability thereby reducing communication and memory overhead. Replication rate is chosen such that resource consumption and collecting time is minimized and at the same time data survivability is assured. The probabilistic theory to assure data survivability and the process of selecting replication rate is well discussed in this work.

2.3.2 Data Authentication in UWSNs

In UWSNs, a forward secure signature schemes are effective and minimizes the number of key compromises. Forward secure signature was first proposed in [44]. In this scheme, time stamp is associated with the signed data item. Secret key of the signer is changed after each time period and the same key cannot be used for previous time periods. For signing, message authentication codes are used in literature. Message Authentication Code is a tag or a signature calculated for a message. MAC's are computed and verified with the same key so that only intended recipients can do the verification. A MAC contains two algorithms (*sign and verify*) and a shared key between the signer and verifier. The sender computes the signature of the message using the *sign* algorithm and sends the message along with the signature to the receiver. The receiver accepts the message and signature pair and validates the message using the *verify* algorithm.

Pietro *et al.* [28] proposed two schemes namely Cooperative MAC (Co-MAC) and Extensive Cooperation (ExCo). In Co-MAC, each sensor, after collecting the data, calculates the message authentication code for the data, stores it and forwards the same to t randomly selected nodes in the network for authentication. In each round, a new key is calculated using the old key and the identities of the nodes which selected it as co-authenticators. At the end of every round, the sink collects all the message authentication codes to compare them with the original data. In ExCo, message authentication codes are calculated and sent to t randomly chosen nodes, but all the received message authentication codes are bundled into a single message authentication code. These schemes work well against both reactive (compromises nodes after identifying the target) and proactive adversaries (compromises nodes before identifying its target), but they suffer from high costs overhead relative to the

level of security achieved.

Butterfly pollination scheme proposed by Dimitriou and Sabouri [29] gives better data authentication with less communication overhead compared to [28]. Here, each node selects another node at random and sends its message authentication tag. Whenever this message authentication code passes through a node, the node downloads the message authentication code and attaches its own message authentication code and forwards it towards the destination. This process is repeated until the node reaches the destination. All the message authentication codes are stored at the destination. Since the single destination node is chosen at random, the node can be anywhere in the network which may be a problem in some cases. If the selected node is in other end of the field from the position of the source node then there will be more communication overhead as the data has to pass through all the intermediate nodes and if the destination node is a neighbouring node then probability of survival decreases.

Recently Yavuz *et al.* in [42] proposed three Hash-Based Sequential Aggregate and Forward Secure Signature (HaSAFSS) schemes. These authentication schemes allow the signer to fixed-size, compact and publicly verifiable signature efficiently. These schemes are named as symmetric HaSAFSS (Sym-HaSAFSS), Elliptic Curve Cryptography (ECC) based HaSAFSS scheme (ECC-HaSAFSS) and self-sustaining HaSAFSS (SU-HaSAFSS). In the proposed schemes, the computational overhead is same for both signers and verifiers. HaSAFSS achieves forward security by using Timed Release Encryption (TRE) [43] (The purpose of TRE is to encrypt a message such that it cannot be decrypted by any entity including the receivers until a predefined future time), by using TRE asymmetry is introduced between the senders and receivers. These schemes contain four algorithms; Key generation, forward secure and aggregate signature generation, time trapdoor release and forward secure aggregate

signature verification. These schemes achieve five conflicting goals; forward security, scalability, flexibility, public verifiability and computational efficiency at the same time. In these schemes, the signer stores and transmits only a single compact signature by aggregating the signatures of all the data items, this reduces the communication overhead. Sym-HaSAFSS and ECC-HaSAFSS has a linear bound on number of time periods a signer can use and all the signers should agree on a pre-determined and fixed data delivery schedule before deployment. Su-HaSAFSS does not have these limitations, it enables the sender to use unbound number of time periods and also enables each sender to decide its own schedule with out communicating with other signers. Despite these advantages, Su-HaSAFSS is more computationally costlier than the other two schemes.

2.3.3 Location Privacy

Location information is very important in many applications of wireless sensor networks. Security mechanisms and routing protocols need the location and distance information of the peers. In UWSNs, critical nodes create replicas of the important data to increase the data survivability. However, multiple replicas in the network trigger security vulnerabilities by exposing the location information of the nodes. The adversary can estimate the location of the critical nodes based on the captured replicas. If the adversary has the knowledge of the location of the critical nodes then there will be more damage to the data collection in the network. In some networks, sensors have GPS or other position systems to identify the location of the nodes. In these networks, location will be measured based on the time of flight of radio or ultra sound signals and the receiving strengths of radio signals of the devices [7].

Chen *et al* in [45] provides the tradeoff between location privacy of a critical node

and data survival rate. In UWSN, to ensure the data survivability of the critical nodes, data from the critical nodes is replicated and sent to nodes placed in different locations in the network. Replication obviously increases the data survival rate but it highly increases the chances of giving away the location of the critical nodes to the adversary. In this work the authors proposed three algorithms, the coordinate median, average of overlapping area and expectation maximizing algorithms for estimating the location of the critical nodes based on the compromised sensors storing data replicas. It is shown that all the three location estimation algorithms achieve the same performance and Coordinate Median is more computationally efficient. According to the authors, a better location performance can be achieved by the adversary if there are a large number of replicas and replicas of the source node are stored in nodes which are few hops away from the source node. If the adversary is capable of compromising more number of nodes between the successive visits of the mobile sink then it has better location estimation performance. However if the location privacy is of high priority in the network then the number of replicas should be reduced to prevent the degradation in location privacy. If the data survivability is of high priority then increase in the number of replicas improve the resistance to attack from the adversary. If both location privacy and the data survivability are considered to be important in the network then a optimal number of replicas and the optimal number of hops the replicas should be sent has be chosen such that the location of the critical nodes cannot be exposed.

In [37], Li *et al.* proposed statistical methods to make two classes of localization: RF-based finger printing and triangulation. For triangulation-based methods, their algorithm switches between two location estimators depending on the status of the nodes. If the nodes are in normal status then Least squares position estimator is used and if the nodes are being attacked then Least median squares position estimators will

be used to achieve robustness. For the RF based finger printing location estimation, instead of using traditional euclidean distance metrics, median based distance metrics have been used. In [38], analysis is done on attack model of node centric and infra structure centric positioning systems. In node centric positioning systems, position will be calculated based on the signals received from the stations whose positions are known. In infrastructure centric position systems, position will be computed based on the mutual communication between the nodes. In this work, they proposed a new approach to secure locations based on hidden and mobile sinks. This approach uses broad spectrum positioning techniques such as ultrasonic or RF to enable secure positioning. In this approach, more verification is done by the sink which involves more communication overhead and is not practical in the case of UWSNs.

Hwang *et al* in [39] proposed a secure localization mechanism to detect the phantom nodes. Phantom nodes are the malicious nodes that has the ability to fake their locations that are different from the existing their existing physical locations. The algorithm proposed in this work has two phases. First is the distance measurement and filtering phase, where each node measures the distance to its neighbours. In the second phase, every node in the network projects its neighbours into a virtual plane which determines largest consistent subset of nodes. A local view without phantom nodes is developed after the second phase. In this scheme all the nodes play the role of verifier by generating a local map which prevents central point of attack. Using this algorithm most of the phantom nodes can be filtered even when the number of phantom nodes are greater then the number of honest nodes.

Chen *et al* in [40] proposed several attack detection schemes for wireless localization systems. Unlike traditional security schemes, proposed schemes can detect non cryptographic attacks such as signal signal attenuation and amplification. Using these schemes, mathematical models and analytical solutions for attack detections can be

defined for any system that uses multilateration and signal strength approaches for localization. In multilateration, an analytical solution has been developed using Linear Least Squares (LLS) regression has been used for easy conducting. In signal strength based approaches, the minimum distance between the observation and database containing signal strength vectors is used for attack detection. The main advantage of this approach is that detection step can be performed before localization.

Liu *et al* in [41] proposed two methods to tolerate attacks against beacon-based location discovery in the network. The first method, "attack resistant Minimum Mean Square Estimation" filters the malicious signals based on inconsistency. The malicious location references introduced by attacks mislead the normal sensor nodes and these references are usually inconsistent. The proposed method identifies these signals based on the inconsistency and removes them from the network. In the second method named "voting based location estimation" deployment field is quantified into cells. Each location reference vote on the cells in which node resides. The results of the voting is iteratively refined to determine the location.

2.3.4 Secure Data Aggregation

In typical data aggregation procedure, each node collects the data and processes the data with the peers to combine the information. The combined information can be collected by the sink. Though this process reduces the communication and memory overhead, it brings more security issues. In general, two types of attacks are possible in the data aggregation process. The first one is that the information sent to the aggregator from the nodes might have been modified by a adversary and as result aggregators receive corrupted information. The second one is that the adversary compromising the aggregator itself and modifying the aggregated data as a result,

falsified information will be sent to the sink. An other challenge with the unattended network is choosing the aggregator node. The aggregator node cannot be same in all the rounds Data aggregation can be made secure in two ways: plaintext based scheme and cipher text based scheme[7].

2.3.4.1 Plaintext Based Schemes

In the plaintext based schemes, the intermediate nodes which are transferring the message to the aggregators know the content of the message. Deng, *et al* in [30] introduced routing algorithms which contain downstream and upstream authentication between sensors and aggregators. In downstream authentication, sensors authenticate the commands sent from the aggregator nodes using hash chains and μTESLA. In the upstream, data is authenticated by the aggregator before it is aggregated. These algorithms require pairwise keys established between aggregator and sensors. Du, *et al* in [31] proposed a witness based approach which ensures that the data sent to the sink by the aggregator is not falsified. In this approach, some neighbouring nodes of the aggregator node are selected as witness nodes which monitor the results of the aggregator. When the sink collects the information from the aggregator, it uses a voting strategy among the neighbouring nodes to decide whether to accept the data or not. In [32] a multi path message authentication method was introduced called Canvas where each node creates two message authentication codes(MAC's) for the next two nodes. So, a message is authenticated twice before it is forwarded to the aggregator. Ye, *et al* in [33] proposed a statistical en-route filtering (SEF) scheme to detect the compromised data. In this scheme, MAC is attached to the data packet and all the intermediate nodes on the path to the aggregator verifies the correctness of the MAC's probabilistically and drops the invalid MAC's.

2.3.4.2 Cipher Based Schemes

Unlike plaintext schemes, the intermediate nodes do not know the content of message. Concealed Data Aggregation (CDA) proposed in [34] is one way to prevent the data disclosure in the intermediate nodes. This work is based on the privacy homomorphism [35] which allows the direct computation on the encrypted data. Peter *et al.* in [36] evaluated three privacy homomorphism algorithms suitable for wireless sensor networks: Domingo-Ferrer (DFPH), CMT (a key based privacy homomorphism) and Elliptic Curve ElGamal (ECEG). ECEG is a asymmetric cryptographic approach which is based on elliptic curve cryptographic algorithm. According to the authors of [36] none of these algorithms achieve desirable security goals and CMT is considered more effective among the three algorithms. Two schemes were proposed in this work, the first one two algorithms were combined so that the weakness one algorithm will be balanced by the strengths of the other algorithm. In the second scheme, two biggest issues of the CMT algorithm were evaluated.

Chapter 3

Proposed Data Survivability Schemes

In Unattended Wireless Sensor Networks (UWSN's) every node has to secure the data until the next visit of the sink. This lengthy intervals of absence of sink greatly increases the susceptibility of attacks focusing on the data collected by the sensors. Security needs should be taken into account to ensure data protection (also called *data survivability*) in these sensors at the time of design. Distributed security schemes are preferable over centralized schemes, because centralized schemes are prone to single point failure. Existing distributed security mechanisms for WSNs [5, 7] are not suitable for the UWSNs due to infrequent visits of the sink. Data security and data authentication is a major concern in UWSNs. Most cryptographic techniques provide data authenticity and integrity but do not ensure data survivability. This implies that if an adversary compromises a sensor and destroys the data contained therein, the data is lost forever. The other drawback of cryptographic schemes is that they are computationally expensive for resource-constrained sensor nodes. For these reasons, non-cryptographic techniques can be considered over cryptographic ones.

Table 3.1: Notations and their meanings

Symbols	Meaning
N	Size of the network
h	Number of hops
ADV	Mobile adversary
K	Public key of sink
s_i	i-th sensor node
$nbd(s_i)$	Set of neighbors of s_i
d_j	Data sensed by sensor j
$E_K(d_j)$	Data d_j encrypted using key K
t	Number of nodes with duplicate data
m_i	Message with encrypted data
T	Round time
v	Number of rounds between successive visits by sink
k	Number of compromised nodes
r	Transmission range of a node
P_{attack}	Probability of attack
$P_{survival}$	Probability of data survivability

3.1 Background

Notations used in this chapter are presented in Table 3.1.

3.1.1 Network model

UWSN consists of N mobile sensors s_1,s_2,... s_N uniformly deployed in the network, programmed to sense and collect data periodically. All the sensors are mobile following *RandomJump* (RJ) [11] mobility model. In RJ, sensors move with speeds such that any point of the deployment area can be reached in one round. Time is divided into rounds, in each round sensors collect data and encrypt with the public key of the sink. We assume that the sink is a trusted authority. The sink visits once in every v rounds to collect the data. As soon as the data has been transferred to the sink,

entire memory of the sensor node is erased. The sink reinitializes all the sensors with its public key K, seed of pseudo random number generator (PRNG) and hop count h, which will be discussed later. The PRNG is used to select a neighbor randomly. It is assumed that the communication between sensors or between sensor and sink is reliable. Every sensor is aware of its location and the location of its immediate neighbors.

3.1.2 Adversarial Model

Adversary(ADV) is mobile and moves randomly over the network. As assumed by authors in [28] and [29], we consider two adversary models, namely *Reactive* and *Proactive*. If the ADV is *Reactive*, it starts compromising only after the target is identified, and if the ADV is *Proactive*, it starts compromising before the knowledge of the target node is obtained. Goals of the ADV is to change or destroy the data of the sensor nodes. It is to be noted that the adversary cannot decrypt data because it is encrypted with sink's public key. An adversary can compromise a constant number of sensors every round. As long as a sensor is compromised, the adversary gets access to the stored information and can eavesdrop traffic through the sensor. ADV has the knowledge of composition and topology of the network and is stealthy; sensors cannot detect the presence of the adversary even after it is compromised.

3.2 Proposed Protocols

Depending on the nature of the adversary, two schemes are proposed in this work. In the following, we construct and evaluate two schemes that enhance the security of UWSN by replicating the data.

3.2.1 Data Survivability against Proactive Adversary (DS-PADV)

Since all the nodes are mobile, at the start of every round, each node gets the information about the location of the neighbouring nodes. At every round, data d_i sensed by a node s_i is encrypted with sink's public key. The encrypted data $m_i = E_K(d_i)$ is stored in its memory and is forwarded to one of the neighbouring nodes. The neighbouring node is selected at random using a pseudo random number generator. A data bit h is attached along with the encrypted data which determines the number of hops the data has to travel. h can be determined by the sink during its visit to the sensor node; it can be based on the amount of energy left over in the node or the level of security needed and can also be pre-determined before deployment based on the size of the network.

The node which receives the data decrements h by one, downloads the received data into its memory. Let node s_i send data m_i to its neighbour s_j. s_i concatenates this data with its sensed data m_j. Node s_j then calculates the angle θ_{s_i,s_j,s_l} between s_i, itself and each of its neighbours $s_l \in nbd(s_j)$. It then forwards it to the neighbour s_q such that

$$\theta_{s_i,s_j,s_q} = \pi - min_{s_l \in nbd(n_j)} |\pi - \theta_{s_i,s_j,s_l}| \tag{3.1}$$

This implies that the concatenated data $m_i||m_j$ is forwarded to the neighbouring node s_q which is in the opposite direction of the node from which the data has been received. If there is no node in the opposite direction, a node close to the opposite angle is chosen. Data is forwarded until h reaches zero. So, source nodes need not know the location of the node to which data is being sent. This scheme can be applied when the adversary is proactive, that is when the adversary starts compromising nodes randomly before the target node is found. The process is explained through Fig. 3.1,

where $h = 3$. The shaded nodes contain the data (m_1) of the node s_1. As explained, s_1 selects s_3 as its random neighbour and sends its data, s_3 decrements hop count to 2, concatenates its data m_3 and sends $m_1||m_3$ to s_4. Same procedure is repeated in s_4 and when the hop count reaches zero in node s_5, data of all the nodes (s_1, s_3, s_4) will be saved in s_5. Similarly, s_2 is transmitting the data with s_6 as random neighbouring node. A copy of data of s_1 will be saved in s_7 as shown in Fig. 3.1. Node s_7 will store $m_2||m_6||m_1$. Every sensor runs the procedure in Algorithm 1.

Algorithm 1 DS-PADV

```
Sense d_j
Compute m_j =E_K(d_j)
Store m_j locally
Set H_j = h
Select a neighbour randomly
Send m_j and H_j to selected neighbour
  while time ≤ T do
    Receive Hop and M          ▷ Hop count and message received from neighbour
    Hop = Hop − 1
    if Hop > 0 then
      Save M locally
      Forward M = M||m_j and Hop to the neighbor
      in the direction opposite to the source node
    else if Hop = 0 then
      Save M locally
    end if
  end while
```

3.2.2 Data Survivability against Reactive Adversary (DS-RADV)

If the adversary is reactive (the adversary starts compromising when the target node is found) the following scheme can be used. In this scheme, at every round sensed data is encrypted, stored in its memory and forwarded to randomly selected t neighbors.

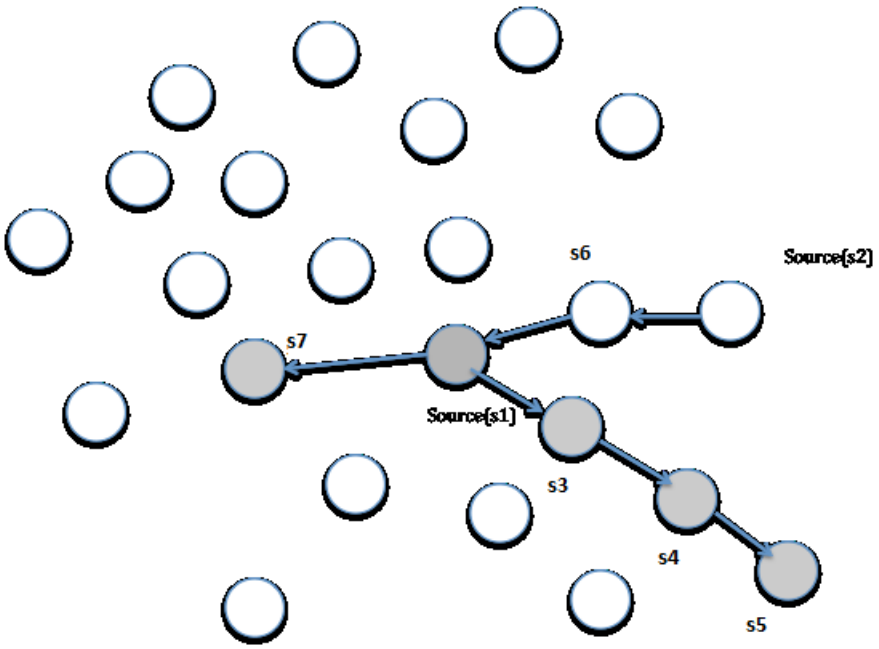

Figure 3.1: Scheme DS-PADV highlighting the message of source node 1

Unlike DS-PADV, data is not stored by the nodes forwarding it. Intermediate nodes just decrement the hop count and forward data until the hop count reaches zero. When the hop count reaches zero, data is saved in the memory of that node. Since the nodes in the network are uniformly distributed, chosen t neighbours cannot send the data to the same node. For each node there will be t copies of data in different directions. For the adversary to win, it has to compromise all the t nodes which are h hops away from the source node. The mobile adversary has to travel in t different directions to compromise all these nodes and it should be done in a single round which is not feasible. The scheme is explained in Algorithm 2.

Working of this is illustrated in Fig. 3.2 with a hop count $h = 3$ and $t = 2$. Shaded nodes in Fig. 3.2 contain the encrypted data m_1 of s_1. s_1 selects s_2 and s_5 as two

Algorithm 2 DS-RADV

Sense d_j
Compute $m_j = E_K(d_j)$
Store m_j locally
Set $H_j = h$
Select t neighbours randomly
Send m_j and H_j to all t neighbours
while $time \leq T$ **do**
  Receive Hop and M ▷ Hop count and message received from neighbour
  $Hop = Hop - 1$
  if $Hop > 0$ **then**
    Forward M and Hop to the neighbour in the direction
    opposite to the source node
  else if $Hop = 0$ **then**
    Save M locally
  end if
end while

random neighbours and sends m_1. s_2 and s_5 decrements h and forwards m_1 to the nodes in the opposite direction, as in the previous scheme. In this case they are s_3 and s_6 respectively. Same procedure is repeated in s_3 and s_6. Finally, when h reaches zero (in nodes s_4 and s_7), m_1 is saved locally.

In both the schemes, at the end of every round when the node moves to a new location, it registers itself with the neighbouring nodes in the new location and gets the location information of all the neighbouring nodes.

Hop count for sending data is very important as a network may contain large number of nodes. If the destination node is chosen randomly, then data may have to travel from one end of the field to the other end to locate the destination node. There might also be a possibility that the chosen node is a neighbouring node, so the probability of data survival may reduce. In case of random destinations, locations of the nodes in the network has to be saved and routing tables should be updated every time which is a overhead. In the proposed schemes, knowledge of locations of

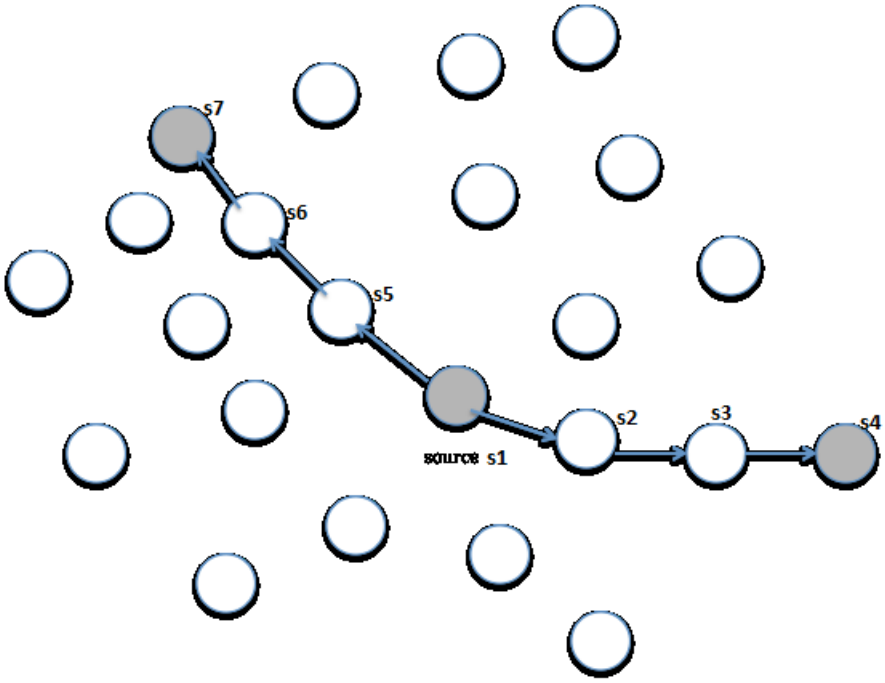

Figure 3.2: Scheme DS-RADV highlighting the message of source node 1

neighbouring nodes alone is sufficient.

3.3 Performance evaluation

In this section, we analyze our proposed schemes in terms of communication costs, memory overhead and data survivability and compare with the schemes proposed in [28], [29] and [1]. Our schemes are implemented using Network Simulator (NS2) [63]. NS2 is a open source simulation tool operating on unix based operating systems. NS2 is implemented using C++ programming language with Object Tool Common Language (OTCL) as front-end interpreter. The version of NS2 used in this thesis is 2.35 and it is installed on Ubuntu 10.04 operating system. The simulation results are generated in two separate files output trace file and a NAM trace file. The trace files contain the information about the events occurred in the network.

Simulations were conducted on a network comprised of 100 sensors, each placed 10m apart and with a coverage radius of 40m. Node are arranged uniformly in a rectangular grid (1000*1000). Nodes collect the data once in every 10 seconds which is considered as a round. The nodes are moved to a different location using the "setdest" command in ns2. To simulate the proactive adversary, random nodes are selected in the network and saved as compromised nodes. For the reactive adversary, a node is selected at random and the nodes surrounding it are saved as compromised nodes. The power of the adversary, i.e., the number of nodes it can compromise, is varied for different simulations. The data traffic type between the nodes is Constant Bit Rate (CBR) and the routing protocol used is AODV. The data is considered secure when selected destination nodes for sending the data are not same as the compromised nodes. The compromised nodes regain its security once in 100 seconds (10 rounds), that is when the sink visits. Information regarding the number of messages sent by each node is calculated using the trace files generated. Survival probability is calculated by comparing the nodes containing the data with the compromised nodes. If all the nodes containing the data are in the list of compromised nodes then the survival probability is zero. Results are also checked with varying number of nodes in the network. Simulations are conducted over 100 times and average results are considered.

3.3.1 Data Survival Probability

We define survival probability (and denote it by $P_{survival}$) as the probability that the data can be retrieved even if the target is compromised. Probability that a node is compromised is denoted by P_{attack} and given by

$$P_{attack} = \frac{k}{N},$$

where k is the number of compromised nodes and N is the size of the network. In order to achieve its goal, a proactive adversary has to compromise all the nodes h hops away from the target node. Since the nodes are mobile, all these nodes should be compromised in a single round by the adversary. Thus, the probability of survival is given by

$$\begin{aligned} P_{survival} &= 1 - (P_{attack})^h \\ &= 1 - (\frac{k}{N})^h \end{aligned} \tag{3.2}$$

From the above equation, it is evident that the probability of survival increases with the increase in hop count. This is illustrated in the Fig. 3.3 with 600 compromised nodes.

In case of DS-RADV, the adversary, in order to achieve its goal, has to compromise the target node and t other nodes which contain the data of the target node. The adversary can learn the value of h after compromising the target node. To compromise t nodes, the adversary has to choose the exact t neighbours of the target node to which the data has been forwarded. All the nodes in the transmission range (r) of the target node are the neighbours of the target node. After choosing the exact t neighbours, the adversary has to traverse through h hops in a direction opposite to the target node to reach the nodes containing the data of the target node. So, the probability of compromising t nodes containing the data is same as the probability of choosing t neighbours around the target node. Since the nodes are arranged uniformly in the field, number of nodes within the transmission range r would be $2r(r+1)$. Thus, the probability that t nodes are compromised amongst the $2r(r+1)$ neighbours is

$$P_{attack} = \frac{t}{2r(r+1)}$$

$$\begin{aligned} P_{survival} &= 1 - P_{attack} \\ &= 1 - \frac{t}{2r(r+1)} \end{aligned} \tag{3.3}$$

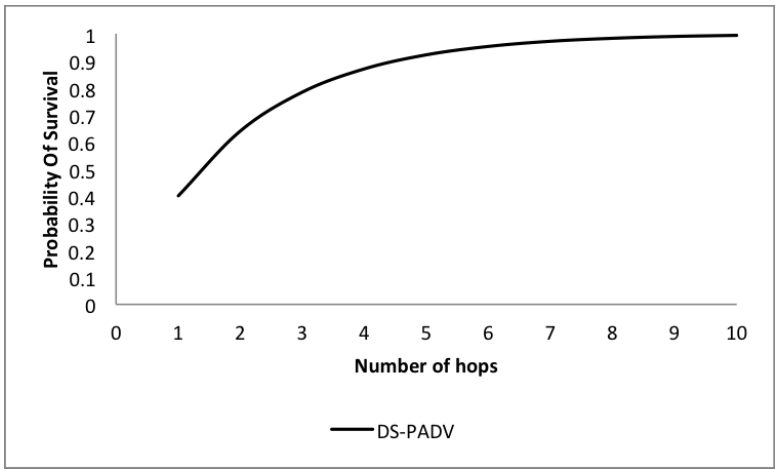

Figure 3.3: Survival Probability with 600 compromised nodes and N=1000

Replication model for data survivability proposed in [1] against is compared with DS-PADV and illustrated in Fig. 3.4.

3.3.2 Communication Costs

In DS-PADV, the data has to travel for h hops, and all the intermediate nodes will have a copy of the data. If N nodes are transmitting messages, each message has to pass through h other nodes. So, the number of communications in the network would be Nh. Considering CoMac and ExCo [28], where every node has to send messages to t randomly chosen nodes in the network. As a result, the communication cost is $O(tN\sqrt{N})$. In [29], each message has to be transmitted to a randomly chosen node, and hence the number of communications are $O(N\sqrt{N})$. For N nodes transmitting the message, the communication cost is $O(Nh)$. Communication cost in DS-RADV

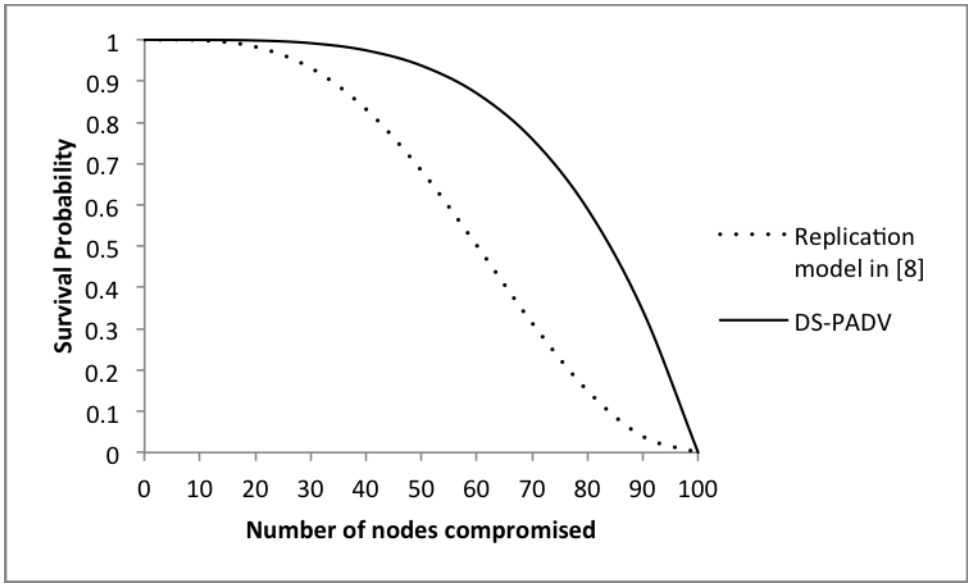

Figure 3.4: Survival Probability of Replication model in [1] and DS-PADV

is t times the communication cost in DS-PADV. Thus, the communication cost in DS-RADV is $O(tNh)$.

Comparisons are done using simulations in a field of 1000 nodes and shown in Fig. 3.5. In the proposed schemes, there is a constant number of point to point communications as the number of hops are pre-determined unlike the other protocols, where communication costs may vary depending on the position of the randomly selected node.

3.3.3 Memory Overhead

Every node in the network should be capable of storing the data until the next visit of the sink. In DS-PADV, each node should be capable of storing $O(vh)$ messages between successive sink visits. This clearly shows the tradeoff between memory and survivability. In DS-RADV, since only t copies of data has to be saved, each node should be capable of storing $O(vt)$ messages between successive visits of sink.

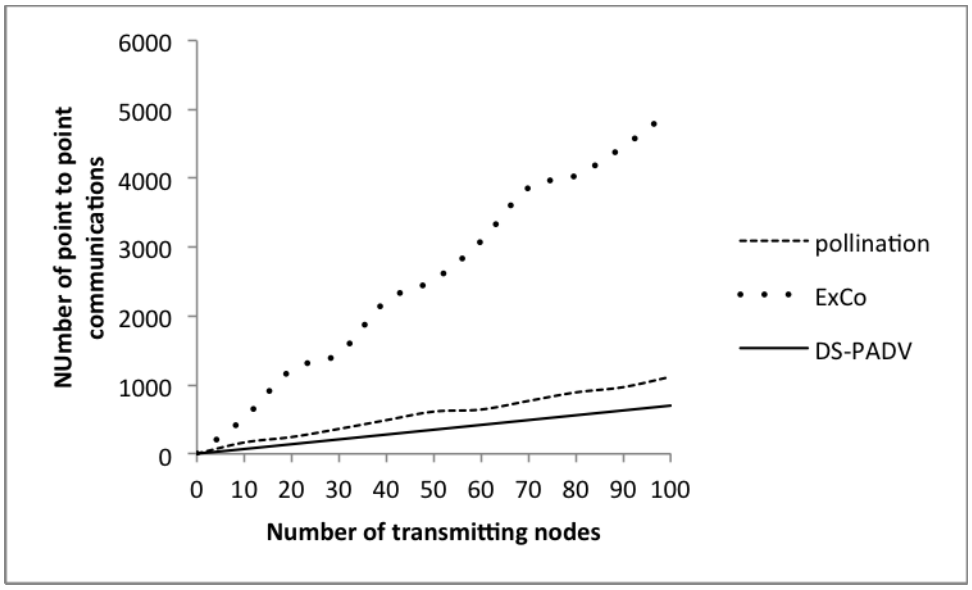

Figure 3.5: Communication costs with $N = 1000$, $h = 7$ for DS-PADV, $t = 5$ for ExCo

Table 3.2: Comparison of storage and Communication costs

Schemes	Number of replicas	Communication per node
DS-PADV	$O(vh)$	$O(h)$
DS-RADV	$O(vt)$	$O(th)$

3.3.4 DS-PADV vs DS-RADV

DS-RADV contains t times more point to point communications than that of DS-PADV. However, the memory overhead is less since the data has to be saved in only t nodes. As t increases, energy consumption in the network increases t times, so the value of t should be chosen wisely. To save the energy of the network, a small h value can be chosen for large t value. DS-PADV and DS-RADV are compared in terms of communication costs and are shown in Fig. 3.6. We have summarized the analysis results for communication costs and storage for both DS-PADV and DS-RADV in TABLE 3.2.

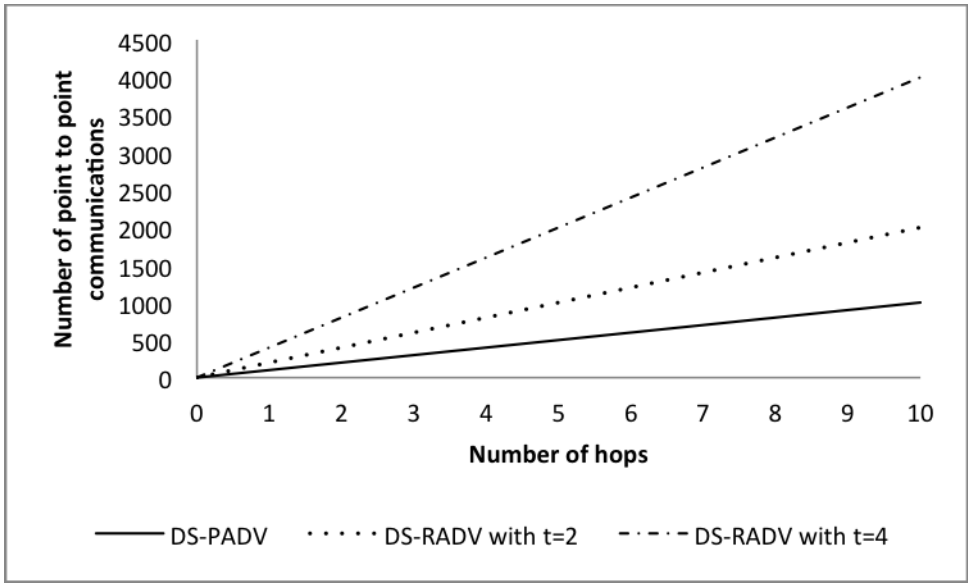

Figure 3.6: Communication costs in DS-PADV and DS-RADV when 100 nodes are transmitting

3.4 Summary

In this chapter, we considered the data survivability problem in mobile unattended wireless sensor networks against two types of mobile adversaries. We proposed two non-cryptographic distributed schemes based on replication of data that maximizes the data survival with little overhead. The proposed schemes were analyzed and evaluated based on survival probability, communication costs and memory overhead and compared against the existing schemes. In existing schemes, sensors remain static between visits from the sink, whereas in our scheme sensors can move between successive visits from the sink. Our schemes perform better than the existing schemes in terms of communication overheads.

Chapter 4

Proposed Data Authentication Scheme

Sensors transmit mission critical information over the network and hence there is a need for security services like authentication, encryption, key management, etc., The unattended nature of the network and the lack of tamper resistant hardware increases the susceptibility of attacks over the data collected by the sensors [29]. So, a mechanism is needed to ensure that the data received at the sink is authentic. Since the UWSN scenario is different from the traditional WSN's, defence solutions from WSN security literature are not suitable for coping with a mobile adversary in UWSN's. The main goal of the adversary [29] is to inject fraudulent data into the information collected by the nodes and remain undetected. The mobile adversary is capable of compromising k out of n nodes in each round and it can also switch to different set of k nodes per round. Authentication schemes for UWSN against a mobile adversary presented in [28] and [29] guarantee good security but suffers from high communication cost relative to the level of security achieved. In this work, we introduce a collaborative authentication mechanism against a mobile adversary

Table 4.1: Notations and their meanings

Symbols	Meaning
N	Size of the network
h	Number of hops
ADV	Mobile adversary
K	Public key of sink
s_i	i-th sensor node
$nbd(s_i)$	Set of neighbors of s_i
d_j^r	Data sensed by sensor j in round r
t	Number of nodes with duplicate data
T	Round time
v	Number of rounds between successive visits by sink
k	Number of compromised nodes
r	Transmission range of a node
P_{attack}	Probability of attack
$P_{survival}$	Probability of data survivability
l	Size of the cluster

focusing on maximizing security while reducing the communication overhead.

4.1 Background

4.1.1 Notations

Notations are presented in TABLE 4.1.

4.1.2 Network model

We consider an UWSN with N nodes s_1, s_2, ..., s_N uniformly deployed in a network programmed to sense and collect data periodically. Every sensor in the network is aware of its location and the location of its immediate neighbors with respect to its location. Network operates time units called rounds and sensors are responsible

to sense the data once in every round. Network consists of a mobile sink which periodically collects the sensed data from the nodes. Each sensor s_i collects data d_i^r every round r and the collected data is stored locally until an authorized mobile sink offloads the data. Sink is assumed as a trusted authority and cannot be compromised. Sink vists all the nodes once in v rounds to collect the data. As the data has been transferred to the sink, it is erased from the memory. Additionally, mobile sink re-initializes the secret keys and also the unique secret seed for the Pseudo-Random Number Generator (PRNG) during every visit. We assume that the nodes in the network form a connected cluster so that there exists a communication path between them.

We assume that every node has a Pseudo-Random Number Generator (PRNG) and can perform cryptographic hashing. All the nodes in the network are synchronized, they are able to operate in the same epoch. A dynamic node election mechanism is also necessary to assign the role of nodes. The node election mechanism is assumed to be non-manipulable. The election process takes place at the beginning of every epoch. Non-manipulable node election protocols are available in literarture e.g., [15], [16].

4.1.3 Adversary Model

The adversary has the capacity to control a set of nodes k out of N nodes in any round. The goal of the adversary is to inject false data (data that deviates from its actual measurement) or to delete the stored data in the nodes. It is to be noted that the adversary cannot decrypt the data since it is encrypted with the sink's public key. Adversary is mobile and moves randomly through out the network. It is assumed that the adversary cannot interfere with the communication between the nodes, in

other words communication between nodes is assumed to be secure. Besides the attacks, ADV can also physically corrupt the nodes. Sensors may fail due to the power depletion or due to any other natural causes resulting in loss of functionality.

4.2 The Proposed Scheme

In this section, we propose an authentication scheme which is based on a three step procedure: (i) Normal nodes calculate message authentication code, (ii) Cluster heads calculate hash value for the received MAC's, (iii) Replicating the calculated hash value and sending it to other nodes. In every round, each node picks one of the two roles: (i) Normal node and (ii) Cluster Head. Normal nodes senses the data and calculate the message authentication code (MAC). The role of the cluster head is to collect the MAC's from a group of nodes and hash them. Cluster head also creates replicas of the calculated hash value and forward them to other nodes in the network to avoid single point of failure.

4.2.1 Normal nodes

During the election process, if a sensor s_j, picks its role as a normal node then at the start of round r, s_j collects its sensed data d_j^r. Normal nodes also act as storage nodes which stores the hash values received from the other cluster heads. Message authentication code (MAC) [20, 23] is calculated for the sensed data using the pre-initialized key in the sensor. At any round, every normal node runs two seperate process $SEND$ and $RECEIVE$. $SEND$ process computes the MAC of the collected data and saves the data in the local storage. Send process is illustrated in Fig. 4.1.

$RECEIVE$ process will be executed only if the hash value is sent to this node for storage or for forwarding. Node receives hash value (H_r) along with another

parameter *hop* which determines the hop count or in other words the number of nodes hash value has to be transmitted. H_r is calculated at the cluster head (CH) using a one way hash function which is explained in the next subsection. If the *hop* has a value of 0, then H_r will not be transmitted any more and will be saved locally. If the value of *hop* is greater then 0 then, it is decremented by 1 and forwarded to the node in the direction opposite to the node from which H_r has been received. If there is no node in the exact opposite direction then a node (among the neighbors) located close to the opposite direction is selected and used to forward the information. Node s_j calculates the angle θ_{s_i,s_j,s_l} between s_i, itself and each of its neighbors $s_l \in nbd(s_j)$. It then forwards it to the neighbor s_q such that

$$\theta_{s_i,s_j,s_q} = \pi - min_{s_l \in nbd(n_j)} |\pi - \theta_{s_i,s_j,s_l}| \tag{4.1}$$

This process is illustrated in the Fig. 4.2. As shown in the figure, cluster head sends the H_r and Hop (initialized to 2) to s_1, s_1 saves the value locally and decrements the *hop* by 1. s_1 inturn locates the neighbour in the direction opposite to the cluster head and forwards H_r and the updated *hop* value. As a result s_2 is selected as the next node to which data has to be sent. Since value of Hop reaches 0 in s_2, there is no more forwarding done. Intial value for the *hop* can be decided by the sink and can be changed during its visit. It determines the number of replications required.

During the next visit, sink collects the data along with the hash value (if present in the node). Every node computes a new key from the old key and a random value generated from a PRNG using a one way hash function $h(.)$. The secret seed for the PRNG will be updated by the sink during its visit.

Algorithm 3 SEND

Sense d_j^r
Compute $z_j^r = MAC(K_j^r, d_j^r)$
Store d_j^r locally
Send z_j^r to the elected cluster head

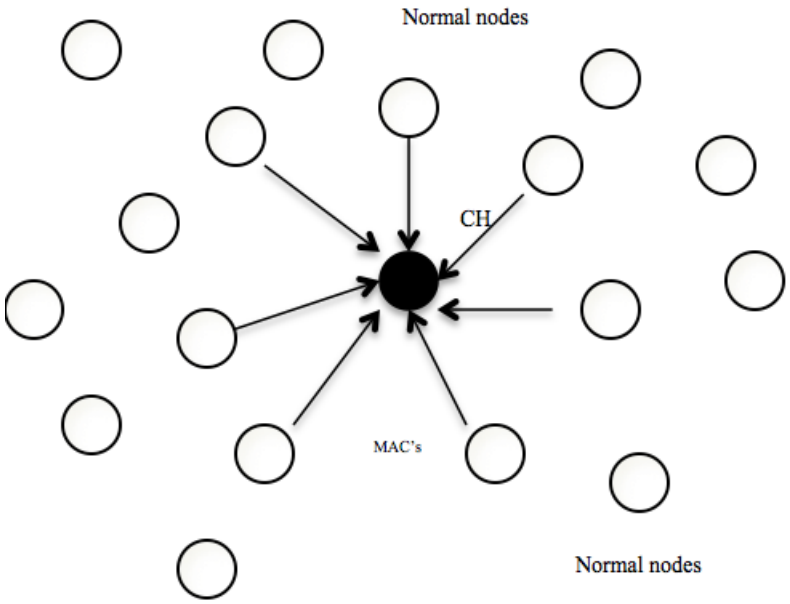

Figure 4.1: Normal nodes sending their MAC's to the Cluster head (CH)

Algorithm 4 RECEIVE

while *roundnotover* **do**
  Receive H_r and *hop*
  $hop = hop - 1$
  if $Hop = 0$ **then**
    Store H_r locally
  else if $Hop > 0$ **then**
    Save H_r locally
    Forward *hop* and H_r to the neighbor
    in the direction opposite to the source node
  end if
end while
$K_j^{(r+1)} = H(K_j^r, rand)$

4.2.2 Cluster Heads

Cluster heads receive message authentication codes from the normal nodes, received MAC's are combined with their own MAC and a hash value is computed for the combined MAC. Hash value is created using a one way hash function. Computed hash value is sent to randomly selected t neighbors, these t neighbors follow the RECEIVE algorithm explained above and forward the value until *hop* reaches 0. Cluster heads initialize the *hop* with a value given by sink during its visit. This value determines the number of replications as explained earlier. Since the data is transmitted in a particular direction, cluster head or other nodes do not keep track of the path of the transmitted data, thus even if the node gets compromised the adversary cant track the next destination.

Algorithm 5 Cluster Heads

Set $R^r = \Phi$
Intialize *hop* to the value given by sink
  while *roundnotover* **do**
    Receive z_j^r
    $R^r = R^r||z_j^r||j$
  end while
$H_r = h(R^r)$
Save H_r locally
Set S_j^r = Select t neighbors randomly
  for p=1 $\ldots t$ **do**
    Send H_r and *hop* to $s_{S_j[p]}$
  end for

The sink visits all the nodes in the network once in every v rounds. It gets the collected data and the hash values from all the nodes. Since the sink knows the initial values of the keys and the roles of every node, it is easy for the sink to simulate the behaviour of the nodes. After calculating the keys and MAC's, sink can compute the hash value and compare with the values received from the nodes to verify the

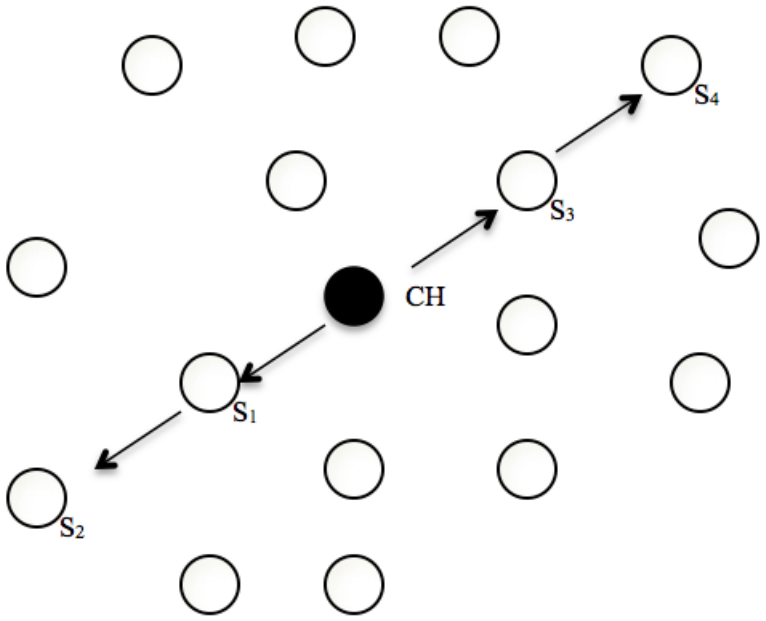

Figure 4.2: Cluster head(CH) replicating the H_r

authenticity of the collected data. Hop count has a major role as network may contain large number of nodes. If the co-autheticator for a node is chosen randomly, then data may have to travel from one end of the field to the other end to locate the destination node. The random choice may also result in choosing a co-authenticator located close to the node, so the probability of survival may reduce. In case of random destinations, locations of the nodes in the network has to be saved and routing tables should be updated which results in a overhead. Since the proposed scheme is based on the knowledge of locations of neighboring nodes, there is no need for the routing tables.

4.3 Performance Evaluation

In this section, we analyze the proposed scheme in terms of communication costs, memory overhead and data survivability and compare with the work in [28] and [29]. Implementation is done using NS2. The simulation environment is similar to the work in Chapter 3.

4.3.1 Security Analysis

For an adversary, to successfully inject falsified data into a single node:

1. It has to know the key of that particular node and also keys of all the nodes present in the cluster in order to recalculate the MAC's of all nodes.

2. It has to locate and compromise the cluster head , recalculate the hash value with the falsified data and replace it with the previous hash value.

3. Since cluster head does not have the information of the replications in its memory, adversary has to locate the replicated copies. It has to compromise all the nodes which are h hops away from the cluster head and replace the hash value with the falsified value.

We define the successful attack probability (P_{Attack}) as the probability with which the adversary can plant the falsified data into the node and sink validating the injected data as the data collected by the sensor s_i. The ideal case for the ADV to make this happen is to use all its power to compromise all k nodes which are h hops away from the the target node and hope that the H_r does not escape the target node area. Probability for a node to be compromised is

$$P_{attack} = \frac{k}{N},$$

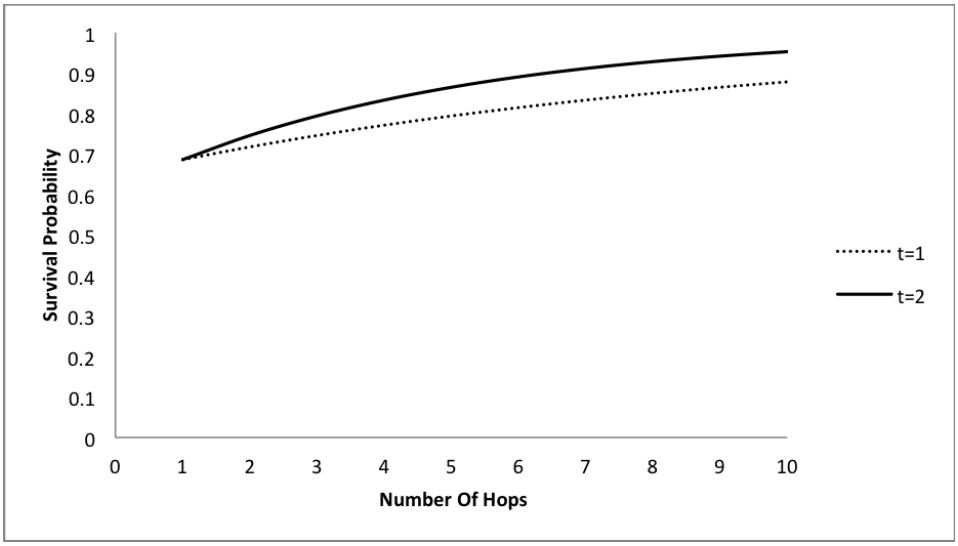

Figure 4.3: Survival Probability with 900 compromised nodes and N=1000

Since the ADV has to compromise all the nodes in the cluster, cluster head and all the nodes located h hops away, the probability of survival is given by

$$\begin{aligned} P_{survival} &= 1 - (P_{attack})^{(ht)+l-t+1} \\ &= 1 - (\frac{k}{N})^{(ht)+l-t+1} \end{aligned} \tag{4.2}$$

From the above equation, it is evident that the probability of survival increases with the increase in hop count and t. This is illustrated in the Fig. 4.3 with 900 compromised nodes in a field of 1000 nodes. Fig. 4.4 shows the survival probability with varying number of compromised nodes.

4.3.2 Communication Costs

During each round, all the nodes send their MAC's to their respective cluster heads, cluster heads calculates the H_r and sends it to t random neighbors. These t random neighbors send this value to h nodes each, resulting in $O(t\sqrt{N})$ point to point com-

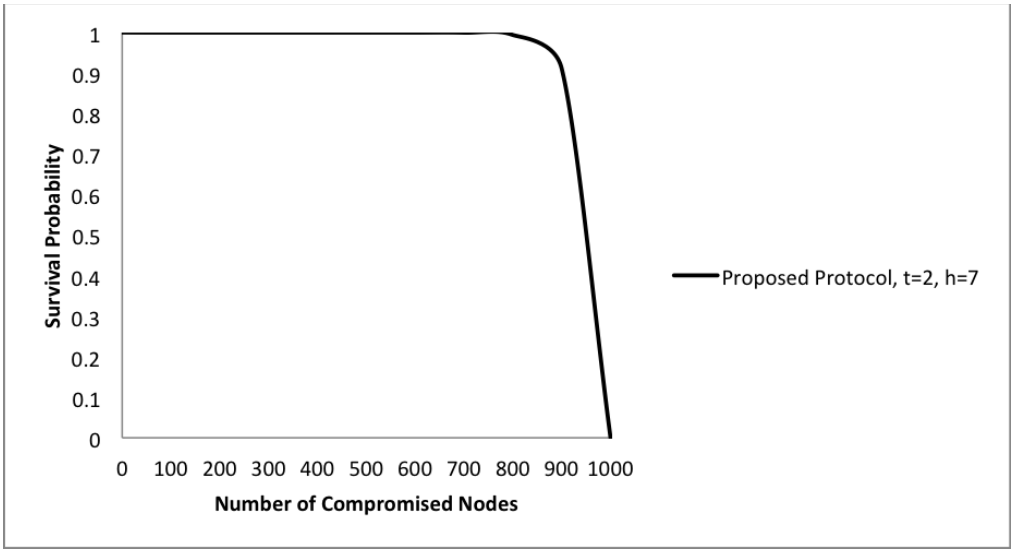

Figure 4.4: Survival Probability with N=1000 and varying k

munications per cluster in a single round. Considering CoMAC and ExCo [28], every node has to send the tag to t randomly chosen co-authenticators. In the proposed scheme, there is a constant number of point-to-point communications as the h and t are pre-determined unlike other schemes where communication costs vary depending on the position of the co-authenticators. As a result, overall communication cost is $O(t\sqrt{N})$ for a single node. So the proposed protocol reduces the communication cost by l (number of nodes in a cluster) times compared the CoMAC and ExCo. In butterfly pollination scheme [29], butterfly initiates from each node and goes through $\sqrt{N}$ nodes on an average, so number of point to point communications in this case is $O(\sqrt{N})$ per node.

A summary of communication costs is shown in TABLE 4.2. Simulation results for the communication costs in a network of 1000 nodes is shown in the Fig. 4.5. For the simulation purposes, it is assumed that cluster has 10 nodes, t=2 and h=7 . For ExCo and CoMAC, a t value of 5 is considered.

Table 4.2: Comparison of Communication costs

	Communication per round
ExCo	$O(t\sqrt{N})$ per node
CoMAC	$O(t\sqrt{N})$ per node
Pollination	$O(\sqrt{N})$ per node
Proposed Scheme	$O(t\sqrt{N})$ per cluster with l nodes

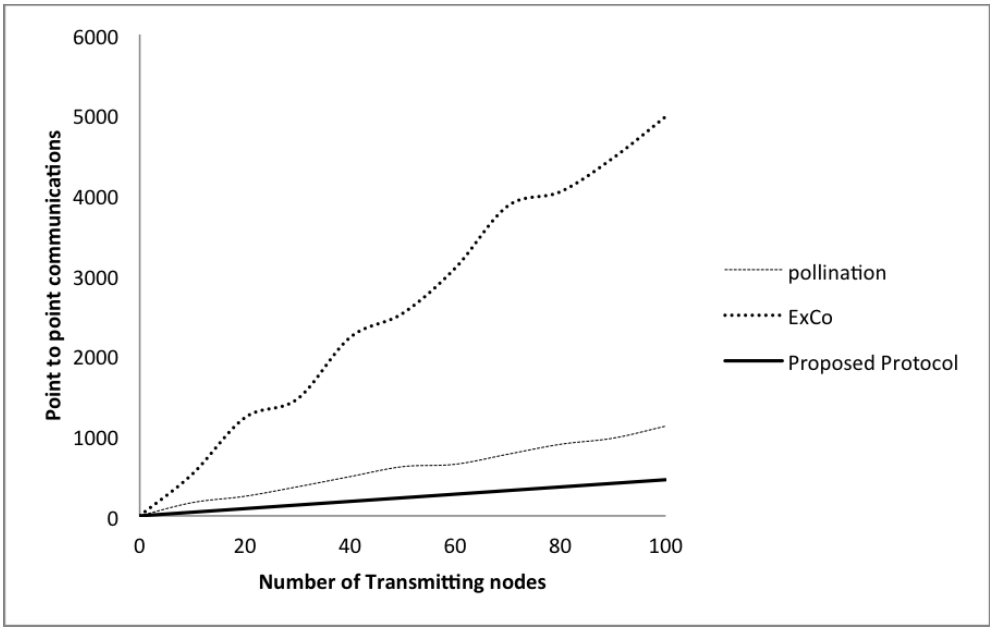

Figure 4.5: Number of point to point communications in a network of 1000 nodes

4.3.3 Storage

Since the sink visits every node once in v rounds, every node should be capable of storing data until the next visit of the sink. In the proposed protocol, every node should be capable of storing $O(vth)$ between succesive visits of the sink. The number of replications is directly proportional to number of co-authenticators (t) and number of hops (h). To save storage and communication costs, a small h value can be chosen for a large t value. TABLE 4.2 gives a summary of the storage required in ExCo, CoMAC, pollination and the proposed protocol.

Table 4.3: Storage

	Storage
ExCo	$O(vt)$ per node
CoMAC	$O(vt)$ per node
Pollination	$O(v)$ per node
Proposed Scheme	$O(vth)$ per cluster with l nodes

4.4 Summary

In this chapter, we considered the data authentication problem in Unattended Wireless Sensor Networks against a mobile adversary which tries to modify the sensor data. We proposed a data authentication scheme based on simple cryptographic hashing and replication which provides a good defence against the mobile adversary with a little communication and memory overhead. Proposed scheme maximizes the level of security in the network forcing the mobile adversary to compromise almost the entire network before manipulating the data in a single node. Proposed scheme is analyzed and evaluated based on survival probability, communication costs and storage overhead. Results prove that our scheme outperform the existing works in this area.

Chapter 5

Conclusion and Future work

5.1 Conclusions

Device compromise is a realistic threat and distributed techniques can be used to mitigate the effects of device compromise. This thesis focused on both theoretical and practical aspects of distributed techniques to protect the data generated and stored in unattended wireless sensors.

- We identified the existing data survivability problems in an emerging type of wireless sensor network called Unattended Wireless Sensor Networks. Besides exposing the problems, we probe into a number of issues having to do with the capability of the adversary and the network's tolerance against the adversary.

- We provide a thorough analysis of the conditions that can assure data survivability in an UWSN.

- We proposed two non-cryptographic schemes (DS-PADV and DS-RADV) suitable for data protection against two types of adversaries with less communication and memory overhead and with a easy routing scheme. DS-PADV can be

used against proactive adversary, and DS-RADV performs well against reactive adversary.

- We proposed a scheme to achieve data authentication against the data modification attempts made by the mobile adversary.
- Finally, we demonstrated the effectiveness of our techniques through formal proofs, theoretical analysis, implementation and simulation.

5.2 Future Work

One drawback of the proposed schemes is that there is a high chance for the nodes to expose their locations. Since every nodes create replicas and spread them in the network, it is easy for the adversary to find the location of the node based on the location of the replicas. So, location privacy is one much needed future work in this area. Apart from the location privacy, cryptographic techniques for data survival with less computational overhead also needs good deal of attention. These techniques can reduce the number of replications in the network thereby providing good security with less memory and communication overhead.

Bibliography

[1] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, G. Tsudik, "Catch me (if you can): Data Survival In Unattended Sensor Networks," in *Proc. IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, pp. 185–194, 2008.

[2] K. Römer, F. Mattern, "The Design Space Of Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey". *Computer Networks*, vol. 34, no. 4, pp. 393–422, 2002.

[4] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, G. Tsudik, "Maximizing Data Survival in Unattended Wireless Sensor Networks Against a Focused Mobile Adversary," *IACR Cryptology ePrint Archive*, pp. 1–23, 2008.

[5] M. A. S. Jr., P. S. L. M. Barreto, C. B. Margi, T. C. M. B. Carvalho, "A Survey on Key Management Mechanisms for Distributed Wireless Sensor Networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[6] S. Das, A. Nayak, S. Rührup, I. Stojmenovic, "Semi-beaconless Power and Cost Efficient Georouting with Guaranteed Delivery Using Variable Transmission Radii for Wireless Sensor Networks," in *Proc. IEEE MASS*, pp. 1–6, 2007.

[7] X. Chen, K. Makki, K. Yen, N. Pissinou, "Sensor Network Security: a Survey," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 52 –73, 2009.

[8] R. D. Pietro, N. V. Verde, "Epidemic Data Survivability in Unattended Wireless Sensor Networks," in *Proc. ACM Conf. on Wireless Network Security (WiSec)*, pp. 11–22, 2011.

[9] V. Shoup, "OAEP reconsidered," in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, pp. 239-259, 2001.

[10] S. Basagni, K. Herrin, D. Bruschi, E. Rosti, "Secure Pebblenets," in *Proc. 2nd ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 156–163, 2001.

[11] R. D. Pietro, G. Oligeri, C. Soriente, G. Tsudik, "Securing Mobile Unattended WSNs Against a Mobile Adversary," in *Proc. IEEE Symp. on Reliable Distributed Systems*, pp. 11–20, 2010.

[12] W. O. Kermack, A. Mckendrick, "A Contribution to the Mathematical Theory of Epidemics," *Royal Society of London Proceedings Series A*, vol. 115, pp. 700–721, 1927.

[13] A. S. S. Mateus, C. B. Margi, M. A. S. Jr., C. C. F. P. Geovandro, B. T. de Oliveira, "Implementation of Data Survival in Unattended Wireless Sensor Networks Using Cryptography," in *Proc. IEEE Conf. on Local Computer Networks (LCN)*, pp. 961–967, 2010.

[14] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, G. Tsudik, "Data Security in Unattended Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 58, no. 11, pp. 1500–1511, 2009.

[15] M. Sirivianos, D. Westhoff, F. Armknecht, J. Girao, "Non-Manipulable Aggregator Node Election Protocols for Wireless Sensor Networks," in *Proc. Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–10, 2007.

[16] L. Buttyan, P. Schaffer, "PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, doi:10.1155/2010/679205, 2010.

[17] R. D. Pietro, D. Ma, C. Soriente, G. Tsudik, "POSH: Proactive Co-operative Self-healing in Unattended Wireless Sensor Networks," in *Proc. IEEE Symp. on Reliable Distributed Systems*, pp. 185–194, 2008.

[18] D. Ma, G. Tsudik, "DISH: Distributed Self-Healing (in Unattended Sensor Networks)," *Cryptography ePrint Archive, Report 2008/158*, pp. 47–62, 2008.

[19] D. Ma, C. Soriente, G. Tsudik, "New Adversary and New Threats: Security in Unattended Sensor Networks," *IEEE Networks*, vol. 23, no. 2, pp. 43–48, 2009.

[20] M. Bellare, S. Miner, "A Forward-Secure Digital Signature Scheme," in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, *Lecture Notes in Computer Science*, vol. 1666, pp. 431-448, 1999.

[21] R. Gennaro , S. Jarecki , H. Krawczyk , T. Rabin, "Robust and Efficient Sharing of RSA Functions," in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, pp 157-172, 1996.

[22] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, "Robust Threshold DSS Signatures," in *Proc. EUROCRYPT*, pp. 354-371, 1996.

[23] M. Bellare, B. Yee, "Forward Integrity for Secure Audit Logs," *UCSD CSE Dept. Technical Report 23*, 1997.

[24] G. Itkis, L. Reyzin, "Forward-Secure Signatures with Optimal Signing and Verifying," in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, *Lecture Notes in Computer Science*, vol. 2139, pp. 332–354, 2001.

[25] "Information Processing Technology Office (IPTO) Defense Advanced Research Projectsv Agency v(DARPA), BAA 07-46 LANdroids Broad Agency Announcement,URL : $http$: $//www.darpa.mil/IPTO/solicit/open/BAA-07-46_PIP.pdf$", 2007.

[26] R. D. Pietro, L .V. Mancini, C. Soriente, A. Spognardi, G. Tsudik, "Playing Hide-and-Seek with a Focused Mobile Adversary in Unattended Wireless Sensor Networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1463–1475, 2009.

[27] R. Ostrovsky, M. Yung,"How to Withstand Mobile Virus Attacks," in *Proc. ACM Symp. on Principles of Distributed Computing*, pp. 51–59, 1991.

[28] R. D. Pietro, C. Soriente, A. Spognardi, G. Tsudik, "Collaborative Authentication in Unattended WSNs," in *Proc. ACM Conf. on Wireless Network Security (WiSec)*, pp. 237–244, 2009.

[29] T. Dimitriou, A. Sabouri, "Pollination: A Data Authentication Scheme for Unattended Wireless Sensor Networks," in *Proc. IEEE Trust, Security and Privacy in Computing and Communications (TrustCom)* pp. 409 –416, 2011.

[30] J. Deng, R. Han, S. Mishra, "Security Support for in-Network Processing in Wireless Sensor Networks," in *Proc. 1st ACM workshop on Security of Ad Hoc and Sensor Networks*, pp. 83–93, 2003.

[31] W. Du J. Deng, Y. S. Han, P. K. Varshney, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks" in *Proc. IEEE GLOBECOM*, pp. 1435–1439, 2003.

[32] H. Vogt, "Exploring Message Authentication in Sensor Networks," in *Proc. European Workshop Security Ad-Hoc Sensor Networks(ESAS)*, 2004; *Lecture Notes in Computer Science*, vol. 3313, pp 19–30, 2005.

[33] F. Ye, H. Luo, S. Lu, L. Zhang,"Statistical en-route Filtering of Injected False Data in Sensor Networks", *IEEE Journal on Selected Areas in Communications,* vol. 23, no. 4, pp. 839–850, 2005.

[34] J. Girao, D. Westhoff, and M. Schneider, "CDA: Concealed Data Aggregation in Wireless Sensor Networks," in *Proc. IEEE ICC*, pp. 3044–3049, 2005.

[35] J. Domingo-Ferrer, " A Provably Secure Additive and Multiplicative Privacy Homomorphism," in *Proc. Information Security Conf.*, pp. 471–483, 2002.

[36] S. Peter, K. Piotrowski, P. Langendoerfer, "On Concealed Data Aggregation for WSNs," in *Proc. 4th IEEE Consumer Communication Networking Conf. (CCNC),* pp. 192–196, 2007.

[37] Z. Li, W. Trappe, Y. Zhang, B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," in *Proc. 4th Int. Symp. on Information Processing in Sensor Networks,* 2005.

[38] S. Capkun, M. Cagalj, M. Srivastava, "Secure Localization with Hidden and Mobile Base Stations," in *Proc. IEEE INFOCOM,* 2006.

[39] J. Hwang, T. He, Y. Kim, "Detecting Phantom Nodes in Wireless Sensor Networks," in *Proc. IEEE INFOCOM,* pp. 2391–2395, 2007.

[40] Y. Chen, W. Trappe, R. P. Martin, "Attack Detection in Wireless Localization," in *Proc. IEEE INFOCOM,* 2007.

[41] D. Liu, P. Ning, W. Du, "Attack-Resistant Location Estimation in Sensor Networks," in *Proc. 4th Int. Symp. on Information Processing in Sensor Networks,* pp. 99–106, 2005.

[42] A. A. Yavuz, P. Ning, "Self-Sustaining, Efficient and Forward-Secure Cryptographic Constructions for Unattended Wireless Sensor Networks", *Ad Hoc Networks*, vol. 10, no. 7, pp. 1204-1220, 2012.

[43] R. Rivest, A. Shamir, D. Wagner, "Time-Lock Puzzles and Timed-Release Crypto," in *Technical Report, Cambridge, MA, USA*, 1996.

[44] R. Anderson, "Two Remarks on Public-Key Cryptology,invited lecture", in *Proc. 4th ACM Conf. on Computer and Communications Security*, 1997.

[45] C. Chen, Y. Tsai, "Location Privacy in Unattended Wireless Sensor Networks upon the Requirement of Data Survivability", *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1480–1490, 2011.

[46] R. Ostrovsky, M. Yung, "How to Withstand Mobile Virus Attacks," in *Proc. 10th ACM Symp. on Principles of Distributed Computing*, pp. 51–59, 1991.

[47] Y. Frankel, P. Gemmel, P. MacKenzie, M. Yung, "Proactive RSA", in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, pp. 440-454, 1997.

[48] T. Rabin, "A Simplified Approach to Threshold and Proactive RSA", in *Proc. Annual Int. Cryptology Conference (CRYPTO)*, 1998.

[49] C. Karlof, D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", in *Proc. IEEE Int. Workshop on Sensor Network Protocols and Applications*, 2003.

[50] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences", in *Proc. Annual Int. Cryptology Conference (CRYPTO)* 1992.

[51] Z. Liu, J. Ma, Y. Park, S. Xiang, "Data Security in Unattended Wireless Sensor Networks with Mobile Sinks," *Wireless Communications and Mobile Computing*, vol. 12, no. 13, pp. 1131–1146, 2012.

[52] M.A.S. Santos, C.B. Margi, "Design and Implementation of Data Survival in Unattended Wireless Sensor Networks," in *Proc. IEEE Int. Performance Computing and Communications Conference*, pp. 1–6, 2011.

[53] L. Wang and T. Jiang, X. Zhu, "Updatable Key Management Scheme with Intrusion Tolerance for Unattended Wireless Sensor Network", in *Proc. IEEE GLOBECOM*, 2011.

[54] A. Rasheed, R. Mahapatra, "Key Predistribution Schemes for Establishing Pairwise Keys with a Mobile Sink in Sensor Networks," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 1, pp. 176–184, 2011.

[55] S. Hussain, F. Kausar, A. Massod, "An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks," in *Proc. Int'l Conf. on Wireless Communications and Mobile Computing (IWCMC)*, 2007.

[56] D. Liu, P. Ning, R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," in *Proc. 10th ACM Conf. Computers and Communication Security*, pp. 52–61, Oct. 2003.

[57] W. Du, R. Wang, P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," in *Proc. ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 58–67, 2005.

[58] A. Wander, N. Gura, H. Everle, V. Gupta, S. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks," in *Proc. IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, pp. 324–328, 2005.

[59] K. Ren, W. Lou, K. Zeng, P. J. Moran, "On broadcast authentication in Wireless Sensor Networks", *IEEE Transactions on Wireless Communications,* vol. 6, no. 11, pp. 4136–4144, 2007.

[60] K. Barr, K. Asanovic, "Energy Aware Lossless Data Compression", *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, 2006.

[61] Texas Instruments Inc., "Msp430 Family of Ultra-lowpower 16-bit RISC Processors", http://www.ti.com. Accessed November 2, 2012.

[62] S. V. L. Reddy, S. Ruj, A. Nayak, "Distributed Data Survivability Schemes in Mobile Unattended Wireless Sensor Networks," in *Proc. IEEE GLOBECOM*, 2012.

[63] "The NS Manual (formerly ns Notes and Documentation)", The VINTProject, A Collaboration between researchers at UC Berkeley, LBL,USC/ISI, and Xerox PARC, August 2006.

[64] NS-2 Trace Formats, "$http : //nsnam.isi.edu/nsnam/index.php/NS - 2\_Trace\_Formats\#New\_Wireless\_Trace\_Formats$", Accessed on January 10, 2012.