Achieving Data Survivability and Confidentiality in Unattended Wireless Sensor Networks

Arpan Sen†, Shrestha Ghosh†, Arinjoy Basak†, Harsh Parsuram Puria‡, Sushmita Ruj*
† Indian Institute of Engineering Science and Technology, Shibpur, India –
Email: s.arpan1993@gmail.com, ghosh_shrestha@yahoo.co.in, basakarinjoy@gmail.com
‡National Institute of Technology, Calicut, India – Email: hpuria1608@gmail.com
*Indian Statistical Institute, Kolkata, India –Email: sush@isical.ac.in

Abstract- In Unattended Wireless Sensor Networks (UWSNs) the nodes are subjected to hostile environment for sensing critical data. Due to the unattended nature of the network the sink is not always present. Hence, the nodes in the network are required to function in a distributed way in order to ensure *Data Survivability* and *Data Confidentiality*. In this work we address these two issues. We have proposed algorithm(s) to ensure Data Survivability by *encryption* and *data replication*. We propose a simple scheme for *key management* which ensures confidentiality by sharing the key among various nodes in the network so that the adversary cannot read the data by compromising a node in the network. We have compared our scheme with the existing ones, both mathematically and by simulations. Analysis shows that our scheme performs better in terms of overheads and efficiency.

I. Introduction

A Wireless Sensor Network (WSN) consists of a very large number of nodes that are often deployed over large geographic areas in hostile environment. The sensors present in the node sense data from an environment and report it to a trusted gateway node called the sink. They are used for collecting classified military information, climate data, data monitoring for health care or maintaining water quality that requires that data collected should always be available. The all pervasive nature of WSNs demands that it be made more energy-efficient, scalable and resilient to security threats than other ad-hoc networks. Security is important for many applications, both for civilian and military purposes. A recent survey [1] highlights the security features and the drawbacks of prominent and emerging wireless ad-hoc networks.

Unattended Wireless Sensor Networks (UWSNs) were introduced in [2]. UWSNs consists of sensor nodes, which are deployed in a hostile environment to sense/collect data. A sink visits the nodes at regular intervals and collects the data stored in them. The absence of a sink during a particular period can be exploited by an adversary by reading, deleting, modifying data or injecting false data into the network.

The main issues in UWSNs are ensuring *Data Survivability, Data Confidentiality, Data Authentication* and *Data Integrity* as data remains exposed for longer periods. Data Survivability refers to the availability of the sensed data till the arrival of the sink and Data Confidentiality means that the adversary is unable to learn anything useful from the data obtained from the nodes. Both cryptographic as well as non-cryptographic techniques are utilized to achieve security and privacy in the networks. [3] addresses the problem of data availability and confidentiality by distributing shares of the data in the network such that an adversary would have to collect at least a fixed number of shares to form the data. Work in Data Authentication in UWSNs has been done in [4], [5], [6]. [2], [7], [8], [9] have proposed schemes for data survivability in mobile and static UWSNs. To recover security after a node has been compromised, co-operative self-healing was proposed in [10]. For stringent security, cryptographic techniques using public key cryptography(PKC) should be preferred.

A proactive adversary is one which starts compromising nodes before it identifies the target. The different types of proactive adversaries are discussed in [2]. We propose a scheme which ensures data survivability, data confidentiality and key management in UWSNs which will protect a data against a curious, a search-and-erase and an eraser type of proactive adversary. In our scheme we have used Data Replication to increase data survivability. Unlike [3] we have used a symmetric key to encrypt replicas of data sensed by a node, make shares of the key using Secret Sharing and distribute the encrypted data and key shares in the network. Using Public Key Encryption we encrypt the incoming data in the node during transmission thereby guaranteeing complete data confidentiality. The incorporation of data replication as well as secret sharing of the symmetric keys as done in this paper has not been addressed before to the best of our knowledge.

The use of public key techniques in WSNs has been observed in recent schemes. TinyECC has been proposed in [11], [12], where Elliptic Curve Cryptography(ECC) has been used. Different optimizations have been made for the use in WSNs. It provides flexibility to the designer as different combinations provide different execution time and resource consumption. TinyPBC presented in [13] uses Pairing-Based Cryptography(PBC) protocols where parties can agree on keys without interaction.

A. Our Contribution

In our proposed scheme we have combined the data replication technique with encryption and Shamir's secret sharing,

as defined in [14], to ensure data survivability and data confidentiality. The data confidentiality is full proof as the adversary is unable to decrypt any data, be it a replica or a key share. The probability of survival of data in our scheme is 0.755 even when all the nodes in the network are compromised by the adversary in v rounds. This result obtained is comparatively much greater and more efficient than the schemes presented in [6], [2], [15], [4]. We have provided detailed mathematical analysis and simulation results for our scheme and compared the theoretical value and the experimental value to show consistency. The communication cost and storage cost in our scheme is $O(rNh)$ and $O(rN)$ respectively, where N, r and h are the number of nodes in the network, the number of rounds, and the number of data replicas.

B. Organization

The rest of the paper is organized as follows. In section II we give the related works in the field of UWSN security. The network model, adversary model and the network assumptions are presented in section III. The detailed description of the proposed scheme is given in section IV. Section V contains the mathematical analysis of the scheme in terms of survival probability of the data. Section VI shows the simulation results. In section VII we evaluate the performance of our scheme and provide comparison with the existing ones. We conclude the work in section VIII.

II. Related Work

The problem of Data Survivability in UWSNs was first addressed in [2], by Di Pietro et al by replicating data and distributing it to other nodes to increase data survivability. Survival probability is increased by the use of replicas. Encryption in the nodes is treated only as an option, since it would increase the resource requirement of the nodes.

The scheme in [15] improves upon the known schemes by presenting two non-cryptographic algorithms (DS-PADV and DS-RADV) to ensure data survivability in mobile UWSNs with lesser communication costs. The former algorithm protects against a proactive adversary -one which compromises nodes before identifying the target. DS-RADV protects against a reactive adversary -one which compromises nodes after identifying the target.

The paper [6] by Dimitrou and Sabouri, presents a data authentication scheme called Pollination, in which the nodes participate collectively to generate the authentication codes for data. The network and adversary models are similar to the ones discussed in [2] and [15]. Compared to the schemes presented by Di Pietro et al [5], this scheme aimed to provide greater security against modification of data at lower communication overheads.

Mobile UWSNs (MUWSNs) are introduced in [3] underlying how their energy efficient scheme of local secret sharing and information diffusion provides security against data availability and confidentiality. They also specify how parameters of secret sharing should be chosen based on different mobility models.

In [4] , an authentication scheme is implemented using which the sink is able to detect any kind of modification made by the adversary, thus denying false data injection. Multiple message authentication codes(MAC) sent by the nodes are hashed and replicated to avoid single point failure.

The paper [16] uses a well known epidemic model SIS [17] and provides a solution to data survivability problem using data replication process. It focuses on at least one node storing the data instead of each node storing it.

Another paper by Di Pietro et al., [10], discusses various methods for nodes to recover from the compromise by adversary and continue to maintain secrecy of data. In particular, it discusses two cooperative self healing schemes that operate through the collaboration of nodes in the network and analyses their performance against an agile and powerful adversary.

In [18] the authors have used replication technique using random hops to ensure data survivability. Since every replica is associated to a source node, the authors have mentioned that increasing the number of replicas exposes the location of the source to the adversary. They have provided three different algorithms by which the location can be estimated- namely the coordinate median, average of overlapping area and expectation-maximization. Finally a trade-off between location privacy and data survivability was developed and the optimum number of generated data replicas was set to three.

In our work we have ensured secure transmission through the channel by using symmetric key encryption. Multiple encrypted replicas have been generated and stored in different nodes to ensure data survivability. To protect the stored data, public key encryption technique is used. Since nowadays the nodes are very powerful and efficient in terms of computational energy requirements, public key encryption schemes are feasible. As the identity of the source node is stored in encrypted form, the location of the source is not compromised. The symmetric key used for encryption of data is divided into certain number of shares and distributed throughout the network to protect the key from the adversary. Unlike other schemes, our scheme ensures that the data cannot be read by the adversary. It has to eavesdrop on multiple channels at the same time which is infeasible. In terms of data survival probability and overheads, our scheme gives better results than the aforementioned schemes.

III. Preliminaries

The system consists of a sink and a set of sensor nodes, arranged at random.

A. Network Model

The network consists of N static sensor nodes, randomly distributed over a large area. Each sensor node operates independent of all other sensor nodes in its neighbourhood and has the same communication range. Each node is considered to have sufficient memory space and computational ability required to efficiently sense and encrypt its own data as well as the data replicas and key shares that it receives from other neighbouring nodes. Each node is capable of generating multiple ciphertexts from the sensed data and secret key, as per the encryption scheme enc_{sym}, and it is assumed that it uses multicast communication to transmit them in parallel over multiple channels.

TABLE I
LIST OF NOTATIONS USED IN THE SCHEME

Symbols	Meanings
N	Total number of nodes in the network
v	Number of rounds a sink makes to cover the entire network
r	Number of data replicas
n	Number of key shares
t	Number of shares needed to construct the key
h	Number of random hops
x_i	Unique random number assigned to the i^{th} node by the sink during initialization phase
d_i	Data sensed by the i^{th} node
k_i	Secret symmetric key of the i^{th} node
pk_i	Public key of the i^{th} node
sk_i	Secret key corresponding to the pk_i of the i^{th} node
c_j	Ciphertext of the j^{th} data replica
$enc_{sym}()$	Probabilistic symmetric key encryption function
$enc_{pub}()$	Probabilistic public key encryption function
$\overline{c_j}$	Data that will be distributed to other nodes
$\overline{\overline{c_j}}$	Encrypted data replica stored in destination node
$\underline{k_i^j}$	j^{th} key share of k_i for the i^{th} node
$\overline{\underline{k_i^j}}$	Key share to be distributed to other nodes
$\overline{\overline{\underline{k_i^j}}}$	Encrypted key share stored in destination node

With each node i is associated the following — data d_i, symmetric key k_i, public key pk_i, random number x_i assigned by the sink, and encryption functions enc_{sym} and enc_{pub}.

The sink is a mobile entity which visits the network repeatedly after a regular interval of time to collect the data replicas and key shares from the nodes. The sink is a trustworthy entity. In other words, the adversary cannot compromise the sink to know any information which is stored in it.

It possesses the following — the corresponding secret key sk_i for each public key pk_i, and the number x_i assigned randomly to each node, as discussed later in the scheme.

B. Adversary Model

An adversary is an entity which tries to corrupt or harm the data of the authorized entities by some means. The adversary is completely aware of the way in which the sensor nodes are arranged in the network. In our adversarial model we assume that the adversary can eavesdrop on a communication channel to extract the data being transferred across the channel. The adversary can compromise a node by reading all the data in the node, or by deleting the data stored in a node.

The adversary attacks the network after the initialization phase, that is, after the data has been sensed, encrypted using the corresponding secret key k_i and the key shares k_i^j are generated. The adversary enters the network at time t_{entry} and leaves the network at time t_{exit} such that the time of stay in the network is very small compared to that of sink. We also assume that the number of nodes the adversary can compromise per round is much smaller compared to the number of nodes accessed by the sink per round.

C. Network Assumptions

During the design of our scheme, we made certain assumptions regarding the entities of our network system. Neither the source node nor any intermediary nodes store any information regarding which node the data has been transferred to. The symmetric key k_i of the i^{th} node is generated after v rounds by the node, and is different each time. The public key pk_i of the i^{th} node is provided by the sink during the initialization of the nodes. It is assumed that each node does not continuously sense data; that is, it senses data for a certain period of time and then waits for the sink to cover all the nodes in the network. The sink is assumed to cover all N nodes in v rounds. So the node remains idle for v rounds. The sink is assumed to be able to distinguish between the encrypted data replicas and key shares that it obtains from a particular node based on their lengths. The communication channels between any two nodes are not secure — an adversary can eavesdrop on the data being transmitted between two nodes. Since the distribution of the shares and data replicas are done parallely on multiple channels by a sensor node, it is assumed that an adversary can eavesdrop on only one channel at a particular instant of time. Thus, it can capture the data being transmitted in only one channel at a time. The adversary can read or delete the data stored in the nodes, but cannot modify the data, that is inject false data.

D. Shamir's Secret Sharing

In Shamir's secret sharing [14] we choose two positive integers t and n such that $t \leq n$. A (t, n) scheme is a method of sharing a key K among a set of n participants, in such a way that any t participants can compute the value of K but no group of $t-1$ participants can do so.

IV. THE PROPOSED SCHEME

In our scheme, the sink first visits all the N nodes and initializes them, that is it provides the random number x_i and the public keys pk_i. Then the nodes start sensing data. Whenever a node senses some data, it is encrypted by $enc_{sym}()$ using key k_i and r replicas are generated. The unique random number x_i and a hop count are then concatenated to each encrypted data. Then they are distributed. A node j on receiving an encrypted replica from another node performs $enc_{pub}()$ using its own pk_j and stores it. For the key k_i, n shares are generated. With every key share, x_i and a hop count are concatenated and these are distributed. The data d_i, key k_i and the random number x_i are then deleted. When a node receives a key share it performs the same activities as it does for data replica, and stores them.

When a sink visits the network, it collects all the stored data. It uses the corresponding sk_j to decrypt the data stored in it. If the sink collects at least one replica and at least t key shares, it is able to form k_i and hence decrypt the corresponding data. During every visit of the sink the nodes are reinitialized with new values.

A. Algorithm for the nodes sensing the Data

The i^{th} node senses data d_i. It probabilistically encrypts d_i using its symmetric key k_i, r times to generate r ciphertexts and also generates n key shares for the symmetric key k_i. For each ciphertext and key share, it generates a random hop number h and concatenates this h and x_i with it. The node then distributes

the concatenated ciphertext/key share and deletes the data d_i, k_i and x_i.

Algorithm 1 $SensorNodes()$ algorithm operated by a node n_i

Input: Sensed data d_i, Symmetric key k_i, Random number x_i
Output: Data Replica $\overline{c_j}$, Key Shares $\overline{k_i^j}$

if data d_i is sensed by i^{th} node **then**
  for j = 1 ; j $\leq$ r ; j++ **do**
    Compute $c_j = enc_{sym}(d_i, k_i)$
    Set $h =$ hop count
    Compute $\overline{c_j} = c_j||x_i||h$
  end for
  for j = 1 ; j $\leq$ r ; j++ **do**
    distribute $\overline{c_j}$
  end for
  delete d_i
  for j = 1 ; j $\leq$ n ; j++ **do**
    Generate share k_i^j of key k_i using Shamir's Secret Sharing
    Set $h =$ hop count
    Compute $\overline{k_i^j} = k_i^j||x_i||h$
  end for
  for j = 1 ; j $\leq$ n ; j++ **do**
    distribute $\overline{k_i^j}$
  end for
  delete k_i
  delete x_i
end if

B. Algorithm for the intermediate nodes

If a node receives a key share or a data replica, it first checks whether the hop count h is 0 or not. If it is not 0, it reduces the hop count h by 1 and forwards the key share or the data replica to a randomly selected neighbouring node.

Algorithm 2 $InterNodes()$ operated by a node

Input: Data Replica $\overline{c_j}$ or Key Share $\overline{k_i^j}$
Output: Data Replica $\overline{c_j}$ or Key Share $\overline{k_i^j}$ with reduced hop count

if $h > 0$ **then**
  Set $h = h - 1$
  if a node receives $\overline{c_j}$ **then**
    forward $\overline{c_j}$ randomly to another node
  end if
  if a node receives $\overline{k_i^j}$ **then**
    forward $\overline{k_i^j}$ randomly to another node
  end if
end if

C. Algorithm for the nodes storing the Replicas and shares

The node which will store the data replica or the key share will encrypt the item to be stored with its own public key. The corresponding secret keys are held by the sink for decryption. After encrypting the data replica or the key share, the node stores it.

Algorithm 3 $StoreNodes()$ operated by a node

Input: Data Replica $\overline{c_j}$ or Key Share $\overline{k_i^j}$
Output: Encrypted Data Replica $\overline{\overline{c_j}}$ or Encrypted Key Share $\overline{\overline{k_i^j}}$

if $h == 0$ **then**
  if a node y receives $\overline{c_j}$ **then**
    Compute $\overline{\overline{c_j}} = enc_{pub}(\overline{c_j}, pk_y)$
    Store $\overline{\overline{c_j}}$
  end if
  if a node y receives $\overline{k_i^j}$ **then**
    Compute $\overline{\overline{k_i^j}} = enc_{pub}(\overline{k_i^j}, pk_y)$
    Store $\overline{\overline{k_i^j}}$
  end if
end if

D. Algorithm for Sink

The sink can distinguish between a key share and a data replica. On obtaining data from a node, it decrypts it using the node's sk_i. For both the key shares and data replicas it searches the x_i in its precomputed table and maps it to the generating node. The sink can decrypt a data if and only if it has at least one data replica and t shares of the corresponding symmetric key to decrypt it.

Algorithm 4 $Sink()$ operated by the mobile sink

Input: Encrypted Data Replica $\overline{\overline{c_j}}$ and Encrypted Key Share $\overline{\overline{k_i^j}}$
Output: Plaintext data d_i

if sink collects $\overline{\overline{k_i^j}}$ from p^{th} node **then**
  Compute $\overline{k_i^j} = dec_{pub}(\overline{\overline{k_i^j}}, sk_p)$
  Match x_i of $\overline{k_i^j}$ with its precomputed table
  if match found AND number of such entries $== t$ **then**
    Compute k_i from the shares
  end if
end if
if sink collects $\overline{\overline{c_j}}$ from p^{th} node **then**
  Compute $\overline{c_j} = dec_{pub}(\overline{\overline{c_j}}, sk_p)$
  Match x_i of $\overline{c_j}$ with its precomputed table
  if match found AND sink has key k_i **then**
    Calculate $d_i = dec_{sym}(c_j, k_i)$
  end if
end if

V. Mathematical Analysis

Here we provide an analysis of our proposed scheme in terms of data survivability where the adversary is capable of *reading* and *deleting* the data.

First we show that the adversary is unable to decrypt any data. The data can be read in two scenarios - when the data

is being transmitted through the channel or when the data has already been stored in the nodes after its distribution. In the second case there is no way the adversary can decrypt the data since every data stored in the nodes is encrypted with its public key.

In the first case the adversary can get hold of either a data replica, $\overline{c_j}$, or a key share, $\overline{k_i^j}$ by eavesdropping on a channel. It is not sufficient to acquire just a data replica as the symmetric key with which it has been encrypted is made into shares and distributed uniformly randomly in the network. The adversary will have to be present in multiple channels simultaneously in order to intercept the data and the key shares from the channels before it reaches the destination node and is encrypted with the node's public key. This can be summarized in the following theorem:

Theorem 1: In the UWSN it is impossible for an adversary to decrypt an intercepted data during transmission.

For an adversary to decrypt an intercepted data, it needs a symmetric key which has to be constructed from the shares of the key that have been distributed in the network via random hops.

The case where the adversary deletes the data it reads, we find the probability of survival of at least one data replica.

The number of nodes covered by the sink after n rounds is given by $(n-1)s$ and number of nodes that the adversary has not compromised is given by $N-(n-1)q$, where s is the number of nodes visited by the sink per round and q is the number of nodes compromised by the adversary per round.

Probability that a node containing $\overline{\overline{c_j}}$ is compromised in n^{th} round is the probability that it is not covered by the sink till the n^{th} round and is compromised by the adversary in round n. It is given by,

$$P_1 = (\frac{N-(n-1)s}{N})(\frac{q}{N-(n-1)q})$$

Probability that a data replica is safe till the sink covers the entire network in v rounds,

$$P_2 = \prod_{a=0}^{v-1}(1-(\frac{N-as}{N})(\frac{q}{N-aq}))$$

Probability that a replica is compromised before the end of v rounds, $P_3 = 1 - P_2$
Probability that all the replicas are compromised, $P_4 = P_3^r$

$$P_4 = [1-\prod_{a=0}^{v-1}(1-(\frac{N-as}{N})(\frac{q}{N-aq}))]^r$$

Probability that at least one replica is safe, $P_D = 1 - P_4$

$$P_D = 1-[1-\prod_{a=0}^{v-1}(1-(\frac{N-as}{N})(\frac{q}{N-aq}))]^r$$

Theorem 2: : If N, v, s, q, r be the total number of nodes in the network, number of rounds, number of nodes visited by sink per round, number of nodes compromised by the adversary per round and the number of data replicas generated respectively, then the survival probability of the data is given by P_D where,

$$P_D = 1-[1-\prod_{a=0}^{v-1}(1-(\frac{N-as}{N})(\frac{q}{N-aq}))]^r$$

VI. Simulation

In this section we perform the simulation of our scheme. Our primary objective is to determine an optimal value for the number of replicas r and the parameters t and n of the (t, n) secret sharing scheme. With the help of various results obtained from our simulation we can successfully determine the values of the above mentioned parameters.

A. Simulation Environment

In our simulation we have considered 1000 nodes to be randomly distributed in the network. Each node has an unique id which is used to identify the nodes. The adversary is not aware of these unique ids. A neighbour of a given node is any node which lies in the communication (transmission) range of the node under consideration. The nodes use random hop method to distribute the data replica and the key shares throughout the network where the number of hops is selected at random. After the distribution of the data replicas and key shares the adversary and the trusted sink visit the network in rounds to compromise and collect data respectively. The number of such rounds is set to 5. The total number of nodes compromised by the adversary in 5 rounds is varied from 100 to 1000 with an interval of 100 to obtain the required data. The number of replicas generated (r) are taken to be equal to 2, 3 and 4. The key sharing schemes used are $(3, 3)$, $(3, 4)$ and $(3, 5)$. The results have been considered on an average over 500 simulations.

B. Simulation Results

In this section we will discuss and analyse the various plots obtained from the simulations and hence, fix an appropriate value for r,t and n.

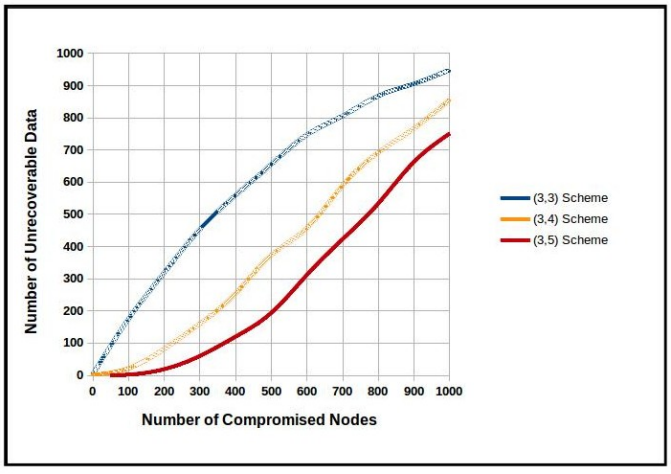

Fig. 1. Number of unrecoverable data with varying number of compromised nodes for 3 data replicas and various key sharing schemes

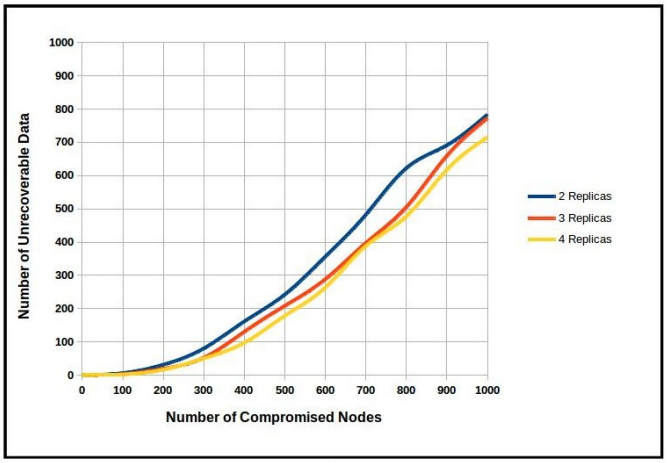

Fig. 2. Number of unrecoverable data with varying number of compromised nodes for (3,5) key sharing and various number of data replicas

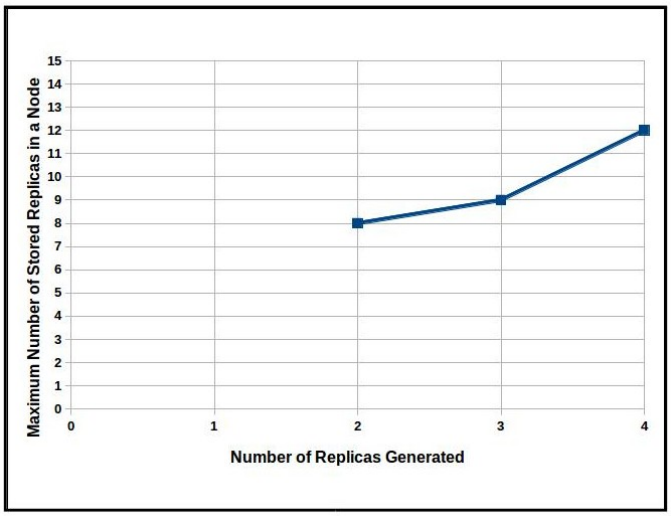

Fig. 3. Maximum number of stored replicas in a node for various number of generated replicas

Figure 1 shows the variation of the number of unrecoverable data with change in number of compromised nodes for various key sharing schemes. We observe that for $(3, 3)$ and $(3, 4)$ key sharing schemes, the number of unrecoverable data are much higher compared to a $(3, 5)$ scheme. This is because the adversary has to compromise only 1 and 2 key shares to destroy the key in case of $(3, 3)$ and $(3, 4)$ sharing schemes respectively. Whereas in case of $(3, 5)$ scheme the adversary has to delete at least 3 shares to destroy the key, thus increasing the probability of key survival. Hence we have chosen $(3, 5)$ key sharing to be used in our proposed scheme.

Figure 2 shows the variation of the number of unrecoverable data with change in number of compromised nodes for various number of generated replicas and a $(3, 5)$ key sharing scheme. The number of unrecoverable data for 2 replicas is notably higher than that for 3 and 4 replicas for all values of the number of compromised nodes. So using 2 replicas is not feasible. The curves for 3 and 4 replicas are very close to one another and they almost overlap when the number of compromised nodes are less than 300. We may use either of them but from the graph in Figure 3 we observe that for a 4 replica scheme, the maximum number of data units stored in a node increases steeply than that for a 3 replica scheme. Hence considering a trade off between storage and security, we have used 3 replicas in our scheme.

From Figure 1 and Figure 2 it can be seen that for all schemes, the number of unrecoverable data is very high when the number of compromised nodes are very high (> 500). In our assumptions we have considered that if the adversary remains undetected, it has to be present in the network for much less duration of time as compared to sink. Hence if the sink visits all the 1000 nodes then the adversary will only be able to visit a small fraction of them (< 250). Thus considering the range of compromised nodes to be between 0 and 250, our chosen scheme runs efficiently giving satisfactory results.
So the value of the parameters that we have fixed are $r = 3$, $t = 3$ and $n = 5$.

VII. Performance Evaluation

We have compared the results obtained from our simulations to the results obtained in [2], [15], [6] and [4] in terms of survival probability, storage and communication costs. We also provide an analysis of the scalability of our scheme for a large network.

The Table II depicts the the issues discussed in the various schemes that have been discussed in the papers. A "✓" indicates that the corresponding feature in the column was addressed by the paper in the corresponding row, and "×" indicates otherwise. It is observed that although some papers have discussed survivability and data authentication, none have discussed data survivability and data confidentiality jointly in any of their schemes. This is what we have attempted to contribute in our paper.

A. Data Survivability

Figure 4 demonstrates the comparison between our scheme and the schemes proposed in [2], [15], [6] and [4] based on the results obtained in our simulations.

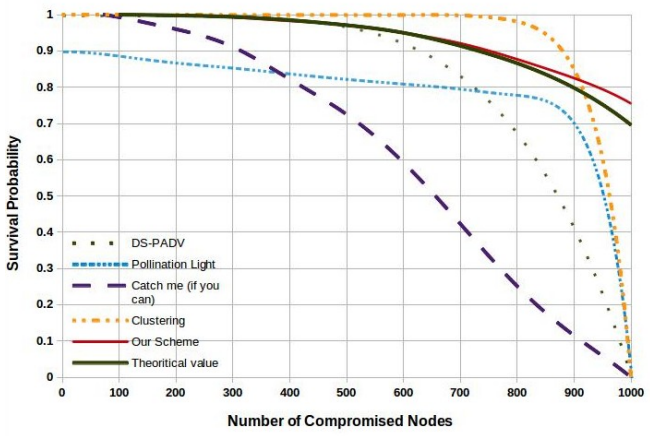

Fig. 4. Comparison of Survival Probability of other Schemes with our scheme

TABLE II
PROBLEMS ADDRESSED IN OTHER SCHEMES AND IN OURS, COMPARED

Schemes	Data Survivability	Data Confidentiality	Data Authentication
Catch me If you can [2]	✓	×	×
DS-PADV [15]	✓	×	×
Pollination [6]	✓	×	✓
Clustering [4]	✓	×	✓
Our Scheme	✓	✓	×

It can be seen from Figure 4 that there is a striking contrast between our scheme and all the others as number of compromised nodes increases. While there is a sudden drop in all the graphs, our plot gradually decreases to somewhere between 0.7 and 0.8. Even when the number of nodes compromised in v rounds equals the total number of nodes present in the network, the data survives with a probability of 0.755. The reason for this is that the sink collects data from the network at the end of each round negating any future attempts by the adversary to delete data from the node already visited by the sink in one of the previous rounds. Therefore even with the adversary visiting the entire network, effectively it is able to compromise only a little above 25% of the nodes.

B. Trade-Off between Data Survivability and Data Confidentiality

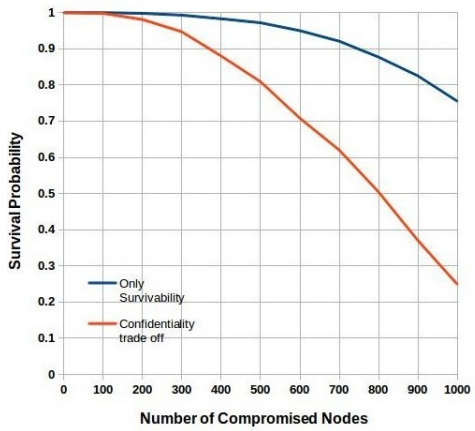

Fig. 5. Survival Probability for Data Confidentiality and Data Survivability

Figure 5 presents the effect of a trade off between data survivability and data confidentiality. When confidentiality is not considered, then the survival probability is high using only replication and public key encryption. Here, the data is transmitted in raw form. The data can be gathered if the transmission channel is attacked. However, if confidentiality is of greater importance then we have to make a trade off between survivability and confidentiality. To implement confidentiality we have used a symmetric key encryption scheme and the keys have been shared by Shamir's Secret Sharing technique. Now for the data to survive it must be ensured that at least one replica survives and minimum number of shares required to construct the symmetric key also survives. Hence there is a decrease in data survivability.

C. Scalability

Figure 6 is a measure of the performance of our scheme in terms of accommodation of increasing number of nodes in the network.

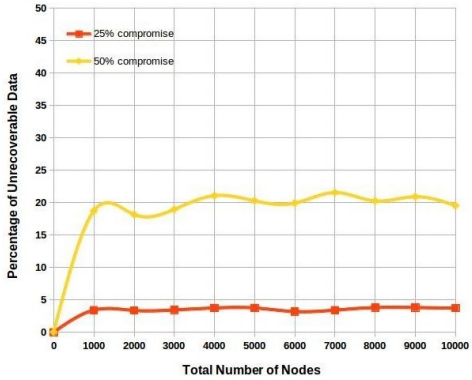

Fig. 6. Scalability of the scheme

There are two plots in which the adversary compromises 25% and 50% of the total number of nodes in the network respectively. By compromising a node we mean that the adversary deletes all data from the visited node. In the first case the graph is almost horizontal while in the second case the graph fluctuates within 2% of the 20% line denoting the percentage of unrecoverable data in the network. The graph is plotted for nodes in the network starting from 1000 and going upto 10000 increasing the number of nodes by 1000 in each step. We performed 1000 simulations each for 1000 nodes upto 10000 nodes. Thus, our scheme is scalable.

D. Overheads

The overheads incurred in communication depends linearly on the number of replicas r, number of nodes in the network, N, and the number of hops, h. The communication overhead is measured as the number of transmissions required to transmit the data from a source node to a destination node. For storage there is a linear dependence on the number of replicas r, and the number of nodes in the network, N. Table III draws a comparison between the scheme we present and [2], [15], [6] and [4]. It is also observed that our scheme performs better than the others in terms of overheads.

TABLE III
COMPARISON WITH OTHER SCHEMES IN TERMS OF COMMUNICATION AND STORAGE COSTS

	Pollination [6]	DS-PADV [15]	DS-RADV [15]	Clustering [4]	Our Scheme
Communication	$O(N\sqrt{N})$	$O(Nh)$	$O(rNh)$	$O(rN\sqrt{N}/l)$	$O(rNh)$
Storage	$O(vN)$	$O(vNh)$	$O(vrN)$	$O(vrNh/l)$	$O(rN)$

Where N = Number of Nodes, v = Number of Rounds, r = Number of Replicas, h = Number of hops, l = Number of nodes per Cluster

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the problems of both data survivability and data confidentiality in UWSNs by presenting a scheme which ensures data survivability using data replication and distribution by multihop paths. Our scheme ensures data confidentiality through a key sharing strategy based on Shamir's Secret Sharing and public key encryption. We have also shown that our scheme ensures that the data cannot be decrypted, no matter what the adversary does within its limited capability. The performance of our scheme has also been analysed, to obtain the optimum values for number of replicas, key shares and the parameters for secret-sharing, and a comparison to other works has been done to show the efficiency of our scheme over other previous works. Our scheme has also been proved to be highly scalable, indicating its practicability.

Our work does not consider an adversary which can modify and/or inject false data. It requires a scheme which facilitates authentication of data when it is collected by the sink, and thus detect the errors in data and the presence of an adversary. This could serve as an improvement upon our current model, and thus form the grounds for future work based on this topic. A further work may explore data confidentiality and survivability in a network model in which the nodes are mobile, as opposed to the static nodes considered in our model. Thus, these are the possible scopes for further development on our current work.

REFERENCES

[1] R. D. Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in wireless ad-hoc networks - A survey," *Computer Communications*, vol. 51, pp. 1–20, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2014.06.003

[2] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch me (if you can): Data survival in unattended sensor networks," in *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), 17-21 March 2008, Hong Kong*. IEEE Computer Society, 2008, pp. 185–194. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/PERCOM.2008.31

[3] R. D. Pietro and S. Guarino, "Data confidentiality and availability via secret sharing and node mobility in UWSN," in *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*. IEEE, 2013, pp. 205–209. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2013.6566764

[4] S. K. V. L. Reddy, S. Ruj, and A. Nayak, "Data authentication scheme for unattended wireless sensor networks against a mobile adversary," in *2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, Shanghai, China, April 7-10, 2013*. IEEE, 2013, pp. 1836–1841. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2013.6554843

[5] R. D. Pietro, C. Soriente, A. Spognardi, and G. Tsudik, "Collaborative authentication in unattended wsns," in *Proceedings of the Second ACM Conference on Wireless Network Security, WISEC 2009, Zurich, Switzerland, March 16-19, 2009*, D. A. Basin, S. Capkun, and W. Lee, Eds. ACM, 2009, pp. 237–244. [Online]. Available: http://doi.acm.org/10.1145/1514274.1514307

[6] T. Dimitriou and A. Sabouri, "Pollination: A data authentication scheme for unattended wireless sensor networks," *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, vol. 0, pp. 409–416, 2011.

[7] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Maximizing data survival in unattended wireless sensor networks against a focused mobile adversary," *IACR Cryptology ePrint Archive*, vol. 2008, p. 293, 2008. [Online]. Available: http://eprint.iacr.org/2008/293

[8] M. A. S. Santos, C. B. Margi, M. A. S. Jr., C. C. F. P. Geovandro, and B. T. de Oliveira, "Implementation of data survival in unattended wireless sensor networks using cryptography," in *The 35th Annual IEEE Conference on Local Computer Networks, LCN 2010, 10-14 October 2010, Denver, Colorado, USA, Proceedings*. IEEE, 2010, pp. 961–967. [Online]. Available: http://dx.doi.org/10.1109/LCN.2010.5735841

[9] R. D. Pietro, G. Oligeri, C. Soriente, and G. Tsudik, "Securing mobile unattended wsns against a mobile adversary," in *29th IEEE Symposium on Reliable Distributed Systems (SRDS 2010), New Delhi, Punjab, India, October 31 - November 3, 2010*. IEEE, 2010, pp. 11–20. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/SRDS.2010.10

[10] R. D. Pietro, D. Ma, C. Soriente, and G. Tsudik, "Self-healing in unattended wireless sensor networks," *TOSN*, vol. 9, no. 1, p. 7, 2012. [Online]. Available: http://doi.acm.org/10.1145/2379799.2379806

[11] A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," *International Conference on Information Processing in Sensor Networks*, vol. 0, pp. 245–256, 2008.

[12] H. Seo, K. Shim, and H. Kim, "Performance enhancement of tinyecc based on multiplication optimizations," *Security and Communication Networks*, vol. 6, no. 2, pp. 151–160, 2013. [Online]. Available: http://dx.doi.org/10.1002/sec.422

[13] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.05.013

[14] D. R. Stinson, *Cryptography-Theory and Practice*. Chapman and Hall/CRC, 2006.

[15] S. K. V. L. Reddy, S. Ruj, and A. Nayak, "Distributed data survivability schemes in mobile unattended wireless sensor networks," in *2012 IEEE Global Communications Conference, GLOBECOM 2012, Anaheim, CA, USA, December 3-7, 2012*. IEEE, 2012, pp. 979–984. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2012.6503240

[16] R. D. Pietro and N. V. Verde, "Epidemic data survivability in unattended wireless sensor networks," in *Proceedings of the Fourth ACM Conference on Wireless Network Security, WISEC 2011, Hamburg, Germany, June 14-17, 2011*, D. Gollmann, D. Westhoff, G. Tsudik, and N. Asokan, Eds. ACM, 2011, pp. 11–22. [Online]. Available: http://doi.acm.org/10.1145/1998412.1998417

[17] W. O. Kermack and A. McKendrick, "A Contribution to the Mathematical Theory of Epidemics," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 115, no. 772, pp. 700–721, Aug. 1927.

[18] C. Chen and 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. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/JSAC.2011.110813