Secure and Efficient DiDrip Protocol for Improving Performance of WSNs

Shobhna Vedprakash Pandey[1], Dr. K. Srujan Raju[2]

[1]M.Tech Student,CMR Technical Campus, Kandlakoya(V), Medchal(M),Ranga Reddy District, Telangana, India
[2]HOD, CSE Department,CMR Technical Campus, Kandlakoya(V), Medchal(M),Ranga Reddy District, Telangana, India

Abstract— *Wireless Sensor Networks consists of a set of resource constrained devices called nodes that communicate wirelessly with each other. Wireless Sensor Networks have become a key application in number of technologies. It also measures the unit of vulnerability to security threats. Several Protocols are projected to make them secure. Some of the protocols within the sensor network specialize in securing data. These protocols are named as data discovery and dissemination protocols. The data discovery and dissemination protocol for wireless sensor networks are utilized for distributing management commands and altering configuration parameters to the sensor nodes. All existing data discovery and dissemination protocols primarily suffer from two drawbacks. Basically, they are support centralized approach (only single station can distribute data item).This approach is not suitable for multiple owner-multiple users. Second, the protocols are not designed with security in mind. This Paper proposes the first distributed knowledge discovery and dissemination protocol called DiDrip which is safer than the existing one. The protocol permits multiple owners to authorize many network users with altogether totally different priorities to at an equivalent time and directly flow into data items to sensor nodes.*

Keywords— Wireless Sensor Networks, Security threats, Distribution Techniques, centralized approach

I. INTRODUCTION

Wireless Sensor Network (WSN) is a dense network consisting of little and light-weight nodes, which are broadcasted over the system in giant numbers by the measurement of physical parameters like temperature, pressure, ratio, etc. [1]. In WSNs some common variables could also be held in every node of the network. The data discovery protocols are accountable to feature, delete, and configure such variables by requesting every node to exchange packets in order that they eventually become consistent across the network [2]. In the literature, several data discovery and dissemination protocols are projected. The proposed protocols specialize in economical and reliable knowledge dissemination. Certain security concerns in several protocols remain an unresolved issue. Security vulnerability, as a result of the open nature of wireless communication channels and lack of protection of individual sensor nodes, makes it easy for the intruders to interrupt the communication. Sensor data for surrounding networks must not be leaked. In many applications such as ones concerning military and security nodes store and communicate sensitive data. Data confidentiality keeps the sensitive data secret by encrypting the data with a secret key that is solely meant for the receivers.

The proposed work will increase the confidentiality of sensitive knowledge throughout transmission and avoid fallacies of the existing system. The main contributions of this paper are: a) replacement approach to handle the safety problems in WSN victimization cryptography technique; b) Cryptography of sensitive knowledge wherever the amount of encryption will be controlled by the key generated. Thus, it limits the usage of resources offered by providing better security.

Our paper is structured as follows: we present an overview of the related literature concerning dissemination protocols in Section 2. We present the DiDrip framework in Section 3. Discussions and Conclusions are presented in Section 4. The contents of each section may be provided to understand easily about the paper.

II. REVIEW OF THE LITERATURE

Considering the objective of the paper, systematic review of the extant literature has been adopted as a suitable approach. The extant literature on Wireless Sensor Network, Drip, CodeDrip, DIP, DHV and Typhoon has been systematically reviewed and presented in the sub-sections below:

2.1 Wireless Sensor Networks:

A WSN consists of variety of nodes used for watching function that pass the data collected through the network to a main location [3]. The usage of wireless sensor networks was intended principally by military applications. However, these days WSN are used popularly in several applications like device and watching, environmental watching, attention

management, construction safety, emergency response information, logistics and inventory management, etc.

2.1.1 Characteristics of WSN:

The WSN consists of nodes starting from few to several hundred, where every node is connected to at least one or many sensors. Every node has many components such as microcontroller, radio transceiver, a circuit for interfacing with the sensors and electric battery. Sensor networks are sometimes setup in remote and hostile environment. Therefore size and price are strict constraints that lead to corresponding constraints on resources like machine speed, energy, memory and communications data measurement. The topology of WSNs is either a star network or a multihop wireless mesh network. WSNs usually operate for long period time and usually do not get any human administration or intervention. The remote nature of WSNs needs the propagation of latest code over the network as manual updating of such networks is not attainable. This method is known as dissemination. But this poses multiple challenges in system and network design [4]. One of the challenges is effective dissemination of data to any sensor node within the network. This is often not a simple task since the amount of nodes in a particular sensor network is large and therefore the environment is dynamic in nature i.e., nodes will die or move, and so topology will keep on modifying continuously. Also, depending upon the application the knowledge to be disseminated can be originating at one node, such as the base station, or at multiple nodes for instance sensor nodes themselves. Therefore knowledge dissemination in neighborhood is to be studied deeply and more meticulously.

2.1.2 Routing in Wireless Sensor Network:

Due to recent technological advances, the bearing of little and low price sensors became technically and economically feasible. The sensing electronics measure close conditions associated with the atmosphere thereby closing the sensor and transforming them into an electrical signal. Processes such as indication reveals some properties regarding objects placed and/or events happening within a given section of the sensor. An oversized number of those disposable sensors are networked in several applications that need unattended operations. A Wireless sensor Network (WSN) contains hundreds and thousands of those sensor nodes. These sensors have the flexibility to either associate among themselves or on to an external Base-Station (BS). A larger number of sensors permit for sensing over larger area with greater accuracy. Within the past few years, an intensive analysis that addresses the potential of collaboration among sensors in knowledge gathering and process and within the coordination and management of the sensing activity were conducted. However, sensor nodes are constrained in energy supply as well as bandwidth. Thus, innovative techniques that eliminate energy inefficiencies that would shorten the lifetime of the network are highly needed. Such constraints combined with a typical deployment of huge number of sensor nodes pose many challenges to the planning and management of WSNs and necessitates energy-awareness at all layers of the networking protocol stack. For instance, at the network layer, it is highly desirable to search out methods for energy-efficient route discovery as well as relaying of information from the sensor nodes to the Base Station (BS) so the lifetime of the network is maximized. A System overview of centralized and distributed data discovery and dissemination approaches is presented in Figure 1.

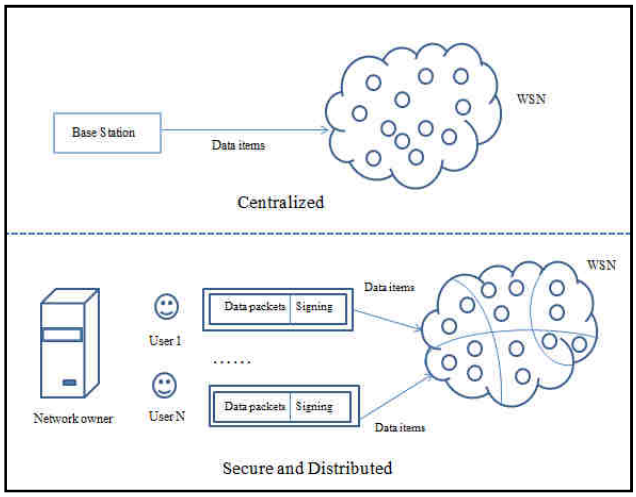

Fig.1: System overview of centralized and distributed data discovery and dissemination

2.2 Drip

Tolle et al. presented Sensor Network Management System (SNMS), which is an application - cooperative management system for WSN and Drip is the dissemination protocol which is employed in it [5]. Drip is the simplest of all dissemination protocols and depends on trickle rule and establishes a self-employed trickle for each variable within the knowledge. Every time an application wants to transmit a message, a replacement version id is generated and used. This can cause the protocol to reset the Trickle timer and hence disseminates the new value. Drip provides a typical message reception interface in WSN. Every node that needs to use Drip can register with a specific symbol that represents a dissemination channel. All messages received thereon in the channel are going to be delivered on to the node. Every node is additionally responsible for caching the data extracted from the recent message received on every channel to which it subscribes, and returns it in response to the periodic transmit requests. Drip achieves effectiveness by avoiding redundant transmissions if an

equivalent data has already been received by the nodes within the neighborhood.

2.3 CodeDrip

It is a data dissemination protocol proposed by Nildo et al. and can be employed in Wireless Sensor Networks [6]. This protocol is especially used for dissemination of small values. Network committal to writing is a mechanism that mixes packets in the network which would increase the out turn and decrease number of messages transmitted. CodeDrip uses Network Coding to improve the reliability and speed of dissemination. Instead of merely retransmitting received knowledge packets, device nodes attempt to mix varied packets of small knowledge items into one, and re-transmit the combined packet to its neighbors. Thus, packet loss is avoided since lost packets could be recovered through the secret writing of others combined packets. By avoiding frequent retransmissions, the dissemination method finishes quicker. CodeDrip uses the Trickle rule for dissemination. It is similar to Drip apart from the very fact that here messages are sometimes combined and sent. To mix messages, coding protocols use completely different operators, here XOR operator is employed. This alternative allows Drip to run with efficiency on resource constraint nodes. In this, the packet format for Drip is changed to accommodate the management fields needed by network coding [7]. The secret writing aspect should apprehend that messages were combined to make the given payload. Thus raising the message header to a point where messages can be combined. At the destination we utilize this field to work out whether a message received is an actual message or a combined message.

2.4 DIP

DIP (Dissemination Protocol) is a data detection and dissemination protocol as suggested by Lin et al [8]. It is a protocol supported by the trickle algorithmic rule. It works in two parts: detection whether a difference in data in nodes has occurred, and distinguishing that data item is completely different. It uses the concept of version variety and keys for every data item. Within the steady state all nodes are up to date and have similar versions. DIP utilizes trickle algorithm to calculate and send hashes that count all of the version numbers [9]. Nodes that receive hashes that are a similar as their own understand they need consistent data with respect to their neighbors. If a node identifies a hash that differs from its own, it detects that a distinction exists, however it does not understand that data item incorporates a newer version. Distinguishing that data item is completely different and that node has the newer version which requires exchange of particular version numbers. Additionally to the version variety, DIP also manages a soft state estimate of whether a given item differs from a neighbors item or not. Whenever a node receives a packet and also the hashes which are same the estimate is decremented to a minimum of zero (0). Otherwise if hashes disagree the estimate is incremented. This goes on till the estimates converge to zero which implies all have similar data.

2.5 DHV

DHV is a code consistency managing protocol proposed by Dang et al [10]. The name DHV originates from three steps in the protocol – Difference detection, Horizontal search and Vertical search. It tries to maintain codes on completely different nodes during a WSN, consistent and up to date. Here data items are described as tuples (key, version).This protocol tries to make up for the disadvantages of previous protocols like DRIP and DIP by reducing the quality concern within the updating of data in the network. It supports the observation that if two versions are completely different, they will only differ during a few least vital bits of their version number rather than entire bits. Hence, it's not forever necessary to transmit and compare the complete version number within the network. Here the version number is given as a bit array. DHV utilizes bit slicing to quickly verify the out of date code, leading to fewer bits being transmitted within the network. DHV includes two necessary phases: detection and identification. In detection, every node can broadcast a hash of all its versions during an outline message. Upon receiving this, a node compares it to its hash. If they're not similar, there is one or a lot of code items with a special version number. In identification, the horizontal search and vertical search steps are accustomed determining the difference in the versions. Throughout horizontal search, a node broadcasts a checksum of all its versions during a HSUM message. Upon receiving this, a node compares the checksum to its own checksum to spot that bits are completely different and moves to the following step. In vertical search, the nodes can broadcast a bit slice, beginning at the LSB of all versions, that a VBIT message. If the bit indices are matching, and also the hashes are completely different, the node can broadcast a bit slice of index zero and increase the bit index to search out varied locations till the hashes become same. Once obtaining a VBIT message, a node compares it to its own VBIT to spot the locations equivalent to the differing tuples. Once distinctive, the node broadcasts those (key, version) tuples during a VECTOR message. Upon receiving a VECTOR message, a node compares it to its own (key, version) tuple to make a decision which has the newer version and whether it ought to broadcast its data. A node with a more recent version can broadcast its data to nodes with an older version.

2.6 Typhoon

It is a reliable data dissemination protocol utilized in wireless sensor networks given by Liang et al [11]. It is

predominantly used for dissemination of large data like Deluge. Therefore in this case as well massive data objects are divided into fixed sized pages or packets. In contrast to different protocols, typhoon sends data packets in unicast fashion. This approach permits receivers to acknowledge the receipt of packets and hence quickly recover lost packets if any. Whereas data packets are sent in unicast manner, interested nodes will receive those packets by snooping on the wireless medium. So through the mixture of unicasting and snooping, this protocol achieves prompt retransmissions and data delivery to any or all the nodes during a broadcast domain through one transmission. Typhoon uses trickle timers for dissemination of meta-data. Here meta-data includes object ID, size and version to point the existence of a freshly created data object. Counting on comparisons of meta-data nodes arrange to accept or not accept new data objects. Here all protocol choices are aiming to minimize the idle listening time of nodes i.e. non transmission or non-receiving data packets. The nodes continuously try to aim to push data into the network as quickly as possible. Also spatial reuse is employed, through which nodes in numerous components of the network will be transmitted at an equivalent time. Different techniques which will be employed are duty cycling, turning nodes off once not in use and so on.

2.7 MNP

Sandeep et al. projected a Multihop Network reprogramming Protocol (MNP) [12], [13]. It provides a reliable service to propagate new program code to all sensor nodes within the network. The primary objective of this dissemination protocol is to confirm reliable, low memory usage and quick knowledge dissemination. It is based on a sender choice protocol within which supply nodes compete with one another based on the number of distinct requests they have received. In every neighborhood, a source node sends out program code to multiple receivers. Once the receiver gets the complete program image at their side, they become source nodes, and send the code into their neighborhood. However there are frequent problems with collisions. This can be solved by choosing an acceptable sensor node based on some parameters maintained by the nodes and a few packaging and download messages changed by the nodes. It is like a greedy algorithmic rule. Pipelining is often included during this protocol to enable quicker knowledge propagation within the case of larger networks. To do pipelining, programs are divided into segments, each of these segments contain a fixed number of packets. Once a detector node receives all the segments of a program, it will reboot with the new program. This continues until all the nodes are thence updated.

Table 1. Comparative summary of various Protocols

Protocols	**Authors**	**Algorithm used**	**Data used**	**Distinctive features**	**Security**	**Base Architecture**	**Network Owner**
Drip	Tolle et al.	Trickle	Short Configuration parameters	Simplest of all protocols	No	Centralized	One owner
CodeDrip	Nildo et al.	Trickle	Short values	Used Network Coding	No	Centralized	One owner
DIP	Lin et al.	Trickle	Multiple data items	Definitive	No	Centralized	One owner
DHV	Dang et al.	Trickle	Multiple values	Quicker	No	Centralized	One owner
Typhoon	Liang et al.	Trickle for metadata	Large Data Objects	Spatial multiplexing	No	Centralized	One owner
MNP	Sandeep et al.	Trickle	Network Reprogramming code	Energy efficient	No	Centralized	One owner
DiDrip	He et al.	Trickle	All data are used	Multi-owner-Multiuser, Authorization through user access, Node	Yes	Distributed	Multiple owners

				compromise tolerance, user collision tolerance, DoS attacks resistance, less energy overhead			

A comparison of all the protocols discussed above is presented in the tabular form in Table 1. highlighting the differences on algorithms, data, security architecture, network owner and other distinctive features.

III. DIDRIP FRAMEWORK

DiDrip consists of four phases as shown in Figure 2, system format, user joining, packet pre-processing and packet verification. For our basic protocol, in system formatting part, the network owner creates its public and private keys, so masses the general public parameters on each node before the network deployment. In user joining part, a user gets the dissemination privilege via registering to the network owner. In packet pre-processing part, if a user enters to the network and needs to disseminate some data items, he/she ought to construct the information dissemination packets so as to send them to the nodes. In packet verification part, a node verifies every received packet. If the result is positive, it updates the information as per the received packet. It is the primary distributed knowledge discovery and dissemination protocol that permits network owners and approved users to check data items into WSNs while not relying on the base station. Moreover, our intensive analysis demonstrates that DiDrip satisfies the safety necessities of the protocols of its kind. Specifically, they apply the obvious security technique to formally prove the authenticity and integrity of the disseminated data items in DiDrip.

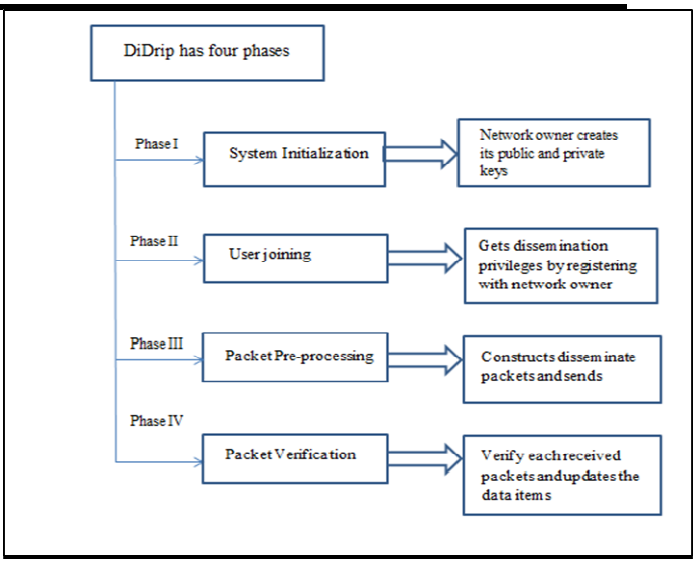

Fig.2: Phases of DiDrip

IV. DISCUSSION AND CONCLUSION

We conclude that, in this paper we projected a secure and distributed data discovery and dissemination protocol in the form of DiDrip. Table 2 presents a summary of merits of various protocols. Besides analyzing the protection of DiDrip, this paper has also through an analysis of DiDrip in an empirical network of resource-limited sensor nodes, showed that DiDrip is possible in practice. We have additionally presented a proper proof of the authenticity and integrity of the disseminated data items in DiDrip. In addition to this it can be concluded that due to the open nature of wireless channels, messages are simply intercepted.

Protocols	Merits
Drip	It avoids redundant transmission and achieves greater efficiency.
CodeDrip	Use of Network coding Increases throughput and decreases number of messages transmitted and thereby improves reliability and speed of dissemination.
DIP	It distinguishes difference of data in a node and identifies the different data items. The version number and keys for each data item is used.
DHV	This protocol tries to overcome the demerits of previous protocols like DRIP and DIP by reducing the problems involved in the updating of data and it uses bit slicing of version numbers in the network.
Typhoon	It uses a combination of spatially-tuned timers, prompt retransmissions, and frequency diversity to reduce contention and promote spatial re-use and also reduce dissemination time and energy consumption.

MNP	It provides reliability, low memory usage and fast data transfer.
DiDrip	ECC cryptography is used for key generation and to make it more secure Hash function is used.

REFERENCES

[1] Mohammad A. Matin, Wireless Sensor Networks: Technology and Protocols: Published by InTech, Croatia, ISBN 978-953-51-0735-4, 2012.

[2] Salvatore La Malfa, Wireless Sensor Networks, 2010.

[3] Jisha Mary Jose, Jomina John, "Data dissemination protocols in wireless sensor networks-a survey", IJARCCE, March 2014.

[4] Daojing He, Sammy Chan, Shaohua Tang and Mohsen Guizani, "Secure Data Discovery and Dissemination based on Hash Tree for Wireless Sensor Networks", IEEE transactions on wireless communications, Vol. 12, No. 9, September 2013.

[5] G. Tolle and D. Culler, "Design of an application cooperative management system for wireless sensor networks," in Proc. EWSN, pp. 121–132, 2005.

[6] Nildo dos Santos Ribeiro Junior, Marcos A. M. Vieira1, Luiz F. M. Vieira1 and Om Gnawali, "CodeDrip: Data Dissemination Protocol with Network Coding for Wireless Sensor Networks", In Proceedings of the 11th European conference on Wireless sensor networks (EWSN 2014), Feb. 2014.

[7] T. Ho and D. Lun. Network Coding: An Introduction. Cambridge University Press, 2008.

[8] Lin, K., Levis, P.: "Data discovery and dissemination with dip." In: Proceedings of the 2008 International Conference on Data Processing in Sensor Networks (IPSN 2008), Washington, DC, USA, IEEE Computer Society (2008) 433-444.

[9] P. Levis, N. Patel, D. Culler and S. Shenker, "Trickle: a self-regulating algorithm for code maintenance and propagation in wireless sensor networks", in Proc. 2004 NSDI, pp. 15-28.

[10] T. Dang, N. Bulusu, W. Feng, and S. Park, "DHV: a code consistency maintenance protocol for multihop wireless sensor networks", in Proc. 2009 EWSN, pp. 327-342.

[11] Liang, Chieh-Jan Mike, and Andreas Terzis. "Rethinking multi-channel protocols in wireless sensor networks." Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors. ACM, 2010.

[12] Kulkarni, Sandeep S., and Limin Wang. "MNP: Multihop network reprogramming service for sensor networks." Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on. IEEE, 2005.

[13] Hailun Tan, "Secure multi-hop network programming with multiple one-way key chains", In: Proceedings of the International conference on Embedded networked sensor systems (Sensys 07), Sydney, Australia, ACM.