Securing Disseminated Data in Wireless Sensor Networks using Reliable Schemes

Nagashree B. N.
PG Scholar
Dept. of CS&E, MCE
Hassan, India

Gururaj H. L.
Assistant Professor
Dept. of CS&E, MCE
Hassan, India

Ramesh B.
Professor
Dept. of CS&E, MCE
Hassan, India

ABSTRACT
Wireless sensor networks are autonomous sensors that are spatially distributed to supervise temperature, pressure, sound and other environmental conditions and to pass the data co-operatively to a main location through the network. The WSN is composed of several thousands of nodes, where each node is linked to one or several sensors. As wireless sensor networks continue to grow, the need for security mechanisms also does. After a WSN is deployed it is necessary to spread data through wireless links in order to update the configuration parameters and to distribute management commands to sensors. This is termed as data dissemination in WSNs. All presented data dissemination protocols experience from two drawbacks. First, they are based on centralized approach. Second, protocols were not designed with security in mind. Hence adversaries can easily launch attacks on the network.

General Terms
Security, Distributed approach, Elliptic Curve Cryptography.

Keywords
Dissemination, protocols, security, wireless sensor networks, efficiency.

1. INTRODUCTION
A wireless sensor network consists of vast number of self organizing nodes used for monitoring purposes which pass information obtained via network to base station. The motive for the development of WSNs is military applications. WSNs are also deployed in remote control and monitoring, healthcare management, environmental monitoring, inventory management etc.

WSN nodes are connected to one or more sensors. Each node of the sensor network consists of 3 subsystem i.e. sensor subsystem that senses the environment, processing subsystem that performs local computations on sensed data, and communication subsystem which is responsible for exchanging messages with neighbor sensor node. Each node has many components: microcontroller, radio transceiver, circuit for interfacing with sensors and battery. WSN topology is either star or multi-hop wireless mesh network.

WSNs operate for long periods of time and don't have human intervention. Considering the sensor nodes could be distributed in harsh environment, remotely disseminating data to sensor nodes through wireless channel is preferred than manual intervention. The propagation of new code over the network in required for wireless sensor network as manual updating is not possible. This process is called as dissemination. But effective dissemination of data to sensor nodes in network is a challenge. It is a difficult task as sensor network consists of vast number of nodes and the environment is dynamic in nature.

2. LITERATURE SURVEY
D.He et.Al [1] proposed Code dissemination protocol called Dicode. Code distribution is the method of propagating a fresh program copy or relevant commands to sensor nodes. After WSN is deployed in hostile environments, secure code dissemination continues to be a major concern. Most of them are based on centralized approach and only base station has the ability to initiate code dissemination. However, it is desirable and necessary to disseminate code images in a distributed manner. A salient feature of DiCode is its resistance to DoS attacks.

T.Dang et.Al [2] proposed a code consistency maintenance protocol called DHV. DHV tries to keep codes up to date and consistent on different nodes of WSN. Here data items are represented as tuples (key, version). DHV protocol overcomes the disadvantages of protocols like DRIP and DIP by reducing the complexity of updating data in the network. The basis is that if two versions are different, they may only differ in few least significant bits of their version number rather than in all of their bits. Hence, there is no need to transmit and compare entire version number in the network. The version number is given as a bit array.

Bit slicing is used to determine the out of date code. DHV uses two phases: detection and identification. In detection, every node will transmit a hash of all its versions in a SUMMARY message. A node compares it to its hash after receiving the above message. If they are not similar, there are one or more code items with a different version number.

In identification, horizontal search and vertical search steps are used to identify which versions differ. In horizontal search, a node transmits a checksum of all its versions in a HSUM message. Upon acceptance of the above message, a node compares the checksum to its own checksum to distinguish which bits are dissimilar and moves to the subsequently step. In vertical search, the nodes will broadcast a bit slice, earliest at the LSB of all versions, which a VBIT message. If the bit indices are identical, and the hashes are dissimilar, the node will transmit a bit slice of index 0 and enhance the bit index to find various locations until the hashes become same. After receiving a VBIT message, a node compares message to its own VBIT to identify the locations equivalent to the differing tuples. After identifying this, the node transmits those (key, version) tuples in a VECTOR message. Upon receiving a VECTOR message, a node compares it to its own (key, version) tuple to make a decision about who has the newer version and whether it should transmit its data. A node with a newer version will transmit its data to nodes with an older version.

K.Lin et.Al [3] proposed data discovery and dissemination protocol called as DIP. This protocol is based on the Trickle algorithm. It works in two parts: detects whether a difference

in data in nodes has occurred, and identifies which data item is different. The concept of version number and keys is used for each data item.

All nodes are up to date and have the same versions in steady state. Trickle is used to calculate and send hashes that cover all version numbers. Nodes that receive hashes same as their own knows that they have consistent data w.r.t their neighbours. If a node has a hash that differs from its own, a difference exists, and it does not know which data item has a newer version. Identifying the data item that is different and the node that has the newer version requires exchange of actual version numbers.

DIP maintains a soft state approximation of whether a given item differs from neighbour items or not. When a node receives packet and if the hashes are same, the estimate is decremented to 0. Otherwise the estimate is incremented. This continues until the estimates meets zero which means all have the similar data.

G.Tolle et.Al [4] introduced Sensor Network Management System (SNMS), which is an application co-operative management system for WSN and Drip is the dissemination protocol used in it. Drip is the simplest dissemination protocol and is based on Trickle algorithm. An independent trickle for each variable in the data is established. Each time an application wishes to broadcast a message, a new version number is generated and used. This causes the protocol to reset trickle timer and disseminate new value.

A standard message reception interface is provided by Drip in WSN. Each node that desires to use Drip will schedule with a specific identifier that depicts a dissemination channel. All inward messages on that channel will be delivered directly to the node. Each node is answerable for caching the data extracted from the current message received on each channel to which it subscribes, and recurring it in reaction to periodic rebroadcast requests. Drip achieves immense effectiveness by avoiding disused transmissions if the same information has previously been received by the nodes in the locality.

3. DIDRIP METHODOLOGY

DiDrip is the first secure and distributed data discovery and dissemination protocol. All previous protocols are based on centralized approach and no security was provided. These drawbacks are overcome by DiDrip. It allows network owners to authorize multiple network users to directly disseminate data items to the sensor nodes. It provides extensive security. It identifies security vulnerabilities in present data discovery and dissemination protocols, and disseminates data in distributive manner.

It uses ECC cryptography for key generation and Hash function for security. DiDrip is implemented by using Data hash chain and Merkle hash tree. It consists of 4 phases, system initialization, user joining, packet preprocessing and packet verification. In system initialization, public and private keys are created by network owner and public parameters are loaded on each node. In user joining, user registers to network owner for dissemination. In packet preprocessing, user enters the network and disseminates the constructed data items to the nodes. In packet verification, each received packet is verified by the node. Data is updated if the result of verification is positive.

3.1 System Architecture

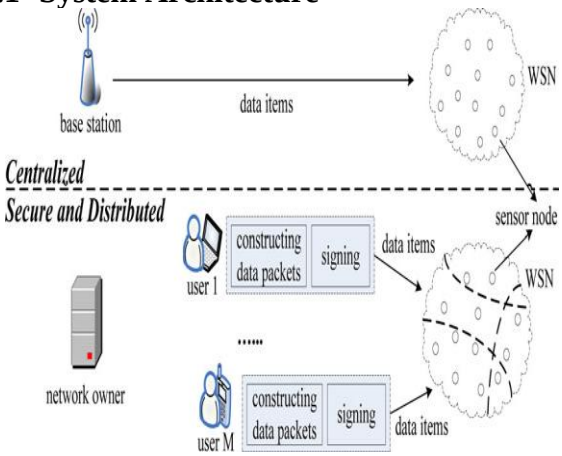

Figure 3.1 Centralized and Distributed approach

All existing data discovery and dissemination protocols employ centralized approach in which, as shown in top sub-figure in Figure 3.1, only base station disseminates data items.

3.1.1 Network Model

As shown in the figure 3.1, a general WSN consists of a large number of sensor nodes which are administered by the owner and accessible by many users. Sensor nodes are usually resource constrained and can perform limited number of cryptographic operations. Network users use mobile devices for dissemination and network owner is responsible for generating keying materials.

3.1.2 Trust Model

Network users are given dissemination privileges by trusted authority on behalf of network owner.

3.1.3 Threat Model

The adversary aims to corrupt the disseminated data. It may launch either external or insider attacks. In external attacks, adversary has no control of any sensor node in the network. The adversary can launch insider attacks by compromising network users or sensor nodes.

3.2 Simulation Design

Modules

1. System initialization phase
2. User joining phase
3. Packet pre-processing phase
4. Packet verification phase
5. Performance analysis

By referring to the lower sub-figure in figure 3.1, DiDrip consists of four phases, system initialization, user joining, packet pre-processing and packet verification phase. The information processing flow of DiDrip is illustrated in figure 3.2.

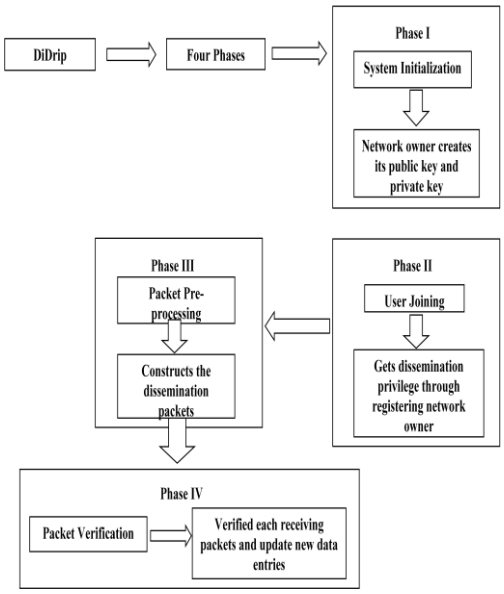

Figure 3.2 Information flow in DiDrip

1. System initialization phase
In this phase, network owner derives the public and private parameters. It then selects the private key and computes the public key. After that, the public parameters are preloaded in each node of the network.

2. User joining phase
This phase is invoked when a user with identity UID hopes to obtain dissemination privilege. User chooses the private key and computes public key. Then user sends UID to network owner, where Pri indicates the dissemination privilege of user. Upon receiving this message, network owner generates the certificate.

3. Packet pre-processing phase
User enters the WSN and wants to disseminate data items. For construction packets, data hash chain and merkle hash tree methods are used.

Before disseminating the data items, user signs the root node with private key and then transmits the packet consisting of user certificate. User disseminates each data item along with appropriate internal nodes for verification purpose. User certificate contains user identity information and dissemination privilege. Before network deployment, network owner assigns a predefined key to identify this packet.

4. Packet verification phase
When a sensor node receives a packet from an authorized user or from its one-hop neighbors, it first checks the packet key field. The data hash chain method incurs less communication overhead than merkle hash tree method. In data hash chain method, only one hash value of a packet is included in each packet. In merkle hash tree, D (tree depth) hash values are included in each packet.

5. Performance analysis
For the proposed system, following measurements are used to evaluate its performance.

1. Packet delivery ratio
2. End-to-End delay
3. Packet loss ratio

4. SIMULATION RESULTS

The performance of simulation is evaluated by using x graph. The evaluation metrics chosen are Packet received ratio, Packet loss ratio and End-to-End delay.

End-to-End Delay

It is the time taken for a packet to be transmitted across a network from source to destination. Figure 4.1 illustrates end-to-end delay of source node1 and source node2.

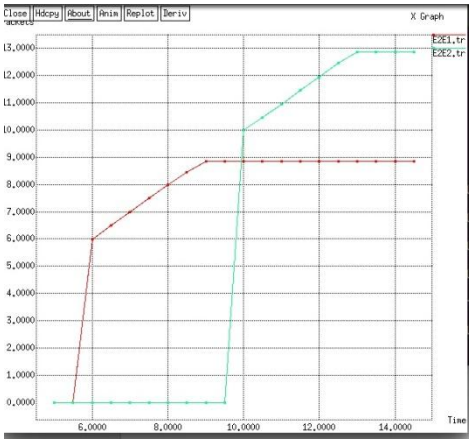

Figure 4.1 Xgraph of delay between two nodes

Packet Loss ratio

It is measured as percentage of packets lost with respect to the number of packets sent. Figure 4.2 illustrates the packet loss ratio of two source nodes.

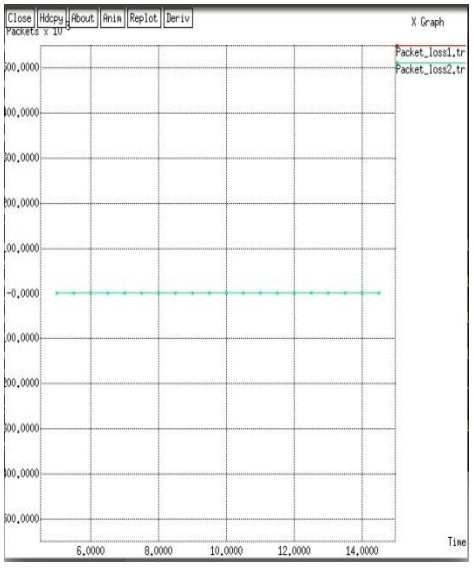

Figure 4.2 Xgraph of packet loss ratio

Packet Received ratio
It is defined as the ratio of number of packets received to the number of packets sent. Figure 4.3 illustrates the packet delivery ratio of two source nodes.

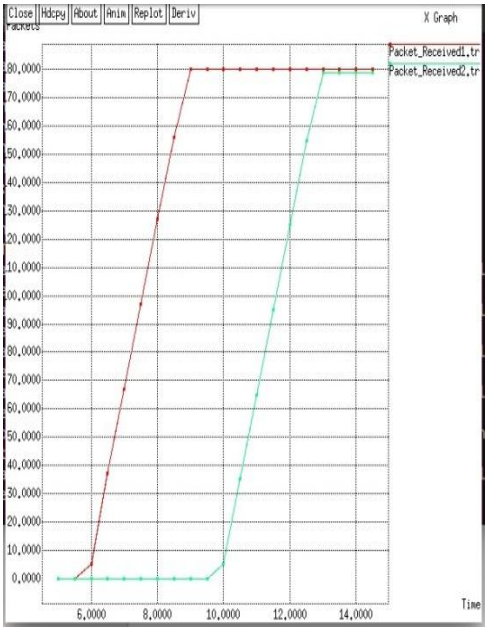

Figure 4.3 Xgraph of packet received ratio

5. CONCLUSION AND FUTURE WORK

Wireless Sensor Network is a wide area in networking research and is increasingly being used for monitoring applications. This demands the must for quickly and efficiently disseminating data and code to sensor nodes to reprogram them to match the current needs of the application. This is achieved by making use of data dissemination protocols. In this research paper, we have identified the security vulnerabilities while disseminating data in WSNs. All of the existing dissemination protocols are designed without security constraint and distributed operation was not supported. Therefore, in this paper, a secure and distributed data discovery and dissemination protocol named DiDrip has been proposed. Besides analyzing the security of DiDrip, the evaluation results show that DiDrip is feasible. Also a formal proof of authenticity and integrity of disseminated data items in DiDrip is given.

Thus, in the future work, we will consider how to disseminate multimedia data items using distributed approach.

6. ACKNOWLEDGMENTS

I would like to take this opportunity to thank one and all who have provided their valuable Advice, without their guidance this work would not have been a success. I have to thank many who have helped me directly or indirectly but some in particularly have to single out since they have given me more than just guidance.

I express my gratitude and sincere thanks to my guide **Mr. Gururaj H L,** Assistant Professor, Department of Computer Science & Engineering, Malnad College of Engineering, Hassan for his guidance, co-operation and encouragement.

I extend my thanks to **Dr. Ramesh B,** Professor and Head, Department of Computer Science & Engineering, Malnad College of Engineering, Hassan for his co-operation and encouragement throughout the course.

7. REFERENCES

[1] D. He, C. Chen, S. Chan and J. Bu, "DiCode: DoS-resistant and distributed code dissemination in wireless sensor networks," IEEE Trans. Wireless Communication., vol. 11, no. 5, pp. 1946-1956, May 2012.

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

[3] K. Lin and P. Levis, "Data Discovery and Dissemination with DIP," in Proc. ACM/IEEE IPSN, pp. 433-444, 2008.

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

[5] J.W. Hui and D. Culler, "The Dynamic behaviour of a Data Dissemination protocol for network programming at scale", in Proc. ACM SenSys, pp. 81-94, 2004.

[6] M. Ceriotti et al., "Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment," in Proc. IEEE IPSN, pp. 277- 288, 2009.

[7] D. He, S. Chan, S. Tang, and M. Guizani, "Secure data discovery and dissemination based on hash tree for wireless sensor networks," IEEE Trans. Wireless Communication., vol. 12, no. 9, pp. 4638-4646, Sept. 2013.

[8] S. Rahman, N. Nasser, T. Taleb, "Secure timing synchronization for heterogeneous sensor network using pairing over elliptic curve," Wireless Communications and Mobile Computing, vol. 10, no. 5, pp. 662-671, May 2010.

[9] M. Rahman, N. Nasser, and T. Taleb, "Pairing-based secure timing synchronization for heterogeneous sensor networks," in Proc. IEEE GLOBECOM, pp. 1-5, 2008.

[10] Geoss. http://www.epa.gov/geoss/.

[11] NOPP, http://www.nopp.org/.

[12] ORION,http://www.joiscience.org/ocean observing/advisors.