Small Data Dissipation Using Se-Drip – An Enhanced Version of Drip

Ms. V. Savitha M.E., Mr. E. U. Iniyan M.E., Ms. M. D. Dafny Lydia
Electronics and Communication Engineering, Prathyusha Institute of Technology and Management
Aranvoyalkuppam, Thiruwallur, Tamil Nadu, India

Abstract—Wireless Sensor networks are a new class of distributed system that is an integral part of the physical space they inhabit. Unlike most computers which work primarily with data created by humans, sensor networks reason about the state of world that embodies them. The network consists of numerous sensor nodes with sensing, wireless communications and computing capabilities. These sensor nodes require time to time update of variables by a process called dissemination. This is done using dissemination protocols like Drip, DIP (Dissemination Protocol) and DHV. These protocols helps improve reliability and efficiency but do not consider security of transmitted data. A special protocol called Se-Drip is proposed to ensure security during data dissemination. It is an enhanced version of Drip protocol. Here security is provided using Merkle hash tree which forms the hash values for the transmitted data. To enhance and complicate the security process an 8 byte number called the salt number is added. This provides better security and authentication.

Keywords—Dissemination, trickle algorithm, merkle hash tree, salt number, wireless sensor network

I. INTRODUCTION

Wireless sensor networks are used widely in monitoring and control of environmental parameters. Several approaches have been proposed for data discovery and dissipation, but they all focus on ensuring reliability and overlook the security vulnerabilities. Se-Drip is a secure, DoS resistant data discovery and dissipation protocol used in WSNs. It also provides authentication for packet delivery by providing hash values to every nodes in the network. The process of dissipation includes the capability to send messages from a control server to the other nodes in the network. The process ensures authenticity and integrity in data discovery and dissemination.

The dissemination protocols may be Code dissemination and Data dissemination protocols. Code dissemination protocols distribute long messages into the network while data dissemination protocols distribute short messages into the WSN, such as two byte parameters. The use of these protocols includes injection of programs, queries, configuration parameters and commands into the network.

The main contributions of this paper include:
1) Investigating the security issues in data discovery and dissipation procedure and analyzing the lack of authentication of data in WSNs.

2) Developing a secure, DoS resistant data discovery and dissipation protocol named Se-Drip, where the received packets are immediately verified based on a signed Merkle Hash tree. By this the base station signs only the root of the Merkle has tree. To further increase the security and to enhance efficiency, salt number is added.

II. SECURITY VULNERABILITIES IN DATA DISCOVERY AND DISSIPATION

A. Process of data discovery and dissipation

Trickle algorithm forms the basics of data discovery and dissipation. The existing protocols of Drip and DIP use this algorithm. According to this algorithm, a node in the network broadcasts periodically a summary of data it has unless it receives the same summary of data from other nodes. Once the data in all nodes are consistent, the broadcast interval increases exponentially. If, on the other hand, a node receives a new data, it reports quickly. Thus when a new data is injected by the base station to a node, it is disseminated immediately using Trickle algorithm.

In the existing protocols of Drip, DIP and DHV, each data item is represented by 3-tuple (key, version, data*), where key indicates the variable to be updated, version indicates if the data item is old or new (larger the version, newer the data) and data* indicates the dissemination data.

B. Security vulnerabilities

The attacks in WSNs might be external or internal attacks. External attacks include eavesdropping of information, replaying messages, injecting forged messages and impersonating a valid node. Internal attacks can occur when the nodes in the network are compromised to attack the other nodes in the network. They are referred to as insiders as they are still part of the network. They remain to be a member of the network until they are identified and excluded from the network. Forging of messages is an important criteria followed by internal attack. By forging the messages the adversary can reboot the entire network or they can also eliminate important variables from the nodes within the network. Rebooting of network includes the conversion of data into data* (wrong message). The next process involves the injection of data* into the network while changing the tuple as (key, version, data*). This is accomplished as the version number is larger than all the other version numbers. Thus the nodes are updated to data* due to higher version number. Similarly the entire data can be erased by simply sending a fake data item (key, version, 0) into the WSN, where the version number can be made as large as possible.

C. Assumptions

The following assumptions can be taken into consideration:
1) The base station acts as the origin of legitimate data updates hence it cannot be compromised. It is a source of dissipated data and so it is trustworthy.
2) Base station has unlimited computational power compared to any other sensor nodes.
3) The sensor nodes have limited resources which are constrained in memory space, bandwidth and energy, and so it can perform limited number of public key cryptography during its lifetime.

III. SE-DRIIP – SECURED VERSION OF DRIP PROTOCOL

A. Se-Drip- an overview

Se-Drip combines ECC (Elliptic Curve Cryptography) and Merkle hash tree. Thus it inherits robustness to packet loss using Trickle algorithm where there is periodic delivery of data to every node in the network.

Se-Drip mainly consists of three phases: system initialization phase, packet preprocessing phase and packet verification phase. The system initialization phase is done before the network distribution. Here the base station creates the public and private key. The public key is loaded into the nodes in the network. Before dissipation of data into the network, packet preprocessing phase is executed. In packet preprocessing phase, merkle hash tree is constructed from the transmitted data items. In the packet verification phase, the received packets in each node are verified. If the result is positive then the data is updated according to the received packet. Each phase is described in detail in the following and the symbols used are listed in Table 1.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_i$</td>
<td>Nodes</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Hash value of message</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Packets</td>
</tr>
<tr>
<td>$d_i$</td>
<td>Data items</td>
</tr>
<tr>
<td>SIG$_{SK}$</td>
<td>Signature of root</td>
</tr>
</tbody>
</table>

TABLE I

SYMBOLS

B. System initialization phase

Base station creates the public and private keys. The public key is loaded into each node. Drip protocol uses 3 tuple (key, version, data), whereas Se-Drip uses 4 tuple (round, key version, data) since it is an enhanced version of Drip. Here round refers to round to which the data items belongs.

C. Packet preprocessing phase

System initialization phase is followed by the packet preprocessing phase. The base station disseminates n data items, $d_i = \{\text{round}, \text{key}, \text{version}, \text{data} \}$, where $i = 1,2,...,n$. Fig. 1 shows Merkle hash tree. Merkle hash tree is used to construct the packets of the respective data items. The leaves in the tree are hashes of authentic packets $P_i$ , where $i = 1,2,...,n$. Hash function is calculated over the packet header, data item $d_i = \{\text{round}, \text{key}, \text{version}, \text{data} \}$. The base station computes $e_i = H (P_i)$, where $i = 1,2,...,n$. The hash tree is computed from the internal children nodes. Each internal node is the hash value of the two children nodes. In Fig. 1, $e_{1,2} = H (e_1 || e_2), e_{1,4} = H (e_1 || e_4)$ and $e_{5,6} = H (e_5 || e_6)$. Similarly $e_{1,6} = H (e_{1,2} || e_{3,4} || e_{5,6})$. The base station signs the root with the private key forming a signature packet $P_0$ which has the root of the Merkle hash tree and the signature value. A predefined key is assigned to identify the signature packet. The packet $P_0$ consists of the packet header, data item $d_0 = \{\text{round}, \text{key}, \text{version}, \text{root} \}$ and the signature $SIG_{SK} (H (d_0))$. The base station broadcasts the signature packet first and the root allows each node to verify each received packet $P_n$ using the authentication path included in the packet.

For example, in Fig. 1, if $e_{1,4}$ has been authenticated using signature packet. On receiving a packet consisting of $d_1, e_2, e_{1,4}$ and $e_{5,6}$, verification can be done by analyzing if $H (H (H (d_1) || e_2) || e_{1,4}) || e_{5,6}) = e_{1,6}$. Thus Se-Drip enables authenticity and integrity even when packets are received out of sequence.

D. Packet verification phase

On packet reception, the key is checked. If the value is new, the node replaces the value, else it is discarded.

IV. SECURITY IMPROVEMENT

A. Adding Salt number

The process of continuous authentication in Merkle hash tree leads to overhead. In order to reduce the transmission overhead a random 8 byte number called the salt number is used. This salt number is added to the data $d_0$ of the signature packet, that is, $d_0 = \{\text{round}, \text{key}, \text{version}, \text{root}, \text{salt number} \}$. Thus $d_0 = \{\text{round}, \text{key}, \text{version}, \text{root}, \text{salt number} \}$ will be the input to the signature generating operation. Also the base station uses salt number as part of its input in the hash value generation, that is, $e_i = H (P_i || \text{salt})$. The node ensures that the signature packet $P_0$ is authentic and updates < round, salt, root > by the corresponding values of $P_0$ which is then picked up according to the round, and then uses the root and salt number to verify the packet by performing hash operation.

Fig. 1 Example of Merkle hash tree.
Fig. 2 Graphical representation of throughput, delivery ratio and delay using Se-Drip protocol

V. CONCLUSION

Thus the security vulnerabilities in data discovery and dissemination of wireless sensor networks is identified and a protocol named Se-Drip is developed to provide secure and efficient authentication of the disseminated data items using Merkle hash tree and Salt number.

VI. FUTURE SCOPE

The process of data dissemination in WSN requires the need of hash generation and authentication for individual packet hence this might not be able to be used for medical purposes as it increases delay. Thus a protocol can be designed for secure transmission of packets in WBSNs.

VII. REFERENCES