The Design and Challenges of Online Reprogramming System for Wireless Sensor Networks

Mande Xie¹, Guoping Zhang²
¹College of Computer Science & Information Engineering
Zhejiang Gongshang University, Hangzhou
Zhejiang, People’s Republic China, 310018
²Faculty of Informatics & Electronics
Zhejiang Sci-Tech University
Hangzhou, Zhejiang, 310018, China
xiemd@mail.zjgsu.edu.cn, zgp4508@mail.zstu.edu.cn

ABSTRACT: At present, the online network reprogramming faces two challenges: 1) How to save the energy consumed during the code dissemination; and 2) How to enhance the security of online network reprogramming. In this paper, we modify the work flow of online reprogramming system and separates an alone reprogramming component from the whole program image which need not be transmitted in order to reduce the energy consumption. According to the security vulnerabilities in the image management aspects of code dissemination, a cipher puzzle-based image management algorithm is proposed. According to the based-request attacks, several counter measures are introduced in our online reprogramming system, such as secure broadcast authentication based on cluster keys and the request count technique. By integrating several key techniques, our online reprogramming system can securely and energy-efficiently disseminate the new program image into the whole WSNs.

Categories and Subject Descriptors

General Terms
Wireless Sensor Networks, Online Programming

Keywords: Online Network Reprogramming, Wireless Sensor Networks, Hash Key Chain, Authentication

Received: 10 June 2011, Revised 19 August 2011, Accepted 24 August 2011

1. Introduction

Due to the potential to provide fine-grained sensing and actuation at a reasonable cost, wireless sensor networks (WSNs) are considered ideal candidates for a wide range of applications, such as industry monitoring and military operations. Program image (PI) updates have become very necessary in WSNs because they may be required to fix bug or to provide new functionalities after WSNs have been deployed. However, if WSNs are large scale or deployed in the harsh environment, it is impossible to manually reprogram all nodes. The online network reprogramming can remotely reprogram all nodes via wireless communication. Hence it becomes a promising technique.

For large WSNs where the sink cannot reach every node through broadcasting, new PI can only be transmitted hop-by-hop within the WSNs, and significant energy is consumed. Recent studies have shown that sending a single bit of data consumes about the same energy as executing 1000 instructions. However, sensor nodes are usually powered by the battery and have the limited energy. As a result, it is essential to conserve the energy in a WSNs during the code and data dissemination, especially when the update happens frequently. In present, lots of researchers have paid much attention on the reprogramming algorithms to minimize the energy consumption. Their main approaches are to optimize the transmitted object.

Almost of the existing online reprogramming protocols are an epidemic protocol and the epidemic behavior can provide resilience to random process and allows rapid code dissemination through purely local interactions in large scale, dynamic environment. However, they do not take account of any security issues. Furthermore, WSNs are generally deployed in hostile and unattended environments for long periods of time. As a result, they are possible to face various secure threats and are vulnerable to lots of attacks. For example, an adversary may employ the suppression mechanism and the dynamic adjustments mechanism of the broadcast rate to prevent the propagation of code update, waste network resources, introduce unnecessary latency or disrupt the normal operation of code dissemination.

Hence, the online reprogramming faces two challenges: 1) How to save the energy consumed during the code dissemination; and 2) How to enhance the security of online reprogramming. In this paper, according to these two challenges, several solutions are presented. The rest of this paper is structured as follows: Section 2 introduces the related work. Section3 gives out the framework of the
online reprogramming system. In Section 4, The secure image management method is described. The energy reserved strategy in the online reprogramming system is proposed in Section 5. Section 6 describes the security strategies against the based-request attacks in PI update. At last, the brief summary is drawn in Section 7.

2. Related Works

2.1 Review On Network Reprogramming With Energy Consideration

The early network reprogramming protocol includes XNP [1], MOAP [2]. XNP only operates over a single hop and does not provide incremental updates of the code image. The Multi-hop Over the Air Programming (MOAP) protocol extends it to operate over multiple hops [2]. MOAP introduces several concepts which are used by later protocols. However, it does not leverage the pipelining effect with segments of the code image. Another three protocols that are substantially more sophisticated than the rest are Deluge [3], MNP [4] and Freshet [5]. All use the three way handshake for locally propagating the code. Deluge [3] is the earliest and lays down some design principles used by the other two. It builds on top of Trickle [6] which is a protocol for a node to determine when to propagate code in a one hop case. The design goal of MNP [4] is to choose a local source of the code which can satisfy the maximum number of nodes. They save energy by turning off the radio of non-sender nodes. Freshet [5] aggressively optimizes the energy consumption for reprogramming. The Stream [7] uses the facility of having multiple code images on a node and switches between them, Stream pre-installs the reprogramming protocol as one image and the application program equipped with the ability to listen to new code updates as the second image. Rajesh et al. propose two incremental reprogramming algorithms Zephyr [8] and Hermes [9] to minimize the energy consumption. Zephyr transfers the delta between the old and the new software and reduces the delta size by using application level modifications to mitigate the effects of function shifts. At the same time, it compares the binary images at the byte-level with a novel method to create small delta. Based on Zephyr, Hermes reduces the delta by using techniques to mitigate the effects of function and global variable shifts caused by the software modifications. According to mobile sensor network, Pradip et al. [10] propose an energy-efficient, multihop reprogramming protocol to consider the prohibitive factor of uncertainty about a nodes location due to their continuous movement.

2.2 Existing work on secure Network Reprogramming

Deluge currently distributed as part of TinyOS, is the de facto standard network reprogramming for WSNs. However, it does not take account of any security issues and is vulnerable to lots of attacks. Hence, based the framework of Deluge, lots of secure network reprogramming algorithms are proposed. The lecture [11] [12] [13] [14] propose some hash chain-based schemes. Sluice [11] integrates signature and cryptographic hash functions to provide efficient authentication for network reprogramming. Sluice builds the hash chain in page level and only perform authentication when an entire page is received. As a result, it cannot authenticate a packet immediately after the packet is received. Different to Sluice, Dutta et al. [12] builds a hash chain in packet level to authenticate a packet immediately after it is received. However, this approach requires the packets are received in order. In fact, it is normal that a packet reached the target node out of node in WSNs. Deng et al. [13] [14] builds a hash chain in page level and then builds a Merkle hash tree for each page. This approach can authenticate a packet immediately after it is received and also allow the packet is received out of order. But, it introduces an excessive traffic for the Merkle hash trees. According to the Deny of Service (DOS) attacks on Deluge, the lectures [15-19] propose several defense methods to mitigate the DoS attacks. Park et al. [15] propose two schemes in the way of providing the recovery method for verification processes at packet loss, using supplementary hashing: redundant hash scheme and page digest scheme. Dong et al. [16] presents two filtering techniques, a group-based filter and a key chain-based filter, to handle DoS attacks against signature verification. Tan et al. [17] firstly integrate confidentiality and DoS-attack-resistance in a multi-hop code dissemination protocol. At the same time, they propose countermeasures against both types of DoS attacks based on requests. Hyun and Ning et al. [18] [19] propose DoS-resistant network reprogramming system named Seluge. Seluge can provide immediate authentication of each packet upon receipt, without disrupting the efficient propagation mechanisms used by Deluge. Pradip et al. [20] propose an epidemic theoretic framework for vulnerability analysis of network reprogramming protocols in WSNs and this framework provides a general analysis for several protocol. According to the vulnerability of “reboot” and “erase” command, Liu et al. [21] propose a hash chain-based scheme to provide the security of image management.

3. The Framework of Online Reprogramming System

Inspired by Stream[7], our online reprogramming system implements the network reprogramming protocol as an alone image and pre-installs it in all sensor nodes. The application program equipped with the ability to listen to new code updates is implemented as another image. As shown in Figure 1, the framework is divided three steps:
1) Reboot to the pre-installed protocol image; 2) Disseminate the new PI (program image); 3) Reboot to the new PI. Before the BS (base station) disseminates the new PI, it must firstly make all of node reboot to the protocol image. If the reboot command is correctly authenticated, the nodes will accept the reboot command and reboot to the protocol image. During the new PI update phase, the BS firstly divides the new PI into a series of pages and each page is further divided into a series of fixed size packets. Next, the BS computes the hash value of each packet and appends it at the end of the corresponding packet. After the above operations finish,
4. The Secure Image Management Method

In present, almost all of secure online reprogramming algorithms focus on securing the propagation of code images and overlook the security vulnerabilities in the image management aspects of code dissemination. Almost no paper on secure image management is published except the lecture [21]. However, according to the image management algorithm based on key chain proposed by Liu et al [21], an attacker can directly manipulate the version number $rVerNum$ to launch an attack. The intuitional countermeasures is to include the $rVerNum$ in the digital signature packet. However, unlike the inject command, a code image may be often rebooted. Hence, digital signature is not suitable because of its high complexity. A new image management algorithm based on a weak authentication mechanism called Cipher Puzzles (CP) is proposed to reduce the computation load of the algorithm in this paper.

The Figure 2 shows the principle of our algorithm. Like [21], our method must also build a hash key chain for each image by the method proposed by [21]. The first element $K_{i,0}$ of each hash key chain is preloaded in all nodes. Let $K_{ij}$ indicate the jth reboot key of code image $i^{th}$ and we also call it puzzle key. For each reboot $j$ of code image $i$, we use the puzzle key $K_{ij}$ to generate a puzzle. We first encrypt the message $rVerNum | K_{ij} | F_i | P_i$ by a symmetric encryption algorithm whose key is the last reboot key. Where, $rVerNum$ is the version number, $K_{ij}$ is the current reboot key, $F_i$ is the other required fields.

Upon receiving the authentication packet, each node first decrypts it. If the decryption is successful, $K_{ij}$ is extracted from the decrypted message. Then the receiving nodes verifies whether the key $K_{ij}$ is valid by the formula $K_{ij-1} = H(K_{ij})$, where $K_{ij-1}$ is the last reboot key. If this verification is successful, the node further verifies the puzzle solution by the method that the first $k$ bits of authentication packet is the same with those of $H(rVerNum)$. If this verification is also successful, the $rVerNum$ must not be modified and we can determine the code image is new or old by it.

5. Energy Reserved Strategy in the Online Reprogramming System

Unlike Deluge, our online reprogramming system separates an alone reprogramming component from the whole program image. That is to say, we implements the reprogramming protocol as an alone program image, denoted $IMAGE_P$, and implement the application as another program image, denoted $IMAGE_A$. Besides the normal application function, $IMAGE_A$ is also equipped with the ability to listen to new code updates. Because the $IMAGE_P$ is simple and invariable, in general, $IMAGE_P$ need not be updated. $IMAGE_A$ may to be updated if the application function is changed or some bugs are found. However, in general, the difference between the modified program image and the old one is small. $EEORP$ transmits the difference instead of the whole $IMAGE_A$. Figure 1 shows the work flow of the reprogramming protocol. The source node first generates the delta script, which is composed of the INSERT commands and COPY commands, by byte level comparison tool [22] between modified version and old version. Then, the delta script is...
disseminated by the WSNs. Upon receiving the delta script, a sensor node stores it and rebuilds the new application with the old application and delta script (as shown in Figure 3 and Figure 4). In addition, any sensor node divides the external flash into several slots and the Figure 4 shows the layout of external flash in the nodes, where $\text{IMAGE}_P$ indicates the reprogramming component, $\text{IMAGE}_A$ indicates the old application version, $\text{IMAGE}_D$ indicates the delta script and $\text{IMAGE}_N$ indicates the new application version. After a node has rebuilt the new application version and saved it, it loads the new application to program memory to run by the boot loader.

Figure 3. The work flow of the online reprogramming system

6. Security Strategies Against The Based-Request Attacks In PI Update

In general, online reprogramming employs the three-stages propagation method which includes advertise, request and updates. During request stage, there are lots of attack types. According to the based-request attacks, several countermeasures are introduced. Firstly, cluster keys are used for local broadcast authentication which is proposed in [13]. Each node in WSNs generates a per-node cluster key and authenticates all the advertisement and request packets transmitted from itself by it. When a node is deployed, it notifies its neighbors through periodic hello packets. Upon receiving a hello packet from a new neighbor, after a random delay, each node replies with its cluster key to the sender encrypted using their pairwise key which can be established by the existing method [23][24]. Moreover, a node that just sends a cluster key to a new neighbor also broadcasts a hello packet so that the new neighbor can reply with its own cluster key. For each incoming request packet, a node uses the sender’s cluster key to verify its integrity. The node simply discards unauthenticated or duplicate request packets.

Secondly, the sequence number denoted $SN$ is introduced. Denote the requester and the sender $A$ and $B$, respectively. $A$ firstly sends a request packet with $SN_A = 0$ to $B$. Next, $A$ sends each request packets with $SN_B = SN_B + 1$ to $B$. Upon receiving a request packet from $A$, if the packet is the first request packet from $A$, $B$ sets the $SN_A = 0$. Otherwise, $B$ compares the $SN_B$ with $SN_A$ and if $SN_B > SN_A$, $A$ accepts the request and set $SN_A = SN_A + 1$. If $SN_B \leq SN_A$, $A$ denies the request.

At last, a counter denoted as $C_{\text{req}}$ is introduced for each node. If $C_{\text{req}}$ is less than a given threshold value $R_{\max}$ and the request is accepted and set the $C_{\text{req}} = C_{\text{req}} + 1$, otherwise, the request is denied. By the introduction of $C_{\text{req}}$, the malicious repeat requests can be avoided.

7. Conclusion

The two challenges of saving the energy consumed during the code dissemination and enhancing the security of online reprogramming is addressed in the current study. In this paper, according to these two challenges, several solutions are introduced. In order to enhance the security of online reprogramming, two methods are introduced. Now, almost all of secure online reprogramming algorithms focus on securing the propagation of code images and overlook the security vulnerabilities in the image management aspects of code dissemination, such as rebooting and erasing code images. In accordance with the situation, we presented a cipher puzzle-based image management algorithm based on the method proposed by [21]. The cipher puzzle-based authentication method is valid and efficient, because it only introduces a few hash function operations and comparisons to authenticate the reboot command and the version number $r\text{VerNum}$. According to the based-request attacks, several countermeasures are introduced in our online reprogramming system, such as secure broadcast authentication based on cluster keys and the request count technique. In additional, in order to save the energy consumed during the code dissemination, we modify the work flow of online reprogramming system and employ the energy reserved strategy by splitting program image.
In our online reprogramming system, we separates an alone reprogramming component from the whole program image. That is to say, we implement the reprogramming protocol as an alone program image and implement the application as another program image. Only the latter is transmitted during the code dissemination.

8. Acknowledgements

This work was partially supported by Grant No Y1101316 from the National Natural Science Foundation of Zhejiang Province and Science and Technology Department of Zhejiang Province Program.

References
