Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks Using Seluge (Demo Abstract)

An Liu, Young-Hyun Oh, Peng Ning
Cyber Defense Laboratory
Department of Computer Science
North Carolina State University

1 Introduction

A wireless sensor network is expected to consist of a potentially large number of low-cost, low-power, and multifunctional sensor nodes that communicate over short distances through wireless links. Due to their potential to provide fine-grained sensing and actuation at a reasonable cost, wireless sensor networks are considered ideal candidates for a wide range of applications, such as industry monitoring, data acquisition in hazardous environments, and military operations.

It is desirable and sometimes necessary to reprogram sensor nodes through wireless links after they are deployed, due to, for example, the need of removing bugs and adding new functionalities. The process of propagating a new code image to the nodes in a network is commonly referred to as *code dissemination*. A few code dissemination protocols (e.g., [2, 5, 7, 10, 12, 13]) have been developed recently to propagate new code images using the ad-hoc wireless network formed by the sensor nodes. In particular, Deluge [5] uses an epidemic protocol [9] for efficient advertisement of code meta data and spatial multiplexing for efficient propagation of code images. Deluge is generally accepted as the state of the art for code dissemination in wireless sensor networks, and has been included in recent TinyOS distributions [1].

In hostile environments, where there may be malicious attacks against wireless sensor networks, code dissemination faces threats from both external attackers and potentially compromised nodes. For example, the adversary may attempt to modify or replace the real code image being propagated to sensor nodes, introducing malicious code into the sensor network. As another example, the adversary may inject bogus code dissemination packets and force normal sensor nodes to verify and/or forward them, thus exhausting their limited battery power.

Several recent works have attempted to provide secure code dissemination for wireless sensor networks [3, 4, 8], which are all extensions to Deluge [5]. Unfortunately, all these approaches have various weaknesses or limitations [6], such as the vulnerability to denial-of-service (DoS) attacks and inefficiency in code dissemination.

2 Seluge

We have developed an efficient system named Seluge that supports secure and DoS-resistant code dissemination in wireless sensor networks [6]. Seluge is an extension to Deluge [5], an open source code dissemination system included in TinyOS [1]. It inherits the efficiency and robustness properties from Deluge, and at the same time provides security protections for code dissemination, including the integrity protection of code images and resistance to the following three classes of DoS attacks: (1) DoS attacks against signature packets; (2) DoS attacks against code dissemination packets; and (3) DoS attacks against maintenance packets. To the best of our knowledge, these are all the DoS attacks that manipulate code dissemination protocols.

The key contribution of Seluge is a novel way to organize the packets used to distribute new code images. By carefully arranging code dissemination data items and their hash images in packets, Seluge provides *immediate authentication* of each packet when it is received, without disrupting the efficient propagation mechanisms used by Deluge. Thus, it can defeat DoS attacks against dissemination packets, which have to exploit authentication delays.

Seluge properly authenticates advertisement and SNACK packets. As a result, it can prevent DoS attacks exploiting the Deluge epidemic propagation and suppression mechanisms.

Seluge uses a signature to bootstrap the authentication of a new code image. However, unlike the previous attempts, Seluge uses a weak authentication mechanism called *message specific puzzle* along with the signature [11]. Message specific puzzle has some nice properties: It can be efficiently verified by a regular sensor node, but it takes a computationally powerful attacker a substantial amount of time to forge a weak authenticator. Moreover, it cannot be pre-computed. Thus, message specific puzzle provides an effective filter of forged signatures. As a result, Seluge is not subject to the same DoS attacks against signature verifications as the previous approaches.

Compared with the previous attempts [3,4,8], Seluge not only provides integrity protection for code images, but is also resistant to various DoS attacks exploiting the expensive signature verifications, authentication delays, and the epidemic propagation strategies in Deluge. Indeed, Seluge is superior to all the previous solutions [3,4,8], and is the only solution that seamlessly integrates the security mechanisms and the efficient Deluge propagation strategies.

3 Demo Plan

In this demo, we will show how Seluge works in wireless sensor networks. Our demo will consist of two parts: (1) demo in an on-site small-scale network, and (2) demo in a remote test-bed.

3.1 Demo in an On-Site Network

We will bring 20-30 MicaZ motes physically to the demo site to form an on-site, small-scale wireless sensor network. Given the space constraints (i.e., each demo only has a table), we will hard code the neighbor relationships in these motes, so that they will behave as a multi-hop network. During the demo, we will reprogram the motes with different programs and take performance measurements on the fly. We will also display selected program states of selected motes to show the dynamics of remote programming.

A limitation of this demo is that there will be interference between the code dissemination packages since the neighbor relationships are simulated. Thus, the performance results obtained in this demo is not exactly what one would see in a real deployment. We address this problem by having the second demo component, as we explain in the following.

3.2 Remote Demo in WiSeNeT Testbed

We have a testbed for wireless sensor networks named WiSeNeT (Wireless Sensor Network Testbed). WiSeNeT consists of over 120 MicaZ motes, deployed on the second and the third floors in the east wing of Engineering Building II at NC State University. Each mote is connected to an Ethernet programming board, which allows remote access to the mote for experimentation purposes. The entire WiSeNeT covers over 30,000 square feet of space.

In the remote demo, we will repeat the same set of experiments in WiSeNeT through remote connection to hosts at NC State University. These experiments will provide features of Seluge in a realistic setting. However, the audience will not be able to observe the motes directly.

References

[1] TinyOS: An open-source OS for the networked sensor regime. `http://www.tinyos.net/`.

[2] Crossbow Technology Inc. Mote in-network programming user reference, 2003.

[3] J. Deng, R. Han, and S. Mishra. Secure code distribution in dynamically programmable wireless sensor networks. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN '06)*, April 2006.

[4] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler. Securing the deluge network programming system. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN '06)*, April 2006.

[5] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, November 2004.

[6] S. Hyun, P. Ning, A. Liu, and W. Du. Seluge: Secure and dos-resistant code dissemination in wireless sensor networks. In *Proceedings of the Seventh International Conference on Information Processing in Sensor Networks (IPSN '08)*, April 2008.

[7] S. Kulkarni and L. Wang. MNP: multihop network reprogramming service for sensor networks. In *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS '05)*, pages 7–16, June 2005.

[8] P. Lanigan, R. Gandhi, and P. Narasimhan. Sluice: Secure dissemination of code updates in sensor networks. In *Proceedings of the 26th International Conference on Distributed Computing Systems (ICDCS '06)*, July 2006.

[9] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the 1st Symposium on Network System Design and Implementation (NSDI '04)*, March 2004.

[10] V. Naik, A. Arora, P. Sinha, and H. Zhang. Sprinkler: A reliable and scalable data dissemination service for wireless embedded devices. In *Proceedings IEEE International Real-Time Systems Symposium*, pages 277–286, December 2005.

[11] P. Ning, A. Liu, and W. Du. Mitigating DoS attacks against broadcast authentication in wireless sensor networks. *ACM Transactions on Sensor Networks*, 4(1), February 2008. To appear.

[12] N. Reijers and K. Langendoen. Efficient code distribution in wireless sensor networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, pages 60–67, September 2003.

[13] T. Stathopoulos, J. Heidemann, and D. Estrin. A remote code update mechanism for wireless sensor networks. Technical Report CENS-TR-30, UCLA, Center for Embedded Networked Computing, November 2003.