Computer Networks 141 (2018) 44–56

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Designing self-destructing wireless sensors with security and performance assurance

Yu Li[a], Xiaotian Wang[a], Dae Wook Kim[c], Junjie Zhang[a], Rui Dai[b,*]

[a] Department of Computer Science and Engineering, Wright State University, Dayton, OH 45435, USA
[b] Department of Electrical Engineering and Computing Systems, University of Cincinnati, Cincinnati, OH 45221, USA
[c] Department of Computer Science, Eastem Kentucky University, USA

ARTICLE INFO

Article history:
Received 4 June 2017
Revised 24 March 2018
Accepted 16 May 2018
Available online 17 May 2018

Keywords:
Wireless sensors
Self-destructing
Assurance
Model checking

ABSTRACT

A lost wireless sensor may lead to the leakage of sensitive information. This paper proposes a framework to facilitate the design of self-destructing wireless sensors with assured security and performance properties. This framework includes a cryptographic self-destructing mechanism that enables autonomous self-destruction of a wireless sensor and a caching strategy to reduce the performance overhead that is consequently introduced. Based on Discrete-Time Markov Chains (DTMC), we have designed models to characterize the proposed self-destructing mechanism, the caching strategy, various components in a sensor, the attacker, and the interactions among all these elements. We have also defined security and performance properties in the form of probabilistic computation tree logic (PCTL), which could be rigorously verified using probabilistic model checking. This framework offers unique capabilities on performing quantitative analysis on security and performance of wireless sensors.

1. Introduction

Wireless networked sensors play an essential role in revolutionizing our world in numerous aspects such as defense, transportation, energy, and healthcare through various applications such as cyber-physical systems, the Internet of Things (IoT), and body sensor networks. Compared to traditional computation systems such as desktops, wireless sensors are more likely to be attached to physical entities, thereby experiencing a much higher probability to be lost. If a sensor is lost and then collected by an attacker, great security concerns will arise: the attacker can break into the system (e.g., by brute-force login accesses, exploiting certain vulnerabilities, or physically disassembling the system) to collect various sensitive information such as program instructions and data from both memory [1] and disk. A body sensor network, which typically consists of surface-mounted wireless sensors, represents a salient example that is subject to such threat. For example, when a person with body sensors walks through a public area, a sensor might be lost and then collected by an attacker. Both programs (i.e., program instructions) and data, no matter in disk or memory, may reveal sensitive information of the monitored person such as personal identity, health concerns (e.g., by the intention of program instructions), and etc.

Addressing such security concerns calls for *self-destruct* sensors that can *autonomously* destroy sensitive information if a sensor is lost. A self-destructing sensor necessitates two capabilities. First, each sensor needs to be able to determine whether it is lost. Second, it needs to assure that the sensitive information is timely destroyed. Fortunately, driven by the fractionated design paradigm [2–4], an increasing number of system (particularly cyber-physical systems) integrate sensors using short-range wireless communications, such as energy-efficient bluetooth [5] and Near-Field-Communication [6]. Some sensors even leverage human body as wireless communication channels [7,8]. All these communications are usually sensitive to distance: if a sensor is lost, it will also be likely to lose the connections with other sensors. Consequently, the availability of the wireless communication among wireless sensors offers a *light-weight* and *autonomous* means to identify whether a sensor is lost. Specifically, if a sensor fails to establish wireless connections with other sensors, it can conclude that it is lost. Despite its pragmatic simplicity, this solution is likely to incur a high false positive rate since the wireless communications are prone to temporal disruption. For example, a user may detach a sensor belonging to his body sensor network (BSN) before he takes a shower and therefore this sensor may lose connection with the network temporally. Therefore, *a self-destructing wireless*

[*] Corresponding author.
E-mail addresses: li.137@wright.edu (Y. Li), wang.137@wright.edu (X. Wang), daewook.kim@eku.edu (D.W. Kim), junjie.zhang@wright.edu (J. Zhang), rui.dai@uc.edu (R. Dai).

https://doi.org/10.1016/j.comnet.2018.05.013
1389-1286/© 2018 Elsevier B.V. All rights reserved.

sensor has to tolerate potential false positives introduced by the autonomous loss detection means.

In response, we propose to adopt a cryptographic method to "destruct" the sensitive information. Specifically, we proactively encrypt the sensitive instructions and data in a sensor node. The encrypted information is stored locally in the sensor while the decryption key is remotely stored in another node (named as a base node) that is unlikely to be lost (e.g., a cellphone that serves as a base station for a body sensor network). When a task is triggered by the monitored physical entity/environment, the sensor will first retrieve the key from the base node, then decrypt the encrypted instructions and data that are relevant to the task, next load them into the memory, and finally execute instructions with data. Once the task is finished, the sensor will delete all decrypted sensitive information as well as the local copy of the key. Such design offers two unique advantages. On the one hand, as long as the task is completed, a sensor will only contain the encrypted sensitive information. On the other hand, it tolerates false positives of loss detection since the sensor node will resume proper operation once it can reestablish wireless communications with other nodes. For example, if a lost sensor belonging to a BSN is found by its user, it can resume its operation after successfully retrieving the key from other nodes.

Nevertheless, this method is likely to significantly degrade system performance considering the resource-constrained nature of typical wireless sensors. Specifically, the direct adoption of this method means that the execution of every single task will trigger communication-intensive key retrieval and computation-intensive decryption. As a result, sensors' resources, particularly the power, could be quickly depleted. One straightforward yet effective mitigation solution is to adopt a caching mechanism. Specifically, decrypted sensitive information can be cached in a sensor for a certain amount of time (T_{cache}) after the task is completed. In this case, if a new task is triggered before the cache expires, both the key retrieval and decryption can be avoided.

Intuitively, T_{cache} implies trade-offs between security and performance: when T_{cache} increases, less times of key retrieval and decryption will be needed, reducing the energy consumption; a larger value of T_{cache} renders more time for attackers to compromise the system. Assigning a proper value to T_{cache}, however, faces great challenges as a result of multiple interacting factors such as task generation patterns, attackers' capabilities, and the cache mechanism.

In order to overcome these challenges, we propose to design a framework to guide the design and configuration of a wireless sensor, enabling assured and optimized balance between security and performance. The focus of this framework is a stochastic model capable of characterizing cache, attackers, the task generation, and the interactions among them. This model makes possible the rigorous and quantitative analysis towards evaluating how the value of T_{cache} impacts the system performance and security. Specifically, we made the following contributions:

1. We have developed a stochastic framework based on Discrete-Time Markov Chains (DTMC) to holistically characterize how three critical factors, including the task generation, the cache time, and an attacker's capability, jointly impact the security and performance properties.
2. We have designed properties of security and performance and specified them in the form of probabilistic computation tree logic (PCTL). We analyze these properties against DTMC-based sensor models using rigorous probabilistic model checking, where the results can guide the design and configuration of a sensor that lead to assured and optimized balance between security and performance.

To the best of our knowledge, this is the first work that targets at the *systematic* design of self-destructing wireless sensors.

2. Related work

Confidentiality is a key security property of wireless sensor networks and active research has been conducted towards this direction. However, most of existing literatures [9–13] focus on securing communication channels against malicious eavesdropping and tampering. Unfortunately, none of these methods protects the sensitive information of a sensor node by taking advantage of wireless connections of sensors and meanwhile overcome their energy constraints [14,15]. Nevertheless, self destruct is highly desired capability to protect the confidentiality of wireless sensors when they are deployed in adversarial environments [15]. A few attempts have been made to integrate self-destruction into sensor design and implementation across both hardware and software layers. For example, The Vanishing Programmable Resources Project from DARPA seeks electronic systems capable of physically disappearing. Open platforms call for software-based self-destruct capabilities [14,15] that imply low cost and wide adoption. Specifically, Plastoi et al. [14] proposed an approach to efficiently monitor the power/battery of a sensor node and active self-destruction code (i.e., deleting sensitive information) before predicted energy depletion. Curiac et al. designed a method to evaluate the trustworthiness of a sensor node by analyzing its readings and then destroy the information of this sensor it is suspicious. Our method differs from these two approaches in a number of fundamental ways. First, different from these two methods that use computationally intensive algorithms to identify whether the sensor is low in battery or compromised, our method takes advantage of the nature that short-range wireless communications are sensitive to distance, which is extremely light-weight. Second, both these methods are sensitive to false positives (e.g., benign cases that are identified as suspicious cases), where falsely destructed information cannot be recovered. Comparably, our method can autonomously recover from false destruction by retrieving keys from another node. Finally, we propose a model to quantitatively study the trade-off between the confidentiality and performance, which, to the best of our knowledge, represents the first attempt towards this direction.

Similar to wireless sensors, traditional computing devices such as laptops and mobile phones also face data leakage concerns if they are lost. A few methods [16,17] have been proposed to address this challenge. Similar to our method, these methods rely on the same cryptographic mechanism to gain the control of data: sensitive information will be proactively encrypted using a key; the key, detached from the protected system, will be retrieved when a legitimate access attempt to the information presents. Sharif et al. [18] employed a similar cryptographic mechanism to thwart static reverse engineering (i.e., analyzing a malware binary without executing it). Specifically, sensitive instructions in the binary will be first encrypted in the memory; they will be decrypted for execution after the binary retrieves the correct key from the remote server controlled by the attacker. Since the static analysis of a malware binary protected by the proposed solution will never trigger key retrieval, reverse engineers will gain no knowledge about the decrypted instructions. However, once the malware is executed, it will retrieve the key and decrypt all sensitive information, making them visible to reverse engineers.

Our proposed framework uses a similar cryptographic mechanism compared to [16–18] to proactively protect the confidentiality of sensitive information in a sensor node. However, it differs from existing work in fundamental ways. First, the self-destructing data designed in [17] has stronger assumptions for both data and attackers compared to ours. Specifically, it considers that the data will never be used after it self-destructs; it also assumes that the