Ad Hoc Networks 7 (2009) 1229–1242

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Predistribution and local collaboration-based group rekeying for wireless sensor networks ☆

Wensheng Zhang [a,*], Sencun Zhu [b], Guohong Cao [b]

[a] Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011, United States
[b] Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 100084, United States

ARTICLE INFO

Article history:
Received 21 January 2008
Received in revised form 19 August 2008
Accepted 19 November 2008
Available online 13 December 2008

Keywords:
Wireless sensor networks
Security
Group key management
False message filtering

ABSTRACT

When a sensor network is deployed in a hostile environment, an adversary may launch such attacks as eavesdropping the communications and compromising sensor nodes. Using the compromised nodes, he may inject false sensing reports or modify the reports sent by other nodes. To defend against these attacks, researchers have proposed symmetric group key-based schemes. In these schemes, however, if a large number of nodes are compromised, many (sub)group keys will be revealed. This greatly endangers the filtering schemes, making them very ineffective or even useless. To address this problem, we propose a family of *predistribution and local collaboration-based group rekeying (PCGR)* schemes, which update the compromised group keys to prevent the compromised nodes from understanding the communications between noncompromised nodes or injecting false data. These schemes are designed based on a simple while controversial idea – preload future group keys into sensor nodes before their deployment. To protect the preloaded keys from being disclosed by compromised nodes, we propose a novel technique that requires neighboring nodes to collaborate to derive the future group keys. To the best of our knowledge, our schemes are the first set of *distributed* group rekeying schemes for sensor networks without involving online key servers. Extensive analysis and simulations are conducted to evaluate the proposed schemes, and the results show that the proposed schemes can achieve a good level of security, outperform several previous group rekeying schemes, and significantly improve the effectiveness of false data filtering.

1. Introduction

In many applications, e.g., battlefield surveillance and habitat monitoring, sensor networks are deployed in unattended or hostile environments. If the communications in the networks are not well protected, they can be easily eavesdropped by adversaries. Due to the lack of tamper resistant hardware, adversaries may even capture and reprogram nodes, or inject their own nodes into the network and induce the network to accept them as legitimate nodes [1]. Once in control of a few nodes, the adversaries can mount various attacks from inside the network. For example, a compromised node (intruder) may inject false sensing reports or maliciously modify reports that go through it. Under such attacks, the data sink may accept wrong sensing data and take inappropriate responses, which could cause catastrophic impacts in some strategic scenarios.

To prevent outsiders from eavesdropping messages, legitimate sensor nodes can share one or more group keys [2] for encrypting the messages exchanged among them. To thwart the message injection attacks, messages should be authenticated. Due to high computational and communication overhead, the digital signature-based authentication techniques are not suitable for sensor networks [3].

☆ A preliminary version of this paper appears in the proceedings of IEEE INFOCOM 2005.

* Corresponding author. Tel.: +1 515 294 2821.
E-mail addresses: wzhang@cs.iastate.edu (W. Zhang), szhu@cse.psu.edu (S. Zhu), gcao@cse.psu.edu (G. Cao).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.11.004

Therefore, researchers proposed to adopt symmetric group key-based techniques such as the *statistical en-route filtering (SEF)* scheme [4]. The basic idea of the SEF scheme is as follows: sensor nodes are randomly assigned to multiple groups before deployment, and nodes in the same group are preloaded with a same group key. When a sensor node wants to send a message to a sink, the message is authenticated using multiple MACs contributed by its neighbors and each MAC is generated using one group key. When such a message is forwarded along a path to the sink, each en-route node uses its group key to verify one MAC carried in the message. If an en-route node is compromised, it may modify a passing message or inject a false message. However, it normally knows only one group key, and thus it can only forge one MAC correctly. So the modification or injection will be detected by other en-route nodes who know different group keys.

The group key-based techniques, however, will become ineffective if some nodes are compromised, since an adversary may obtain group keys from the compromised nodes. To deal with node compromise, the compromised nodes should be identified and then the noncompromised nodes should update their group keys to prevent the adversary from making use of the captured keys. In the literature, various techniques have been proposed for identifying compromised nodes. For example, nodes may use the *watchdog* mechanism [5] to monitor its neighbors and identify the compromised nodes when observing misbehavior. The identification accuracy can be further improved by using some collaborative intruder identification schemes [6]. In addition to identifying compromised nodes based on misbehavior observed, Seshadri et al. [7] have proposed a software-based attestation technique to detect nodes that have been reprogrammed.

Although the problem of detecting compromised nodes has attracted lots of interest and been studied extensively, the problem of efficiently updating group keys in sensor networks has not been delved. In the context of secure multicast in wired networks, many centralized schemes [8–11] and a few distributed schemes [12] have been proposed. However, most of them are not suitable for sensor networks. For example, in SKDC [8], each key updating requires N key encryptions and N transmissions (N is the number of nodes in the networks) of keys from the central controller to each individual node, which results in very high communication overhead and rekeying delay. The logic tree-based schemes proposed by Wallner et al. [9], Wong et al. [10], and Balenson et al. [11] can achieve logarithmic broadcast size, storage, and computational cost. However, the communication cost and the rekeying delay are still high when they are applied to a large scale sensor network with high resource constraints. Furthermore, a central controller has to be online to trace the status of all nodes, and maintain a large logic tree connecting all the trusted nodes, leading to high management overhead. The distributed solutions, e.g., Blundo's scheme [12], allow a set of nodes to set up a group key in a distributed way. However, it is still not scalable or efficient since each node must communicate with other trusted members in the same group, and the storage cost of each node increases rapidly as the group size increases.

To address the group rekeying problem for sensor networks, we propose a family of distributed and localized group rekeying schemes, called the *predistribution and local collaboration-based group rekeying* (*PCGR*) schemes. The design of these schemes are motivated by the following ideas: (1) future keys can be preloaded to individual nodes before deployment to avoid the high overhead in securely disseminating new keys at the key updating time. (2) Neighbors can collaborate with each other to effectively protect and appropriately use the preloaded keys; the local collaboration also avoids the high cost of the centralized management. Based on the above ideas, we first propose a *basic PCGR* (*B-PCGR*) scheme, in which one-hop neighbors collaborate to protect their group key polynomials. In B-PCGR, a group key polynomial is revealed if a node is compromised along with a certain threshold number of its one-hop neighbors. To address this limitation and achieve higher resilience to node compromise, we propose an enhanced PCGR scheme called *cascading PCGR*, in which each node can collaborate with nodes beyond its one-hop neighborhood to protect its group key polynomial. To the best of our knowledge, PCGR schemes are the first set of distributed group rekeying schemes for sensor networks. This distributed property is critical for unattended sensor networks deployed in adversarial environments because the central authority is a single point of failure from security and performance perspectives. Extensive analysis is conducted to evaluate the security level and the performance of the proposed schemes. We also compare the performance of the proposed schemes with several existing group rekeying schemes. Simulations are used to evaluate the effectiveness of the proposed group rekeying scheme in filtering false data. The analysis and simulation results show that the proposed schemes can achieve a good level of security, outperform several previously proposed schemes, and significantly improve the effectiveness and efficiency of false data filtering.

The rest of this paper is organized as follows. The next section presents the system model. In Section 3, we describe and analyze the basic PCGR scheme. An enhanced PCGR scheme is presented in Section 4. Section 5 reports the performance evaluation results. Section 6 discusses a number of issues related to the proposed schemes. Section 7 concludes the paper.

2. System model

We consider a large scale wireless sensor network, which is deployed in an unattended and hostile environment. The network is composed of low-complexity sensor nodes, e.g. the Berkeley MICA mote [13], which has a processor running at 4 MHz and 4KB RAM. These nodes are also limited in power supply, bandwidth, and computational capability. Therefore, public key-based per packet authentication cannot be afforded [14]. On the other hand, each node has enough space for storing a few kilobytes of keying information.

Node deployment is managed by a (offline) central controller (or setup server), which is responsible for picking group keys and preloading keying information to every