Science of Computer Programming 81 (2014) 53–78

Contents lists available at ScienceDirect

Science of Computer Programming

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

A semantic analysis of key management protocols for wireless sensor networks☆

Damiano Macedonio, Massimo Merro*

Dipartimento di Informatica, Università degli Studi di Verona, Italy

ARTICLE INFO

Article history:
Received 16 September 2011
Received in revised form 29 May 2012
Accepted 18 January 2013
Available online 6 February 2013

Keywords:
Wireless sensor network
Key management protocol
Security analysis
Process calculus

ABSTRACT

Gorrieri and Martinelli's *timed Generalized Non-Deducibility on Compositions* (*tGNDC*) schema is a well-known general framework for the formal verification of security protocols in a concurrent scenario. We generalise the *tGNDC* schema to verify wireless network security protocols. Our generalisation relies on a simple *timed broadcasting process calculus* whose operational semantics is given in terms of a labelled transition system which is used to derive a standard *simulation theory*. We apply our *tGNDC* framework to perform a security analysis of three well-known *key management protocols* for wireless sensor networks: μTESLA, LEAP+ and LiSP.

1. Introduction

Wireless sensors are small and cheap devices powered by low-energy batteries, equipped with radio transceivers, and responding to physical stimuli, such as pressure, magnetism and motion, by emitting radio signals. Such devices are featured with *resource constraints* (involving power, storage and computation) and low transmission rates. *Wireless sensor networks* (WSNs) are large-scale networks of sensor nodes deployed in strategic areas to gather data. Sensor nodes collaborate using wireless communications with an asymmetric many-to-one data transfer model. Typically, they send their sensed events or data to a specific node, called sink node or base station, which collects the requested information. WSNs are primarily designed for monitoring environments that humans cannot easily reach (e.g., motion, target tracking, fire detection, chemicals, temperature); they are used as embedded systems (e.g., biomedical sensor engineering, smart homes) or mobile applications (e.g., when attached to robots, soldiers, or vehicles).

An important issue in WSNs is *network security*: sensor nodes are vulnerable to several kinds of threats and risks. Unlike wired networks, wireless devices use radio frequency channels to broadcast their messages. An adversary can compromise a sensor node, alter the integrity of the data, eavesdrop on messages, inject fake messages, and waste network resource. Thus, one of the challenges in developing trustworthy WSNs is to provide high-security features with limited resources.

Generally, in order to have a secure communication between two (or more) parties, a secure association must be established by sharing a secret. This secret must be created, distributed and updated by one (or more) entity and it is often represented by the knowledge of a *cryptographic key*. The management of such cryptographic keys is the core of any security protocol. Due to resource limitations, all key management protocols for WSNs, such as μTESLA [1], LiSP [2], LEAP [3], PEBL [4] and INF [5], are based on *symmetric cryptography* rather than heavy public-key schemes, such as Diffie–Hellman [6] and RSA [7].

☆ The first author is supported by research fellowship n. AdR1601/11, funded by Dipartimento di Informatica Verona. Work partially supported by the PRIN 2010–2011 national project "Security Horizons".

* Corresponding author. Tel.: +39 0458027992; fax: +39 0458027068.
E-mail addresses: massimo.merro@univr.it, massimo.merro@gmail.com (M. Merro).

0167-6423/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2013.01.005

In this paper, we adopt a process calculus approach to formalise and verify real-world key management protocols for WSNs. A process calculus is a formal and concise language that allows us to express system behaviour in the form of a process term. We propose a simple *timed broadcasting process calculus*, called aTCWS, for modelling wireless networks. The time model we adopt is known as the *fictitious clock* approach (see e.g. [8]): A global clock is supposed to be updated whenever all nodes agree on this, by globally synchronising on a special timing action σ.[1] Broadcast communications span over a limited area, called *transmission range*. Both broadcast actions and internal actions are assumed to take no time. This is a reasonable assumption whenever the duration of those actions is negligible with respect to the chosen time unit. The operational semantics of our calculus is given in terms of a labelled transition semantics in the SOS style of Plotkin. The calculus enjoys standard time properties, such as: *time determinism*, *maximal progress*, and *patience* [8]. The labelled transition semantics is used to derive a (weak) *simulation theory* which can be easily *mechanised* by relying on well-known interactive theorem provers such as Isabelle/HOL [10] or Coq [11].

Based on our simulation theory, we generalise Gorrieri and Martinelli's *timed Generalized Non-Deducibility on Compositions* (*tGNDC*) schema [12,13], a well-known general framework for the formal verification of timed security properties. The basic idea of *tGNDC* is the following: a protocol M satisfies $tGNDC^{\rho(M)}$ if the presence of an arbitrary *attacker* does not affect the behaviour of M with respect to the abstraction $\rho(M)$. By varying $\rho(M)$ it is possible to express different timed security properties for the protocol M. Examples are the *timed integrity* property, which ensures the freshness of authenticated packets, and the *timed agreement* property, when agreement between two parties must be reached within a certain deadline. In order to avoid the universal quantification over all possible attackers when proving *tGNDC* properties, we provide a *compositional* proof technique based on the notion of *the most powerful attacker*.

We use our calculus to provide a formal specification of three well-known key management protocols for WSNs: (i) μTESLA [1], which achieves *authenticated broadcast*; (ii) the *Localized Encryption and Authentication Protocol*, LEAP+ [3], intended for large-scale wireless sensor networks; (iii) the *Lightweight Security Protocol*, LiSP [2], that, through an efficient mechanism of re-keying, provides a good trade-off between resource consumption and network security.

We perform a *tGNDC*-based analysis on these three protocols. As a result of our analysis, we formally prove that the *authenticated-broadcast phase* of μTESLA enjoys both timed integrity and timed agreement. Then, we prove that the *single-hop pairwise shared key* mechanism of LEAP+ enjoys timed integrity but not timed agreement, due to the presence of a replay attack despite the security assessment of [3]. Finally, we prove that the LiSP protocol satisfies neither timed integrity nor timed agreement. Again, this is due to the presence of a replay attack. To our knowledge both attacks are new and they have not yet appeared in the literature.

We end this introduction with an outline of the paper. In Section 2, we provide syntax, operational semantics and behavioural semantics of aTCWS. In the same section we prove that our calculus enjoys time determinism, maximal progress and patience. In Section 3, we adapt Gorrieri and Martinelli's *tGNDC* framework to aTCWS. In Sections 4–6 we provide a security analysis of the three key management protocols mentioned above. The paper ends with a section on conclusions, future and related work.

2. The calculus

In Table 1, we provide the syntax of our *applied Timed Calculus for Wireless Systems*, in short aTCWS, in a two-level structure: A lower one for *processes* and an upper one for *networks*. We assume a set *Nds* of logical node names, ranged over by letters m, n. *Var* is the set of *variables*, ranged over by x, y, z. We define *Val* to be the set of values, and *Msg* to be the set of *messages*, i.e., closed values that do not contain variables. Letters u, $u_1 \ldots$ range over *Val*, and w, $w' \ldots$ range over *Msg*.

Both syntax and operational semantics of aTCWS are parametric with respect to a given *decidable* inference system, i.e. a set of rules to model operations on messages by using constructors. For instance, the rules

$$(\text{pair})\ \frac{w_1 \quad w_2}{\text{pair}(w_1, w_2)} \qquad (\text{fst})\ \frac{\text{pair}(w_1, w_2)}{w_1} \qquad (\text{snd})\ \frac{\text{pair}(w_1, w_2)}{w_2}$$

allow us to deal with pairs of values. We write $w_1 \ldots w_k \vdash_{\mathrm{r}} w_0$ to denote an application of rule r to the closed values $w_1 \ldots w_k$ to infer w_0. Given an inference system, the *deduction function* $\mathcal{D} : 2^{Msg} \to 2^{Msg}$ associates a (finite) set ϕ of messages to the set $\mathcal{D}(\phi)$ of messages that can be deduced from ϕ, by applying instances of the rules of the inference system.

Networks are collections of nodes running in parallel and using a unique common channel to communicate with each other. All nodes have the same transmission range (this is a quite common assumption in models for ad hoc networks [14]). The communication paradigm is *local broadcast*: only nodes located in the range of the transmitter may receive data. We write $n[P]^{\nu}$ for a node named n (the device network address) executing the sequential process P. The tag ν contains the neighbours of n ($\nu \subseteq Nds \setminus \{n\}$). In other words, ν contains all nodes in the transmission cell of n (except n itself), thus

[1] Time synchronisation relies on some clock synchronisation protocol [9].