Available online at www.sciencedirect.com

SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 135 (2005) 95–114

www.elsevier.com/locate/entcs

Formalizing and Analyzing the Needham-Schroeder Symmetric-Key Protocol by Rewriting

Monica Nesi[1], [2] and Giuseppina Rucci

Dipartimento di Informatica
Università degli Studi di L'Aquila
via Vetoio, 67010 L'Aquila, Italy

Abstract

This paper reports on work in progress on using rewriting techniques for the specification and the verification of communication protocols. As in Genet and Klay's approach to formalizing protocols, a rewrite system $\mathcal{R}$ describes the steps of the protocol and an intruder's ability of decomposing and decrypting messages, and a tree automaton $\mathcal{A}$ encodes the initial set of communication requests and an intruder's initial knowledge. In a previous work we have defined a rewriting strategy that, given a term t that represents a property of the protocol to be proved, suitably expands and reduces t using the rules in $\mathcal{R}$ and the transitions in $\mathcal{A}$ to derive whether or not t is recognized by an intruder. In this paper we present a formalization of the Needham-Schroeder symmetric-key protocol and use the rewriting strategy for deriving two well-known authentication attacks.

Keywords: Protocol verification, term rewriting, tree automata, rewriting strategy.

1 Introduction

In the past few years several approaches have been applied to protocol specifications in order to formally verify various properties of interest, such as authentication, secrecy or confidentiality, freshness, etc. These approaches range from model checking [24,27,4] to theorem proving [26,35,36,37,22] through process calculi [1,8,9], Horn clauses [6], multiset rewriting and strand spaces [5,10],

[1] Work partially supported by MIUR ex-40% project "Infrastrutture e architetture software per applicazioni globali, adattabili e attendibili (GHIBLI)".

[2] Email: monica@di.univaq.it

doi:10.1016/j.entcs.2005.06.002

rewriting techniques and strategies [11,14,23] using tree automata and abstract interpretation [18,19,28]. Most of these verification approaches have also been implemented using either specific-purpose tools, such as AVISPA [2], CASRUL [23], NRL [26] and Timbuk [19], or general-purpose tools, such as ELAN [11,18], FDR [24], Isabelle [35,36], Maude [14] and SPASS [37]. There has also been some work on comparing and combining different approaches, e.g. the combination of Genet and Klay's approximation technique with Paulson's inductive method [33,34].

We are interested in the use of rewriting based techniques for the formalization and the verification of communication protocols. Rewrite systems provide a very natural approach to operationally describe the behaviour of a protocol. In particular, rewrite systems and tree automata are used in [17,18,19] to specify and verify properties of security protocols by developing an approximation technique that aims at finding that there are no attacks on a protocol, rather than at discovering attacks. The protocol is specified through a rewrite system $\mathcal{R}$, while the initial set E of communication requests and an intruder's initial knowledge are described through a tree automaton $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) \supseteq E$. Starting from $\mathcal{R}$ and $\mathcal{A}$, the approximation technique by Genet and Klay builds a tree automaton which over-approximates the set of the messages exchanged among the protocol agents. The quality of the approximation depends on an approximation function γ which defines the subterms that can be approximated. The approximation technique can be seen as a particular completion process between $\mathcal{R}$ and $\mathcal{A}$, as critical pairs are computed between the rules in $\mathcal{R}$ and the transitions in $\mathcal{A}$. The rules derived from the critical pairs are new transitions that are normalized using γ and then added to $\mathcal{A}$. Thus, the language recognized by the resulting approximation automaton $\mathcal{T}_{\mathcal{R}}\!\uparrow\!(\mathcal{A})$ includes all $\mathcal{R}$-descendants of E. In this way, in order to prove whether a property p is satisfied, it is sufficient to consider the intersection between the language of $\mathcal{T}_{\mathcal{R}}\!\uparrow\!(\mathcal{A})$ and the language of a tree automaton $\mathcal{A}_{\overline{p}}$ which models the negation of p and thus contains the "prohibited" terms. If such intersection is empty, then p is satisfied.

In developing our approach to verifying security protocols, we have been borrowing Genet and Klay's formalization of a protocol, i.e. a rewrite system $\mathcal{R}$ and a tree automaton $\mathcal{A}$. Then, given a term t that describes a property to be proved, we apply a rewriting strategy (defined in [31]) that suitably expands and reduces t using the rules in $\mathcal{R}$ and the transitions in $\mathcal{A}$ to derive whether or not t is recognized by an intruder. This is done by simulating a completion process in a bottom-up manner starting from t and trying to derive if a transition $t \rightarrow q_f$ can be generated from critical pairs, where q_f is a final state of $\mathcal{A}$. If the transition $t \rightarrow q_f$ is derived by the strategy, this means that