LTL Model-Checking for Security Protocols

Roberto Carbone *
Fondazione Bruno Kessler, Trento, Italy
Phone: +39.0461.314.192 Fax: +39.0461.302.040
E-mail: carbone@fbk.eu

Abstract. This thesis is about the application of automated reasoning techniques to the formal analysis of security protocols. More in detail, it proposes a general model-checking framework for security protocols based on a set-rewriting formalism that, coupled with the use of Linear Temporal Logic, allows for the specification of assumptions on principals and communication channels as well as complex security properties that are normally not handled by state-of-the-art protocol analyzers. The approach successfully combines encoding techniques originally developed for planning with bounded model-checking techniques.

The effectiveness of the approach proposed is assessed against the formal analysis of relevant security protocols, with the detection of a severe security flaw in Google's SAML-based SSO for Google Apps and a previously unknown attack on a patched version of the ASW contract signing protocol.

Keywords: security protocols, bounded model-checking, planning, graphplan, propositional satisfiability (SAT)

1. Introduction

Security protocols are communication protocols that aim at providing security guarantees through the application of cryptographic primitives. Since these protocols are at the core of security-sensitive applications in a variety of domains (e.g. healthcare, e-commerce, and e-government), their proper functioning is crucial as a failure may undermine the customer and, more in general, the public trust in these applications. In spite of their apparent simplicity, security protocols are notoriously error-prone. It is thus of utmost importance to have tools and specification languages that support the activity of finding flaws in protocols.

Model checking [1] is a powerful and automatic technique for verifying concurrent systems. It has been applied widely and successfully in practice to verify digital sequential circuit designs, and, more recently, important results have been obtained for the analysis of security protocols [2]. Most model checking techniques for security protocols make a number of simplifying assumptions on the protocol and/or on its execution environment that prevent their applicability in some important cases. For instance, most techniques assume that communication between honest principals is controlled by a Dolev-Yao intruder [3], i.e. a malicious agent capable to overhear, divert, and fake messages. Yet we might be interested in establishing the security of a protocol that relies on a less insecure channel (e.g. confidential and/or authentic channels provided by some other protocol sitting lower in the protocol stack).

This thesis presents a general model-checking framework for security protocols that allows for the specification of assumptions on principals and communication channels as well as complex security properties that are normally not handled by state-of-the-art protocol analyzers. This approach extends the one described in [4] based on a bounded model checking technique for security protocols analysis that reduces the problem of determining whether a security protocol violates a security property in $k > 0$ steps to the propositional satisfiability (SAT) problem. The extension provides support to Linear Temporal Logic (LTL). The tool SATMC, implementing this approach, has been extended accordingly.

The effectiveness of the approach proposed is assessed by running SATMC against the formal analysis of relevant security protocols.

2. Contributions of the thesis

The main contribution of this thesis is the definition of a framework for security protocols based

*I am indebted with Alessandro Armando for his help and support during the whole period of the thesis. This work was partially supported by the FP7-ICT-2007-1 Project no. 216471, "AVANTSSAR" (`www.avantssar.eu`), and the "SIAM" project (`http://st.fbk.eu`).

on a rewriting-based formalism (as done in [5]) and LTL. It allows for the formal specification of the transition system specifying all the possible actions of the participating agents in a given protocol scenario. The thesis also shows that the proposed framework supports the modeling of a number of relevant intruder models as well as the straightforward specification of important security properties. A *protocol security problem* is a pair $\Xi = \langle \Gamma, \varphi \rangle$, where Γ is a set-rewriting system, defined by a set of *facts* (i.e. atomic formulae) and a set of *rewrite rules* (rules, for short), which models the behaviors of the honest principals and of the Dolev-Yao intruder. φ is the following LTL formula:

$$(C_I \wedge C_H) \supset G$$

where C_I and C_H are LTL formulae that constrain the allowed behaviors of the intruder and of the honest principals respectively, and G is an LTL formula stating the security properties that the protocol is expected to enjoy. The protocol security problem is the problem of establishing whether all the execution traces of Γ satisfy φ. Any trace violating φ is an *attack trace* of the protocol specified by Γ.

After having modelled a security protocol and its requirements as a protocol security problem, we focus on the problem of reducing a protocol security problem to SAT. Given a protocol security problem $\Xi = \langle \Gamma, \varphi \rangle$ and an integer $k > 0$, we build a propositional formula $[\![\Xi]\!]_k$ such that every model of $[\![\Xi]\!]_k$ corresponds to a (linearizable) partial-order solution of Ξ of length k and vice versa. This is done by adding a time-index to the rules and facts to indicate the state at which the rules apply or the facts hold. Facts and rules are thus indexed by 0 through k. If p is a fact or a rule and i is an index, then $i{:}p$ is the corresponding time-indexed propositional variable. If $\mathbf{p} = p_1, \ldots, p_n$ is a tuple of facts or rules and i is an index, then $i{:}\mathbf{p} = i{:}p_1, \ldots, i{:}p_n$ is the corresponding time-indexed tuple of propositional variables. The formula $[\![\Xi]\!]_k$ is defined by

$$[\![\Xi]\!]_k = [\![\Gamma]\!]_k \wedge [\![\varphi]\!]_k$$

where $[\![\Gamma]\!]_k$ is a propositional formula encoding the behavior of the set rewriting system Γ and $[\![\varphi]\!]_k$ is a propositional formula encoding the set of attack traces of length k, as done in [6]. The propositional formula $[\![\Gamma]\!]_0 = I(0{:}\mathbf{f})$, while $[\![\Gamma]\!]_k$, for $k > 0$ is of the form:

$$[\![\Gamma]\!]_k = I(0{:}\mathbf{f}) \wedge \bigwedge_{i=0}^{k-1} T_i(i{:}\mathbf{f}, i{:}\boldsymbol{\rho}, i+1{:}\mathbf{f})$$

where $\mathbf{f}$ and $\boldsymbol{\rho}$ are tuples of facts and rules respectively. The formula $I(0{:}\mathbf{f})$ encodes the initial state whereas the formula $T_i(i{:}\mathbf{f}, i{:}\boldsymbol{\rho}, i+1{:}\mathbf{f})$ encodes all the possible evolutions of the system from step i to step $i+1$.

The protocol security problem of length k is then constructed by conjoining $[\![\Gamma]\!]_k$ with $[\![\varphi]\!]_k$. Initially $k = 0$, and then k is incremented till an attack is found or the available resources are exhausted.

Traditional techniques for the encoding of the system $[\![\Gamma]\!]_k$ lead to propositional formulae of unmanageable size, even for relatively simple problems. To overcome this difficulty, we have followed the approach proposed in [4], adapted from an encoding technique originally introduced for AI planning, namely the graphplan-based encoding [7]. Such encoding can drastically reduce the size of resulting SAT formulae, and lead to significant speed-ups of the SAT solver.

These ideas have been implemented by realizing an extension of the tool SATMC.[1]

The effectiveness of the approach proposed is assessed against the formal analysis of relevant security protocols. SATMC has revealed a severe security flaw in Google's SAML-based SSO for Google Apps [8], that allowed a dishonest service provider to impersonate a user at another service provider. Moreover, we detected a previously unknown attack on a patched version of the protocol proposed by Asokan, Shoup, and Waidner (ASW) [9,10], one of the most prominent optimistic fair exchange protocol, for online contract-signing.

All the details about the work can be found in the thesis [11].

References

[1] Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge, Massachusetts (1999)

[2] Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR. In Margaria, T., Steffen, B., eds.: Proceedings of TACAS'96. LNCS 1055, Springer-Verlag (1996) 147–166

[3] Dolev, D., Yao, A.: On the Security of Public-Key Protocols. IEEE Transactions on Information Theory **2**(29) (1983)

[1] The tool is available at `http://www.ai-lab.it/satmc`

[4] Armando, A., Compagna, L.: SAT-based Model-Checking for Security Protocols Analysis. International Journal of Information Security **7**(1) (2008) 3–32

[5] Cervesato, I., Durgin, N., Lincoln, P., Mitchell, J., Scedrov, A.: A meta-notation for protocol analysis. In: Proceedings of the 12th IEEE Computer Security Foundations Workshop: CSFW'99. IEEE Computer Society Press (1999) 55–69

[6] Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic Model Checking without BDDs. In: Proceedings of TACAS'99. LNCS 1579, Springer-Verlag (1999) 193–207

[7] Kautz, H., McAllester, H., Selman, B.: Encoding Plans in Propositional Logic. In Aiello, L.C., Doyle, J., Shapiro, S., eds.: KR'96: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann (1996) 374–384

[8] Armando, A., Carbone, R., Compagna, L., Cuellar, J., Abad, L.T.: Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In Shmatikov, V., ed.: Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering (FMSE 2008), ACM Press (2008) 1–10

[9] Asokan, N., Shoup, V., Waidner, M.: Asynchronous protocols for optimistic fair exchange. In: Proceedings of the IEEE Symposium on Research in Security and Privacy. (1998) 86–99

[10] Armando, A., Carbone, R., Compagna, L.: LTL model checking for security protocols. In: 20th IEEE Computer Security Foundations Symposium (CSF20), Venice (Italy) (2007)

[11] Carbone, R.: LTL Model-Checking for Security Protocols. PhD thesis, University of Genova, Italy (2009) Available at `http://www.ai-lab.it/carbone/papers/Phd-thesis/carbone_phd_thesis.pdf`.