The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications*

A. Armando[1], D. Basin[2], Y. Boichut[3], Y. Chevalier[4], L. Compagna[1], J. Cuellar[5], P. Hankes Drielsma[2], P.C. Heám[3], O. Kouchnarenko[3], J. Mantovani[1], S. Mödersheim[2], D. von Oheimb[5], M. Rusinowitch[4], J. Santiago[4], M. Turuani[4], L. Viganò[2], and L. Vigneron[4]

[1] AI-Lab, DIST, Università di Genova, Italy
[2] Information Security Group, ETH Zurich, Switzerland
[3] LIFC, Université de Franche-Comté, Besancon, France
[4] LORIA-INRIA-Lorraine, Nancy, France
[5] Siemens AG, CT IC 3, Munich, Germany

Abstract. AVISPA is a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques. To the best of our knowledge, no other tool exhibits the same level of scope and robustness while enjoying the same performance and scalability.

1 Introduction

With the spread of the Internet and network-based services, the number and scale of new security protocols under development is out-pacing the human ability to rigorously analyze and validate them. To speed up the development of the next generation of security protocols, and to improve their security, it is of utmost importance to have tools that support the rigorous analysis of security protocols by either finding flaws or establishing their correctness. Optimally, these tools should be completely automated, robust, expressive, and easily usable, so that they can be integrated into the protocol development and standardization processes to improve the speed and quality of these processes.

A number of (semi-)automated protocol analysis tools have been proposed, e.g. [1, 4, 6, 7, 13, 14], which can analyze small and medium-scale protocols such as those in the Clark/Jacob library [10]. However, scaling up to large scale Internet security protocols is a considerable challenge, both scientific and technological. We have developed a push-button tool for the *A*utomated *V*alidation of *I*nternet

* This work was supported by the FET Open Project IST-2001-39252 and the BBW Project 02.0431, "AVISPA: Automated Validation of Internet Security Protocols and Applications" (`www.avispa-project.org`).

*S*ecurity-sensitive *P*rotocols and *A*pplications, the *AVISPA Tool*[6], which rises to this challenge in a systematic way by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating different back-ends that implement a variety of automatic analysis techniques ranging from *protocol falsification* (by finding an attack on the input protocol) to *abstraction-based verification* methods for both finite and infinite numbers of sessions. To the best of our knowledge, no other tool exhibits the same scope and robustness while enjoying the same performance and scalability.

2 The AVISPA Tool

As displayed in Fig.1, the AVISPA Tool is equipped with a web-based graphical user interface (`www.avispa-project.org/software`) that supports the editing of protocol specifications and allows the user to select and configure the different back-ends of the tool. If an attack on a protocol is found, the tool displays it as a message-sequence chart. For instance, Fig.1 shows part of the specification of Siemens' H.530 protocol (top-right window) and the attack that AVISPA has found (bottom window), reported on in [3]. The interface features specialized menus for both novice and expert users. An XEmacs mode for editing protocol specifications is available as well.

The AVISPA Tool consists of independently developed modules, interconnected as shown at the bottom left of Fig.1. A protocol designer interacts with the tool by specifying a *security problem* (a protocol paired with a security property that it is expected to achieve) in the *High-Level Protocol Specification Language HLPSL* [8]. The HLPSL is an expressive, modular, role-based, formal language that allows for the specification of control flow patterns, data-structures, alternative intruder models, complex security properties, as well as different cryptographic primitives and their algebraic properties. These features make HLPSL well suited for specifying modern, industrial-scale protocols.

The HLPSL enjoys both a declarative semantics based on a fragment of the Temporal Logic of Actions [12] and an operational semantics based on a translation into the rewrite-based formalism *Intermediate Format IF*. HLPSL specifications are translated into equivalent IF specifications by the *HLPSL2IF translator*. An IF specification describes an infinite-state transition system amenable to formal analysis. IF specifications can be generated both in an untyped variant and in a typed one, which abstracts away type-flaw attacks (if any) from the protocol; this is particularly useful as in many cases type-flaws can be prevented in the actual implementation of a protocol [11]. IF specifications are input to the back-ends of the AVISPA Tool, which implement different analysis techniques. The current version of the tool integrates the following four back-ends.

The **On-the-fly Model-Checker (OFMC)** [3] performs protocol falsification and bounded verification by exploring the transition system described by

[6] The AVISPA Tool is a successor to the AVISS Tool [1], which automated the analysis of security protocols like those in the Clark/Jacob library. The AVISPA Tool significantly extends its predecessor's scope, effectiveness, and performance.

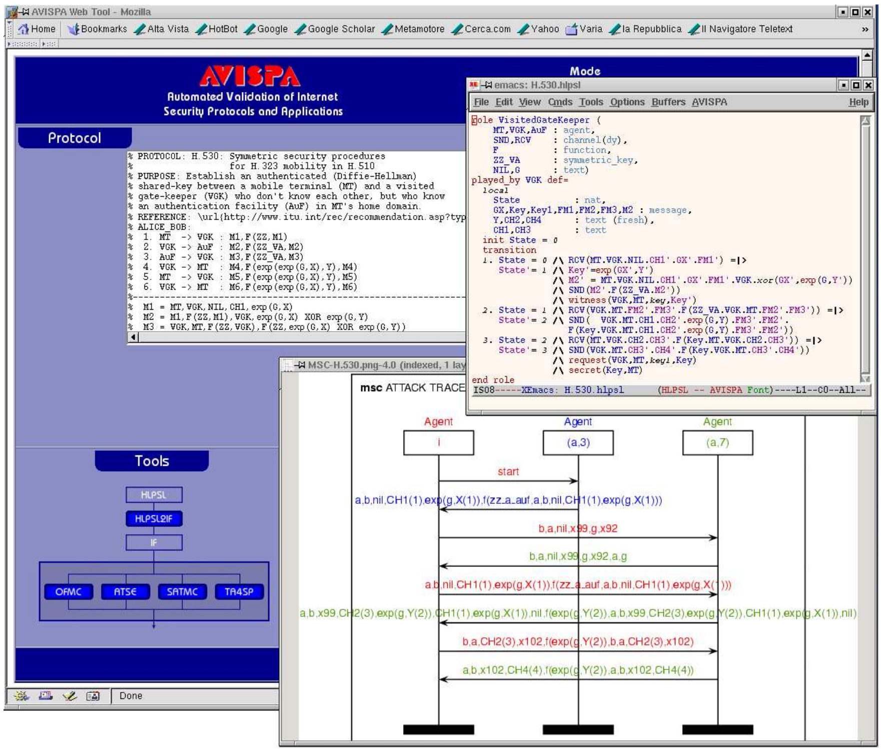

Fig. 1. A screen-shot of the AVISPA Tool.

an IF specification in a demand-driven way. OFMC implements a number of correct and complete symbolic techniques. It supports the specification of algebraic properties of cryptographic operators, and typed and untyped protocol models.

The **Constraint-Logic-based Attack Searcher (CL-AtSe)** applies constraint solving as in [9], with some powerful simplification heuristics and redundancy elimination techniques. CL-AtSe is built in a modular way and is open to extensions for handling algebraic properties of cryptographic operators. It supports type-flaw detection and handles associativity of message concatenation.

The **SAT-based Model-Checker (SATMC)** [2] builds a propositional formula encoding a bounded unrolling of the transition relation specified by the IF, the initial state and the set of states representing a violation of the security properties. The propositional formula is then fed to a state-of-the-art SAT solver and any model found is translated back into an attack.

The **TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols)** back-end [5] approximates the intruder knowledge by using regular tree languages and rewriting. For secrecy properties, TA4SP can show whether a protocol is flawed (by under-approximation) or whether it is safe for any number of sessions (by over-approximation).

Upon termination, the AVISPA Tool outputs the result of the analysis stating whether the input problem was solved (positively or negatively), the available resources were exhausted, or the problem was not tackled for some reason. In order to demonstrate the effectiveness of the AVISPA Tool on a large collection of practically relevant, industrial protocols, we have selected a substantial set of security problems associated with protocols that have recently been, or are currently being standardized by organizations like the Internet Engineering Task Force IETF. We have then formalized in HLPSL a large subset of these protocols, and the result of this specification effort is the *AVISPA Library* (publicly available at the AVISPA web-page), which at present comprises 112 security problems derived from 33 protocols. We have thoroughly assessed the AVISPA Tool by running it against the AVISPA Library. The experimental results are summarized in the appendix. In particular, the AVISPA Tool has detected a number of previously unknown attacks on some of the protocols analyzed, e.g. on some protocols of the ISO-PK family, on the IKEv2-DS protocol, and on the H.530 protocol.

References

1. A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Mödersheim, M. Rusinowitch, M. Turuani, L. Viganò, L. Vigneron. The AVISS Security Protocol Analysis Tool. In *Proc. CAV'02*, LNCS 2404. Springer, 2002.
2. A. Armando and L. Compagna. SATMC: a SAT-based Model Checker for Security Protocols. In *Proc. JELIA'04*, LNAI 3229. Springer, 2004.
3. D. Basin, S. Mödersheim, L. Viganò. OFMC: A Symbolic Model-Checker for Security Protocols. *International Journal of Information Security*, 2004.
4. B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proc. CSFW'01*. IEEE Computer Society Press, 2001.
5. Y. Boichut, P.-C. Heam, O. Kouchnarenko, F. Oehl. Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols. In *Proc. AVIS'04*, ENTCS, to appear.
6. L. Bozga, Y. Lakhnech, M. Perin. Hermes: An Automatic Tool for the Verification of Secrecy in Security Protocols. In *Proc. CAV'03*, LNCS 2725. Springer, 2003.
7. The CAPSL Integrated Protocol Environment: `www.csl.sri.com/~millen/`.
8. Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, J. Mantovani, S. Mödersheim, L. Vigneron. A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. In *Proc. SAPS'04*. Austrian Computer Society, 2004.
9. Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In *Proc. CAV'02*, LNCS 2404. Springer, 2002.
10. J. Clark and J. Jacob. A Survey of Authentication Protocol Literature: Version 1.0, 17. Nov. 1997. URL: `www.cs.york.ac.uk/~jac/papers/drareview.ps.gz`.
11. J. Heather, G. Lowe, S. Schneider. How to prevent type flaw attacks on security protocols. In *Proc. CSFW'00*. IEEE Computer Society Press, 2000.
12. L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, 1994.
13. L. C. Paulson. The Inductive Approach to Verifying Cryptographic Protocols. *Journal of Computer Security*, 6(1):85–128, 1998.

14. D. Song. Athena: A new efficient automatic checker for security protocol analysis. In *Proc. CSFW'99*. IEEE Computer Society Press, 1999.

Appendix: Experimental Results

The following table displays the results of running the AVISPA Tool against the 112 security problems from 33 protocols in the AVISPA Library. For each of the protocols, the table gives the number of security problems (#P), and for each back-end, the number of problems successfully analyzed with the given resources[7] (P), the number of problems for which attacks are detected (A), and the time (T) spent by the back-end to find the attacks or to report that no attack exists in the given (bounded) scenario, where "-" indicates that the back-end does not support the analysis of that problem. For SATMC, we list both the time spent to generate the SAT formula (TE) and that spent to solve the formula (TS). Note that the times of unsuccessful attempts (due to time out or memory out) are not taken into account. By running the TA4SP back-end

Problems		OFMC			CL-atse			SATMC			
Protocol	#P	P	A	T	P	A	T	P	A	TE	TS
UMTS_AKA	3	3	0	0,02	3	0	0,01	3	0	0,11	0,00
AAAMobileIP	7	7	0	0,75	7	0	0,20	7	0	1,32	0,01
ISO-PK1	1	1	1	0,02	1	1	0,00	1	1	0,05	0,00
ISO-PK2	1	1	0	0,05	1	0	0,00	1	0	1,62	0,00
ISO-PK3	2	2	2	0,04	2	2	0,01	2	2	0,27	0,00
ISO-PK4	2	2	0	0,54	2	0	0,03	2	0	1.153	1,16
LPD-MSR	2	2	2	0,02	2	2	0,02	2	2	0,17	0,02
LPD-IMSR	2	2	0	0,08	2	0	0,01	2	0	0,43	0,01
CHAPv2	3	3	0	0,32	3	0	0,01	3	0	0,55	0,00
EKE	3	3	2	0,19	3	2	0,04	3	2	0,22	0,00
TLS	3	3	0	2,20	3	0	0,32	3	0	-	0,00
DHCP-delayed	2	2	0	0,07	2	0	0,00	2	0	0,19	0,00
Kerb-Cross-Realm	8	8	0	11,86	8	0	4,14	8	0	113,60	1,69
Kerb-Ticket-Cache	6	6	0	2,43	6	0	0,38	6	0	495,66	7,75
Kerb-V	8	8	0	3,08	8	0	0,42	8	0	139,56	2,95
Kerb-Forwardable	6	6	0	30,34	6	0	10,89	0	0	-	-
Kerb-PKINIT	7	7	0	4,41	7	0	0,64	7	0	640,33	11,65
Kerb-preauth	7	7	0	1,86	7	0	0,62	7	0	373,72	2,57
CRAM-MD5	2	2	0	0,71	2	0	0,74	2	0	0,40	0,00
PKB	1	1	1	0,25	1	1	0,01	1	1	0,34	0,02
PKB-fix	2	2	0	4,06	2	0	44,25	2	0	0,86	0,02
SRP_siemens	3	3	0	2,86	0	0	-	0	0	-	-
EKE2	3	3	0	0,16	0	0	-	0	0	-	-
SPEKE	3	3	0	3,11	0	0	-	0	0	-	-
IKEv2-CHILD	3	3	0	1,19	0	0	-	0	0	-	-
IKEv2-DS	3	3	1	5,22	0	0	-	0	0	-	-
IKEv2-DSx	3	3	0	42,56	0	0	-	0	0	-	-
IKEv2-MAC	3	3	0	8,03	0	0	-	0	0	-	-
IKEv2-MACx	3	3	0	40,54	0	0	-	0	0	-	-
h.530	3	1	1	0,64	0	0	-	0	0	-	-
h.530-fix	3	3	0	4.278	0	0	-	0	0	-	-
lipkey-spkm-known	2	2	0	0,23	0	0	-	0	0	-	-
lipkey-spkm-unknown	2	2	0	7,33	0	0	-	0	0	-	-

on a subset of the AVISPA Library, the AVISPA Tool is able to establish in a few minutes that a number of protocols in the library (namely, EKE, EKE2, IKEv2-CHILD, IKEv2-MAC, TLS, UMTS_AKA, CHAPv2) guarantee secrecy.

[7] Results are obtained by each single back-end with a resource limit of 1 hour CPU time and 1GB memory, on a Pentium IV 2.4GHz under Linux.