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

Alessandro Armando

AI-Lab, DIST, Università di Genova

Università di Genova INRIA-Lorraine ETH Zurich Siemens AG

Automated Validation of Internet Security Protocols and Applications

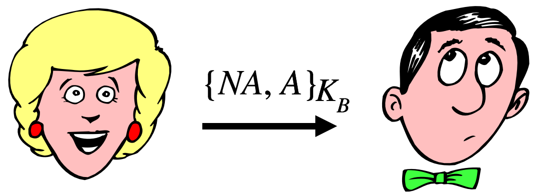
Shared cost RTD (FET open) project IST-2001-39252

Motivation

- 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.

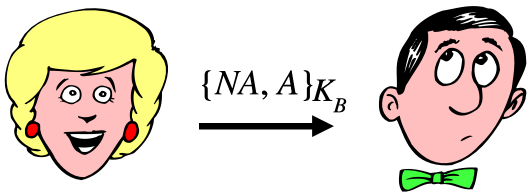
The state of the art

- Several (semi-)automated protocol analyzers have been proposed, BUT automatic analysis limited to small and medium-scale protocols.
  - For example, Clark/Jacob protocol library: NSPK, NSSK, Otway-Rees, Yahalom, Woo-Lam, Denning-Sacco, ...



The state of the art

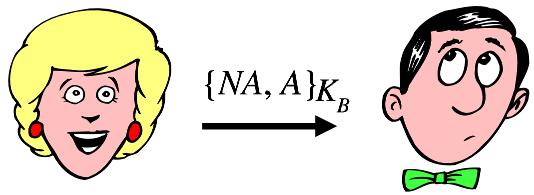
- Several (semi-)automated protocol analyzers have been proposed, BUT automatic analysis limited to small and medium-scale protocols.
  - For example, Clark/Jacob protocol library: NSPK, NSSK, Otway-Rees, Yahalom, Woo-Lam, Denning-Sacco, ...



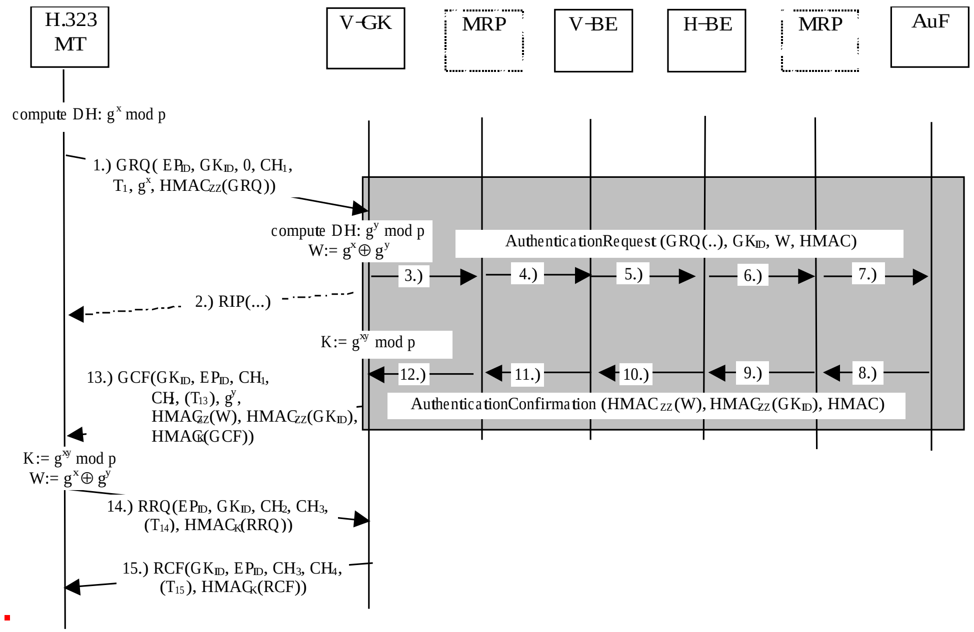
  - Most tools come with their own specification language and user interface.

The state of the art

- Several (semi-)automated protocol analyzers have been proposed, BUT automatic analysis limited to small and medium-scale protocols.
  - ▶ For example, Clark/Jacob protocol library: NSPK, NSSK, Otway-Rees, Yahalom, Woo-Lam, Denning-Sacco, ...



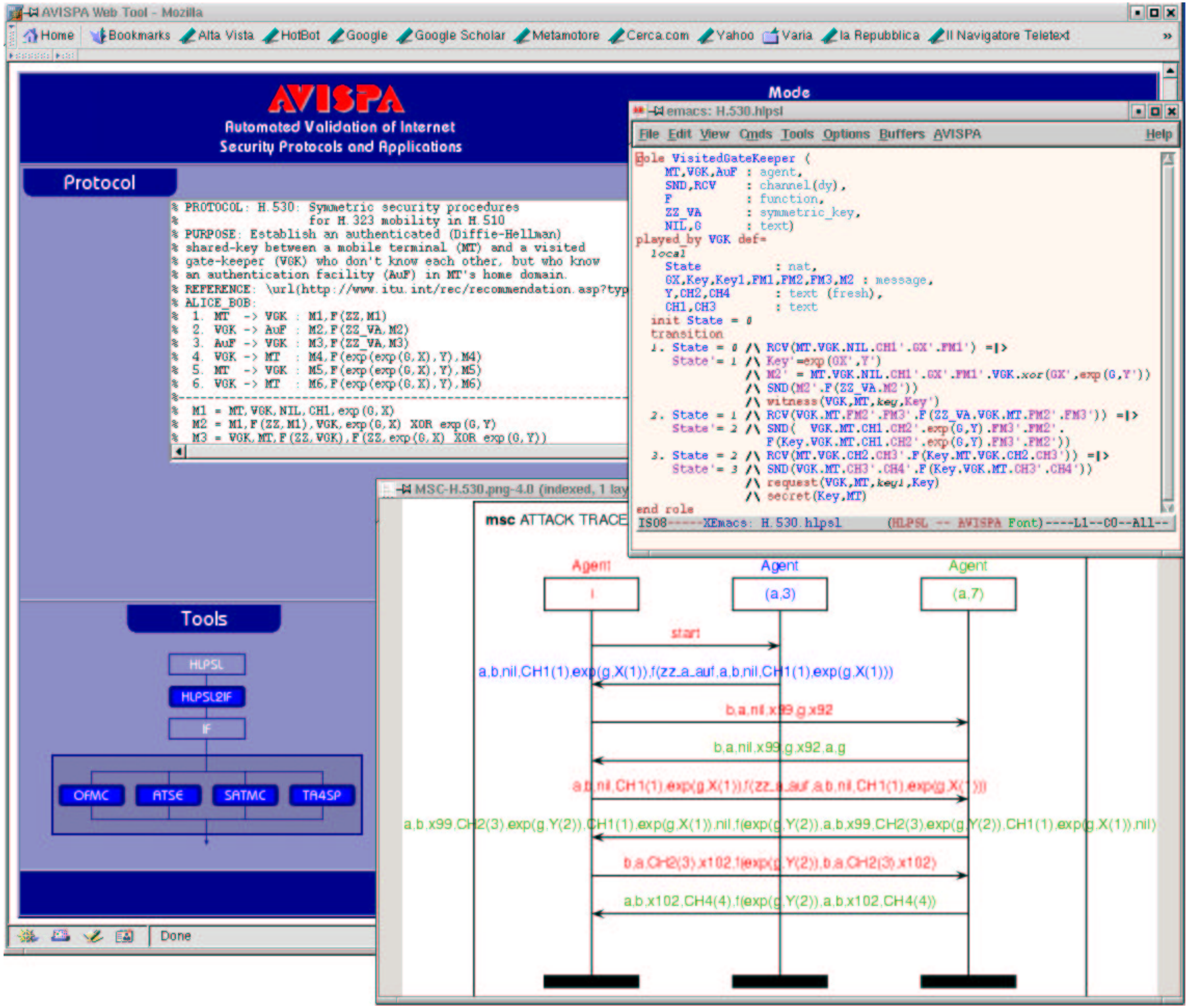
  - ▶ Most tools come with their own specification language and user interface.
  - ▶ Scaling up to large-scale Internet security protocols is a considerable scientific and technological challenge.



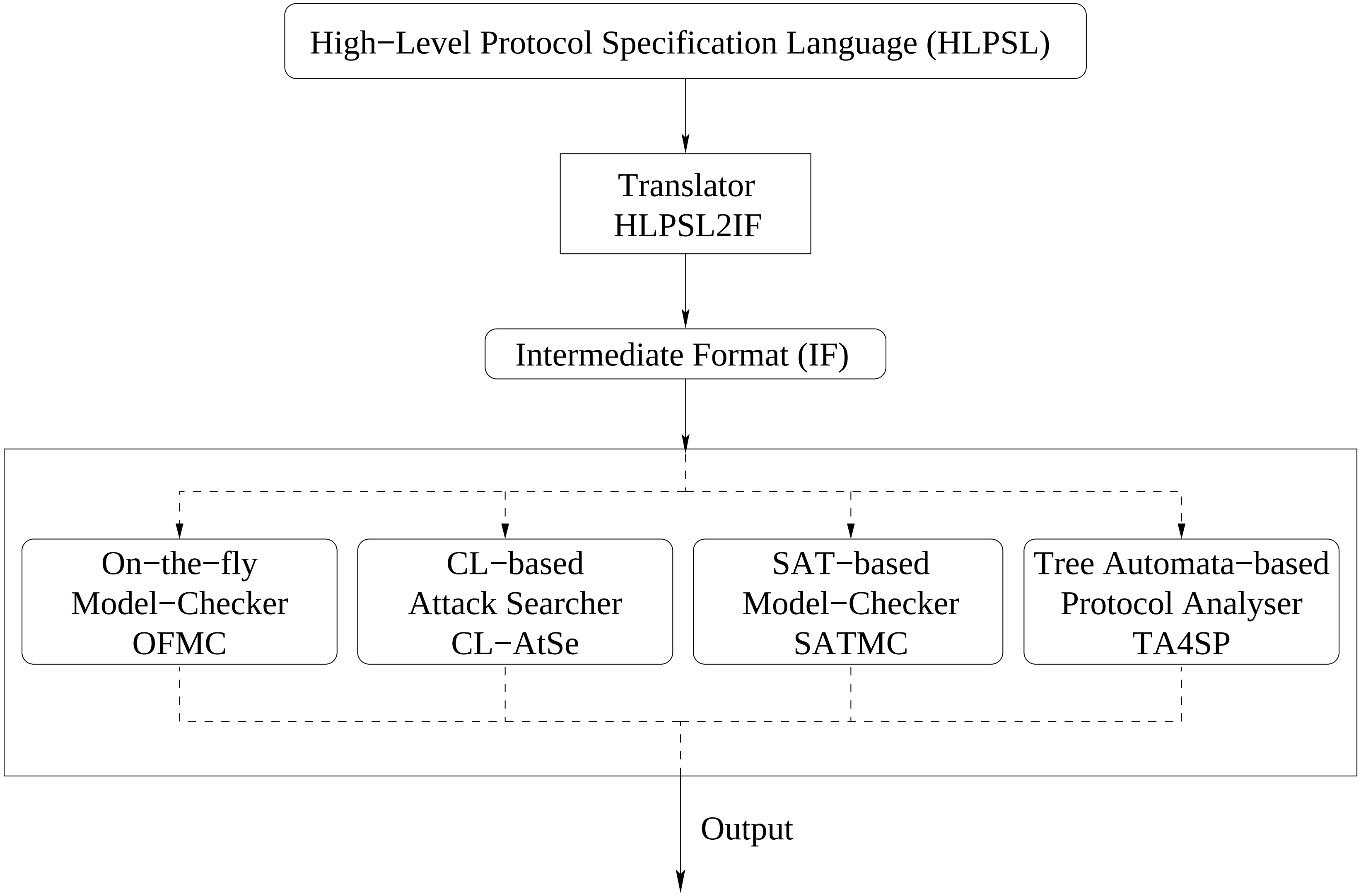
The AVISPA Tool

- Push-button security protocol analyzer.
- Supports the specification of security protocols and properties by means of a modular and expressive specification language.
- Integrates different back-ends implementing a variety of state-of-the-art automatic analysis techniques for
  - protocol falsification (by finding an attack on the input protocol)
  - abstraction-based verification methods

  both for finite and infinite numbers of sessions.
- User interaction facilitated by an emacs mode and a Web interface.


AVISPA Web Tool - Mozilla
Home
Bookmarks
Alta Vista
HotBot
Google
Google Scholar
Metamotore
Cerca.com
Yahoo
Varia
la Repubblica
Il Navigatore Teletext
AVISPA
Automated Validation of Internet
Security Protocols and Applications
Mode
Protocol
% PROTOCOL: H.530: Symmetric security procedures
%   for H.323 mobility in H.510
% PURPOSE: Establish an authenticated (Diffie-Hellman)
% shared-key between a mobile terminal (MT) and a visited
% gate-keeper (VGK) who don't know each other, but who know
% an authentication facility (AuF) in MT's home domain.
% REFERENCE: \url(http://www.itu.int/rec/recommendation.asp?typ
% ALICE_BOB:
% 1. MT -> VGK : M1,F(ZZ,M1)
% 2. VGK -> AuF : M2,F(ZZ_VA,M2)
% 3. AuF -> VGK : M3,F(ZZ_VA,M3)
% 4. VGK -> MT : M4,F(exp(exp(G,X),Y),M4)
% 5. MT -> VGK : M5,F(exp(exp(G,X),Y),M5)
% 6. VGK -> MT : M6,F(exp(exp(G,X),Y),M6)
% M1 = MT,VGK,NIL,CH1,exp(G,X)
% M2 = M1,F(ZZ,M1),VGK,exp(G,X) XOR exp(G,Y)
% M3 = VGK,MT,F(ZZ,VGK),F(ZZ,exp(G,X) XOR exp(G,Y))
emacs: H.530.hlpsl
File Edit View Cmds Tools Options Buffers AVISPA Help
role VisitedGateKeeper (
MT,VGK,AuF : agent,
SND,RCV : channel(dy),
F : function,
ZZ_VA : symmetric_key,
NIL,G : text)
played_by VGK def=
local
State : nat,
GX,Key,Key1,FM1,FM2,FM3,M2 : message,
Y,CH2,CH4 : text (fresh),
CH1,CH3 : text
init State = 0
transition
1. State = 0 /\ RCV(MT.VGK.NIL.CH1'.GX'.FM1') =|>
State'= 1 /\ Key'=exp(GX',Y')
/\ M2' = MT.VGK.NIL.CH1'.GX'.FM1'.VGK.xor(GX',exp(G,Y'))
/\ SND(M2'.F(ZZ_VA.M2'))
/\ witness(VGK,MT,key,Key')
2. State = 1 /\ RCV(VGK.MT.FM2'.FM3'.F(ZZ_VA.VGK.MT.FM2'.FM3')) =|>
State'= 2 /\ SND(VGK.MT.CH1.CH2'.exp(G,Y).FM3'.FM2'.
F(Key.VGK.MT.CH1.CH2'.exp(G,Y).FM3'.FM2'))
3. State = 2 /\ RCV(MT.VGK.CH2.CH3'.F(Key.MT.VGK.CH2.CH3')) =|>
State'= 3 /\ SND(VGK.MT.CH3'.CH4'.F(Key.VGK.MT.CH3'.CH4'))
/\ request(VGK,MT,key1,Key)
/\ secret(Key,MT)
end role
ISO8-----XEmacs: H.530.hlpsl (HLPSL -- AVISPA Font)----L1--C0--All--
MSC-H.530.png-4.0 (indexed, 1 lay
msc ATTACK TRACE
Agent
Agent
Agent
i
(a,3)
(a,7)
start
a,b,nil,CH1(1),exp(g,X(1)),f(zz_a_auf,a,b,nil,CH1(1),exp(g,X(1)))
b,a,nil,x99,g,x92
b,a,nil,x99,g,x92,a,g
a,b,nil,CH1(1),exp(g,X(1)),f(zz_a_auf,a,b,nil,CH1(1),exp(g,X(1)))
a,b,x99,CH2(3),exp(g,Y(2)),CH1(1),exp(g,X(1)),nil,f(exp(g,Y(2)),a,b,x99,CH2(3),exp(g,Y(2)),CH1(1),exp(g,X(1)),nil)
b,a,CH2(3),x102,f(exp(g,Y(2)),b,a,CH2(3),x102)
a,b,x102,CH4(4),f(exp(g,Y(2)),a,b,x102,CH4(4))
Tools
HLPSL
HLPSL2IF
IF
OFMC
ATSE
SATMC
TA4SP
Done

The AVISPA Tool: Architecture



High-Level Protocol Specification Language (HLPSL)

- Supports symmetric and asymmetric keys, non-atomic keys, key-tables, Diffie-Hellman key-agreement, hash functions, algebraic functions, typed and untyped data, etc.
- Security properties: different forms of authentication and secrecy.
- The intruder is modeled by the channel(s) over which the communication takes places:
  - Dolev-Yao intruder and (preliminarily) other intruder models.
- Role-based language:
  - a role for each (honest) agent,
  - parallel and sequential composition glue roles together.

HLPSL: Basic Roles

```
role NSPK-Initiator (A, B: agent, Ka, Kb: public_key,
                     SND, RCV: channel (dy))
 played_by A def=
   local State:nat, Na:text (fresh), Nb:text
   init State = 0
   transition
     1. State =0 /\ RCV(start)
        =|>
        State'=2 /\ SND({Na'.A}_Kb) /\ witness(A,B,na,Na')
     2. State =2 /\ RCV({Na.Nb'}_Ka)
        =|>
        State'=4 /\ SND({Nb'}_Kb) /\ request(A,B,nb,Nb')
                  /\ secret(Na,B)
end role
```

HLPSL: Parallel and Sequential Composition

```
role Kerberos (...)
 composition
  Client(...) /\
  Authn_Server(...) /\
  Server(...) /\
  TGS(...)
end role

role Alice (...)
 composition
  establish_TLS_Tunnel(server_authn_only);
  present_credentials;
  main_protocol(request, response)
end role
```

High-Level Protocol Specification Language (HLPSL)

- The HLPSL enjoys both
  - a declarative semantics based on a fragment of Lamport's Temporal Logic of Actions,
  - an operational semantics based on a translation into a rewrite-base formalism: the Intermediate Format (IF).
- This translation is automatically carried out by the HLPSL2IF translator.

The AVISPA Tool: The Back-Ends

Protocol falsification, and bounded and un-bounded verification.

The On-the-fly Model-Checker (OFMC) employs several symbolic techniques to explore the state space in a demand-driven way.

CL-AtSe (Constraint-Logic-based Attack Searcher) applies constraint solving with simplification heuristics and redundancy elimination techniques.

The SAT-based Model-Checker (SATMC) builds a propositional formula encoding all the possible attacks (of bounded length) on the protocol and feeds the result to a state-of-the-art SAT solver.

TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) approximates the intruder knowledge by using regular tree languages and rewriting to produce under and over approximations.

The AVISPA Library

- The AVISPA Library: HLPSL specifications of security problems associated with protocols that have recently been or are currently being standardized by the IETF.
- The AVISPA Library comprises 112 security problems derived from 33 protocols.
- AVISPA Tool assessed by running it against the AVISPA Library.

The AVISPA Tool: Results

Experimental Results (excerpt of)

Protocol	#P	OFMC P	OFMC A	OFMC T	CL-atse P	CL-atse A	CL-atse T	SATMC P	SATMC A	SATMC TE	SATMC 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
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
TSIG	2	2	1	0,04	2	1	0,00	2	1	0,12	0,01
DNSSEC	4	3	3	2,01	1	1	0,13	1	1	0,64	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-DSx	3	3	0	42,56	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	-	-

The AVISPA Tool: Results

- The experimental results show that:
  - Most problems are analysed in a few seconds
  - Back-ends exhibit complementary strengths
- Moreover, TA4SP establishes in a few minutes that a number of protocols (EKE, EKE2, IKEv2-CHILD, IKEv2-MAC, TLS, UMTS_AKA, CHAPv2) guarantee secrecy.

Conclusions

- The AVISPA Tool is a state-of-the-art, integrated environment for the automatic analysis and validation of Internet security protocols.
  - Try/download it at `www.avispa-project.org`.
- Current work:
  - Extending the AVISPA library with further protocols and properties.
  - Unbounded verification using abstractions.
  - Algebraic properties.
  - Guessing intruder and other intruder models (and channels).
  - Web-services.
- Integration of other tools via HLSPL/IF (e.g. translator from HLPSL to Applied Pi Calculus to then apply ProVerif).