A Tool helping to Design Cryptographic Protocols

Laurent Vigneron

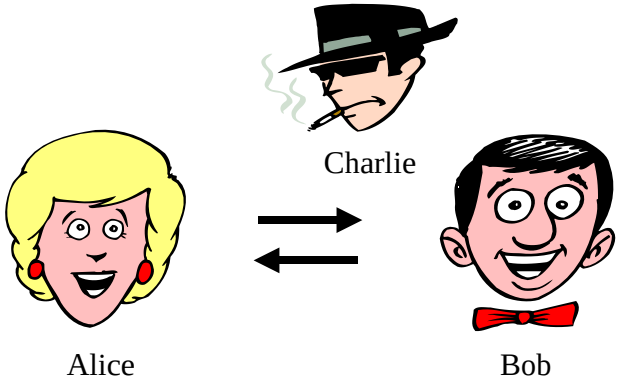
LORIA – Université Nancy 2 – CNRS

Università di Genova ETH Zurich Siemens AG

Automated Validation of Internet Security Protocols and Applications
Shared cost RTD (FET open) project IST-2001-39252

Motivation

- The world is distributed:
  - Our basic infrastructures are increasingly based on networked information systems.
  - Business, finance, communication, transportation, energy distribution, entertainment, . . .



Alice → Bob@Bank: "Transfer 100€ to account X"

Bob@Bank → Alice: "Transfer carried out"

  - How does Bob know that he is really speaking with Alice?
  - How does Bob know Alice just said it?
  - Confidentiality, integrity, accountability, non-repudiation, privacy, . . . ?

Motivation

- The world is distributed:
  - Our basic infrastructures are increasingly based on networked information systems.
  - Business, finance, communication, transportation, energy distribution, entertainment, . . .

- Protocols essential for developing networked services and new applications.
- Security errors in protocol design are costly.

  Money: security updates are costing hundreds of millions \$/€.
  Time: protocols are delayed by years.
  Confiance: eroding confidence in Internet Security and new applications.

Motivation

- The number and scale of new security protocols under development is out-pacing the human ability to rigorously analyse 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.

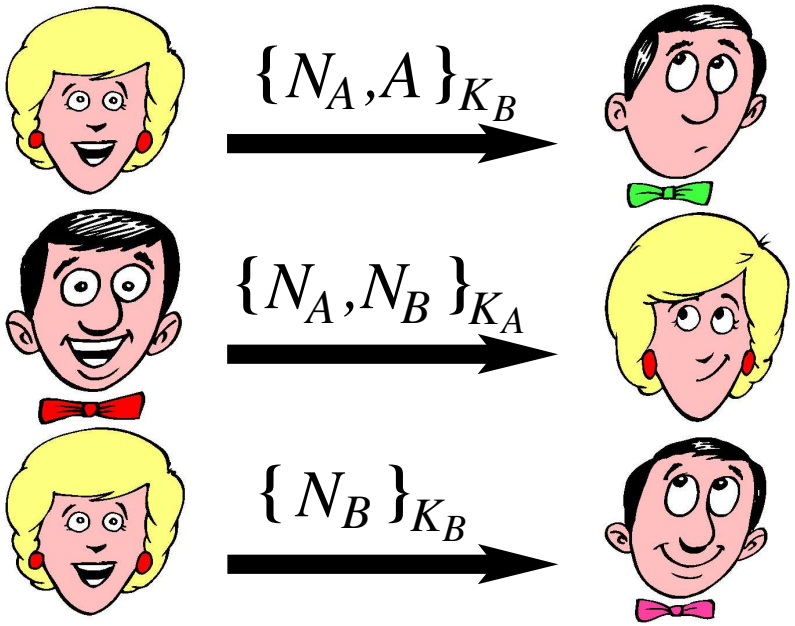
- Optimally, these tools should be completely automated, robust, expressive, and easily usable, so that they can be integrated into the protocol development and standardization processes.

The State of the Art. . .

- Several (semi-)automated protocol analyzers have been proposed,
  BUT automatic analysis limited to small and medium-scale protocols, e.g.:

  Mutual authentication protocol (NSPK):

  1. $A \rightarrow B : \quad \{N_A, A\}_{K_B}$
  2. $B \rightarrow A : \quad \{N_A, N_B\}_{K_A}$
  3. $A \rightarrow B : \quad \{N_B\}_{K_B}$



"This is Alice and I have chosen a nonce N_A."

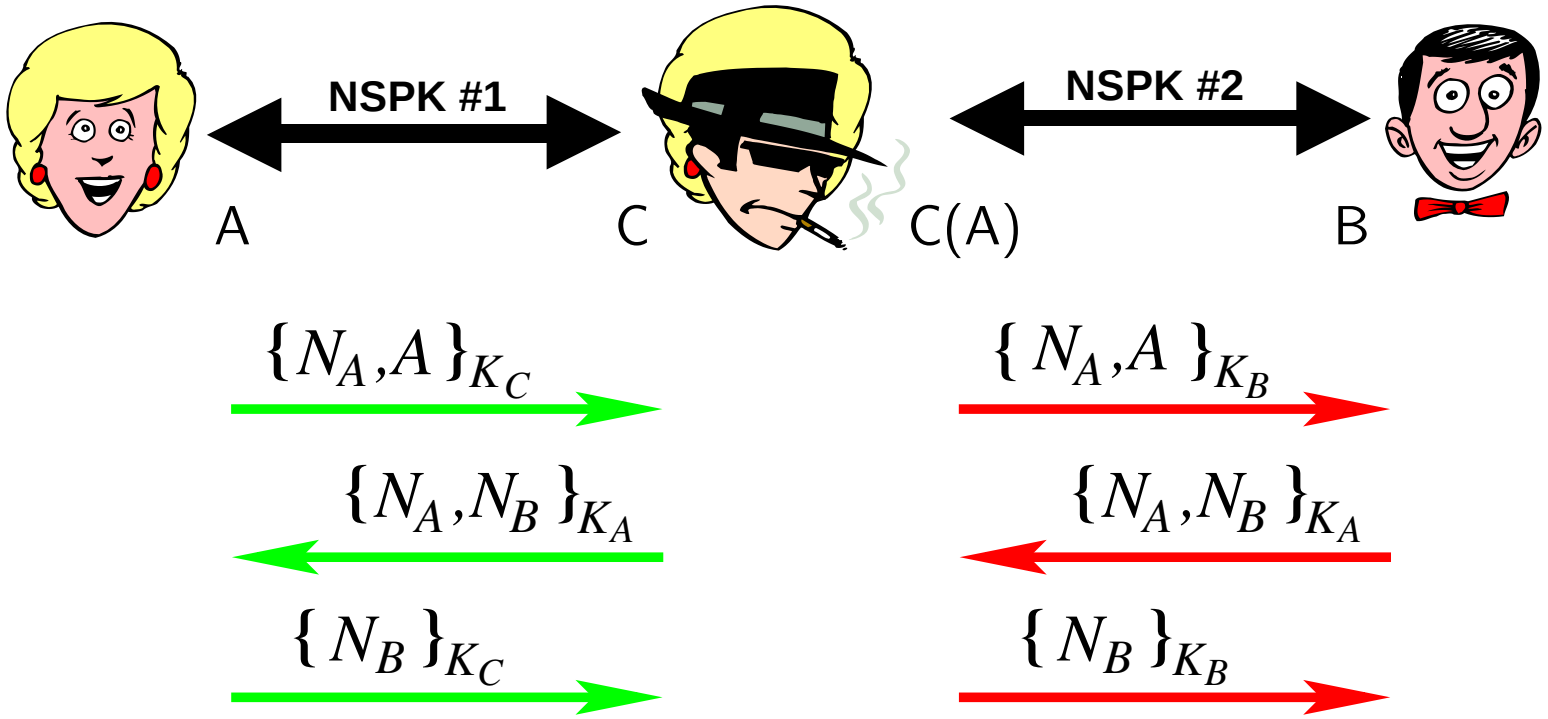
"Here is your nonce N_A. Since I could read it, I must be Bob. I also have a challenge N_B for you."

"You sent me N_B. Since only Alice can read this, and I sent it back, I must be Alice."

Protocols are typically small and convincing. . . but often wrong!

Man-in-the-Middle Attack

1. $A \rightarrow B: \quad \{N_A, A\}_{K_B}$
2. $B \rightarrow A: \quad \{N_A, N_B\}_{K_A}$
3. $A \rightarrow B: \quad \{N_B\}_{K_B}$



B believes he is speaking with A!

What went wrong?

- Problem in step 2: $B \rightarrow A : \{N_A, N_B\}_{K_A}$
  Agent B should also give his name: $\{N_A, N_B, B\}_{K_A}$

- Is this new version correct now?

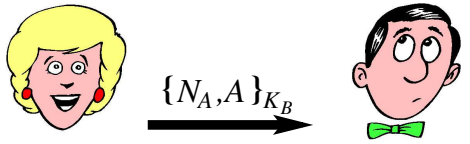
  . . . against this or other kinds of attacks?

  Use formal methods (and automated tools)!

The State of the Art. . .

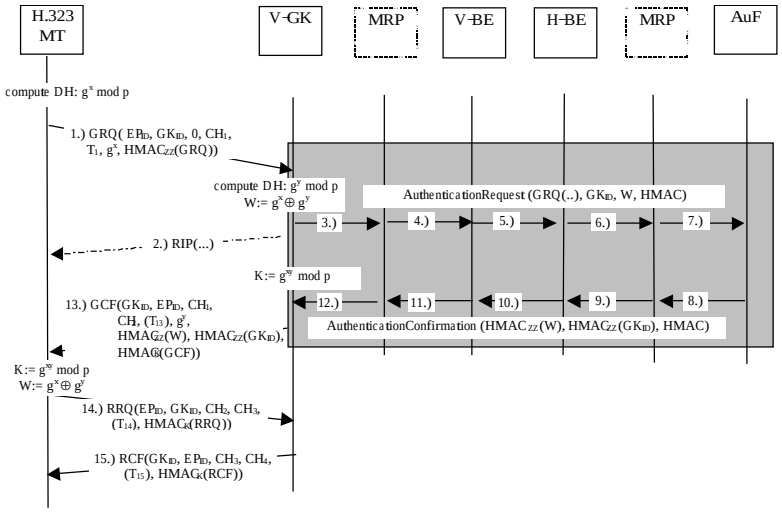
- Several (semi-)automated protocol analyzers have been proposed, BUT automatic analysis limited to small and medium-scale protocols.

  – 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.



...and beyond: the AVISPA Project

- A rich specification language for formalising industrial strength security protocols and thier properties.
- Advance state-of-the-art analysis techniques to scale up to this complexity.
- An integrated tool supporting the protocol designer in the debugging and validation of protocols via a uniform and user-friendly interface.
  $\Longrightarrow$ AVISPA Tool
- Tool assessed on a large collection of practically relevant, industrial protocols
  $\Longrightarrow$ AVISPA Library
- Migration of this technology to companies and standardisation organisations.

AVISPA Tool

- Push-button security protocol analyzer.
- Supports the specification of security protocols and properties by means of a rich protocol specification language.
- Integrates different back-ends implementing a variety of state-of-the-art automatic analysis techniques.
- User interaction facilitated by:
  - XEmacs mode.
  - Web interface.

A state-of-the-art (for level of scope and performance), integrated environment for the automatic analysis and validation of Internet security protocols.

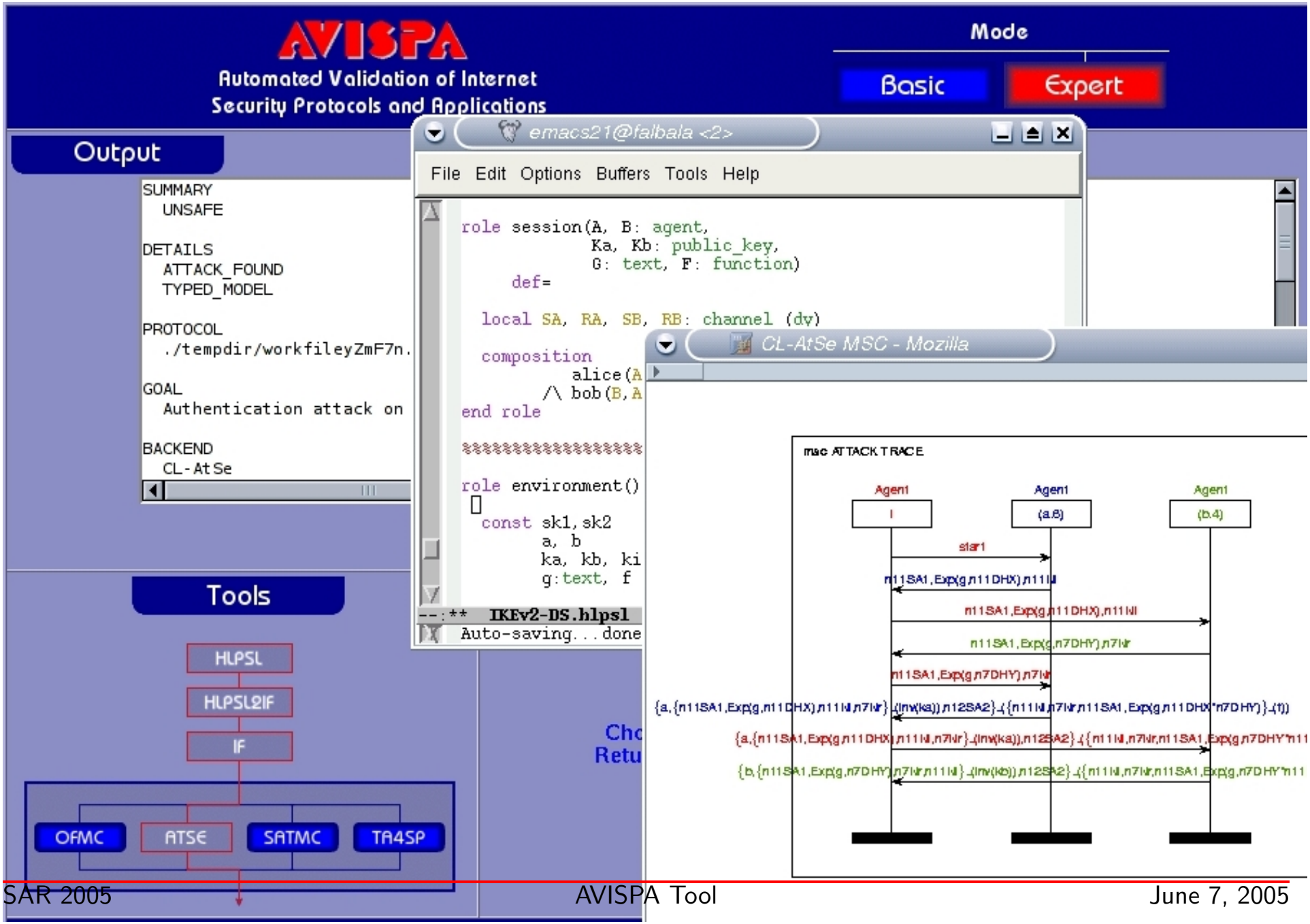
AVISPA Tool: Web Interface

- L'outil AVISPA est librement accessible à l'adresse:

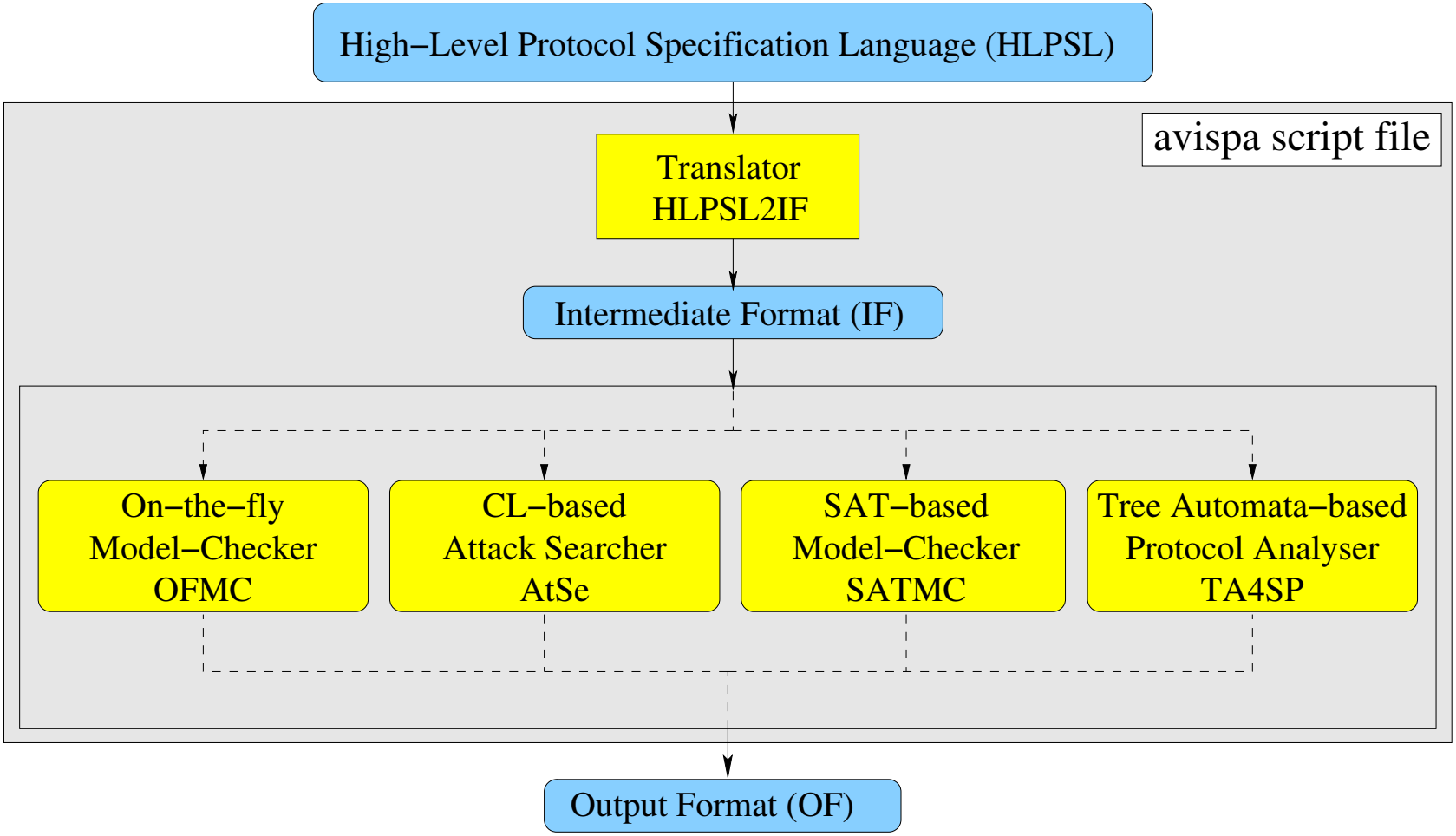
  http://www.avispa-project.org

- The interface features:
  - A simple editor for HLPSL specifications.
  - Basic/Expert user modes.
  - Attacks are graphically rendered with message-sequence charts.

just a quick look. . .


AVISPA
Automated Validation of Internet Security Protocols and Applications
Mode
Basic
Expert
Output
SUMMARY
UNSAFE
DETAILS
ATTACK_FOUND
TYPED_MODEL
PROTOCOL
./tempdir/workfileyZmF7n.
GOAL
Authentication attack on
BACKEND
CL-AtSe
emacs21@falbala <2>
File Edit Options Buffers Tools Help
role session(A, B: agent,
Ka, Kb: public_key,
G: text, F: function)
def=
local SA, RA, SB, RB: channel (dy)
composition
alice(A
/\ bob(B,A
end role
role environment()
const sk1, sk2
a, b
ka, kb, ki
g:text, f
--:** IKEv2-DS.hlpsl
Auto-saving...done
CL-AtSe MSC - Mozilla
msc ATTACK TRACE
Agent1
I
Agent1
(a,6)
Agent1
(b,4)
start
Tools
HLPSL
HLPSL2IF
IF
OFMC
ATSE
SATMC
TA4SP

AVISPA Tool: Architecture



AVISPA Tool: HLPSL

(High-Level Protocol Specification Language)

A powerful specification language:

- modular, role-based: basic roles (participants) and composed roles (sessions, instances);
- various cryptographic bases: symmetric keys (non-atomic), public/private keys, hash functions, nonces;
- typed information (or not): simple and compound types;
- algebraic properties: concatenation, exclusive-or, exponentiation;
- channels: for exchanging messages (Dolev-Yao);
- flow control: guarded transitions;
- studied properties: secrecy, weak and strong authentication.

Explicit semantics:

- a declarative semantics based on a fragment of Lamport's temporal logic of actions (TLA);
- an operational sémantics based on a translation into a rewriting-based formalism: the Intermediate Format (IF).

AVISPA Tool: Back-ends

Implemented methods: protocol falsification, bounded and unbounded verification.

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

AtSe: Constraint-Logic-based Attack Searcher
applies constraints solving with simplification heuristics and redundancy elimination techniques.

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

AVISPA Library

- Beyond Clark/Jacob (a few seconds for the entire library, with new attacks).
- Selection of a substantial set of security problems associated with protocols that have recently been or are currently being standardized by the IETF.
- Formalisation in HLPSL of a large subset of these protocols
  $\Longrightarrow$ AVISPA Library.
- At present the AVISPA Library comprises 112 security problems derived from 33 protocols.
- AVISPA Tool assessed by running it against the AVISPA Library.

Some protocols: AAAMobileIP, CHAPv2, CRAM-MD5, DHCP-delayed-auth, EKE2, IKEv2-CHILD, IKEv2-DS, IKEv2-MAC, ISO-9798, Kerb-Basic, Kerb-Cross-Realm, Kerb-PKinit, Kerb-Preauth, LPD-MSR, PBK, SRP, TLS, UMTS_AKA.

Also: TA4SP establishes in a few minutes that a number of protocols (EKE, EKE2, TLS, UMTS_AKA, CHAPv2) guarantee secrecy.

AVISPA Tool: How to use it?

Main steps:

Given, for example, a *RFC document*,

1. Write the messages exchange in an *Alice&Bob notation*,
2. Write this exchange in the *point of vue* of each participant,
3. Specify a *basic role* for each participant, with parameters, local variables, . . .
4. Specify *composition roles*: one role representing a session, one environment role representing the required instances,
5. Specify *properties* to check: goal section, plus goal predicates in some transitions of basic roles.

Run the AVISPA Tool: the translator will verify the syntax; the chosen back-end will verify the properties.

Conclusion and Future Work

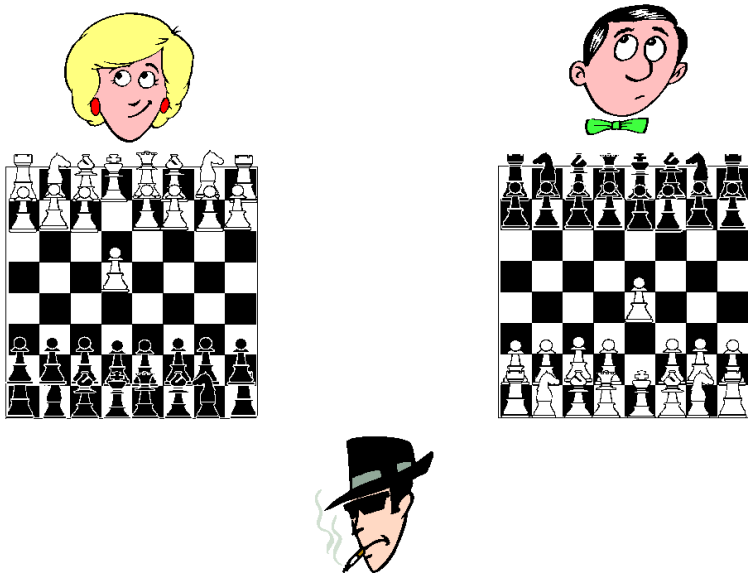
Bilan of the AVISPA Project:

- AVISPA Tool: a state-of-the-art, integrated environment, for automated analysis and validation of Internet security protocols.
- Information on `http://www.avispa-project.org`:
  - Web interface for demos;
  - AVISPA package v1.0: officially released in a couple of days!
  - Scientific papers, Slides,. . .

Future Work:

- Collaborations: Siemens AG (IETF), France Telecom.
  Large distribution of the AVISPA Tool $\Longrightarrow$ many future collaborations?
- Diversification of the activities: more properties (non-repudiation, fairness,. . .); more kinds of protocols (contributing protocols, web services, . . .).

Questions?



Demonstration on demand, or online:

`http://www.avispa-project.org`