

CASSIS

Combining ApproacheS for the Security of Infinite state Systems

Michael Rusinowitch
INRIA Lorraine

November 13, 2006



Team

Current permanent members:

Nancy

V. Cortier, S. Ranise,
C. Ringeissen, M. Rusinowitch,
M. Turuani, L. Vigneron.

Besançon

F. Bouquet, A. Giorgetti,
P.C. Héam, O. Kouchnarenko,

Former members:

F. Ambert, F. Bellegarde, G. Cécé, B. Legeard, F. Peureux

Non permanent members:

PhD students: 10 Postdocs: 2 Engineers: 2



Overall Objectives

Objective: Verification of security properties of systems with high or infinite number of states

Sources of infinity: program data, recursion, protocol topology, time, system parameters . . .

Target Applications: Embedded software, Cryptographic protocols, Distributed systems

Approach:

- Modeling systems with logic, formal languages, constraints and rewriting
- Encoding security properties as reachability ones
- Verification with automated deduction, regular model-checking and model-based testing



Research directions

Positioning: Push-button validation tools

- Automated Deduction - Decision Procedures
Harvey to discharge proof obligations generated by deductive verification (proof assistants, Why . . .)
- Cryptographic Protocol Verification
AVISPA to search for attacks or show their absence
- Symbolic and Constraint-based Verification
BZ-TT to validate implementations by model-based testing



Decision Procedures - Context

- Software verification often requires checking some formula is **satisfiable** in a **background theory** modeling data structures, memory model, or type system . . .
 - ➔ This is the **Satisfiability Modulo Theories** (**SMT**) problem
 - ➔ Examples of background theories: linear arithmetic, arrays, bit vectors, pointers, difference constraints, their combination as in

$$\text{store}(a, i, d)[j] \neq e \wedge f(i) = d \wedge i = j + k$$

- Standard deduction techniques do not work
 - ➔ **SMT solvers**: Simplify, CVC, STeP, MathSAT, ICS, Barcelogic tool, **haRVey**, Yices, Ario, CVC-Lite, ...



Our Approach to Solving SMT problems

- **Scalability**: integration of
 - SAT technology: to reason on Boolean structure
 - decision procedures: to reason on background theories
- **Expressiveness** (*originality of our approach!*): joint use of
 - **uniform methodology** to develop decision procedures for interesting data-structures
 - **combination** methods to re-use other existing procedures (e.g., fragments of Arithmetics)



Uniform Methodology to Design Satisfiability Procedures

- produces **rewriting-based satisfiability procedures**
- relies on **superposition** calculus \approx resolution + rewriting
- simple correctness argument
- **direct implementation from existing provers**
- the methodology
 - works for many data structures, e.g. lists, arrays, records
 - does not work for (decidable fragments of) Arithmetics!
 \implies **Need of combining black-box decision procedures!**



Combination of Theories

- Satisfiability problem for unions of (disjoint) theories
 - combining 2 rewriting-based procedures [AB+05]
example: array + list
 - combining 1 rewriting-based and 1 black-box
example: array + arithmetics
Efficient and automatic check of Nelson-Oppen conditions
for combination [KR+05/06]
- New combination methods (beyond Nelson-Oppen) for useful theories, e.g. **enumerated data types** [BG+06] or **container data structures** storing arbitrary elements [RRZ05]



haRvey: a cocktail of theories

The logo for haRvey, featuring the word "haRvey" in a stylized, blue, italicized font with a thin blue line above the letters.

developed in collaboration with D. Déharbe (U. Natal, Brasil) and P. Fontaine (MOSEL project)

- Integration of a SAT solver (MiniSAT) and a combination of a superposition-based prover (E prover) with a decision procedure for Linear Arithmetic (home made)
- Applications:
 - Debugging/verification of units of C code [DR03]
 - Verification of an industrial smart-card [CD+03]. Outperforms Atelier B.
 - Certification of auto-generated code from NASA applications [KR+05]



Formal Approaches to Security Protocol Verification

- Messages are abstracted using terms
These terms are built over a fixed signature
E.g., $\Sigma = \{< >, \text{enc}, \text{dec}, \dots\}$
- The attacker can do **symbolic manipulations on terms**.

$$\frac{S \vdash \text{enc}(M, \text{key}) \quad S \vdash \text{key}^{-1}}{S \vdash M} \qquad \frac{S \vdash \langle M_1, M_2 \rangle}{S \vdash M_2}$$

This approach allows to detect any **logical attack** that does not rely on weaknesses of the encryption algorithm.



Protocol Description

Protocol:

$$\begin{array}{l} \textit{Terminal} \rightarrow \textit{Card} : \textit{rand} \\ \textit{Card} \rightarrow \textit{Terminal} : \textit{enc}(\textit{rand}, \textit{key}) \end{array} \qquad \frac{S \vdash x}{S \vdash \textit{enc}(x, \textit{key})}$$

Secrecy properties:

$$S \vdash s?$$

known: undecidable in general and
NP complete for finite number of sessions

Too abstract model: misses many attacks!



Adding Cryptographic Primitives - Results

Cryptographic primitives have algebraic properties modeled using **equational theories** and by extending the intruder power.

Secrecy is decidable for interesting protocol families using XOR [CS06] or blind signatures [CRZ05].

Secrecy remains NP-complete for finite number of sessions with XOR [CK+03]
Abelian groups + modular exponentiation [CK+03]

General combination result : If secrecy is decidable for theories on disjoint signatures then it is decidable for the union of these theories [CR05].



Relating Formal and Cryptographic Approaches

	Formal approach	Cryptographic approach
Messages	terms	bitstrings
Encryption	idealized	algorithm
Adversary	idealized	any polynomial algorithm
Proof	automatic	by hand, tedious and error-prone

Link between the two approaches ?



A Combination Result

Theorem (CW05)

- for protocols with only public key encryption and signatures
- if a protocol is secure in the formal approach (proof given by a tool for example),
- if the public key encryption algorithm is IND-CCA2,

then the protocol is secure in the cryptographic approach.

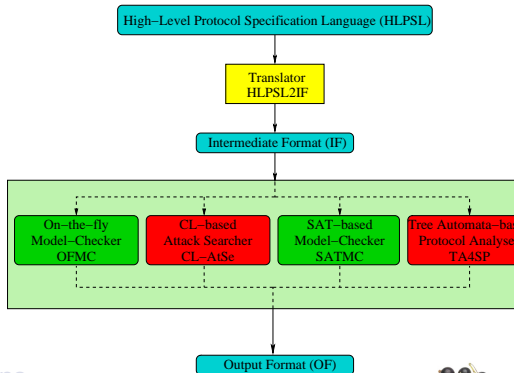
Extended to hash [CKKW06]



AVISPA Tool - U.Genova, ETH Zurich, INRIA, Siemens

Development of the AVISPA Tool, **push-button** security protocol analyser featuring:

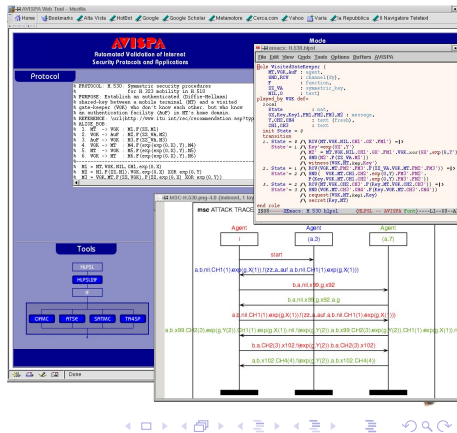
- an automatic translator from HLPSSL to IF
- four back-ends two provided by Cassis:
 - falsification: **CL-AtSe**
 - verification: **TA4SP**
- two interaction modes:
 - advanced editor for HLPSSL specifications
 - web interface



AVISPA Tool - U.Genova, ETH Zurich, INRIA, Siemens

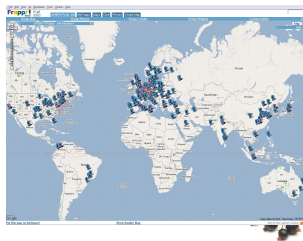
Development of the AVISPA Tool, push-button security protocol analyser featuring:

- an automatic translator from HLPSL to IF
- four back-ends two provided by Cassis:
 - falsification: **CL-AtSe**
 - verification: **TA4SP**
- two interaction modes:
 - advanced editor for HLPSL specifications
 - web interface



Impact

- The AVISPA Tool is engineered for usability:
 - Assessed on large number of protocols (> 79)
 - GUI, comprehensive documentation, user mailing list
- Dissemination in industry forums: IETF
- Used in many security courses, and active interest from SAP, Telefonica, France Telecom R&D, Thomson, Gemalto, . . . (RNTL PROUVE with SECSI)



CL-Atse: Constraint-based attack searching

- finite sessions case
- secrecy, authentication, non repudiation
- handle **xor**, **exponential**, type flaws
- pre-processing, chaining steps, partial instantiation for reducing non-determinism
- fully automatic

Protocol	Diagnosis	Time (s)
ASW - Abort	Secrecy flaw	0.03
DNSsec	Auth. flaw	0.98
EAP with IKEv2 Exp.	Safe	2.26
EKE Exp.	Auth. flaw	0.01
Fair Zhou-Gollmann	Auth. flaw	0.20
Fair Zhou-Gollmann (patch)	Safe	4.99
IKEv2 with MAC Exp.	Safe	7.38
IKEv2 with DS Exp.	Auth. flaw	0.03
Kerberos, cross-realm	Safe	0.32
Kerberos, forwardable ticket	Safe	0.14
Purpose Built Keys	Auth. flaw	0.01
Next Steps In Signaling	Safe	2.61
SET - Purchase Request	Secrecy flaw	0.17
SPEKE, with strong pwd. Exp.	Safe	0.04
SSH Exp.	Safe	1.22
TSIG	Safe	0.08



TA4SP: Tree Automata based on Automatic Approximations for the Analysis of Security Protocols.

- uses Timbuk tree automata library (developed by LANDE)
- unbounded number of sessions
- **fully automatic**, answers in few seconds or minutes
- handles algebraic operators (xor)
- produces attack traces

Protocol	Time (s)	Diagnosis
NSPKL	1.45	SAFE
NSPK	4.81	FLAWED
RSA	5.93	FLAWED
NSSK	115.34	SAFE
Denning-Sacco	8.82	SAFE
Yahalom	97.68	SAFE
Andrew Secure RPC	23.97	SAFE
Wide Mouthed Frog	7.20	SAFE
Kaochow v1	209.60	SAFE
Kaochow v2	353.91	??
TMN	8.02	FLAWED
Neumann	66.45	SAFE
AAA Mobile IP	754.19	SAFE
UMT-AKA	0.55	SAFE
CHAPv2	16.46	SAFE
CRAM-MD5	0.37	SAFE
DHCP-Delayed-Auth	6.84	SAFE
EKE	2.87	SAFE
LPD-IMSR	3.25	SAFE
LPD-MSR	0.61	FLAWED
TSIG	4.46	SAFE
SHARE	1.90	SAFE
ViewOnly (xor)	6276.00	SAFE

BZ-Testing-Tools

Based on a customized set-oriented constraint solver and applied to

- Smart Cards (GSM 11.11, Java Card transaction mechanism, key management 2G/3G)
- Urban systems (EMV payment, Transport ticket, Parkeon)
- Automotive software (wiper controller, lightings, PSA)
- JML Animator and Test generator

Spin-off: **Leirios Technologies**

Founded in 2003 by B. Legoard

Smart Testing from B Model

September 1st 2006: 26 employees



Decision Procedures

Model checking infinite state systems : integration of LTL model checking algorithms and decision procedures for data structures (application to software model-checking)

Heap allocated data structures :

- incorporate local reasoning *à la* Separation Logic
- generalize to arbitrary recursive data structures, such as trees, DAGs ➡ **ARA ARROWS**

New combination methods : the rewriting approach as an automatic proof technique for checking combinability conditions in extensions of the Nelson-Oppen method



Protocol Verification

Composition of protocols

- Verification of composed protocols
- Secure protocol design

New classes of protocols

- Group protocols: new models and security properties (e.g. time-sensitive) [BC+06]
- Cryptographic API [CS06]

Computational analysis

- Weaker cryptographic assumptions
- Contract-signing protocols, ➔ **ARA FORMACRYPT**



Model-Based Testing

- Incorporate security requirements in test generation process - Analysis of IAS 2.0 (european citizen id card) and GlobalPlatform standards (RNTL POSE)
- Collaboration test generation / theorem proving



Verification of Web Services

Many security challenges, e.g.

Secure messaging and authentication : specific vulnerabilities of XML protocols

Composition of web services : integration of temporal constraints and security policies into existing formal models -

➔ Application of constraint solving and decision procedures: composition viewed as a satisfiability problem.

➔ Validation by regular approximations and model-based testing.

➔ **ARA COPS**

