AVISPA

www.avispa-project.org

IST-2001-39252

Automated Validation of Internet Security Protocols and Applications

Deliverable D1.4: Final Project Report

Covering period 01.01.2003 — 30.06.2005

Abstract

This report provides a comprehensive view of the results obtained, of the methodologies and approaches employed, and of changes in the state-of-the-art since the project was contracted, and elaborates on the degree to which the objectives have been reached.

Deliverable details

Deliverable version: *1.1*	Person-months required: *1*
Date of delivery: *13.10.2005*	Due on: *30.08.2005*
Classification: *public*	Total pages: *41*

Project details

Start date: *January 1st, 2003*
Duration: *30 months*
Project Coordinator: *Alessandro Armando*
Partners: *Università di Genova, INRIA Lorraine, ETH Zürich, Siemens AG*

Information Society Technologies

Project funded by the European Community under the *Information Society Technologies* Programme (1998-2002)

[This page has been intentionally left blank.]

Contents

1 Executive Summary 3

2 Project Objectives 8

3 Methodologies 8
3.1 Design and Development of Specification Languages for Security Protocols 8
3.2 Protocol Selection 10
3.3 Modelling and Specification of Security Protocols 11
3.4 Automatic Analysis Techniques for Security Protocols 11
3.5 Tool Development 12
3.6 Tool Assessment 14

4 Project Results and Achievements 15
4.1 Comparison to the original project objectives 17
4.2 Relations and synergies with other relevant projects 19
4.2.1 The Project PROUVE 19
4.2.2 ECSS Group, Eindhoven 19
4.2.3 Blanchet's Logic Programming Approach 20
4.2.4 The Project DEGAS 20
4.2.5 The project MyThS 21
4.3 Implications for EU policies and standards 21
4.4 Benefits to society 22

5 Deliverables and Publications 22
5.1 Deliverables 22
5.2 Publications and Conference Presentations 29
5.3 Tutorials 33
5.4 Invited Talks 34
5.5 International Workshops Organisations 35
5.6 Editorial Activities 36

6 Potential Impact of Project Results 36

7 Future Outlook 38

8 Conclusions 38

1 Executive Summary

AVISPA is a FET Open Project that has developed a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. This technology will speed up the development of the next generation of network protocols, improving their security, and therefore increasing the public acceptance of advanced, distributed IT applications based on them.

The partners of the project are:

1. Università di Genova (UNIGE), Italy (project coordinator),
2. INRIA Lorraine, France,
3. ETH Zürich (ETHZ), Switzerland, and
4. Siemens AG, Munich, Germany.

The project activity was divided in 8 work packages. The project objectives and milestones are detailed in the Technical Annex, and can be summarised as follows:

WP1 – Protocol Management To manage the project and all of its objectives and milestones (including, in particular, the organisation of the project meetings and the production of the project reports and other required documents).

WP2 – Protocol Specification Languages To define the High-Level Protocol Specification Language HLPSL capable of supporting the specification of security-sensitive, state-of-the-art Internet protocols. To design and develop a translator from the high-level language to the rewrite-based declarative Intermediate Format IF amenable to formal analysis. To provide a graphical user interface that supports the editing, specification and push-button validation of protocols with the AVISPA Tool.

WP3 – Context & Properties Specification To build constructs for expressing sophisticated security goals and assumptions about the environment into both the high-level and the intermediate specification languages.

WP4 – Scalability To improve the automated deduction techniques and prototype tools previously developed by the partners and scale them up to large-scale, state-of-the-art security protocols such as those selected in WP6.

WP5 – Verification To investigate and integrate mechanisms to derive positive statements about protocol security, i.e. verify that they achieve their security objectives.

WP6 – Selection & Specification of Protocols To define the AVISPA selection and library, a set of formalised security problems (i.e., protocols and security properties) drawn from Internet protocols that have recently been standardised or are currently undergoing standardisation.

WP7 – Tool Assessment To evaluate the technical achievements of the project with respect to measurable criteria. Classes of protocols, threat models, and security goals for which each automated deduction technique behaves optimally will be also identified.

WP8 – Dissemination To disseminate the project results through appropriate channels and in appropriate forums.

All the expected results have been achieved, all success criteria set out in the Technical Annex have been met, and all planned deliverables have been produced on time.

The results of the scientific work packages WP2—WP7 consist of a series of deliverables, whose purposes and contents are detailed in the next sections. We here summarise the main project achievements:

WP2&3 We have formalised the High-Level Protocol Specification Language HLPSL and the Intermediate Format IF, and have implemented the automated translator HLPSL2IF from HLPSL to IF. Both the HLPSL and the IF are more expressive than other specification languages used for the same purpose. The HLPSL is a very expressive language supporting the specification of security-sensitive protocols with a formal semantics based on an expressive first-order temporal logic. The IF is a tool-independent, low-level protocol specification language that supports the specification of sophisticated typed protocol models and that is suitable for automated deduction. Specifications of security protocols and properties written in HLPSL are automatically translated into IF specifications, which are then given as input to the different back-ends that constitute the AVISPA Tool. We have also devised and implemented a number of advanced techniques and optimisations that allow users of the AVISPA Tool to formally specify complex protocol analysis contexts, environments, and properties.

WP4&5 We have devised a number of heuristics, optimisations, and reduction and abstraction techniques, both general and specific to the individual back-ends. Moreover, we have introduced a verification algorithm for time-sensitive security protocols, and we have investigated the completeness of protocol validation procedures and the compositionality of protocols, obtaining a number of results on the composition of intruder theories, of different protocols, and of different communication channels.

These protocol analysis techniques are implemented in the 4 back-ends of the AVISPA Tool:

OFMC, an on-the-fly model-checker developed and maintained by ETHZ,

CL-AtSe, a protocol analyser based on Constraint Logic developed and maintained by INRIA,

SATMC, a SAT-based model-checker developed and maintained by UNIGE, and

TA4SP, a tree automata based protocol analyser developed and maintained by the LIFC group affiliated with INRIA.

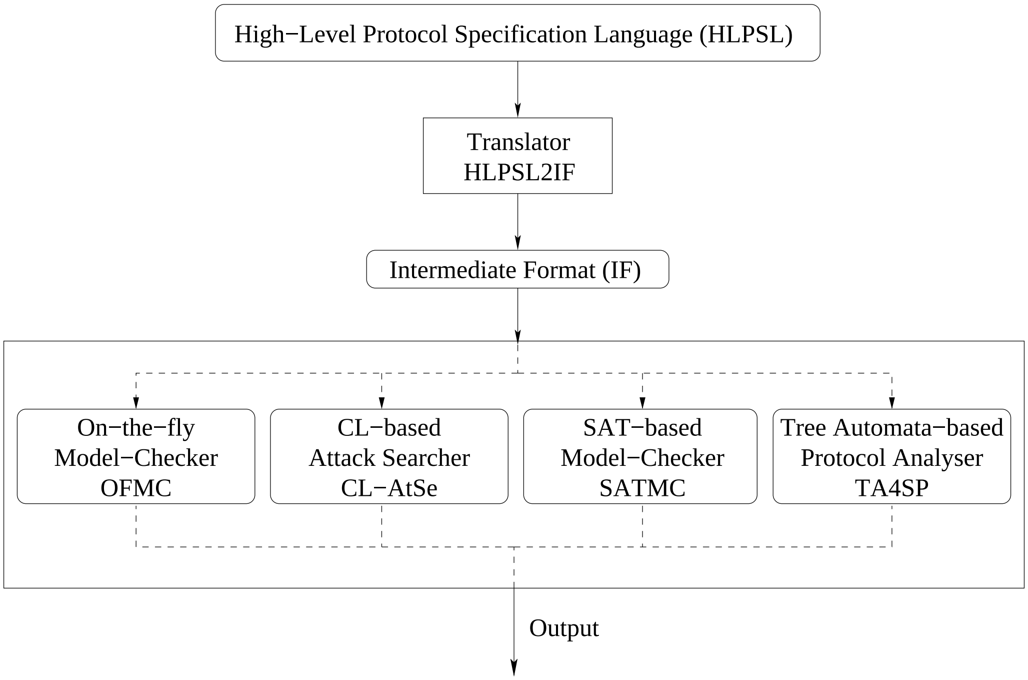


Figure 1: Architecture of the AVISPA Tool

The resulting architecture of the AVISPA Tool is depicted in Figure 1.

WP6&7 In order to assess the strength of the back-ends of the AVISPA Tool, and to demonstrate proof-of-concept on a large collection of practically relevant, industrial protocols, we have selected a set of candidate protocols currently being drafted by the IETF, along with the security properties these protocols are expected to enjoy. We have thus identified a set of security problems, where a problem is given by both a protocol and a security property the protocol should satisfy. This set, which we call the *AVISPA Selection*, contains a total of 384 security problems and 79 protocols, mostly from the IETF, divided into 33 groups.

We have used the AVISPA Selection as the basis for the success criteria for the project. More specifically, the following criteria, which refine the ones given in the Technical Annex, are used for the final assessment of the AVISPA tool at month 30 (i.e. the end of the project):

Coverage: at least 80 security problems from 20 groups of the AVISPA Selection should be specifiable in the HLPSL.

Effectiveness: the AVISPA Tool should successfully analyse at least 75% (i.e. 60) of these 80 problems, including at least one problem from each of the first seven groups given in [57], by either verifying that the protocol satisfies the desired security property (mainly for scenarios consisting of a bounded number of protocol sessions) or by finding a counterexample demonstrating that the

Table 1: Results of the AVISPA Tool at the end of the project

Success criteria at month 30	**Objectives**	**Results**
Coverage	80 problems from 20 groups	215 problems from 22 groups
Effectiveness	60 problems	215 problems
Performance	< 1 hour per problem	< 24 minutes per problem (all 215 problems in 87 minutes)

property is violated.

Performance: the verification of each problem should be carried out in less than 1 hour of CPU time.

The results demonstrate the success of the project. As summarised in Table 1, we have been able to formalise in the HLPSL 215 problems from 22 groups (including 38 problems from the seven main groups described in [57]), and the tool successfully analyses all of them in less than 24 minutes of CPU time per problem (globally, the entire library of 215 problems requires 87 minutes of CPU time to be analysed). All the above requirements (namely coverage, effectiveness, and performance) are therefore more than fulfilled.

The activities for the Project Management work package (WP1) included *(i)* the supervision of the technical activity and of the production of the deliverables, *(ii)* the organisation of project meetings (restricted to the AVISPA personnel), and *(iii)* the writing of the different *Reports* as well as the production of a *Project Presentation*, of a *Dissemination and Use Plan*, and of the *Consortium Agreement*. All these objectives have been realised successfully. 11 project meetings with a large number of attendees from all project partners have been held, and a large number of exchange visits took place to address technical issues. Project work proceeded in compliance with the plan set out in the Technical Annex, meeting all the success criteria.

The activities for the Dissemination work package (WP8) included the creation and management of the AVISPA Web-Site (`www.avispa-project.org`), the organisation of the project workshops and tutorials, the presentation of the project's achievements at conferences and in invited talks, and the preparation of the *D8.7 - Technology and Implementation Plan*. All these objectives have been realised successfully.

Dissemination has followed standard scientific channels: 47 papers have been published in international conferences and journals; 2 PhD theses are about to be completed and other 4 PhD students have developed a significant part of their thesis in the context of the project; 4 international workshops have been organised, and a large number of invited talks, presentations, and tutorials were given in the context of major scientific events. Additionally, we have actively sought for contacts and exchanges of ideas with representatives of research projects on related themes that are currently being carried out at the national, EU, or international level.

We have also initiated an extremely fruitful dialogue between AVISPA and the Internet Engineering Task Force (IETF). This is particularly important as the large collection of practically relevant, security-sensitive, industrial protocols that AVISPA has been studying are mostly being standardised by the IETF. The list of chosen candidate protocols and related problems, as described in Deliverable 6.1 [57], has been made available to the IETF and discussed with the security area directors. We have also presented the results of the Project at the Open Security Area Directorate meetings held in the context of the 59th and the 62nd meetings of the IETF.

Summarising, we have met all the objectives that we had set out at the beginning of the project and satisfied all success criteria. The work we have carried out has led the foundations of a push-button technology, based on automated deduction, for validating security-sensitive protocols like those used in electronic commerce, telecommunications, multi-media, and other application areas. We believe that this technology will pave the way to the construction of industrial-strength protocol validation tools that will reduce time-to-market and increase trust in the security of applications, thereby improving the competitiveness of European companies working in these application areas.

2 Project Objectives

The project objectives are described in detail in Section 2 of the Technical Annex. So here we simply quote the relevant text:

> The overall goals of the AVISPA project are
>
> **(O1)** to develop a rich specification language for formalising protocols, security goals, and threat models of industrial complexity,
>
> **(O2)** to advance the state-of-the-art in automated deduction techniques to scale up to this complexity,
>
> **(O3)** to build a tool based on these techniques that will allow industry and standardisation organisations to automatically validate or detect errors in their products,
>
> **(O4)** to tune this tool and demonstrate proof-of-concept on a large collection of practically relevant, industrial protocols, and
>
> **(O5)** to begin the migration of this technology into industry standardisation organisations such as the IETF so that both the scientific and the industrial community can benefit from the advances achieved by this project.
>
> More technically, the AVISPA project aims to design a push-button technology, based on automated deduction, for validating security-sensitive protocols like those used in electronic commerce, telecommunications, multi-media, and other applications. This technology will pave the way to the construction of industrial-strength protocol validation tools that will reduce time-to-market and increase trust in the security of applications, thereby improving the competitiveness of European companies working in these application areas.

3 Methodologies

3.1 Design and Development of Specification Languages for Security Protocols

The solution we adopted to cope with the contrasting requirements of getting specifications (1) writable efficiently by engineers and (2) amenable to formal analysis was to design two specification languages, a high-level one and a low level one.

AVISPA has designed an expressive high-level protocol specification language HLPSL [51, 24], which provides the basis for a formal description of a large class of industrial-scale protocols, their security properties, and assumptions on their environment and threat models.

The HLPSL [51, 24] provides the basis for a formal description of a large class of industrial-scale protocols, their security properties, and assumptions on their environment and threat models.

HLPSL draws its semantic roots from Lamport's Temporal Logic of Actions (TLA, [73]). TLA is an elegant and powerful language which lends itself well to specifying concurrent systems (see, e.g., [74]) precisely like the types of protocols we seek to model here. Syntactically, however, specifying protocols in a raw logic can be a daunting task. Moreover, the domain of protocol analysis calls for several syntactic constructs (such as message structure) and semantic concepts (like the notion of an intruder) that are problem-independent and arise in every model. Ideally, it would be convenient to model protocols in a language which offers such commonalities built-in. The development of HLPSL was thus undertaken with the following design objectives:

- It must provide a convenient, human-readable, and easy to use language, which is however powerful enough to support the specification of modern Internet protocols. To this end, HLPSL has been defined in such a way as to closely resemble a language for defining guarded transitions within a state-transition system and is equipped with constructs that allow for the modular specification of protocols.

- It must enjoy a formal semantics. To this end, HLPSL has been based on Lamport's TLA and its semantics is given by a translation to a subset of TLA.

- It must be amenable to automated formal analysis. This is achieved by a translation of HLPSL into the Intermediate Format (IF [54]). The main goal in the design of the IF was to provide a low-level description of the protocol that is suitable for automatic analysis (rather than being abstract and easy to read for human users like the HLPSL), and yet this format should be independent from the analysis methods employed by the various back-ends. The IF describes a protocol in terms of rewrite rules describing an infinite-state transition system with an initial state, transition rules, and a state-based safety property, namely a goal (attack) predicate that defines if a given state is an attack state or not.[1]

A number of alternative approaches to the specification of security protocols have been proposed recently. For instance, different protocol analysis tools (e.g. Casper, ProVerif, Scyther) take as input protocol specifications written using process algebraic formalisms such as CSP, the spi calculus, or the applied pi calculus, e.g. [49, 61, 67, 69, 75, 76]. Our work has several advantages with respect to these approaches. First, our HLPSL is more expressive than many of these languages, which would thus require extensions order to formally model the protocols that we have been considering in the AVISPA project. Second, AVISPA also currently has a broader application scope than the techniques and tools based on these formalisms, as they have only been tested on a limited number of small to medium scale protocols. In other words, there are no tools that have been applied to a library of problems of the breadth and complexity as done for the AVISPA Tool. Moreover, there are no process algebraic tools that effectively combine advanced technologies for both verification and flaw detection. Only the CAPSL [68] environment, developed at SRI International (Menlo Park, CA) in the context of a DARPA-project, is designed to support

[1] An *attack trace* is a path that leads from the initial state to an attack state.

multiple analysis techniques. The CAPSL environment and the AVISPA Tool share the idea of using both a high-level specification language close to the language used in textbooks and by engineers and a low-level language that provides an interface to different back-ends for protocol analysis. Based on the available experiments and the results in [68], our prototype is considerably more effective than CAPSL and its current connectors on the protocols in the Clark/Jacob library [66] (and on those few protocols of the AVISPA Library that have also been considered in the CAPSL project). For example, CAPSL cannot handle protocols where a principal receives a message that he cannot decrypt immediately. In our case, the principal will store it and decrypt it when he later receives the key. This feature is necessary for the analysis of non-repudiation protocols.

3.2 Protocol Selection

From our perspective as the developers of the AVISPA Tool, in order to guide the design of the tool, to check its coverage, effectiveness, correctness and performance, as well as to demonstrate its practical usefulness, it is indispensable to have a broad collection of relevant protocols as reference examples. Complementarily, from the perspective of users in standardisation bodies like the IETF, it is of enormous value to be able to check whether their security protocols, which have been recently standardised or are currently undergoing standardisation, actually fulfil their respective security requirements.

We thus selected a large number of protocols forming the *AVISPA Selection* — 79 protocols of 33 groups — of practically-important Internet security protocols and their related security goals, in order to provide relevant reference examples for the development of the AVISPA Tool, and afterwards formalised and checked as many of the protocols as possible. To this end, taking advantage of Siemens' good contacts with the standardisation bodies and own contributions to the area of security protocols, we have identified, categorised, and briefly described a large number of candidate protocols as well as their required properties and done a thorough coverage and relevance assessment. The protocols have been selected in such a way to be representative of the many protocol groups currently being developed by the IETF and other standardisation bodies.

The AVISPA Library serves as a suite of benchmark problems for protocol formalisation and analysis that can be readily used to assess the coverage, effectiveness, correctness and performance of rival approaches. Note that, in contrast to the AVISPA Tool, no other state-of-the-art approach is able to deal with these protocols.

Already in the project proposal, we committed to tackle at least 80 security problems from 20 groups which include seven so-called *"Main Protocols"* of particular interest. The even larger number of protocols that we actually modelled and analysed indeed have given invaluable feedback on the language and tools developed, as well as on the security (or flaws) of the protocols themselves, and have proven the usefulness and broad applicability of the AVISPA Tool.

3.3 Modelling and Specification of Security Protocols

Throughout the project, we chose protocols from the protocol selection introduced above, and formalised them in HLPSL. The resulting set of protocols and problems from the AVISPA Selection is called the *AVISPA Library*.

We did part of the formalisations even at stages where HLPSL was not yet fully defined. This gave immediate practical feedback on the ergonomy and expressiveness of HLPSL, for example checking if the syntax is intuitive to the modellers, or pointing out that some important modelling concepts were missing.

The original, more or less informal, protocol specifications, typically given in the form of one or more IETF RFCs or drafts, served as the source of the formal model and as the definite reference in case of questions. Of course, not all details of such specifications (which are often rather technical and not fully focused on security) need to modelled, but only those aspects required for achieving the goals to be reached. Moreover, some aspects like timing are beyond the scope of the AVISPA Tool. So, the challenge is to define a model that faithfully describes the important aspects within the expressiveness of HLPSL as concisely as possible, such that the model is easy to read and model checking is feasible. Independent reviews, as has been common practice within the AVISPA team, are very important to guarantee high quality of the formal models. Very helpful in this respect are also automatic checks, like executability of all transitions, and the diagnostics given by the compiler and attack traces given by the back-ends.

The resulting large set of (currently 66, including variants, of 27 groups) protocol models comprising the AVISPA Library demonstrates that the essentials of industrial-scale protocols, as well as their security requirements, can be faithfully and adequately modelled using HLPSL.

The AVISPA Library is publicly available and can serve the scientific community as a suite of benchmark problems for automatic protocol analysis that can be readily used to assess the performance of rival approaches.

The only related recent effort we know to provide a uniform library of formalised protocols is the SPORE library (Security Protocols Open Repository, `http://www.lsv.ens-cachan.fr/spore/`). It gives Alice&Bob descriptions (with informal semantics) of protocols in the style of Clark and Jacob's first list [66]. More interesting than the format itself, are the comments given in structured ways about attacks and requirements about the protocols. However, SPORE does not seem to be active anymore since the last update was on March 06, 2003.

3.4 Automatic Analysis Techniques for Security Protocols

The back-ends of the AVISPA Tool take as input a security problem formally specified in the IF, automatically carry out an analysis of the problem, and — whenever they terminate — they output the results of their analysis. The AVISPA Tool v1.0 includes for back-ends: OFMC, an on-the-fly model-checker based on lazy data types (developed and maintained by ETHZ); CL-AtSe, a protocol analyser based on Constraint Logic (developed and maintained by INRIA); SATMC, a SAT-based model-checker (developed and maintained by UNIGE); and the T4SP system (developed and maintained by INRIA), which is based on tree automata approximations for validating protocols without restrictions on the number of sessions.

The AVISPA Tool allows for efficient discovery of flaws and validation of protocols. We decided to develop several back-ends based on different technologies since depending from the hypothesis on the protocol model some techniques are more appropriate than others. For instance, whether the messages are typed or not will favour constraint-solving or lazy intruder exploration technique or sat-based model checking.

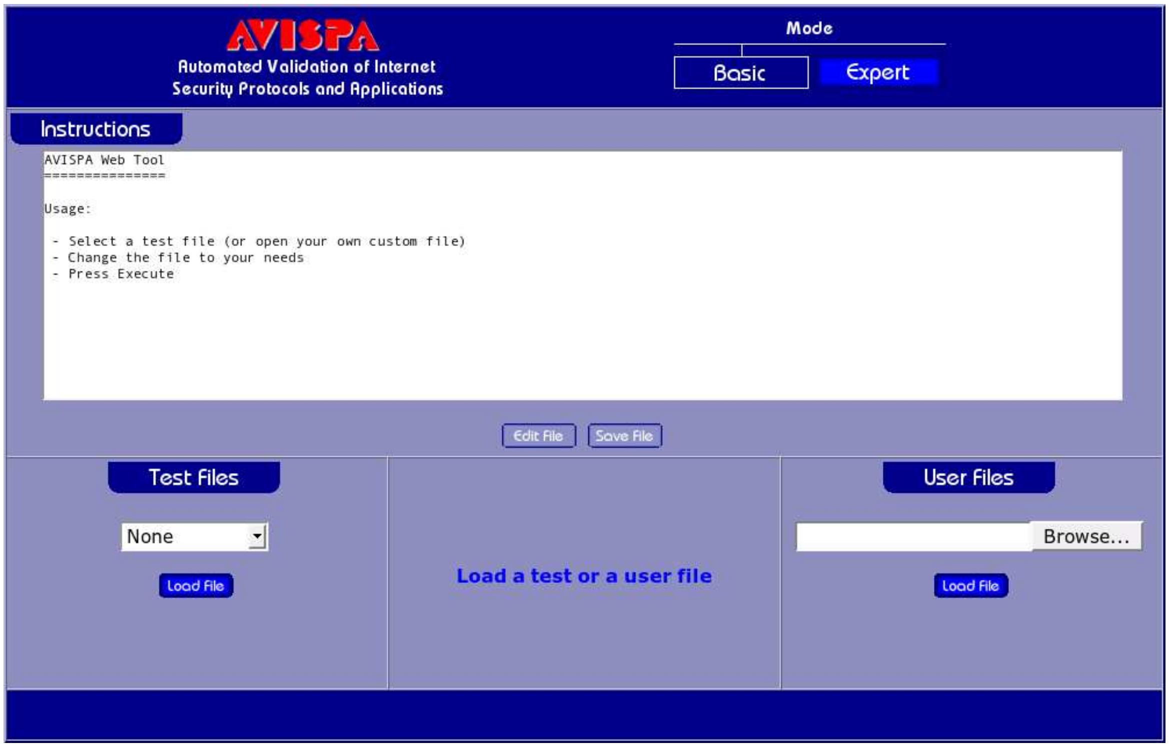
The results obtained give evidence of the following: the technique underlying OFMC is insensitive to the size of messages; the CL-AtSe's approach was particularly suited in tackling protocols where special properties of cryptographic operators are required and the results after the implementation and incorporation of a lot of theoretical work confirm our expectations. The SAT-based Model-Checking technique is effective on scenarios with a bounded number of sessions but is particularly sensitive to the size of messages. Finally the technique underlying TA4SP is currently the only one in the AVISPA framework able to verify secrecy under an unbounded number of sessions

3.5 Tool Development

Following the project objectives, we developed the AVISPA Tool to be easily usable by IT-professionals, engineers, and protocol designers working in industry or standardisation organisations like the IETF. We thus deployed the AVISPA Tool v1.0 as a downloadable single "package" to be installed on the users' local machines, as well as a remote tool that can be employed by external users thanks to the web-interface accessible from the project web-site (more specifically, the web-interface is a graphical front-end to the tool, which provides a suite of applications for the analysis of formal models of security protocols).

The architecture of the tool is depicted in Figure 1 (see also [2]). Specifications of security protocols and properties written in the High-Level Protocol Specification Language (HLPSL) are automatically translated (by the translator HLPSL2IF) into IF specifications, which are then given as input to the different back-ends of the AVISPA Tool: OFMC, CL-AtSe, SATMC, and TA4SP. Whenever it terminates, each back-end of the AVISPA Tool outputs the result of its analysis using a common and precisely defined Output Format stating whether the input problem was solved (positively or negatively), some of the system resources were exhausted, or the problem was not tackled by the required back-end for some reason.

Such a modular architecture has a number of advantages. First, it allows the users to need to understand and be able to use the high-level input language, i.e. write the specifications of their protocols and the properties the protocols should satisfy in HLPSL. In particular, by using the interface, the user can easily load a protocol specification among the ones provided or write a specification on his/her own, and invoke one of the back-ends. In case an attack is found, the attack trace is also output in a graphical format, using Message Sequence Charts, or in a postscript file. As different users have different needs



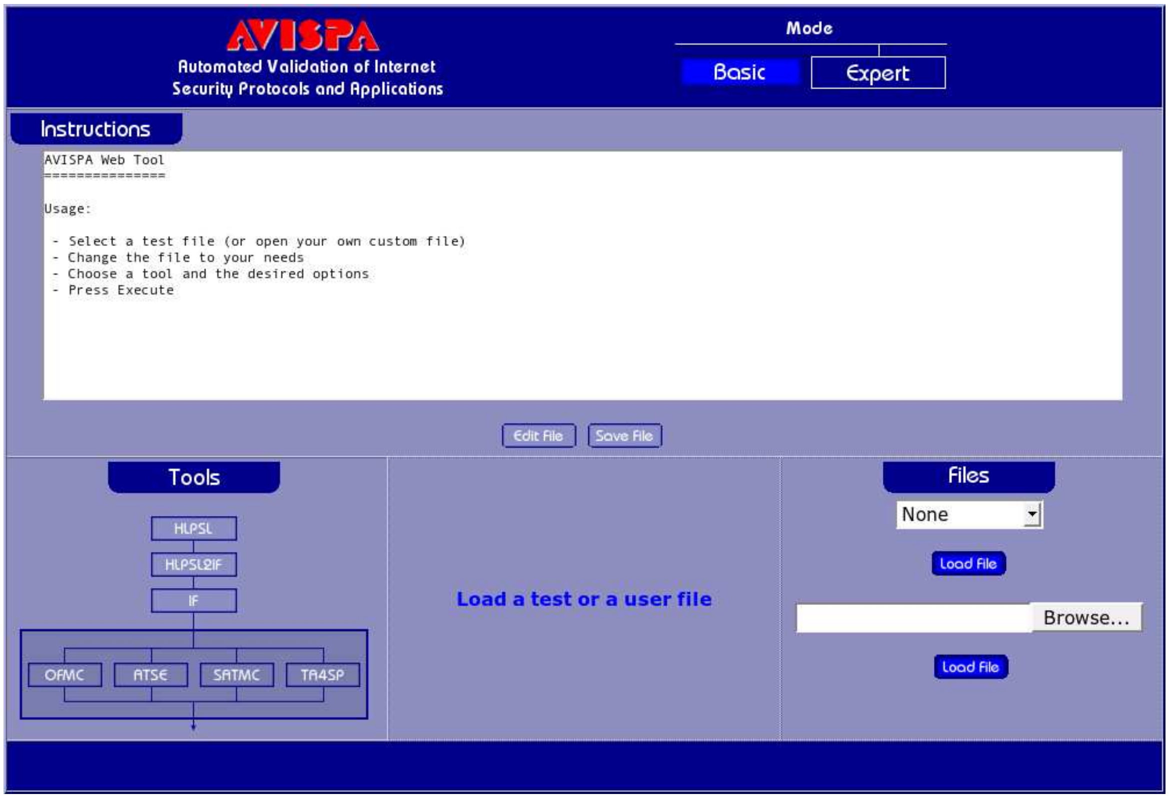
```
AVISPA Web Tool
===============

Usage:

 - Select a test file (or open your own custom file)
 - Change the file to your needs
 - Press Execute
```

Load a test or a user file

Figure 2: Basic Mode

and skills, we designed two kinds of user interaction. As shown in the Figures 2 and 3, and described in more detail in [55], the web-interface is composed of a *basic* and an *expert mode*. One of the main advantages of an on-line tool/interface is that users will always be using the latest version of the analysis tools and do not have to worry about handling different tools versions and doing manual installations. However, if desired, the interface can also be installed locally on the user machine for an off-line use.

The second advantage of our modular architecture is that, thanks to the well-defined syntax and semantics of both languages HLPSL and IF, the HLPSL2IF translator is capable of automatically translating HLPSL specifications into IF specifications, which are then given as input to the AVISPA Tool back-ends. (As discussed in more detail in deliverables [52, 53], similar translations from a high-level language into a low-level one have been developed for CAPSL/CIL [68], CASRUL [72], and from Alice&Bob protocol notation to a process algebra that is similar to the spi-calculus [62] in the context of the projects Mefisto and Degas.) The Intermediate Format thus provides low-level descriptions of protocols and their properties that are suitable for automatic analysis (rather than being abstract and easy to read for human users like the HLPSL), and yet this format is independent from the analysis methods employed by the various back-ends. This independence allows for the independent development of the high-level language and of the back-ends, and also provides an interface for the future connection of other (possibly, project-external) tools to the AVISPA Tool. A first example of this is the work of Gotsman, Massacci, and Pistore [70],



```
AVISPA Web Tool
===============

Usage:

 - Select a test file (or open your own custom file)
 - Change the file to your needs
 - Choose a tool and the desired options
 - Press Execute
```

Load a test or a user file

Figure 3: Expert Mode

who have recently published a translation procedure from protocol descriptions in HLPSL to descriptions in the applied pi calculus, which allowed them to apply the ProVerif tool to some of our HLPSL protocol specifications.

Finally, our modular architecture allowed the different partners of the project to develop their own back-ends independently and compare the results based on the common input (the Intermediate Format) and output (the Output Format).

3.6 Tool Assessment

We have periodically tested the AVISPA Tool against the security problems contained in the AVISPA Library throughout the life of the project with assessment points at month 12, 24, and 30.

Each security problem was analysed under a variety of assumptions, namely whether the messages exchanged are assumed to be well-formed or not and whether the sessions established by the honest parties are known in advance or not. Moreover multiple instances of the individual back-ends have been used in the experiments, each instance resulting from using a given back-end with different settings. (See [58] for the details.)

In order to cope with the large number of tests to be executed we set up a number of scripts to automatically execute the experiments. Moreover, in order to obtain the results in a reasonable amount of time we distributed the computation on a cluster of 8 PCs. Finally we implemented scripts that collect the results of the experiments and automatically build tables summarising the results.

The assessments points proved very useful as they gave us very useful feedback on the relative merits of the various techniques implemented in the back-ends. Moreover the experiments have been a very effective means to detect programming bugs in the back-ends as well as in the HLPSL specifications.

4 Project Results and Achievements

The main results of the project are the following.

R1. Specification Languages for Security Protocols. The High Level Protocol Specification Language (HLPSL) is an expressive language for modelling large-scale communication and security protocols and their properties. HLPSL draws its semantic roots from Lamport's Temporal Logic of Actions (TLA, [73]). HLPSL provides a high level of abstraction and has many features that are common to most protocol specifications – such as intruder models and encryption primitives – built in. In contrast, the Intermediate Format (IF) is a lower-level, tool-independent protocol specification language suitable for automated deduction. The HLPSL2IF translator automatically translates a HLPSL protocol specification provided by the user into an IF specification, which is then given as input to the 4 back-ends of the AVISPA Tool.

R2. On-the-fly model checking of security protocols. The On-the-fly Model-Checker OFMC developed by ETHZ takes as input a specification of a security problem written in IF and performs both protocol falsification and bounded verification in an automatic way, by exploring the transition system described by an IF specification in a demand-driven way (i.e. on-the-fly, hence its name). OFMC is a state-of-the-art protocol analysis tool both in terms of coverage and performance: we have successfully applied it to 215 problems in the AVISPA Library and have been able to re-discover known attacks as well as find new attacks. OFMC's effectiveness is due to the integration of a number of symbolic, constraint-based techniques, which are correct and complete, in the sense that no attacks are lost nor new ones are introduced by them. Moreover, OFMC also implements a number of efficient search heuristics. It supports the specification of algebraic properties of cryptographic operators, and typed and untyped protocol models.

R3. Constraint logic based verification of security protocols. CL-AtSe, INRIA's protocol verifier based on constraint logic, is an implementation of deduction rules for analysing automatically and symbolically security protocols against a generic intruder with various capabilities. Any protocol written in the AVISPA specification languages and given with a bound on the number of sessions can be verified: whenever an attack exists it is produced as output; otherwise, the protocol can be considered as secure for the given number of sessions. The protocol messages can be typed or untyped, and the pairing can be considered associative or not. Several properties of the Xor operator can be handled too. Finally, CL-AtSe performs several kind of optimisations to reduce, and often eliminate, redundancies or useless branches in the protocol symbolic execution. Benchmarks have shown the efficiency of the approach: fractions of seconds are sufficient to explore many protocol analysis problems.

R4. SAT-based model checking of security protocols. The SAT-based Model Checker SATMC developed by UNIGE takes as input a specification of a security problem written in IF and performs both protocol falsification and bounded verification in an automatic way by reducing the input problem to a sequence of invocation to a state-of-the-art SAT-solver. The interface between the SATMC and the SAT solver complies with the DIMACS format (a de facto standard for SAT problems) and therefore SATMC can easy incorporate and exploit new SAT solvers as soon as they will become available. The performance of the tool has improved of several orders of magnitude since the beginning of the project. Currently SATMC successfully analyses most protocols in the AVISPA Library whose cryptographic operators do not enjoy algebraic properties.

R5. Tree Automata based verification of security protocols. We have proposed an extension of a tree automata based approximation method (introduced by Genet and Klay in 1998) for verifying security protocols. The previous procedures of this kind required the presence of an expert who has to transform by hand a security protocol into a term-rewriting system and compute an ad hoc approximation function. Our result allows one to compute an approximation function automatically. We also offer some technical improvements for automatically translating a protocol expressed in the Intermediate Format specification language of AVISPA into a term rewriting system. All these improvements have been implemented in the AVISPA tool and many protocols have been verified in reasonable time using this approach.

R6. Library of formally specified industrial scale security protocols: the AVISPA Library. The AVISPA Selection is a broad collection of 79 practically-important Internet protocols and 384 security properties related to them. The AVISPA Library is a large subset of these, namely 66 protocols (including variants) and their properties that have been modelled in the HLPSL and checked with the AVISPA Tool.

The AVISPA Selection identifies, categorises, and briefly describes a large number of protocols as well as their required properties. It has undergone a thorough coverage and relevance assessment: the protocols have been selected in such a way to be representative of the many protocol groups currently being developed by the IETF and other standardisation bodies.

The AVISPA Library comprises a significant part of the AVISPA Selection, formalising in HLPSL the original, more or less informal, protocol specifications, typically given in the form of one or more IETF RFCs or drafts. The formalisations have been carefully reviewed and cross-checked to make sure that they faithfully describe the important aspects within the expressiveness of HLPSL, while keeping them easy to read and model checking feasible.

The AVISPA Selection and Library are publicly available and can serve the scientific community as a suite of benchmark problems for protocol formalisation and analysis that can be readily used to assess the coverage, effectiveness, correctness and performance of rival approaches. Note that, in contrast to the AVISPA Tool, no other state-of-the-art approach is able to deal with these protocols.

R7. Environment for the automatic validation of security protocols: the AVISPA Tool. The AVISPA Tool is a modular environment for the automatic validation of Internet security protocols and applications, which can be employed by external users thanks to the web-interface or can be installed on the users' local machines. The tool takes as input a specification of a security problem written in IF and outputs the results of the analysis by the 4 back-ends of the tool: OFMC, CL-AtSe, SATMC, and TA4SP. The back-ends implement a variety of analysis techniques, ranging from falsification, to bounded verification, and to unbounded verification. In the latter case, abstraction techniques allow the tool to prove whether protocols satisfy secrecy properties in unbounded execution scenarios, but this comes at the cost of preventing the detection of attacks in some protocols. The IF also provides the interface via which other protocol analysis tools can be connected to the AVISPA environment.

Whenever it terminates, each back-end of the AVISPA Tool outputs the result of its analysis stating whether the input problem was solved (positively or negatively), some of the system resources were exhausted, or the problem was not tackled by the back-end for some reason. The results are output in AVISPA's Output Format, so that protocol attacks can then also be represented graphically, in the form of message sequence charts or as postscript files.

The experiments that we have carried out on security problems in the AVISPA Library demonstrate that the AVISPA Tool is a state-of-the-art protocol analysis tool in terms of coverage (number of different security problems that can be specified), effectiveness (number of different security problems that can be analysed), and performance (amount of time required for the analysis). The AVISPA Tool has been able to re-discover all known attacks to protocols in the library as well as find a number of new attacks.

4.1 Comparison to the original project objectives

All the project objectives have been successfully achieved:

(O1) We have developed the HLPSL and the IF specification languages for formalising security protocols and their properties. By supporting symmetric and asymmetric keys, non-atomic keys, key-tables, Diffie-Hellman key-agreement, hash functions, algebraic functions, typed and untyped data, and many other features needed to specify large-scale Internet security protocols, the HLPSL fully achieves the initial objective of defining "a rich specification language for formalising protocols, security goals, and threat models of industrial complexity". The IF is a language at an accordingly lower abstraction level, which is thus more suitable for automated deduction, and which provides an interface to different back-ends for protocol analysis for the future connection of other (possibly, project-external) tools to the AVISPA Tool.

(O2) We have devised a number of symbolic reduction and abstraction techniques that have led *all* the automated deduction techniques used in the project to scale from small or medium security protocols to large-scale Internet security protocols. We have also developed a novel verification technique based on automata and rewriting that was not originally planned in the Technical Annex.

(O3) Prototype implementations of the translator from HLPSL to IF as well as back-ends implementing the automatic analysis techniques designed within the project have been developed and integrated into the AVISPA Tool. We have also provided the AVISPA Tool with a user-friendly web-interface, a tool for graphically rendering attacks via MSCs, and an emacs-mode for editing and automatically validating protocol specifications. To the best of our knowledge, the AVISPA Tool is the most advanced, yet user-friendly, environment for the automatic validation of security protocols currently available.

(O4) The AVISPA Tool has been thoroughly tuned and assessed against a large collection (the AVISPA Library) of security problems drawn from a significant set of practically relevant, industrial protocols. To the best of our knowledge, the AVISPA Library is the best publicly available library of security protocols both in terms of number and scale of security. We expect that the AVISPA Library will be used as the reference benchmark suite for automatic security protocol analysers for several years to come.

(O5) Both the AVISPA Tool and the AVISPA Library have been made available to the public, and the results of the project have been widely disseminated both to the academic and the industrial communities by means of 18 invited talks, 47 journal and conference papers, 35 paper presentations at conferences, 4 tutorials, 4 project workshops, 2 invited talks at IETF Meetings, as well as the editing and publication of 5 special issues of journals or workshop proceedings.

On the negative side we have not incorporated timestamps in the specification languages as it was initially planned. However this was not much a problem with the corpus of protocols we had to verify. There was also several channel assumptions and more generally environment assumptions that we have not built in the language as planned. Finally there was some other security properties such as anonymity and non-repudiation that we have studied but not yet implemented in the AVISPA Tool. We leave them for future projects and future collaborations with specialists.

4.2 Relations and synergies with other relevant projects

Many projects in recent years have been working on security protocol analysis. We address here only the most representative, advanced or concerted, efforts, which are closely related to the AVISPA project.

4.2.1 The Project PROUVE

The French project PROUVE has started in 2003 and INRIA-Lorraine is a partner of this project (administrative coordinator), together with France Telecom R&D, laboratories LSV (ENS de Cachan, France) and Verimag (Grenoble, France). The objective of PROUVE project is to verify security-sensitive protocols provided by France Telecom R&D. Like for the AVISPA Tool, the PROUVE platform will rely on a high-level specification language close to the language used in textbooks. Three tools will be applied to the case-studies: H1 (LSV), Hermes (Verimag) [63] and CL-AtSe. In contrast to AVISPA, PROUVE is oriented towards the specific applications provided by France Telecom R&D.

H1 and Hermes are designed for verifying secrecy properties for an unbounded number of sessions, in order to prove properties on protocols. Hermes can also detect attacks however it is limited to atomic keys. Hermes has been successfully applied to construct secrecy proofs for about 15 protocols of the Clark/Jacob library. But they are not efficient in finding attacks: when a proof attempt fails, this does not automatically mean that there is an attack. In these cases, the tools provide some reasons for the failure but the user has to find a real attack by himself. Note that actually it is not possible to design tools that are able to prove and disprove secrecy properties automatically for an unbounded number of sessions. Thus, these tools implement some abstractions that allow them to prove secrecy properties, but prevent the detection of attacks in some protocols.

Action taken Collaboration between AVISPA and PROUVE is ongoing: although the specification languages, as well as the case-studies, are different, we believe that some back-end technologies can be shared. France Telecom RD has experimented on verifying an electronic purse protocol with CL-AtSe.

We have frequent contacts with LSV Cachan and Verimag with many reciprocal visits.

4.2.2 ECSS Group, Eindhoven

The Eindhoven Computer Science Security Group (led by Prof. Sjouke Mauw) has been working on the formal verification of black box security protocols, which is closely related to our research. Their approach is based on process algebra and on the μCRL language, which allows one to combine data and processes. The approach is quite similar to the model-checking one of CASPER. In general some approximations have to be done and there is no guarantee that an attack was not overlooked [71]. They have not investigated completeness of their approach.

From its publications list it seems that this group has shifted its interest only recently towards security protocols (the related papers dating from the end of 2003 and 2004).

It seems that only a few protocols have been analysed by ECSS and whether there is a uniform, fully automatic methodology is not obvious from their work. It is questionable whether a non-specialist of μCRL model-checking would be able to apply the technique easily.

Action taken We have begun communication with the Eindhoven Computer Science Security Group, and Sjouke Mauw has accepted to join the program committee of our third year workshop, namely ARSPA'05, which will be held in Lisbon, Portugal, in co-location with ICALP 2005 (`http://www.avispa-project.org/arspa/`).

4.2.3 Blanchet's Logic Programming Approach

Bruno Blanchet (MPI for computer science, Saarbrücken, Germany, and ENS, France) has developed a tool where protocols and security properties are expressed as Horn clauses and he provides strategies to saturate these sets of clauses [49, 59, 60, 61]. The tool ProVerif allows one to prove security properties for an unbounded number of sessions, in particular strong secrecy (which means that an intruder cannot see any difference when the value of the secret changes). Note, however, that the tool may raise some false attack since nonces are abstracted by constants or function symbols, so that attacks have to be constructed by the user itself. However recently ProVerif has been enhanced to reconstruct automatically the attacks [50]. Note also that at the ARSPA'05 workshop, which we organise, Gotsman, Massacci, and Pistore will present a translation procedure from protocol descriptions in HLPSL to descriptions in the applied pi calculus, which allowed them to apply the ProVerif tool to some of our HLPSL protocol specifications [70]. This will provide the basis for further comparison and cross-fertilisation between AVISPA and ProVerif.

Action taken We have discussed at several conferences with Bruno Blanchet to share experience.

4.2.4 The Project DEGAS

A related European project, DEGAS IST-2001-32072 (`http://www.omnys.it/degas/`), is dedicated to the design of an environment for developing global applications. For instance, a case study considered in this project is mobile home-banking. The specification language is UML and the abstract language for verification tasks is based on process algebra. The related Italian project Mefisto (`http://mefisto.web.cs.unibo.it`) on formal methods for security ended in November 2003. In particular, this project has attempted to extend protocol verification to more realistic and detailed models including time and probabilistic information flow.

In the context of these projects, a translation has been designed from Alice&Bob protocol notation to a process algebra that is similar to the spi-calculus [62]. This translation allows one to derive a precise description of the protocol behaviour, and is similar to translations previously devised for CAPSL/CIL [68], CASRUL [72], and HLPSL2IF in our projects AVISS and AVISPA. The verification is then performed by a polynomial-time static analysis and related approximation techniques. Some experiments with classical protocols from the Clark/Jacob Library are given, and both real flaws and false ones are detected on these protocols.

Note that, as discussed in [62], the approach does not address asymmetric cryptography, imperfect cryptography, timing issues, type flaw attacks related to bit-string representations. All these topics have been investigated by the AVISPA project.

Action taken We have invited some of the principal investigators of the DEGAS project to our next workshop: Pierpaolo Degano of the University of Pisa will co-chair the ARSPA'05 workshop together with Luca Viganò of ETHZ, and Hanne Riis Nielson of the Technical University of Denmark has joined the program committee of the workshop. Moreover, we have begun a detailed comparison of the approach of the DEGAS project with our abstraction techniques, and in particular with the tree automata techniques discussed in [56]. We believe that it will be possible for them to reuse our technology, which is more efficient according to the experimental results.

4.2.5 The project MyThS

The project MyThS ("Models and Types for Security in Mobile Distributed Systems") is funded by the Global Computing pro-active initiative (GC) of the Future and Emerging Technologies (FET, contract IST-2001-32617). MyThS addresses the foundations of programming languages and paradigms that allow for the static detection of security violations, and aims at developing type-theoretic methods and tools that enable formal analysis of security guarantees appropriate for systems and applications on the global computing platform. As AVISPA, MyThS addresses the problems of secure communication in open-ended networks. using cryptographic means. However, Myths is more focused on mobility and highly dynamic topologies and expects to derive new mechanisms for decentralised (dynamic) type-checking of distributed computing sites and migrating agents. In contrast, AVISPA is more focused on providing a fully automated analysis tool although for a less general class of processes than those considered in MyThs. To our knowledge MyThs will not provide an integrated tool.

4.3 Implications for EU policies and standards

The AVISPA project fits the objectives of the FET OPEN scheme focused on developing new technologies for significant breakthroughs in industrial and societal terms. The project is also closely related to several other action lines, such as that on computing communications and networks, and to the cross-programme on mobile applications and services, and all those actions and policies where security-sensitive applications are developed or exploited, in areas such as health-care, e-commerce, and e-government. Moreover, the project addresses European policy objectives such as those of the *e*Europe and *e*Europe+ action plans, namely to promotes a cheaper, faster and more secure Internet. The dissemination of the project results to industry and standardisation and regulation bodies such as the IETF contributes to improving the competitiveness of European industry.

4.4 Benefits to society

The AVISPA project has a relevant and immediate impact on social and economic development in Europe. In particular, the project contributes to the efforts of the European Commission to bring the *Information Society* closer to the European citizen. The presentation of the project results in international conferences and workshops has ensured their dissemination to the European (and international) research community and the dissemination of the results to industry and standardisation and regulation bodies such as the IETF contributes to improving the competitiveness of European industry.

In other words, the technology developed in the AVISPA project contributes to the standardisation and industrial consensus of (security-sensitive) Internet protocols and applications, and thereby improves the reliability and efficiency of protocol and networks, and hence reduces their social and marketing costs. It promotes a cheaper, faster and more secure Internet. Moreover, it stimulates the use of the Internet for accelerating and achieving social objectives such as electronic access to public services and online public health services.

5 Deliverables and Publications

5.1 Deliverables

(D2.1) The High-Level Protocol Specification Language HLPSL is the language through which end-users and protocol modellers make use of the AVISPA Tool [24]. As such, it is designed to be accessible: it should be easy for human users to both read and write HLPSL specifications. To this end, HLPSL provides a high level of abstraction and has many features that are common to most protocol specifications – such as intruder models and encryption primitives – built in. In contrast, the Intermediate Format (IF), the language into which HLPSL specifications are translated, is a language at an accordingly lower abstraction level and is thus more suitable for automated deduction.

Protocol specifications in HLPSL are divided into *roles*. Some roles, the so-called *basic* roles, serve to describe the actions of one single agent in a run of a protocol or sub-protocol. Other roles, namely *composed* roles, instantiate these basic roles to model an entire protocol run (potentially consisting of the execution of multiple sub-protocols), a session of the protocol between multiple agents, or the protocol model itself. This latter role is also called the *environment* role.

Given a set of roles describing the protocol and an *environment* role in which we define the concrete sessions whose execution we wish to consider, we then define our security goals. Currently, HLPSL supports only different forms of authentication and secrecy goals, but

Table 2: Summary of the deliverables

No.	Title	WP	Leader	Nature
1.1	Year 1 Progress Report	1	UNIGE (CO1)	R
1.2	Year 2 Progress Report	1	UNIGE (CO1)	R
1.3	Year 3 Progress Report	1	UNIGE (CO1)	R
1.4	Final Project Report	1	UNIGE (CO1)	R
2.1	The High-Level Protocol Specification Language	2	INRIA (CR2)	R&O
2.2	Algebraic properties	2	INRIA (CR2)	R&O
2.3	The Intermediate Format	2	INRIA (CR2)	R&O
2.4	Interface	2	INRIA (CR2)	O
3.1	Security Properties	3	ETHZ (CR3)	R&O
3.2	Assumptions on environment	3	ETHZ (CR3)	R&O
3.3	Sessions instances	3	ETHZ (CR3)	R&O
4.1	Compositionality	4	ETHZ (CR3)	R&O
4.2	Partial-Order Reduction	4	ETHZ (CR3)	R&O
4.3	Heuristics	4	ETHZ (CR3)	R&O
4.4	AVISPA Tool v.1	4	ETHZ (CR3)	R&O
4.5	AVISPA Tool v.2	4	ETHZ (CR3)	R&O
4.6	AVISPA Tool v.3	4	ETHZ (CR3)	R&O
5.1	Abstractions	5	INRIA (CR2)	R&O
5.2	Infinite-state model checking	5	INRIA (CR2)	R&O
5.3	Completeness issue	5	INRIA (CR2)	R
6.1	List of Selected Problems	6	SIEMENS (CR4)	R&O
6.2	Specification of the Problems in the high-level specification language	6	SIEMENS (CR4)	R&O
7.1	Experimental Setup	7	UNIGE (CO1)	O
7.2	Assessment of the AVISPA Tool v.1	7	UNIGE (CO1)	R&O
7.3	Assessment of the AVISPA Tool v.2	7	UNIGE (CO1)	R&O
7.4	Assessment of the AVISPA Tool v.3	7	UNIGE (CO1)	R&O
8.1	AVISPA web-site	8	UNIGE (CO1)	O
8.2	Project Presentation	8	UNIGE (CO1)	R
8.3	Dissemination and Use Plan	8	UNIGE (CO1)	R
8.4	Year 1 Project Workshop	8	UNIGE (CO1)	R
8.5	Year 2 Project Workshop	8	UNIGE (CO1)	R
8.6	Year 3 Project Workshop	8	UNIGE (CO1)	R
8.7	Technology Implementation Plan	8	UNIGE (CO1)	R

more general security objectives, are planned for future versions. We may consider to handle fairness and non-repudiation properties for contract-signing protocols.

(D2.2) Algebraic properties The current automated methods for protocol analysis assume the so-called *perfect encryption hypothesis*, which assumes that there are no relations between the messages apart from the standard ones entailed by the Dolev-Yao model. A first step towards a less abstract model is to take into account some algebraic properties of the cryptographic primitives, and we have thus extended our analysis techniques in order to account for important algebraic properties satisfied by protocol primitives. We have considered different approaches to the formalisation and implementation of such algebraic properties and we have also extended the back-ends of the AVISPA Tool in order to analyse protocols that are based on such properties.

(D2.3) The Intermediate Format The Intermediate Format (IF) is a tool-independent protocol specification language suitable for automated deduction. The HLPSL2IF translator automatically translates a HLPSL protocol specification provided by the user into an IF specification, which is then given as input to the different back-ends of the AVISPA Tool. Hence, the main goal in the design of the IF was to provide a low-level description of the protocol that is suitable for automatic analysis (rather than being abstract and easy to read for human users like the HLPSL), and yet this format should be independent from the analysis methods employed by the various back-ends. The IF describes a protocol in terms of rewrite rules describing an infinite-state transition system with an initial state, transition rules, and a set of state-based safety properties, namely goal predicates that define if a given state is an attack state or not. If one or more properties are not verified at some state, then we can report an attack.

(D2.4) Interface The AVISPA Tool Web Interface is a graphical front-end to the AVISPA Tool v1.0, which provides a suite of applications for the analysis of formal models of security protocols. It consists of an interactive set of dynamically generated HTML [78] pages, which allows for the invocation of the four back-ends OFMC, CL-AtSe, SATMC, and TA4SP, with the respective options.

Since different users may have different needs and skills, we devised two kinds of user interaction: the web interface supports both a *basic* and an *expert mode*. By using the interface, the user can easily load a protocol specification among the ones provided or write his own specification, and invoke one of the back-ends. In case an attack is found, the attack trace is also output in a graphical format, using Message Sequence Charts, or is output in a postscript file.

(D3.1) Security Properties We have investigated the specification and formalisation of a number of security properties, such as different forms of authentication, secrecy, anonymity, and non-repudiation. Our current methods are well suited to automatically – and very efficiently – check for violations of security properties goals like authentication and secrecy. Furthermore, many other security properties closely resemble authentication and secrecy. It is, therefore, particularly desirable to find reductions from complex properties into Boolean or temporal combinations of these simpler properties for which efficient analysis techniques already exist. We adopt this approach wherever possible.

(D3.2) Assumptions on environment In protocol analysis, the *environment* refers to the formal definition of the conditions under which a security protocol executes. This environment comprises many elements. Among them we count, for instance, the deductive powers assumed of the intruder and the properties of the communication channels over which messages are sent, including the (perhaps differing) intruder types to which said channels are vulnerable. Modern Internet protocols are designed to be executed in a variety of environments. The ability to specify a rich set of assumptions on the protocol analysis environment is therefore important for the specification and analysis of such protocols. We have devised a four techniques, described in detail in this deliverable, which yield a more expressive set of assumptions on the environment.

(D3.3) Sessions instances We have devised a number of techniques that allow the users of the AVISPA Tool to specify parallel protocol sessions and thereby accelerate search and increase the performance and efficiency of our protocol validation tool.

(D4.1) Compositionality We have developed three different approaches to tackle different aspects of compositionality and thereby further scale up the automated deduction techniques and tools that we have been developing in the AVISPA project. The approaches are: a general method for reasoning about contract-signing protocols using a specialised protocol logic, an algorithm for combining decision procedures for intruder constraints on disjoint signatures, and (Abstract) Secure Communication Channels as a means of modelling larger application protocols which make use of several sub-protocols and where one wishes to specify different intruder models for different parts of a protocol.

(D4.2) Partial-Order Reduction We have investigate that use of partial-order reduction techniques in the validation of security protocols, and have hence introduced *constraint differentiation*, a new general technique that we have developed for substantially reducing search when model-checking security protocols, and which considerably improves the performance of the AVISPA Tool, enabling its application to a wider class of problems.

(D4.3) Heuristics We have developed a collection of heuristics that substantially reduce search and improve the performance of the back-ends of the AVISPA Tool.

(D4.4, D4.5, D4.6) AVISPA Tool The AVISPA Tool can be employed by external users thanks to the web-interface accessible from the project web-site, and is also downloadable as a single "package" to be installed on the users' local machines. The architecture of the tool is depicted in Figure 1. Specifications of security protocols and properties written in the High-Level Protocol Specification Language (HLPSL) are automatically translated (by the translator HLPSL2IF) into IF specifications, which are then given as input to the different back-ends of the AVISPA Tool: OFMC, CL-AtSe, SATMC, and TA4SP. Whenever it terminates, each back-end of the AVISPA Tool outputs the result of its analysis using a common and precisely defined format stating whether the input problem was solved (positively or negatively), some of the system resources were exhausted, or the problem was not tackled by the required back-end for some reason.

(D5.1) Abstractions For automatic protocol verification to become feasible, it is often needed to abstract away from specification details or to introduce finite or regular descriptions for infinite sets of data. Abstractions in our protocol verification context are mappings from the original rewrite rules model into a simpler model such that every protocol flaw (of the original model) is contained in the abstract model. The converse usually doesn't hold, since, due to the simplification, we may have false positives in the abstract model even when the original model is flawless. In this deliverable, we present a number of different abstractions that we have been considering in our protocol analysis tools. More specifically, we present abstractions of the nonces and of the intruder knowledge, as well as a general approach for designing abstractions.

(D5.2) Infinite-state model checking Protocol verification requires the analysis of an infinite number of configurations and therefore calls for infinite-state model checking techniques. In Deliverable 5.1, we have proposed several sound abstractions for reducing the verification to a finite number of states. In this deliverable, we first introduce a verification algorithm for time-sensitive security protocols. The verification is performed by symbolic exploration of upward closed set of configurations. We then present an extension of the tree automata technique introduced in Deliverable 5.1, which allows us to reduce the number of states generated by the approximation function and therefore handle more protocols from the AVISPA library. Finally, we report on some finer abstractions on nonces in fixed-point computations, which allow us to analyse the ASW contract signing protocol and to point to a new attack on it.

(D5.3) Completeness issue Our back-ends either consider a finite number of protocol sessions or perform abstractions. In the former case, we cannot be sure that there are no attacks at all; in the latter case some false attacks might be reported. In this deliverable we have investigated two classes of protocols for which we can decide the existence of secrecy flaws. For the first class, we have encoded the protocol analysis problem as a logical satisfiability problem that we have decided by a resolution strategy. For the second class, we have applied tree automata techniques.

(D6.1) List of Selected Problems In this deliverable, after a thorough evaluation of numerous Internet protocols and their properties, we define the *AVISPA Selection*, a list of problems (recall that a problem is given by both a protocol and a security property the protocol should satisfy). On average, a protocol has around four properties, each of them counting as one problem. The list contains a total of 384 security problems and 79 protocols, mostly from the IETF, divided into 33 groups.

We also present a coverage and relevance assessment of the resulting problem set, with comments of IETF representatives taken into account, and give an extensive bibliography.

As a basis for the success criteria for the projects, we have committed to specify at least 80 security problems from 20 groups, and for at least 60 of these 80 problems, including at least one problem from each of the first seven groups (called the *"Main Protocols"*), to verify that the protocol satisfies the desired security property.

(D6.2) Specification of the Problems in the high-level specification language In this deliverable, we present the *AVISPA Library*, consisting of specifications of the protocols and security problems that we have modelled in the HLPSL and analysed with the AVISPA Tool. This set of protocols is a substantial subset of those described in Deliverable 6.1 and contains meanwhile 66 protocol models from 27 groups. For each protocol, we describe its purpose, variants (if any), the message exchange in the Alice&Bob notation, the security problems formalised, and the actual HLPSL code. If attacks were found, we explain these and give fixes for them. Where appropriate, we add further explanations and comments.

The AVISPA Library is presented as a large document type-set in LaTeXand as HTML pages on the AVISPA web-site. A significant part of the models is used in the assessment, and some of them are included with the AVISPA package as educational examples.

Assessment of the AVISPA Tool (D7.1, D7.2, D7.3, D7.4) The AVISPA tool has been periodically tested against the AVISPA Library in order to monitor and assess its coverage, effectiveness, and performance. In Deliverable 7.1 a machinery to automatically run the experimental analysis has been set up. The results of the assessments at project months 12, 24, and 30—presented in Deliverable 7.2, Deliverable 7.3 and Deliverable 7.4 respectively—demonstrate the achievement of the project's objectives for the reporting periods. The results of the last assessment are given in Table 1. We have been able to formalise in the HLPSL 215 problems from 22 groups (including 38 problems from the seven main groups described in Deliverable 6.1 [57]), and the tool successfully analyses all of them in less than 24 minutes of CPU time per problem (globally, the entire library of 215 problems requires 87 minutes of CPU time to be analysed). Therefore, all the above requirements (namely coverage, effectiveness, and performance) are largely fulfilled by the AVISPA Tool. Moreover, the tool is able to detect new (i.e. previously unknown) attacks to some of the protocols analysed.

AVISPA web-site (D8.1) In order to provide a comprehensive and timely dissemination of the project's results we have set up at the very beginning of the project and since then maintained a publicly available web-site at the URL `http://www.avispa-project.org`. Currently the web-site includes the following sections: *(i)* a general introduction to the project: the objectives, the expected results, the milestones, the detailed description of the consortium and its coordinates within the Fifth Framework Programme; *(ii)* the list of events related to AVISPA: meetings, conferences, workshops, and their availability to the public; *(iii)* the list of present and past collaborators; *(iv)* publications related to AVISPA, both in the scientific community and in the general press; *(v)* a "Software" section which contains the downloading instructions for the AVISPA Tool as well as a link to the AVISPA web-based graphical user interface; and *(vi)* three sections dedicated to (1) internal communication among AVISPA partners, (2) communication with the European Commission, (3) communication with the IETF.

Project Workshops (D8.4, D8.5, D8.6) In order to disseminate the results of the AVISPA project to academia and industry, we published a large number of papers (see Section 5.2), gave a large number of conference presentations, talks, tutorials and demos (see Sections 5.2, 5.3 and 5.4, and also organised the Workshop on Security Protocols Verification SPV as well as three project workshops (see Section 5.5).

The First Project Workshop was held at INRIA-Lorraine (Nancy) on January 23, 2004, and was devoted to recent advances on the specification of security protocols and their properties, as well as on the techniques for their automatic analysis. The technical program of the workshop consisted of several talks by members of the AVISPA project and was enriched by the talks of two internationally renown invited speakers: Prof. Peter Ryan (University of Newcastle, U.K.) and Dr. Yassine Lakhnech (Verimag, Grenoble, France). The results of the workshop have been significant in terms of dissemination, cross-fertilisation of ideas, spill-over effects, and establishment of new synergies with other research teams.

The Second Project Workshop, titled "Automated Reasoning for Security Protocol Analysis" (ARSPA), was held at the University College, Cork (Ireland), on July 4, 2004 in the context of the 2nd International Joint Conference on Automated Reasoning (IJCAR'04). The workshop brought together researchers and practitioners from both the security and the automated reasoning communities, from academia and industry, who are working on developing and applying automated reasoning techniques and tools for the formal specification and analysis of security protocols. The results of the workshop have been significant in terms of dissemination and cross-fertilisation of ideas. The workshop proceedings have been published as volume 125(1) of the Electronic Notes in Theoretical Computer Science. Moreover, the members of the program committee of ARSPA (namely, Alessandro Armando, David Basin, Jorge Cuellar, Michael Rusinowitch, and Luca Viganò) have guest-edited a soon-to-appear Special Issue of the Journal of Automated Reasoning collecting original papers on automated reasoning techniques and tools for the formal specification and analysis of security protocols.

The Third Project Workshop, titled "The Second Workshop on Automated Reasoning for Security Protocol Analysis" (ARSPA'05), will be held on July 16, 2005, in the context of The 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), in Lisbon, Portugal. As for its predecessor, the ARSPA'05 workshop will bring together researchers and practitioners from both the security and the automated reasoning communities, from academia and industry, who are working on developing and applying automated reasoning techniques and tools for the formal specification and analysis of security protocols. The workshop proceedings have been published as volume 135(1) of the Electronic Notes in Theoretical Computer Science. Moreover, the workshop organisers (namely, Pierpaolo Degano of the University of Pisa, Italy, and Luca Viganò of ETHZ) are planning a Special Issue of an international journal to collect original papers on automated reasoning techniques and tools for the analysis of security protocols.

5.2 Publications and Conference Presentations

Almost 50 journal or conference papers related to the project have been published and we envisage at least 20 more paper will be generated after project's completion. All papers published in conference proceedings have been presented by members of the AVISPA team.

[1] P. Ammirati and G. Delzanno. Constraint-based Automatic Verification of Time Dependent Security Properties. In *Proceedings of SPV'03*, 2003.

[2] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Heám, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*. Springer-Verlag, 2005.

[3] A. Armando, C. Castellini, E. Giunchiglia, and M. Maratea. The SAT-based Approach to Separation Logic. Technical report, UNIGE, 2005. To be published in the Journal of Automated Reasoning, 2005.

[4] A. Armando and L. Compagna. Abstraction-driven SAT-based Analysis of Security Protocols. In *Proceedings of SAT 2003*, LNCS 2919. Springer-Verlag, 2003.

[5] A. Armando and L. Compagna. An optimized intruder model for sat-based model-checking of security protocols. *Electronic Notes in Theoretical Computer Science (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2004)*, 125(1):91–108, 2005.

[6] A. Armando and L. Compagna. SATMC: a SAT-based model checker for security protocols. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume 3229 of *LNAI*, pages 730–733, Lisbon, Portugal, 2004. Springer-Verlag.

[7] A. Armando, L. Compagna, and P. Ganty. SAT-based Model-Checking of Security Protocols using Planning Graph Analysis. In K. Araki, S. Gnesi, and D. Mandrioli, editors, *Proceedings of the 12th International Symposium of Formal Methods Europe (FME)*, LNCS 2805, pages 875–893. Springer-Verlag, 2003.

[8] A. Armando, L. Compagna, and Y. Lierler. Automatic compilation of protocol insecurity problems into logic programming. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume 3229 of *LNAI*, pages 617–627, Lisbon, Portugal, 2004. Springer-Verlag.

[9] A. Armando and L. Viganò. Preface (editorial). *Electronic Notes in Theoretical Computer Science (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2004)*, 125(1):1, 2005.

[10] A. Armando and L. Viganò, editors. *Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA 2004)*, Amsterdam, The Netherlands, 2005. Electronic Notes in Theoretical Computer Science 125 (Elsevier Science Direct).

[11] M. Backes, A. Datta, A. Derek, J. Mitchell, and M. Turuani. Compositional analysis of contract signing protocols. In *Proceedings of 18th IEEE Computer Security Foundations Workshop*, 2005.

[12] D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In E. Snekkenes and D. Gollmann, editors, *Proceedings of ESORICS'03*, LNCS 2808, pages 253–270. Springer-Verlag, 2003.

[13] D. Basin, S. Mödersheim, and L. Viganò. Constraint Differentiation: A New Reduction Technique for Constraint-Based Analysis of Security Protocols. In V. Atluri and P. Liu, editors, *Proceedings of CCS'03*, pages 335–344. ACM Press, 2003.

[14] D. Basin, S. Mödersheim, and L. Viganò. Constraint Differentiation: A New Reduction Technique for Constraint-Based Analysis of Security Protocols (Extended Abstract). In *Proceedings of SPV'03*. Available at `www.loria.fr/~rusi/spv.html`, 2003.

[15] D. Basin, S. Mödersheim, and L. Viganò. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, 4(3):181–208, June 2005. Published online December 2004.

[16] Y. Boichut, P.-C. Heam, O. Kouchnarenko, and F. Oehl. Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols. In *Proceedings of Automated Verification of Infinite States Systems (AVIS'04)*, ENTCS, 2004. To appear.

[17] Y. Boichut, P.-C. Heam, O. Kouchnarenko, and F. Oehl. Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols. In *Proc. Int. Workshop on Automated Verification of Infinite-State Systems (AVIS'2004), joint to ETAPS'04*, pages 1–11, Barcelona, Spain, 2004. The final version will be published in EN in Theoretical Computer Science, Elsevier.

[18] C. Caleiro, L. Viganò, and D. Basin. Towards a metalogic for security protocol analysis. In W. A. Carnielli, F. M. Dionísio, and P. Mateus, editors, *Proceedings of the Workshop on the Combination of Logics: Theory and Applications (Comblog'04)*, pages 187–196. Center for Logic and Computation, Departamento de Matemática, Instituto Superior Técnico, Lisbon, Portugal, 2004.

[19] C. Caleiro, L. Viganò, and D. Basin. Deconstructing Alice and Bob. *Electronic Notes in Theoretical Computer Science 135(1):3–22 (Proceedings of the Second Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2005)*, 2005.

[20] C. Caleiro, L. Viganò, and D. Basin. Metareasoning about security protocols using distributed temporal logic. *Electronic Notes in Theoretical Computer Science (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2004)*, 125(1):67–89, 2005.

[21] C. Caleiro, L. Viganò, and D. Basin. Relating strand spaces and distributed temporal logic for security protocol analysis. *Logic Journal of the IGPL*, to appear.

[22] Y. Chevalier. *Résolution de problèmes d'accessibilité pour la compilation et la validation de protocoles cryptographiques.* Phd, Université Henri Poincaré, Nancy, December 2003.

[23] Y. Chevalier. A simple constraint combination procedure for cryptographic protocols with xor. In M. Kohlhase, editor, *18th Int. Workshop on Unification*, Cork, Ireland, July 2004. Long version available as INRIA Research Report RR-5224.

[24] Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, J. Mantovani, S. Mödersheim, and L. Vigneron. *A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols*, volume 180 of *Automated Software Engineering*, pages 193–205. Austrian Computer Society, Austria, September 2004.

[25] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proceedings of the Logic In Computer Science Conference, LICS'03*, pages 261–270, 2003.

[26] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science, FST TCS'03*, LNCS 2914. Springer-Verlag, 2003.

[27] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Commuting Public Key Encryption. *Electronic Notes in Theoretical Computer Science (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2004)*, 125(1):55–66, 2005.

[28] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. *Theoretical Computer Science*, 338(1-3):247–274, June 2005.

[29] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions. In M. Baaz, editor, *Proceedings of CSL'2003*, LNCS 2803. Springer-Verlag, 2003.

[30] Y. Chevalier and M. Rusinowitch. Combining Intruder Theories. Technical report, INRIA — extended abstract to appear in the proceedings of ICALP'05, 2005. `http://www.inria.fr/rrrt/rr-5495.html`.

[31] Y. Chevalier and M. Rusinowitch. Combining Intruder Theories. In L. Vigneron, editor, *Proceedings of the 19th International Workshop on Unification*, pages 63–76, Nara, Japan, April 2005.

[32] Y. Chevalier and L. Vigneron. Rule-based Programs describing Internet Security Protocols. In S. Abdennadher and C. Ringeissen, editors, *5th Int. Workshop on Rule-Based Programming (RULE)*, Aachen, Germany, June 2004.

[33] Y. Chevalier and L. Vigneron. Strategy for Verifying Security Protocols with Unbounded Message Size. *Journal of Automated Software Engineering*, 11(2):141–166, April 2004.

[34] V. Cortier, M. Rusinowitch, and E. Zalinescu. A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In *Proceedings of the 7th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP'05), Lisboa, Portugal.* ACM press, July 2005.

[35] P. Degano and L. Viganò. Preface (editorial). *Electronic Notes in Theoretical Computer Science 135(1):1–2 (Proceedings of the Second Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2005)*, 2005.

[36] P. Degano and L. Viganò, editors. *Proceedings of the Second Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA 2005).* Electronic Notes in Theoretical Computer Science 135(1), 2005.

[37] G. Delzanno and P. Ganty. Symbolic Methods for Automatically Proving Secrecy and Authentication in Infinite-state Models of Cryptographic Protocols. In *Proceedings of the Workshop on Issues in Security and Petri Nets (WISP'03)*, 2003.

[38] G. Delzanno and P. Ganty. Automatic verification of time sensitive cryptographic protocols. In K. Jensen and A. Podelski, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*, volume 2988 of *Lecture Notes in Computer Science*, pages 342–356. Springer, 2004.

[39] P. Hankes Drielsma and S. Mödersheim. The ASW Protocol Revisited: A Unified View. *Electronic Notes in Theoretical Computer Science (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2004)*, 125(1):141–156, 2005.

[40] P. Hankes Drielsma, S. Mödersheim, and L. Viganò. A formalization of off-line guessing for security protocol analysis. In F. Baader and A. Voronkov, editors, *Proceedings of LPAR'04*, volume 3452 of *LNAI*, pages 363–379. Springer, 2005.

[41] M. Rusinowitch. Automated Analysis of Security Protocols. In G. Vidal, editor, *Proceedings of the 12th International Workshop on Functional and (Constraint) Logic Programming, WFLP'03*, volume 86(3). Electronic Notes in Theoretical Computer Science, 2003.

[42] M. Rusinowitch. A decidable analysis of security protocols. In J.-J. Lévy, E. Mayr, and J. Mitchell, editors, *18th IFIP World Computer Congress on Theoretical Computer Science - TCS'2004*, Toulouse, France, August 2004. Kluwer Academic Publishers.

[43] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *Theoretical Computer Science*, 299:451–475, 2003.

[44] T. Truderung. Regular protocols and attacks with regular knowledge,. In *Proceedings of the International Conference on Automated Deduction (CADE)*, LNCS. Springer, 2005.

[45] M. Turuani. *Sécurité des Protocoles Cryptographiques: Décidabilité et Complexité*. Phd, Université Henri Poincaré, Nancy, December 2003.

[46] L. Vigneron. Automatic verification of security protocols. In M. Kohlhase, editor, *18th Int. Workshop on Unification*, Cork, Ireland, July 2004. Invited talk.

[47] L. Vigneron, editor. *Proceedings of the 19th International Workshop on Unification*, Nara, Japan, April 2005. Available as LORIA Research Report A05-R-022, Nancy, France.

5.3 Tutorials

- Full day tutorial at the International Conference on Software Engineering And Formal Methods, in the context of SEFM 2003, Brisbane, Australia, September 22–27, 2003. URL: `http://www.svrc.uq.edu.au/Events/SEFM03/`

- Full day tutorial on Automated Validation of Security Protocols (AVASP'04) held on July 5, 2004 at the University College Cork, Ireland in the context of the Second

International Joint Conference on Automated Reasoning (IJCAR'04). (Tutorial Web-Site: http://www.avispa-project.org/avasp)

- Full day tutorial on Automated Validation of Security Protocols (AVASP'05), in the context of the 8th European Joint Conference on Theory and Practice of Software (ETAPS 2005), Edinburgh, Scotland, April 3, 2005.
- Tutorial entitled "A Tool helping to Design Cryptographic Protocols", presented at the 4th Conference on Security and Network Architectures (SAR), Batz sur Mer, France, June 2005.

5.4 Invited Talks

- Invited talk by J. Cuellar (Siemens) at the Industrial Day of the Formal Methods Europe Conference (FME 2003): Specifying and Verifying real-world Security Protocols. URL: http://fme03.isti.cnr.it/iday-progr.htm.
- Invited talk on Automated Analysis of Security Protocols and the AVISPA Project, by M. Rusinowitch. WFLP'03, 12th International Workshop on Functional and (Constraint) Logic Programming. Valencia, Spain, June 12–13, 2003. ENTCS Volume 86(3).
- Invited talk on Deciding Security of Cryptographic Protocols and the AVISPA Project, by M. Rusinowitch. Sixth International Workshop in Formal Methods (IWFM'03). School of Computing, Dublin City University, July 11, 2003.
- Invited talk on the OFMC back-end and the AVISPA project by L. Viganò (ETHZ) at the Dagstuhl Seminar on Language-Based Security. Dagstuhl, Germany, October 5–10, 2003.
- Invited Presentation by J. Cuellar (Siemens) of the AVISPA protocols and problems at the Open Security Area Directorate Meeting (SAAG) at Seoul, March 4, 2004. http://www.ietf.org/meetings/IETF-59.html (Number of attendees: about 300).
- Invited talk on Automatic Verification of Security Protocols and the AVISPA Project, by L. Vigneron, 18th Int. Workshop on Unification. Cork, Ireland. July 2004.
- Invited talk on A Decidable Analysis of Security Protocols and the AVISPA Project, by M. Rusinowitch. 18th IFIP World Computer Congress - Theoretical Computer Science. Toulouse, August 22–27, 2004. Kluwer Academic Publishers.
- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Faculty of Mathematics of the University of Bologna, Italy, October 25, 2004.
- Invited talks on the AVISPA Project by A. Armando (UNIGE) and on SATMC by L. Compagna (UNIGE) at the Istituto per la Ricerca Scientifica e Tecnologica (ITC-IRST), Trento, Italy, Nov 4, 2004.

- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Dagstuhl Seminar "SPP". Dagstuhl, Germany, November 5–10, 2004.

- Invited talk by Y. Boichut (INRIA) on the TA4SP back-end and HLPSL at the Institut de Recherche en Informatique et Système Aléatoire (IRISA), Rennes, France, December 9, 2004.

- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Faculty of Computer Science of the University of Pisa, Italy, February 17, 2005.

- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Faculty of Computer Science of the University of Verona, Italy, February 22, 2005.

- Invited presentation of the AVISPA Project by A. Armando (UNIGE) at the Open Security Area Directorate Meeting (SAAG) held in the context of the 62nd Meeting of the IETF, Minneapolis, March 5, 2005. The slides of the talk are available online in the proceeding of the IETF at `http://www3.ietf.org/proceedings/05mar/saag.html` (Number of attendees: about 200).

- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Faculty of Informatics of the University of Lugano, Switzerland, April 21, 2005.

- Invited talk on the AVISPA Project by L. Viganò (ETHZ) at the Special Session on Security of the Twenty-First Conference on the Mathematical Foundations of Programming Semantics (MFPS XXI), Birmingham, U.K., May 19, 2005.

- Invited talk on the OFMC back-end and the AVISPA Project by L. Viganò (ETHZ) at the Faculty of Computer Science of the Imperial College London, U.K, July 6, 2005.

- Invited talk by J. Cuellar (Siemens) at Colloque sur les Risques et la Sécurité d'Internet et des Systèmes : "Analyse et Vérification de protocoles cryptographiques". Bourges, France. October 13–14, 2005.

5.5 International Workshops Organisations

- SPV — Workshop on Security Protocols Verification, September 6, 2003, Marseille, France. Workshop Web-Site: `http://www.loria.fr/~rusi/spv.html`. (Number of attendees: about 70.)

- First Project Workshop, INRIA-Lorraine Nancy, January 23, 2004. Workshop Web-Site: `http://qsl.loria.fr/Externe/Evennements/JourneeQSL/Journee23-01-2004/programme.htm`. Invited guests: Prof. Peter Ryan (University of Newcastle, U.K.)

and Dr. Yassine Lakhnech (Verimag, Grenoble, France). (Number of attendees: about 25.)

- Second Project Workshop: Workshop on Automated Reasoning for Security Protocols Analysis (ARSPA'04) co-located with the Second International Joint Conference on Automated Reasoning (IJCAR'04) in Cork (Ireland), July 4, 2004. Workshop Web-Site: `http://www.avispa-project.org/arspa/`. (Number of attendees: about 50.)

- Third Project Workshop: The Second Workshop on Automated Reasoning for Security Protocols Analysis (ARSPA'05) will take place on July 16th, 2005 (i.e. just after the end of the project), in the context of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005) in Lisbon, Portugal. Workshop Web-Site: `http://www.avispa-project.org/arspa/`. (Number of attendees: about 40.)

5.6 Editorial Activities

- Proceedings of the IJCAR'04 Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA'04), Alessandro Armando and Luca Viganò, editors. Electronic Notes in Computer Science 125(1), Elsevier Science, 2005.

- Proceedings of The Second Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA'05, co-located with ICALP'05), Pierpaolo Degano and Luca Viganò, editors. Electronic Notes in Computer Science 135(1), Elsevier Science, 2005.

- Special issue on First-Order Theorem Proving of the Journal of Automated Reasoning. Deepak Kapur and Laurent Vigneron, editors. Volume 33, Numbers 3-4, Springer Science+Business Media B.V., October 2004.

- Special Issue of the Journal of Automated Reasoning on "Automated Reasoning for Security Protocol Analysis". A large number of papers (21) has been submitted and we selected 5 high-quality contributions for publication.

- Proceedings of the 19th International Workshop on Unification. Laurent Vigneron, editor. Available as LORIA research report A05-R-022, Nancy, France.

6 Potential Impact of Project Results

With the spread of the Internet and network-based services, and the development of new technologies, the number and scale of Internet security protocols and applications under development has been out-pacing the ability of both industry and academia to rigorously analyse and validate them. This has become an increasingly serious problem especially for standardisation bodies like the *Internet Engineering Task Force* (IETF), the *International*

Telecommunication Union (ITU), the *World Wide Web Consortium* (W3C), the *Liberty Alliance*, and the *Open Mobile Alliance* (OMA), as well as for companies whose products and/or services whose time-to-market requirements depend on the rapid standardisation and correct functioning of these protocols. It is also a problem for users of these technologies, whose possessions, rights and freedoms, e.g. the protection of payment transactions and the privacy of personal data, depend on a secure infrastructure. Striking negative examples include the tremendous vulnerabilities of WEP used in the WLAN IEEE 802.11b standard and the enormous delays in Mobile IPv6 and Geopriv, due to unsolved security issues, in particular in the area of correct protocol design, where possible subtle forms of attacks are most easily overlooked.

The AVISPA project has significantly pushed forward the state-of-the-art of validating security protocols. Now, the protocol designers have a tool at their disposal that — in contrast to earlier efforts in this direction — is actually capable of checking the major security properties of the protocols that they are currently designing. Analysing a protocol with the AVISPA Tool usually takes just hours or a few days (in case of a particularly complex protocol or of an unexperienced user), and is therefore not only considerably quicker and cheaper than the usual review procedures, but it also provides a formal result, which is not the case for such procedures.

As shown by our experience with students and the first AVISPA users who did not have any previous experience with formal protocol analysis shows, using the documentation and examples at hand, a person familiar with a security protocol to be checked can learn within a couple of days to make effective use of the AVISPA Tool, i.e. to model the protocol and its environment with the expressive formal specification language HLPSL and to validate the intended security properties with the available set of powerful and easy-to-use model checkers. Moreover, since the collection of well-checked protocols in the AVISPA Library provides examples and trusted re-usable modules, it serves as a good basis for developing new protocols. Thus, the AVISPA project results speed up the development of the next generation of network protocols and thereby to improve their security and ultimately the acceptance of products based on them.

The initial set of users of the AVISPA technology are the security protocol designers of companies and institutes related to the companies and institutes of the project participants. In particular, Siemens has already applied the AVISPA Tool within various other projects and benefited from it for enhancing trust and confidence in their own protocol proposals, and in the protocol specifications like Geopriv, NSIS and HIP that are currently being designed in the international standardisation communities and that will be adopted by the Internet, mobile communications, and e-commerce industries. This usage is paving the way to the migration of our technology into industry standardisation bodies such as the IETF, which already have shown their interest using it, so that both the scientific and the industrial communities benefit from the advances achieved by the project.

7 Future Outlook

All the project partners will continue to develop the AVISPA technologies and tools. Moreover, given the experience gained with the AVISPA project and given also the excellent experimental results we have obtained in analysing security protocols, the partners have decided, jointly with three further participants, to submit a follow-up project proposal with an extremely challenging goal, namely, the modular design and validation of composed security services. We have submitted this proposal, called *VAMOS* (for Validating Modular Security Services), on June 17 as a STREP project in the FET-Open program.

Our motivation is that IT infrastructures are increasingly conceived as service architectures providing inherently complex solutions in a demand-driven way. Service-oriented computing requires that services and resources are exposed to the network and thus to the outside world. This in turn requires trust and security: system and application owners, service providers, and users must have confidence that their protection needs are met. However, establishing trust and security of composed systems is highly involved, as the composition of services may subject their individual independent security to subtle and new types of interference. Instead of just a combination of security services, we thus need validated security services as well as their validated modular integration.

Tool support is of utmost importance both for the provision of validated security services and for their composition into architectures meeting application security needs. In particular, tool support is required to accelerate and improve the deployment of trustworthy systems and applications.

The VAMOS project thus proposes the development of a framework for the formal specification of security requirements, the formal analysis of security services, and their composition in an automated and validated way. In particular, the project will develop: (i) a logic comprising a language with precise syntax and semantics for describing security primitives (channels, objectives, policies, trust, etc.), and a deduction system for reasoning about security properties of system modules and about their combination and usage at the application and the system level; (ii) a methodology for the validation of modular security services, based on combining model-checking for validating basic services and deductive reasoning for validating their composition.

The framework will be implemented in a prototype tool and will be driven by a set of challenging large-scale case studies, mostly provided by the industrial partners, but also from the public domain, and linked to standardisation efforts.

8 Conclusions

We have successfully developed a number of methodologies and technologies for the automatic validation of Internet security protocols and applications. We have implemented the AVISPA Tool and tested it on a large corpus of industrial protocols, namely to 215 problems in the AVISPA Library of protocols and properties that we have been formalising. The tool has been made available to the academic and industrial communities. We expect a substantial feedback and many suggestions from users to extend the tool both in the direction of higher expressiveness (more protocols and properties) and adding further analysis techniques, possibly even by connecting new back-ends with complementary features. Several potential back-ends are already on the line. We also plan, in a future project, to embed the tool in a verifying environment for web and other security services.

References

[49] M. Abadi, B. Blanchet, and C. Fournet. Just Fast Keying in the Pi Calculus. In *Proceedings of the 13th European Symposium on Programming (ESOP'04)*, LNCS 2986, pages 340–354. Springer, 2004.

[50] X. Allamigeon and B. Blanchet. Reconstruction of Attacks against Cryptographic Protocols. In *18th IEEE Computer Security Foundations Workshop (CSFW-18)*, Aix-en-Provence, France, June 2005. IEEE Computer Society. To appear.

[51] AVISPA. Deliverable 2.1: The High-Level Protocol Specification Language. Available at `http://www.avispa-project.org/publications.html`, 2003.

[52] AVISPA. Deliverable 1.3: Periodic Progress Report N°: 3. Available at `http://www.avispa-project.org`, 2005.

[53] AVISPA. Deliverable 1.4: Final Project Report. Available at `http://www.avispa-project.org`, 2005.

[54] AVISPA. Deliverable 2.3: The Intermediate Format. Available at `http://www.avispa-project.org/publications.html`, 2003.

[55] AVISPA. Deliverable 2.4: Interface. Available at `http://www.avispa-project.org/publications.html`, 2005.

[56] AVISPA. Deliverable 5.1: Abstractions. Available at `http://www.avispa-project.org`, 2003.

[57] AVISPA. Deliverable 6.1: List of selected problems. Available at `http://www.avispa-project.org`, 2003.

[58] AVISPA. Deliverable 7.4: Assessment of the AVISPA tool v.3. Available at `http://www.avispa-project.org/publications.html`, 2005.

[59] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings of CSFW'01*, pages 82–96. IEEE Computer Society Press, 2001.

[60] B. Blanchet. Automatic verification of cryptographic protocols: A logic programming approach (invited talk). In *Proceedings of PPDP'03*, pages 1–3. ACM Press, 2003.

[61] B. Blanchet. Automatic Proof of Strong Secrecy for Security Protocols. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 86–100. IEEE Computer Society Press, 2004.

[62] C. Bodei, M. Buchholtz, P. Degano, F. Nielson, and H. Riis Nielson. Automatic validation of protocol narration. In *Proceedings of CSFW'03*, pages 126–140. IEEE Computer Society Press, 2003.

[63] L. Bozga, Y. Lakhnech, and M. Perin. Pattern-based abstraction for verifying secrecy in protocols. In *Proceedings of TACAS 2003*, LNCS 2619. Springer-Verlag, 2003.

[64] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In P. Kolaitis, editor, *Proceedings of LICS'2003*. IEEE, 2003.

[65] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of FSTTCS'2003*, Lecture Notes in Computer Science. Springer-Verlag, 2003.

[66] 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`.

[67] CSP — Communicating Sequential Processes. `http://www.formal.demon.co.uk/CSP.html`.

[68] G. Denker, J. Millen, and H. Rueß. The CAPSL Integrated Protocol Environment. Technical Report SRI-CSL-2000-02, SRI International, Menlo Park, CA, October 2000. Available at `http://www.csl.sri.com/~millen/capsl/`.

[69] FDR2 System — Failures-Divergence Refinement. `http://www.formal.demon.co.uk/CSP.html`.

[70] A. Gotsman, F. Massacci, and M. Pistore. Towards an Independent Semantics and Verification Technology for the HLPSL Specification Language. *Electronic Notes in Theoretical Computer Science 135(1):59–77 (Proceedings of the Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA 2005)*, 2005.

[71] J. Groote, S. Mauw, and A. Serebrenik. Analysing the BKE-security protocol with mCRL. Computer Science Report CSR-04-30, Department of Mathematics and Computer Science, Eindhoven University of Technology, 2004.

[72] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In M. Parigot and A. Voronkov, editors, *Proceedings of LPAR 2000*, LNCS 1955, pages 131–160. Springer-Verlag, 2000.

[73] L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, May 1994.

[74] L. Lamport. *Specifying Systems*. Addison-Wesley, 2002.

[75] G. Leduc and F. Germeau. Verification of Security Protocols using LOTOS – Method and Application. *Computer Communications, special issue on "Formal Description Techniques in Practice*, 23(12):1089–1103, 2000.

[76] G. Lowe. Casper: a Compiler for the Analysis of Security Protocols. *Journal of Computer Security*, 6(1):53–84, 1998. See `http://web.comlab.ox.ac.uk/oucl/work/gavin.lowe/Security/Casper/`.

[77] Z. Manna and A. Pnueli. The temporal framework for concurrent programs. In R. Boyer and J. Moore, editors, *The Correctness Problem in Computer Science*, pages 215–274. Academic Press, 1981.

[78] `http://www.w3.org/MarkUp/`. W3c html standard.