Review Report

The AVISPA Team

July 18, 2005

Abstract

In this document we give an executive summary of the activities done since the last review. We discuss how we have addressed the reviewers' recommendations given after the previous evaluation meeting (end of Year 1). Then we give a self-evaluation of the work done and of its compliance with the technical annex. We report no deviations from the initial plan.

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 – Project 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 library and formalisations, 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.

1.1 Deliverables

(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.

(D2.4) 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 [10] 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.

(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.

(D4.5, 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 down-loadable as a single "package" to be installed on the users' local machines. 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.

(D5.2) 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.

(D6.2) Specification of the Problems in the high-level specification language In this deliverable, we present the *AVISPA Formalisations*, 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. 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 Formalisation 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 (D7.3, D7.4) The AVISPA tool has been periodically tested against the AVISPA Formalisations in order to monitor and assess its coverage, effectiveness, and performance. The results of the assessments at project months 24, and 30—presented in Deliverable 7.3 and Deliverable 7.4 respectively—demonstrate the achievement of the project's objectives for the reporting periods. 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 [1]), 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 website (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 website 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 (D8.5, D8.6) In order to disseminate the results of the AVISPA project to academia and industry, we published a large number of papers, gave a large number of conference presentations, talks, tutorials and demos and also organised project workshops.

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 (IJ-CAR'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.

2 Reviewers' Recommendations

We discuss how we have addressed the reviewers' recommendations given after the previous evaluation meeting (end of Year 1).

R1. AVISPA Software Distribution

> *1. Work towards the development of a clearly identifiable AVISPA software distribution, for there are clearly users around the world who would appreciate access, even at this stage of development. At present, the AVISPA tool seems to be a collection of tools available from different places and glued together through the representation formalism, IF. Practitioners from industry will benefit from an integrated solution. For example, a user should be able to define a protocol in HLPSL and verify its properties without being aware of IF and the related intermediate steps.*

We have dedicated a considerable amount of activity to packaging the software developed within the project into the AVISPA Tool v1.0. The AVISPA Tool v1.0 provides a modular and expressive formal language for specifying protocols and their security properties, and integrates four different back-ends that implement a variety of state-of-the-art automatic analysis techniques. The components are not just glued together; rather, we have invested a lot of effort to find define the common input specification language in such a way that it has a well-defined semantics and also allows for the specification of various security properties such as different forms of authentication and secrecy, as well as other properties. Moreover, we have defined a common output format for reporting the results of the analyses and it is now supported by all the back-ends integrated into the AVISPA Tool. This facilitates the understanding of the output produced by the different back-ends, and allows also for the generation of message-sequence charts and postscript files depicting the attacks the back-ends have found.

The package also comprises a tutorial (53 pages), a number of detailed specification examples, and a user-manual (86 pages). A special emacs mode has been designed and implemented in order to ease the writing of protocol specifications. Moreover we have set up a mailing list so that users can ask questions, provide comments, or report on bugs that they have found.

We have tested the AVISPA Tool v1.0 on various computer platforms. After a preliminary phase of beta testing, restricted to a number of known users, the first official release of the package, the AVISPA Tool v1.0, has been publicly announced and made available for downloading on the AVISPA web-site on June 14, 2005. We maintain a list of users: so far, we have had 56 downloads and 45 people are officially registered as AVISPA users.

We have also considerably enhanced the web-interface of the AVISPA Tool (available at the URL: `http://www.avispa-project.org/web-interface/`) so to enable even inexperienced users to get immediate access to the functionalities of the AVISPA Tool through their favourite web-browser and without having to install any software on their computers. The web-interface has been carefully designed so to be used effectively by users with different levels of expertise. It supports a *basic mode*, in which the user does not need to know any detail about the actual architecture of the tool or the intermediate format IF (which is the low-level language into which input specifications written in the high-level language HLPSL are automatically translated by the HLPSL2IF translator), and an *expert mode* that allows more experienced users to select the most suited back-end and/or settings for the problem at hand.

R2. Standing of AVISPA with respect to other Approaches

> *2. Clarify the standing of AVISPA with respect to other approaches in formal verification of security. Analyze the pros and cons of the AVISPA approach. Specifically, contrast the pros and cons of tailored model checking (AVISPA) with respect to general model checking.*

The most important alternative approaches to AVISPA's are interactive theorem proving and finite-state model-checking. With interactive theorem proving using a proof assistant like Isabelle or Coq one has the advantage that more complex protocols and security goals can be handled. For instance, the verification of contract-signing protocols where security properties such as fairness cannot be simply encoded in the protocol specification language currently supported by the AVISPA Tool but it can be encoded in Isabelle's logic, as experimented by Vu Van, a master student at INRIA. On the other hand the verification with such proof-assistant still lacks automation and is quite time consuming, even for simple protocols. Finite-state model checking is as efficient as AVISPA Tool on small protocols however it seems to fail with large ones. For instance the Scyther system of S. Mauw group [5] works well for secrecy compared to Brutus, Casper, and Constraint Logic systems but does not handle authentication. The [5] reference reports experiments with Scyther only for a 3 step Bilateral Key Exchange protocol and up to 7 sessions. Moreover, with finite-state model-checking the absence of flaw detection even when considering finite sessions (scenarios) does not guarantee at all the absence of flaws in the Dolev-Yao model, due to several approximations (e.g. message size) performed to run the specification. There are some completeness results by Gavin Lowe [8] but they are very restricted (as they apply to almost no realistic protocol). Moreover it seems non trivial to integrate algebraic properties with finite-state model-checking.

R3. Scope of AVISPA

> *3. Try to characterize the different aspects of security that might be handled by a (any) formal verification tool, and clearly identify which of them can be analyzed using AVISPA. Give reasons not to consider the aspects left out.*

Our tools are well-suited for analysing message passing processes with crypted data, but they are not suited for analysing QoS (time) or memory resources. In particular, our tools are well-suited for proving trace-based properties that can be encoded as reachability properties such as secrecy and authentication. More precisely:

- Security often relies on timeliness issues, especially in the case of embedded systems: real-time systems should satisfy some hard delay constraints. These aspects can be handled using some extension of model-checking but are not covered by our approach since it is usually not a major issue in the case of most security protocols. More generally, our tools cannot handle DoS attacks, at least not directly, and thus this topic is left for future research. For instance, it would be interesting to analyse attacks that result in downgrading the quality of service in the case of video-on-demand services (pay-per-view protocols).

- Some security issues are related also to the protection of resources like memory. It is often the case that leaks in security protocols lead to buffer-overflow attacks. A considerable amount of work has been devoted to using static analysis techniques to verify the part of memory that is written. Our tool is better suited for analysing message-passing communicating systems with crypted data. A possible interesting future work would be to combine the two types of analysis. A good candidate for that would be the back-end TA4SP, which uses tree automata techniques for over-approximating sets of messages as it might be combined with a tool for approximating the reachable part of memory.

- New protocols for electronic commerce, such as contract-signing protocols, require to check properties like fairness or abuse-freeness that are not trace-based ones (they need to envisage several executions simultaneously) and are more of a game-theoretic nature: a participant should not gain some advantage on the other one when signing the contract. In general these properties are difficult to verify with automated tools. But some experiments have been successful performed using Murphi in [9] and in AVISPA using OFMC in [6]. There is also a possibility to get automated proofs for more complex fairness properties thanks to the recent results by [7] that give effective and complete verification algorithms for contract signing protocols.

R4. Dissemination

> *4. Strengthen the efforts on dissemination of the AVISPA project, results and the toolset capabilities to industrial/professional engineers, especially those new to the field of security - the latter being a group which is most likely to gain through the use of the analysis tools.*

The results of the project are useful to a wide audience ranging from researchers interested in the theoretical issues to IT professional interested in applications with or without a background in computer security. We have then identified four different target audiences, namely:

- Researchers working in formal verification or automated reasoning both with no specific expertise in computer security, e.g. researchers working in formal methods, automated reasoning, and software engineering.
- Researchers with specific expertise in computer security and formal methods, e.g. researchers working at the development of techniques for the formal analysis of security protocols or applications.
- IT professionals with no specific expertise in security or formal methods, e.g. engineers working on the design, implementation, deployment, or assessment of communication protocols or infrastructures.
- IT professionals with expertise in security, e.g. engineers working at the design, development, deployment, or certification of security protocols or infrastructures.

We have thus differentiated our dissemination activities accordingly. The distribution of the dissemination activities over the different target audiences is summarised in Table 1.

Specific measures have been undertaken in order to reach IT professionals as recommended: we have given talks and demos of the AVISPA Tool to several companies involved in the development of security protocols: France Telecom R&D, Kotio, SAP, Thomson Multimedia R&D, Credit Suisse, UBS, IBM Switzerland, Microsoft U.K., ... This proved to be an effective means to show them the potential of our tool for validating the security of their products. As a matter of fact, several new collaborations have stemmed from these presentations and new research projects are currently being prepared.

Table 1: Dissemination activities for different types of audience

	Security Expert?	
	no	yes
Researchers	• 3 tutorials • 1 invited talk at IRST • system description at CAV'05 • web interface • AVISPA package • AVISPA web site	• 28 paper presentations • 33 publications • editing of 4 proceedings or special issues of journals • 14 invited talks • 2 PhD Theses • 3 project workshops • AVISPA package • AVISPA web site
IT Professionals	• 3 tutorials • 2 invited talks at IETF • tool demos • web interface • AVISPA web site	• 3 tutorials • 2 invited talks at IETF • tool demos • web interface • AVISPA package • AVISPA web site

Since the last review meeting, we have also organised 2 workshops, 3 tutorials, we have presented the AVISPA Tool at the 17th International Conference on Computer Aided Verification (CAV'05), and we have presented the results of the project in 2 invited talks at the Open Security Area Directorate (SAAG) meetings held in the context of the 59th and 62nd meetings of the IETF.

For a detailed description of the dissemination activity carried out since the last review meeting please refer to [2, 3, 4].

3 Self-evaluation of the work done

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.

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 library, contains a total of 384 security problems and 79 protocols, mostly from the IETF, divided into 33 groups.

We have used the AVISPA library 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 library 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 [1], 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 property is violated.

Table 2: 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)

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 2, we have been able to formalise in the HLPSL 215 problems from 22 groups (including 38 problems from the seven main groups described in [1]), 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 workpackage (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: 33 papers have been published in international conferences and journals; 2 PhD theses are about to be completed and 4 PhD student 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 [1], has been made available to the IETF and discussed with the security area directors. We have also presented the results of the Project during an invited presentation at the Open Security Area Directorate Meeting held in the context of the 62nd Meeting of the IETF.

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.

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.

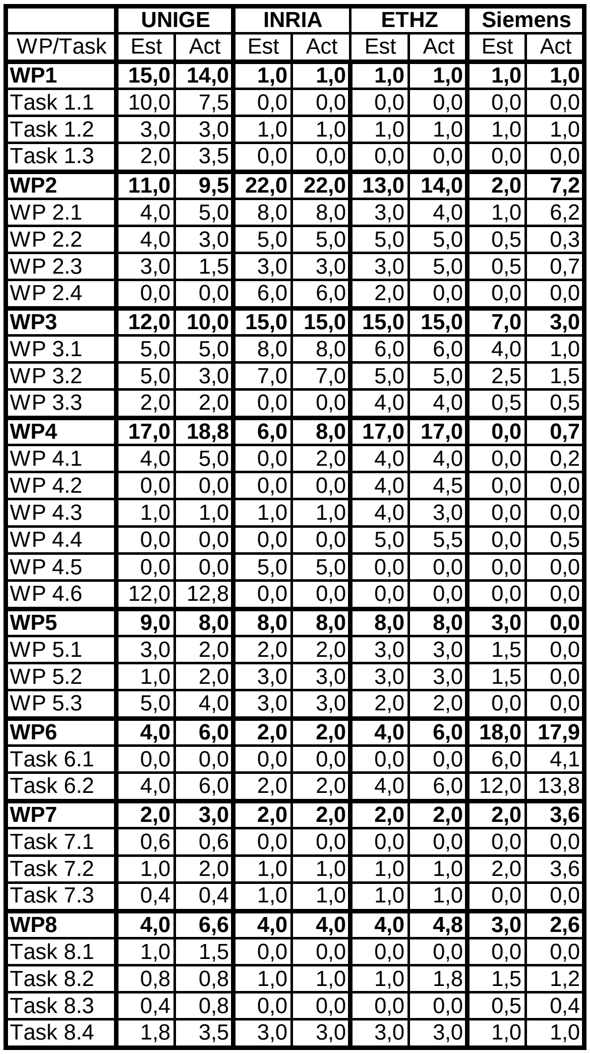
4 Effort spent per partner

The effort (in person months) spent per partner on each task and the total effort spent on each deliverable are summarised in Table 3 and in Table 4 respectively.

5 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.

Table 3: Breakdown of the effort (in person months) spent per partner on each task



	UNIGE		INRIA		ETHZ		Siemens	
WP/Task	Est	Act	Est	Act	Est	Act	Est	Act
WP1	**15,0**	**14,0**	**1,0**	**1,0**	**1,0**	**1,0**	**1,0**	**1,0**
Task 1.1	10,0	7,5	0,0	0,0	0,0	0,0	0,0	0,0
Task 1.2	3,0	3,0	1,0	1,0	1,0	1,0	1,0	1,0
Task 1.3	2,0	3,5	0,0	0,0	0,0	0,0	0,0	0,0
WP2	**11,0**	**9,5**	**22,0**	**22,0**	**13,0**	**14,0**	**2,0**	**7,2**
WP 2.1	4,0	5,0	8,0	8,0	3,0	4,0	1,0	6,2
WP 2.2	4,0	3,0	5,0	5,0	5,0	5,0	0,5	0,3
WP 2.3	3,0	1,5	3,0	3,0	3,0	5,0	0,5	0,7
WP 2.4	0,0	0,0	6,0	6,0	2,0	0,0	0,0	0,0
WP3	**12,0**	**10,0**	**15,0**	**15,0**	**15,0**	**15,0**	**7,0**	**3,0**
WP 3.1	5,0	5,0	8,0	8,0	6,0	6,0	4,0	1,0
WP 3.2	5,0	3,0	7,0	7,0	5,0	5,0	2,5	1,5
WP 3.3	2,0	2,0	0,0	0,0	4,0	4,0	0,5	0,5
WP4	**17,0**	**18,8**	**6,0**	**8,0**	**17,0**	**17,0**	**0,0**	**0,7**
WP 4.1	4,0	5,0	0,0	2,0	4,0	4,0	0,0	0,2
WP 4.2	0,0	0,0	0,0	0,0	4,0	4,5	0,0	0,0
WP 4.3	1,0	1,0	1,0	1,0	4,0	3,0	0,0	0,0
WP 4.4	0,0	0,0	0,0	0,0	5,0	5,5	0,0	0,5
WP 4.5	0,0	0,0	5,0	5,0	0,0	0,0	0,0	0,0
WP 4.6	12,0	12,8	0,0	0,0	0,0	0,0	0,0	0,0
WP5	**9,0**	**8,0**	**8,0**	**8,0**	**8,0**	**8,0**	**3,0**	**0,0**
WP 5.1	3,0	2,0	2,0	2,0	3,0	3,0	1,5	0,0
WP 5.2	1,0	2,0	3,0	3,0	3,0	3,0	1,5	0,0
WP 5.3	5,0	4,0	3,0	3,0	2,0	2,0	0,0	0,0
WP6	**4,0**	**6,0**	**2,0**	**2,0**	**4,0**	**6,0**	**18,0**	**17,9**
Task 6.1	0,0	0,0	0,0	0,0	0,0	0,0	6,0	4,1
Task 6.2	4,0	6,0	2,0	2,0	4,0	6,0	12,0	13,8
WP7	**2,0**	**3,0**	**2,0**	**2,0**	**2,0**	**2,0**	**2,0**	**3,6**
Task 7.1	0,6	0,6	0,0	0,0	0,0	0,0	0,0	0,0
Task 7.2	1,0	2,0	1,0	1,0	1,0	1,0	2,0	3,6
Task 7.3	0,4	0,4	1,0	1,0	1,0	1,0	0,0	0,0
WP8	**4,0**	**6,6**	**4,0**	**4,0**	**4,0**	**4,8**	**3,0**	**2,6**
Task 8.1	1,0	1,5	0,0	0,0	0,0	0,0	0,0	0,0
Task 8.2	0,8	0,8	1,0	1,0	1,0	1,8	1,5	1,2
Task 8.3	0,4	0,8	0,0	0,0	0,0	0,0	0,5	0,4
Task 8.4	1,8	3,5	3,0	3,0	3,0	3,0	1,0	1,0

Table 4: Effort (in person months) spent per deliverable

No.	Title	Person months
1.1	*Year 1 Progress Report*	*0.5*
1.2	*Year 2 Progress Report*	*1*
1.3	*Year 3 Progress Report*	*1*
1.4	*Final Project Report*	*1*
2.1	*HLPSL*	*20*
2.2	*Algebraic properties*	*15*
2.3	*IF*	*16*
2.4	*Interface*	*2*
3.1	*Security Properties*	*19*
3.2	*Assumptions on environment*	*20*
3.3	*Session instances*	*10*
4.1	*Compositionality*	*8*
4.2	*Partial-Order Red.*	*6*
4.3	*Heuristics*	*5*
4.4	*AVISPA tool v.1*	*10*
4.5	*AVISPA tool v.2*	*10*
4.6	*AVISPA tool v.3*	*2*
5.1	*Abstractions*	*14*
5.2	*Infinite-state Model Checking*	*9*
5.3	*Completeness issue*	*7*
6.1	*List of Selected Problems.*	*3*
6.2	*Spec. in HLPSL*	*14*
7.1	*Experimental Setup*	*1*
7.2	*Assessment of tool v1*	*1*
7.3	*Assessment of tool v2*	*5*
7.4	*Assessment of tool v3*	*3*
8.1	*AVISPA website*	*2*
8.2	*Project presentation*	*1*
8.3	*Dissemination and Use Plan*	*0.5*
8.4	*Year 1 Project WS*	*0.3*
8.5	*Year 2 Project WS*	*0.3*
8.6	*Year 3 Project WS*	*0.5*
8.7	*Technology and Implementation Plan*	*1*

References

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

[2] AVISPA. Deliverable 1.2: Periodic Progress Report N°: 2. Available at `http://www.avispa-project.org`, 2004.

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

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

[5] S. Cremers, C.J.F.; Mauw. Checking secrecy by means of partial order reduction. In A. Amyot, D.; Williams, editor, *Sam 2004: security analysis and modelling*, volume LNCS 3319 of *4th Workshop on SDL and MSC*, pages 177–194, Ottawa, Canada. Springer-Verlag, Berlin.

[6] P. Hankes Drielsma and S. Mödersheim. The ASW protocol revisited: A unified view. In *Proceedings of the IJCAR04 Workshop ARSPA*, 2004. To appear in ENTCS, available at `http://www.avispa-project.org`.

[7] D. Kähler and R. Küsters. Constraint Solving for Contract-Signing Protocols. In *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR 2005)*, 2005. To appear.

[8] G. Lowe. Towards a completeness result for model checking of security protocols. In *Proceedings of the 11th IEEE Computer Security Foundations Workshop (CSFW'98)*, pages 96–105. IEEE Computer Society Press, 1998.

[9] V. Shmatikov and J. C. Mitchell. Finite-state analysis of two contract signing protocols. *Theoretical Computer Science*, 283(2):419–450, 2002.

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