INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

Contract for:

Shared-cost RTD (FET Open) Project

Annex 1 - Description of Work

Project acronym: AVISPA
Project full title: Automated Validation of Internet Security Protocols and Applications
Proposal no: IST-2001-39252
Contract no: IST-2001-39252
Related to other Contract no.:

Date of preparation of Annex 1: July 14, 2005

Version: 2

Operative commencement date of contract: 01/01/2003

Contents

1 Project summary

1.1 Abstract

This project aims to develop a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. This technology will speed the development of the next generation of network protocols, improve their security, and therefore increase the public acceptance of advanced, distributed IT applications based on them. We will achieve this by advancing specification and deduction technology to the point where industry protocols can be specified and automatically analyzed. This technology will be integrated into a robust automated tool, tuned on practical, large-scale problems, and migrated to standardization bodies, whose protocol designers are in dire need of such tools.

1.2 Objectives

This project aims to develop techniques and tools for the analysis of security-sensitive protocols, required to support the next generation of distributed, Internet applications. The main objectives are five fold. First, to develop a rich specification language for formalizing protocols, security goals, and threat models of industrial complexity. Second, to advance the state-of-the-art in automated deduction techniques to scale up to this complexity. Third, to build a tool based on these techniques that will allow industry and standardization organizations to automatically validate or detect errors in their products. Fourth, to tune this tool and demonstrate proof-of-concept on a large collection of practically relevant, industrial protocols. And finally, to begin the migration of this technology into industry standardization organizations such as the IETF so that both the scientific and the industrial community can benefit from the advances achieved by this project.

1.3 Description of the Work

The work will be carried out by accomplishing the following tasks.

- We will design a high-level language for specifying Internet security protocols, and implement a translator from protocol descriptions to a declarative format amenable to formal analysis. The language will support the description of Internet protocol suites, security goals, and assumptions about the environment.
- We will develop a technology for automated protocol error detection based on three automated deduction techniques operating on the translator's output. The first technique, on-the-fly model-checking, uses lazy data-types and specialized algorithms that can automatically handle infinite state spaces; it will be backed up by powerful search heuristics. The second technique, theorem-proving with constraints, provides an efficient way of representing an infinite state-space using a constraint store, and supports the specification of built-in theories for cryptographic operators. The third technique employs model-checking methods based on propositional satisfiability checking that efficiently find errors in protocols by reducing an approximation of the problem to a propositional satisfiability problem. Although each technique can work independently, they will be integrated into a single analysis tool, AVISPA, where they will interact and benefit from each other's strengths.
- To verify protocols we will develop techniques for infinite-state verification, like use of abstractions and infinite-state symbolic model-checking, and integrate them in our tool. To avoid combinatorial blow-up in search, for both verification and falsification, we shall exploit the fact that Internet protocols are often built, compositionally from subprotocols and we will develop compositional reasoning techniques.
- A set of representative security problems drawn from IETF drafts will be selected and used to thoroughly evaluate the AVISPA tool according to well-defined and measurable criteria.

1.4 Milestones and Expected Results

The AVISPA tool will be tested against a library of problems drawn from 26 protocol groups developed by the IETF. As main results, we expect:

1. Specification of at least 80 problems taken from 20 protocol groups.

2. Analysis of at least 75% of the specified problems in under 1 CPU-hour per problem.

The main milestones monitoring the project are the delivery and assessment of 3 versions of the AVISPA tool at months 12, 20 and 27, with a final milestone on the project's success at month 30.

2 Project Objectives

2.1 Context and Motivation

Our economic, political, and social life today depends vitally on communication and IT infrastructures, in particular the Internet. The acceptance and continued expansion of these infrastructures depends on trust: all participants must have confidence in their security, which is integrated into the infrastructure either by means of specific security protocols or protocol families (cf. protocols like IPSec, IKE, TLS, and SSH) aimed at satisfying general security requirements, or by augmenting protocols and applications with security sensitive parts (cf. Geopriv, SIP, etc.), i.e. security specific functional elements that satisfy application-oriented security requirements. With the spread of the Internet and network based services, and the development of new technological possibilities, the number and scale of new protocols under development is out-pacing our ability to rigorously analyze and validate them. This is an increasingly serious problem for standardization organizations like the *Internet Engineering Task Force* (IETF), the *International Telecommunication Union* (ITU), and the *World Wide Web Consortium* (W3C) as well as for companies whose products and/or services depend on the rapid standardization and correct functioning of these protocols. It is also a problem for users of these technologies whose rights and freedoms, e.g. the right to privacy of personal data, depend on a secure infrastructure.

Designing secure protocols is a very challenging problem. In open networks, such as the Internet, protocols should work even under worst-case assumptions, namely messages may be eavesdropped or tampered with by an *intruder* (also called the attacker or spy) or dishonest or careless principals (where we call *principals* the agents participating in a protocol execution). Surprisingly, severe attacks can be conducted even without breaking cryptography, but by exploiting weaknesses in the protocols themselves, for instance by carrying out *man-in-the-middle attacks*, where an attacker plays off one protocol participant against another, or *replay attacks*, where messages from one session (i.e. execution of the protocol) are used in another session. The possibility of these attacks sometimes stems from subtle misconceptions in the design of the protocols. Typically, these attacks are simply overlooked, as it is difficult for humans, even by careful inspection of simple protocols, to determine all the complex ways that different protocol sessions could be interleaved together, with possible interference of a malicious intruder.

To speed up the development of the next generation of network protocols and to improve their security and ultimately the acceptance of products based on them, it is of utmost importance to have tools that support the activity of finding flaws in protocols or of establishing their absence. Optimally, these tools should be robust, expressive, and easily usable, so that they can be integrated into the protocol development and the standardization process to improve the speed and quality of that process.

2.2 Objectives

The overall goals of the AVISPA project are

- to develop a rich specification language for formalizing protocols, security goals, and threat models of industrial complexity,
- to advance the state-of-the-art in automated deduction techniques to scale up to this complexity,
- to build a tool based on these techniques that will allow industry and standardization organizations to automatically validate or detect errors in their products,
- to tune this tool and demonstrate proof-of-concept on a large collection of practically relevant, industrial protocols, and
- to begin the migration of this technology into industry standardization organizations 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.

The AVISPA consortium consists of four partners with complementary scientific and practical competence: a group from the University of Genova UNIGE, a group from the Institut National de Recherche en Informatique et en Automatique of Nancy INRIA, a group from the Swiss Federal Institute of Technology of Zurich ETHZ, and a group from the Siemens Aktiengesellschaft SIEMENS (CO1, CR2, CR3, and CR4, respectively). The partners are leaders in their respective areas and have a long and successful history of international collaboration and strong bilateral relations. While the SIEMENS partner is among the leaders in the design of protocols and actively participates in their standardization at IETF, ITU, 3GPP, and W3C (and also has experience in utilizing formal methods for the validation of protocols and applications [93, 131, 89, 91]), the partners UNIGE, INRIA and ETHZ have previously carried out extensive joint work that lays the foundations of the proposed technology; see, for example, [5, 125] as well as the description of the FET-Open Assessment Project IST-2000-26410, "AVISS: Automated Verification of Infinite State Systems" in Section 14. The three partners have developed a prototype verification tool based on three complementary automated deduction techniques: on-the-fly model-checking using lazy data-types, theorem-proving with constraints, and model-checking based on propositional satisfiability checking. In this tool, as it is often done, a security protocol and the properties it should satisfy, as well as the intruder activities, are modeled as an infinite state transition system. The first technique of the prototype tool, on-the-fly model-checking, uses lazy data-types and specialized algorithms that can automatically handle infinite state-spaces. The second technique, theorem-proving with constraints, provides an efficient way of representing an infinite state space using a constraint store. The third technique employs model-checking methods based on propositional satisfiability checking that efficiently find errors in protocols by reducing an approximation of the problem (which is iteratively improved) to a propositional satisfiability problem.

The effectiveness of these approaches was demonstrated by the application of the resulting system to the Clark/Jacob library [48], a well-known library of 51 authentication protocols of small and medium size. For detecting errors, our prototype tool is better than all other existing analysis tools worldwide, in that it has either better coverage or better performance, or both. In particular, our prototype tool can detect many subtle attacks (e.g. based on typing ambiguities) that are missed by most other tools. (For more details on related and previous work, see Section 5.)

However, whether the approach scales-up to the new generation of large-scale, security-sensitive protocols remains a challenge. Rising to this challenge leads us to investigate:

1. Methods for specifying large-scale protocols, security properties and goals, and models of intruders and threats.

2. Complementary techniques based on automated deduction and abstraction for efficiently detecting flaws in these protocols or for verifying them.

2.3 Operational Goals

Given a protocol validation problem expressed in a high-level specification language (e.g. a description of a protocol together with a security property to be checked), the AVISPA tool will first translate the problem description into an operational semantics description, which is further processed by inference engines implementing the selected automated deduction techniques. Whenever the input protocol is flawed and when the analysis carried out by the tool completes successfully, the tool will return an execution trace witnessing an attack on the protocol. Alternatively, proofs of security properties satisfied by the protocol may be provided by the verification tool using abstraction and infinite-state model-checking techniques.

More technically the operational objectives of the AVISPA project are:

- To design a high-level specification language for the faithful description of security-sensitive protocols, security properties and goals, and intruder/threat models. The language will be extended with constructs to express control-flow, data-structures, and operator properties, and to allow for the flexible specification of assumptions on environments and on sessions and properties of operators.

- To develop a compiler that maps the high-level problem descriptions into an intermediate format that is suitable for analysis by complementary automated deduction techniques for falsification (i.e. for error-detection).
- To develop three back-ends that carry out such analyses and enhance them with new search techniques. The compiler and the back-ends, suitably wrapped by a user-friendly graphical user interface (GUI), will constitute the AVISPA tool.
- To incorporate in the tool verification techniques for establishing the absence of flaws.
- To build and make publicly available a library of formalized IETF protocols and associated security problems called "the AVISPA library". The protocols and problems will be selected in such a way to be representative of the many protocol groups currently being developed by the IETF. The library will be proposed to 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.
- To apply the AVISPA tool against the AVISPA library and quantitatively assess its coverage, effectiveness, and performance.
- To analyze the experiments in order to establish a (best practice) methodology for future AVISPA tool users. This will include a classification of the complementary analysis techniques implemented in the back-ends as well as a characterization of the protocols and properties that each back-end is likely to successfully handle.

2.4 Success Criteria

Currently, very few large-scale protocols have been formally analyzed and even fewer with automated tools, and in those cases where automatic analysis was performed, the level of abstraction was quite high [109, 32, 103]. Considerable work has been devoted to the application of formal methods to protocol analysis in academia, but no method seems to be generally adopted and the only available benchmarks concern small and medium scale protocols, such as the protocols in the Clark/Jacob library [48].

In order to measure the success of the project, the tool developed by the partners will be thoroughly tested against the AVISPA library, a library of security problems drawn from a large number of protocols developed by the IETF (see Section 15 for a survey). Since these protocols are beyond the reach of current formal methods, a measure of the success of our approach will be the security problems from this collection that can be successfully analyzed with our tool. In particular, the success criteria for the project are:

Coverage: at least 80 security problems taken from 20 of the 26 IETF groups of protocols (cf. Section 15) should be specifiable in the high-level specification language.

Effectiveness: the tool should either falsify or verify at least 75% (i.e. 60) of the problems specified in the high-level specification language. Moreover the set of successfully analyzed problems should come from one protocol of each of the Main Protocols Groups listed in Section 15.

Performance: the processing of the successfully analyzed security problems should take less than 1 hour of CPU time per problem on standard commercially available computers.

Notice that each of the above criteria, even when considered in isolation, is far beyond the scope of state-of-the-art tools for the formal analysis of security protocols. Therefore the successful completion of the project represents a substantial improvement in the state-of-the-art for this domain.

3 Participant List

LIST OF PARTICIPANTS						
Role	**Number**	**Participant name**	**Participant short name**	**Country**	**Date enter project**	**Date exit project**
CO (C)	1	Università di Genova	UNIGE (or CO1)	IT	Start of project	End of project
CR (P)	2	Institut National de Recherche en Informatique et en Automatique	INRIA (or CR2)	F	Start of project	End of project
CR (P)	3	Swiss Federal Institute of Technology, Zurich	ETHZ (or CR3)	CH	Start of project	End of project
CR (P)	4	Siemens Aktiengesellschaft	SIEMENS (or CR4)	DE	Start of project	End of project

4 Contribution to Programme/Key Action Objectives

The AVISPA project is well positioned within the FET OPEN scheme as the technology we will develop for the automated security analysis of Internet protocols and applications will strongly contribute to developing new applications and services and it will improve the trust and confidence in them. In particular, it will contribute to "reinforcing *e*Europe objectives beyond 2002", as called for in the IST Workprogramme 2002 [82, p. 6]. The AVISPA project addresses a number of the programme's priorities, including:

- Network management, interoperable networks and distributed systems.
- Take-up activities in e-commerce, e-work, and security and privacy.
- New paradigms for *e*business, *e*work, and trust and confidence solutions.
- New concepts for knowledge and interface technologies.
- Systems to improve security and for more efficient crisis management.
- Next generation mobile systems and networked audio-visual systems.

More specifically, the AVISPA project will contribute not only to the

- Action Line VI.1.1 – Open domain.

of the

- Future and emerging technologies

activity of the WP2002 of the IST, but also to the other programme action lines:

- II.1 – Addressing *e*Europe and *e*Europe+ objectives.
- IV.2 – Computing communications and networks.
- IV.3 – Technologies and engineering for software, systems and services.
- IV.4 – Real-time and large-scale simulation and visualization technologies.
- IV.5 – Mobile and personal communications and systems (including, in particular, the validation of wireless and mobile systems and technologies).

as well as to the cross-programme theme

- V.1.14 CPA14 – Mobile applications and services.

The AVISPA project also addresses existing European policy objectives, as it contributes to standardization and industrial consensus of (security-sensitive) Internet protocols and applications. (See also Section 6 "Community added value and contribution to EU policies".) The project is also closely related to the themes of WP2002 where security-sensitive applications are developed or exploited, and where system failures or lack of properties such as confidentiality, integrity or authorization could have drastic or even life-threatening consequences. For example, we see a great potential for the exploitation of the results of our project in the lines of the "Key Action I – Systems and services for the citizen", such as line "I.1.1. – Intelligent systems for monitoring of health status".

The ideas underlying the new technology that we will develop in the AVISPA project are highly innovative and their realization is highly risky and requires longer term research. The project thus fits well into the objectives of the FET action line "VI.1.1 – Open domain", which is focussed on developing new technologies for significant breakthroughs in industrial and societal terms. In particular, the use of a multi-technology approach like ours requires a research effort with a truly European dimension: The expertise required by the AVISPA project cannot be found in a single national research group or site. Indeed, it would be very difficult to pursue the technical and socio-economical objectives of the proposed project without the participation of different research groups, from both academia and industry. The development and exploitation of the techniques to be applied in the project require a considerable variety and a high degree of advanced technical and technological skills. The partners of the AVISPA consortium are leading European experts both in the different automated deduction techniques upon which the project is based, and/or in the design and analysis of protocols and applications. Moreover, the partners have a common background on automated verification strengthened by a long and successful history of international collaboration and strong bilateral relations. The AVISPA consortium, therefore, collects the set of complementary technological skills required to tackle such a high-risk project, to a degree only possible in such a multi-national cooperation.

Let us expand on the question of coveraging by giving several examples showing how the AVISPA project addresses different action lines. First, the technology that we will develop will support activities to progress the *e*Europe and *e*Europe+ initiative of "Action line II.1" as the validation techniques and tools will contribute to the development of information society technologies that enable the realization of secure infrastructures for the electronic marketplace. This will exploit European strengths such as electronic payments, smart cards, mobile systems, and software for business process modeling and enterprise management and consumer protection. Second, the AVISPA project will contribute to the "Key Action IV – Essential technologies and infrastructures", in particular to the four action lines listed above, as our work will contribute to, and support, today's European strengths in communications and network technologies, digital broadcasting, consumer electronics, software and embedded systems. It will thus contribute to the development of the next generations of the essential component technologies, integrated systems, and networks underpinning today's converging industries and infrastructures. Finally, our project will strongly contribute to the cross-programme theme "V.1.14 CPA14 – Mobile applications and services" as our validation technology will support the development of secure user-friendly applications and services that take full advantage of the possibilities of leading-edge mobile and wireless technologies, including third generation mobile systems and wireless LANs, in the framework of a transition from IPv4 to IPv6 based provisioning.

Last but not least, the project will train young researchers in the European Union on state-of-the-art techniques for reasoning about systems, protocols, and IT security.

All these contributions of the AVISPA project are discussed in more detail in the following sections.

5 Innovation

5.1 The Challenge

Experience over the last twenty years has shown that, even assuming perfect cryptography, the design of protocols for secure electronic communication and transactions is highly error-prone, and that conventional validation techniques based on informal arguments or testing are not up to the task. High quality requirements, stemming from the need to protect the rights of individuals (e.g. data privacy), have resulted in

the promotion and standardization of rigorous evaluation criteria (such as ITSEC and the Common Criteria [126]). Formal methods and formal proofs are generally considered to ensure the highest level of trustworthiness. But applying formal methods usually involves considerable effort, both for formalization and for proof. To apply formal methods on a large-scale, to complex protocols, requires the development of new automated techniques.

The challenge for the AVISPA project is to develop a new push-button technology for formally analyzing the security of protocols, and thereby drastically reduce the amount of work and the costs necessary to analyze the security of the new generation of Internet protocols. We aim to develop a technology so automated and robust that it can be routinely applied to advanced protocols, like those designed by the IETF, both speeding up this process and producing higher-quality results. Hence, the AVISPA project aims at pushing the boundaries of automated analysis based on the following innovations:

- AVISPA will create a high-level protocol specification language and modeling environment for the formal description of protocols, their security properties, and assumptions on their environment. This language must be very expressive to allow the formulation of a large class of realistic protocols, threats, and environmental assumptions. To decrease the distance from the protocol model to actual implementations we will go beyond the standard assumption that "cryptography is perfect", where properties of cryptographic operators are typically abstracted away.
- We will push-forward the state-of-the-art on different automated deduction techniques, incorporated in the analysis back-ends (which operate on the intermediate format produced by a specification compiler). Achieving the ambitious success criteria will require significant innovations and advancements in the different deduction techniques; we must achieve improvements of many orders of magnitude on the prototype previously developed (see [5, 125] and Section 14), which itself was state-of-the-art in terms of coverage and performance for the domain of authentication protocols (i.e. for the small and medium scale protocols in the Clark/Jacob library [48]). We must also develop and advance the state-of-the-art for techniques like abstraction and model-checking for verifying infinite-state systems. To do this, we will carry out innovative research, as well as investigate optimizations, abstractions and generalizations of the prototype specification and validation techniques.
- The uniqueness of the AVISPA project also lies in its challenging goal of analyzing large real-life protocols from the IETF drafts, with the same automated tool, in a uniform way. A breakthrough would be to influence the design of new generations of IETF protocols by the early detection of potential weakness or through their verification. An even larger breakthrough would be to have a tool powerful and successful enough to influence the standardization process itself, leading to the long-term integration of formal methods tools into this process.

5.2 State of the Art

We now briefly describe the state of the art in the design and analysis of Internet security protocols and applications, against which the innovation of the AVISPA project can be judged. We first contrast our proposed contributions with the capabilities of the current generation of security protocol analysis tools. Afterwards, we consider the new generation of security-sensitive protocols and applications, in particular those put forward by the IETF.

The current generation of security protocol analysis tools

The current generation of formal methods tools for the analysis of security protocols are mostly based on either interactive verification or automatic finite-state model-checking [35, 36, 39, 41, 43, 49, 66, 67, 88, 96, 100, 101, 102, 108, 112, 113, 132]. The interactive tools require a considerable investment of time by expert users and provide no support for error detection when the protocols are flawed. The automatic tools generally require strong assumptions that bound the information analyzed, so that an infinite-state system can be approximated by a finite-state one. Moreover, the current level of automated support scales poorly and is insufficient for the validation of realistic protocols. Finally, tuning the specification (such as building relevant approximations) and tuning the tool's parameters often requires significant expertise too.

Probably the largest restriction on both kinds of tools is their current scope. The vast majority of these techniques are tested on a limited number of small to medium scale protocols. Moreover, the vast majority of protocols considered are authentication protocols rather than more complex e-commerce protocols or, more generally, security-sensitive protocols. There are no tools that have been applied to problems of the breadth and complexity planned for the AVISPA tool. Moreover, there are no tools that effectively combine advanced technologies for both verification and falsification (flaw detection).

We now focus on several representatives of the state of the art that (1) are automatic, (2) have features similar to those of the prototype tool [5, 125] of previous work of the first three project partners, which is described in detail in Section 14, and (3) share similarities with the approach underlying the AVISPA project. To begin with, only the CAPSL [66] environment, developed at SRI International (Menlo Park, CA) in the context of a DARPA-project, is designed to support 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 text-books and by engineers and a low-level language that provides an interface to different back-ends for protocol analysis. In [66], methods are given for integration of CAPSL with SRI's own PVS [110] and Maude [98], with the special-purpose NRL protocol analyzer [40], and with the automatic verifier Athena [123]. There are a number of important differences. The most important ones are that the CAPSL environment only focuses on authentication protocols and that it has only very limited (published) results with its different backends.

As we previously mentioned, we have employed the Clark/Jacob library [48], which consists of 50 authentication protocols of small and medium size, as a benchmark for the prototype techniques and tools we have developed previously. For detecting errors, our prototype tool is better than all other existing analysis tools worldwide, in that it has either better coverage or better performance, or both. In fact, based on the available experiments and the results in [66], our prototype is considerably more effective on the library than CAPSL and its current connectors. 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.

We have presented our prototype to the main developers of the CAPSL environment, Grit Denker and Jonathan Millen from SRI International, and they have expressed their interest in our project and have suggested possible collaborations.

The Casper tool [67, 96], developed by Gavin Lowe and his group, is the only tool that both allows for the specification of almost all protocols in the Clark/Jacob library and detects a number of attacks close to those detected by our prototype. Casper, however, does not support non-atomic keys, which is a real drawback for extending this approach to more ambitious protocols of the kind considered in the AVISPA project. Moreover, the Casper approach is oriented towards finite-state verification by model-checking with CSP and FDR2 [52, 71], and thus requires a number of strong assumptions for bounding information necessary to get a finite number of states; other finite-state model-checking approaches suffer from similar problems. For example, in the Casper approach it is necessary to limit the depth of messages that can be generated and sent by the intruder, which has the consequence that attacks (e.g., resulting from type-flaws) can be missed. In contrast, in the AVISPA project we will consider a much less restrictive infinite-state model and use abstractions and symbolic model-checking techniques to cope with this infinity (e.g. apply a symbolic representation for intruder-generated messages), and thus also scale-up to large-scale protocols and applications.

In addition to the above, a number of other model-checking systems have been employed for the analysis of small and medium scale security protocols. For example, in [88], Leduc and Germau illustrate how to employ the ISO-standardized formal language LOTOS to specify protocols and cryptographic operations, and how a model-based verification method can then be used to verify the robustness of a protocol against attacks. The LOTOS language uses CSP as one of its parts and this approach is closely related to Lowe's, and like that and other model-checking approaches, is limited to finite sets of messages.

We conclude this overview of relevant related work by discussing the recently proposed Athena approach [123], which combines model-checking and theorem-proving techniques with the strand space model (used also in [42, 70]) to reduce the search space and automatically prove the correctness of protocols with arbitrary number of concurrent runs. Athena has been applied successfully to analyze a number of small and medium scale protocols, but it is questionable whether the approach will be able to scale to large In-

ternet protocols and applications. For example, like for Casper/FDR2, this approach is not able to handle type flaws. Even more critically, it is not well-suited for modeling and reasoning about protocols requiring non-atomic keys, which is needed for e-commerce protocols (like SET) and many of new generation Internet protocols.

The new generation of security-sensitive protocols and applications

The Internet Engineering Task Force (IETF) provides a forum for working groups to coordinate the technical development of new protocols. Its most important function is the development and selection of standards within the Internet protocol suites.

The history of cryptography and cryptanalysis has repeatedly shown that open discussion and analysis of algorithms exposes weaknesses not thought of by the original authors, and thereby leads to better and more secure algorithms. Within the IETF, frank discussion of the flaws of proposed and actual protocols has led to improved versions. Hence, the IETF supports and encourages the open, prudent, discussion of vulnerabilities in hardware and software in all appropriate IETF venues, giving the manufacturer of the vulnerable product a short but useful early warning of discovered vulnerabilities so that they have an opportunity to repair them and/or to prepare patches or work-arounds.

Formal methods are starting to find their way in the IETF, but the current work in IETF standardization, including the few rigorous assessment activities, is dominated by US contributors. For instance, Catherine Meadows has applied the NRL protocol analyzer (developed by her and her group at the Center for High Assurance Computer Systems of the Naval Research Laboratory, Washington, D.C.) to carry out preliminary analyses of some IETF protocols, in particular the Internet Key Agreement protocol IKE and the Group Domain of Interpretation GDOI [30]. As a consequence of this increased awareness for the need of formal methods, the IETF has begun to consider explicit formal requirements. For example, one of the explicit requirements for Just Fast Keying JFK (one of the proposed successors of IKE) is that it is provably secure.

To meet the requirement of provable security, it is important to recognize that the type of protocols currently discussed at the IETF imposes new challenges to the formal specification and verification communities. Writing a formal specification presupposes a common formal language for doing so, and current approaches are inadequate here as they tend to be to narrowly focused on subproblems and cannot handle the richness present in real, large-scale protocols. For instance, the protocols are typically characterized as depending on assumptions of properties of related protocols, by being integrated in protocol suites with distributed responsibility on security, by offering variants and options that allow for adaption to different application contexts, and by security properties expected to be established by protocols mainly designed to offer non-security services. These requirements, in particular the last one, leads to protocols with a rich structure of messages, including both security-relevant and non-security-relevant components. The intrinsic complexity of interaction between security-specific and other parts leads to increased demands on abstraction techniques, in particular, abstraction being integrated in specification techniques and thus being formally treated. For instance, the use of bounded counters for the purposes of replay-protection or the use of known or assumed secure channels should be subject of abstraction. In contrast to related approaches discussed above, AVISPA explicitly addresses the type of protocols characterized above, by extending its scope to security-sensitive Internet protocols. That is, we do not only consider security protocols, but protocols whose goals strongly depend on security properties of parts of the protocol, like mobile-ip or SIP that contain large security related sub-protocols.

6 Community Added Value and Contribution to EU Policies

6.1 European Dimension of the Consortium

The development of an approach like ours, based on multiple advanced technologies, requires a research effort with a truly European dimension: The expertise required by the AVISPA project cannot be found in a single national research group or site. Indeed, it would be very difficult to pursue the technical and socio-economical objectives of the proposed project without the participation of different research groups, from both academia and industry. The development and exploitation of the techniques to be applied in the

project require a considerable variety and a high degree of advanced technical and technological skills. The partners of the AVISPA consortium are leading European experts both in the different automated deduction techniques upon which the project is based, and/or in the design and analysis of protocols and applications. Moreover, the partners have a common background on automated verification strengthened by a long and successful history of international collaboration and strong bilateral relations. The AVISPA consortium, therefore, collects the set of complementary technological skills required to tackle such a high-risk project like ours, to a degree only possible in such a multi-national cooperation.

6.2 European Added Value and Contributions to EU Policies, Standardization and Regulation

As discussed in detail in Section 4 ("Contribution to programme/Key Action objectives"), the AVISPA project fits well the highly innovative objectives of the FET action line "VI.1.1 – Open domain", which is focussed 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 the line 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.

The AVISPA project will have a relevant and immediate impact on social and economic development in Europe. The project will, in particular, contribute 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 will ensure their dissemination to the European (and international) research community and the dissemination of the results to industry and standardization and regulation bodies such as the IETF will contribute to improving the competitiveness of European industry.

In other words, the new technology developed in the AVISPA project will contribute to the standardization and industrial consensus of (security-sensitive) Internet protocols and applications, and thereby improve the reliability and efficiency of protocol and networks, and hence reduce their social and marketing costs. Thus, AVISPA will address European policy objectives such has those of the *e*Europe and *e*Europe+ action plans: it will promote a cheaper, faster and more secure Internet. Moreover, it will stimulate the use of the Internet for accelerating and achieving social objectives such as electronic access to public services and online public health services. We will illustrate the project's contribution to Community social objectives in the next section (Section 7), and discuss the project's impact on the economic development and S&T prospects in Section 8.

7 Contribution to Community Social Objectives

Our economic, political, and social life today depends vitally on communication and IT infrastructures, in particular the Internet, which offers tremendous possibilities for communication and information exchange both nationally and internationally. This dependence will continue its explosive development with ongoing activities aiming at supporting complex individual and administrative transactions with advanced distributed, communicating, IT systems. In order to assure the free flow of information, media pluralism, and cultural diversity, it is critical that European citizens participate more vigorously in the development and use of the Internet technologies.

Substantial advances in distributed applications and infrastructures have taken place in areas such as health-care, e-commerce and e-government, and have consequences for every individual European Community citizen. These consequences include the ubiquitous availability of communication, which reduces the impact of spatial and temporal distances between people and/or institutions and thus brings communities closer together.

Continued expansion of these technologies and their benefits relies on several economical and social factors that must be guaranteed by industry. One of the major enabling factors is trust: all participants must have confidence in the security of electronic transactions. Thus, security is a major concern with respect to communication and IT technology, influencing important parts of business and private life. For instance, the fact that personal and transaction data are distributed over networks raises serious privacy

concerns, and financial transactions relying on the infrastructure require protection against attackers and must enforce fairness when exchanging money for goods and services. Overall, the technological development and the new, widespread ways of collecting and processing personal data on information highways carries risks for fundamental personal rights and freedoms, in particular the right to privacy, when personal data is processed. Here the European Community has strong commitments regarding the protection of individuals, which in turn imposes strong requirements on new technologies.

From a technical viewpoint, the infrastructure for these new distributed applications is basically given by a set of protocols that has been defined, or is currently being defined, by standardization and regulation bodies such as IETF, ITU, IEEE, and W3C. Because of the infrastructure moving towards mobility support, distributed services and applications, application integration etc., we face the situation that more protocols are developed in even shorter time, increasing the number of protocols currently under discussion and the need for providing evidence that these protocols achieve their goals.

Security is integrated into the infrastructure either by means of specific security protocols or protocol families aimed at satisfying general security requirements (cf. protocols like IPSec, IKE, TLS, SSH, etc.) or by augmenting protocols and applications with security-sensitive parts, i.e. security specific functional elements that satisfy application-oriented security requirements (cf. Geopriv, SIP, etc.). In all cases, it is crucial that requirements are given precisely and that security mechanisms are assessed with respect to these requirements. Experience in the scientific community has shown that designing security mechanisms is particularly error-prone. Even assuming perfect cryptography, protocols are often flawed and can potentially be exploited in ways that undermine personal privacy or are financially ruinous. Example of flaws in security or Internet applications are: violations of secrecy, privacy, or anonymity of participants, failure of authentication, etc. We thus have a strong motivation to apply rigorous methods when performing the assessment of protocols and solutions.

The need for rigorous assessment has been recognized by the European Community, cf. the following guideline for Internet service providers taken from the Council Of Europe, Committee Of Ministers Recommendation No. R (99) 5 Of The Committee Of Ministers To Member States For The Protection Of Privacy On The Internet (adopted on Feb 1999 at the 660th meeting of the Ministers' Deputies): "Use appropriate procedures and available technologies, preferably those which have been certified, to protect the privacy of the people concerned (even if they are not users of the Internet), especially by ensuring data integrity and confidentiality as well as physical and logical security of the network and of the services provided over the network." We see the AVISPA project as a crucial step for enabling the high-level quality certification of protocols, including those that protect the privacy of the natural persons or institutions involved.

The AVISPA approach emphasizes the need for analysis and assessment of security sensitive protocols and applications, thus extending the scope of the approach to those infrastructure elements for which security requirements really are being defined (compared to pure security protocol analysis, where requirements are often anticipated). Providing the community with methods and tools to efficiently and rigorously assess the security of sensitive protocols and applications is a key issue to provide secure technology and to achieve trustworthiness and acceptance of the communication infrastructure, and the applications built on top of them.

Finally, but not less importantly, the project will train young researchers in the European Union on state-of-the-art techniques for reasoning about security-sensitive protocols, systems and applications.

8 Economic Development and S&T Prospects

Recent developments in distributed, communicating IT systems, including e-commerce, e-business, and e-government applications in office, home, and mobile environments, have given rise to completely new business sectors comprising both technology, application and content providers and operators. Although the recent economic slow down has shown that many of the anticipated revenues and growth rates were overestimates, business and private life have undergone substantial changes ranging from procurement, human resources, and controlling processes for business, to online banking and shopping for private persons. The success of these new developments depends on the quality of the communications infrastructure and the corresponding applications.

This dependence may be one of the reasons why growth rates have not developed as expected: though

people willingly accept potential risks and use existing IT infrastructure for conducting low-value transactions, the technology lacks acceptance when it comes to high-value or mission-critical transactions. Lack of acceptance is mainly caused by lack of trust: if vital transactions are processed by an infrastructure that is not under the complete control of the user, then it is essential that users have confidence in the correct operation of systems, their robustness against malicious attack, and the privacy of the data that they maintain. This becomes even more important in a technology sector where new requirements and technical solutions appear with increasing speed, and where systems are subject to a broad range of possible attacks: providing confidence by gaining experience with the system and collecting empirical evidence at the same time is simply impossible!

The AVISPA approach addresses the problem of trust in the quality and security of communications infrastructure. It aims at replacing empirical and informal reasoning by a rigorous, mathematical methodology providing the strongest possible grounds for trust. By providing automatic techniques for security analysis and verification of large-scale security-sensitive protocols, the AVISPA project provides a means of overcoming the acceptance problem, and thus significantly contributes to the economic development of the communications technology and e-commerce business sector.

To prove its applicability to real-world industrial-scale systems, it is important to address the security of those protocols and applications that provide the basis of today's e-commerce applications. They are given by the results of standardization bodies such as IETF, ITU, W3C, IEEE, and 3GPP. There is considerable motivation in industry for standardizing the basic constituents of the infrastructure since this avoids the provision of proprietary, incompatible solutions and it is considered to be the best approach to make long-term decisions and to achieve long product cycles.

By recognizing the role of the Internet in today's communication systems, AVISPA will concentrate on IETF protocols. This decision also takes into account the fact that activities to security assessment have been welcomed by the IETF, see for instance "Encryption and Security Requirements for IETF Standard Protocols", by Jeffrey Schiller, (IETF draft *draft-ietf-saag-whyenc-00.txt*).

Current work in IETF standardization, including the few rigorous assessment activities, is dominated by US contributors. AVISPA has the potential to strengthen the European role by providing the reference assessment technology for IETF security-sensitive protocols. This potential can be realized by providing an effective tool with wide coverage that features push-button automation, which should increase the acceptance and dissemination of the approach. AVISPA significantly contributes to the strong European position in verification of security properties, which in turn enables the European industry to provide high-quality products and solutions with rigorously assessed properties.

The contribution of AVISPA to both the security verification community and the communication systems community, particularly the IETF, will be supported by a broad spectrum of dissemination activities; for details, see the description of Workpackage 8 in Section 9. Moreover, the results obtained by AVISPA can be immediately exploited in industry. For instance, the AVISPA industrial partner, Siemens AG, particularly its mobile communications business group ICM, has a substantial interest in supporting its standardization work in IETF, 3GPP and other bodies by means of rigorously validating its proposals and thus increasing their acceptance. Previous experience based on pen-and-paper work (e.g. the UMTS authentication and key exchange protocol being standardized by 3GPP) showed that verification can be the key to success in standardization. The automation provided by the AVISPA approach will substantially reduce the effort involved, enabling formal analysis to be carried out as a routine, every-day activity, thus leading to substantial increase in quality of standardized solutions.

9 Workplan

9.1 General Description

The project can be subdivided in three main parts, corresponding to 8 workpackages (where WP1 and WP8 are administrational workpackages, devoted to project management and assessment):

1. Specification languages for Internet security protocols (WP2&3).

   We will design a high-level specification language for expressing protocol analysis problems close to the language used in text-books and by engineers. We will develop a compiler that translates analysis problems into a standard declarative format, called *intermediate format*, well-suited for automated deduction.

2. Error-detection and verification procedures (WP4&5).

   We will develop effective tools, based on automated deduction, for error-detection as well as complementary techniques for compositional verification.

3. Analysis of industrial protocols (WP6&7).

   We will select a corpus of representative protocols from IETF drafts. We shall use these protocols to tune the back-ends, assess the success of our tools, and develop a methodology for optimally using our tools.

WP1: Project Management

Project management will aim to keep the project on target in a way that the individual task objectives and the overall project objectives can be best achieved. Given the relatively small size of our consortium and the past history of successful collaboration between all partners, we expect that project management will be effective and unproblematic.

Overall, this workpackage describes how we will coordinate the cooperation between the project partners, as well as the communication between the partners and the European Commission. It also fixes responsibilities for financial and other kinds of administration involved in running the project. Finally, it sets standards for monitoring the technical content and progress of each workpackage, supervising the evolving project results at each milestone, and coordinating the synergies between the different workpackages.

Besides two annual project evaluation meetings and a final evaluation meeting, we also anticipate three project workshops per year attended by all partners in order to synchronize and assess the results, as well as additional meetings and bilateral visits, which will be arranged as needed.

WP2: Protocol Specification Languages

The objective of this workpackage is to design an expressive high-level protocol specification language close to the one used by text-books and IETF drafts, and to build a translator from this high-level language to a rewrite-based declarative intermediate format. Protocol specifications expressed in the intermediate format will be mapped to deduction problems that can be processed by automated deduction techniques. The design of the translator raises non-trivial issues related to the formal definition of semantics for protocol executions. The intermediate format simplifies the connections of automated tools to the high-level specification language by code sharing. A minimal version of the specification languages has been developed for authentication protocols without loops or branches in a recent joint work by the first three partners UNIGE, INRIA and ETHZ (see [5, 125] and Section 14). It will serve as the basis for the definition of the specification languages of the AVISPA tool, which should also support the formalization of Internet protocol-suites and take into account the algebraic properties of cryptographic operators, the security goals and the assumptions under which the protocols are analyzed.

WP2.1: Modularity & Control Structures

Complex Internet protocols are typically built from smaller sub-protocols. This sub-workpackage is dedicated to the design of syntactic facilities and control structures for composing complex protocols from smaller ones.

Many of the more complex Internet protocols allow principals to choose their actions depending on received messages or other criteria. In other words, these protocols contain choice points. For example, in the SSL protocol, the server chooses the cryptographic system he uses depending on the message he has received. This allows for some bidding-down attacks where the intruder can force the server to use weak encryption.

To model such protocols, we will introduce in the syntax of the high-level specification language both conditional branching and non-deterministic branching (for modeling "random" decisions), and then develop the corresponding rewrite rules for the intermediate format. We expect that the problem of the translation to intermediate format will be reduced to solving inequalities on messages terms.

We will also define a syntactic construction for timestamps. Timestamps are important for authentication protocols like Kerberos [87], which deliver short-time tickets for authentication. The compilation of timestamps into intermediate format should take into account the actions to be taken by the receiver of a message whose timestamps are no longer valid, and will be handled similarly to choice points.

Finally, we will also extend our protocol specification languages with control structures such as iteration. This construction will be important for the analysis of large protocol-suites like SET, where after registration the client can iterate the purchase sub-protocol any number of times.

WP2.2: Algebraic Properties of Operators

This sub-workpackage aims at refining the analysis of protocols by modeling special properties of the operators employed for building messages.

Standard protocol analysis techniques rely on the assumption that one can decrypt a cipher only if one has the corresponding key (this is a so-called *black-box* cryptography model). Hence, in the traditional approach the equality relation on message expressions is restricted to the property that decryption and encryption are inverses.

These assumptions are, however, quite restrictive, for example in the cases of concatenation and of some symmetric encryption algorithms. Algebraic properties of encryption cannot be ignored, as the case of the Diffie-Hellman group protocol demonstrates. In [107] it is proved that this protocol is secure if one ignores the properties of the *XOR*-encryption algorithm, but in [118] it was then shown that (an implementation of) the protocol suffers from a flaw relying on the properties of the *XOR*-encryption.

Equalities on terms can also be used to model arithmetic equalities, e.g. for calculating a key, as in the Diffie-Hellman protocol or the Internet Key Exchange protocol. In these protocols, the two participants compose a common key in two different ways, using two different knowledge sets. Being able to express the equality between those keys will allow us to specify and analyze such protocols.

More specifically, in this sub-workpackage, we will extend the high-level specification language with equalities between terms, which will allow us to check if the principals playing in the different roles have the knowledge necessary to compose the messages they are supposed to according to the protocol. These equalities will then be automatically translated into the intermediate format, in order to be used by the tools for modeling the intruder's behavior. We will consider two different approaches for analyzing a protocol using equalities:

- Designing specific unification algorithms for the algebraic properties (to identify terms that are equal modulo the properties). This technique is not suited for all kinds of equalities, but we will investigate it in detail in the case of associative and/or commutative operators.
- Using explicit intermediate format rules for describing the equalities.

The first approach is more efficient, whereas the second is more general.

WP2.3: Data Structures

This sub-workpackage provides syntactical extensions for dealing with complex data-structures required by the emerging Internet protocols.

Complex digital certificates are used in several protocols as authentication structures. For example, the SET protocol, based on the X.509 certificate format, adds several standard and specific extensions which make certificates more complex. We will design new syntactical constructs to declare data records and deal with such formats.

New negotiation mechanisms have been introduced in recent IETF and e-commerce protocols. For example, establishing security associations in the IKE protocol requires messages with variable cryptographic options and data structures. Moreover, in other protocols like the SET payment capture, messages contain fields of arbitrary size and number. We will use a suitable syntax (e.g., with variadic constructors) to express the different possibilities of composing a message, where the number of arguments and their types are fixed during the protocol execution. Moreover, we will use the control structures described in WP2.1 to formalize the choices that can be made during protocol execution.

Finally, we will provide information about message sizes in the specification. This will permit one to express and analyze block-ciphers, and to get much closer to the actual implementation of a protocol than with other approaches.

WP2.4: (GUI) Engineering Issues

To address engineering needs for specifying large protocols, we will provide an extensible graphic front-end for editing the high-level protocol specification files. This editor will offer a visual representation of the different protocol sections. Moreover, it will provide ergonomic features to manage complex specifications (highlighting critical data and displaying protocol specifications in a hierarchically structured way).

The graphical interface will also allow for direct access to the error-detection and verification back-ends on the high-level protocol descriptions (i.e. it will allow for push-button validation of protocol specifications in the high-level language) and will support compositional reasoning (cf. WP4). It will also provide functionality for displaying flaw scenarios by means of interactive simulations and animations. Therefore this sub-workpackage will also concern WP3 and WP4.

WP3: Context & Properties Specification

The objective of this workpackage is to extend the specification languages of our AVISPA tool with constructs for expressing different security properties, for specifying assumptions about the environment and the abilities of the intruder, as well as for specifying ways in which principals communicate with each other.

WP3.1: Security Properties

In the prototype tool developed in [5, 125], protocols are validated by investigating whether they satisfy two kinds of security properties, namely secrecy and authentication. Intuitively, secrecy of a message part exchanged by some honest principals means that the intruder is not able to see the contents of this message part, while authentication means that the claimed identity of principals is correct and that the contents of the messages they have sent has not been tampered with; for various definitions of authentication and other security properties see [95] and [79, 120].

The emerging Internet protocols aim at providing a number of other security properties, which we will consider in our project. For example, e-commerce protocols often require not only secrecy and authentication but also anonymity or non-repudiation. Intuitively, anonymity means that an observable communication between agents does not leak information about their identities (see, e.g., the definitions in [44, 119, 121, 124]), so that the specification of anonymity can be considered as a challenging extension of secrecy, while non-repudiation aims at preventing principals from denying that they have sent or received certain messages [79, 120, 133]. For example, non-repudiation protocols avoid conflicts by collecting evidence of message reception or emission so that this evidence can be presented to a "judge" in the event of a dispute. However, this evidence should not be derivable by a dishonest participant from an incomplete session.

We will investigate the formalization of anonymity and non-repudiation as reachability problems for the intermediate format rules (which formalize the protocols) by extending the encodings of secrecy and authentication or by introducing new appropriate specifications.

There are a number of important security properties that require a quantitative model, different from that of the properties considered above. For example, Internet servers are usually required to be invulnerable to denial-of-service attacks. In order to detect this kind of attacks, we will also investigate how to extend our tool by assigning time-costs to protocol steps and intruder actions.

WP3.2: Assumptions on Environment

Whether a security property is satisfied by a protocol may depend on assumptions made on honest principals, on the environment, and/or the intruder. These assumptions should be specified concisely along with the description of the protocol and they should be taken into account by the validation tools.

This sub-workpackage aims at providing facilities to declare assumptions about protocols and environments, and techniques to compile the protocol specification accordingly. To this end, we will, for instance, introduce assumptions on the way received messages are interpreted by the honest principals. Typically, principals like smart cards or terminals with low computing power will skip the analysis of some part of a message they receive in order to accelerate the protocol execution. This restriction may, however, give rise to more attacks since less verification is performed. In order to correctly model such principals, we will specify possible assumptions on the interpretation of messages received by principals, such as defining parts that will not be decrypted.

To specify assumptions about the environment and the intruder ability, we will introduce a mechanism to specify protected communication channels: in such a model, messages exchanged between two principals on a protected channel cannot be eavesdropped or manipulated by the intruder. We can then consider new kinds of attacks. For example, in mobile-IP protocols (see the list of protocols in Section 15), there are communications on two different channels designed so that the intruder is able to attack one of these channels but not both (but he can choose the one to attack).

We will also model extended abilities for the intruder by specifying, in the high-level description of a protocol, new knowledge and new deduction rules that he can apply. For example, the intruder could use another protocol as an oracle to gain more power, e.g. to decrypt some particular ciphering. Another example of extended deductive capability available to the intruder is the possibility to deduce the encryption of the prefix of a message from the encryption of the whole message, in the case where some block-cipher algorithms are used.

WP3.3: Session Instances

Session instances are obtained by assigning concrete values to the protocol variables. In particular, a variable representing a communicating agent in the protocol specification defines a role that can be played by a concrete agent or principal. In order to carry out effectively the analysis of security protocols, it is often necessary to restrict the search space by introducing assumptions about the possible protocol execution scenarios considered by the analysis tools. For instance, the prototype [5, 125] allows one to specify a set of session instances by specifying a finite set of principals and different protocol roles played by a principal in that scenario. Notice that the use of session instances does not make the search space finite (since there is an infinite number of messages that can be exchanged within a session and since a session may be restarted at any time with new values) but it nevertheless allows one to find attacks in a large number of authentication protocols, as documented in [5, 125].

In this sub-workpackage, we will investigate generalizations and abstractions of session instances in order to relax the restrictions they introduce. More specifically, we will investigate ways in which, given a set of principals, the compiler of the high-level specification language can automatically generate session instances to be considered during the analysis, rather than having the user specify such instances manually. To this end, we will define an appropriate notion of equivalence of session instances, e.g. by symmetry, which will allow us to remove "redundant" instances thereby avoiding unnecessary analysis work. Moreover, we will investigate methods in which the different back-ends of our tool can automatically explore "reasonable" protocol execution scenarios that best suit the kind of analysis they perform.

WP4: Scalability

We will improve the automated deduction techniques and tools previously developed by the partners and will make them scale up to the security protocols selected in WP6. Our goal here is both to reduce the analysis time for the protocols and to increase the number and kinds of protocols that can be analyzed using bounded computational resources. Both the generality and effectiveness of the improvements will be thoroughly evaluated against the library of protocols selected in WP6 and quantitatively assessed using the criteria set in WP7.

We plan the following activities:

WP4.1: Compositionality

We will exploit the modular structure of protocols (cf. WP2 & WP3) during analysis by using compositional reasoning techniques: properties of sub-protocols will be analyzed in isolation and these properties will then be combined to establish properties of the main protocol as a whole. Several systems of compositional inference rules have already been put forward to support "assume-guarantee" style of reasoning using different temporal logics [99, 4, 81]. We will apply and adapt these techniques to the domain of security protocol analysis. This will enable us to decompose the specification of large protocols into manageable pieces that can be automatically tackled by the available back-ends. We will develop an infrastructure for supporting compositional reasoning within the AVISPA tool. Since the full automation of compositional reasoning is infeasible, user guidance through the GUI is envisaged. We expect that the combination of compositional reasoning with the error-detection and verification capabilities provided by the available back-ends will be a key feature of the AVISPA tool as it will significantly speed up the design, modification, and analysis of complex security protocols.

WP4.2: Partial-Order Reduction

Partial-order reduction (POR) is a technique that has proved to be very successful in explicit state model-checking: different interleavings of actions can be ignored if they result in equivalent successor states. Reducing the number of traces considered this way can dramatically reduce the search.

We will systematically study the possibilities for using POR in the context of protocol verification with respect to the particular model used by the AVISPA system. POR techniques have already proved very effective in this domain. In previous work [5, 125] (see also Section 14), the partners UNIGE (CO1), INRIA (CR2) and ETHZ (CR3) showed that an optimization based on these ideas allows one to simplify the model by eliminating certain kinds of interleavings and replacing several transitions by their composition. This leads to a dramatic reduction of the number of execution traces to be considered and, consequently, to an improvement of the execution times of the model-checking procedures of several orders of magnitude.

We recall that POR techniques are based on the definition of an equivalence relation between action sequences. We will analyze different possible abstractions (that lead to larger equivalence classes) and their relation to families of verification problems. Moreover we will investigate the soundness of the equivalence relations considered, i.e. whether from two equivalent states the same attacks are reachable. This will guarantee that no attacks will be excluded by the application of the POR. We will also carefully analyze the implementation of POR techniques as it must be carefully implemented so that the additional computations performed do not outweigh the savings achieved.

WP4.3: Heuristics

Heuristics guide search during state-space exploration by giving precedence to those parts that are most likely to contain a goal state. The literature contains a variety of approaches to heuristic search, which we will study and adapt to our domain. In particular, we will design heuristics specific to the domain of security protocol analysis, e.g. heuristics based on estimations of the quantity and quality of information that the intruder can glean from the application of the available actions. Moreover, we investigate heuristics that depend on the property to be established. For example, against certain kinds of goals (secrecy and authentication) there

are a number of standard kinds of attacks (replay, man-in-the-middle, etc.) and exploring these possibilities can also be integrated into heuristics.

WP4.4: On-the-Fly Model-Checking

The preliminary implementation of the on-the-fly model-checker [19, 20, 28] uses relatively simple algorithms and data structures, e.g. unordered linear lists (within linear-time lookup) to represent multi-sets. In this work-package, we will profile this implementation over a wide range of problems, determine which optimizations will bring most benefit, and improve the corresponding algorithms and data-structures. For example, we currently compute what the intruder can analyze, using a closure operation, after each intercepted message; but as the intruder's knowledge increases monotonically, this can be accelerated using an incremental algorithm that avoids recomputing the closure from already analyzed information.

WP4.5: Theorem Proving with Constraints

In the theorem-proving with constraints approach, the detection of flaws is reduced to a constraint solving problem. In previous work [45, 46, 47], this was implemented by a direct encoding within a constrained-based theorem prover taken off the shelf. The approach is simple but can be made more efficient by applying a more deterministic strategy for the choice of the constraint solving rules to be applied and by reducing the number of these rules. We plan to develop a structure sharing approach for managing the intruder's knowledge in order to avoid redundant deductions. Finally we will isolate the unification algorithm (for message terms) from the other constraint solving rules in order to facilitate the incorporation of the algebraic properties of encryption operators.

WP4.6: SAT-based Model-Checking

The SAT-based approach to model-checking amounts to reducing the detection of attacks of bounded length to a propositional problem. This problem is then fed to an off-the-shelf, state-of-the-art SAT solver. In previous work [7], the reduction of the bounded (in)security problem to propositional logic is done in two steps: the security problem is first translated into a planning problem which is afterwards reduced to a propositional formula using standard encoding techniques. Computer experiments, described in the same work, show that the time spent to generate the SAT formula largely dominates that spent by the SAT solver. We will thus focus on the design, implementation, and experimental evaluation of new, more elaborate, encoding techniques. Improvements will be obtained by adopting more sophisticated, but still general purpose, encoding techniques (see e.g. [85, 86]) as well by the specialization of the encoding techniques to the domain of security protocol analysis. In particular, we expect that great improvements will be obtained by building encryption properties into the encoding.

WP5: Verification

The three currently implemented analysis tools are specialized for falsification, i.e. finding flaws in protocols. In this workpackage, we will also integrate mechanisms to derive positive statements about protocol security, i.e. to prove the correctness of a protocol if no flaw can be found. In general, it is undecidable whether a protocol is correct with respect to a given security property [68, 69].

WP5.1: Use of Abstraction Techniques

Verification is already possible with the techniques implemented so far whenever the number of agents and the number of sessions between the agents (and hence the state-space) is finite. However, it would be desirable to verify protocols even without these restrictions. The use of abstractions [51] represents an attractive way of attacking this problem. Abstraction techniques are extensively used in program analysis and the verification of reactive systems and hardware systems. The use of abstractions often provides a way to prove

the correctness/security of systems by over-estimating the possibility of failure. We will apply abstraction techniques to protocols verification following two lines of research.

A first approach will be to abstract systems with an infinite number of principals running the protocol by restricting the generation of nonces (terms newly created in each session). Some preliminary and promising work has already been conducted by one of the project partners, based on abstracting all nonces that were created by the same participant and intended for the communication with the same participants. In this sub-workpackage, we will formalize this abstraction, prove its completeness, and most importantly, comprehensively test and refine it on large-scale examples.

The second approach that we will investigate, initiated by [73] (but see also [80, 104]), is to over-estimate the intruder knowledge by using regular tree languages. This method allows one to show that some states are unreachable, and hence that the intruder will never be able to know certain terms. Regular tree-languages can be used here to effectively model the knowledge that the intruder might have acquired from previous sessions.

WP5.2: Infinite-state Symbolic Model-Checking

The verification of security protocols where the number of parallel sessions and principals is not fixed a priori can also be tackled by the so-called paradigm of infinite-state symbolic model-checking. In this setting powerful assertional languages (e.g. regular languages, first or second order logic) are used to symbolically represent or approximate infinite collections of protocol configurations (e.g. with an unbounded number of principals). Symbolic forward or backward reachability procedures are then used as semi-algorithms to prove safety properties. Abstractions play here an important role in order to enforce the termination of the resulting procedures. Specifically, we will investigate the applicability of the following techniques to our case studies:

- automatic generation of finite abstractions via the combination of weak second-order monadic logic, theorem proving, and model-checking as in [31];
- forward reachability in which sets of states are approximated by regular languages combined with accelerations and widening operators as in [1, 37];
- symbolic backward reachability for special classes of safety properties (properties that can be represented or approximated via upward closed sets of states) as in the general framework proposed in [2]. Some preliminary experiments are described [38].

These approaches are compatible with the general approach taken in the AVISPA project in that they try to avoid human ingenuity during proof construction. Thus, they represent an important step towards push-button verification technology for infinite-state systems.

WP5.3: Completeness of Model-Checking Procedures

One of the drawbacks of the use of abstractions (as in WP5.1 and WP5.2) is that they might return false negative answers. An alternative approach, which does not suffer from this problem, is to reduce the (infinite) model associated with a protocol to a bounded model [111, 97]. This amounts to proving that only a bounded number of sessions and agents needs to be considered in order to find an attack if there is any; in other words, showing that if the restricted model has some property, then so does the original.

The aim of this workpackage is to implement the check for the applicability of such bounds, i.e. whether the considered protocol is inside the class of protocols for which the bounds can safely be applied. If so, the analysis is automatically restricted to the required bounds. The bound checks are performed dynamically, i.e. in the back-ends during the search.

WP6: Selection and Specification of Protocols

The objective of this work package is to analyze and formally specify a large collection of protocols and security properties that have recently been standardized or are currently undergoing standardization, especially

at the IETF. The goal of this workpackage is to select and formalize at least 40 different protocols of this kind, with an average of at least 2 security properties per protocol.

WP6.1: Selection

Not all protocols are equally important for the proper operation of the Internet or for the secure support of e-commerce or telecommunications (Voice over IP, VoIP) applications; nor all protocols are suitable for the kind of formal verification that we plan to perform, either because their strength or weakness depends on the particulars of the cryptographic algorithms used, or on policy-based operation, or on their performance. Some protocols are just "containers" to pass further unspecified authentication information. Some protocols have so many different layers, exceptions, configurations, and message exchanges that a reasonable abstraction and formalization is out of the scope of our proposal.

Within these constraints, after an analysis of the protocols, the purpose of this task is to select the candidates for formal specification. They are given by both the protocol description and a set of security properties the protocol should satisfy. In general, a protocol is designed to achieve a multitude of security goals, e.g. secrecy of session key k, or authentication of peer in role A with strong agreement on nonce n. We therefore introduce the notion of *security problem*, which is given by a protocol paired with a security property. We expect an average of 2–3 problems defined per protocol.

The introduction of problems emphasizes the point that a protocol can only be considered to offer appropriate security if all of its associated security properties are satisfied. In order to assure that the problems identified during the selection process show this kind of completeness, and taking into account that in many cases required security properties are not explicitly stated in IETF documents, work in WP6.1 includes both thorough analysis of the protocols and the initiation of a discussion process with IETF representatives, with the selection being based on their comments on coverage and relevance to problem proposals by AVISPA.

In order to show the broad applicability of the AVISPA tool, the selection is directed by the requirement for covering a broad range of protocol groups. In section 15, a list of 26 groups is given that, together with a few additional groups suggested for further work, cover most of the ongoing IETF activities on security and security-sensitive applications. Work in WP6.1 will lead to the definition of a library of security problems, called "AVISPA Library" that will be proposed to the scientific community as a suite of benchmark problems for automatic protocol analysis.

The analysis during the compilation of the AVISPA library also gives insight into new requirements for our languages for protocol specification (WP2) and for the specification of context and properties (WP3).

WP6.2: Specification

In specifying the IETF protocols and their associated problems, we will often need to perform a certain amount of abstraction and simplification. Some of this *may* be necessary to deal with the limitations of our tools (say, on the number of concurrent sessions, or agents, or on some data types), and some because certain features of the protocol (such as the cipher suites used, techniques for negotiating them, policy issues, or strength against some denial of service attacks) will be outside the scope of our analysis. Besides this, most of the IETF protocols are not written in a language that readily is translatable to a formal specification. We expect that in many cases there will be a need to discuss the protocols with their developers in order to properly interpret the intended meaning of the (proposed) standards. These discussions are conducted in continuation of the process initiated in WP6.1.

WP7: Tool Assessment

The technical results of the project will be evaluated with respect to measurable criteria that allow us to establish whether the project has achieved the technical objectives set in Section 2. Moreover, the assessment activity will be used to keep the project on target. To this end, the assessment activity will be conducted in parallel to the other activities and this will allow us to continuously monitor their progress and take corrective actions at an early stage. We thus expect significant interplay between WP7 and work in WP4, WP5, and WP6.

The technical achievements of the project will be assessed by thoroughly testing the tool developed by the partners against the AVISPA library, a library of security problems drawn from the protocols developed by the IETF (cf. WP6). The following criteria will be used for the assessment of the tool:

Coverage: number and variety of security problems specified in the high-level specification language and successfully translated in the intermediate format.

Effectiveness: number of security problems successfully falsified or verified by the tool.

Performance: CPU time spent by the tool to process the successfully analyzed security problems on standard commercially available computers.

Assessment points are placed at months 12, 24, and 30 to check and quantitatively measure the progress. The project will be considered on track at months 12, 24, and 30 if the tool meets the target requirements indicated in Table 1. In particular, a coverage requirement of "P problems from G groups" means that the tool must be able to successfully analyze P security problems drawn from G of the 26 groups selected in WP6; an effectiveness requirement of "E problems" means that the tool should either falsify or verify at least E of the security problems specified in the high-level specification language; finally the performance requirement is set to 1 hour per problem in all the assessment points.

	Month 12	Month 24	Month 30
Coverage	20 problems from 5 groups	40 problems from 10 groups	80 problems from 20 groups
Effectiveness	15 problems	30 problems	60 problems
Performance	< 1 hour	< 1 hour	< 1 hour

Table 1: Target requirements of assessment points

The project will thus be considered successful if the tool meets the following criteria at the end of the project:

Coverage: at least 80 security problems taken from 20 of the 26 groups given in Sections 15 should be specifiable in the high-level specification language.

Effectiveness: the tool should either falsify or verify at least 75% (i.e. 60) of the problems specified in the high-level specification language. Moreover the set of successfully analyzed problems should come from one protocol of each of the Main Protocols listed in Section 15.

Performance: the processing of the successfully analyzed security problems should take less than 1 hour of CPU time per problems on standard commercially available computers.

Notice that each of the above criteria, even when considered in isolation, is far beyond the scope of state-of-the-art tools for the formal analysis of security protocols. Therefore the successful completion of the project implies a substantial advancement of the state-of-the-art in this domain.

The analysis of the results of the assessment activity will allow for the identification of the classes of protocols, threat models, and security goals for which each back-end performs best. For example, we expect that for highly combinatorial problems (e.g. with many similar instances) the SAT-approach will have the best performance, whereas the approach based on theorem-proving with constraints is likely to be very good in handling specific properties of encryption operators (like associativity of XOR); finally for extending the capacity of intruders, and in general for incorporating search heuristics to find particular kinds of flaws, the on-the-fly model-checking approach is likely to be the most suited. As a result of this analysis we will produce a classification of error-detection and verification techniques with respect to the coverage, effectiveness, and performance criteria.

WP8: Dissemination

In order to coordinate the dissemination of the project results through appropriate channels and in appropriate forums, we will organize the project workshops and the preparation of presentations and reports, and

coordinate the dissemination of scientific publications. This workpackage will also maintain the AVISPA Website, which will contain all publicly available results, and will make available an on-line demo of the AVISPA tool.

Moreover, in combination with the project meetings, we will also organize annual Project Workshops to disseminate results among the project partners and the European Commission. We will also invite researchers, scientists and users from academia, from standardization bodies (e.g. the IETF), and from industry to attend the workshops in the second project half, and in particular at the final workshop. Researchers from the different project partners are attending, or will attend, meetings of the IETF, and we also plan to present the results of our project at these meetings.

We will organize the efforts of all partners in preparing the presentation of, and reporting on, the main project results, which will illustrate the methods and tools developed, and highlight their applications at the European level. In order to reach a wide academic and industrial audience, all project partners will aim to publish project results in international journals, conferences, and symposia.

9.2 Workpackage list

The workplan is organized in 8 workpackages. WP1 is an administrative workpackage devoted to project management and is aimed at keeping the project on target. WP2 and WP3 address the evolution of the protocol description languages and their reflection in the implemented back-ends. WP4 will improve the automated deduction techniques and tools previously developed by the partners and will make them scale up to the security problems arising in state-of-the-art Internet protocols. WP5 is devoted to the investigation and integration of mechanisms to derive positive statements about protocol security. In WP6 we set up a library of security problems drawn from a large collection of protocols and security properties that have recently been standardized or are currently undergoing standardization. In WP7 we assess the technical results of the project. Finally, in WP8 we coordinate the dissemination of the project results.

WORKPACKAGE LIST						
No	**Title**	**Lead Contractor**	**Person-months**	**Start month**	**End month**	**Deliverable No**
WP1	Project Management	UNIGE (CO1)	18 (15+1+1+1)	1	30	D1.1-1.4
WP2	Protocol Specification Languages	INRIA (CR2)	48 (11+22+13+2)	1	24	D2.1-2.4
WP3	Context & Properties Specification	ETHZ (CR3)	49 (12+15+15+7)	7	24	D3.1-3.3
WP4	Scalability	ETHZ (CR3)	40 (17+6+17+0)	1	28	D4.1-4.6
WP5	Verification	INRIA (CR2)	28 (9+8+8+3)	7	27	D5.1-5.3
WP6	Selection, Analysis, & Specification of Protocols	SIEMENS (CR4)	28 (4+2+4+18)	1	24	D6.1-6.2
WP7	Tool Assessment	UNIGE (CO1)	8 (2+2+2+2)	10	30	D7.1-7.4
WP8	Dissemination	UNIGE (CO1)	15 (4+4+4+3)	1	30	D8.1-8.7

Legend: in the workpackage tables below, we follow the *Guide for Proposers* and indicate the nature of the deliverable by writing "R" for "Report", "O" for "Other" (including a prototype implementation deliverable and a website deliverable), and "PU" for the dissemination level "public".

9.3 Workpackage Descriptions

WP1 – Project Management

Start date:	month 1			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	15	1	1	1
WP leader:	×			

Objectives
Project management, coordination, and administration.

Description of work
This workpackage describes general administrative responsibilities and tasks for coordinating the project, including the project meetings, and the management of reports, deliverables and milestones.

T1.1: Project coordination. The project coordination will be carried out by the partner UNIGE (CO1), who will locally appoint a *Scientific Coordinator* and a *Financial and Administrative Coordinator*.

The Scientific Coordinator will chair the *Project Coordination Committee (PCC)*, which will address high-level management and financial issues, and consists of the *Local Scientific Coordinators* (i.e. the responsible scientists of the project partners) and the Financial and Administrative Coordinator.

The Scientific Coordinator will coordinate the communication between the project partners and the European Commission, and coordinate tasks and optimize synergistic interaction between the project partners. To this end, the Scientific Coordinator will manage both a common distributed repository of the project code and documentation (using the concurrent version-management tool CVS) and the AVISPA Website, which will also be used for the dissemination of the project results (see WP8).

The software management and development will be carried out following the standard software engineering practice, i.e. following a software development process like the waterfall or the V model [122], guiding the development from semi-formal specification (of the code and, in particular, of the interfaces) to the integration of the different, independently developed, software components, and to their testing and validation.

The coordination task will also include guidelines for deliverables, presentation standards, deadlines, information flow, dissemination, and reporting. This will allow the Scientific Coordinator to *(i)* consolidate the project planning, *(ii)* manage the input of the project partners on the different WPs, *(iii)* supervise the evolving project results at each milestone, *(iv)* supervise the assessment and evaluation of the results, and *(v)* assemble and control the project reports and deliverables. The Scientific Coordinator will be supported in this task by the PCC and by the different *Workpackage Leaders*, who will be responsible for the detailed coordination, planning, monitoring, realization and reporting of the respective workpackages and the detailed coordination of tasks between the different workpackages.

T1.2: Project meetings. Besides two annual project evaluation meetings and a final evaluation meeting, we also anticipate several project workshops per year attended by all partners in order to synchronize and assess the results, as well as additional meetings and bilateral visits, which will be arranged as needed. The following is a preliminary plan for these meetings:

Month 1: Kick-off meeting.
Month 6: Assessment of the preliminary results and plan for future work (including the Dissemination and Use Plan – Deliverable D8.3).
Month 9: Synchronization and assessment meeting.
Month 12: First evaluation meeting.
Month 16: Synchronization and assessment meeting.
Month 20: Synchronization and assessment meeting.
Month 24: Second evaluation meeting.
Month 27: Synchronization and assessment meeting.
Month 30: Final evaluation meeting.

T1.3: Project administration. The Financial and Administrative Coordinator of UNIGE (CO1) will coordinate, with the support of the PCC, the financial and bureaucratic administration of the project, managing in particular the cost statements, the budgetary overviews, the budget for the management task, etc.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D1.1	Year 1 Progress Report	12	R	PU	UNIGE (CO1)
D1.2	Year 2 Progress Report	24	R	PU	UNIGE (CO1)
D1.3	Year 3 Progress Report	30	R	PU	UNIGE (CO1)
D1.4	Final Project Report	30	R	PU	UNIGE (CO1)

The Final Project Report includes also a final management report.

Milestones and expected results

Month 6: Dissemination and Use plan, Consortium Agreements.
Month 12: First Evaluation Meeting.
Month 24: Second Evaluation Meeting.
Month 30: Third Evaluation Meeting & Final Project Report.

Interaction with other WPs

This WP interacts with all other WPs.

WP2 – Protocol Specification Languages

Start date:	month 1			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	11	22	13	2
WP leader:		×		

Objectives
To define a high-level protocol specification language capable of supporting the specification of security-sensitive parts of state-of-the-art Internet protocols. To design and develop a translator from the high-level language to a rewrite-based declarative intermediate format amenable to formal analysis.

Description of work
A minimal version of the specification language has been developed for authentication protocols without loops or branches in recent joint work by the partners CO1, CR2, and CR3. In this workpackage we will incorporate new constructs needed to express the features of state-of-the-art Internet protocols. Three sub-workpackages are devoted to this: WP2.1 (Control Structures), WP2.2 (Algebraic Properties), and WP2.3 (Data Structures); a fourth sub-workpackage, WP2.4 (Interface), is dedicated to the development of a graphical user interface that will provide the user with an integrated environment for the editing, analysis, and simulation of protocols. The activities in the sub-workpackages are broken down as follows:

WP2.1 Control Structures.

T2.1.1 Modules. Add the definition of sub-protocols and their invocation.

T2.1.2 Conditionals. Add conditional constructs and non-determinism.

T2.1.2 Iteration. Add constructs for iteration.

T2.1.3 Timestamps. Support the specification of timestamps.

WP2.2 Algebraic Properties.

T2.2.1 Unification algorithms. Design new unification algorithms for specific algebraic properties of operators (e.g. associativity and/or commutativity).

T2.2.2 Translation into rules. Design suitable translations of the algebraic properties into rules of the intermediate format.

WP2.3 Data Structures.

T2.3.1 Records. Support the definition of messages with (extensible) records.

T2.3.2 Message size. Support the specification of message size in messages.

WP2.4 Interface. Design and develop an extensible graphic facility for protocol specification and interpreting results. The work will be organized along the following main lines of activity:

T2.4.1 Editor. Design and develop an editor for high-level protocol specifications. This will allow protocol designers to use appropriate graphical components and diagrams to specify protocol sections.

T2.4.2 Interface to the back-ends. Design a bi-directional interface with the back-ends. This will allow one to invoke and present in a uniform way the activity and results of the back-ends.

T2.4.3 Simulation. Design and develop a simulator capable of displaying flaw scenarios by means of interactive simulation and animation.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D2.1	The High-Level Protocol Specification Language	8	R&O	PU	INRIA (CR2)
D2.2	Algebraic properties	18	R&O	PU	INRIA (CR2)
D2.3	The Intermediate Format Specification Language	8	R&O	PU	INRIA (CR2)
D2.4	Interface	24	O	PU	INRIA (CR2)

Milestones and expected results

Month 5: Definition of new control constructs and new data structures, as well as their translation into the intermediate format.

Month 12: Design and implementation of the translator of the algebraic properties into rules of the intermediate format.

Month 18: Unification algorithms modulo algebraic properties.

Month 24: Graphical User Interface (GUI).

Interaction with other WPs

Interaction with WP3, WP4 and WP5 is foreseen as the translation of sophisticated control constructs and data structures into the intermediate format will affect the specification of the problems input to the back-ends. The adequacy of the new constructs will be tested during the selection and specification of Internet protocols carried out in WP6. Finally, the graphical user interface will support the activities in all the following workpackages, in particular the specification of protocols carried out in WP6.

WP3 – Context & Properties Specification

Start date:	month 6			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	12	15	15	7
WP leader:			×	

Objectives
To build components for expressing security goals and assumptions about the environment into both the high-level and the intermediate specification languages.

Description of work
In recent joint work, the partners UNIGE (CO1), INRIA (CR2), and ETHZ (CR3) designed high-level and intermediate specification languages. These were only capable of specifying authentication and secrecy properties and a fixed set of assumptions about the environment and they also required information about session instances to be given. In this workpackage we will lift these limitation by incorporating components capable of expressing a wide-range of security properties, threat models, and complex communication scenarios among the agents. Three sub-workpackages are devoted to this:

WP3.1 Security Properties. Model security properties as reachability problems for the intermediate format rules. Since this has already been done in previous work by the partners for secrecy and authentication properties, here we will focus on anonymity and non-repudiation properties. We will also study invulnerability with respect to denial-of-service attacks by adding quantitative elements to our model.

T3.1.1 Anonymity. Model anonymity as a reachability problem for the intermediate format rules.

T3.1.2 Non-repudiation. Model non-repudiation as a reachability problem for the intermediate format rules.

T3.1.3 Invulnerability with respect to denial-of-service attacks. Extend the model based on the idea of assigning time-costs to protocol steps and intruder actions.

WP3.2 Assumptions on Environment. Support the concise specification of assumptions made on honest principals, on the environment, and the intruder.

T3.2.1 Message interpretation by honest principals. Specify possible assumptions on the interpretation of messages received by principals, such as defining parts that will not be decrypted by the receiver.

T3.2.2 Protected communication channels. Define mechanisms for defining protected communication channels.

T3.2.3 Extended abilities of the intruder. Strengthen the intruder model with new knowledge and new deduction rules.

WP3.3 Sessions Instances. Restrict the search space by specifying possible protocol execution scenarios.

T3.3.1 Automatic generation of session instances. Design and develop a technique for automatically generating session instances.

T3.3.2 Symmetry Reduction. Define equivalence relations over session instances, e.g. by exploiting symmetry, so to discharge "redundant" instances.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D3.1	Security Properties	24	R&O	PU	ETHZ (CR3)
D3.2	Assumptions on environment	18	R&O	PU	ETHZ (CR3)
D3.3	Sessions instances	12	R&O	PU	ETHZ (CR3)

Milestones and expected results

Month 12:	Design and development of a procedure for the automatic generation of session instances that exploits symmetry reduction.
Month 18:	Mechanisms to define message interpretation and protected communication channels.
Month 24:	Formal definitions of anonymity, non-repudiation, and invulnerability with respect to denial-of-service attacks.

Interaction with other WPs

WP3 is strongly connected to WP2, as the encoding of the assumptions on message interpretation and channels depends from the translation of protocols into the intermediate format that is obtained in WP2. Interaction with WP4 and WP5 is foreseen as the translation of new security properties, sets of session instances, and assumptions on the environment into the intermediate format will affect the specification of the problems input to the back-ends. The adequacy of the new concepts will be tested during the selection and specification of Internet protocols carried out in WP6.

WP4 – Scalability

Start date:	month 1			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	17	6	17	0
WP leader:			×	

Objectives
To improve the automated deduction techniques and tools previously developed by the partners and scale them up to the security protocols selected in WP6.

Description of work

In order to cope with great complexity of security-sensitive Internet protocols, the back-ends previously developed by the partners will be strengthened with new, state-of-the-art techniques and optimizations. The work is broken down as follows:

WP4.1: Compositionality. Develop within the AVISPA tool an infrastructure capable of exploiting the modular structure of complex protocols by supporting compositional reasoning.

T4.1.1 Compositional reasoning. Design and develop an infrastructure for compositional reasoning.

T4.1.2 Integration. Integrate the infrastructure for compositional reasoning with the back-ends and the GUI.

WP4.2: Partial-order reduction. Use POR techniques to prune search.

T4.2.1 Equivalences between action sequences. Design and analyze different abstractions leading to larger equivalence classes. Prove the soundness of the equivalence relations considered.

T4.2.2 Implementation and experimental evaluation of POR. Design and develop data structures and algorithms for an effective application of POR techniques. Experimentally assess the gains obtained by the adoption of POR techniques.

WP4.3: Heuristics. Design heuristics for the domain of security protocol analysis.

T4.3.1 Definition of heuristics. Define heuristics based on estimations of the quantity and quality of information that the intruder can obtain from the application of the available actions.

T4.3.2 Implementation and experimental evaluation of heuristics. Implement the heuristics and carry out experimental evaluations to assess the gains obtained by their adoption.

WP4.4: Improvements to the on-the-fly model-checker. Improve the algorithms and data-structures used by the on-the-fly model-checker.

T4.4.1 Profiling. Profile the implementation of the back-end previously developed by the partner over a wide range of complex problems and determine which optimizations bring the greatest benefits.

T4.4.2 Implementation and experimental evaluation. Design and implement the optizations. Carry out experiments to tune them and assess the resulting improvements.

WP4.5: Improvements to the theorem prover with constraints. Improve and tune the strategies and data structures of the theorem-prover.

T4.5.1 Structure sharing. Develop a structure sharing approach for managing the intruder's knowledge in order to avoid redundant deductions.

T4.5.2 Separation of the unification algorithm. Separate the unification algorithm (for message terms) from the other constraint solving rules in order to facilitate the incorporation of the algebraic properties of encryption operators.

WP4.6: Improvements to the SAT-based model-checker. Design, implement, and experimentally evaluate new techniques for the reduction of bounded (in)security problems to propositional logic.

T4.6.1 General encodings. Investigate and integrate into the back-end the state-of-the-art domain-independent encoding techniques available in the literature (e.g. graphplan-based encodings [85, 86]).

T4.6.2 Domain specific encodings. Design and develop new encodings based on domain-specific features such as, e.g., the properties of cryptographic operators.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D4.1	Compositionality	28	R&O	PU	ETHZ (CR3)
D4.2	Partial-order reduction	8	R&O	PU	ETHZ (CR3)
D4.3	Heuristics	11	R&O	PU	ETHZ (CR3)
D4.4	AVISPA tool v. 1	11	R&O	PU	ETHZ (CR3)
D4.5	AVISPA tool v. 2	19	R&O	PU	ETHZ (CR3)
D4.6	AVISPA tool v. 3	27	R&O	PU	ETHZ (CR3)

Note that in this case the "O"(ther) in the deliverable "Nature" stands for a prototype implementation of the techniques in the AVISPA tool.

Milestones and expected results

Month 12: Delivery of the AVISPA tool v. 1 strengthened with POR techniques and heuristics.

Month 20: Delivery of the AVISPA tool v. 2 featuring optimized back-ends and first implementation of compositional reasoning.

Month 28: Delivery of the AVISPA tool v. 3 featuring fully optimized back-ends and completed support to compositional reasoning.

Interaction with other WPs

There will be considerable feedback between WP4 and WP7. Moreover, WP4 strongly depends on WP2 and WP3 as the new features added to the intermediate specification language will greatly affect the complexity of the problems input to the back-ends.

WP5 – Verification

Start date:	month 6			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	9	8	8	3
WP leader:		×		

Objectives
To investigate and integrate mechanisms to derive positive statements about protocol security.

Description of work
To complement the error-detection tools described in WP4, this workpackage introduces techniques for proving automatically the correctness of a protocol with respect to a security property. The activities in the sub-workpackages are:

WP5.1 Abstraction Techniques. Design abstractions for handling an unbounded number of sessions.

T5.1.1 Abstraction of Nonces. Investigate the search of flaws with unbounded number of sessions by reducing the set of nonces generated by a participant to a finite one.

T5.1.2 Intruder knowledge. Approximate intruder knowledge by a regular tree language.

WP5.2 Infinite-state Symbolic Model Checking. Symbolically represent or approximate infinite collections of protocol configurations.

T5.2.1 Automatic generation of finite abstractions. Investigate techniques based on weak second-order monadic logic, theorem proving, and model checking for automated generation of abstractions.

T5.2.2 Approximation of forward reachability sets. Approximate sets of protocol configurations by regular languages combined with accelerations and widening operators.

T5.2.3 Approximation of backward reachability sets. Develop proof techniques for properties that can be represented using upward closed sets of states.

WP5.3 Completeness of Model Checking. Identify security problems where the abstraction does not introduce false negatives.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D5.1	Abstractions	12	R&O	PU	INRIA (CR2)
D5.2	Infinite-state model checking	20	R&O	PU	INRIA (CR2)
D5.3	Completeness issues	27	R	PU	INRIA (CR2)

Milestones and expected results

Month 10: Design of abstraction techniques for nonces and intruder knowledge and implementation in the back-ends.

Month 18: Design of infinite-state model checking procedures for verification and implementation in the back-ends.

Month 27: Identification of problems where the absence of flaws in a bounded model implies the correctness of the protocol.

Interaction with other WPs
Since the abstraction techniques have to be implemented in the back-ends, WP5 is strongly connected to WP4. There should also be interaction between the treatment of the security properties considered in WP3.1 and the abstraction techniques considered in WP5.2. Finally a tight interaction is expected between WP5 and the assessment workpackage WP7.

WP6 – Selection & Specification of Protocols

Start date:	month 1			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	4	2	4	18
WP leader:				×

Objectives
To define the AVISPA library, a set of formalized security problems (protocols and security properties) drawn from Internet protocols that have recently been standardized or are currently undergoing standardization.

Description of work
Since not all protocols are equally important for the proper operation of the Internet or for the secure support of e-commerce or telecommunications applications, the first task will be the selection of the most representative protocols (and associated security properties) out of the large body of IETF protocols (cf. Section 15). The selected security problems will be then formalized in the high-level specification language of the AVISPA tool.

T6.1: Problem Selection. Select candidate protocols for formal specification and analyze them with respect to their required security properties. Analyze the resulting problem set for coverage and relevance, with comments of IETF representatives being taken into account.

T6.2: Problem Specification. Formally specify the set of problems defined in T6.1 in the AVISPA high-level specification language.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D6.1	List of selected problems	11	R & O	PU	SIEMENS (CR4)
D6.2	Specification of the problems in the high-level specification language	24	R & O	PU	SIEMENS (CR4)

Milestones and expected results
The project review meetings at months 12 and 24 will be the milestones of this workpackage.

Interaction with other WPs
Interaction with all other WPs is expected but a particularly tight correlation with WP7 (Assessment) is envisaged.

WP7 – Tool Assessment

Start date:	month 9			
Participant number:	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	2	2	2	2
WP leader:	×			

Objectives
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.

Description of work
Test the AVISPA tool developed by the partners against the AVISPA library of security problems set in WP6 and quantitatively assess the results with respect to the coverage, effectiveness, and performance criteria. Since the tool will be systematically assessed three times during the project's life-time, we will develop an experimental set up that will automate the execution and reporting activities.

T7.1 Experimental setup. Design and develop scripts for the automatic execution of tests and result reporting.

T7.2 Assessment. Perform the tests of the AVISPA tool on the number of security problems required in Table 1, for the given time point.

T7.3 Comparative analysis. Identify the classes of protocols, threat models, and security goals for which each back-end performs best and produce a classification with respect to the coverage, effectiveness, and performance criteria.

Deliverables

No.	**Name**	**Delivery date**	**Nature**	**Dissemination level**	**Lead**
D7.1	Experimental Setup	11	O	PU	UNIGE (CO1)
D7.2	Assessment of the AVISPA tool v. 1	12	R&O	PU	UNIGE (CO1)
D7.3	Assessment of the AVISPA tool v. 2	24	R&O	PU	UNIGE (CO1)
D7.4	Assessment of the AVISPA tool v. 3	30	R&O	PU	UNIGE (CO1)

Milestones and expected results

Month 12: Decision on consequences of assessment results of D7.2.
Month 24: Decision on consequences of assessment results of D7.3.
Month 30: Final decision on project success based on assessment results of D7.4.

Interaction with other WPs
This workpackage will be an important source of feedback for WP4 (Scalability) and WP5 (Verification).

WP8 – Dissemination

Start date:	month 1			
Participant name (number):	UNIGE (CO1)	INRIA (CR2)	ETHZ (CR3)	SIEMENS (CR4)
Person-months:	4	4	4	3
WP leader:	×			

Objectives
To disseminate the project results through appropriate channels and in appropriate forums.

Description of work
This workpackage coordinates the dissemination of the results by *(i)* maintaining the AVISPA website, *(ii)* organizing the project workshops, *(iii)* organizing the writing of the Project Presentation, of the Dissemination and Use Plan and of the Technology Implementation Plan, and *(iv)* coordinating the dissemination of scientific publications. This will be achieved by carrying out the following tasks.

T1.1: AVISPA Website. The project website will be maintained by the Scientific Coordinator of the partner UNIGE (CO1). The site will contain information about the project workshops and meetings, and, most importantly, it will contain all publicly available results, and will make available an on-line demo of the tools of AVISPA.

T1.2: Project Workshops. The Scientific Coordinator, in accordance with the Project Coordination Committee, will organize annual Project Workshops (combined with the project meetings) to disseminate results among the project partners and the European Commission. We will also invite researchers, scientists and users from academia, from the IETF, and from industry to attend to the workshops in the second project half, and in particular at the final workshop. Researchers from the different project partners are, and will be regularly attending the meetings of the IETF, and we plan to present the results of our projects at a number of such meetings.

T1.3: Project Presentation and TIP. The Scientific Coordinator will organize the efforts of all partners in writing the Project Presentation and the Technological Implementation Plan (TIP), as well as the different project deliverables and reports (cf. WP1). Based on the Final Project Report (Deliverable 1.4), the TIP will survey the main project results, illustrate the methods and tools developed, and highlight their applications at the European level. We anticipate that the Final Project Report, as well as many of the deliverables, will give rise to publications in international journals, conferences, and symposia.

T1.4: Scientific Publications. In order to reach a wide academic and industrial audience, all project partners will aim at publishing the project results in international journal, conferences, and symposia. This will be done either individually or in collaboration with other partners, as appropriate.

Deliverables

No.	Name	Delivery date	Nature	Dissemination level	Lead
D8.1	AVISPA Website	1—30	O	PU	UNIGE (CO1)
D8.2	Project Presentation	3	R	PU	UNIGE (CO1)
D8.3	Dissemination and Use Plan	6	R	PU	UNIGE (CO1)
D8.4	Year 1 Project Workshop Report	10	R	PU	UNIGE (CO1)
D8.5	Year 2 Project Workshop Report	19	R	PU	UNIGE (CO1)
D8.6	Year 3 Project Workshop Report	28	R	PU	UNIGE (CO1)
D8.7	Technology Implementation Plan	30	R	PU	UNIGE (CO1)

Note that in this case the "O"(ther) in the field "Nature" stands for a website. Moreover, we write "1—30" to indicate that the website will be continuously updated during the course of the project.

Milestones and expected results

AVISPA Website.

Project Workshops.

Technology Implementation Plan.

Scientific Publications.

Interaction with other WPs

This WP interacts with all other WPs.

9.4 Deliverables List

Legend: Following the *Guide for Proposers*, we indicate the nature of the deliverable by writing "R" for "Report" and "O" for "Other" (such as a website), and "PU" for the dissemination level "public". Moreover, we write "1—30" as the delivery date of deliverable D8.1 to indicate that the AVISPA Website will be continuously updated during the course of the project.

Note also that deliverables are ordered by the workpackage numbers rather than by the delivery dates.

DELIVERABLES LIST						
Deliv. No	**Deliverable title**	**WP No**	**Lead participant**	**Nature**	**Dissemination level**	**Delivery date (proj. month)**
1.1	Year 1 Progress Report	1	UNIGE (CO1)	R	PU	12
1.2	Year 2 Progress Report	1	UNIGE (CO1)	R	PU	24
1.3	Year 3 Progress Report	1	UNIGE (CO1)	R	PU	30
1.4	Final Project Report	1	UNIGE (CO1)	R	PU	30
2.1	The High-Level Protocol Specification Language	2	INRIA (CR2)	R&O	PU	8
2.2	Algebraic properties	2	INRIA (CR2)	R&O	PU	18
2.3	The Intermediate Format Specification Language	2	INRIA (CR2)	R&O	PU	8
2.4	Interface	2	INRIA (CR2)	O	PU	24
3.1	Security Properties	3	ETHZ (CR3)	R&O	PU	24
3.2	Assumptions on environment	3	ETHZ (CR3)	R&O	PU	18
3.3	Sessions instances	3	ETHZ (CR3)	R&O	PU	12
4.1	Compositionality	4	ETHZ (CR3)	R&O	PU	28
4.2	Partial-Order Reduction	4	ETHZ (CR3)	R&O	PU	8
4.3	Heuristics	4	ETHZ (CR3)	R&O	PU	11
4.4	AVISPA tool v.1	4	ETHZ (CR3)	R&O	PU	11
4.5	AVISPA tool v.2	4	ETHZ (CR3)	R&O	PU	19
4.6	AVISPA tool v.3	4	ETHZ (CR3)	R&O	PU	27
5.1	Abstractions	5	INRIA (CR2)	R&O	PU	12
5.2	Infinite-state model checking	5	INRIA (CR2)	R&O	PU	20
5.3	Completeness issue	5	INRIA (CR2)	R	PU	27

DELIVERABLES LIST (CONT.)						
Deliv. No	**Deliverable title**	**WP No**	**Lead participant**	**Nature**	**Dissemination level**	**Delivery date (proj. month)**
6.1	List of Selected Problems	6	SIEMENS (CR4)	R&O	PU	11
6.2	Specification of the Problems in the high-level specification language	6	SIEMENS (CR4)	R&O	PU	24
7.1	Experimental Setup	7	UNIGE (CO1)	O	PU	11
7.2	Assessment of the AVISPA tool v.1	7	UNIGE (CO1)	R&O	PU	12
7.3	Assessment of the AVISPA tool v.2	7	UNIGE (CO1)	R&O	PU	24
7.4	Assessment of the AVISPA tool v.3	7	UNIGE (CO1)	R&O	PU	30
8.1	AVISPA Website	8	UNIGE (CO1)	O	PU	1—30
8.2	Project Presentation	8	UNIGE (CO1)	R	PU	3
8.3	Dissemination and Use Plan	8	UNIGE (CO1)	R	PU	6
8.4	Year 1 Project Workshop	8	UNIGE (CO1)	R	PU	10
8.5	Year 2 Project Workshop	8	UNIGE (CO1)	R	PU	19
8.6	Year 3 Project Workshop	8	UNIGE (CO1)	R	PU	28
8.7	Technology Implementation Plan	8	UNIGE (CO1)	R	PU	30

9.5 Project Planning and Timetable

A GANTT chart depicting the scheduling of the workpackages is given in Figure 9.5. Note that the assessment in WP7 will take place in three phases, indicated by the solid line, whereas the wavy pattern indicates "monitoring" phases preparing the next round of assessment.

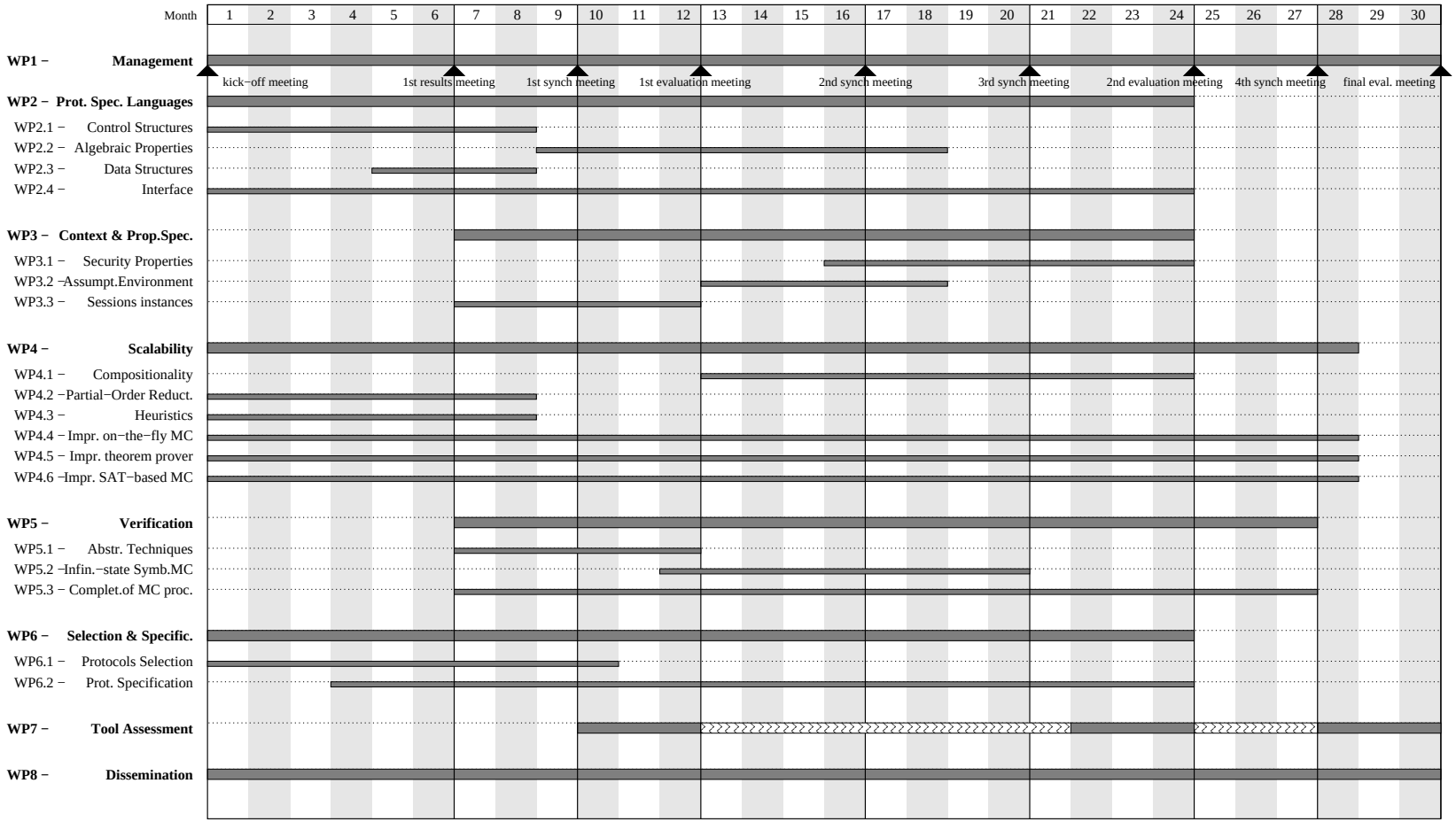

Figure 1: GANTT Chart of the AVISPA Project

9.6 Graphical Presentation of Project Components

A PERT chart representing the logical dependencies between the workpackages is given in Figure 2.

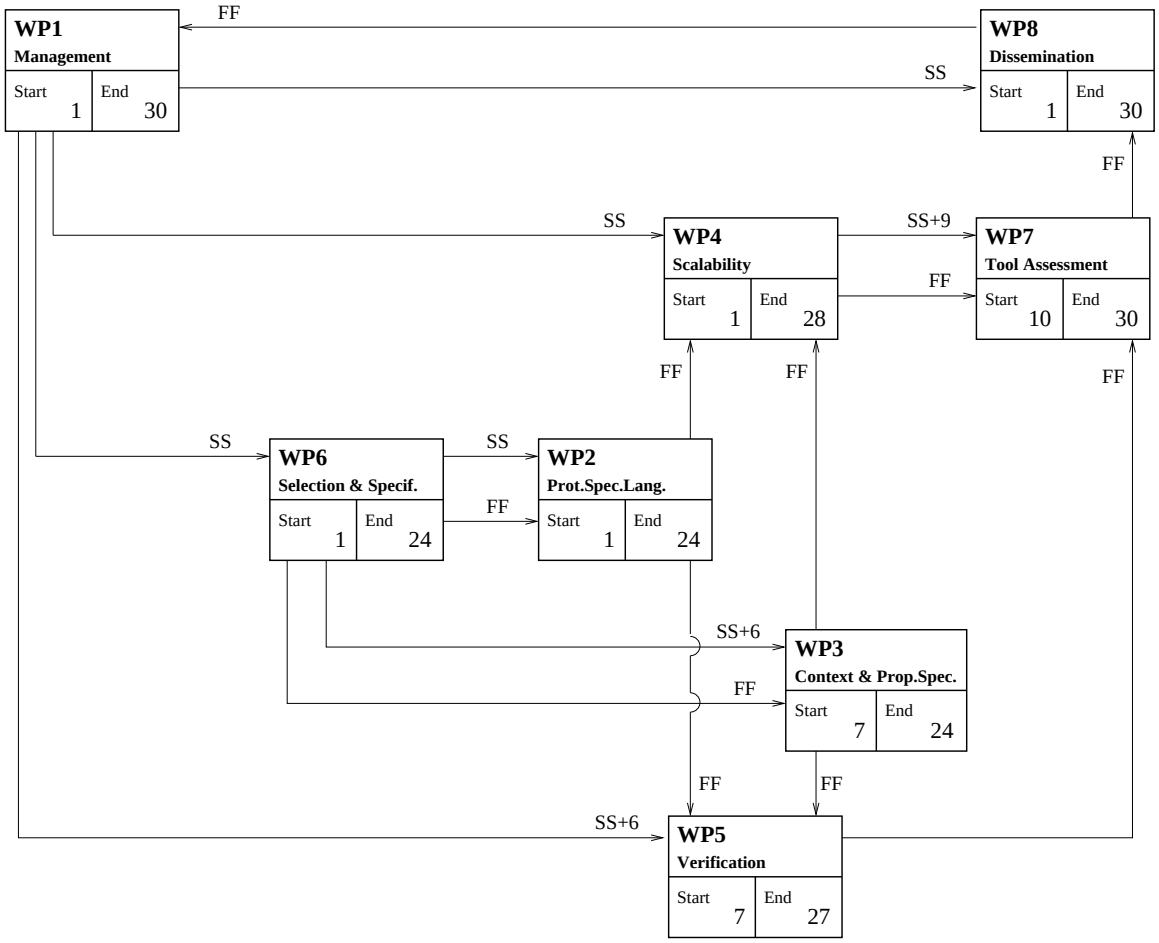

Figure 2: PERT Chart of the AVISPA Project

9.7 Project Management

As described in Section 9, we have devised a special workpackage WP1 that will be devoted to the management of the project. This workpackage describes general administrative responsibilities and tasks for coordinating this project, including the project meetings, as well as the management of deliverables and milestones. We here describe in more detail how the management will proceed.

Project Coordination

The project management and coordination will be carried out by UNIGE (CO1), which will act as the central node in the project, chairing and moderating the project meetings, assembling and controlling the deliverables, and supervising the evolving project results at each milestone as well as the assessment results.

The *Scientific Coordinator (SC)*, Dr. Alessandro Armando has considerable research management experience. In particular, he led the UNIGE (CO1) partner in the FET Open Assessment Project IST-2000-26410, "AVISS: Automated Verification of Infinite State Systems", the predecessor to this proposal, which was carried out by the partners UNIGE (CO1), INRIA (CR2), and ETHZ (CR3); more information on AVISS is

given in Section 14 below. Dr. Alessandro Armando will be assisted in this task by members of his group, including Dott. Luca Compagna who also collaborated to the AVISS project.

The SC will chair the *Project Coordination Committee (PCC)*, which will address high-level management and financial issues, and consists of the *Site Leaders* (i.e. the scientific/administrative coordinator of the different project partners). The PCC will also be responsible for the formulation of appropriate *Consortium Agreements* to place on a legal basis the relationship between the project partners and their responsibilities for the duration of the work. The Consortium Agreements will be delivered at month 6, together with a *Dissemination and Use Plan*. The PCC will also be responsible for the writing of the *Mid-Project Assessment Report* and of the *Final Management Report* (which will be included in the *Final Project Report*, deliverable D8.3), and for the formulation of the *Technology and Implementation Plan*, which will also be part of the dissemination work of workpackage WP8.

With the support of the PCC, the SC will also devise a *project management plan*. In particular, the SC will

- coordinate the communication flow between the project partners and the European Commission, and
- coordinate tasks and optimize synergistic interaction between the project partners.

To this end, the SC will manage both

- a common distributed repository of the project code and documentation (using the version-management tool CVS), and
- the AVISPA Website, which will allow also for the dissemination of the project results (see WP8 in Section 9).

The project management plan will guide the coordination task, which will also include guidelines for deliverables, presentation standards, deadlines, information flow, dissemination and reporting, as well as quality assurance measures. This will allow the SC to

- consolidate the project planning,
- manage the input of the project partners on the different WPs,
- supervise the evolving project results at each milestone,
- supervise the assessment and evaluation of the results,
- control changes in the workplan and taking executive decisions (e.g. stopping certain activities or re-distributing resources),
- resolve possible conflicts, and
- assemble and control the project reports and deliverables.

The SC will be supported in this task by the PCC and by the different *Workpackage Leaders*, who will be responsible for

- the detailed coordination, planning, monitoring, realization and reporting of the respective workpackages,
- the detailed coordination of tasks between the different workpackages.

In the unlikely case of a conflict or dispute, the four members of the PCC will vote on the issue, with the SC's vote counting twice in the case of a tie.

Project management will aim to keep the project on target in a way that the individual task objectives and the overall project objectives can be best achieved. Given the relatively small size of our consortium and the past history of successful collaboration between all partners, we expect that project management will be effective and unproblematic. In addition to the use of a common on-line repository, e-mail and telephone will be means for effective communication within the consortium.

Project Meetings

Besides the two annual project evaluation meetings and the final evaluation meeting, we anticipate at least three project workshops per year attended by all partners in order to synchronize and assess the results, as well as additional meetings and bilateral visits, which will be arranged as required. The following is a preliminary plan for such meetings:

- Month 1: Kick-off meeting.
- Month 6: Assessment of the preliminary results and plan for future work (including the Dissemination and Use Plan).
- Month 9: Synchronization and assessment meeting.
- Month 12: First evaluation meeting.
- Month 16: Synchronization and assessment meeting.
- Month 20: Synchronization and assessment meeting.
- Month 24: Second evaluation meeting.
- Month 27: Synchronization and assessment meeting.
- Month 30: Final evaluation meeting.

Project Administration

The *Financial and Administrative Coordinator (FAC)* will also be Dr. Alessandro Armando. He will be supported by the administrative staff of DIST (Department of Information, Computers and Systems Science at the University of Genova) which has extensive experience with the financial and administrative coordination of European Community projects. Each project partner will be responsible for carrying out its planned contribution within its budget, to contribute to deliverables, to provide full documentation of project activities, to provide documentation on the financial situation of the project to the FAC for reports to EC officers and authorities (such as cost statements). The FAC will coordinate, with the support of the PCC, the financial and bureaucratic administration of the project, managing in particular the cost statements, the budgetary overviews, the budget for the management task, etc.

10 Clustering

Not applicable.

11 Other contractual conditions

The Contract Preparation Forms include (under "travel/subsistance costs" and "other significant project costs") provisions for audit certificates, as well as provisions for participation of project members to relevant international (also outside the EU MS/AS) conferences and meetings, such as those of the IETF and other standardization bodies. We envisage that at least one representative of each group will attend at least one conference as well as one meeting outside the EU MS/AS during the lifetime of the project.

12 (Optional) Supplementary reports and concertation activity: Other action-specific conditions

Not applicable.

13 (Optional) Other considerations

Not applicable.

14 The FET-Open Assessment Project AVISS (IST-2000-26410)

14.1 AVISS and AVISPA

The AVISPA project is the follow-up of the FET-Open Assessment Project IST-2000-26410, "AVISS: Automated Verification of Infinite State Systems". The 12 month assessment phase was completed successfully in April 2002 and the Commission Services invited us to submit a full version of our proposal. Note that we chose a new name and acronym following the suggestion of the final review document, which praised the excellent results of the AVISS project but recommended the choice of a different name, giving "a better idea on the focus of project on the proposed subject of validation of cryptographic (security) protocols".

In this final section of the proposal, we briefly summarize the main achievements of the AVISS project. Detailed information about the project is available at the AVISS website:

URL: `http://www.informatik.uni-freiburg.de/~softech/research/projects/aviss`

which also includes the final project report [125], the publications documenting the project results [5, 7, 28, 45, 46, 47, 114, 116, 117], and an online demonstration of the *prototype analysis tool* that we developed.

14.2 Motivations of the AVISS Project

As we previously observed, the current generation of formal methods tools for the analysis of security protocols is insufficient in terms of coverage, effectiveness, and performance, to scale from simple example protocols to realistic ones. While exploratory work has shown the potential of automated deduction techniques for protocol analysis, most of them have never been tested comprehensively across a large range of problems. A FET OPEN assessment project provided thus an important preliminary step before starting a wide-scale industrial involvement within a full project.

14.3 The Main Achievements of the AVISS Project

The AVISS project lays the foundations of a new generation of analysis tools for automatic error detection for e-commerce and related security protocols.

To assess the potential of this new technology, the project was structured into two phases:

1. A *development phase* aimed at the design and implementation of a prototype analysis tool incorporating inference engines based on three promising automated deduction techniques: on-the-fly model-checking based on lazy data-types, theorem-proving with constraints, and model-checking based on propositional satisfiability checking.

2. An *analysis phase* aimed at measuring the success of the assessment project by thoroughly testing and evaluating the prototype tool (and the techniques) by applying it to the protocols in the Clark/Jacob library [48], which contains 51 protocol verification problems.

14.4 The AVISS Consortium

The AVISS project was carried out by the first three partners of the AVISPA project, namely UNIGE, INRIA, and the group lead by Prof. Basin, which at the time of the assessment project was at the Institute for Software Engineering of the Albert-Ludwigs-Universität Freiburg (ALUFR) but will move to the ETH Zurich in January 2003. (Dr. Luca Viganò and Sebastian Mödersheim, the two staff members of the ALUFR group who collaborated to the AVISS project, will also move to ETHZ and participate to the AVISPA project.) These three groups have a long and successful history of international collaboration and strong bilateral relations. Moreover, each partner is a leading experts on one of the three techniques upon which this project is based.

The success of the project has naturally paved the way to turning the prototype into a mature technology, and the transfer of this technology into the industrial sector, in a follow-up project with industry. The AVISS consortium was thus joined by the SIEMENS group, which routinely designs protocols and actively participates in their standardization at IETF, ITU, 3GPP, and W3C. Moreover, the Siemens group also has extensive experience in utilizing formal methods for the validation of protocols and applications.

14.5 The Prototype AVISS Tool

The prototype AVISS tool for security protocol analysis has the architecture shown in Figure 3. The *High-Level Protocol Specification Language* HLPSL is a language close to that used in text-books and by engineers. Given a specification of a protocol analysis problem in the HLPSL (i.e. a description of a protocol together with a security property to check), the HLPSL2IF translator translates this specification into the more detailed, tool-independent format suitable for automated deduction, called the *Intermediate Format* (IF). Specifications in the IF are then translated into tool-specific encodings that are fed into the inference engines that implement the selected automated deduction techniques. The three back-ends of the tool developed in the context of the AVISS project are:

- The on-the-fly model-checker OFMC developed by the ALUFR/ETHZ group.
- The theorem-prover based on constraint logic CL developed by the Nancy group INRIA.
- The model-checker based on propositional satisfiability checking SATMC developed by the Genova group UNIGE.

Whenever the input protocol is flawed and when the analysis carried out by the tool completes successfully, the tool will return as a counter-example an execution trace witnessing an attack on the protocol.

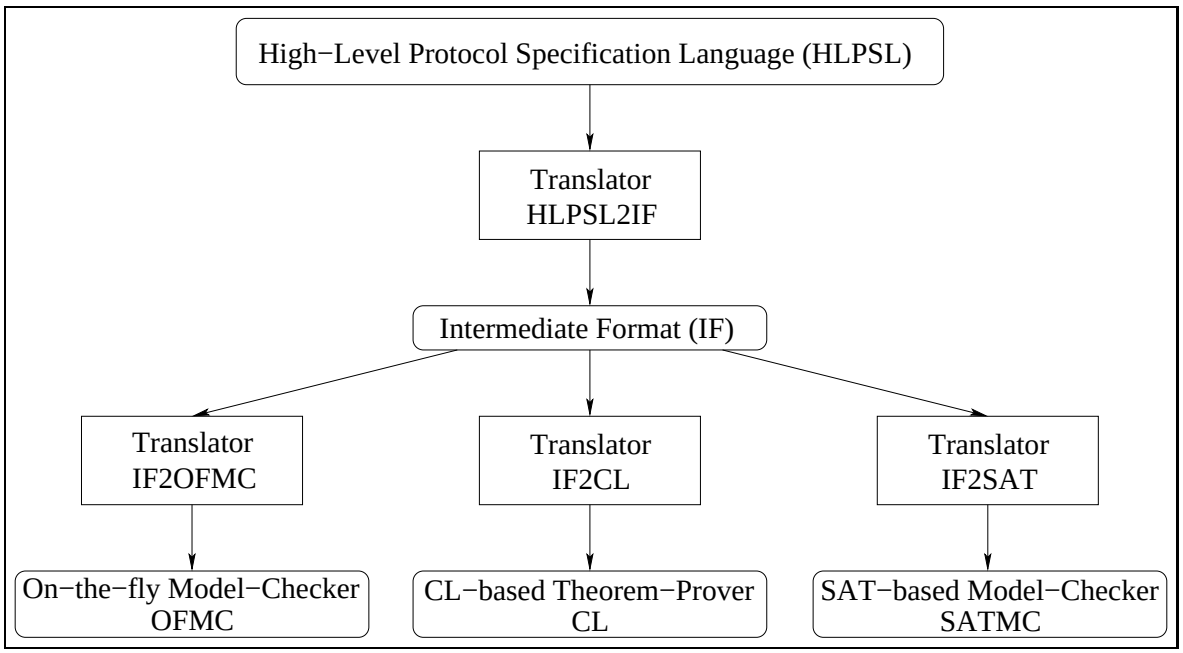

Figure 3: Architecture of the prototype AVISS tool

Ease of tool integration was an important design consideration for the AVISS tool. We currently have implemented three back-ends for performing complementary automated protocol analysis techniques, and the IF provides a specification language that allows for the integration of other existing tools as further back-ends into the AVISS architecture. We have begun to investigate in this direction and the first results are very promising; further investigations will be carried out in the context of the AVISPA project. The following are brief descriptions of the three back-ends we implemented:

The On-the-fly model-checker (OFMC) The transition relation specified by the IF is unrolled starting from the initial state producing an infinite tree that is model-checked on-the-fly. We use Haskell, a compiled lazy functional programming language, to modularly specify the search space, reduction methods, heuristics, and procedures, generalizing the method of [19]. When an attack is found, it is reported to the user by means of the sequence of exchanged messages.

Success criteria		Objectives	Results
Coverage	(number of protocols specified)	80%	90%
Effectiveness	(number of flaws detected)	70%	91 %
Performance	(CPU time)	$<$ 1 hour on 70%	$<$ 1 min on 91%

Table 2: Summary of the AVISS project results (according to the success criteria)

The Constraint-Logic-based theorem-prover (CL) The IF is translated into a first-order theory which is input to the daTac prover [84]. The CL back-end combines rewrite-based first-order theorem proving with constraint logic in order to handle properties such as associativity/commutativity of operators for representing sets of messages. Message exchanges and intruder activities are directly translated from the IF rewrite rules into clauses; searching for a flaw then amounts to searching for an incoherence in the resulting formula.

The SAT-based Model-Checker (SATMC) The SAT-based model-checker builds a propositional formula encoding a bounded unrolling of the transition relation specified by the IF, the initial state, and the set of states representing a violation of the security properties. The propositional formula is then fed to a state-of-the-art SAT solver (currently Chaff, SIM, and SATO are supported) and any model found by the solver is translated back into an attack, which is reported to the user.

14.6 Objectives and Results of the AVISS Project

The project results demonstrate the success of the assessment phase: the prototype tool more than satisfies the project objectives, according to the success criteria adopted as a measure of the assessment, i.e.

Coverage: the number of protocols, security properties, and types of security threats that can be specified in the tool's input specification language.

Effectiveness: the number of insecure protocols detected as flawed and the number of different types of attacks on the insecure protocols detected.

Performance: the time spent by the tool to detect errors in the protocols.

The AVISS tool achieves the following results, which we display in a summarized form in Table 2.

Coverage: 46 of 51 of the protocols in the Clark/Jacob library [48] are specifiable in the tool's input specification language HLPSL, which amounts to a coverage of 90% (80% was requested).[1]

Effectiveness: the tool detects attacks in 91% of the protocols in the library known to be insecure (70% was requested). More specifically, 35 out of the 51 protocols are flawed, and we can find a flaw in 32 of them, including a previously unknown flaw (the remaining 3 protocols can not be specified in HLPSL yet).

Performance: the tool can find attacks for 91% of the flawed protocols in the Clark/Jacob library (i.e. the 32 flawed protocols that can be modeled), in less than one minute of CPU time on a 1.4 GHz processor (the criterion required that at least 70% of the (flawed) protocols were processed in less than 1 hour of CPU time).

Experimental results on the performance of the back-ends of the AVISS tool over the testsuite of flawed protocols are given in Table 3. Note that the lists includes two variants of the Needham-Schroeder public key protocol that are not taken into account in [48]: without key-server (*Needham-Schroeder Public Key*) and with Lowe's fix (*Needham-Schroeder with Lowe's fix*). Moreover for the Neuman-Stubblebine protocol

[1] For some protocols, several variants are presented in [48]; according to the case study in [67], we count them as different protocols.

(*Neuman Stubblebine (complete)*) we consider as single protocols the initial part (*Neuman Stubblebine initial part*) and the repeated part (*Neuman Stubblebine repeated part*) since the repeated part contains a flaw that results from a type flaw in the initial part; these additional case studies are marked with a "*" in Table 3.

Preliminary to the execution of the back-ends we generated both an untyped and a typed version of the IF specifications by means of the HLPSL2IF translator. The OFMC back-end and the CL back-end are run against the untyped and the typed IF specifications of each protocol. The kind of the attack found (if any) and the time spent by the back-end are given in the corresponding columns.[2] Note that the analysis of the untyped and typed IF specifications may lead to the detection of different kinds of attacks. When this is the case, in Table 3 we report the two attacks found. Since, as is discussed in more detail in [5, 125], the SATMC is not suited to analyze the untyped IF specifications, we applied it to the typed IF specifications only. As a consequence, we did not apply SATMC on protocols suffering from type flaw attacks.

For SATMC we give a pair of values T_e/T_s, where T_e is the *encoding time*, i.e. the time spent to generate the propositional formula, and T_s is the *search time*, i.e. the time spent by the SAT solver to check the formula (the SAT timings are obtained using the Chaff solver [105].) The labels TO and MO indicate a failure to analyze the protocol due to time-out and memory-out, respectively.[3] The label NS indicates that the back-end does not support some of the features occurring in the corresponding IF specification (this happens to the SATMC on the *Hwang and Chen's modified SPLICE* protocol and the *Denning Sacco Key Distribution with Public Key* protocol since key-tables are not yet supported by this back-end). Finally the label NA indicates the problem has not been attempted.

The experimental results clearly indicate that the effectiveness and the performance criteria are met. Table 3 allows us also to analyze the performance of the individual back-ends:

OFMC: The OFMC model-checker performs uniformly well on all the protocols: most of the attacks are found in a fraction of a second, and detecting all the attacks requires a total time of less than one minute. It is immediate to see that the OFMC tool achieves alone both the effectiveness and the performance criteria set in the proposal for the whole tool.

CL: The poorer timings of the CL theorem-prover are balanced by the fact that it is based on an off-the-shelf prover (daTac) and it offers other advantages such as the simple integration of algebraic relations on message constructors (e.g. commutativity of encryptions in RSA). In any case, it must be noticed that also this back-end achieves alone both the effectiveness and the performance criteria set in the proposal for the whole tool.

SATMC: The experiments show that the time spent to generate the SAT formula largely dominates the time spent to check the satisfiability of the SAT instance. Nevertheless, in many cases the overall timing is not too far from that of the OFMC back-end and it is better than that of the CL back-end. It is also interesting to observe that in many cases the time spent by the SAT solver is smaller than the time spent by the OFMC back-end for the same protocol.

14.7 The Graphical Web-Based Interface

We developed a graphical web-based user interface for the AVISS tool, which is also available from the project website. This eases interaction with the tool, e.g., selecting back-ends, options, and the like, and interpreting the results. It also ensures a permanent access to the latest version of the tool, and it eliminates the need for installation and tool configuration; this makes it straightforward for others to use the system and provide feedback.

Figure 4 shows a snapshot of the interface, which, as displayed on the right, imports the tree structure of the project architecture. In particular, there are four main icons corresponding to the HLPSL2IF translator and the three implemented back-ends (OFMC, CL and SATMC). As shown on the left, input protocols are

[2] Times are obtained by running our back-ends on a PC with a 1.4 GHz Pentium III Processor and 512 Mb of RAM.

[3] We set a time limit of 1 hour for each attempt. Due to a limitation of SICStus Prolog the SAT-based model-checker is bound to use 128Mb during the encoding generation.

Table 3: Performance of the back-ends of the AVISS tool over the testsuite

Protocol	**Attack**	**OFMC**	**CL**	**SATMC**
ISO symmetric key 1-pass unilateral authentication	Replay	0.01	1.98	0.18/0.00
ISO symmetric key 2-pass mutual authentication	Replay	0.01	3.86	0.43/0.01
Andrew Secure RPC Protocol	Type flaw	0.03	4.26	NA
	Replay	0.05	32.74	80.57/2.65
ISO CCF 1-pass unilateral authentication	Replay	0.00	2.23	0.17/0.00
ISO CCF 2-pass mutual authentication	Replay	0.01	4.55	0.46/0.01
Needham-Schroeder Conventional Key	Replay STS	0.31	63.43	29.25/0.39
Denning-Sacco (symmetric)	Type flaw	0.02	15.98	NA
Otway-Rees	Type flaw	0.02	10.71	NA
Yahalom	Type flaw	0.02	44.08	NA
Woo-Lam Π_1	Type flaw	0.01	0.81	NA
Woo-Lam Π_2	Type flaw	0.01	0.80	NA
Woo-Lam Π_3	Type flaw	0.00	0.82	NA
Woo-Lam Π	Parallel-session	0.24	1074.95	3.31/0.04
Woo-Lam Mutual Authentication	Parallel-session	0.27	245.56	1024.08/7.95
Needham-Schroeder Signature protocol	Man-in-middle	0.13	53.88	3.77/0.05
**Neuman Stubblebine initial part*	Type flaw	0.03	6.19	NA
**Neuman Stubblebine repeated part*	Replay STS	0.03	3.54	15.17/0.21
Neuman Stubblebine (complete)	Type flaw	0.04	46.78	NA
Kehne Langendorfer Schoenwalder (repeated part)	Parallel-session	0.20	199.43	MO/-
Kao Chow Repeated Authentication, 1	Replay STS	0.45	76.82	16.34/0.17
Kao Chow Repeated Authentication, 2	Replay STS	0.48	45.25	339.70/2.11
Kao Chow Repeated Authentication, 3	Replay STS	0.49	50.09	1288/MO
ISO public key 1-pass unilateral authentication	Replay	0.02	4.23	0.32/0.00
ISO public key 2-pass mutual authentication	Replay	0.01	11.06	1.18/0.01
**Needham-Schroeder Public Key*	Man-in-middle	0.04	12.91	1.77/0.05
Needham-Schroeder Public Key with key server	Man-in-middle	1.12	TO	4.29/0.04
**Needham-Schroeder with Lowe's fix*	Type flaw	0.01	31.12	NA
SPLICE/AS Authentication Protocol	Replay	4.02	352.42	5.48/0.05
Hwang and Chen's modified SPLICE	Man-in-middle	0.02	13.10	NS
Denning Sacco Key Distribution with Public Key	Man-in-middle	0.52	936.90	NS
Shamir Rivest Adelman Three Pass Protocol	Type flaw	0.03	0.70	NA
Encrypted Key Exchange	Parallel-session	0.10	240.77	75.39/1.78
Davis Swick Private Key Certificates, protocol 1	Type flaw	0.05	106.15	NA
	Replay	1.19	TO	1.37/0.02
Davis Swick Private Key Certificates, protocol 2	Type flaw	0.16	348.49	NA
	Replay	0.86	TO	2.68/0.03
Davis Swick Private Key Certificates, protocol 3	Replay	0.03	2.68	1.50/0.02
Davis Swick Private Key Certificates, protocol 4	Replay	0.04	35.97	8.18/0.13

Legend: TO: Time Out MO: Memory Out
NA: Not Attempted NS: Not Supported
Replay STS: Replay attack based on a Short-Term Secret

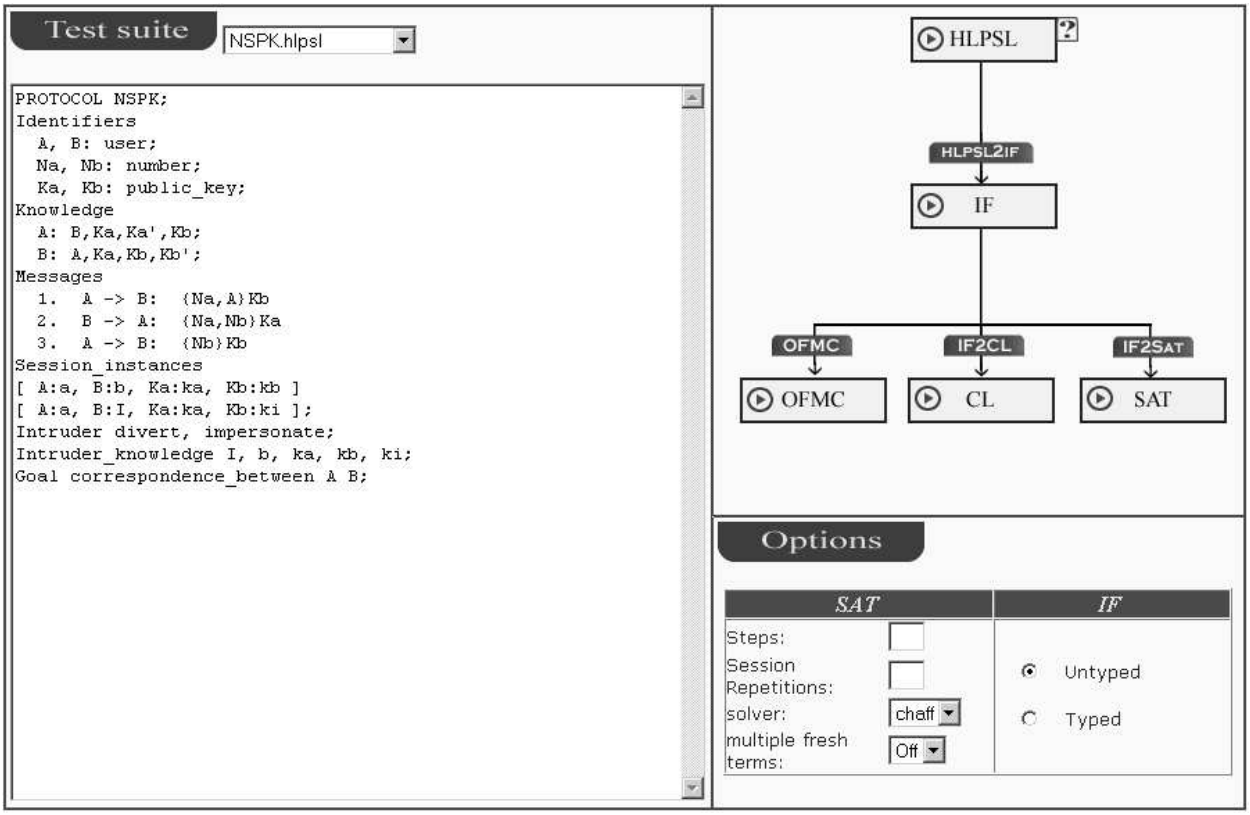

Figure 4: Snapshot of the graphical web-based interface of the AVISS tool

either selected from a given testsuite, including a number of protocols in the library [48], or they are defined by the user.

Overall, the interface provides a prototype for the tool integration platform that we will further develop in the AVISPA project.

15 A List of Protocols

Here we list the 30 groups of protocols that we initially plan to investigate. A small number of them may be obsolete in a few months, and new ones may appear.

IETF group descriptions are found in:

`http://search.ietf.org/html.charters/`

and the drafts can be found in

`http://search.ietf.org/internet-drafts/` .

For instance the mobile-ip working group home page is:

`http://search.ietf.org/html.charters/mobileip-charter.html`

and the draft *draft-ietf-mobileip-ipv6-17.txt* is found at:

`http://search.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-17.txt` .

Notice that Internet-Drafts are only work-in-progress documents: from the point of view of the IETF, they have a maximum lifetime of six months; after that time, they must be updated, or they will be deleted of the IETF repository. When an Internet-Draft becomes an RFC, it will be replaced in the RFC repository maintained by the IETF Secretariat.

15.1 Main Protocols: Mobility, VoIP, QoS, Location Services, Presence

mobile-ip

This protocol provides routing support to permit mobile IP nodes to seamlessly "roam" among IP subnetworks. The Mobile IP method supports the maintenance of active TCP connections and UDP port bindings. The Working Group is searching for solutions to address known security deficiencies and shortcomings. In March 2001, the IESG returned the Mobile IPv6 draft to the working group due to concerns about the security of binding updates sent to correspondent nodes and the associated IPsec processing that is specified in the draft. Since that time, discussions have continued to attempt to define what is really needed to make binding updates secure while taking into consideration the aspect of scalability as well as the fact that IPsec may not be the most suitable security mechanism for securing BUs between Mobile Nodes and Correspondent Nodes.

In the longer term, the WG needs to address Location Privacy.

Working group(s) homepage(s): mobileip-charter.html

Protocol description:

- *draft-ietf-mobileip-ipv6-17.txt*
- *draft-ietf-mobileip-mipv6-scrty-reqts-02.txt*
- *draft-glass-mobileip-security-issues-01.txt*
- *draft-le-aaa-diameter-mobileipv6-01.txt*
- *draft-ietf-mobileip-rfc3012bis-02.txt* extensions for the Mobile IP Agent Advertisements and the Registration Request that allow a foreign agent to use a challenge/response mechanism to authenticate the mobile node.
- *draft-le-mobileip-dh-01.txt* Dynamic Diffie Hellman based Key Distribution for Mobile IPv6

Context Transfer, Handoff Candidate Discovery, and Dormant Mode Host Alerting (seamoby)

In IP access networks that support host mobility, the routing paths between the host and the network may change frequently and rapidly. In some cases, the host may establish certain routing-related services on subnets that are left behind when the host moves. Examples of such services are AAA, header compression, and QoS. In order for the host to obtain those services on the new subnet, the host must explicitly re-establish the service by performing the necessary signalling flows from scratch. The success of time sensitive services like VoIP telephony, video, etc., in a mobile environment depends heavily on the ability to minimize the impact of the traffic redirection during a change of packet forwarding path. An efficient solution is to transfer information on the existing state associated with these services, or context, to the new subnet, a process called 'context transfer'.

Working group(s) homepage(s): seamoby-charter.html

Protocol description:

- *draft-ietf-seamoby-context-transfer-problem-stat-04.txt*
- *draft-ietf-seamoby-ct-reqs-03.txt*
- *draft-forsberg-seamoby-aaa-relocate-00.txt*
- *draft-koodli-seamoby-ctv6-03.txt*
- *draft-gopal-seamoby-ipsec-relocate-00.txt* Handovers also imply that a terminal such as a MN may perform network access authentication in order to obtain connectivity and access to network features such as QoS, header compression, etc. Security context associated with several Security Associations (SA) may need to be transferred in order to achieve this. This enables the MN to regain authenticated connectivity and establish new SAs without having to reinitiate time-consuming operations.
- *draft-gopal-seamoby-ipsecctxt-issues-01.txt*
- *draft-forsberg-seamoby-aaa-relocate-00.txt* Network access authentication and authorization is used for providing protected access to users and for billing and accounting purposes. AAA has been proposed as an infrastructure that could provide such support. With AAA, an access router can be configured with packet filtering rules to allow controlled access.

Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is an application-layer control (signalling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution and multimedia conferences.

SIP itself offers a list of security mechanisms at different layers of the IP protocol stack, including simple password-based challenge-response exchanges as well as the TLS protocol that is commonly used to authenticate HTTP-based Web-servers in the Internet. The spectrum of security means for SIP is wide, including quite different aspects like authentication, integrity protection, confidentiality and privacy as well as authorization and policy management aspects. In addition, security is always applied to certain scenarios. This may be quite specific as e.g. the IP multimedia subsystem (IMS), which is based on SIP session control, or the scenario may be quite generic, for instance end-to-middle or end-to-end SIP signalling.

Spam mail is already a problem in current Internet. Anonymity mechanisms on the Internet make it possible for a sender to send offending mail without being traced back. On the other hand, in conventional telephony this is not the case because most telephone numbers are linkable to the real identity of the caller. Besides that, the use of black lists of numbers are a remedy for the called party, since it is expensive for the sender to obtain every time new telephone numbers. But things will change when internet telephony arrives. A real menace to the service will appear: imagine receiving untraceable junk voice mails on a phone every ten minutes. On the other hand privacy (anonymity) will be a requirement too. SIP will require light-weight authentication procedures for end-to-end purposes where an authorization indication is required, but the

"real" identity is not necessary. Thus the protocols tend to hide the true identities of the users (perhaps with or without the possibility of an "identity escrow" by an enforcement entity), but the end user has some security regarding the authorization status (or the accountability, or the consistency or another attribute) of the communication partner.

Working group(s) homepage(s): sip-charter.html
sipping-charter.html

Protocol description:

- *draft-arkko-sip-sec-agree-01.txt*
- *draft-uusitalo-sipping-algorithm-agreement-00.txt*
- *draft-peterson-sip-identity-00.txt*
- *draft-loughney-sip-aaa-req-00.txt*
- *draft-watson-sipping-nai-reqs-00.txt*
- *draft-beck-sipping-billing-scen-00.txt* A caller causes costs on the called party's side. With SIP's authentication methods and few additional headers, a SIP billing service could be established that would enable callees to charge their callers.
- *draft-ietf-sip-refer-04.txt* This document defines the REFER method. This SIP extension requests that the recipient REFER to a resource provided in the request. This can be used to enable many applications, including Call Transfer.
- *draft-uusitalo-sipping-authentication-00.txt* 3GPP Requirements for SIP Authentication

H323 Suite: H5300

The ITU-T Study Group 16 is working on multimedia security for H.323 mobility. One of the general problems addressed aims at enabling mobility in small and medium sized H.323-based corporate and enterprise networks.

Working group(s) homepage(s): `http://www.itu.int/ITU-T/studygroups/com16/index.html`

Protocol description:

- *H.530* in `http://www.itu.int/rec/recommendation.asp`. In the scenario envisioned, a mobile H.323 user would attach at a foreign visited H.323 network domain in order to obtain H.323 services from the corresponding subscribed home domain; e.g. forwarded H.323 Voice-over-IP calls. The security issues of concern in such a scenario include mutual authentication of the mobile user and the visited domain, authentication of the visited domain to the home domain and secure key agreement between the mobile user terminal and the visited gatekeeper.

Next Steps in Signalling (nsis)

When considering end-to-end communication, it is likely that several administrative domains are traversed. Interworking of authentication, authorization, accounting, and QoS mechanisms between the domains is problematic. The existing (weak) trust relationships between them, the mobility and roaming additionally add complexity to the picture. As with nearly all work being done in the IETF, security is a very important concern for NSIS. The working group studies the security requirements for QoS Signaling, and is producing a threat analysis of QoS signaling.

Working group(s) homepage(s): nsis-charter.html

Requirements description:

- *draft-ietf-nsis-req-02.txt* Requirements document, including security.
- *draft-hancock-nsis-framework-00.txt* Framework
- *draft-westberg-nsis-edge-edge-framework-00.txt* Edge-to-edge framework
- *draft-tschofenig-nsis-threats-00.txt* Description of threats for the nsis protocol

applications such as SIP or location services

geopriv

Already, the first "location-based services" have been launched, capable of sending text messages to mobile phone users in particular network cells. Applications include: asset and vehicle tracking, information and proximity services, interactive games, location-sensitive messaging, location-sensitive portal services, maps and directions, network management systems, personal and employee safety systems, targeted advertising and promotions, telematics, traffic services, and many more.

If not properly addressed, privacy concerns are as a long-term impediment to the success of e-business ventures and in particular of location dependent value added services. A protocol will be needed to securely transfer private information to a server and, more importantly, to authorize this server to release securely some information to a client. The main principle has to be that the "user" may specify which information may be released to whom under which further conditions.

To quote the International Working Group on Data Protection in Telecommunications of the OECD: "The enhanced precision of location information and its availability to parties other than the operators of mobile telecommunications networks create new unprecedented threats to the privacy of the users of mobile devices linked to telecommunications networks. The Working Group therefore deems it necessary that the technology for locating mobile devices is designed as little invasive to privacy as possible."

Working group(s) homepage(s): geopriv-charter.html

Requirements description:

- *draft-cuellar-geopriv-reqs-02.txt*
- *draft-cuellar-geopriv-scenarios-00.txt* scenarios

Instant Messaging and Presence Protocol (impp) and SIP for Instant Messaging and Presence Leveraging Extensions (simple)

Instant Messaging (IM) refers to the transfer of messages between users in near real-time. These messages are usually, but not required to be, short. IMs are often used in a conversational mode, that is, the transfer of messages back and forth is fast enough for participants to maintain an interactive conversation.

Working group(s) homepage(s): `http://search.ietf.org/html.charters/impp-charter.html` `http://search.ietf.org/html.charters/simple-charter.html`

Protocol description:

- *draft-ietf-impp-cpim-02.txt*
- *draft-mankin-im-session-guide-00.txt*
- *draft-riikonen-presence-attrs-00.txt*
- *draft-ietf-simple-presence-06.txt*
- *draft-ietf-sip-message-04.txt*

15.2 Security Infrastructure

Kerberos

The Kerberos protocol provides a mechanism for mutual authentication between entities before a secure network connection is established. It defines how clients interact with a network authentication service. Clients obtain tickets from the Kerberos Key Distribution Centre (KDC), and they present these tickets to servers when connections are established. Kerberos tickets represent the client's network credentials. This protocol is increasingly important in applications.

Working group(s) homepage(s): krb-wg-charter.html

Protocol description:

- *draft-ietf-krb-wg-kerberos-clarifications-00.txt*
- *draft-ietf-cat-kerberos-pk-cross-08.txt* Extension to provide a method for using public key cryptography to enable cross-realm authentication.
- *draft-ietf-cat-iakerb-08.txt* Extension that enables a client to obtain Kerberos tickets for services where the KDC is not accessible to the client, but is accessible to the application server.
- *draft-ietf-krb-wg-hw-auth-01.txt* Passwordless Initial Authentication to Kerberos by Hardware Preauthentication for performing initial authentication of a user without using that user's long-lived password. Any 'hardware preauthentication' method may be employed instead of the password and the key of another principal must be nominated to encrypt the returned credential.

TLS

The Transport Layer Security (TLS) Protocol suite is a standard designed to protect the privacy of information communicated over the Internet. TLS assumes that a connection-oriented transport, typically TCP, is in use. It is divided in two sub-protocols: The TLS Handshake Protocol is responsible for the authentication and key exchange necessary to establish or resume secure sessions. The TLS Record protocol secures application data using the keys created during the Handshake. The Record Protocol is responsible securing application data and verifying its integrity and origin.

Working group(s) homepage(s): tls-charter.html

Protocol description:

- *draft-ietf-tls-rfc2246-bis-01.txt*
- *draft-ietf-tls-extensions-04.txt*

IPsec / IKE

IPsec/IKE is a suite of security protocols in the network layer developed to provide cryptographic security services that flexibly support combinations of authentication, integrity, access control, and confidentiality. The key agreement protocol IKE is seen as already too complex, and complexity leads to security bugs. Therefore in the new version of the IPsec Key Agreement Protocol (known as Son-of-IKE) the IETF will be regarding provable correctness, (formal proofs of security) as one important criteria for choosing the protocol.

Working group(s) homepage(s): ipsec-charter.html

Protocol description:

- `http://search.ietf.org/rfc/rfc2409.txt` IKE
- *draft-ietf-ipsec-properties-01.txt* IPsec / IKE properties and issues

- *draft-spencer-ipsec-ike-implementation-02.txt*
- *draft-ietf-ipsec-sonofike-rqts-00.txt* son-of-ike requirements
- *draft-ietf-ipsec-son-of-ike-protocol-reqts-00.txt* son-of-ike requirements
- *draft-ietf-ipsec-soi-features-00.txt* son-of-ike comparison of features of the proposals
- *draft-ietf-ipsec-revised-identity-00.txt*
- *draft-ietf-ipsec-jfk-03.txt* JFK
- *draft-ietf-ipsec-ikev2-02.txt* IKEv2
- *draft-ietf-ipsec-ikev2-rationale-00.txt* IKEv2 rationale
- *draft-richardson-ipsec-opportunistic-08.txt* Opportunistic encryption using IKE and IPsec.

S/MIME Mail Security (smime)

CMS Symmetric Key Management and Distribution a mechanism to manage (i.e., set-up, distribute, and rekey) keys used with symmetric cryptographic algorithms. Also defined herein is a mechanism to organize users into groups to support distribution of encrypted content using symmetric cryptographic algorithms.

Working group(s) homepage(s): `http://search.ietf.org/html.charters/smime-charter.html`

Protocol description:

- *draft-ietf-smime-symkeydist-07.txt*

AAA

This protocol provides an Authentication, Authorization, and Accounting framework for applications such as network access or IP mobility. Diameter is also intended to work both with local AAA and with roaming situations.

Working group(s) homepage(s): aaa-charter.html use in mobile-ip:

- *draft-ietf-aaa-diameter-mobileip-10.txt*
- *draft-dupont-mipv6-aaa-01.txt*
- *draft-ietf-mobileip-aaa-key-09.txt* Mobile IP requires strong authentication between the mobile node and its home agent. When the mobile node shares a security association with its home AAA server, however, it is possible to use that security association to create derivative security associations between the mobile node and its home agent, and again between the mobile node and the foreign agent currently offering connectivity to the mobile node. This document specifies such extensions to Mobile IP.

Protocol for Carrying Authentication for Network Access (pana)

Most of the commercial networks today require users (or devices on behalf of users) to provide their authentication information, such as identity, device identifier, etc., before being allowed access to network resources. It is expected that future IP devices will have a variety of access technologies to gain network connectivity. Currently there are access-specific mechanisms for providing client information to the network for authentication and authorization purposes. In addition to being limited to specific access media, some of these protocols are limited to specific network topologies.

Working group(s) homepage(s): pana-charter.html

Protocol description:

- *draft-ietf-pana-usage-scenarios-01.txt*
- *draft-le-pana-lsa-tsk-00.txt*
- *draft-engelstad-pana-paa-discovery-00.txt*

IP Security Remote Access (ipsra) and Securely Available Credentials (sacred)

Working group(s) homepage(s): ipsra-charter.html
sacred-charter.html

- *draft-ietf-ipsra-reqmts-05.txt*
- *draft-ietf-sacred-framework-03.txt*
- *draft-ietf-sacred-protocol-bss-02.txt*
- *draft-ietf-ipsra-pic-05.txt* PIC, Pre-IKE Credential Provisioning Protocol. This is a method to bootstrap IPsec authentication via an 'Authentication Server' (AS) and legacy user authentication (e.g., RADIUS). The client machine communicates with the AS using a key exchange protocol where only the server is authenticated, and the derived keys are used to protect the legacy user authentication.

Secure Shell (secsh)

SSH is a protocol for secure remote login and other secure network services over an insecure network.

Working group(s) homepage(s): secsh-charter.html

Protocol description:

- *draft-ietf-secsh-userauth-15.txt* This document describes the SSH authentication protocol framework and public key, password, and host-based client authentication methods.

Secure Password Protocols

Secure Password protocols for authenticated key exchange should be designed to work despite the use of passwords drawn from a space so small that an adversary might well enumerate all possible passwords and run online or offline dictionary attacks.

Working group(s) homepage(s): none

Protocol description:

- *draft-mackenzie-ips-iscsi-pak-00.txt* PAK: Password-Authenticated Key Exchange for iSCSI
- `http://www.ietf.org/rfc/rfc2945.txt` Secure Remote Password Protocol (SRP) Authentication and Key Exchange System
- *draft-jablon-speke-01.txt* The SPEKE Password-Based Key Agreement Methods

The Host Identity Payload (HIP)

The purpose of HIP is to establish a rapid authentication between two hosts and continuity between those hosts independent of the Networking Layer.

Working group(s) homepage(s): none

Protocol description:

- *draft-moskowitz-hip-05.txt*
- *draft-irtf-nsrg-report-03.txt*

Point-to-Point Protocol Extensions (pppext)

Extensible Authentication Protocol (EAP) is an authentication protocol that supports multiple authentication mechanisms. EAP typically runs directly over the link layer without requiring IP and therefore includes its own support for in-order delivery and retransmission. Fragmentation is not supported within EAP itself; however, individual EAP methods may support this. While EAP was originally developed for use with PPP, it is also now in use with IEEE802.

Working group(s) homepage(s): pppext-charter.html

Protocol description:

- *draft-ietf-pppext-rfc2284bis-04.txt*
- *draft-arkko-pppext-eap-aka-03.txt* The Authentication and Key Agreement (AKA) mechanism performs user authentication and session key distribution in Universal MobileTelecommunications System (UMTS) networks. AKA is a challenge-response based mechanism that uses symmetric cryptography. mechanism for authentication and session key distribution using the AKA mechanism and a UMTS Subscriber Identity Module, a smart card like device.
- *draft-niemi-sipping-digest-aka-00.txt*
- *draft-kniveton-sim6-00.txt*
- *draft-berrendo-chabanne-pppext-eapmake-01.txt* Authentication protocol based on EAP, which emphasizes on simplicity and scalability. Authentication is provided through a mechanism derived from the Diffie-Hellman scheme, and it is possible to derive and check a common symmetric key for the purpose of privacy. Scalability is provided by the underlying support of legacy PKI systems.
- *draft-aboba-pppext-key-problem-01.txt* The EAP Session Key Problem, issues involved in key derivation by EAP methods

Secure Network Time Protocol (stime)

Working group(s) homepage(s): stime-charter.html

Protocol description:

For trust models to be truly portable across the Internet, transactions must be anchored so they are comparable. The one shared commodity that can be widely agreed upon is time, and the ability to authenticate the source of the time can assist in providing such portability in trust. The ability to securely obtain time from authenticated sources is thus becoming a key factor in security and non-repudiation. The purpose of this working group is to define the message formats and protocols — specifically, modifications to the existing Network Time Protocol (NTP) — which are necessary to support the authenticated distribution of time for the Internet.

- *draft-ietf-stime-ntpauth-03.txt* This proposes a scheme for authenticating servers to clients using the Network Time Protocol. It extends prior schemes based on symmetric key cryptography to a new scheme based on public key cryptography. The new scheme, called Autokey, is based on the fact that the IPSEC schemes proposed by the IETF cannot be adopted intact, since that would preclude stateless servers and severely compromise timekeeping accuracy.

Further Authentication Proposals

There are several other authentication protocols that are not associated with a single working group.

Protocol description:

- *draft-trostle-slam-00.txt* Secure Lightweight Authentication Mechanism (SLAM) allows network entities to mutually authenticate and establish shared secret keys for protection of application data. This draft specifies confidential and integrity protected messages for application level security (e.g. GSS-API protected messages). This proposal is based on an earlier version of lightweight Kerberos.
- *draft-montenegro-sucv-02.txt* Statistic Uniqueness and Cryptographic Verifiability (SUCV) a general method to solve the "identifier ownership problem", that is, how an entity can prove that it owns a certain identifier (for instance an IP-Address).
- *draft-riikonen-silc-spec-05.txt* Secure Internet Live Conferencing (SILC), Protocol Specification
- *draft-riikonen-silc-ke-auth-05.txt* The Secure Internet Live Conferencing (SILC) Key Exchange and Authentication Protocols provides secure key exchange between two parties resulting into shared secret key material.
- *draft-zandbelt-idsec-01.txt* IDsec: Virtual Identity on the Internet. IDsec is a mechanism that provides an identity for users on the Internet. Identity in IDsec means that a user is known by a certain profile that contains precisely those attributes that the user wants to reveal to the requester of his profile. Access to profile attributes is managed by the user himself. Certificates and public/private key mechanisms ensure that information is exchanged in a secure way only between parties that trust each other.

15.3 E-Commerce Applications

Electronic Data Interchange-Internet Integration (ediint)

Electronic Data Interchange (EDI) is a set of protocols for conducting highly structured inter-organization exchanges, such as for making purchases or initiating loan requests. The current additional requirements for obtaining multi-vendor, inter-operable service, concentrate on security issues such as EDI transaction integrity, privacy and non-repudiation.

Working group(s) homepage(s): ediint-charter.html

Protocol description:

- *draft-ietf-ediint-req-09.txt* This document is a functional specification, discussing the requirements for inter-operable EDI, but also containing high-level description of recommended protocol exchanges to protect EDI traffic.

15.4 Routing and Management Infrastructure

unap

Secure Network Access Using Router Discovery and AAA (SNAP) extend the standard router discovery protocol to use it as a carrier for the authentication purposes.

Working group(s) homepage(s): none

Protocol description:

- *draft-yegin-unap-snard-00.txt*

Service Location Protocol, (svrloc)

The Service Location Protocol (SLP) allows clients to discover services with little or no prior configuration. SLP can be used to advertise services of any kind through the use of URLs.

Working group(s) homepage(s): svrloc-charter.html

Protocol description:

- *draft-guttman-svrloc-rfc2608bis-02.txt*
- *draft-guttman-svrloc-serviceid-01.txt*
- *draft-guttman-svrloc-as-00.txt*

Domain Name System (DNS) Extensions (dnsext)

Security extensions to DNS protocol that provide data integrity and authentication to security aware protocols and applications through the use of cryptographic digital signatures.

Working group(s) homepage(s): dnsext-charter.html

Protocol description:

- `http://www.ietf.org/rfc/rfc2535.txt` DNS Security Extensions
- *draft-ietf-dnsext-dnssec-roadmap-05.txt* DNS Security Document Roadmap: Several documents exist to describe these extensions and the implementation-specific details regarding specific digital signing schemes. The interrelationship between these different documents is discussed here.
- *draft-ietf-dnsext-dnssec-intro-01.txt* DNS Security Introduction and Requirements
- *draft-ietf-dnsext-dns-threats-01.txt* Threat Analysis Of The Domain Name System
- *draft-bellovin-dnsext-bloomfilt-00.txt* Some answers of DNSSEC must be authenticated, which requires complex mechanisms, online storage of the zone's secret key, expensive online computations, or massive zone files. As an alternative, this draft proposes storage of authenticated pointers to Bloom filters.
- *draft-ietf-dnsext-ad-is-secure-05.txt* The AD bit should only set on answers where signatures have been cryptographically verified or the server is authoritative for the data and is allowed to set the bit by policy.
- *draft-josefsson-siked-framework-00.txt* This paper is an introduction to the ongoing debate over whether to store cryptographic keys used by applications in DNS.

Inter-Domain Routing (idr) and Inter-Domain Multicast Routing (idmr)

The Border Gateway Protocol (BGP) is an inter-Autonomous System routing protocol. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. A compromise of BGP would lead to wrong routing of IP packets.

Working group(s) homepage(s): idr-charter.html
idmr-charter.html

Protocol description:

- *draft-ward-bgp-ipsec-00.txt*
- *draft-ietf-idr-rfc2385bis-01.txt* Protection of BGP Sessions via the TCP MD5 Signature Option
- *draft-murphy-bgp-secr-04.txt* BGP Security Analysis
- *draft-murphy-bgp-vuln-00.txt* Vulnerabilities of Border Gateway Protocol (BGP)
- *draft-murphy-bgp-protect-00.txt* BGP Security Protections
- *draft-irtf-gsec-igmpv3-security-issues-01.txt* The protocol used by hosts and routers to communicate their multicast group membership.

Mobile Ad-hoc Networks (manet) and Zero Configuration Networking (zeroconf)

Before a device is installed into a secure network, certain security parameters (such as keys) must be configured. Many common TCP/IP protocols such as DHCP [RFC2131], DNS [RFC1034] [RFC1035], MADCAP [RFC2730], and LDAP [RFC2251] must be configured and maintained by an administrative staff. This is unacceptable for emerging networks such as home networks, automobile networks, airplane networks, or ad hoc networks at conferences, emergency relief stations, and many others. Such networks may be nothing more than two isolated laptop PCs connected via a wireless LAN. For all these networks, an administrative staff will not exist and the users of these networks neither have the time nor inclination to learn network administration skills. Instead, these networks need protocols that require zero user configuration and administration. This document is part of an effort to define such zero configuration (zeroconf) protocols.

Somewhat related is the Peer-to-Peer Messaging Protocol (PPMP). (There is no working group associated with this protocol).

Working group(s) homepage(s): manet-charter.html
zeroconf-charter.html

Protocol description:

- *draft-hanna-zeroconf-seccfg-00.txt*
- *draft-hessing-p2p-messaging-00.txt* The Peer-to-Peer Messaging Protocol (PPMP) supports the membership of servents of one or more communities with each community supporting one or more applications. Every community has a security profile. This profile defines the requirements for servents to join a community and the requirements for the transport of messages. A servent can join a community by making a connection to one of more other servents that are already member of the community and by negotiating with these peers the use of the community.

References

1. P. A. Abdulla, A. Annichini, S. Bensalem, A. Bouajjani, P. Habermehl, and Y. Lakhnech. Verification of infinite-state systems by combining abstraction and reachability analysis. In N. Halbwachs and D. Peled, editors, *Proceedings of CAV'99*, LNCS 1633, pages 146–159. Springer-Verlag, 1999.
2. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 313–321, 1996.
3. R. Accorsi, D. Basin, and L. Viganò. Towards an awareness-based semantics for security protocol analysis. In J. Goubault-Larrecq, editor, *Proceedings of CAV Workshop on Logical Aspects of Cryptographic Protocols Verification*, ENTCS 55(1). Elsevier Science Publishers, 2001.
4. R. Alur, T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Tasiran. MOCHA: Modularity in model checking. In *Proceedings CAV'98*, pages 521–525, 1998.
5. A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Mödersheim, M. Rusinowitch, M. Turuani, L. Viganò, and L. Vigneron. The AVISS Security Protocol Analysis Tool. In E. Brinksma and K. G. Larsen, editors, *Proceedings of CAV'02*, LNCS 2404, pages 349–353. Springer-Verlag, 2002. URL of the AVISS project: `http://www.informatik.uni-freiburg.de/~softech/research/projects/aviss`.
6. A. Armando, C. Castellini, and E. Giunchiglia. SAT-Based Procedures for Temporal Reasoning. In *Proceedings of ECP-99*, pages 98–109. Springer-Verlag, 1999.
7. A. Armando and L. Compagna. Automatic SAT-Compilation of Protocol Insecurity Problems via Reduction to Planning. In *22nd IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems*, Houston, Texas, November 2002. Also presented at the FCS & Verify Workshops, Copenhagen, Denmark, July 2002.
8. A. Armando, J. Gallagher, A. Smaill, and A. Bundy. Automating the Synthesis of Decision Procedures in a Constructive Metatheory. *Annals of Mathematics and Artificial Intelligence*, 22(3-4):259–279, 1998.
9. A. Armando and E. Giunchiglia. Embedding Complex Decision Procedures inside an Interactive Theorem Prover. *Annals of Mathematics and Artificial Intelligence*, 8:475–502, 1993.
10. A. Armando and S. Ranise. Constraint Contextual Rewriting. In *Proceedings of FTP'98*, pages 65–75, 1998.
11. A. Armando and S. Ranise. Constraint Solving in Logic Programming and in Automated Deduction: a Comparison. In F. Giunchiglia, editor, *Proceedings of AIMSA'98*, LNAI 1480, pages 28–38. Springer-Verlag, 1998.
12. A. Armando and S. Ranise. From Integrated Reasoning Specialists to "Plug-and-Play" Reasoning Components. In J. Calmet and J. Plaza, editors, *Proceedings of AISC98*, LNCS 1476, pages 42–54. Springer-Verlag, 1998.
13. A. Armando and S. Ranise. A Practical Extension Mechanism for Decision Procedures. In *Proceedings of FM-TOOLS 2000*, pages 53–57, 2000. Extended version to appear on the Journal of Universal Computer Science.
14. A. Armando and S. Ranise. Termination of Constraint Contextual Rewriting. In H. Kirchner and C. Ringeissen, editors, *Proceedings of FroCoS'2000*, LNCS 1794, pages 47–61. Springer-Verlag, 2000.
15. A. Armando, S. Ranise, and M. Rusinowitch. A rewriting approach to satisfiability procedures. *Information and Computation*, 2002. to appear.
16. A. Armando, A. Smaill, and I. Green. Automatic Synthesis of Recursive Programs: The Proof-Planning Paradigm. *Automated Software Engineering*, 6(4):329–356, 1999.
17. L. Bachmair, I. V. Ramakrishnan, A. Tiwari, and L. Vigneron. Congruence Closure modulo Associativity-Commutativity. In H. Kirchner and C. Ringeissen, editors, *Proceedings FroCoS'2000*, LNCS 1794, pages 242–256. Springer Verlag, 2000.
18. D. Basin. Logical Framework Based Program Development. *ACM Computing Surveys*, 30(3):1–4, 1998.
19. D. Basin. Lazy infinite-state analysis of security protocols. In R. Baumgart, editor, *Secure Networking — CQRE'99*, LNCS 1740, pages 30–42. Springer-Verlag, 1999.
20. D. Basin and G. Denker. Maude versus Haskell: an Experimental Comparison in Security Protocol Analysis. In K. Futatsugi, editor, *ENTCS 36*. Elsevier, 2001.
21. D. Basin and S. Friedrich. Modeling a Hardware Synthesis Methodology in Isabelle. *Formal Methods in Systems Design*, 15(2):99–122, 1999.
22. D. Basin, S. Friedrich, M. Gawkowski, and J. Posegga. Bytecode Model Checking: An Experimental Analysis. In D. Bosnacki and S. Leue, editors, *Model Checking Software, 9th International SPIN Workshop*, LNCS 2318, pages 42–59. Springer-Verlag, 2002.

23. D. Basin, S. Friedrich, J. Posegga, and H. Vogt. Java Byte Code Verification by Model Checking. In *Proceedings of CAV'99*, LNCS 1633, pages 491–494. Springer-Verlag, 1999.

24. D. Basin and N. Klarlund. Automata Based Symbolic Reasoning in Hardware Verification. *Formal Methods in Systems Design*, 13(3):255–288, 1998.

25. D. Basin and B. Krieg-Brückner. Formalization of the Development Process. In E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner, editors, *Algebraic Foundations of System Specification*, pages 521–562, Berlin, 1998. Springer-Verlag.

26. D. Basin, S. Matthews, and L. Viganò. Labelled Propositional Modal Logics: Theory and Practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

27. D. Basin, S. Matthews, and L. Viganò. Labelled Modal Logics: Quantifiers. *Journal of Logic, Language and Information*, 7(3):237–263, 1998.

28. D. Basin, S. Mödersheim, and L. Viganò. An on-the-fly model-checker for security protocol analysis. 2002.

29. D. Basin, F. Rittinger, and L. Viganò. A Formal Analysis of the CORBA Security Service. In D. Bert, J. P. Bowen, M. C. Henson, and K. Robinson, editors, *Proceedings of ZB'2002*, LNCS 2272, pages 330–349. Springer-Verlag, 2002.

30. M. Baugher, T. Hardjono, H. Harney, and B. Weis. The Group Domain of Interpretation. Technical report, Internet Engineering Task Force, `http://www.ietf.org/ietf/1id-abstracts.txt`, 2002.

31. K. Baukus, S. Bensalem, Y. Lakhnech, and K. Stahl. Abstracting ws1s systems to verify parameterized networks. In S. Graf and M. I. Schwartzbach, editors, *Proceedings of TACAS 2000*, LNCS 1785, pages 188–203. Springer-Verlag, 2000.

32. G. Bella, F. Massacci, L. C. Paulson, and P. Tramontano. Formal verification of cardholder registration in SET. In F. Cuppens, Y. Deswarte, D. Gollmann, and M. Waidner, editors, *Proceedings of the 6th European Symposium on Research in Computer Security: ESORICS 2000*, LNCS 1895, pages 159–174. Springer-Verlag, 2000.

33. N. Berregeb, A. Bouhoula, and M. Rusinowitch. Observational Proofs with Critical Contexts. In E. Astesiano, editor, *Fundamental Approaches to Software Engineering - ETAPS'98*, LNCS 1382, pages 38–53, Berlin, 1998. Springer-Verlag.

34. D. Bjørner and J. Cuellar. Software Engineering Education: The Rôle of Formal Specifications and Design Calculi. *Annals of Software Engineering Journal*, 1999.

35. D. Bolignano. Towards the Formal Verification of Electronic Commerce Protocols. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 133–146. IEEE Computer Society Press, 1997.

36. M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th International Conference on Automata, Language and Programming: ICALP'01*, LNCS 2076, pages 667–681. Springer-Verlag, Berlin, 2001.

37. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Proceedings of CAV'00*, pages 403–418, 2000.

38. M. Bozzano and G. Delzanno. Automated Protocol Verification in Linear Logic. In *Principle and Practice of Declarative Languages, PPDP 2002, to appear*, 2002.

39. S. Brackin. Evaluating and Improving Protocol Analysis by Automatic Proof. In *Proceedings of the 11th IEEE Computer Security Foundations Workshop: CSFW'98*. IEEE Computer Society Press, 1998.

40. S. Brackin, C. Meadows, and J. Millen. CAPSL Interface for the NRL Protocol Analyzer. In *Proceedings of ASSET'99, IEEE Symposium on Application-Specific Systems and Software Engineering Technology*. IEEE Computer Society Press, 1999.

41. M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

42. I. Cervesato, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. Relating strands and multiset rewriting for security protocol analysis. In P. Syverson, editor, *Proceedings of the 13th IEEE Computer Security Foundations Workshop: CSFW'00*, pages 35–51. IEEE Computer Society Press, 2000.

43. I. Cervesato and P. F. Syverson. The logic of authentication protocols. In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, LNCS 2171, pages 63–136. Springer-Verlag, 2001.

44. D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *CACM*, 24(2):84–88, Feb. 1981.

45. Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols. In *Proceedings of ASE'01*. IEEE Computer Society Press, 2001. Long version available as Technical Report A01-R-140, LORIA, Nancy (France).

46. Y. Chevalier and L. Vigneron. Towards Efficient Automated Verification of Security Protocols. In *Proceedings of VERIFY'01*, pages 19–33. Università degli studi di Siena, TR DII 08/01, 2001.

47. Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In E. Brinksma and K. G. Larsen, editors, *Proceedings of CAV'02*, LNCS 2404, pages 324–337. Springer-Verlag, 2002.

48. J. Clark and J. Jacob. A Survey of Authentication Protocol Literature: Version 1.0, 17. Nov. 1997. URL: `http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz`.

49. E. Cohen. TAPS: A first-order verifier for cryptographic protocols. In *Proceedings of the 13th Computer Security Foundations Workshop*. IEEE Computer Society Press, 2000.

50. H. Comon, P. Narendran, R. Nieuwenhuis, and M. Rusinowitch. Decision Problems in Ordered Rewriting. In V. Pratt, editor, *Proceedings 13th IEEE Symposium on Logic in Computer Science*, pages 117–125. IEEE Computer Society Press, 1998.

51. P. Cousot. Abstract interpretation. *Symposium on Models of Programming Languages and Computation, ACM Computing Surveys*, 28(2):324–328, 1996.

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

53. J. Cuellar, D. Barnard, and M. Huber. A Solution Relying on the Model Checking of Boolean Transition Systems. In M. Broy, S. Merz, and S. K., editors, *Formal Systems Specification; The RPC-Memory Specification Case Study*, LNCS 1169. Springer-Verlag, 1996.

54. J. Cuellar, D. Barnard, and M. Huber. Rapid Prototyping for an Assertional Specification Language. In B. Steffen and T. Margaria, editors, *Proceedings of TACAS'96*, LNCS 1055. Springer-Verlag, 1996.

55. J. Cuellar and M. Huber. TLT. In C. Lewerentz and L. T., editors, *Formal Development of Reactive Systems, Case Study Production Cell*, LNCS 1165. Springer-Verlag, 1995.

56. J. Cuellar and I. Wildgruber. The Steam Boiler Problem - A TLT Solution. In J. R. Abrial, E. Boerger, and H. Langmaack, editors, *Formal Methods for Industrial Applications. Specifying and Programming the Steam Boiler Control*, LNCS 1165. Springer-Verlag, 1996.

57. J. Cuellar, I. Wildgruber, and D. Barnard. Combining the Design of Industrial Systems with Effective Verification Techniques. In M. Bertran, T. Denvir, and M. Naftalin, editors, *Proceedings of FME 94*, LNCS 873. Springer-Verlag, 1994.

58. G. Delzanno. Automatic verification of parameterized cache coherence protocols. In E. A. Emerson and A. P. Sistla, editors, *Proceedings of CAV'00*, LNCS 1855. Springer-Verlag, 2000.

59. G. Delzanno. Verification of Consistency Protocols via Infinite state Symbolic Model Checking, a Case Study. In *Proceedings of the FORTE XIII/PSTV XX*, pages 171–188. Kluwer, 2000.

60. G. Delzanno. Specifying and Debugging Security Protocols via Hereditary Harrop Formulas and λProlog - A Case-study -. In H. Kuchen and K. Ueda, editors, *Proceedings of FLOPS'01*, LNCS 2024, pages 123–137. Springer-Verlag, 2001.

61. G. Delzanno and T. Bultan. Constraint-based verification of client-server protocols. In *Proceedings of Constraint Programming 2001*, LNCS 2239, pages 286–301. Springer-Verlag, 2001.

62. G. Delzanno, J. Esparza, and A. Podelski. Constraint-based Analysis of Broadcast Protocols. In *Proceedings of CSL'99*, LNCS 1683, pages 50–66. Springer-Verlag, 1999.

63. G. Delzanno and A. Podelski. Model checking in CLP. In R. Cleaveland, editor, *Proceedings of TACAS'99*, LNCS 1579, pages 223–239. Springer-Verlag, 1999.

64. G. Delzanno and A. Podelski. Constraint-based deductive model checking. *Software Tools for Technology Transfer*, 3(3):250–270, 2001.

65. G. Delzanno, J.-F. Raskin, and L. V. Begin. Attacking Symbolic State Explosion. In G. Berry, H. Comon, and A. Finkel, editors, *Proceedings of CAV'01*, LNCS 2102, pages 298–310. Springer-Verlag, 2001.

66. 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/`.

67. B. Donovan, P. Norris, and G. Lowe. Analyzing a Library of Security Protocols using Casper and FDR. In *Proceedings of the Workshop on Formal Methods and Security Protocols*, 1999.

68. N. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. Undecidability of Bounded Security Protocols. In *Proceedings of the FLOC'99 Workshop on Formal Methods and Security Protocols (FMSP'99)*, 1999. Available at `http://www.cs.bell-labs.com/who/nch/fmsp99/program.html`.

69. S. Even and O. Goldreich. On the security of multi-party ping pong protocols. Technical Report 285, Israel Institute of Technology, 1983.

70. F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7:191–230, 1999.

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

72. P. Ferraris and E. Giunchiglia. Planning as Satisfiability in Nondeterministic Domains. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2000.

73. T. Genet and F. Klay. Rewriting for cryptographic protocol verification. In *Proceedings of CADE'00*, LNCS 1831. Springer-Verlag, 2000.

74. E. Giunchiglia, A. Armando, and P. Pecchiari. Structured Proof Procedures. *Annals of Mathematics and Artificial Intelligence*, 15(I):1–18, 1995.

75. E. Giunchiglia, F. Giunchiglia, R. Sebastiani, and A. Tacchella. More Evaluation of Decision Procedures for Modal Logics. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of KR'98*, pages 626–635. Morgan Kaufmann, 1998.

76. E. Giunchiglia, F. Giunchiglia, and A. Tacchella. *SAT, KSATC, DLP and TA: a comparative analysis. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Collected Papers from the International Description Logics Workshop (DL'99)*. CEUR, 1999.

77. E. Giunchiglia, F. Giunchiglia, and A. Tacchella. SAT-Based Decision Procedures for Classical Modal Logics. *Journal of Automated Reasoning*, 2000.

78. E. Giunchiglia, M. Maratea, A. Tacchella, and D. Zambonin. Evaluating Search Heuristics and Optimization Techniques in Propositional Satisfiability. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of IJCAR'2001*, LNAI 2083, pages 347–363. Springer-Verlag, 2001.

79. D. Gollmann. *Computer security*. John Wiley & Sons, 1999.

80. J. Goubault-Larrecq. A method for automatic cryptographic protocol verification (extended abstract). In *Proceedings of IPDPS'00*, LNCS 1800, pages 977–984. Springer-Verlag, 2000.

81. T. A. Henzinger, S. Qadeer, and S. K. Rajamani. You assume, we guarantee: Methodology and case studies. In *Proceedings of CAV'98*, pages 440–451, 1998.

82. IST 2002 Workprogramme.

83. F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Narrowing Cryptographic Protocols. Technical Report 99-R-303, LORIA, Vandoeuvre les Nancy, Dec. 1999. URL: `www.loria.fr/equipes/protheo/SOFTWARES/CASRUL`.

84. 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.

85. H. Kautz, H. McAllester, and B. Selman. Encoding Plans in Propositional Logic. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of KR'96*, pages 374–384. Morgan Kaufmann, 1996.

86. H. Kautz and B. Selman. BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving. In *Working notes of the Workshop on Planning as Combinatorial Search, held in conjunction with AIPS-98*, 1998.

87. Kerberos: The Network Authentication Protocol. URL: `http://web.mit.edu/kerberos/www/`.

88. 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.

89. V. Lotz. Threat Scenarios as a Means to Formally Develop Secure Systems. *Journal of Computer Security*, 5:31–67, 1997.

90. V. Lotz. Ein methodischer Rahmen zur formalen Entwicklung sicherer Systeme. In *Arbeitskonferenz Systemsicherheit*, pages 31–67. Vieweg Verlag, 2000.

91. V. Lotz. Formally Defining Security Properties with Relations on Streams. *Electronic Notes on Theoretical Computer Science*, 32, 2000.

92. V. Lotz, V. Kessler, and G. Walter. A Formal Security Model for Microprocessor Hardware. In *Proc. of FM'99 World Congress on Formal Methods*, LNCS 1708, pages 718–737. Springer-Verlag, 1999.

93. V. Lotz, V. Kessler, and G. Walter. A Formal Security Model for Microprocessor Hardware. *IEEE Transactions on Software Engineering*, 26(8):702–712, Aug. 2000.

94. V. Lotz and G. Walter. Formally Modelling Hardware Processor Security. In *Proc. EUROSMART Security Conference*, pages 361–373, June 2000.

95. G. Lowe. A hierarchy of authentication specifications. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop: CSFW'97*, pages 31–43. IEEE Computer Society Press, 1997.

96. 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/`.

97. 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.

98. The Maude System. URL: `http://maude.csl.sri.com`.

99. K. L. McMillan. A compositional rule for hardware design refinement. In *Proc. 9th International Computer Aided Verification Conference*, pages 24–35, 1997.

100. C. Meadows. The NRL Protocol Analyzer: An Overview. *Journal of Logic Programming*, 26(2):113–131, 1996. See `http://chacs.nrl.navy.mil/projects/crypto.html`.

101. C. Meadows. Open issues in formal methods for cryptographic protocol analysis. In *Proceedings of the DARPA Information and Survivability Conference and Exposition: DISCEX 2000*, pages 237–250. IEEE Computer Society Press, January 2000.

102. J. Mitchell, M. Mitchell, and U. Stern. Automated Analysis of Cryptographic Protocols Using Murphi. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 141–153, 1997.

103. J. Mitchell, V. Shmatikov, and U. Stern. Finite-state analysis of SSL 3.0. In *Seventh USENIX Security Symposium*, pages 201–216, 1998.

104. D. Monniaux. Abstracting cryptographic protocols with tree automata. In *Sixth International Static Analysis Symposium (SAS'99)*, LNCS 1694. Springer-Verlag, 1999.

105. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, 2001.

106. P. Narendran, M. Rusinowitch, and R. Verma. RPO constraint solving is in NP. In G. Gottlob, E. Grandjean, and K. Seyr, editors, *Computer Science Logic*, LNCS 1584, pages 385–398, Berlin, 1998. Springer-Verlag.

107. L. C. Paulson. Mechanized proofs for a recursive authentication protocol. In *10th Computer Security Foundations Workshop*, pages 84–95. IEEE Computer Society Press, 1997.

108. L. C. Paulson. The Inductive Approach to Verifying Cryptographic Protocols. *Journal of Computer Security*, 6(1):85–128, 1998.

109. L. C. Paulson. Inductive analysis of the internet protocol TLS. *ACM Transactions on Computer and System Security*, 2(3):332–351, 1999.

110. The PVS Specification and Verification System. URL: `http://pvs.csl.sri.com`.

111. A. Roscoe and P. J. Broadfoot. Proving security protocols with model checkers by data independence techniques. *Journal of Computer Security*, page 147, 1999.

112. A. W. Roscoe. Modelling and verifying key-exchange protocols using CSP and FDR. In *Proceedings of the 8th IEEE Computer Security Foundations Workshop CSFW'95*, pages 98–107. IEEE Computer Society Press, 1995.

113. A. W. Roscoe and M. Goldsmith. The perfect "spy" for model-checking cryptoprotocols. In *Proceeding of DIMACS Workshop on Design and Formal Verification of Crypto Protocols*, 1997.

114. M. Rusinowitch. The practice of cryptographic protocols verification. In J. Goubault-Larrecq, editor, *CAV Workshop on Logical Aspects of Cryptographic Protocols Verification*, Electronic Notes in Theoretical Computer Science ENTCS 55(1). Elsevier Science Publishers, July 2001. Invited talk.

115. M. Rusinowitch, S. Stratulat, and F. Klay. Mechanical Verification of an Incremental ABR Conformance Algorithm. In A. Emerson and P. Sistla, editors, *Proceedings of CAV'2000*. Springer-Verlag, 2000.

116. M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions is NP-complete. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 2001.

117. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions and composed keys is np-complete. *Theoretical Computer Science A*, 2002. to appear.

118. P. Ryan and S. Schneider. An attack on a recursive authentication protocol. *Information Processing Letters 65*, 1998.

119. S. Schneider and A. Sidiropoulos. CSP and anonymity. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *Proceedings of ESORICS'96*, LNCS 1146, pages 198–218. Springer-Verlag, 1996.

120. B. Schneier. *Applied Cryptography*. John Wiley & Sons, New York, 1996.

121. V. Shmatikov and D. Hughes. Defining anonymity and privacy. In *Workshop on Issues in the Theory of Security (WITS '02)*, 2002.

122. I. Sommerville. *Software Engineering*. Addison-Wesley, 2000.

123. D. Song, S. Berezin, and A. Perrig. Athena: a novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 9:47–74, 2001.

124. P. F. Syverson and S. G. Stubblebine. Group principals and the formalization of anonymity. In *World Congress on Formal Methods (1)*, pages 814–833, 1999.

125. The AVISS Team. Deliverable 1.3: Final project report. URL of the AVISS project: `http://www.informatik.uni-freiburg.de/~softech/research/projects/aviss`, 2002.

126. The Common Criteria Implementation Board (CCIB). The common criteria for information technology security evaluation (cc) version 2.1. http://csrc.nist.gov/cc/ccv20/.

127. L. Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, Dordrecht, 2000.

128. L. Vigneron. Positive Deduction modulo Regular Theories. In H. Kleine-Büning, editor, *Proceedings of Computer Science Logic*, LNCS 1092, pages 468–485. Springer-Verlag, 1995. URL: `www.loria.fr/equipes/protheo/SOFTWARES/DATAC/`.

129. L. Vigneron. Automated Deduction Techniques for Studying Rough Algebras. *Fundamenta Informatica*, 33(1):85–103, 1998.

130. L. Vigneron and A. Wasilewska. Rough Sets based Proofs Visualisation. In R. N. Davé and T. Sudkamp, editors, *Proceedings of NAFIPS'99*, pages 805–808. IEEE Computer Society Press, 1999.

131. D. von Oheimb and V. Lotz. Formal Security Analysis with Interacting State Machines. In *Proceedings of ESORICS 2002*. Springer-Verlag, 2002.

132. C. Weidenbach. Towards an Automatic Analysis of Security Protocols. In H. Ganzinger, editor, *Proceedings of CADE'99*, LNCS 1632, pages 378–382. Springer-Verlag, 1999.

133. J. Zhou and D. Gollmann. Towards verification of non-repudiation protocols. In *Towards verification of non-repudiation protocols. Proceedings of 1998 International Refinement Workshop and Formal Methods Pacific, pages 370–380, Canberra, Australia, September 1998.*, 1998.

Appendix A: Consortium Description

LIST OF PARTICIPANTS						
Role	**Number**	**Participant name**	**Participant short name**	**Country**	**Date enter project**	**Date exit project**
CO (C)	1	Università di Genova	UNIGE (or CO1)	IT	Start of project	End of project
CR (P)	2	Institut National de Recherche en Informatique et en Automatique	INRIA (or CR2)	F	Start of project	End of project
CR (P)	3	Swiss Federal Institute of Technology, Zurich	ETHZ (or CR3)	CH	Start of project	End of project
CR (P)	4	Siemens Aktiengesellschaft	SIEMENS (or CR4)	DE	Start of project	End of project

The consortium combines partners with complementary scientific and practical competence. The group from UNIGE has considerable experience on the combination of decision procedures and their application to different problem domains [86, 10, 11, 12, 13, 14, 74, 75, 76, 77]. The group from INRIA has been working on theoretical foundations and system support for automated deduction for many years [17, 50, 115, 128, 129, 130]. The group from ETHZ has broad experience in formal methods, model-checking, and application of model-checking to security [18, 19, 21, 22, 23, 24, 25, 26, 27, 29, 127]. These groups are leaders in their respective areas and have a long and successful history of international collaboration and strong bilateral relations. Moreover, each partner is among the leading experts on one of the three techniques upon which this project is based. The group from SIEMENS is routinely designing protocols and actively participating in their standardization at IETF, ITU, 3GPP, and W3C. It is among the few industrial groups utilizing formal methods for the validation of industrial-scale security-sensitive applications, e.g. [34, 53, 54, 55, 56, 57, 92, 93, 94].

The coordinator, Dr. Armando from the University of Genova, previously led the UNIGE group in AVISS, the FET Open Assessment Project that is the predecessor to AVISPA, and which we summarize in Section 14. In addition, he is leading the UNIGE group in the context of the EU funded Research Training Network *CALCULEMUS: Systems for Integrated Computation and Deduction* (HPRN-CT-2000-00102) and led two international projects with Great Britain and France funded by CRUI *(Conferenza dei Rettori delle Università Italiane)*. Moreover he is coordinator of a project for the internationalization of PhD programs involving the University of Edinburgh (Scotland), the University of Saarbrücken (Germany), the University of Linz (Austria), and the University of Genova.

A.1 UNIGE – Università di Genova, Dipartimento di Informatica Sistemistica e Telematica

Research in the Artificial Intelligence Group of DIST (Department of Information, Computers and Systems Science) at the University of Genova is focused on the design of automated reasoning tools and their effective use in a variety of application areas such as hardware and software verification and synthesis, planning, and validation of safety or security critical systems.

The group has been involved in several national projects including a project on the design of a SAT-based model-checking tool with application to the validation of safety critical systems and a project for the co-tutoring of PhD students in a network of European Research Institutions. Internationally, the laboratory has been supported by two grants with Germany (University of Freiburg and University of Saarbrücken), a grant with France (INRIA Lorraine), and a grant with the UK (University of Edinburgh). The group is also

partner of the EU Research Training Network "CALCULEMUS: Systems for Integrated Computation and Deduction" and was a partner of the AVISS FET Open Assessment Project.

The group will benefit of the administrative support from DIST, which has a long record of international projects (more than 70 EU-funded projects since 1984).

People Involved

Dr. Alessandro Armando (`http://www.mrg.dist.unige.it/~armando`) is assistant professor at DIST. In February 2001 he got the qualification of associate professor in Computer Engineering. His appointments include a research position at the University of Edinburgh (1994-1995) and one at INRIA-Lorraine, Nancy (1998-1999). His research focuses on the integration of automated reasoning tools [9, 10, 11, 12, 14, 74] and their application to the verification and synthesis of programs and hardware circuits [8, 15, 16], the automatic analysis of security protocols via reduction to planning and propositional logic [7], and to the automation of temporal reasoning [6]. He is scientific representative for DIST of the EU-funded Research Training Network CALCULEMUS. He is member of the Steering Committees of the "First Order Theorem Proving" and of the "Frontiers of Combining Systems" Workshop Series as well as of the International Joint Conference on Automated Reasoning (IJCAR). He is the author of more than 40 research works in international journals and conferences. He has been program committee member of a number of international workshops and conferences and program chair of the 4th International Workshop on Frontiers of Combining Systems (FroCoS 2002).

Dr. Giorgio Delzanno (`http://www.disi.unige.it/person/DelzannoG`) is assistant professor at the Department of Computer Science (DISI) of the University of Genova. In July 2002 he got the qualification of associate professor in Computer Science. He received his PhD in 1997 from the University of Pisa. He was postdoctoral student at the Max-Planck-Institut für Informatik from October 1997 to October 1999. His research interests include verification of protocols, concurrent and distributed systems [58, 59, 61, 62, 63, 64] data structures for specialized constraint solvers [65], and practical applications of constructive logics [60].

Prof. Mauro Di Manzo (`http://www.mrg.dist.unige.it/~mauro`) is full professor at DIST. He became full professor of Computer Science at the University of Ancona in 1987 and since 1991 he has been at the University of Genova. He was director of the Department of Computer Science at the University of Ancona from 1987 until 1991, and director of DIST from 1995 until 2002. He was national coordinator of the National Project on Artificial Intelligence from 1989 until 1994, and has been chairman of the National Research Council committee for the Research Program in AI since 1995. He was founder and vice-president of the Italian Association for Artificial Intelligence. His research interests include multi-agent systems and automated reasoning.

Prof. Dr. Enrico Giunchiglia (`http://www.mrg.dist.unige.it/~enrico`) is associate professor at DIST. In June 2001, he got the qualification of full professor in Computer Engineering. He is Area editorial committee of the journal "Electronic Transactions on Artificial Intelligence", he has been and will be Program Co-Chair of various international events (among them, some relevant for the project, are the "13th International Conference on Automated Planning & Scheduling (ICAPS 2003)" and the "Sixth International Symposium on the Theory and Applications of Satisfiability Testing (SAT 2003)"), and he has been Program Committee of various international conferences in the field of Artificial Intelligence. He will be guest editor of a special issue of the journal "Artificial Intelligence", devoted to Nonmonotonic Reasoning. He has coordinated MURST, CNR, ASI and industrial projects. He is responsible for an ongoing cooperation with Intel Corp. on "SAT Solvers for Symbolic Model Checking and Formal Verification".

Expertise: SAT-based State-Exploration

The Genova group has a long history of research in SAT-based state-exploration [9, 74, 75, 76, 77] and in the combination and integration of automated reasoning tools [10, 11, 12, 6, 14]. The group has shown how it is possible to define different encodings and/or search heuristics for verifying properties of domains specified in

a variety of formalisms [72, 86]. Some of the proposed encodings and search heuristics led to improvements in the overall performances of the system of several orders of magnitude. More recently the group has shown how protocol insecurity can be reduced to planning and, in turn, to a sequence of SAT problems [7]. The group has also a solid experience in the design and development of state-of-the-art SAT-solvers [78] (URL: `http://www.mrg.dist.unige.it/~starsat`).

A.2 INRIA Lorraine, Cassis Group

INRIA (National Institute for Research in Computer Science and Control) is a French public-sector scientific and technological institute. The research carried out at INRIA brings together experts from the fields of computer science and applied mathematics. INRIA gathers in its premises around 2,100 people including 1,600 scientists, many of which belong to partner organizations (CNRS, industrial labs, universities) and work on common projects.

This FET Open project at INRIA Lorraine will be based on the Cassis group (created in January 2002). The members of the Cassis group are well-known for more than a decade of research on term rewriting systems, constraint logic programming and automated deduction. The group has developed several widely distributed software packages, including SPIKE (an induction based theorem-prover), daTac [128] (a first-order theorem-prover for associative-commutative theories). Current research in the Cassis group focuses on designing tools for system specification and verification using automated deduction, model-checking, and constraint solving. Based on these techniques the group has developed a security protocol verifier, Casrul.

At the national level, Cassis participates in a Vernam project (ACI Cryptologie) on security protocol verification and to an RNTL project on tests generation.

The members of the Cassis group have been involved in the Basic Research Esprit project CCL and in the Working Group CoFI, in the COMPULOG network of excellence and in the ERCIM Working Groups, and collaborates with numerous foreign institutes. Cassis has industrial collaborations on formal verification with France-Telecom R&D.

People Involved

Dr. Michaël Rusinowitch (`http://www.loria.fr/~rusi`) received a Thèse d'État in Computer Science in 1987 at the University Henri Poincaré in Nancy. Since 1994 he is Directeur de Recherche at INRIA. His research is mainly concerned with theorem-proving, term-rewriting, and their application to software verification. He contributed to the development of automated deduction with constraints (e.g. [50, 106]), to new proof methods based on induction and rewriting (e.g. [33]) and to the verification of security protocols [83, 116, 117]. Dr. Rusinowitch has been responsible in 1998/99 for an INRIA Cooperative Research Action on the Validation of Infinite State Systems. He is currently leader of the Cassis group at INRIA Lorraine. He has published his works in 30 international conferences and 20 journal papers, and is the author of a book on automated deduction. He has been a member of program committees for several international conferences and co-chairman of the conference on Rewriting Techniques and Applications, was invited speaker at LPAR'00 and RTA'01.

Expertise: Theorem-Proving with Constraints

In a previous work [83], the group has shown how to automatically translate standard descriptions of security protocols into rewrite rules. This both defines an operational semantics for protocol executions and permits one to simulate protocol execution by exploiting the built-ins of the first-order prover daTac that was developed in the same group [128]. Flaws can be detected by deriving an inconsistency in the theory associated with the protocol. During this work the group has developed a prototype translator CASRUL [83] whose target language can serve as a basis for the common declarative format to be used by the project partners. Furthermore, the group plans to develop theorem-proving and constraint-solving strategies that are well-adapted both to the high-level syntax used in this project and to the back-ends applied for analyzing the protocols.

A.3 ETHZ – Swiss Federal Institute of Technology, Zurich

The Swiss Federal Institute of Technology Zurich (ETH Zurich, or simply ETHZ) was founded by the Swiss government in 1854 as a polytechnic and opened its doors in Zurich in 1855. Until 1969 it was the only national university in Switzerland. The ETH Zurich is an institution of the Swiss Confederation dedicated to higher learning and research. Together with the ETH Lausanne and four research institutes it forms the federally directed, and to a major degree financed, ETH domain. Since 1993 (when the 3rd Framework Programme of the European Union started and Switzerland first took part in it), ETH Zurich has been involved in 377 EU-projects. The University has central infrastructure for, and considerable experience with, the administration of EU-projects.

Prof. Dr. David Basin founded and headed the Institute for Software Engineering at the Albert-Ludwigs-Universität Freiburg (ALUFR) from 1997 to 2002. In January 2003, he will start at ETH Zurich as the professor for Information Security. Research in Zurich will be focused around all aspects of information security, including security protocols.

Over the last 3 years, the Institute for Software Engineering of ALUFR, has been involved in several national projects including a Germany Economic Ministry project on generating security infrastructures for distributed applications from high-level models, a DFG project on applications of formal methods to verifying multilateral security properties of distributed systems, and a BMBF-funded VIROR project, where the institute has developed multi-medial educational material on computer security. Internationally, aside from coordinating and participating in the AVISS project, the institute has been supported by two DAAD grants with France (Procope) and Italy (Vigoni), supporting collaborative work on formal methods for secure systems.

The Institute for Software Engineering has also worked on several projects with German industry related to the proposed project. The institute has had projects in 1998–2000 funded by Deutsche Telekom, where they developed formal methods for reasoning about security properties of mobile code, in particular Java byte-code. Since 2000, several doctoral positions have been funded by the German company "Interactive Objects GmbH", which support the development of formal methods for secure electronic commerce, in particular developing verification and test methodologies for validating software architectural descriptions and protocols used by Interactive Objects in their e-commerce applications. The institute is also partner of the IST Working Group "Computer-Assisted Reasoning Based on Type Theory (TYPES)", which started in 2000.

People Involved

Prof. Dr. David Basin (`http://www.informatik.uni-freiburg.de/~basin`) received his Ph.D. in Computer Science from Cornell University in 1989 and his habilitation in Informatik from the University of Saarbrücken in 1996. From 1997 to 2002 he was a full professor at the University of Freiburg where he headed the Institute for Software Engineering. From January 2003 will be a full professor at the ETHZ. His research focuses on methods for the specification, development, and verification of security and mission critical systems, as well as computer support for these activities. He has published more than 20 journal papers and 60 papers in international conferences on these and related topics, e.g. [18, 19, 21, 23, 24, 25, 26, 27]. Recent IT Security projects include developing, with Deutsche Telekom, methods for automatically analyzing security properties of Java byte-code [22, 23], developing new methods for modeling and analyzing security protocols [3, 19] and developing techniques for modeling security services for distributed (middleware based) systems [29]. He has taught numerous courses on IT Security, organized, or been a member of, the program committee of over 50 international conferences and workshops, has organized a "Dagstuhl Seminar" on security protocols, and is an editor of both "Higher-Order and Symbolic Computation" and "Acta Informatica".

Dr. Luca Viganò (`http://www.informatik.uni-freiburg.de/~luca`) received his Masters in Electronic Engineering from the University of Genova in 1994 and his Ph.D. in Computer Science from the University of Saarbrücken in 1997. His appointments include a research position at the Max-Planck-Institut für Informatik in Saarbrücken (1994-1997), and an assistant professor position at the Institute for Software Engineering at

the University of Freiburg (1997-2002). From January 2003 he will be a senior research scientist at ETHZ. His research focuses on methods for the specification, verification, and construction of secure systems. His work includes foundational work on the theory and applications of non-classical and security logics, of proof development systems, and of logical frameworks. On these topics he has co-organized several classes and seminars, and has published a book and more than 20 papers in international journals and conferences, e.g. [3, 26, 27, 28, 29, 127].

Expertise: On-the-fly Model-Checking

The ETHZ group has developed a model-checker for security protocols that offers a number of advantages over conventional model-checking approaches [19]. In this work, a protocol and an attacker model give rise to an infinite state space that formalizes the interleaving semantics of the protocol. The group has developed specialized data structures and algorithms that are then used to represent and compute with the infinite state-space associated with a protocol. The key idea is that the state-space is constructed in a lazy, "on-the-fly" way, i.e. in a demand-driven fashion, where heuristics can be easily integrated with state-space construction.

A.4 Siemens – Siemens Aktiengesellschaft, Corporate Technology

Siemens, headquartered in Berlin and Munich, is one of the world's largest electrical engineering and electronics companies, founded more than 150 years ago. In the fiscal year 2001 (ended September 30, 2001), the company had roughly 484,000 employees and posted sales of EUR 87 billion, an increase of 12 percent over the previous year. Net income including all special items and special charges totaled EUR 2.1 billion.

Siemens boasts an impressive international presence, focusing on the core business areas of Information and Communications, Automation and Control, Power, Transportation, Medical and Lighting. The company currently does business in over 190 countries around the world, generating 80 percent of its sales outside of Germany. Siemens operates more than 600 facilities in over 50 countries.

Siemens' international business is conducted by thirteen operating groups. Recently Siemens has repeatedly strengthened its market position through targeted strategic acquisitions. Following the acquisition of the U.S. company Shared Medical Systems, Siemens Medical Solutions is now the world's largest provider of IT services in the health care industry. The integration of the activities acquired from Atecs Mannesmann created leading suppliers in logistics and automotive electronics. In the high-growth field of e-logistics, Siemens Dematic will attain a world-leading position as a complete provider of logistics solutions. The product portfolio of Siemens VDO Automotive is focused on the highest-growth fields in the automotive supply industry — including infotainment, engine management and passenger safety systems.

In fiscal 2001, spending on research and development rose to more than EUR 6.8 billion, or over eight percent of sales, underscoring the enormous importance of innovation at Siemens. The company submits approximately 9,000 invention disclosure reports each year. According to the official records, Siemens is the largest patent applicant at both the German Patent and Trade Mark Office and the European Patent Office. This is made possible by the efforts of some 60,000 R&D employees at Siemens, approximately one-third of whom work outside of Germany. The company has its own R&D departments in over 30 countries. Roughly three-fourths of all Siemens' products and services are less than five years old.

Work in AVISPA will be conducted by the Security department of Siemens Corporate Technology, Information & Communications. Corporate Technology, with its 1,800 employees, is responsible for research, development and consulting in strategically important technologies, development of business scenarios and identification of new applications, markets and businesses, safeguarding the company's interests in the area of intellectual property, and corporate functions of quality management, environmental affairs & technical safety, standardization & regulation and Information Research Center.

The security department consists of 45 researchers working on a broad spectrum of security topics including cryptography, formal methods in security analysis, Internet and multimedia security, security for e-business, and security for mobile communications. People working in AVISPA will come from both the formal methods and the mobile communications security group.

People Involved

Dr. Jorge Cuellar Jorge R. Cuellar studied mathematics (BA. and MA.) at the Universidad de los Andes, Bogota and obtained a Ph.D. from the University of Mainz. He was faculty member of the Ohio State University and Universidad de los Andes. Since 1987 he has been with Siemens, where he is Principal Research Scientist and has held visiting teaching positions at Technical University of Chemnitz, Technical University of Munich, University of Dortmund, University of Freiburg, and the University of Canterbury (Christchurch, New Zealand). He has worked in operating systems, formal methods, neural networks, performance, network and mobile security and Internet protocols. He has been in the editorial board of Science of Computer Programming (Elsevier).

Volkmar Lotz Volkmar Lotz has received his diploma in Computer Science from the University of Kaiserslautern in 1988. He joined Siemens Corporate Technology in 1989, first in the Software Engineering Department, then, since 1994, in the Security Department. He is currently leading the Formal Methods in Security Analysis group, focusing on security requirements engineering, evaluation and certification, and cryptographic protocol verification. He has been the main contributor to the LKW model [92, 93, 94], a formal security model for smart card processors, which allowed for the Infineon SLE66 processor to be the first smartcard hardware being certified according to ITSEC E4 and Common Criteria EAL5. He is currently preparing a PhD thesis on formal development of secure systems based on relations on streams [89, 90, 91].

Appendix B: Contract Preparation Forms