INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team SECSI

Sécurité des systèmes d'information

Futurs

THEME SYM

Table of contents

1. Team

SECSI is a project common to INRIA and the Laboratoire Spécification et Vérification (LSV), itself a common lab between CNRS (UMR 8643) and the École Normale Supérieure (ENS) de Cachan.

Head of project-team
Jean Goubault-Larrecq [Professor, ENS Cachan]

Vice-head of project-team
Hubert Comon-Lundh [Professor, ENS Cachan - until Nov. 30]
Florent Jacquemard [INRIA Research Scientist - since Dec. 01]

Staff member, INRIA
Steve Kremer [Research Scientist]
Julien Olivain [Junior technical staff]

Staff member, CNRS
Stéphane Demri [Research Scientist]

Staff member, ENS Cachan
Ralf Treinen [Associate professor, in "délégation" INRIA since Sep. 01]
Fabrice Parrennes [Part-time teaching and research assistant, until May 31]

PhD students
Mathieu Baudet [Corps des Télécoms INRIA grant]
Vincent Bernat [Student at ENS Cachan]
Stéphanie Delaune [CIFRE grant with France Télécom R&D, École Doctorale Sciences Pratiques (Cachan)]
Pascal Lafourcade [MENRT grant on ACI "sécurité" Rossignol, École Doctorale Sciences Pratiques (Cachan) & Université de Provence (Marseilles), since Oct. 01]
Benjamin Ratti [MENRT grant, École Doctorale Sciences Pratiques (Cachan), since Oct. 01]
Yu Zhang [MENRT grant on ACI "cryptologie" funding, École Doctorale Sciences Pratiques (Cachan)]

2. Overall Objectives

This section is unchanged from the SECSI 2003 report.

SECSI is a common project between INRIA Futurs and the LSV (Laboratoire Spécification et Vérification), itself a common research unit of CNRS (UMR 8643) and the ENS (École Normale Supérieure) de Cachan.

The SECSI project is a research project on the security of information systems. It is organized around three main themes, and their mutual relationships:

- Automated verification of cryptographic protocols;
- Intrusion detection;
- Static analysis of programs, in order to detect security holes and vulnerabilities at the protocol level.

The objectives of the SECSI project are:

- to design new models and new logics for describing security properties: secrecy, authentication, anonymity, privacy, fair exchange, resistance to dictionary attacks, etc;
- to design and implement new automated cryptographic protocol verification algorithms;
- to invent, improve, implement and experiment with new model-checking techniques, particularly on-line model-checking techniques, with application to intrusion detection;
- to design and implement new static analysis techniques to evaluate the level of assurance of actual cryptographic code;
- to integrate static analysis techniques and dynamic monitoring techniques (intrusion detection).

3. Scientific Foundations

3.1. What is computer security? Do we need some?

Keywords: *computer security*, *cryptographic protocol*, *intrusion detection*, *model-checking*, *static analysis*, *verification*.

This section is unchanged from the SECSI 2003 report.

verification see model-checking.

model-checking a set of automated techniques aiming at ensuring that a formal model of some given computer system satisfies a given specification, typically written as a formula in some adequate logic.

protocol a sequence of messages defining an interaction between two or more machines, programs, or people.

cryptographic protocol a protocol using cryptographic means, in particular encryption, that attempts to satisfy properties of secrecy, authentication, or other security properties.

static analysis set of automated techniques that determine some properties satisfied by given programs, without having to execute them; based on analyzing source code, sometimes object code; essentially identical to abstract interpretation of programs.

intrusion detection set of methods attempting to detect attacks, intrusions, or anomalies in computer systems, by real-time monitoring networks and systems.

Security has been getting more and more attention recently, as attacks against even personal computers (viruses, worms, spam), or banking cards, or mobile phones, etc., are becoming more and more frequent, and more and more well-known to the general public.

The first and foremost property that one would like to enforce is *secrecy*, or *confidentiality*. You certainly would not like to be robbed by somebody who got hold of all the necessary information on your banking card; you would not like your health record to be public either; and you would not like your next (hopefully) big-selling software project to be known by your competitors in advance. This problem, ensuring that some given data are concealed to external, non-authorized people (or machines), is not new. Encryption has been used as a means of ensuring confidentiality in every armed forces around the world for ages. The new factor here is that computers and networks make it so easy to access any kind of information: in modern computer networks, reading data from your computer for an intruder may be just as easy as connecting a wire to an outlet on the wall.

A second property of interest is *authentication*. Maybe you'd like to communicate with trusted parties. But how can you be sure you're really talking to the right person? A long time ago, when you met face to face, it was easy enough to recognize whom you were talking to. Nowadays, computers talk through digital lines. Even payphones talk to smartcards (see [54] for an authentication attack on second-generation pay-phone cards), and mobile phones talk to servers and back, using encrypted channels. Each of these appliances need to check that they are really talking to the right appliance or computer. Otherwise you could spy on someone else's conversation on the phone, or you could intercept an encrypted email between two competitors, for example.

There are many other properties to be checked, in practice. *Denial of service* attacks do not steal valuable information from your hard disk (secrecy does not fail), they do not attempt at making you believe you're receiving an email from your old friend Joa (authentication), rather they just make your machine unusable: suddenly your machine freezes, reboots, your network is overloaded: you may be victim of a denial of service attack.

Another one is *fair exchange*: when you sign a contract over Internet—and you do, as soon as you buy a train ticket or the latest Harry Potter book on the Internet—, you would like to be sure that you agree to buy

and the reseller agrees to sell, or none of you agrees to the transaction, but that nothing else may happen. In particular, you would like to be sure that nobody can get a competitive advantage by first having the other agree to the transaction, then reporting the sales condition you obtained to a competitor, to eventually resign the transaction and make a deal with the competitor.

There would be many other properties that are worth considering. The goal of the SECSI project is, foremost, to design algorithms and tools to check such *security properties*. First, on abstract and idealized versions of what actually runs on your computer, banking card, or mobile phone: namely, on *cryptographic protocols*. This is important: one can cite dozens of published cryptographic protocols which nonetheless have been found faulty later on—the award certainly going to the Needham-Schroeder public-key protocol [79], which was believed to be correct for 17 years before an attack was found, and the protocol fixed, by G. Lowe [75].

Second, it would be desirable to check the same security properties on more and more concrete algorithms, until a level is reached where actual code can be analyzed. This is a technical challenge, involving the design of new static analysis techniques that mix reasoning on cryptographic protocols (only at a larger scale) and reasoning on pointers, functions, and other features of standard programming languages.

Third, once various more or less abstract versions of some piece of software have been proved correct, it may still be the case that some attacks remain. This may sound like a paradox, but look at it this way. When we reason on an abstract version of the given piece of software, we may have forgotten some important aspects of reality in the model. For instance, we may have modeled possible intruders on our system as being dishonest, all other participants being honest; but Lowe's attack on Needham-Schroeder's public-key protocol involves an intruder that is both honest *and* dishonest at the same time (in different sessions). It is all too easy to overlook the fact that anybody might be both good and evil. Another example is the fact that, to be able to say anything at all on a protocol, or some piece of code, simplifying assumptions have to be made. For example, a very convenient assumption until now was that of *perfect cryptography*, where the only way to get the plaintext from the ciphertext is to decipher the latter, using the right key. But many cryptographic primitives are not perfect, and A. Joux [72] has shown that Lowe's corrected version of the Needham-Schroeder public-key protocol was in fact flawed again, if you used the El Gamal encryption scheme to encrypt messages.

One of our efforts in the themes of cryptographic protocol verification, and also static code analysis to a lesser extent, is to take into account such weaknesses in the models, and repair them. This will provide us with more and more reliable security assessment tools.

However, there will always remain something that the models overlook. To take a last example, consider static analysis of code. When one analyzes actual programs, it is useful to simplify the semantics of the analyzed programming language, and e.g., assume that no pointer runs wild; otherwise, basically the analyzer must assume that anything may happen, and will more often than not that the analyzed program is probably vulnerable—even when it is not (in the given model, of course!). It is then fair to assume that some other means is used to ensure that no pointer indeed goes wild ([85] is a good start), and voila, we don't have to care about out-of-bounds access to arrays and records. In the present case, ignoring out-of-bounds accesses through pointers is precisely what makes the so-called *buffer-overflow attacks* so easy [64]. Let us say right away that the great majority of viruses, worms, and trojans propagate through such buffer-overflow vulnerabilities. It is therefore definitely relevant to monitor system activity *in real time* to detect and counter such attacks. The SECSI project team has had some preliminary success in doing so as part in 2003, using a new intrusion detection tool developed at LSV/SECSI and based on a novel approach to *on-line model-checking*: the attack is detected and reported in real-time, the sessions of the offender are killed and his account closed, in a jiffy. This is just an example of what can be achieved through intrusion detection, and this technology has already been applied to other system-level, and network-level security issues.

While SECSI is interested in many aspects of computer security, no cryptology per se is being done at SECSI. This is better left to cryptologists. SECSI does not guarantee either that your system can be made absolutely secure. After all, one of the most reliable source of unauthorized access to information is through *social engineering* (more or less subtle uses of the gullibility of people), against which science is impotent: see Mitnick and Simon's book [77].

To sum up, the focus of SECSI is on making small (PC) to large (mainframe) systems more secure, by checking once and for all (statically) security properties at a fairly abstract level, and going all the way to the concrete by monitoring (dynamically) security properties on actual computers and networks.

Scientifically, all themes are united by our reliance on rigorous approaches and logic: automated deduction, tree automata, abstract interpretation, model-checking.

3.2. Logic as a tool for assessing computer security

This section is unchanged from the SECSI 2003 report.

The various efforts of the SECSI team are united by the reliance on *logic* and rigorous methods. As already said in Section 3.1, SECSI does not do any cryptology per se.

As far as cryptographic protocol verification is concerned, one popular kind of model is that of Dolev and Yao (after [63], see [59] for a survey), where: the intruder can read and write on every communication channel, and in effect has full control over the network; the intruder may encrypt, decrypt, build and destruct pairs, as many times as it wishes; and, finally, cryptographic means are assumed to be *perfect*. The latter in particular means that the only way to compute the plaintext M from the ciphertext $\{M\}_K$ is to decrypt the latter using the inverse key K^{-1}. It also means that no ciphertext can be confused with any message that is not a ciphertext, and that $\{M\}_K = \{M'\}_{K'}$ implies $M = M'$ and $K = K'$. Thus, messages can be simply encoded as first-order terms, a fact which has been used by many authors.

This observation may be seen as the foundations for encoding cryptographic protocols in first-order logic [91][51]. Cryptographic protocols can also be analyzed using tree automata [58], as shown in [78][66], or using set constraints [57][49]. All these tools can be seen from an automated deduction perspective, as shown in [68] and [17]. Extensions to encryption primitives obeying algebraic laws are now being considered in the SECSI project, using deduction techniques modulo equational theories, as well as direct proof-theoretic techniques [60]. This is one of the themes of the RNTL project Prouvé.

Our work on intrusion detection also relies on logic. The crux of our method is a fast implementation of a fast algorithm for *on-line* model-checking of an application-specific temporal logic to *linear* Kripke models [83]. It also relies on specific *abstract interpretation* techniques to dramatically improve the speed of detection, by showing that certains threads waiting for specific sequences of events cannot succeed and therefore can be killed safely [70][67]. Of course, abstract interpretation is at the heart of our static analysis of C code project, too. In this framework, SECSI designs static analyses that generation sets of Horn clauses as constraints, which are then solved by automated deduction techniques... and this loops the loop.

Finally, it should be mentioned that SECSI also looks at alternative techniques. The most prominent is research conducted at LSV/SECSI on *logical relations* for λ-calculi enriched with primitives for fresh name creation, encryption and decryption, following Sumii and Pierce [87]. This is continuous work, started in [69] and pursued in [92]. The puzzling thing here is that the logical relations obtained there generalize the notion of *bisimulations* used in process algebra to a richer, higher-order framework.

4. Application Domains

4.1. Introduction

Keywords: *SSL*, *TLS*, *intrusion detection*, *mobile phones*, *secure distributed architectures*, *security*, *smart-cards*.

This section is unchanged from the SECSI 2003 report.

The application domains of SECSI cover a large part of computer security.

4.2. Cryptographic Protocols

Cryptographic protocols are used in more and more domains today, including smart card protocols, enterprise servers, railroad network architectures, secured distributed graphic user interfaces, mobile telephony,

on-line banking, on-line merchant sites, pay-per-view video, etc. The SECSI project is not tied to any specific domain as far as cryptographic protocols are concerned. Our industrial partners in this domain are Trusted Logic S.A., France Télécom R&D, and CRIL Technology.

4.3. Static Analysis

Analyzing cryptographic protocols per se is fine, but a more realistic approach consists in analyzing actual code implementing specific roles of cryptographic protocols, such as `ssh` or `slogin`, which implement the SSL/TLS protocols [89] are are used on every personal computer running Unix today. SSL and TLS are, more widely, used in every Web browser today: as soon as you connect to a secured server, you are running SSL or TLS. Being able to analyze actual C implementations of these or similar protocols is a concrete application we would like to be able to deal with in the long term.

4.4. Intrusion Detection

Making sure that cryptographic protocols are secure is not enough to guarantee that your system is secure. In all these domains, and in general in every domain where you need to set up a computer or a computer network, intrusion detection is needed. A new application domain for intrusion detection is smartcard security. While intrusion detection, and in particular the kind addressed in SECSI, used to be impractical on smartcards, the amount of available memory has soared on modern smartcards, making our intrusion detection techniques attractive on small devices: banking cards perhaps, SIM cards in GSM mobile phones certainly.

Standard application domains include securing enterprise-wide networks, and telephony servers. Our industrial partners in this domain today are France Télécom R&D and Calyx/NetSecure, a small company specialized in intrusion detection solutions.

A slightly less standard application of our intrusion detection techniques is tracking, where the intrusion detection system is not used to detect attacks, but to sort clients' activities per client type/user preferences (e.g., in GSM user tracking, as done by GSM operators), or to sort hardware and software failures according to client, hardware type or brand in remote maintenance applications.

5. Software

5.1. Software Packages and Prototypes

The SECSI project started in 2002 with a relatively large software basis: tools to parse, translate, and verify cryptographic protocols which are part of the RNTL project EVA (including CPV, CPV2, Securify), a static analysis tool in the course of being developed (CSur), an intrusion detection tool (logWeaver). These programs were started before SECSI was created.

The SPORE Web page was new in 2002. It is a public and open repository of cryptographic protocols. Its purpose is to collect information on cryptographic protocols, their design, proofs, attacks, at the international level.

2003 brought new developments. In intrusion detection, a completely new project has started, which benefited from the lessons learned in the DICO project (Section 7.1.1): faster, more versatile, the ORCHIDS intrusion detection system promises to become the most powerful intrusion detection system around.

In 2004, development of ORCHIDS continued. It is now a relatively mature prototype, which can and has been demonstrated in various occasions.

The CSur project was born in 2002. In 2003, some semantic problems delayed the completion of the first prototype. This prototype was finalized in 2004 by Fabrice Parrennes. Fabrice is now working at RATP. A paper on CSur was accepted at VMCAI'05 [71].

To support our work based on automated deduction and tree automata in cryptographic protocol verification and static analysis, two tools were created. The MOP modular platform, created in 2003, allows one to

experiment with new equational theories (and more), so as to enable testing new ideas quickly. This was tested on the verification of an actual protocol in 2004 [5].

And the H1 tool suite was created to support the discovery for security proofs, to output corresponding formal proofs in the Coq proof assistant, and also to provide a suite of tools allowing one to manipulate tree automata automatically [17].

5.2. SPORE: the Security Protocols Open Repository

Participants: Florent Jacquemard [in charge], Ralf Treinen, Hubert Comon-Lundh, (non-exclusive list).

This section is unchanged from the SECSI 2003 report.

SPORE is a publicly accessible Web page (http://www.lsv.ens-cachan.fr/spore/). Its purpose is to provide anybody who wishes so a public repository of cryptographic protocoles, their various versions, the security properties that they have been claimed to satisfy, those that they genuinely satisfy and under which assumptions, and the known attacks against these protocols.

A similar catalog had been published in 1997 by John Clark and Jeremy Jacob, in the second part of a widely distributed survey [56]. Notably, their catalog has often been used as a source of case studies for designers of automated cryptographic protocol verification tools.

The goal of the SPORE page is to continue Clark and Jacob's endeavor, first by updating the protocols in their survey, second by adding new entries. The whole repository is accessible on line, so as to cater for some interactivity with users and to promote its reusability by tool designers.

Each entry in the repository contains the description of one cryptographic protocol (in the semi-formal syntax of the reference paper [53], its claimed security properties, as well as comments and links to papers and pages about the protocol. References to works that propose formal proofs, or attacks, are given. The protocols of the repository can be downloaded in several formats, including printable ones (postscript, pdf) and text-only specifications of the sequence of messages defining the protocol.

The SPORE page contains about fifty protocols today. It was designed to be open: readers may comment on entries by email. More importantly, they may submit new protocols. However, it did not evolve in 2004.

5.3. The H1 Tool Suite: h1, pl2tptp, auto2pl, pldet, autodot, tptpmorph, linauto, h1trace, h1logstrip, h1mc, h1mon, h1getlog

Participant: Jean Goubault-Larrecq [in charge].

The initial purpose of the `h1` tool is to decide Nielson, Nielson and Seidl's class $\mathcal{H}_1$ [80], as well as an automated abstraction engine that converts any clause set to one in $\mathcal{H}_1$.

The main application of `h1` is to verify sets of clauses representing cryptographic protocols. The $\mathcal{H}_1$ class is decidable, and accordingly `h1` always terminates. In case a contradiction is found, the `h1` proof is an indication of a plausible attack on the input protocol. In case no contradiction is found, then the input protocol is secure.

In accordance with goals of the former RNTL EVA project, when no contradiction is found, `h1` is also able to produce a proof of security, in the form of an alternating tree automaton describing a finite model.

A novelty of 2004 is that, while `h1` itself formerly model-checked the input clause set against the alternating tree automaton, and generated a Coq proof script automatically, this task is now delegated to the external model-checker `h1mc`, which is a standalone model-checking tool. This allows for a more modular architecture, and possibly eases interfacing with other tools. This proof script can then be re-checked under the Coq proof assistant, so as to get a formal proof of security. This is the topic of [17].

Another new tool is the `h1trace` utility, which attempts to explain the proofs (or proof candidates) of contradiction, alternatively, the attacks found, in various more or less readable formats, from formal Coq proofs to tree-like, hierarchical descriptions of proofs that can be explored interactively using Emacs' outline mode.

A word of warning: `h1` is fast, but is currently limited to first-order reasoning, without taking into account any equational theory. For more versatility, but less speed, choose MOP [5].

The `h1` program is just the keystone of the H1 *tool suite*. Since `h1` can also be understood as a finite tree automaton workbench ($\mathcal{H}_1$ clause sets can be seen as fancy extensions to alternating, two-way tree automata, and always define regular tree languages), there was an opportunity to make `h1` the cornerstone of a more ambitious platform for handling rational tree languages.

While `h1` takes its input in TPTP format [88], `h1` outputs alternating tree automata in Prolog syntax, and—if asked so—the corresponding finite models (a.k.a., complete deterministic automata) in XML syntax. Various tools are provided in the H1 tool suite to convert between these formats: `pl2tptp` converts clause sets in Prolog notation to clause sets in TPTP format, while `auto2pl` converts XML deterministic tree automata to Prolog notation. The `auto2pl` utility also has an option to produce complements of automata; the combination of `h1` and `auto2pl` then allows one to decide sets of $\mathcal{H}_1$ clauses plus Prolog's stratified negation. (This negation is exactly automaton complementation.) The `pldet` utility determinizes alternating tree automata in Prolog notation, outputting deterministic tree automata in XML syntax.

The `autodot` utility converts deterministic tree automata in XML syntax to files in `dot`'s input format: `dot` is a publicly available graph layout engine. This allows one to visualize automata, at least small ones.

The `tptpmorph` utility computes the image of regular tree languages under term algebra homomorphisms. Additionally, `linauto` solves quantifier-free Presburger formulas, exploiting Comon and Boudet's efficient encoding of solutions to quantifier-free Presburger formulas into deterministic word automata [52]. Note that word automata are merely particular cases of tree automata. The `tptpmorph` utility can then in particular be used to apply projection morphisms, thus implementing existential quantification. That is, `linauto`, `tptpmorph`, and `auto2pl` together provide all needed tools to decide full Presburger arithmetic.

The H1 tool suite, consisting of all these utilities, is written in HimML (http://www.lsv.ens-cachan.fr/~goubault/himml-dwnld.html), a variant of the ML language with fast finite set and map operations, due to Jean Goubault-Larrecq. H1 consists of about 20 000 lines of HimML and C code.

H1 was mostly written in 2003. In 2004, additional optimizations were included, which make `h1` suitable for large sets of clauses such as those generated by CSur [71]. Also, the proof explanation (`h1trace`) and the model-checking (explanation of absence of proofs) facilities were separated from the main `h1` tool and improved. Additionally, helper utilities such as `h1logstrip` (purging proof files from irrelevant inferences), `h1mon` (monitoring utility), and `h1getlog` were added.

5.4. The CSur Static Analysis Tool

Keywords: *C*, *Horn clauses*, *Static analysis*, *cryptographic protocols*, *pointer analysis*.

Participants: Jean Goubault-Larrecq [in charge], Fabrice Parrennes, Julien Olivain.

CSur was started before SECSI was created, early 2002. It was developed by Fabrice Parrennes, once a postdoc on the ACI jeunes chercheurs "Sécurité informatique, protocoles cryptographiques et détection d'intrusions", attributed to Jean Goubault-Larrecq in 2001. Since September 01, 2003, Fabrice Parrennes has been part-time teaching assistant (1/2 ATER) at ENS Cachan. Fabrice Parrennes left to work for RATP early June 2004.

CSur is meant to detect cryptographic protocol-related vulnerabilities in C source code. It parses, analyzes C source files, then produces lists of Horn clauses that can be passed on to tools like SPASS [90] or H1 (Section 5.3) to detect plausible attacks, or better, to prove formally that the input C program is secure.

The basic idea is that Horn clauses can be used to represent both interaction with the network, in Dolev-Yao style, and to describe in-memory pointer aliasing. For example, calls to the `write`(2) primitive will trigger the generation of a clause of the form $\texttt{knows}(M) \Leftarrow \ldots$, where M is the message written to the network, and calls to the `read`(2) primitive triggers clauses $P(X) \Leftarrow \texttt{knows}(X)$, where `knows` is the predicate recognizing all messages known to a Dolev-Yao intruder, and which is axiomatized as in classical Dolev-Yao analysis of cryptographic protocols. On the other hand, points-to analyses [50][86] can also be recast as generating Horn clauses.

The challenge here is in the subtle interaction between the Dolev-Yao world and the C pointer world. CSur reached the status of functional prototype as of late 2003. The abstract semantics was reengineered in 2004: while the abstract semantics was correct, it was too inaccurate on some examples of cryptographic code.

This basics of this work are described in a forthcoming paper by Goubault-Larrecq and Parrennes [71].

5.5. The ORCHIDS Intrusion Detection Tool

Participants: Julien Olivain [in charge], Jean Goubault-Larrecq, Stéphane Demri.

ORCHIDS is an intrusion detection tool, capable of analyzing and correlating events over time, in real time. Its purpose is to detect, report, and take countermeasures against intruders in real time. The core of the engine is based on the algorithm in the second part of the paper by Muriel Roger and Jean Goubault-Larrecq [83]. The precise algorithm is described in two reports [70][67].

In 2004, the ORCHIDS prototype, started in 2003, was considerably enhanced, and reached a status of working prototype. It was deployed on the computer network of the LSV, to be developed and tested on a real-life network.

Most of these enhancements aimed at scaling ORCHIDS to as to be functional on real-life enterprise networks, combining performance and security. Amongst these enhancements, we emphasize:

Automata with cuts: The intrusion detection rules are compiled to non-deterministic automata with cuts and timeouts. The '*cut*' is similar the Prolog language keyword '!'. Its purpose is to reduce the scenario search space by skipping some equivalent event sequences that are known to be a part of an unique attack. Additionally, the '*shortest run*' search criterion makes use of this cut, by keeping only the first occurence of a set of possible attacks which are considered as equivalent. Additionnaly, the cut allow a certain form of negation (search for a sequence *without* seeing another).
A number of improvements related to cuts and shortest runs were the direct consequence of our interaction with the partners of the DICO project (Section 7.1.1).

Programmable input module: To be deployed easily and quickly on a real network, ORCHIDS supports a large number of common standards equipments and applications. This is provided by a module, programmable with a input language based on Posix regular expressions augmented with type conversion facilities. This module includes 40 common Unix programs definition, with 300 data fields.

SNMP traps receiver module: The ORCHIDS tool includes a SNMP-trap module receiver [84], [55]. SNMP stands for *Simple Network Management Protocol* and is widely used in network equipments and servers. A trap is a message sent by an equipment (an agent), in real-time, to a manager (a trap receiver) to inform it of an event. An Ethernet network switch can, for example, inform that a port has been activated, or when Ethernet address collision occurs. A server can inform that its hard disk drives are almost full, or that it is overloaded. A printer can inform that it is out of paper, etc. SNMP-traps are a good source of events for writing intrusion detection rules, because they allow an intruder detection system such as ORCHIDS to use network hardware equipments as intrusion detection sensors.

SNMP monitor tool: SNMP also allows one to access management information bases (*MIB*) that reflect the state of equipments (agents) at any time. The main difference between *MIB*s and *traps* is that traps are sent in real-time, while a *MIB* property must be queried explicitly. Additionally, SNMP can remotely set writable properties of agents. *SNMP Traps* events are usually predefined in the equipment (some useful and/or interesting events for intrusion detection are not reported).
Most advanced SNMP equipments (usually *manageable* switches or routers) have a *Remote MONitoring (RMON) MIB* that computes network statistics. RMON allows one to dynamically define *alarms* reported by SNMP Traps, when some specified integer values reach given thresholds.
The *SNMP monitor tool* designed for ORCHIDS generalizes this concept to any logical expression of any property types, and defines several kind of actions (sending a trap, sending a mail, write a syslog entry). The tool also computes on-line statistics from other values and reports it by SNMP.

Net-Entropy: an entropy checker for ciphered network layers: This tool performs a low-level checking of statistical entropy property of ciphered network connections. This tool effects a passive network capture, and can run in strategic points on a network (on routers or on an host connected on a monitoring port of a switch). It works on all cryptographic secured protocols: all SSH (Secure Shell) versions, all protocols secured by Secure Socket Layer (SSL) and Transport Layer Security (TLS) [62] (https, pops, imaps), IPSec/AH/ESP, and so on (http://www.lsv.ens-cachan.fr/~olivain/net-entropy/).

EvtGen: a generic discrete event simulator: This discrete event simulator was designed to generate random scenarios to experiment, profile and benchmark the core analyzer of the ORCHIDS intrusion detection system. It was extensively used in the DICO project to evaluate and compare algorithms and implementations of the tools done by the various partners.
EvtGen is based on Markov chains and includes some improvements to simulate as close as possible *real* computer system activity. The tool allows one to simulate characteristics like *scenario density* (the number of parallel instances of a scenario), *time density* (control of the time gaps between discrete events), *signalisation delays* (some communication channels may more or less randomly delay message delivery). EvtGen can simulate 'fake' events by generating a string by random substitutions and writing it into a file, sending it over the network, or by executing the string as a system command.
EvtGen includes a pseudo-random generator, so simulations can be replayed; this is useful for testing and comparing different implementations. The pseudo-random number generator includes the following random variable distributions: uniform, Gaussian, log-normal, exponential and Weibull.
Additionally, EvtGen can generate simulation data into files, to precompile huge data sets, and can work in real-time by using its network signalisation and its system command execution.
For more information, see the web page: http://www.lsv.ens-cachan.fr/~olivain/evtgen/).

In addition, ORCHIDS was applied to recent and real attacks, with some important results:

- ORCHIDS is able to detect several variations of the *Linux Kernel* `ptrace` system call attack with the same detection rule. Not only does the `ptrace` attack require the correlation of several events over time, but having just one rule to detect several variants of it debunks the myth that misuse detection tools such as ORCHIDS would have to be periodically upgraded to detect any new attack.
  The generic rule for the `ptrace` attack is also able to build a complementary report of malicious actions that the attacker might have done between the time of detection and the time of reaction (in a real attack, this short lapse of time corresponds to executing a *shellcode*).
  This starts making ORCHIDS an intrusion *prevention* system, i.e., a system that can not only detect but also analyze and counter attacks.
- ORCHIDS introduces a new kind of rule that detects a wide range of unknown attacks in a precise modeled context. This concept is illustrated with the `do_brk()` attack, which is one of the most malicious of the year 2004.
  This attack can gain (locally) a *root* access without generating *any* suspicious activity. Only a permission violation can be observed, if system calls are correlated.
  Detecting it using ORCHIDS is done by describing how programs inherit permissions and rights on *Unix* operating systems. This rule can detect all kernel-level permission violation. (`do_brk` attack, kernel backdoors, etc...) This strategy is well-known in the literature as run-time monitoring. This shows that the ORCHIDS engine also contains this particular form of detection.

- Finally, another result shows how ORCHIDS detects network attacks, by helping the main core analyzer with simple auxiliary analysis tools, such as Net-Entropy, described above.
  This was illustrated with the detection of an attack of the *Apache* web Server and its security module *mod_ssl*. This rule consists of the correlation of some warning about a cryptographic key exchange and abnormal statistical properties of ciphered data. This rule also block the attacker connexion by adding a specific rule on a firewall.

6. New Results

6.1. Verification of Cryptographic Protocols with Explicit Destructors

Participants: Stéphanie Delaune, Florent Jacquemard.

Delaune and Jacquemard proposes a decision procedure solving the problem of insecurity for cryptographic protocols containing explicit destructor symbols (such as decryption and projection) and equality tests, in the presence of a bounded number of sessions and of a passive or an active intruder. The destructor operators are axiomatized by an arbitrary convergent rewrite system satisfying some syntactic restrictions. This approach, with parameterized semantics, allows one to weaken the security hypotheses for verification, i.e., to address a larger class of attacks than for models based on free algebras.

The decision procedure is defined by an inference system for symbolic constraint solving based on basic narrowing techniques. It is polynomial in time in the case of a passive intruder and non-deterministic polynomial for an active intruder.

This result was published and presented at CCS 2004 [12].

6.2. Verification of Protocol Security against Dictionary Attacks

Participants: Mathieu Baudet, Stéphanie Delaune, Florent Jacquemard.

An interesting problem in the verification of cryptographic protocols occurs when some data, like poorly chosen passwords, can be guessed by dictionary attacks (with an off-line brute force enumeration of values in a dictionary, also called guessing attacks). Some countermeasures were proposed in the 1990's to improve protocol resistance to this kind of attack. More recently, some existing protocol verification tools have been modified in order to consider this problem.

However, until 2004, no formal definition and no decision results were known for dictionary attacks.

Stépanie Delaune and Florent Jacquemard have proposed an inference system modeling the deduction capabilities of an intruder who is able mount dictionary attacks. This system extends a set of well-studied deduction rules for symmetric and public key encryption often called Dolev-Yao rules. It is shown in [13] that the protocol insecurity problem in this extended model, and for a bounded number of sessions, is decidable in PTIME for a passive intruder and is NP-complete for an active intruder. In the passive case, the proof is based on a locality lemma for the extended inference system. In the active case, a procedure is given for simultaneously solving symbolic constraints with variables which represent intruder deductions.

A long version of this work has been submitted for a Special Issue of the Journal of Automated Reasoning (JAR) on *Automated Reasoning for Security Protocol Analysis*. Compared to [13], the long version provides a more general definition of the verification problem and the intruder deduction model has been augmented to deal with a probabilistic encryption operator. Indeed, probabilistic encryption (with which encrypting twice the same plaintext with the same key returns distinct ciphertexts) appears to be particularly relevant in the setting of dictionary attacks, e.g., for electronic vote protocols.

Independently, Corin *et al.* [61] have proposed a general definition for off-line guessing attacks, based on the notion of static equivalence from the applied pi-calculus [48]. This new definition is very natural and general (it applies to any set of cryptographic primitives). However, no corresponding decision results were proposed in [61].

Mathieu Baudet [23] showed that for a set of primitives modeled by a subterm confluent rewriting system, the existence of such off-line guessing attacks is decidable for a bounded number of sessions. The decision procedure relies on a practical algorithm for deciding the equivalence between a class of second-order E-unification problems. The overall work can be seen as an extension both of the results of Delaune and Jacquemard [12] (see Section 6.1, adding guessing attacks), and Abadi and Cortier [47] (adding active adversaries).

6.3. Intruder Deduction for AC-like Equational Theories with Homomorphisms

Participants: Pascal Lafourcade, Denis Lugiez [Université Aix-Marseille 1; on sabbatical leave at LSV until January 2004], Ralf Treinen.

Pascal Lafourcade, Denis Lugiez, and Ralf Treinen have investigated the intruder deduction problem, that is the existence of passive attacks, in presence of several variants of *AC*-like axioms (from *AC* to Abelian groups, including the theory of *exclusive or*) and homomorphism that are the most frequent axioms arising in cryptographic protocols. Solutions to this problem have previously been known for the cases of *exclusive or*, of Abelian groups, and of homomorphism alone. This work addresses the combination of these *AC*-like theories with the law of homomorphism which leads to much more complex decision problems.

They have shown decidability of the intruder deduction problem in all cases considered. The decision procedure is in EXPTIME, except for a restricted case in which they have been able to get a PTIME decision procedure using a property of one-counter and pushdown automata.

This result was published as a technical report [35]; a short version is submitted for publication.

6.4. Logical Relations for Name Creation and Encryption

Participants: David Nowak, Yu Zhang, Jean Goubault-Larrecq.

Pitts and Stark's nu-calculus [82] is a typed lambda-calculus which forms a basis for the study of interaction between higher-order functions and dynamically created names. A similar approach has received renewed attention recently through Sumii and Pierce's cryptographic lambda-calculus [87], an original approach to security protocols.

Logical relations are a powerful tool to prove properties of such a calculus, notably observational equivalence. While works done in previous years, by David Nowak and Yu Zhang, consisted in proposing Kripke logical relation alternatives to Pitts and Stark's logical relation, work started in 2003, and pursued actively in 2004, concentrated on two aspects:

- Extending these logical relations to richer lambda-calculi including general encryption and decryption mechanisms. This provides simpler and more uniform descriptions of logical relations for cryptographic lambda-calculi than the rather intricate relations of Sumii and Pierce. This is the main subject of [18].
- Showing that these logical relations are complete for behavioral equivalence. In other words, showing that two processes are related by some logical relation as above if and only if they are behaviorally equivalent. (Behavioral equivalence is, just as in the spi-calculus, the basis for formal definitions of secrecy and authentication, at least.) This required the passage to lax logical relations in [18]. These completeness results were then refined for computational lambda-calculi equipped with other monads than the name creation monad [74].

6.5. Analysis of Multi-party Contract Signing

Participant: Steve Kremer.

This result found in 2003, was first published in 2004: using the model-checker MOCHA, two multi-party contract signing protocols have been analyzed. The main result is that the Garay-MacKenzie multi-party contract signing protocol, published in 1999 and believed correct since then, was shown to contain a fundamental flaw. Due to the high complexity of the protocol, a rigorous hand analysis seems impossible and tool support is required. Moreover, the flaw is not present when three signers participate in the protocol. At least four signers are needed to break the protocol. This partially explains that the protocol was believed correct during four years.

These results were published at CSFW 2004 [9]. An extended version has been submitted to the Special Issue of The Journal of Automated Reasoning on Automated Reasoning for Security Protocol Analysis. This extended version also contains new theoretical results on how to verify efficiently the properties of contract signing protocols. In particular, it shows that when the protocol guarantees the timeliness property, the fairness property, which is the most fundamental one, can be reduced to invariant checking.

This work was carried out before Steve Kremer joint the SECSI project on September 1st and is joined work with Rohit Chadha (University of Sussex, Brighton, UK) and Andre Scedrov (University of Pennsylvania, Philadelphia, USA).

6.6. Analysing the vulnerability of protocols to produce known-pair and chosen-text attacks

Participant: Steve Kremer.

This result shows that it is possible to automatically analyze the existence of known-pair and chosen-text attacks in protocols. As these attacks are at the level of blocks, the attacker is extended by special capabilities related to block chaining techniques. The analysis is automated using Blanchet's protocol verifier and illustrated on two well-known protocols, the Needham-Schroeder-Lowe public-key protocol as well as the Needham-Schroeder symmetric-key protocol. On the first protocol, it is shown how the special intruder capabilities related to chaining may compromise the secrecy of nonces and that chosen-ciphertext attacks are possible. Two modified versions of the protocol which strengthen its security are presented. Known-pair and chosen-plaintext attacks are also illustrated on the second protocol.

These results were first published at SecCo 2004 [19].

This work was carried out before Steve Kremer joint the SECSI project on September 1st. The work was realized while Steve Kremer visited the University of Birmingham and is joined work with Mark Ryan (University of Birmingham, UK).

6.7. Analysis of a Multi-Party Fair Exchange Protocol and Formal Proof of Correctness in the Strand Space model

Participant: Steve Kremer.

A multi-party fair exchange protocol is a cryptographic protocol allowing several parties to exchange commodities in such a way that everyone gives an item away if and only if it receives an item in return. A multi-party fair exchange protocol, originally proposed by Franklin and Tsudik, was subsequently shown to have flaws and was fixed by González and Markowitch. Here, new flaws in the "fixed" version of the protocol are identified. A corrected version of the protocol is proposed and a formal proof of correctness in the strand space model is provided. One of the tricky points is that the proof needs to deal with modular multiplication. Therefore the classical Dolev-Yao intruder needs to be extended, much in the same way as in Section 7.1.4.

These results have been accepted for publication at Financial Crypto 2005 [73].

This work was carried out before Steve Kremer joint the SECSI project on September 1st. The work was realized while Steve Kremer visited the University of Birmingham and is joined work with Aybek Mukhamedov and Eike Ritter (University of Birmingham, UK).

6.8. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus

Participant: Steve Kremer.

Electronic voting promises the possibility of a convenient, efficient and secure facility for recording and tallying votes in an election. Recently highlighted inadequacies of implemented systems have demonstrated the importance of formally verifying the underlying voting protocols. The applied pi calculus is a formalism for modelling such protocols, and allows verifying properties by using automatic tools, and to rely on manual proof techniques for cases that automatic tools are unable to handle. The applied pi calculus is used to model a known protocol for elections known as FOO 92, and three of its expected properties, namely fairness, eligibility, and privacy are formalized. Blanchet's tool ProVerif is used to prove that the first two properties are satisfied. In the case of the third property, ProVerif is unable to prove it directly, because its ability to prove observational equivalence between processes is not complete. A manual proof is provided for the required equivalence. Moreover, the proof emphasizes the need to divide the protocol into three phases in order for privacy to hold. Although the original description of the protocol describes three phases, no explanation for this choice is given and therefore the proof might increase the understanding of this property.

These results have been submitted for publication.

This work was started before Steve Kremer joint the SECSI project on September 1st. The work was initiated while Steve Kremer visited the University of Birmingham and is joined work with Mark Ryan (University of Birmingham, UK).

6.9. Analysis of a Public-Key Protocol with MGS

Participant: Florent Jacquemard.

Since the 1994 landmark demonstration by Adleman of the possibilities of DNA to solve a class of combinatorial problems, biocomputing has often be advocated to develop "chemically combinatorial problem solvers". Jean-Louis Giavitto, Olivier Michel (LAMI, CNRS and Université d'Évry) and Florent Jacquemard have used an approach belonging to the membrane computing [81] area to address the combinatorial problem of analysis of a public key protocol. This analysis is used to validate the protocol and exhibits, as expected, a well known logical attack. The novelty of the approach is to use multiset rewriting in a nest of membranes. The use of membranes enables to tighten the conditions for detecting an attack.

The approach, presented in [21][36] has been validated by developing a full implementation in the declarative language MGS [65] for several versions of the analysis.

A paper on this work will appear [76] as a chapter in a special issue of Springer's Natural Computing Series on applications of membrane computing (George Paun editor).

6.10. Automated Proof by Induction Driven by Constrained Tree Grammars

Participant: Florent Jacquemard.

Adel Bouhoula (École supérieure des communications de Tunis) and Florent Jacquemard have developed a new approach for mechanizing induction on complex data structures (like bags, sorted lists, trees, powerlists, ...) by adapting and generalizing works in tree automata with constraints. The key idea of their approach is to compute a tree grammar with constraints which describes the initial model of the given specification. This grammar is used as an *induction scheme* for the generation of subgoals during the proof. The procedure, presented in [7], is sound and refutationally complete even when the axioms for constructors are not left-linear, constrained, non-terminating. Moreover, it subsumes all test set induction approaches. A long version of [7] was submitted in July to the Journal of Automated Reasoning (JAR).

7. Other Grants and Activities

7.1. National Initiatives

7.1.1. RNTL project DICO (preexisting SECSI)

Participants: Jean Goubault-Larrecq, Julien Olivain, Stéphane Demri.

This exploratory project, funded by the national network for software technology (RNTL), groups NetSecureOne (formerly Calyx/NetSecure, Maisons-Alfort, and before that NetSecure Software, Neuilly; leader of the project), France Telecom R&D (Caen), the LSV (Cachan), the IRISA (Rennes), ONERA/DTIM (Toulouse), FERIA/IRIT (Toulouse), and the École Supérieure d'Électricité (Rennes). It was notified in december 2001, and ended in december 2003; the project leader is waiting for an answer from the ministry of research and technology to a request to prolong the project until March 2004, for technical reasons.

The title, DICO, means **Co**operative **I**ntrusion **D**etection. It is therefore, first, an intrusion detection project, mixing signature-based approaches such as the one currently implemented in the ORCHIDS tool (Section 5.5) at LSV, behavior-based approaches (e.g., Bayesian networks), and alert correlation. The last item is the keystone of the project, and consists in merging alerts, or infering alerts coming from various tools, even from different servers. ORCHIDS also counts as one of the tools providing alerts in the DICO architecture.

The DICO project was officially terminated in June 2004.

7.1.2. ACI cryptologie "PSI-Robuste" (preexisting SECSI)

Participants: Jean Goubault-Larrecq, Stéphane Demri, Fabrice Parrennes, Julien Olivain, Yu Zhang.

The ACI "PSI-Robuste" is a crystallization action at LSV, on the theme of protecting computer systems. More specifically, it consists in developing new intrusion detection approaches, new static code analysis techniques aiming at detecting possible vulnerabilities, and making both interact. This is an ACI (action concertée incitative) on the theme of cryptology of the ministry of research. It started in fall 2001, for three years, and ended in fall 2004.

7.1.3. ACI jeunes chercheurs "Sécurité informatique, protocoles cryptographiques et détection d'intrusions" (preexisting SECSI)

Participants: Jean Goubault-Larrecq, Fabrice Parrennes, Stéphane Demri, Yu Zhang.

This is an ACI on the "jeunes chercheurs" programme ("young researcher"), attributed to Jean Goubault-Larrecq. This provides him and his colleagues funding for three years, starting from fall 2001. This ended in fall 2004.

The themes of this ACI are essentially the same as those of SECSI: automated cryptographic protocol verification, intrusion detection mainly, with a gist of static analysis. Fabrice Parrennes was funded through this ACI until September 2003, starting from September 2002.

7.1.4. RNTL project Prouvé

Participants: Ralf Treinen [in charge], Hubert Comon-Lundh, Jean Goubault-Larrecq, Steve Kremer [since September 01], Florent Jacquemard, Stéphanie Delaune, Pascal Lafourcade.

The exploratory project "Prouvé", funded by the national network for software technology (RNTL), is a collaboration between CRIL Technology, France Télécom R&D (Lannion), the CASSIS project at LORIA, INRIA Lorraine (Nancy), LSV (Cachan), and Verimag (Grenoble). The notification of acceptance, dated November 2003, was received end of January 2004.

The Prouvé project (for "Protocoles cryptographiques: outils de vérification automatique", i.e., cryptographic protocols: automated verification tools) is based on the foundations layed by the EVA project, which ended late 2003. Significant progress can be reported for the three following tasks of the project:

1. One major goal of the project is to define a semantics of cryptographic protocols that would be independent of the particular security property under consideration, and to define a language of

security properties which would allow one to express all properties of interest, independently of the protocol studied.
A first version of the PROUVÉ language for the specification of cryptographic protocols is described in [45]. This language features a restricted imperative programming language for the specification of the *roles* which are to be executed by the protocol participants, and a specification language for *scenarios* which express how roles are instantiated. The language allows the user to specify equational axioms for the (pre-defined or user-defined) operators. Mutable state (like for instance the credit of an agent in a payment protocol) can be expressed by using imperative variables. A first version of a parser for the PROUVÉ language is available for evaluation by the project participants, and will be published under a free license as soon as the language design has stabilized.

2. Another goal of the project is to extend the known methods of protocol verification by weakening the so-called perfect cryptography assumption. In particular, it should be possible to verify cryptographic protocols while taking into consideration algebraic properties of cryptographic primitives (such as those of modular arithmetic, as frequently used in public key cryptography), and substitution of nonces by timestamps or counters.
   A survey of algebraic properties of cryptographic operators that are relevant for the security of cryptographic protocols, along with examples of protocols or attacks using these properties, was given in [28]. This report also gives an overview of the existing formal methods for analyzing cryptographic protocols. Decidability and complexity results for the Intruder Deduction Problem (security against passive attacks in the Dolev-Yao model) for equational theories combining important AC-like theories with laws of homomorphism were given in [35].
   A general technique for automatically proving security against passive or active attacks for a bounded number of sessions, and which applies to any set of equational axioms satisfying certain syntactic criteria, was presented in [12][14]. See Section 6.1.
   Another important aspect of weakening the perfect cryptography assumption consists in so-called dictionary attacks, that is the case of an attacker who can exploit a small search space to guess a secret. It has been shown in [13] that in this case the intruder deduction problem is PTIME-complete, and that security against active attacks is NP-complete for a bounded number of sessions. See Section 6.2.

3. The techniques to be developed in this project will be validated in case studies provided by one of the industrial partners of the project, France Télécom R&D. This work is carried out by a PhD student (Stéphanie Delaune, CIFRE grant) under the direction of Francis Klay (France Télécom R&D).
   A comparative study of different languages for the specification of cryptographic protocols, based on the case study of an electronic payment protocol provided by France Télécom, is presented in [38]. This case study was also a driving force behind the design of the PROUVÉ specification language.

7.1.5. ACI Sécurité "Rossignol"

Participants: Hubert Comon-Lundh [in charge], Pascal Lafourcade, Vincent Bernat, Stéphanie Delaune, Jean Goubault-Larrecq, Florent Jacquemard, Ralf Treinen.

The "Rossignol" project, submitted and accepted as an ACI sécurité informatique, started in december 2003. The partners of the projects are the LIF (Laboratoire d'Informatique Fondamentale de Marseille), the CoMeTe action of INRIA Futurs (Laboratoire d'Informatique de l'École Polytechnique, Saclay), the LSV (Cachan) and Verimag (Grenoble). All the participants at LSV are members of the SECSI project. This ACI funds in particular the PhD of Pascal Lafourcade, under the direction of Ralf Treinen and Denis Lugiez (LIF), on subjects of the project.

The goal of the project Rossignol is to create a framework for security protocol verification that takes into account the operational semantics of protocols, different theories of the intruder that defines the attackers capabilities, and the semantics of the intended decurity properties, which will be defined independently from

the description of the protocols using appropriate formalisms and logics. The project aims at fitting more closely to actual cryptographic protocol practice.

Some results came by this year:

- The finer integration of probabilistic aspects in cryptographic protocol implementations, and in the specification of properties such as anonymity, as proposed by C. Palamidessi, CoMeTe.
- Some breakthroughs on the formal relation between the computational and formal models of security. The former is used mostly by cryptologists, and consists in establishing security by showing that any complexity-bounded adversary will have only a negligible probabilistic advantage in running a protocol compared to trying to break the cryptographic primitives by brute force. The latter is the core of the cryptographic protocol verification methods used at SECSI, in particular. Two papers accepted at ESOP 2004, one by V. Cortier (CNRS, Nancy, formerly PhD student at LSV, SECSI), the other by R. Janvier, Y. Lakhnech, and L. Mazaré (Verimag, Grenoble), establish a strong relationship between the two models, extending considerably the pioneering work by Abadi and Rogaway.
- The study of the so called "intruder deduction problem" in presence of cryptographic operators following AC-like axioms (from Associativity and Commutativity to Abelian groups, including the theory of exclusive or) and homomorphism that are the most frequent axioms arising in cryptographic protocols (see Section 6.3).
- The definition of extended intruder models dealing with dictionary attacks (see Section 6.2) and protocols with explicit destructors (see Section 6.1).

7.2. Industrial Contracts

7.2.1. Evaluating the Security of the Aud Tool

Participants: Jean Goubault-Larrecq, Julien Olivain, Hubert Comon-Lundh.

The SECSI project participated in the evaluation of the security of the implementation of the Aud tool, a document authentication tool and architecture by a small Parisian firm, Aud'System. This was a contract with INRIA Transfert (June 2004).

The first phase consisted in rereading code and documentation, and report bugs or vulnerabilities. While we agreed to keep the results of this study confidential, whatever the outcome, it can be said that this first phase identified several points to be improved or corrected in Aud'System's code, while confirming that the global architecture was sane.

The second phase of the study will be conducted in 2005, directly with Aud'System, and will consist in providing formal models of the tool's interactions with the outside world, and assessing its security formally and automatically as much as can be (using the tools of Section 5.).

8. Dissemination

8.1. Teaching

Jean Goubault-Larrecq and Hubert Comon-Lundh gave a series of lectures on cryptographic protocol verification and automated deduction at the DEA "Programmation"; amount: 30 h. (TD equivalent), 15 h. each.

Jean Goubault-Larrecq and Hubert Comon-Lundh gave a series of lectures on computability and complexity, a module in the first term of the magistère STIC, ENS Cachan. Total amount: 30 h. (TD equivalent).

Jean Goubault-Larrecq gave the first half of the module "Programmation 1" in the first term of the magistère STIC, ENS Cachan. Total amount: 30 h. (TD equivalent).

Jean Goubault-Larrecq gave 5 lectures on automated deduction in course 2-5 of the "Mastère Parisien de Recherche en Informatique" (MPRI). This is a series of lectures in common with Claude Marché and Jean-Pierre Jouannaud. Total amount: 22.5 h. (TD equivalent).

Jean Goubault-Larrecq gave a course on automated deduction to ENS Cachan students of the magistère STIC, second term, second year. While the above lectures mostly focus on resolution, this one focused on tableaux methods, and modal and temporal logics. In collaboration with Alain Finkel. Total amount: 15 h. (TD equivalent).

Jean Goubault-Larrecq gave the course on logic and computer science ("logique et informatique"), second term of first year, common to the magistère STIC, ENS Cachan, and the magistère de mathématiques fondamentales et appliquées à l'informatique (MMFAI), ENS (rue d'Ulm). Amount: 36 h. (TD equivalent).

Florent Jacquemard gave the TDs (exercise sessions) of the above course on logic and computer science. Amount: 24 h. (TD equivalent).

Jean Goubault-Larrecq gave a course on static analysis and automated deduction, second term of second year, magistère STIC, ENS Cachan. Amount: 30. (TD equivalent).

Vincent Bernat gave the TDs (exercise sessions) of the above course on static analysis. Amount: 12 h. (TD equivalent).

Jean Goubault-Larrecq gave an introductory talk on "computer pirates: how they work, how to counter them" at the freshman conference (conférence de rentrée), ENS Cachan. Total amount: 1 h.

Jean Goubault-Larrecq gave an introduction to cryptography and cryptographic protocols to ENS Cachan economy students, May 2004. Total amount: 3 h.

Hubert Comon-Lundh gave a series of introductory lectures to formal logic, a module in the first term of the magistère STIC, ENS Cachan. Total amount: 30 h. (TD equivalent)

Hubert Comon-Lundh gave a course on formal language theory, second term of first year, magistère STIC, ENS Cachan. Amount: 22.5 h. (TD equivalent).

Steve Kremer gave an invited lecture on fair exchange protocols in Mark Ryan's course "Computer Security" at the University of Birmingham. Amount: 1h.

Stéphane Demri gave a series of lectures on the computational complexity of LTL variants for formal verification at the DEA "Algorithmique". Amount: 15 h.

Ralf Treinen gave a series of lectures end exercise sessions on Logic and Automata, a module in the second term of the magistère STIC, ENS Cachan. Total volume: 14 h. (TD equivalent).

Ralf Treinen gave lectures end exercise sessions on Logic, a module in the first term of the magistère STIC, ENS Cachan. Total volume: 23 h. (TD equivalent).

Ralf Treinen gave lectures end exercise sessions on Programming 2, a module in the first term of the magistère STIC, ENS Cachan. Total volume: 44 h. (TD equivalent).

Ralf Treinen gave lectures end exercise sessions on Complexity, a module in the first term of the magistère STIC, ENS Cachan. Total volume: 56 h. (TD equivalent).

Ralf Treinen supervised the installation of a new computer room for the Computer Science students of ENS Cachan, and which was inaugurated in September 2004.

Julien Olivain gave an introductory course on computer networks (amount: 6 h. TD equivalent), and gave exercise sessions (amount: 4 h.) to students of the magistère STIC, ENS Cachan.

As moniteur, Stéphanie Delaune gave TPs (programming exercise sessions) on JAVA to students of DEUG MIAS, University Paris 7. Amount 31h. TD equivalent.

Vincent Bernat gave a series of TPs (programming exercise sessions) of a course of C++ programming for students of the electronics department at ENS Cachan. Amount: 34 h. (TD equivalent).

Vincent Bernat gave a series of TPs (programming exercise sessions) of a course on network programming in C, first term of second year, magistère STIC, ENS Cachan. Amount: 32 h (TD equivalent).

Pascal Lafourcade gave TDs and TPs (exercise and programming exercise sessions) of the "Introduction to programming" course at DEUG MIAS, first year, University Paris 12, Créteil. Amount: 64 h. (TD equivalent).

Pascal Lafourcade gave TDs (exercise sessions) of the "Databases" course, first year, IUT of Fontainebleau. Amount: 32h. (TD equivalent).

Pascal Lafourcade gave TPs (programming exercise sessions) on system programming and networks, second year, IUT of Fontainebleau. Amount: 32h. (TD equivalent).

8.2. Scientific and Administrative Charges

Hubert Comon-Lundh and Jean Goubault-Larrecq are members of the scientific board of the Action Concertée Incitative (ACI) "Sécurité Informatique", and members of the bureau.

Hubert Comon-Lundh is in charge of the computer science teaching department at ENS Cachan.

Florent Jacquemard is member of the board (vice secretary) of the French Association for Information and Communication Systems (ASTI).

Florent Jacquemard is member of the board (treasurer) of the French Association for Theoretical Computer Science, French chapter of the European for Theoretical Computer Science (EATCS).

Ralf Treinen is member of the commission de spécialistes of University Lille 1, Section 27 (Computer Science), and of the commission de spécialistes of ENS de Cachan, Number 6 (Computer Science).

Steve Kremer represented the SECSI project at the kick-off meeting of the European Network of Excellence "Artist 2", October 2004.

Stéphane Demri is supplementary member of the commission de spécialistes, Number 6 of ENS de Cachan, Section 27.

8.3. Supervision, Advisorship

Jean Goubault-Larrecq supervised the following students:

- Yu Zhang (together with David Nowak), third-year PhD student, working on verification of cryptographic protocols, the cryptographic λ-calculus, and logical relations;
- Mathieu Baudet, second-year PhD student, working on cryptographic protocol verification, in particular off-line and on-line guessing attacks.
- Benjamin Ratti, first-year PhD student, working on extensions of tree automata to second-order situations, with applications to opacity properties in cryptographic protocols.

Hubert Comon-Lundh supervised the following students:

- Vincent Bernat, currently third-year PhD student, working on the automatic verification of cryptographic protocols, more specifically on proof normalization and decidability results for a bounded number of sessions.
- Stéphanie Delaune, currently second-year PhD student, working on the automatic verification of cryptographic protocols, more specifically on weakening the perfect cryptography assumption. This work is also supervised by Florent Jacquemard, together with Francis Klay (France Télécom R&D). This is part of the Prouvé project, and is a supported by a CIFRE grant with France Télécom.

Florent Jacquemard supervised Benjamin Ratti's PhD work in collaboration with Jean Goubault-Larrecq (see above).

Florent Jacquemard and Jean Goubault-Larrecq supervised Benjamin Ratti's DEA work, which led to his PhD, on second-order tree automata.

Florent Jacquemard and Jean Goubault-Larrecq supervised Ankit Gupta, a student of IIT Delhi in the INRIA summer internship program, May-July 2004. The subject was experimenting with finite-model-finding techniques to find security proofs of protocols, in particular modulo equational theories.

Ralf Treinen and Denis Lugiez (Marseilles, visiting LSV on sabbatical leave until February 2004) supervised Pascal Lafourcade, a second-year PhD student since October 01, 2003. The field is verification of cryptographic protocols. The subject of his thesis is the extension of the Dolev-Yao intruder model by algebraic properties of cryptographic primitives. His thesis is funded by a grant from the ACI Rossignol, see Section 7.1.5.

Stéphane Demri supervised the following students:

- Régis Gascon (together with Philippe Schnoebelen), currently first-year PhD student, working on the verification of qualitative and quantitative properties.
- Rémi Brochenin (together with David Nowak), first-year Master student co-habilited by ENS de Lyon on logical characterizations of ω-regular languages, June 2004.
- Clément Franchini (together with Philippe Schnoebelen), first-year Master student co-habilited by ENS de Lyon on accessibility problems for weighted graphs, June 2004.

David Nowak supervised Yu Zhang's PhD work in collaboration with Jean Goubault-Larrecq (see above), and Rémi Brochenin's work with Stéphane Demri (see above).

8.4. Participation to PhD or habilitation juries

Jean Goubault-Larrecq was reviewer (rapporteur) of Daniel Méry's PhD thesis in Nancy, France, Nov. 2004, and of Emmanuel Coquery's PhD thesis in Paris, France, Dec. 2004. He was examiner at Vincent Simonet's PhD thesis in Paris, France, March 2004.

Hubert Comon-Lundh was reviewer of Miquel Bofill's PhD thesis in Barcelona, Spain, July 2004.

8.5. Participation to conference program committees or journal editorial boards

Florent Jacquemard was member of the program committee of the 16th International Conference on Rewriting Techniques and Applications (RTA 2005) April 19–23, Nara, Japan.

Jean Goubault-Larrecq is vice-president of the steering committee of the automated theorem proving with tableaux and related methods conference, from September 2003, for three years.

Hubert Comon-Lundh was member of the program committee of the Conference on Computer Science Logic (CSL'04), Varsaw, Spetember 2004.

Ralf Treinen is member of the steering committee of the International Conference on Rewriting Techniques and Application (RTA), where he is publicity chair of RTA.

Ralf Treinen was member of the program committee of the Second International Joint Conference on Automated Reasoning (IJCAR 2004), July 4–8, 2004, Cork, Ireland.

Ralf Treinen was member of the program committee of the 15th International Conference on Rewriting Techniques and Applications (RTA 2004) June 3–5, 2004, Aachen, Germany.

Ralf Treinen was member of the program committee of the 11th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR), March 14-18, 2005, Montevideo, Uruguay.

Ralf Treinen was member of the organizing committee of the 18th International Workshop on Unification (UNIF'04), which was held as a satellite workshop of the International Joint Conference on Automated Reasoning in Cork, Ireland, July 4–8, 2004.

8.6. Participation to symposia, seminars, invitations

Jean Goubault-Larrecq gave an invited talk at JFLA'04, Sainte-Marie-de-Ré, France, in January 2004, on "Once you've found no proof, how do you convince a proof assistant?".

Jean Goubault-Larrecq gave an invited talk at the seminar of cryptography of the IRMAR and CELAR, Rennes, France, January 2004, on cryptographic protocols, code analysis, and intrusion detection—an overview of the scientific themes of the SECSI project.

Jean Goubault-Larrecq gave an invited talk at SASYFT'04, Orléans, France, June 2004: "on cryptographic protocols, regular tree languages, and automated deduction".

Jean Goubault-Larrecq gave an invited talk at the second Workshop on Logic for Pragmatics, Créteil, France, July 2004, on the geometry of intuitionistic modal S4 proofs.

Jean Goubault-Larrecq was invited to talk at Prof. H. Seidl's seminar at the Technische Universität München, March 2004, on the same topic.

Jean Goubault-Larrecq was invited to talk at the seminar of LIAFA, University Paris 7, Nov. 2004, to talk on intrusion detection and on-line model-checking.

Jean Goubault-Larrecq was invited at LFSM, université Laval, Québec City, Québec, for three weeks in July 2004 to work with Nadia Tawbi on static analysis and security on the one hand, with Josée Desharnais and François Laviolette on bisimulation for Markov processes on the other hand.

Hubert Comon-Lundh gave an invited conference at FOSSACS'04, the international conference on Foundations of Software Science and Computation Structures, Barcelona, Spain, April 2004. Title: "Intruder Theories".

Hubert Comon-Lundh gave a talk at the French-Taiwanese workshop, Ecole Polytechnique, April 2004.

Florent Jacquemard gave a talk at the 11th conference on Computer and Communication Security (CCS 2005), Washington DC, USA, September 2004 (accepted paper [12]).

Florent Jacquemard gave a talk at the 18th International Workshop on Unification (UNIF 2004), Cork, Ireland, July 2004 (accepted paper [14]).

Florent Jacquemard gave a talk at the 5th International Workshop on Strategies in Automated Deduction (STRATEGIES 2004), Cork, Ireland, July 2004 (accepted paper [7]).

Ralf Treinen was invited for a stay of three weeks at the Research Center for Verification and Semantics of Japan's National Institute of Advanced Industrial Science and Technology (AIST) at Amagasaki, Japan. During his stay he worked with Hitoshi Ohsaki on the semantics of specification languages for cryptographic protocols, and in particular on the translation of the language developed in the PROUVÉ project (see Section 7.1.4) to the ACTAS verification tool.

Steve Kremer gave a talk on "Formal Analysis of Multi-Party Contract Signing" at the Workshop on Issues in the Theory of Security (WITS'04), Barcelona, Spain, April 2004 (accepted paper, [10]).

Steve Kremer gave a talk on "Juggling with Pattern Matching" at the 3rd International Conference on Fun with Algorithms (FUN'04), Isola d'Elba, Italy, May 2004 (accepted paper, [8]).

Steve Kremer gave a talk on "Formal Analysis of Optimistic Fair Exchange Protocols" at the seminar of LIAFA, University Paris 7, May 2004.

Steve Kremer gave a talk on "Formal Analysis of Optimistic Fair Exchange Protocols" at the seminar of the Computer Science Department of Sussex University, Brighton, UK, June 2004.

Steve Kremer attended the 17th IEEE Computer Security Foundations Workshop (CSFW'04), Asilomar, Pacific Grove, California, USA, June 2004 (accepted paper, [9]).

Steve Kremer gave a talk on "Analysing the Vulnerability of Protocols to Produce Known-Pair and Chosen-Text Attacks" at the 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems (SecCo'04), London, UK, August 2004 (accepted paper, [19]).

Steve Kremer spent 6 months at the University of Birmingham (02/2004–07/2004). He has been working with Mark Ryan on the vulnerability of protocols to known-pair and chosen pair attacks and on the analysis of electronic voting protocols. He has also been working with Aybek Mukhamedov and Eike Ritter on the analysis of a multi-party fair exchange protocol.

Stéphane Demri was invited to talk at the seminar of "Polish Association for Logic and Philosophy of Science", Warsaw, April 2004, on "Linear-Time Temporal Logics over Presburger Constraints".

Stéphane Demri gave a talk at FOSSACS'04 (accepted paper, [16]), Barcelona, Spain, April 2004.

Stéphane Demri was invited to talk at the meeting of the Action Spécifique "Automates, modèles distribués et temporisés" of the RTP 23 of the CNRS, Cachan, January 2004 on "Deciding regular grammar logics with converse through GF2".

Stéphanie Delaune gave a talk at CSFW'04 in California, USA, in June 2004 (accepted paper [13]) and at JDIR'04 in Lannion, France, in November 2004 (accepted paper [15]). She also participated in the poster session of the 32nd Spring School in Theoretical Computer Science in Luminy, France, in May 2004.

Stéphanie Delaune participated in the ETAPS'04 conference in Barcelona, Spain, in March 2004, and in the CCS'04 conference, in October 2004 (accepted paper [12]).

Stéphanie Delaune went to Lannion, France several times (a few days each time) to work with Francis Klay, France Télécom R&D. She gave a talk at the France Télécom R&D site of Caen, France, in January 2004,

to present her work on dictionary attacks and to discuss with Marc Girault (Senior Expert for Security at France Telecom R&D) about an e-commerce protocol recently developed by his team. She also gave a talk on protocols verification in presence of explicit destructors at the annual security area meeting of France Télécom R&D, site of Issy-les-Moulineaux, France, in December 2004.

Stéphanie Delaune gave an invited talk on Verification of Cryptographic Protocols in Presence of Algebraic Properties at the seminar of the GREYC and LMNO laboratories of the University of Caen, France, in December 2004.

Vincent Bernat gave a talk on steganography at the EEA department (electrical engineering), ENS Cachan, 2004.

Yu Zhang gave a talk at the International Workshop on Formal Methods and Security on "Merging Encryption into Kripke Logical Relations of Dynamic Name Creation", Nanjing, China, May 17-20, 2004.

Yu Zhang gave a talk at CSL'04 in Karpacz, Poland, September 20-24, 2004 (accepted paper, together with Jean Goubault-Larrecq, Sławomir Lasota and David Nowak, [18]).

Mathieu Baudet gave a talk at the 1st Workshop on Security Analysis of Systems: Formalism and Tools (SASYFT-2004), Orléans, France, June 2004 (accepted paper: [6]).

Mathieu Baudet gave an invited talk at the seminar of cryptography of the IRMAR and CELAR, Rennes, France, September 2004, on "Relating formal and computational models for the security of protocols".

Mathieu Baudet gave an invited talk at the seminar of the LORIA, Nancy, France, November 2004, on the same topic.

Pascal Lafourcade gave a talk at RNTL Prouvé project meeting, Nancy, May 2004, on "Locality with Xor and Homomorphism".

Pascal Lafourcade participated in the ICCL Summer School on Proof Theory and Automated Theorem Proving and PCC Workshop, Dresden, Germany, June 2004.

8.7. Miscellaneous

Ralf Treinen maintains, together with Nachum Dershowitz (Tel Aviv University, Israel), the list of open problems of the conference series Rewriting Techniques and Applications (RTA). The list contains currently 101 problems, 27 of which are solved. The list is online at the address http://www.lsv.ens-cachan.fr/rtaloop/.

Ralf Treinen moderates the mailing list Constraints in Computational Logics, which was created in the Esprit working group of the same name, and which continues to operate after the end of the working group. The mailing list currently has 114 subscribers in the field of computational logics and mainly carries announcements of interest to the community. Further information about the mailing list, including an archive of past messages, is available at http://www.lsv.ens-cachan.fr/ccl/.

Ralf Treinen maintains the home page of the International Workshop on Unification (UNIF), which provides detailed information about the past events in UNIF's 18-years history. The UNIF home page is available at http://www.lsv.ens-cachan.fr/unif/.

Yu ZHANG maintains the home page of International Workshop on Formal Methods and Security (IWFMS), which is available at http://www.lsv.ens-cachan.fr/~zhang/workshop/.

9. Bibliography

Articles in referred journals and book chapters

[1] H. COMON-LUNDH, V. CORTIER. *Security Properties: Two Agents Are Sufficient*, in "Science of Computer Programming", vol. 50, n° 1–3, 2004, p. 51–71, http://www.lsv.ens-cachan.fr/Publis/PAPERS/ComonCortier-step2.ps.

[2] H. COMON-LUNDH, V. CORTIER. *Tree Automata with One Memory, Set Constraints and Cryptographic Protocols*, in "Theoretical Computer Science", To appear, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/ComonCortierTCS1.ps.

[3] H. COMON-LUNDH, Y. JURSKI. *Counter Automata, Fixed Points and Additive Theories*, in "Theoretical Computer Science", To appear, 2004.

[4] J. GOUBAULT-LARRECQ. *Extensions of Valuations*, in "Math. Struct. in Comp. Science", To appear, 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-17.rr.ps.

[5] J. GOUBAULT-LARRECQ, M. ROGER, K. N. VERMA. *Abstraction and Resolution Modulo AC: How to Verify Diffie-Hellman-like Protocols Automatically*, in "Journal of Logic and Algebraic Programming", To appear, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/GLRV-acm.ps.

Publications in Conferences and Workshops

[6] M. BAUDET. *Random Polynomial-Time Attacks and Dolev-Yao Models*, in "Proc. Workshop on Security Analysis of Systems: Formalism and Tools (SASYFT-2004), Orléans, France, June 2004", S. ANANTHARAMAN (editor)., Proceedings published as LIFO Technical Report 2004-11, Laboratoire d'Informatique Fondamentale d'Orléans, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/B04sasyft.ps.

[7] A. BOUHOULA, F. JACQUEMARD. *Constrained Tree Grammars to Pilot Automated Proof by Induction*, in "Proc. 5th Workshop on Strategies in Automated Deduction (STRATEGIES 2004), Cork, Ireland, July 2004", T. BOY DE LA TOUR (editor)., 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/BJ-strategies04.pdf.

[8] J. CARDINAL, S. KREMER, S. LANGERMAN. *Juggling with Pattern Matching*, in "Proceedings of the 3rd International Conference on Fun with Algorithms (FUN'04), Isola d'Elba, Italy", P. FERRAGINA, R. GROSSI (editors)., Edizioni Plus, Università di Pisa, May 2004, p. 147-158, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-fun04.ps.gz.

[9] R. CHADHA, S. KREMER, A. SCEDROV. *Formal Analysis of Multi-Party Contract Signing*, in "Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04), Asilomar, Pacific Grove, California, USA", IEEE Computer Society Press, June 2004, p. 266-279, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-csfw04.ps.gz.

[10] R. CHADHA, S. KREMER, A. SCEDROV. *Formal Analysis of Multi-Party Contract Signing*, in "Proceedings of the 4th IFIP WG1.7 Workshop on Issues in the Theory of Security (WITS'04), Barcelona, Spain", April 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-wits04.ps.gz.

[11] H. COMON-LUNDH. *Intruder Theories (Ongoing Work)*, in "Proc. 7th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS 2004), Barcelona, Spain, Apr. 2004", Lecture Notes in Computer Science, vol. 2987, Springer, 2004, p. 1–4.

[12] S. DELAUNE, F. JACQUEMARD. *A Decision Procedure for the Verification of Security Protocols with Explicit Destructors*, in "Proc. 11th ACM Conf. on Computer and Communications Security (CCS 2004), Washington, DC, USA, Oct. 2004", To appear, ACM Press, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DJ-ccs-2004.ps.

[13] S. DELAUNE, F. JACQUEMARD. *A Theory of Dictionary Attacks and its Complexity*, in "Proc. 17th IEEE Computer Security Foundations Workshop (CSFW 2004), Asilomar, CA, USA, June 2004", IEEE Comp. Soc. Press, 2004, p. 2–15, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DJ-csfw2004.ps.

[14] S. DELAUNE, F. JACQUEMARD. *Narrowing-Based Constraint Solving for the Verification of Security Protocols*, in "Proc. 18th Int. Workshop on Unification (UNIF 2004), Cork, Ireland, July 2004", M. KOHLHASE (editor)., 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-8.rr.ps.

[15] S. DELAUNE, F. KLAY. *Vérification automatique appliquée à un protocole de commerce électronique*, in "Actes 6ième Journées Doctorales Informatique et Réseau (JDIR 2004), Lannion, France, Nov. 2004", To appear, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DK-jdir-2004.pdf.

[16] S. DEMRI. *LTL over Integer Periodicity Constraints*, in "Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'04), Barcelona, Spain", I. WALUKIEWICZ (editor)., Lecture Notes in Computer Science, vol. 2987, Springer, March 2004, p. 121-135, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-6.rr.ps.

[17] J. GOUBAULT-LARRECQ. *Une fois qu'on n'a pas trouvé de preuve, comment le faire comprendre à un assistant de preuve ?*, in "Actes 15èmes journées francophones sur les langages applicatifs (JFLA 2004), Sainte-Marie-de-Ré, France, Jan. 2004", INRIA, collection didactique, 2004, p. 1–40, http://www.lsv.ens-cachan.fr/Publis/PAPERS/JGL-JFLA2004.ps.

[18] J. GOUBAULT-LARRECQ, S. LASOTA, D. NOWAK, Y. ZHANG. *Complete Lax Logical Relations for Cryptographic Lambda-Calculi*, in "Proc. 18th Int. Workshop Computer Science Logic (CSL 2004), Karpacz, Poland, Sep. 2004", Lecture Notes in Computer Science, vol. 3210, Springer, 2004, p. 400–414, http://www.lsv.ens-cachan.fr/Publis/PAPERS/GLLNZ-csl04.ps.

[19] S. KREMER, M. D. RYAN. *Analysing the Vulnerability of Protocols to produce known-pair and chosen-text attacks*, in "Proceedings of the 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems (SecCo'04), London, UK", Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, August 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-secco04.ps.gz.

[20] O. MICHEL, F. JACQUEMARD. *An Analysis of the Needham-Schroeder Public-Key Protocol with MGS*, in "Proc. 5th Workshop on Membrane Computing (WMC 2004), Milano, Italy, June 2004", G. PĂUN (editor)., 2004, http://psystems.disco.unimib.it/procwmc5.html.

[21] O. MICHEL, F. JACQUEMARD. *An Analysis of the Needham-Schroeder Public-Key Protocol with MGS*, in "Proceedings of the 5th Workshop on Membrane Computing (WMC 2004), Milano, Italy", June 2004.

[22] S. SAEEDNIA, S. KREMER, O. MARKOWITCH. *An Efficient Strong Designated Verifier Signature Scheme*, in "Revised Papers of the 6th International Conference on Information Security and Cryptology (ICISC'03), Seoul, Korea", J. IN LIM, D. HOON LEE (editors)., Lecture Notes in Computer Science, vol. 2971, Springer, 2004, p. 40-54, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-icisc03.ps.gz.

Internal Reports

[23] M. BAUDET. *Deciding Security of Protocols Against Guessing Attacks (Ext. Version)*, http://www.lsv.ens-cachan.fr/~baudet/, Technical report, LSV Research Report, 2004.

[24] V. BERNAT. *First-Order Cyberlogic Hereditary Harrop Logic*, To appear, Technical report, Stanford Research Institute, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/Bernat-cyberlogic1.ps.

[25] A. BOUHOULA, F. JACQUEMARD. *Constrained tree grammars to pilot automated proof by induction*, 20 pages, Research Report, n° LSV-04-14, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-14.rr.ps.

[26] R. CHADHA, S. KREMER, A. SCEDROV. *Analysis of Multi-Party Contract Signing*, Technical report, n° 516, Université Libre de Bruxelles, Belgique, 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Kremer-RT516.ps.gz.

[27] H. COMON-LUNDH, S. DELAUNE. *The finite variant property: How to get rid of some algebraic properties*, 21 pages, Research Report, n° LSV-04-17, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-17.rr.ps.

[28] V. CORTIER, S. DELAUNE, P. LAFOURCADE. *A Survey of Algebraic Properties Used in Cryptographic Protocols*, 35 pages, Research Report, n° LSV-04-15, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-15.rr.ps.

[29] S. DELAUNE, F. JACQUEMARD. *A theory of guessing attacks and its complexity*, 25 pages, Research Report, n° LSV-04-1, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-1.rr.ps.

[30] S. DELAUNE, F. JACQUEMARD. *Narrowing-Based Constraint Solving for the Verification of Security Protocols*, 24 pages, Research Report, n° LSV-04-8, Laboratoire Spécification et Vérification, ENS Cachan, France, April 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-8.rr.ps.

[31] S. DEMRI. *LTL over Integer Periodicity Constraints*, 35 pages, Research Report, n° LSV-04-6, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-6.rr.ps.

[32] J. GOUBAULT-LARRECQ, S. LASOTA, D. NOWAK. *Logical Relations for Monadic Types*, 80 pages, Research Report, n° LSV-04-13, Lab. Specification and Verification, ENS de Cachan, Cachan, France, June 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-13.rr.ps.

[33] J. GOUBAULT-LARRECQ, S. LASOTA, D. NOWAK, Y. ZHANG. *Complete Lax Logical Relations for Cryptographic Lambda-Calculi*, 16 pages, Research Report, n° LSV-04-4, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-4.rr.ps.

[34] J. GOUBAULT-LARRECQ, M. ROGER, K. N. VERMA. *Abstraction and Resolution Modulo AC: How to Verify Diffie-Hellman-like Protocols Automatically*, 40 pages, Research Report, n° LSV-04-

7, Laboratoire Spécification et Vérification, ENS Cachan, France, March 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-7.rr.ps.

[35] P. LAFOURCADE, D. LUGIEZ, R. TREINEN. *Intruder Deduction for AC-like Equational Theories with Homomorphisms*, 69 pages, Research Report, n° LSV-04-16, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-16.rr.ps.

[36] O. MICHEL, F. JACQUEMARD, J.-L. GIAVITTO. *Three Variations on the Analysis of the Needham-Schroeder Public-Key Protocol with MGS*, 25 pages, Technical report, n° LaMI-98-2004, LaMI – UniversitÃ© d'Evry - CNRS, May 2004.

[37] K. N. VERMA, J. GOUBAULT-LARRECQ. *Karp-Miller Trees for a Branching Extension of VASS*, 21 pages, Research Report, n° LSV-04-3, Lab. Specification and Verification, ENS de Cachan, Cachan, France, January 2004, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2004-3.rr.ps.

Miscellaneous

[38] L. BOZGA, S. DELAUNE, F. KLAY, R. TREINEN. *Spécification du protocole de porte-monnaie électronique*, 12 pages, June 2004, http://www.lsv.ens-cachan.fr/prouve/prouve-rap1.ps.gz, Rapport Technique 1 du projet RNTL PROUVÉ.

[39] V. CORTIER, S. DELAUNE, P. LAFOURCADE. *A Survey of Algebraic Properties Used in Cryptographic Protocols*, 19 pages, June 2004, http://www.lsv.ens-cachan.fr/prouve/prouve-rap2.ps.gz, Rapport Technique 2 du projet RNTL PROUVÉ.

[40] J. GOUBAULT-LARRECQ. *On Cryptographic Protocols, Regular Tree Languages, and Automated Deduction*, June 2004, http://www.lsv.ens-cachan.fr/~goubault/talk_sasyft.pdf, Invited talk, Workshop on Security Analysis of Systems: Formalism and Tools (SASYFT-2004), Orléans, France.

[41] D. NOWAK. *Logical Relations for Monadic Types*, May 2004, Invited talk. Int. Workshop on Formal Methods and Security (IWFMS 2004), Nanjing, P. R. China.

[42] J. OLIVAIN. *EVTGEN v1.0: A Programmable Generic Generator of Event Sequences*, Written in C (about 5000 lines), July 2004.

[43] B. RATTI. *Automates d'arbre d'ordre deux*, 45 pages, Rapport de DEA, DEA Programmation, Paris, France, September 2004, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/BRatti-dea2004.ps.gz.

[44] R. TREINEN. *Notes de cours préliminaires: Cours de complexité, second semestre, première année, magistère STIC*, 2004.

[45] R. TREINEN. *The PROUVÉ specification language*, 10 pages, August 2004, http://www.lsv.ens-cachan.fr/prouve/prouve-rap3.ps.gz, Rapport Technique 3 du projet RNTL PROUVÉ.

[46] R. TREINEN. *RTALOOP: The RTA List of Open Problems*, Size as of July 2004: 100 problems, 90 pages, 432 references, 2004, Web site at www.lsv.ens-cachan.fr/rtaloop, started 1997.

Bibliography in notes

[47] M. ABADI, V. CORTIER. *Deciding Knowledge in Security Protocols under Equational Theories*, in "Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP)", Springer-Verlag LNCS 3142, 2004, p. 46–58.

[48] M. ABADI, C. FOURNET. *Mobile Values, New Names, and Secure Communications*, in "Proc. 28th Annual ACM Symposium on Principles of Programming Languages (POPL)", ACM, 2001, p. 104–115.

[49] R. AMADIO, W. CHARATONIK. *On Name Generation and Set-Based Analysis in the Dolev-Yao Model*, in "CONCUR'02", Springer-Verlag LNCS 2421, 2002, p. 499-514.

[50] L. O. ANDERSEN. *Program Analysis and Specialization for the C Programming Language*, Ph. D. Thesis, DIKU, University of Copenhagen, 1994, ftp://ftp.diku.dk/pub/diku/semantics/papers/D-203.dvi.Z.

[51] B. BLANCHET. *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules*, in "14th IEEE Computer Security Foundations Workshop (CSFW-14)", IEEE Computer Society Press, 2001, p. 82–96.

[52] A. BOUDET, H. COMON. *Diophantine Equations, Presburger Arithmetic and Finite Automata*, in "Colloquium on Trees in Algebra and Programming (CAAP'96)", H. KIRCHNER (editor)., Springer Verlag LNCS 1059, 1996, p. 30–43.

[53] M. BURROWS, M. ABADI, R. NEEDHAM. *A Logic of Authentication*, in "Proceedings of the Royal Society", vol. 426, n° 1871, December 1989, p. 233–271.

[54] BY <JBS>. *La carte à puce nouvelle génération T2G est hackable*, in "The Hackademy Journal", vol. 9, June 2003, p. 3–6.

[55] J. CASE, K. MCCLOGHRIE, M. ROSE, S. WALDBUSSER. *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*, January 1996.

[56] J. CLARK, J. JACOB. *A Survey of Authentication Protocol Literature: Version 1.0.*, 1997, http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz.

[57] H. COMON, V. CORTIER, J. MITCHELL. *Tree Automata with One Memory, Set Constraints and Ping-Pong Protocols*, in "Proc. 28th International Conference on Automata, Languages and Programming (ICALP)", Springer-Verlag LNCS 2076, 2001, p. 682–693.

[58] H. COMON, M. DAUCHET, R. GILLERON, F. JACQUEMARD, D. LUGIEZ, S. TISON, M. TOMMASI. *Tree Automata Techniques and Applications*, 1997, http://www.grappa.univ-lille3.fr/tata.

[59] H. COMON, V. SHMATIKOV. *Is it possible to decide whether a cryptographic protocol is secure or not ?*, in "Journal of Telecommunications and Information Technology, Special Issue on Models and Methods for Cryptographic Protocol Verification", J. GOUBAULT-LARRECQ (editor)., vol. 4, Instytut Łącsności (Institute of Telecommunications), Warsaw, Poland, December 2002, p. 3–13.

[60] H. COMON-LUNDH, R. TREINEN. *Easy Intruder Deductions*, in "Proc. Int. Symp. Verification (Theory & Practice). Celebrating Zohar Manna's 1000000 2 -th Birthday, Taormina, Italy", Springer Verlag LNCS 2772, June–July 2003.

[61] R. CORIN, J. DOUMEN, S. ETALLE. *Analysing Password Protocol Security Against Off-line Dictionary Attacks*, in "Proc. 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP)", Electronic Notes in Theoretic Computer Science, Elsevier, 2004.

[62] T. DIERKS, C. ALLEN. *RFC 2246: The TLS Protocol Version 1*, January 1999.

[63] D. DOLEV, A. C. YAO. *On the Security of Pubic Key Protocols*, in "IEEE Transactions on Information Theory", vol. IT-29, n° 2, March 1983, p. 198–208.

[64] O. GAY. *Exploitation avancée de buffer overflows*, http://www.k-otik.com/papers/Exploitation_Avancee_BOF.pdf, Technical report, Security and Cryptography Laboratory (LASEC), École Polytechnique Fédérale de Lausanne, June 2002.

[65] J.-L. GIAVITTO, O. MICHEL. *MGS: a Rule-Based Programming Language for Complex Objects and Collections*, in "Electronic Notes in Theoretical Computer Science", M. VAN DEN BRAND, R. VERMA (editors)., vol. 59, Elsevier Science Publishers, 2001.

[66] J. GOUBAULT-LARRECQ. *A Method for Automatic Cryptographic Protocol Verification (Extended Abstract)*, in "Proceedings of the Workshop on Formal Methods in Parallel Programming, Theory and Applications (FMPPTA'2000)", Lecture Notes in Computer Science LNCS 1800, Springer Verlag, 2000, p. 977–984.

[67] J. GOUBAULT-LARRECQ. *Un Algorithme pour l'Analyse de Logs*, 33 pages, Research Report, n° LSV-02-18, Lab. Specification and Verification, ENS de Cachan, Cachan, France, November 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-18.rr.ps.

[68] J. GOUBAULT-LARRECQ. *Vérification de protocoles cryptographiques : la logique à la rescousse !*, in "Actes du 1er workshop international sur la sécurité des communications sur Internet (SECI'02)", J. GOUBAULT-LARRECQ (editor)., INRIA, collection didactique, 2002, p. 119–152, http://www.lsv.ens-cachan.fr/~goubault/SECI-02/Final/JGL/jgl.ps.

[69] J. GOUBAULT-LARRECQ, S. LASOTA, D. NOWAK. *Logical Relations for Monadic Types*, in "Proc. 16th Int. Workshop Computer Science Logic (CSL'2002), Edinburgh, Scotland", Springer-Verlag LNCS 2471, September 2002, p. 553–568.

[70] J. GOUBAULT-LARRECQ, J.-P. POUZOL, S. DEMRI, L. MÉ, P. CARLE. *Langages de Détection d'Attaques par Signatures*, 30 pages, June 2002, Sous-projet 3, livrable 1 du projet RNTL DICO. Version 1.

[71] J. GOUBAULT-LARRECQ, F. PARRENNES. *Cryptographic Protocol Analysis on Real C Code*, in "Proceedings of the 6th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05), Paris, France", R. COUSOT (editor)., Lecture Notes in Computer Science, To appear, vol. 3385, Springer, January 2005, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/GouPar-VMCAI2005.pdf.

[72] A. JOUX. *Qu'est-ce que la sécurité d'un algorithme de chiffrement?*, Talk at the DCSSI-LogiCal-SECSI meeting, Rocquencourt, 18 September 2002.

[73] S. KREMER, A. MUKHAMEDOV, E. RITTER. *Analysis of a Multi-Party Fair Exchange Protocol and Formal Proof of Correctness in the Strand Space Model*, in "Proceedings of the 9th International Conference on Financial Cryptography and Data Security (FC'05), Roseau, The Commonwealth Of Dominica", Lecture Notes in Computer Science, To appear, Springer, February-March 2005.

[74] S. LASOTA, D. NOWAK, Y. ZHANG. *On the Completeness of Logical Relations for Monadic Types*, in "Proceedings the 7th International Conference on Typed Lambda Calculi and Application (TLCA'04), Nara, Japan", Lecture Notes in Computer Science, Submitted., Springer, April 2005.

[75] G. LOWE. *An Attack on the Needham-Schroeder Public-Key Authentication Protocol*, in "Information Processing Letters", vol. 56, n° 3, 1996, p. 131–133.

[76] O. MICHEL, F. JACQUEMARD. *Applications of Membrane Computing. Achievements and Perspectives*, To appear, chap. An Analysis of a Public-Key Protocol with Membranes, Springer, 2004.

[77] K. D. MITNICK, W. L. SIMON. *The Art of Deception: Controlling the Human Element of Security*, ISBN 0471237124, Wiley Publishing Company, October 2002.

[78] D. MONNIAUX. *Abstracting Cryptographic Protocols with Tree Automata*, in "6th International Static Analysis Symposium (SAS'99)", Springer-Verlag LNCS 1694, 1999, p. 149–163, http://www.dmi.ens.fr/~monniaux/biblio/Monniaux_SAS99.ps.gz.

[79] R. M. NEEDHAM, M. D. SCHROEDER. *Using Encryption for Authentication in Large Networks of Computers*, in "Communications of the ACM", vol. 21, n° 12, 1978, p. 993–999.

[80] F. NIELSON, H. R. NIELSON, H. SEIDL. *Normalizable Horn Clauses, Strongly Recognizable Relations and Spi*, in "9th Static Analysis Symposium (SAS)", Springer Verlag LNCS 2477, 2002.

[81] G. PAUN. *Membrane Computing. An Introduction*, Springer-Verlag, 2002.

[82] A. PITTS, I. STARK. *Observable Properties of Higher Order Functions that Dynamically Create Local Names, or: What's new?*, in "Mathematic Foundations of Computer Science (MFCS'93)", LNCS, n° 711, Springer-Verlag LNCS 711, 1993, p. 122–141, http://www.dcs.ed.ac.uk/~stark/publications/obsphown.html.

[83] M. ROGER, J. GOUBAULT-LARRECQ. *Log Auditing through Model Checking*, in "Proc. 14th IEEE Computer Security Foundations Workshop (CSFW'01), Cape Breton, Nova Scotia, Canada, June 2001", IEEE Comp. Soc. Press, 2001, p. 220–236, http://www.lsv.ens-cachan.fr/Publis/PAPERS/RogGou-csfw01.ps.

[84] M. SCHOFFSTALL, M. FEDOR, J. DAVIN, J. CASE. *A Simple Network Management Protocol (SNMP)*, May 1990.

[85] A. SIMON, A. KING. *Analyzing String Buffers in C*, in "Intl. Conf. on Algebraic Methods and Software Methodology (AMAST'2002)", 2002, p. 365–379.

[86] B. STEENSGARD. *Points-to Analysis in Almost Linear Time*, in "24th ACM SIGPLAN-SIGACT Symp. Principles of Programming Languages", January 1997, p. 32–41.

[87] E. SUMII, B. C. PIERCE. *Logical Relations for Encryption*, in "Proc. 14th Computer Security Foundations Workshop", 2001, p. 256–272.

[88] G. SUTCLIFFE, C. SUTTNER. *The TPTP Problem Library for Automated Theorem Proving*, 2001, http://www.cs.miami.edu/~tptp/.

[89] S. A. THOMAS. *SSL & TLS Essentials: Securing the Web*, ISBN 0471383546, Wiley, 2000.

[90] C. WEIDENBACH, U. BRAHM, T. HILLENBRAND, E. KEEN, C. THEOBALD, D. TOPIĆ. *SPASS Version 2.0*, in "Proceedings of the 18th International Conference on Automated Deduction", A. VORONKOV (editor)., Springer-Verlag LNAI 2392, 2002.

[91] C. WEIDENBACH. *Towards an Automatic Analysis of Security Protocols*, in "Proceedings of the 16th International Conference on Automated Deduction (CADE-16)", H. GANZINGER (editor)., Springer-Verlag LNAI 1632, 1999, p. 378–382.

[92] Y. ZHANG, D. NOWAK. *Logical Relations for Dynamic Name Creation*, in "Proc. 17th Intl. Workshop Computer Science Logic (CSL'2003)", Springer Verlag LNCS 2803, 2003, p. 575–588, http://www.lsv.ens-cachan.fr/Publis/PAPERS/ZN-csl2003.ps.