INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Cassis

Combining approaches for the security of infinite state systems

Lorraine

THEME SYM

Table of contents

1. Team

Head of project-team
Michaël Rusinowitch [Research Director (DR), INRIA-LORIA]

Vice-Head of project-team
Françoise Bellegarde [PR, Université Franche-Comté, LIFC, retired in 2006]

Administrative assistant
Sophie Drouot [Until September 30]
Emmanuelle Deschamps [From October 1st]

Staff members
Véronique Cortier [Research Associate (CR), CNRS-LORIA]
Silvio Ranise [Research Associate (CR), INRIA-LORIA]
Christophe Ringeissen [Research Associate (CR), INRIA-LORIA]
Mathieu Turuani [Research Associate (CR), INRIA-LORIA]

Faculty members (LORIA)
Laurent Vigneron [MC, Université Nancy 2]

Faculty members (Université Franche-Comté)
Fabrice Bouquet [MC, HdR]
Alain Giorgetti [MC]
Pierre-Cyrille Héam [MC]
Olga Kouchnarenko [PR]
Bruno Legeard [PR, until January 31]

Post-doctoral fellows
Nikolaï Kosmatov [RNTL PROUVÉ, LIFC, until August 31]
Bogdan Warinschi [ACI CRYPTO, LORIA]

Ph. D. Students
Yohan Boichut [INRIA, LIFC]
Thibaut Brocard [BDI-CNRS, LIFC, from October 1st]
Najah Chridi [MENRT, LORIA]
Jean-François Couchot [PRAG UFC, LIFC, until August 31]
Frédéric Dadeau [MENRT, LIFC, until September 30]
Stéphane Debricon [INTERREG, LIFC, from January 1st]
Heinrich Hoerdegen [MENRT, LORIA]
Abdessamad Imine [ATER, LORIA]
Vincent Pretre [INTERREG, LIFC]
Augusto Oliveira Viana da Silva [ALBAN, LORIA, until February 28]
Judson Santos Santiago [INRIA, LORIA]
Duc-Khanh Tran [MENRT, LORIA]
Eugen Zalinescu [MENRT, LORIA]

2. Overall Objectives

2.1. Background

Cassis is a joint project between *Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA - UMR 7503)* and *Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC - FRE 2661)*.

The objective of the project is to design and develop tools to verify the safety of systems with an infinite number of states. The analysis of such systems is based on a symbolic representation of sets of states in terms of formal languages or logical formulas. Safety is obtained via automatic proof, symbolic exploration of models or test generation. These validation methods are complementary. They rely on the study of accessibility problems and their reduction to constraint solving.

An originality of the project is its focus on infinite systems, parameterized or large scale, for which each technique taken separately shows its limits. This is the case for example of protocols operating on topologies of arbitrary size (ring networks), systems handling data structures of any size (sets), or whose control is infinite (automata communicating through an unbounded buffer). Ongoing or envisioned applications concern embedded software (e.g., smart cards, automotive controllers), cryptographic protocols (IKE, SET, TLS, Kerberos) designed to ensure trust in electronic transactions, and distributed systems.

The problem of validating or verifying reactive systems is crucial with respect to the increasing number of security-sensitive systems. The failure of these critical systems can have dramatic consequences since for instance they are embedded in vehicles components, or they control power stations or telecommunication networks. Beside obvious security issues the reliability of products whose destination is millions of end-users has a tremendous economical impact.

There are several approaches to system verification: automated deduction, reachability analysis or model-checking, and testing. These approaches have different advantages and drawbacks. Automated deduction can address practical verification however it remains complex to handle and requires a lot of expertise and guidance from the user. Model-checking is exhaustive but must face combinatorial explosion and becomes problematic with large-size or infinite systems. Testing is fundamental for validating requirements since it allows discovering many errors. However, it is almost never exhaustive and therefore only leads to partial solutions. Hence we believe that these approaches should not be considered as competing but complementary.

The goal of our project is to contribute to new combinations of these three verification techniques in a framework that would allow applying them in an industrial context. In particular we expect some breakthrough in the infinite-state verification domain by joint applications of deductive, model-checking and testing techniques.

2.2. Context

For verifying the security of infinite state systems we rely on

- Different ways to express the safety, reachability or liveness properties of systems, linear-time or branching-time logics, and the application of abstraction or abstract interpretation.
- Test generation techniques.
- The modeling of systems by encoding states as words, terms or trees and by representing infinite sets of states by languages. To each of these structures corresponds appropriate action families, such as transductions or rewritings.

Our goal is to apply these different approaches for ensuring the security of industrial systems by providing adequate methods and tools. In more details we aim at the following contributions (see continuous lines in Figure 1):

1. verification of abstract models derived from existing systems;
2. tests generation from the abstract model for validating the existing model;
3. cross-fertilization of the different validation techniques (deduction, model-checking, test) by taking advantage of the complementarity scopes and of their respective algorithmic contributions.

Let us mention that all these techniques comply with various development methodologies.

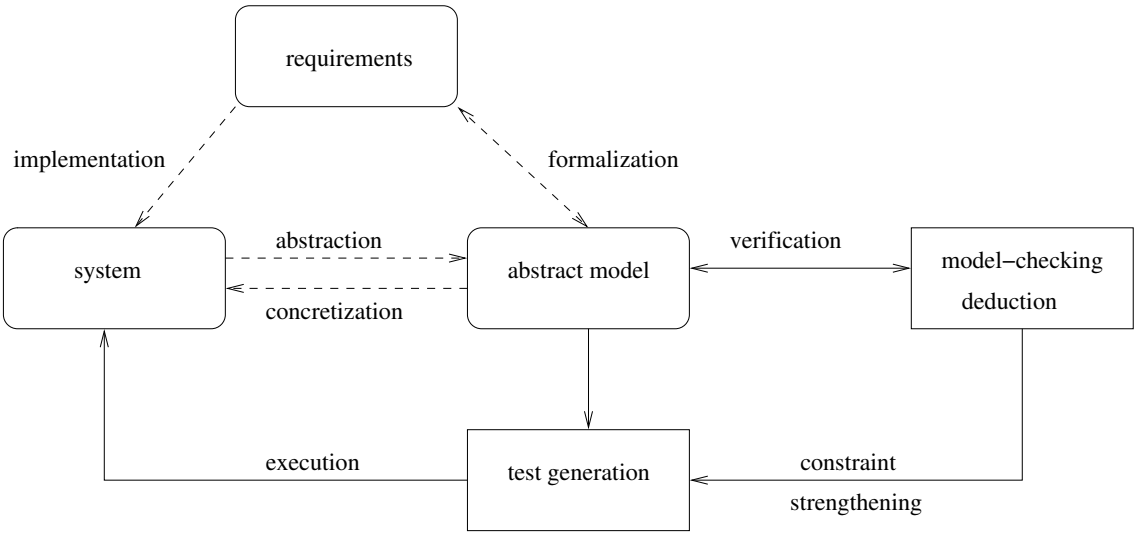

Figure 1. Software validation in Cassis

2.3. Challenge

Verifying the safety of infinite state systems is a challenge: nowadays algorithmic techniques only apply to very specific infinite state systems. On the other hand the deductive approaches are good candidates to capture infinite system safety verification but are difficult to bring into operation and require a deep expertise. A solution consists of integrating several verification methods by combining theorem-proving and model-checking for instance.

The behavior of infinite states systems is expressed in the various models by composing or iterating actions. One of the main problems with algorithmic techniques is to compute the effect of these actions on the initial state. This computation is called *reachability analysis*. The verification of safety properties as well as the automatic generation of test cases rely heavily on the accuracy of reachability analysis.

The transverse goal is to push away the limitations on the use of formal verification techniques, to ease their applications, and to let them scale-up.

1. For properties that can be checked by reachability analysis we have proposed models based on regular languages and rational transductions. We have completed them by designing algorithms for verifying a refinement relation between two models $\mathcal{S}$ and $\mathcal{T}$ [70]. This refinement relation when satisfied preserves the safety properties and therefore allows them to be inherited. We shall investigate this approach with other representations.
2. In order to generate boundary-value functional test cases, we abstract models as constrained states. These constraints are solved by a customized solver, called CLPS. The test cases are derived in two steps [6]:
    1. partitioning of the formal model and extraction of boundary values,
    2. reachability graph exploration from constrained states in order to reach boundary values and generate state sequences (trace) as test cases with the oracle.

    After the generation phase, a concretization is used to produce the test drivers [7]. Furthermore, the kernel of the engine allows one to perform specification animations in order to validate the model [73].

3. For the safety of infinite state systems we have designed automated deduction tools based on term rewriting (*SPIKE*, *daTac*, *haRVey*) and an extensible and modular platform for detecting flaws and potential attacks on security protocols (*AVISPA*). The tools have been built on the modeling of systems by terms and rewrite rules. Our works with other models based on regular languages of words or trees and of transducers should complement these term rewriting models.

In order to address this challenge, we rely on complementary skills within the project. We believe that each of the three techniques will benefit from concepts and algorithms designed for the two others.

3. Scientific Foundations

3.1. Introduction

Our main goal is to design techniques and to develop tools for the verification of (safety-critical) systems, such as programs or protocols. To this end, we develop a combination of techniques based on automated deduction for program verification, constraint resolution for test generation, and reachability analysis for the verification of infinite state systems.

3.2. Automated deduction

The main goal is to prove the validity of assertions obtained from program analysis. To this end, we develop techniques and automated deduction systems based on rewriting and constraint solving. The verification of recursive data structures relies on inductive reasoning or the manipulation of equations and it also exploits some form of reasoning modulo properties of selected operators (such as associativity and/or commutativity).

Rewriting, which allows us to simplify expressions and formulae, is a key ingredient for the effectiveness of many state-of-the-art automated reasoning systems. Furthermore, a well-founded rewriting relation can be also exploited to implement reasoning by induction. This observation forms the basis of our approach to inductive reasoning, with high degree of automation and the possibility to refute false conjectures.

The constraints are the key ingredient to postpone the activity of solving complex symbolic problems only when this is really necessary. They also allow us to increase the expressivity of the specification language and to refine theorem-proving strategies. As an example of this, the handling of constraints for unification problems or for the orientation of equalities in the presence of interpreted operators (e.g., commutativity and/or associativity function symbols) will possibly yield shorter automated proofs.

Finally, decision procedures are being considered as a key ingredient for the successful application of automated reasoning systems to verification problems. A decision procedure is an algorithm capable of efficiently deciding whether formulae from certain theories (such as Presburger arithmetic, lists, arrays, and their combination) are valid or not. We develop techniques to build and combine decision procedures for the domains which are relevant to verification problems. We also perform experimental evaluation of the proposed techniques by combining propositional reasoning (implemented by means of Boolean solvers – Binary Decision Diagrams or SAT solvers) and decision procedures, and their extensions to semi-decision procedures for handling larger (possibly undecidable) fragments of first-order logic.

3.3. Synthesizing and solving set constraints

Applying constraint logic programming technology in the validation and verification area is currently an active way of research. It usually requires the design of specific solvers to deal with the description language's vocabulary. We are interested in using a solver for set constraints based on the CLPS core [2], to evaluate set-oriented formal specifications. By evaluation, we mean the encoding of the formal model into a constraint system, and the ability for the solver to verify the invariant on the current constraint graph, to propagate preconditions or guards, and to apply the substitution calculus on this graph. The constraint solver is used for animating specifications and automatically generating abstract test cases.

3.4. Rewriting-based safety checking

Invariant checking and strenghtening is the dual of reachability analysis, and can thus be used for verifying safety properties of infinite-state systems. In fact, many infinite-state systems are just parameterized systems which become finite state systems when parameters are instantiated. Then, the challenge is to automatically discharge the maximal number of proof obligations coming from the decomposition of the invariance conditions. For parameterized systems, we develop a deductive approach where states are defined by first order formulae with equality, and proof obligations are checked by the automatic theorem prover *haRVey*. Thanks to this tool, we study the applicability of the superposition calculus (a modern version of resolution with a built-in treatment of the equality predicate and powerful techniques for reducing the search space) for deciding conditions arising from program verification.

4. Application Domains

4.1. Verification of security protocols

Security protocols such as SET, TLS and Kerberos, are designed for establishing the confidence of electronic transactions. They rely on cryptographic primitives, the purpose of which is to ensure integrity of data, authentication or anonymity of participants, confidentiality of transactions, etc.

The experience has shown that the design of those protocols is often erroneous, even when assuming that cryptographic primitives are perfect, i.e., that an encoded message cannot be decrypted without the appropriate key. An intruder can intercept, analyze and modify the exchanged messages with very few computations and therefore, for example, generate important economic damage.

Analyzing cryptographic protocols is complex because the set of configurations to consider is very large, and can even be *infinite*: one has to consider any number of sessions, any size of messages, sessions interleaving, algebraic properties of encryption or data structures.

Our objective is to automatize as much as possible the analysis of protocols starting from their specification. This consists in designing a tool easy to use, permitting to specify a large number of protocols thanks to a standard high-level language, and permitting either to look for flaws in a given protocol or to check whether it satisfies a given property. Such a tool is essential for verifying existing protocols, but also for helping in designing new ones. For our tool to be easy to use, it has to provide a graphical interface allowing a user to do only click-button.

Our tools for verifying security protocols are available as components of the AVISPA platform. As an extension of the AVISPA specification language, we are working on a new environment called *CASRUL* for handling more general protocols like e-business protocols for example.

4.2. Automated boundary testing from formal specifications

In [7], we have presented a new approach for test generation from set-oriented formal specifications: the BZ-TT method. This method is based on Constraint Logic Programming (CLP) techniques. The goal is to test every operation of the system at every boundary state using all input boundary values of that operation. It has been validated in several industry case studies for smart card OS and application validation (GSM 11-11 standard [71] and Java Card Virtual Machine Transaction mechanism [72]) and for embedded automotive software (an automobile wind-screen wiper controller).

This test generation method can be summed up as follows: from the formal model, the system computes boundary values to create boundary states; test cases are generated by traversal of the state space with a preamble part (sequences of operations from the initial state to a boundary state), a body part (critical invocations), an identification part (observation and Oracle state computation) and a post-amble part (return path to initial or boundary state). Then, an executable test script file is generated using a test pattern and a table of correspondence between abstract operations (from the model) and concrete ones. This approach differs on

several main points from the work of Dick, Faivre *et al*: first, using boundary goals as test objectives avoids the complete construction of the reachability graph; second, this process is fully automated and the test engineer could just drive it at the boundary value computation level or for the path computation.

The BZ-TT method is fully supported by the BZ-Testing-Tools tool-set. This environment is a set of tools dedicated to animation and test cases generation from B, Z or State-Chart formal specifications. It is based on the CLPS constraint solver, able to simulate the execution of the specification. By execution, we mean that the solver computes a so-called constrained state by applying the pre- and post-condition of operations. A constrained state is a constraint store where state variables and also input and output variables support constraints.

One orientation of the current work is to go beyond the finiteness assumption limitations by using symbolic constraint propagation during the test generation process and to extend the result to object oriented specifications.

4.3. Program debugging and verification

Catching bugs in programs is difficult and time-consuming. The effort of debugging and proving correct even small units of code can surpass the effort of programming. Bugs inserted while "programming in the small" can have dramatic consequences for the consistency of a whole software system as shown, e.g., by viruses which can spread by exploiting buffer overflows, a bug which typically arises while coding a small portion of code. To detect this kind of errors, many verification techniques have been put forward such as static analysis and software model checking.

Recently, in the program verification community, there seems to be a growing demand for more declarative approaches in order to make the results of the analysis readily available to the end user[1]. To meet this requirement, a growing number of program verification tools integrate some form of theorem proving.

The goals of our research are twofold. First, we perform theoretical investigations of various combinations of propositional and first-order satisfiability checking in order to automate the theorem proving activity required to solve a large class of program analysis problems which can be encoded as first-order formulae. Second, we experimentally investigate how our techniques behave on real problems so to make program analysis more precise and scalable. Building tools capable of providing a good balance between precision and scalability is one of the crucial challenges to transfer theorem proving technology to the industrial domains.

5. Software

5.1. Protocols verification tools

Keywords: *Cryptography*, *Security Protocols*, *Verification*.

Participants: Laurent Vigneron, Yohan Boichut, Pierre-Cyrille Héam, Olga Kouchnarenko, Nikolaï Kosmatov, Michaël Rusinowitch, Judson Santos Santiago, Mathieu Turuani.

5.1.1. AVISPA

Cassis has been one of the 4 partners involved in the European project AVISPA, which has resulted in the distribution of a tool for automated verification of security protocols, named *AVISPA* Tool. It is freely available on the web [2] and supported. The *AVISPA* Tool significantly extends its predecessor's scope, effectiveness, and performance, by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating 4 back-ends that implement automatic analysis techniques ranging from *protocol falsification* (by finding an attack on the input protocol) to *abstraction-based verification* methods for both finite and infinite numbers of sessions.

[1] See, for example, the challenge at http://research.microsoft.com/specncheck/consel_challenge.htm.
[2] http://www.avispa-project.org

In 2006, we have delivered a new release (v1.1): minor modifications in the specification language; more semantic verifications; back-ends, including *CL-AtSe* and *TA4SP* (see below), improved (algebraic properties better supported, better performances); addition of some contributions (documentation generator, XEmacs mode).

5.1.2. CASRUL

CASRUL is the subsystem of *AVISPA* that comprises the translator and the Cassis verification back-ends. In the context of RNTL project PROUVÉ, we have been working on the design of a different version of the translator. Its input specification language has been defined as an evolution of the AVISPA specification language, for considering more complex protocols such as electronic purses and electronic vote systems, that have been provided by France Telecom R&D. This language has been linked to different verification tools, including those developed in the Cassis group, *CL-AtSe* and *TA4SP* [33].

5.1.3. CL-AtSe

We develop *CL-AtSe*, a Constraint Logic based Attack Searcher for cryptographic protocols. The *CL-AtSe* approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see [75]) allows *CL-AtSe* to be correct and complete, i.e., any attack found by *CL-AtSe* is a valid attack, and if no attack is found, then the protocol is secure for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. *CL-AtSe* now includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing fairness, non-abuse freeness, etc..., advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators. In particular, *CL-AtSe* is now able to analyze protocols modulo the properties of XOR (exclusive or) or Exp (modular exponentiation). This has required to implement an optimized version of the combination algorithm of Baader & Schulz [69] for solving unification problems in disjoint unions of arbitrary theories.

In particular, *CL-AtSe* has been successfully used by Cassis members to analyse France Telecom R&D, Siemens AG, IETF, or Gemalto protocols in funded projects. It is also employed by external users, e.g., from the AVISPA's community. Moreover, *CL-AtSe* achieves very good analysis times, comparable and sometimes better than state-of-the art tools in the domain like OFMC (see [66] for tool details and precise benchmarks).

5.1.4. TA4SP

We have developed *TA4SP* (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions. This tool provides automatic computations of over and under approximations of the knowledge accessible by an intruder. This knowledge is encoded as a regular tree language and protocol steps and intruder abilities are encoded as a term rewriting system. Completions and tree automata computations are performed by Timbuk, a tool developed by project-team LANDE. When given a reachability problem such as secrecy, *TA4SP* reports that (1) *the protocol is safe* if it manages to compute an over-approximation of intruder's knowledge that does not contain a secret term or (2) *the protocol is unsafe* if it manages to compute an under-approximation of intruder's knowledge containing a secret term or (3) *I don't know* otherwise. *TA4SP* has verified 28 industrial protocols and case (3) occurred only once, for EKE protocol.

Several enhancements, such as an attack trace generator [31] and an algebraic properties module [32] are on the way.

As far as we know, two teams – the project-team LANDE at IRISA and the National Institute of Advanced Industrial Science and Technology in Japan – are working on the verification of security protocols using tree automata approximations. Both use tree automata dedicated tools, respectively Timbuk and Ceta-ACTAS, that can be freely downloaded on the web. However, these tools are not connected to any high level protocol specification language, and over-approximations are not fully automatically computed.

5.2. Testing tools

Keywords: *Animation of Specifications*, *CLP*, *Formal Specification*, *Test generation*.

Participants: Fabrice Bouquet, Frédéric Dadeau, Bruno Legeard.

The Testing Tools is a tool-set for animation and test generation from B, JML, Z and State-chart specifications. It consists of two components:

- **BZ-Testing-Tools** – BZ-TT – is a tool-set for animation and test generation from B, Z and State-chart specifications. BZ-TT provides several testing strategies (partition analysis, cause-effect testing, boundary-value testing and domain testing), and several test model coverage criteria (multiple condition coverage, boundary coverage and transition coverage).
- **JML-Testing-Tools** – JML-TT – is a framework for the symbolic animation of formal models written using JML annotations [76] embedded within Java programs. JML-TT provides a simple and efficient way to semi-automatically validate a JML specification and to check model properties such as class invariant or history constraints during the animation. This tool is used in the ACI GECCOO project[3].

We develop a third tool **Test-For-Testing-Tools** to valid the tests. The tool takes as input a code program and a test suite (realized by several approaches such as BZ-TT/random/properties driven tests). The system performs a mutation of the code program. We observe how many mutants are killed with each test suite.

5.3. Automated deduction tools: haRVey

Keywords: *Automated Deduction*, *Boolean Reasoning*, *Equational Reasoning*, *Satisfiability*, *Saturation Theorem Proving*.

Participants: Jean-François Couchot, Alain Giorgetti, Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran.

haRVey[4] is a theorem prover for first-order logic with equality [74]. It works by refutation and checks whether a first-order formula is a logical consequence of a first-order theory T, axiomatized by a finite set of formulae. Recently, the capability of reasoning in the combination of T and the theory of linear arithmetic over integers has been added. The main feature of *haRVey* is its capability of behaving as a decision procedure for the problem of checking the validity of certain classes of quantifier-free formulae modulo some theories of relevance in verification such as lists, arrays, and their combinations. The system features a combination of Boolean reasoning (supplied by a BDD or a SAT solver) to efficiently handle the boolean structure of formulae and a (generalization of the) Nelson-Oppen combination method between superposition theorem proving to flexibly reason in T and an implementation of Fourier-Motzkin method for linear arithmetic. The version of *haRVey* integrating a SAT solver has been designed and implemented by P. Fontaine (MOSEL project). *haRVey* has been especially designed to be integrated in larger verification systems. It is integrated in Barvey, a tool to check the consistency of B specifications. It takes a B abstract machine as input, generates proof obligations encoding the fact that the invariant is inductive, and translates them into a validity problem that *haRVey* can discharge. The tool *Why* developed by J.-C. Filliâtre (LRI, Université Paris Sud, Orsay) can generate proof obligations for *haRVey* to check the correctness of ML or C programs.

5.4. Others tools

Most of the software tools described in previous sections are using tools that we have developed in the past: BZ-TT uses the set constraints solver CLPS; the first version of *CASRUL* was using the theorem prover *daTac*; and *SPIKE*, our induction-based theorem prover, is used in the system VOTE in collaboration with the ECOO project.

[3] http://geccoo.lri.fr
[4] http://www.loria.fr/equipes/cassis/softwares/haRVey/

6. New Results

6.1. Automated deduction

Keywords: *Consistency*, *Decision Procedure*, *Proof*, *Satisfiability*, *Tree Automata*.

6.1.1. Decision procedures and their extensions

Participants: Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran.

We develop general techniques which allow us to re-use available tools in order to build a new generation of satisfiability solvers offering a good trade-off between expressiveness, flexibility, and scalability. Our original approach is based on the careful integration of rewriting techniques to design satisfiability procedures for a wide range of theories formalizing data structures, together with combination techniques to build satisfiability procedures for unions of theories in a modular way.

In [58], [57], we address the problems of combining rewriting-based satisfiability procedures and consider two combination scenarios: (i) the combination within the class of rewriting-based satisfiability procedures and (ii) the Nelson-Oppen combination of rewriting-based satisfiability procedures and arbitrary satisfiability procedures. For each scenario, we use in [57] meta-saturation, which schematizes saturation of the set containing the axioms of a given theory and an arbitrary set of ground literals, to syntactically decide sufficient conditions for the combinability of rewriting-based satisfiability procedures. For (i), we give a sufficient condition for the modular termination of meta-saturation. When meta-saturation for the union of theories halts, it yields a rewriting-based satisfiability procedure for the union. For (ii), we use meta-saturation to prove the stable infiniteness of the component theories and deduction completeness of their rewriting-based satisfiability procedures. These properties are important to establish the correctness of the Nelson-Oppen combination method and to obtain an efficient implementation.

In [34], we strengthen the Nelson-Oppen decidability result, by showing that it applies to theories over disjoint signatures, whose satisfiability problem, in either finite or infinite models, is decidable. Furthermore, this result covers rewriting-based satisfiability procedures.

We study in [50] extensions of the theory of arrays whose satisfiability problem (i.e., checking the satisfiability of conjunctions of ground literals) is decidable. In particular, we consider extensions where the indexes of arrays have the algebraic structure of Presburger Arithmetic and the theory of arrays is augmented with axioms for additional symbols such as dimension, sortedness, or the domain of definition of arrays. We provide methods for integrating available decision procedures for the theory of arrays and Presburger Arithmetic with automatic instantiation strategies which allow us to reduce the satisfiability problem for the extension of the theory of arrays to that of the theories decided by the available procedures. We show how to use both model-theoretic and equational theorem proving techniques to implement the instantiation strategies of the various extensions.

To reason about pointer-based data structures we have considered in [63] the data structure of singly-linked lists and defined a *Theory of Linked Lists* (*TLL*). The theory is expressive since it is capable of precisely expressing both data and reachability constraints, while ensuring decidability. Furthermore, its satisfiability problem (for existentially quantified formulae) is decidable but *NP*-complete. We also design a practical decision procedure for *TLL* which can be combined with a wide range of available decision procedures for theories in first-order logic.

In [21], we study the Satisfiability Modulo Theories (SMT) framework when the background theory is a combination of (disjoint) theories. The SMT framework relies on the integration of a Boolean solver and decision procedures for component theories. When combining decision procedures for several theories, we have investigated an efficient alternative to the common practice of integrating a Boolean solver with a decision procedure obtained by applying one of the well-known combination methods (e.g., the one by Nelson-Oppen). The key idea is to synchronize the various decision procedures for the component theories via the Boolean solver. The experimental evaluation of the proposed method has proved quite encouraging.

In [62], we consider the problem of augmenting decision procedures with the capability of computing conflict sets without degrading performances, as well as the problem of modularly constructing conflict sets for a combined theory. We also study how the computed conflict sets relate to an appropriate notion of minimality.

6.1.2. Parametric invariant checking by superposition

Participants: Françoise Bellegarde, Jean-François Couchot, Alain Giorgetti, Nikolaï Kosmatov, Silvio Ranise.

The Ph.D. dissertation of J.-F. Couchot [15] gathers our experience in applying the superposition calculus to the verification of safety for parameterized systems with structured data types. We have investigated the cases of data structured in sets, total functions and arrays. All these cases are reduced to an extension of the theory of arrays with extensionality. Since the full theory of arrays is undecidable, the work consists in identifying decidable fragments expressive enough to verify classes of programs with a practical interest. This work is implemented in *bam2rv* This tool repeatedly discharges proof obligations in the *haRVey* prover.

We have also identified two decidable fragments which are expressive enough to verify safety properties for two classes of distributed systems composed of any number of similar processes communicating by rendez-vous and broadcast. Invariants are checked and automatically strenghtened when the proof fails. Many examples of uniform distributed systems and a tool illustrate these results.

6.1.3. Tree automata and their extensions

Participants: Michaël Rusinowitch, Laurent Vigneron.

We have considered classes of tree automata combining automata with equality test and automata modulo equational theories with F. Jacquemard (SECSI project) [56]. These tree automata are obtained by extending their standard Horn clause representations with equational conditions and rewrite systems. We show in particular that a generalized membership problem (embedding the emptiness problem) is decidable by proving that the saturation of tree automata presentations with suitable paramodulation strategies terminates. These tree automata classes can be applied to the reachability problem for a fragment of pi-calculus.

6.1.4. Verification of copies convergence in distributed groupware systems

Participants: Abdessamad Imine, Michaël Rusinowitch.

We are interested in synchronizing replicated data using the Operational Transformation (OT) approach. Based on our algebraic framework for designing OT algorithms [25], we have proposed a compositional method for specifying complex collaborative objects [55]. The most important feature of this method is that designing an OT algorithm for the composed object can be obtained by reusing the OT algorithms of component objects. As a continuation of our previous work, we have proposed a new integration environment which is appropriate for collaborative edition based on linear objects such as a text or an ordered XML tree. The novelty of this environment is that it is scalable. Indeed, it can ensure data convergence for any number of group members and it can be deployed in Peer-to-Peer (P2P) network accordingly.

6.2. Security protocol verification

Keywords: *Exclusive-Or*, *Exponentiation*, *Protocol*, *Security*, *Verification*.

Cryptographic protocols are successfully analyzed using formal methods and many techniques have appeared in the litterature. However, formal approaches usually consider the encryption schemes as black boxes and assume that an adversary cannot learn anything from an encrypted message except if he has the key. Such an assumption is too strong in general since some attacks exploit in a clever way the interaction between protocol rules and properties of cryptographic operators. In [23], we give a list of some relevant algebraic properties of cryptographic operators, and for each of them, we provide examples of protocols or attacks using these properties. We also give an overview of the existing methods in formal approaches for analyzing cryptographic protocols under equational theories.

6.2.1. Extension of the Dolev-Yao model

Participants: Véronique Cortier, Michaël Rusinowitch, Mathieu Turuani.

Some attacks exploit in a clever way the interaction between protocol rules and algebraic properties of cryptographic operators. In [23], we provide a list of such properties and attacks as well as existing formal approaches for analyzing cryptographic protocols under algebraic properties.

Unbounded number of sessions. We have proposed a new class of security protocols using XOR, for which secrecy after an unbounded number of sessions is decidable [45]. The new class is important as it contains examples of key-management APIs, such as the IBM 4758 CCA API, which lie outside the classes for which secrecy has previously been shown to be decidable.

Bounded number of sessions. In [65] we present a unification algorithm modulo exclusive-or adapted from the algorithm by Baader & Schulz, but optimised for this operator. This was implemented in *CL-AtSe* [66], one of AVISPA's backend developed in Cassis for bounded number of sessions. Along with implementations of the unification modulo abelian groups and of intruder's deduction rules for the exclusive-or and exponential operators, this allowed *CL-AtSe* to analyse cryptographic protocols modulo these operators and their properties.

General equational theories. Most of the decision procedures for symbolic analysis of protocols are limited to a fixed set of algebraic operators associated with a fixed intruder theory. Focusing on ground deducibility and static equivalence (checking whether two sequences of messages are indistinguishable to an attacker), we have established general decidability theorems, requiring only loose, abstract conditions on the equational theory for messages [19]. These results apply to many useful theories like blind digital signatures, homomorphic encryption, XOR, and other associative-commutative functions.

We have obtained an algorithm for combining decision procedures for arbitrary intruder theories with disjoint sets of operators, provided that solvability of ordered intruder constraints, a slight generalization of intruder constraints, can be decided in each theory. This is the case for many decidable intruder theories. We have extended this algorithm for a class of non-disjoint theories including exponential with exponents ranging in an Abelian group (the exponential basis is not fixed unlike previous works) [40].

6.2.2. Soundness of the Dolev-Yao model

Participants: Véronique Cortier, Mathieu Turuani, Bogdan Warinschi, Eugen Zalinescu.

All the previous results rely on symbolic models of protocol executions in which cryptographic primitives are abstracted by symbolic expressions. This approach enables significantly simple and often automated proofs. However, the guarantees that it offers have been quite unclear compared to cryptographic models that consider issues of complexity and probability. Cryptographic models capture a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. Bridging the gap between the two approaches has deserved a lot of attention in the past recent years.

We have shown last year that it is possible to obtain the best of both cryptographic and formal worlds in the case of public encryption: fully automated proofs and strong, clear security guarantees. Specifically, for the case of protocols that use signatures and asymmetric encryption, we have established that symbolic integrity and secrecy proofs are sound with respect to the computational model. This result has been recently extended in order to obtain computationally sound symbolic secrecy in the presence of hash functions [43]. Computational soundness for symmetric encryption usually requires that encryption cycles cannot be generated during the execution. We have proved that detecting key cycles for a bounded number of sessions is decidable [46].

These soundness results require to explicitly represent the dependency of ciphertexts on randomness as labels. We have shown [42] that for a large class of security properties (that includes rather standard formulations for secrecy and authenticity properties), security of protocols in the simpler model implies security in the label-based model. Based on these results, we have recently implemented an AVISPA module for verifying security properties in a standard cryptographic model.

The indistinguishability of two pieces of data (or two lists of pieces of data) can be represented formally in terms of a relation called static equivalence. Static equivalence depends on an underlying equational theory. We have defined and justified an equational theory for standard, fundamental cryptographic operations. This equational theory yields a notion of static equivalence that implies computational indistinguishability [27]. In particular, we develop and analyze a principled formal account of guessing attacks in terms of static equivalence.

Following a different method, we have also studied how to directly prove properties on cryptographic protocols without any restriction, and with respect to an active, polynomial-time adversary [47]. This approach relies on a cryptographically sound formal logic, which does not require explicit reasoning about probability, asymptotic complexity, or the actions of a malicious intruder. This approach has been extended to reason symbolically about the security of key exchange protocols [48].

6.2.3. Security properties and advanced class of protocols

Participants: Véronique Cortier, Najah Chridi, Michaël Rusinowitch, Judson Santiago, Laurent Vigneron, Eugen Zalinescu.

Most previous results focus on secrecy and authentication for simple protocols like the ones from Clark & Jacob library. We explore several directions to cover more complex protocols and security properties.

Security Properties. For secrecy properties, decidability results and automatic tools have mainly focused on reachability-based secrecy while equivalence-based secrecy ensures a higher level of security and is closer to cryptographic notions of secrecy. We have studied in [44] under which hypotheses reachability-based secrecy can actually imply equivalence-based secrecy.

Non-repudiation services ensure that when two parties exchange informations over a network, neither one nor the other can deny having participated in this communication. We have proposed [64] a formal description for non-repudiation services, and a technique for verifying them automatically.

Group Protocols. Verifying security properties of group key agreement protocols is challenging: the number of participants is variable and unbounded, and the security properties related to dynamic membership in groups are difficult to express and usually not formally specified in the literature. We have investigated [22] the modeling of group protocols and more generally key contributing ones. This analysis was able to rediscover several attacks on well-known protocols such as A-GDH.2, SA-GDH.2, Asokan-Ginzboorg and Bresson-Chevassaut-Essiari-Pointcheval. From these preliminary results, we have defined a strategy based on constraint solving for automatically discovering flaws [41].

6.2.4. Intruder knowledge approximation

Participants: Yohan Boichut, Pierre-Cyrille Héam, Nikolaï Kosmatov, Olga Kouchnarenko, Laurent Vigneron.

When the number of sessions is unbounded, the security problem of cryptographic protocols is undecidable. Hence, we have proposed automated computations of over and under-approximations of the intruder knowledge using tree automata techniques [14].

In order to make this kind of technique available from high level specification languages like HLPSL and PROUVE [33], we define safe and sound abstractions of protocol transition systems into rewriting systems. In addition, we propose a new representation of secrecy properties, suitable to semi-decide the security problem [68]. These abstraction-based approximation techniques are implemented in TA4SP (see Section 5.1.4), one of the tools of the AVISPA platform. For better protocols analysis, we have developed in [31] a semi-algorithm to generate attack traces in the context of our approximations approach.

In [32], we extend our over-approximation approach to verify security protocols using algebraic properties. We have successfully verified the View Only protocol, a component of the Smartright system[5].

[5] http://www.smartright.org In the context of home digital network, this protocol - using the XOR operator - prevents users from unlawfully copying movies broadcast on the network.

6.3. Reachability analysis

Keywords: *Formal Specifications*, *Model-based Testing*, *Parametric Systems*, *Reachability*, *Regular Languages*, *Test Case Generation*.

6.3.1. Security in embedded systems

Participants: Fabrice Bouquet, Alain Giorgetti, Olga Kouchnarenko.

Security and safety have to be ensured for a large number of different mobile devices, and more and more embedded systems need to be certified. Nowadays not only smart card applications and aeronautics are concerned by this certification process but also transportation area - automotive, rail, etc.

We have started studying and developing methods and tools for verifying and validating embedded systems using model-checking, proof and test. Our aim is to automate as much as possible the analysis of embedded systems to allow a non specialist software engineer to ensure their safety and security. These new techniques are currently applied to smart-card case studies (RNTL POSE).

Model-checking by generation of annotations. In the Java Card framework, embedded software is developed in Java-like code. Its security is reinforced by enriching the Java code with JML annotations.

In [54], we have proposed a way to verify temporal properties of a Java class in an extension of JML (Java Modeling Language) called JTPL (Java Temporal Pattern Language). This extension particularly addresses the verification of liveness properties by automatically translating the temporal properties into JML annotations for this class. This automatic translation is implemented in a tool [53], [52] called *JAG* (JML Annotation Generator). Correctness of the generated annotations ensures that the temporal property is established for the executions of the class in isolation.

6.3.2. Model-based testing

Participants: Fabrice Bouquet, Thibaut Brocard, Stéphane Debricon, Frédéric Dadeau, Bruno Legeard, Vincent Pretre.

The need to offer better methods and tools for functional black-box testing of large scale systems has raised a large amount of research on generating tests from formal specifications. The BZ-TT approach is based on an original method of boundary-value extraction and preamble computation based on a customized constraint logic programming technology. This method has been validated on several real-size industrial applications. A book has been written on the Model-based testing [13].

We use the model-based testing in several application domains such as Smart Cards, Web Services and the security of airports [30].

We propose an integration of the model-based approach in the development process [39]. Our goal is to integrate several modeling processes (testing and developing). To achieve this integration, a UML notation is supported. We use a subset of this langage to generate test sequences [29]. The concept of boundary values is extended to objects [38].

We have started a new study for test generation. We propose to use security requirements into the generation process. In our proposal, the security requirements are translated into the behavior model for a functional test generation approach or into security properties. A case study on smart cards is considered to validate this approach [36]. In [37], a schema of security properties is used to define test generation strategies.

7. Contracts and Grants with Industry

7.1. RNTL

- RNTL project PROUVÉ[6] — *"Protocoles cryptographiques: Outils de Vérification automatique"*, duration: 3 years, started on November 2003. The goal of this project is the automatic verification of cryptographic protocols, for a large class of security properties and algebraic properties of the cryptographic primitives. There are five partners: CRIL Technology Systèmes Avancés, France Telecom R&D, INRIA Lorraine, LSV (ENS de Cachan), Verimag (Grenoble).
- RNTL project DANOCOPS — *"Détection Automatique de NOn-COnformités d'un Programme vis-à-vis de ses Spécifications"*, duration: 39 months, started on 1st January 2004. The goal of this project is to confront specification and program to find non-conformity. We propose to use an abstract representation of specification and source program, with constraints. There are five partners, two industrials: Thales division Systèmes Aéroportés, Axlog (SS2I), and three academics: I3S/Nice, LSR/Grenoble and LIFC/Besançon. The local coordinator is F. Bouquet.
- RNTL project POSÉ — *Security policies conformance testing for embedded systems*, duration: 2 years, started in December 2005. The objective is to provide automated tools for generating tests of security policies conformance of embedded systems. There are five partners: LEIRIOS, AXALTO, SILICOMP AQ, IMAG/LSR and INRIA Lorraine. The local coordinator is F. Bouquet.

7.2. Research result transfer

The BZ-Testing-Tools technology has been transfered to LEIRIOS Technologies, at the end of 2004. The partnership between the Cassis project and the R&D LEIRIOS Department, located at the TEMIS Scientific and Industrial area at Besançon, will be continued through projects (national and international call of work) or with a new transfer protocol. According to the law of innovation, F. Ambert, F. Bouquet, B. Legeard and F. Peureux are scientific consultants of LEIRIOS Technologies.

7.3. INTERREG

INTERREG VALID — We are working with the university of Geneve, LEIRIOS Technologies and Centre des Technologies de l'Information - État de Genève. The project concerns the test generation for the web services. The duration of the project is 18 months and it was started in July 2005.

8. Other Grants and Activities

8.1. International grants

- Project INRIA-CNPq (Brazil), DA CAPO — *Automated deduction for the verification of specifications and programs*. It is a project on the development of proof systems (like *haRVey*) for the verification of specifications and software components. The coordinators of this project are David Déharbe (UFRN Natal, Brazil) and Christophe Ringeissen. On the french side, DA CAPO also involves the PROTHEO project.
- Project INRIA-Tunisian Universities — *"Vérification et analyse de la sécurité et de la sûreté des systèmes critiques"*. The coordinators of this project are Nejib Ben Hadj-Alouane (ENSI Tunis, University Manouba) and Michaël Rusinowitch. On the french side, this project also involves the laboratory LAG (Grenoble).

8.2. National grants

- ACI GECCOO[7] —*"Génération de code certifié pour des applications orientées objet (Spécification, raffinement, preuve et détection d'erreurs)"*, duration: 3 years, started on July 2003.

[6] http://www.lsv.ens-cachan.fr/prouve/
[7] http://geccoo.lri.fr

This project aims at developing methods and tools for the design of object-oriented systems that require a high degree of security. In particular, the project focuses on the design of smart card applications, written in a subset of Java (like JavaCard), annotated with JML specifications. Partners are: TFC (LIFC), EVEREST (INRIA Sophia-Antipolis), PROVAL (INRIA Futurs & LRI), VASCO (LSR). The local coordinators are F. Bellegarde (LIFC) and S. Ranise (Nancy).

- ACI V3F —*Validation & Verification of programs with floating-point numbers*, duration: 3 years, started on October 2003. Cassis (B. Legeard) is the principal coordinator.

  The goal of this project is to provide tools to support the verification and validation process of programs with floating-point numbers. The underlying technology is based on constraint solving. Partners are: I3S-INRIA Sophia Antipolis, IRISA-INRIA Vertecs & Lande, CEA-LIST.

- ACI EDEMOI[8]—*Formal Modeling and Verification of Airport Security*, duration: 3 years, started on October 2003.

  The EDEMOI project aims at defining an approach for the construction and analysis of a precise reference document that models and structures current standards and associated recommendations. The exploitation of this model by the civil aviation authorities will improve airport security. Partners are: LSR-IMAG, CEDRIC-CNAM, ONERA Toulouse, GET ENST Paris.

- ACI SATIN —*Security Analysis for Trusted Infrastructures and Network protocols*, duration: 3 years, started on July 2004. Cassis (M. Rusinowitch) is the principal coordinator.

  The SATIN project aims at working on formal analysis and design of secure distributed systems, by taking advantage of the recent advances in algebraic modeling techniques. Partners are: CEA-DAM, France Telecom R&D, LANDE project - IRISA, VPS team, LIFO.

- ACI Jeunes Chercheurs CRYPTO[9] —*"Lien entre la cryptanalyse et l'étude logique des protocoles cryptographiques"*, duration: 3 years, started on September 2004.

  The CRYPTO project aims at establishing a link between the formal and the computational approaches for cryptographic protocols.

- ARA SSIA FormaCrypt—*Formal proofs and probabilistic semantics in cryptography*, duration: 3 years, started in January 2006.

  The verification of cryptographic protocols is a very active research area. Most works on this topic use either the computational approach, in which messages are bitstrings, or the formal approach, in which messages are terms. The computational approach is more realistic but more difficult to automate. The FormaCrypt project aims at bringing together these orthogonal approaches in order to get the best of the two worlds. Partners are: Liens (coordinator), SECSI project - LSV, Cachan.

- ARA SSIA COPS—*Composition Of Policies and Services*, duration: 3 years, started in December 2005.

  The aim is to build technologies enabling the security analysis of web services that take into account the potential flaws at communication level, at the access policy level or at the interface between communications and access policy. Partners are: IRIT Toulouse, LIM Marseille, Microsoft R&D.

- ARA SSIA ARROWS—*Safe Pointer-Based Data Structures: A Declarative Approach to their Specification and Analysis*, duration: 3 years, started in autumn 2005.

  Programming with pointers is quite a powerful and widely used technique to build many software systems with limited resources such as embedded systems or programs requiring recursive data structures. The goal of this project is to develop new specification languages for programs manipulating pointers which are sufficiently precise to express many interesting properties and, at the same time, support automatic analyses. Partners are: CAPP-LEIBNIZ Grenoble (coordinator), LILaC-Irit Toulouse. The local coordinator is S. Ranise.

[8] http://www-lsr.imag.fr/EDEMOI
[9] http://www.loria.fr/~cortier/aci.html

- QSL VALDA2—*Automated Software Verification using Automated Deduction*, duration: 2 years, started in 2005. With this action, we are working in the *Pôle de Recherche Scientifique et Technologique Intelligence Logicielle* within the theme *Qualité et sûreté des logiciels et systèmes informatiques*, funded by the *Contrat de Plan État-Région Lorraine 2000-2006*.
- QSL COWS—*Constraints for the Composition of Web Services*, duration: 2 years, started in 2006. This action is coordinated by O. Perrin (ECOO project) and L. Vigneron. It is another action of the theme *Qualité et sûreté des logiciels et systèmes informatiques*, funded by the *Contrat de Plan État-Région Lorraine 2000-2006*.
- VALMI — *Automated Validation of embedded micro-systems for electronic transaction*, duration: 18 months, started in November 2006. The aim of this project is to provide a methodology and automated tools for generating tests of electronic transaction for urban embedded system. There are three partners: ERG, LEIRIOS, Parkeon. The local coordinator is F. Bouquet.

8.3. International collaborations

- In the continuation of a PAI PROCOPE, we are working on the combination of automata-theoretic and rewriting techniques for the analysis of cryptographic protocols with the group of professor Thomas Wilke, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel.
- In the area of automated test generation from a formal model, we have an active collaboration with Dr Mark Utting from the Formal Method group from the University of Waikato[10]. This cooperation is supported by the France-New-Zealand scientific program.
- In the area of business applications, we are working on the soundness problem of coloured work-flow Petri nets with the Information System group of Professor K. van Hee from the Technical University of Eindhoven. This cooperation is supported by the NWO scientific program (The Netherlands).

8.4. Individual involvement

F. Bellegarde: director of the research team *Techniques Formelles et à Contraintes (TFC)* of the *Laboratoire d'informatique de Franche Comté (LIFC)*, board member of the *LIFC*, Editorial committee member of *Techniques et Science Informatique (TSI)*.
F. Bouquet: in charge of the Mobilization area (9 research projects, with 46 researchers and 8 laboratories) in ISTI Institute[11]; coordinator of Tools session of B'07; coordinator of INTERREG VALID. PC Member of two workshops: "Perspective on Integrating MDA and V&V", Modeva'06, workshop co-located with Models'06 and "Constraints in Software Testing, Verification and Analysis", CSTVA'06, co-located with CP'06.
V. Cortier: local coordinator of the ARA SSIA FormaCrypt (started in January 2006); coordinator of the ACI Jeunes Chercheurs CRYPTO; PC member of *19th IEEE Computer Security Foundations Workshop (CSFW'06), Venice*, *6th International Workshop on Automated Verification of Critical Systems (Avocs'06), Nancy*, *École de printemps 2006 Sécurité Informatique (EPSI'06), Alger*, *International Conference on Security and Cryptography (Secrypt'06), Setubal*; co-organizer and co-chair of the *Workshop on Formal and Computational Cryptography (FCC'06), Venice*.
O. Kouchnarenko: director of the research team *Techniques Formelles et à Contraintes (TFC)* of the *Laboratoire d'informatique de Franche Comté (LIFC)* since July 2006; PC member of "*Approches Formelles dans l'Assistance au Développement de Logiciels*", AFADL'06 and co-chair of "*Formal Specification and Development in B*". Co-chair of the "CSE 27" of the University of Franche-Comté.
B. Legeard: member of the Scientific council of the University of Franche-Comté. Coordinator of the ACI V3F.

[10] http://www.cs.waikato.ac.nz/Research/fm/index.html
[11] http://www.isti.info

S. Ranise: trustee of the project CALCULEMUS (Systems for Integrated Computation and Deduction); coordinator (with Cesare Tinelli) of the Satisfiability Modulo Theories Library (SMT-LIB) initiative; co-chair of "Calculemus 2006", co-located with ISSAC'06; PC member of two workshops affiliated with IJCAR'06, "Pragmatics of Decision Procedures in Automated Reasoning (PDPAR) 2006" and "Empirically Successful Computerized Reasoning (ESCoR) 2006".
C. Ringeissen: PC member of the 13th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning CALCULEMUS 2006; co-editor (with Alessandro Armando) of a special issue of Information & Computation on "Combining Logical Systems" [11].
M. Rusinowitch: member of the IFIP Working Group 1.6 (Rewriting); member of COST-GTAP prospective committee at INRIA; coordinator of the project ACI Sécurité SATIN. PC member of ASIAN'06, Asian Computing Science Conference; LPAR 2006; IEEE Symposium on Security and Privacy 2006; 19th IEEE Computer Security Foundations Workshop 2006; 8th International Symposium on Functional and Logic Programming 2006; COLSEC, Workshop on Collaboration and Security 2006; Rencontres Sécurité et Architecture Réseaux, Seignosse, June 6-9, 2006; Taiwanese-French Conference on Information Technology 2006.
L. Vigneron: member of the FTP steering committee; secretary of the IFIP Working Group 1.6 (Rewriting); Organising Committee Chair of the first International School on Rewriting, ISR'2006; PC member of the FCS-ARSPA'06 and Strategies'06 workshops; web master of the site *Rewriting Home Page*, of the RTA conference site, and of the web page for the IFIP Working Group 1.6.
We are involved in several lectures of the "Master Informatique" of the universities of Nancy. V. Cortier is in charge of the lecture on *Theory of the security*, S. Ranise and C. Ringeissen are in charge of the lecture on *Decision procedures and program verification*.

8.5. Visits of team members

V. Cortier visited Ralf Küsters at the University of Kiel, Germany, during one week in April 2006 and at the ETH Zurich, Switzerland, during one week in October 2006. The subject of the collaboration was about computational branching properties like fairness for contract-signature protocols.
O. Kouchnarenko visited N. Sidorova at the Technical University of Eindhoven, The Netherlands, during 5 weeks in July/August 2006. The subject of the collaboration concerns the soundness (a kind of security) problem in work-flow Petri Nets.

9. Dissemination

9.1. Awards

The **2006 A'DOC Award of Université de Franche-Comté** has been obtained by Yohan Boichut for his Ph. D. dissertation "*Approximation pour la vérification automatique de protocoles de sécurité* ".

9.2. Committees

V. Cortier is member of the SPECIF committee to award the best Ph. D. dissertations in theoretical computer science.
C. Ringeissen is referee for the Ph. D. thesis of Enrica Nicolini (Milan).
M. Rusinowitch is referee for the thesis of Mathieu Blanc (Orléans), Yacine Bouzida (Rennes), Stéphanie Delaune (Cachan) and examiner for the thesis of Richard Bonichon (Paris).

9.3. Seminars, workshops, and conferences

Besides conference talks mentioned in the publication list, we have given the following talks.
F. BOUQUET, *Model-Based Testing from UML models* , talk at "Modeling and Verifying Software: Model based Testing", Ph.D. tutorial, Geneva (Switzerland), October 12th 2006.

V. CORTIER, *Computationally Sound Security Proof using Formal Models*, talk at the Cryptography Seminar, Rennes, January 27th 2006.
V. CORTIER, *Où va la recherche en France? Des jeunes chercheurs témoignent.* Salon du livre, dans le cadre du Bar des Sciences, Paris, March 17th 2006.
V. CORTIER, *When reachability-based secrecy implies equivalence-based secrecy in security protocols*, talk at the Artist 2 Workshop on Specification and Verification of Secure Embedded Systems, Pisa, Italy, May 18th 2006.
V. CORTIER, *On the use of formal models for proving cryptographic security notions*, invited talk at the Information-MFCSIT'06 conference, Special Session on Formal Approaches to Security, Cork, Irland, August 2006.
V. CORTIER, *Verification of cryptographic protocols: techniques and link to cryptanalysis*, invited talk at the International Workshop on Automated Verification of Critical Systems Avocs 2006, Nancy, France, September 19th 2006.
P.-C. HÉAM, *Clôtures transitives de semi-commutations*, seminar at IRISA, Rennes, April 2006.
O. KOUCHNARENKO, Seminar on Verification of component-based systems at the Technical University of Eindhoven, The Netherlands, July 27th 2006.
S. RANISE, C. RINGEISSEN, *Decision Procedures for the formal analysis of software* [49], tutorial (in collaboration with P. FONTAINE and D. DÉHARBE), 3rd International Colloquium on Theoretical Aspects of Computing (ICTAC), Tunis, November 21st 2006.
M. RUSINOWITCH, *Invited Talk on Rewriting Approach for Security Analysis*, SecReT workshop, Venise, July 15th 2006.
M. RUSINOWITCH, *Tutorial on Security*, International School on Rewriting ISR 2006, Nancy, July 6th 2006.
M. RUSINOWITCH, Seminar on Protocol Analysis at SupCom Tunis, April 24th 2006.

10. Bibliography

Major publications by the team in recent years

[1] A. ARMANDO, S. RANISE, M. RUSINOWITCH. *A Rewriting Approach to Satisfiability Procedures*, in "Journal of Information and Computation — Special Issue on Rewriting Techniques and Applications (RTA'01)", vol. 183, n^{o} 2, June 2003, p. 140–164.

[2] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B: A Constraint Solver to Animate a B Specification*, in "International Journal of Software Tools for Technology Transfer, STTT", vol. 6, n^{o} 2, August 2004, p. 143–157.

[3] Y. CHEVALIER, L. VIGNERON. *Strategy for Verifying Security Protocols with Unbounded Message Size*, in "Journal of Automated Software Engineering", vol. 11, n^{o} 2, April 2004, p. 141–166.

[4] H. COMON-LUNDH, V. CORTIER. *Security properties: two agents are sufficient*, in "Science of Computer Programming", vol. 50, n^{o} 1-3, March 2004, p. 51–71, http://www.loria.fr/~cortier/Papiers/ComonCortierSCP03.ps.

[5] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Compiling and Verifying Security Protocols*, in "Logic for Programming and Automated Reasoning (LPAR'00), Reunion Island, France", A. VORONKOV, M. PARIGOT (editors). , Lecture Notes in Computer Science, vol. 1955, Springer, 2000, p. 131–160.

[6] B. LEGEARD, F. PEUREUX. *B-Testing-Tools : génération de tests aux limites à partir de spécifications B*, in "TSI, Techniques et Sciences Informatiques, Hermès-Lavoisier", vol. 21, n^{o} 9, 2002, p. 1189–1218.

[7] B. LEGEARD, F. PEUREUX, M. UTTING. *Automated Boundary Testing from Z and B*, in "Formal Methods Europe (FME 2002)", L.-H. ERIKSSON, P. LINDSAY (editors). , Lecture Notes in Computer Science, vol. 2391, Springer, 2002, p. 21–40.

[8] M. RUSINOWITCH, M. TURUANI. *Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete*, in "Theoretical Computer Science", vol. 299, April 2003, p. 451–475, http://www.loria.fr/~rusi/pub/tcsprotocol.ps.gz.

[9] C. TINELLI, C. RINGEISSEN. *Unions of Non-Disjoint Theories and Combinations of Satisfiability Procedures*, in "Theoretical Computer Science", vol. 290, n^{o} 1, 2003, p. 291–353.

Year Publications

Books and Monographs

[10] A. ARMANDO, D. BASIN, J. CUELLAR, M. RUSINOWITCH, L. VIGANÒ (editors). *Special Issue of the Journal of Automated Reasoning on Automated Reasoning for Security Protocol Analysis*, to appear, Kluwer, 2006.

[11] A. ARMANDO, C. RINGEISSEN (editors). *Special Issue on Combining Logical Systems*, vol. 204, n^{o} 10, Information and Computation, Elsevier, 2006.

[12] A. BIGATTI, S. RANISE (editors). *13th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning (Calculemus)*, July 2006.

[13] M. UTTING, B. LEGEARD. *Practical Model-Based Testing - A tools approach*, 550 pages, ISBN 0-12-372501-1, Morgan and Kaufmann, Elsevier Science, 2006.

Doctoral dissertations and Habilitation theses

[14] Y. BOICHUT. *Approximations pour la vérification automatique de protocoles de sécurité*, Thèse de Doctorat, LIFC, Université de Franche-Comté, Besançon (France), septembre 2006.

[15] J.-F. COUCHOT. *Vérification d'invariant de systèmes paramétrés par superposition*, Thèse de Doctorat, LIFC, Université de Franche-Comté, avril 2006, http://lifc.univ-fcomte.fr/publis/manuscrits/theseCouchot06.pdf.

[16] F. DADEAU. *Évaluation symbolique à contraintes pour la validation - Application à Java/JML*, Thèse de Doctorat, LIFC, Université de Franche-Comté, Besançon (France), 2006.

[17] A. IMINE. *Conception Formelle d'Algorithmes de Réplication Optimiste. Vers l'Edition Collaborative dans les Réseaux Pair-à-Pair*, Thèse de doctorat, Université Henri Poincaré, Nancy, décembre 2006.

[18] J. SANTOS SANTIAGO. *Spécification et analyse de protocoles complexes dans AVISPA*, Thèse de Doctorat, Université Nancy 2, novembre 2006.

Articles in refereed journals and book chapters

[19] M. ABADI, V. CORTIER. *Deciding knowledge in security protocols under equational theories*, in "Theoretical Computer Science", To appear, 2006, http://www.loria.fr/~cortier/Papiers/AbadiCortierTCS06.ps.

[20] M. BACKES, A. DATTA, A. DEREK, J. C. MITCHELL, M. TURUANI. *Compositional Analysis of Contract Signing Protocols*, in "Theoretical Computer Science", To appear, 2006.

[21] M. BOZZANO, R. BRUTTOMESSO, A. CIMATTI, T. JUNTTILA, P. VAN ROSSUM, S. RANISE, R. SEBASTIANI. *Efficient Theory Combination via Boolean Search*, in "Journal of Information and Computation", Special Issue on Combining Logical Systems, vol. 10, n^{o} 204, 2006, p. 1411–1596.

[22] N. CHRIDI, L. VIGNERON. *Sécurité des communications de groupe*, in "Revue de l'Électricité et de l'Électronique", vol. 6/7, juin/juillet 2006, p. 51–60.

[23] V. CORTIER, S. DELAUNE, P. LAFOURCADE. *A Survey of Algebraic Properties Used in Cryptographic Protocols*, in "Journal of Computer Security", vol. 14, n^{o} 1, 2006, p. 1–43, http://www.loria.fr/~cortier/Papiers/survey.ps.

[24] V. CORTIER, X. GOAOC, M. LEE, H.-S. NA. *A note on maximally repeated sub-patterns of a point set*, in "Discrete Mathematics", vol. 16, August 2006, p. 1965–1968, http://hal.inria.fr/inria-00070247.

[25] A. IMINE, M. RUSINOWITCH, G. OSTER, P. MOLLI. *Formal Design and Verification of Operational Transformation Algorithms for Copies Convergence*, in "Theoretical Computer Science", vol. 351, n^{o} 2, February 2006, p. 167–183.

[26] S. RANISE, C. TINELLI. *Satisfiability Modulo Theories*, in "IEEE Magazine on Intelligent Systems", vol. 21, n^{o} 6, November/December 2006, p. 71–81.

Publications in Conferences and Workshops

[27] M. ABADI, M. BAUDET, B. WARINSCHI. *Guessing Attacks and the Computational Soundness of Static Equivalence*, in "Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS, Vienna, Austria", L. ACETO, A. INGÓLFSDÓTTIR (editors). , Lecture Notes in Computer Science, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS, vol. 3921, Springer, March 2006, p. 398–412.

[28] C. ARORA, M. TURUANI. *Adding Integrity to the Ephemerizer's Protocol*, in "Proc. of Sixth International Workshop on Automated Verification of Critical Systems - AVoCS", September 2006, p. 146–151.

[29] E. BERNARD, F. BOUQUET, A. CHARBONNIER, B. LEGEARD, F. PEUREUX, M. UTTING, E. TORREBORRE. *Model-based Testing from UML Models*, in "Workshop on Model-based Testing, INFORMATIK'06, Dresden, Germany", LNI, Lecture Notes in Informatics, ISBN 978-3-88579-188-1, vol. P-94, October 2006, p. 223–230.

[30] D. BERT, F. BOUQUET, Y. LEDRU, S. VIGNES. *Validation of Regulation Documents by Automated Analysis of Formal Models*, in "REMO2V'06, Int. Workshop on Regulations Modelling and their Validation and Verification (in conjunction with CAiSE'06), Luxembourg", June 2006.

[31] Y. BOICHUT, T. GENET. *Feasible Trace Reconstruction for Rewriting Approximations*, in "Rewriting Techniques and Applications, 17th International Conference, RTA-06, Seattle, USA", Lecture Notes in Computer Science, vol. 4098, Springer, August 2006, p. 123–135.

[32] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Handling Algebraic Properties in Automatic Analysis of Security Protocols*, in "3rd International Colloquium on Theoretical Aspects of Computing, ICTAC, Tunis, Tunisia", Lecture Notes in Computer Science, vol. 4281, November 2006, p. 153–167.

[33] Y. BOICHUT, N. KOSMATOV, L. VIGNERON. *Validation of Prouve Protocols using the Automatic Tool TA4SP*, in "Proceedings of 3rd Taiwanese-French Conference on Information Technology (TFIT), Nancy, France", March 2006, p. 467–480.

[34] M. P. BONACINA, S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Decidability and Undecidability Results for Nelson-Oppen and Rewrite-Based Decision Procedures*, in "Proc. of the 3rd Int. Conference on Automated Reasoning (IJCAR), Seattle, WA, USA", LNAI, n$^{\text{o}}$ 4130, August 2006, p. 513–527.

[35] M. S. BOUASSIDA, N. CHRIDI, I. CHRISMENT, O. FESTOR, L. VIGNERON. *Automatic Verification of a Key Management Architecture for Hierarchical Group Protocols*, in "Proceedings of 5th Conference on Security and Network Architectures (SAR), Seignosse, France", F. CUPPENS, H. DEBAR (editors). , June 2006, p. 381–397.

[36] F. BOUQUET, F. CELLETTI, G. DEBOIS, A. DE LAVERNETTE, E. JAFFUEL, J. JULLIAND, B. LEGEARD, J. LIDOINE, J.-C. PLESSIS, P.-A. MASSON. *Model-Based Security Testing, Application to a Smart Card Identity Applet*, in "7th Int. Conf. on Smart Cards, eSmart, Sophia-Antipolis, France", September 2006.

[37] F. BOUQUET, F. DADEAU, J. GROSLAMBERT, J. JULLIAND. *Safety Property Driven Test Generation from JML Specifications*, in "1st Int. Workshop on Formal Approaches to Testing and Runtime Verification, FATES/RV, Seattle, WA, USA", LNCS, vol. 4262, Springer, August 2006, p. 225–239.

[38] F. BOUQUET, F. DADEAU, B. LEGEARD. *Automated Boundary Test Generation from JML Specifications*, in "FM'06, 14th Int. Conf. on Formal Methods, Hamilton, Canada", LNCS, vol. 4085, Springer, August 2006, p. 428–443.

[39] F. BOUQUET, S. DEBRICON, B. LEGEARD, J.-D. NICOLET. *Extending the Unified Process with Model-Based Testing*, in "Proceedings 3rd International Workshop, MoDeVa: Model Development, Validation and Verification, Genova, Italy", October 2006, p. 2-15.

[40] Y. CHEVALIER, M. RUSINOWITCH. *Hierarchical Combination of Intruder Theories*, in "Proceedings of 17th International Conference, RTA 2006, Seattle (WA)", F. PFENNING (editor). , Lecture Notes in Computer Science, vol. 4098, Springer, August 2006, p. 108–122.

[41] N. CHRIDI, L. VIGNERON. *Strategy for Flaws Detection based on a Services-driven Model for Group Protocols*, in "Proceedings of the 1st Workshop on Constraints in Software Testing, Verification and Analysis, CSTVA, Nantes, France", B. BLANC, A. GOTLIEB, C. MICHEL (editors). , September 2006, p. 88–99.

[42] V. CORTIER, H. HÖRDEGEN, B. WARINSCHI. *Explicit Randomness is not Necessary when Modeling Probabilistic Encryption*, in "Workshop on Information and Computer Security (ICS 2006), Timisoara, Romania", September 2006.

[43] V. CORTIER, S. KREMER, R. KÜSTERS, B. WARINSCHI. *Computationally Sound Symbolic Secrecy in the Presence of Hash Functions*, in "Proceedings of the 26th Conference on Fundations of Software Technology and Theoretical Computer Science (FSTTCS'06), Kolkata, India", N. GARG, S. ARUN-KUMAR (editors). , Lecture Notes in Computer Science, vol. 4337, Springer, December 2006, p. 176–187.

[44] V. CORTIER, M. RUSINOWITCH, E. ZALINESCU. *Relating two standard notions of secrecy*, in "Proceedings of 20th Int. Conference on Computer Science Logic (CSL'06), Szeged, Hungary", Z. ESIK (editor). , Lecture Notes in Computer Science, vol. 4207, Springer, September 2006, p. 303–318, http://www.loria.fr/~cortier/Papiers/Secrecy_CSL06.pdf.

[45] V. CORTIER, G. STEEL. *On the Decidability of a Class of XOR-based Key-management APIs*, in "Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'06), Seattle, Washington", August 2006, http://www.loria.fr/~cortier/Papiers/CortierSteelFCS06.pdf.

[46] V. CORTIER, E. ZALINESCU. *Deciding key cycles for security protocols*, in "Proc. of the 13th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), Phnom Penh, Cambodia", Lecture Notes in Artificial Intelligence, vol. 4246, Springer, November 2006, p. 317–331.

[47] A. DATTA, A. DEREK, J. MITCHELL, A. ROY, V. SHMATIKOV, M. TURUANI, B. WARINSCHI. *Computationally Sound Compositional Logic for Security Protocols*, in "2nd Workshop on Formal and Computational Cryptography, FCC, Venice, Italy", V. CORTIER, S. KREMER (editors). , 2006, http://hal.inria.fr/inria-00080593/en/.

[48] A. DATTA, A. DEREK, J. C. MITCHELL, B. WARINSCHI. *Computationally Sound Compositional Logic for Key Exchange Protocols*, in "Proceedings of 19th IEEE Computer Security Foundations Workshop", 2006, p. 321–334.

[49] D. DÉHARBE, P. FONTAINE, S. RANISE, C. RINGEISSEN. *Decision Procedures for the Formal Analysis of Software*, in "3rd International Colloquium on Theoretical Aspects of Computing, ICTAC, Tunis, Tunisia", Lecture Notes in Computer Science, Tutorial, vol. 4281, Springer, November 2006, p. 366–370.

[50] S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Deciding Extensions of the Theory of Arrays by Integrating Decision Procedures and Instantiation Strategies*, in "Proc. of the 10th European Conference on Logics in Artificial Intelligence (JELIA)", LNCS, vol. 4160, September 2006, p. 177–189.

[51] S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Deciding Extensions of the Theory of Arrays by Integrating Decision Procedures and Instantiation Strategies*, in "Proc. of the IJCAR'06 Ws. PDPAR: Pragmatical Aspects of Decision Procedures in Automated Reasoning, Seattle, WA, USA", B. COOK, R. SEBASTIANI (editors). , August 2006.

[52] A. GIORGETTI, J. GROSLAMBERT. *JAG: JML Annotation Generation for Verifying Temporal Properties*, in "FASE'2006, Fundamental Approaches to Software Engineering, Vienna, Austria", LNCS, vol. 3922, Springer, March 2006, p. 373–376, http://dx.doi.org/10.1007/11693017_27.

[53] A. GIORGETTI, J. GROSLAMBERT. *JAG : Génération d'annotations JML pour vérifier des propriétés temporelles*, in "AFADL'06, Approches Formelles dans l'Assistance au Développement de Logiciels, Paris, France", Session outils, March 2006, http://lifc.univ-fcomte.fr/publis/papers/pub/2006/RT2006-02.pdf.

[54] J. GROSLAMBERT, J. JULLIAND, O. KOUCHNARENKO. *JML-based Verification of Liveness Properties on a Class*, in "SAVCBS'06, Specification and Verification of Component-Based Systems, Portland, Oregon, USA", November 2006.

[55] A. IMINE. *Component-based Specification of Collaborative Objects*, in "The Second International Workshop on Views On Designing Complex Architectures (VODCA), Bertinoro, Italy", September 2006.

[56] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Tree Automata with Equality Constraints Modulo Equational Theories*, in "Proceedings of 3rd International Joint Conference on Automated Reasoning, IJCAR, Seattle (WA)", U. FURBACH, N. SHANKAR (editors). , Lecture Notes in Artificial Intelligence, vol. 4130, Springer, August 2006, p. 557–571.

[57] H. KIRCHNER, S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Automatic Combinability of Rewriting-Based Satisfiability Procedures*, in "Proc. of the 13th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), Phnom Penh, Cambodia", Lecture Notes in Artificial Intelligence, vol. 4246, Springer, November 2006, p. 542–556.

[58] H. KIRCHNER, S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Building and Combining Satisfiability Procedures for Software Verification*, in "Proceedings of 3rd Taiwanese-French Conference on Information Technology (TFIT), Nancy, France", March 2006, p. 125–139.

[59] N. KOSMATOV. *A Constraint Solver for Sequences and its Applications*, in "Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06), Dijon, France", April 2006, p. 404–408.

[60] G. OSTER, P. URSO, P. MOLLI, A. IMINE. *Data Consistency for P2P Collaborative Editing*, in "ACM Conference on Computer Supported Cooperative Work, (CSCW), Alberta, Canada", November 2006.

[61] S. RANISE. *Satisfiability Solving for Program Verification: towards the efficient Combination of Automated Theorem Provers and Satisfiability Modulo Theory Tools*, in "Proc. of the IJCAR'06 Ws. DISPROVING: Non-Theorems, Non-Validity, Non-Provability, Seattle, WA, USA", W. AHRENDT, P. BAUMGARTNER, H. DE NIVELLE (editors). , Invited paper, August 2006, p. 49–58.

[62] S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Producing Conflict Sets for Combination of Theories*, in "Pragmatics of Decision Procedures in Automated Reasoning (PDPAR), Seattle (WA)", B. COOK, R. SEBASTIANI (editors). , Workshop affiliated to the 3rd International Joint Conference on Automated Reasoning, IJCAR, August 2006.

[63] S. RANISE, C. ZARBA. *A Theory of Singly-Linked Lists and its Extensible Decision Procedure*, in "Proc. of the 4th IEEE International Conference on Software Engineering and Formal Methods (SEFM), Pune, India", IEEE Computer Society Press, September 2006.

[64] J. SANTIAGO, L. VIGNERON. *Automatically Analysing Non-repudiation with Authentication*, in "Proceedings of 3rd Taiwanese-French Conference on Information Technology (TFIT), Nancy, France", March 2006, p. 541–554.

[65] M. TUENGERTHAL, R. KUESTERS, M. TURUANI. *Implementing a Unification Algorithm for Protocol Analysis with XOR*, in "Proc. of UNIF'06 - 20th International Workshop on Unification", 2006, p. 1–5.

[66] M. TURUANI. *The CL-AtSe Protocol Analyser*, in "Term Rewriting and Applications - Proc. of RTA, Seattle, WA, USA", Lecture Notes in Computer Science, vol. 4098, 2006, p. 277–286.

Internal Reports

[67] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Handling Algebraic Properties in Automatic Analysis of Security Protocols*, Research Report, n^{o} 5857, INRIA, March 2006, http://hal.inria.fr/inria-00070169.

[68] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Automatic Abstraction Generation : How to Make an Expert Verification Technique for Security Protocols available to Non-expert Users*, 21 pages, Research Report, n^{o} 6039, INRIA, November 2006, https://hal.inria.fr/inria-00116918.

References in notes

[69] F. BAADER, K. U. SCHULZ. *Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures*, in "Journal of Symbolic Computation", vol. 21, n^{o} 2, February 1996, p. 211–243.

[70] F. BELLEGARDE, C. DARLOT, J. JULLIAND, O. KOUCHNARENKO. *Reformulation: a Way to Combine Dynamic Properties and Refinement*, in "International Symposium Formal Methods Europe (FME 2001)", LNCS, vol. 2021, Springer-Verlag, 2001.

[71] E. BERNARD, B. LEGEARD, X. LUCK, F. PEUREUX. *Generation of Test Sequences from Formal Specifications: GSM 11-11 Standard Case-Study*, in "International Journal on Software Practice and Experience", vol. 34, n^{o} 10, 2004, p. 915–948.

[72] F. BOUQUET, B. LEGEARD. *Reification of Executable Test Scripts in Formal Specification-Based Test Generation: The Java Card Transaction Mechanism Case Study*, in "Formal Methods, FME 2003", vol. 2805, Springer-Verlag, September 2003, p. 778–795.

[73] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B - A Constraint Solver for B*, in "International Conference on Tools and Algorithms for Construction and Analysis of Systems, TACAS2002, Grenoble, France", Lecture Notes in Computer Science, vol. 2280, Springer, April 2002, p. 188–204.

[74] D. DÉHARBE, S. RANISE. *Light-Weight Theorem Proving for Debugging and Verifying Units of Code*, in "Proc. of the International Conference on Software Engineering and Formal Methods (SEFM03), Brisbane, Australia", IEEE Computer Society Press, September 2003, http://www.loria.fr/~ranise/pubs/sefm03.ps.gz.

[75] S. EVEN, O. GOLDREICH. *On the Security of Multi-Party Ping-Pong Protocols*, in "IEEE Symposium on Foundations of Computer Science", 1983, p. 34-39, http://citeseer.ist.psu.edu/46982.html.

[76] G. T. LEAVENS, A. L. BAKER, C. RUBY. *JML: a Java Modeling Language*, in "Formal Underpinnings of Java Workshop (at OOPSLA '98)", October 1998, http://www-dse.doc.ic.ac.uk/~sue/oopsla/cfp.html.