INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Project-Team Cassis*

# *Combining approaches for the security of infinite state systems*

*Nancy - Grand Est*

Theme : Programs, Verification and Proofs

Activity Report

2010

# Table of contents

# 1. Team

**Research Scientists**

Véronique Cortier [DR, CNRS-LORIA, HdR]
Christophe Ringeissen [CR, INRIA-LORIA, HdR]
Michaël Rusinowitch [Team Leader, DR, INRIA-LORIA, HdR]
Mathieu Turuani [CR, INRIA-LORIA]

**Faculty Members**

Fabrice Bouquet [PR, Université Franche-Comté, HdR]
Frédéric Dadeau [MC, Université Franche-Comté]
Alain Giorgetti [MC, Université Franche-Comté]
Pierre-Cyrille Héam [MC, Université Franche-Comté, HdR]
Olga Kouchnarenko [Vice-head of project team, PR, Université Franche-Comté, LIFC, HdR]
Abdessamad Imine [MC, Université Nancy 2]
Laurent Vigneron [MC, Université Nancy 2]

**Technical Staff**

Philippe Paquelier [Engineer FP7 SecureChange, LIFC, from February 1]

**PhD Students**

Mumtaz Ahmad [SFERE (Pakistan), LORIA]
Mathilde Arnaud [project AVOTÉ]
Tigran Avanesov [project FP7 AVANTSSAR, LORIA]
Pierre-Christophe Bué [MENRT, LIFC]
Kalou Cabrera [project TASCCC, LIFC]
Jérome Cantenot [Council of Great Besançon, LIFC]
Asma Cherif [MENRT, LORIA]
Stefan Ciobaca [project AVOTÉ]
Roméo Courbis [LIFC]
Stéphane Debricon [project FP7 SecureChange, LIFC]
Aloïs Dreyfus [MENRT, LIFC since November 1]
Elizabeta Fourneret [project FP7 SecureChange, LIFC]
Vincent Hugot [DGA, LIFC]
Adrien de Kermadec [project VALMI and ATER UFC since September 1, LIFC]
Jonathan Lasalle [project VETESS, LIFC]
Mohamed Anis Mekki [project FP7 AVANTSSAR, LORIA]
Elena Tushkanova [INRIA, LIFC]

**Post-Doctoral Fellow**

Valerio Senni [Post-doctoral ERCIM, since May 1, 2010]

**Administrative Assistant**

Emmanuelle Deschamps

# 2. Overall Objectives

## 2.1. Background

Cassis is a joint project between the *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 with 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 because of the increasing number of security-sensitive systems. The failure of these critical systems can have dramatic consequences since they may be 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 the discovery of 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 as complementary.

The goal of our project is to contribute to new combinations of these three verification techniques in a framework that would apply 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 the 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, testing) by taking advantage of the complementary scopes and of their respective algorithmic contributions.

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

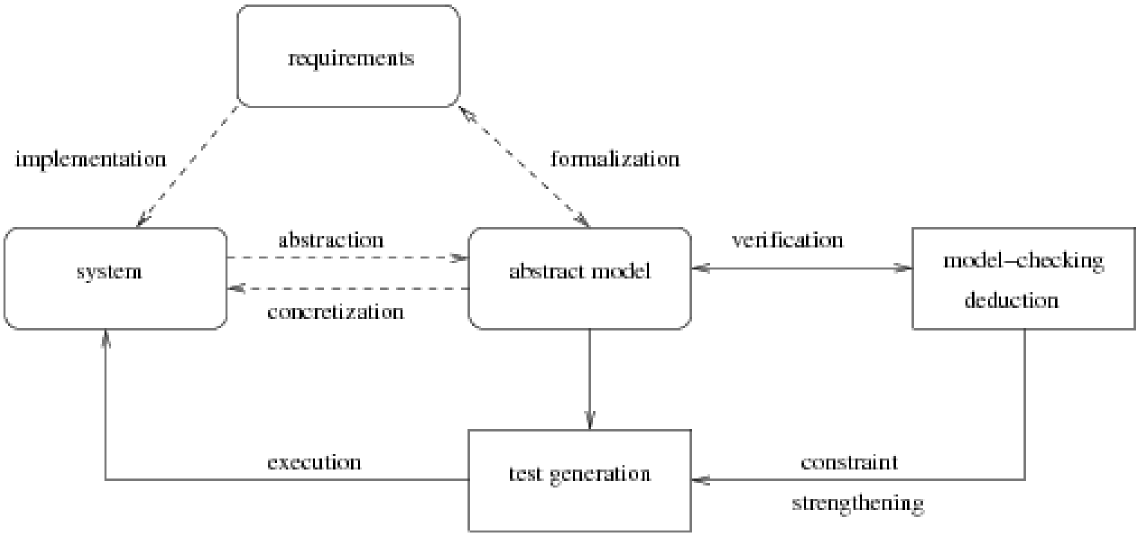

*Figure 1. Software validation in Cassis.*

## 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, for example, theorem-proving and model-checking.

The behavior of infinite states systems is expressed in 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 relies 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 $\mathcal{T}$ [61]. 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:
    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 (traces) as test cases with the oracle.

    After the generation phase, a concretization is used to produce the test drivers.Furthermore, the kernel of the engine allows one to perform specification animations in order to validate the model [65].

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 work 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.

## 2.4. Highlights

Véronique Cortier has received a *starting grant* from the European Research Council (ERC). Her project, called *ProSecure* (Provably secure systems: foundations, design, and modularity), will start in 2011 for five years.

# 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 until it 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.

We investigate techniques to incorporate the use of decision procedures in the model-checking of infinite state systems. The state of such systems is described by the models of theories specifying data types (such as integers or arrays) and their behavior is identified by (possibly infinite) sequences of these models which share the interpretation of the symbols interpreted in the theories (e.g., the addition over the integers). In this context, checking if a system satisfies a certain property may be reduced to checking the satisfiability of a formula in the theory obtained as the combination of the theories describing the sequence of states in the computation. To solve this problem, it is crucial to develop new combination methods for non-disjoint unions of theories.

## 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 [6], 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.

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, some 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

We have introduced 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 [62] and Java Card Virtual Machine Transaction mechanism [64]), information system and for embedded software [74].

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 in several main points from previous works (e.g. [69]): 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. Second orientation is to extend the result to object oriented specifications as UML/OCL.Third orientation is to extend the coverage of method for security aspect.

## 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. 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.

## 4.4. Verification of Web Services

Driven by rapidly changing requirements and business needs, IT systems and applications are undergoing a paradigm shift: components are replaced by services, distributed over the network, and composed and reconfigured dynamically in a demand-driven way into service-oriented architectures [1]. Exposing services in future network infrastructures means a wide range of trust and security issues need to be adressed. Solving them is extremely hard since making the service components trustworthy is not sufficient: composing services leads to new subtle and dangerous vulnerabilities due to interference between component services and policies, the shared communication layer, and application functionality. Thus, one needs validation of both the service components and their composition into secure service architectures. In this context, there is an obvious need of applying formal methods. Our project aims at applying our proof and constraint solving techniques to reason on web services. More precisely, we focus on the composition problem in the presence of security policies.

# 5. Software

## 5.1. Protocols Verification Tools

**Participants:** Pierre-Cyrille Héam, Olga Kouchnarenko, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

### *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 it is well supported. The *AVISPA* Tool compares favourably to related systems in 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.

### *5.1.2. CL-AtSe*

We develop, as a first back-end of *AVISPA*, *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 [70]) 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* includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing secrecy, authentication, fairness, non-abuse freeness, advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators such as XOR (exclusive or) and Exp (modular exponentiation). The handling of XOR and Exp has required to implement an optimized version of the combination algorithm of Baader & Schulz [59] for solving unification problems in disjoint unions of arbitrary theories.

*CL-AtSe* has been successfully used [58] 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 (see [76] for tool details and precise benchmarks).

[1] see e.g. http://osoa.org/display/Main/Service+Component+Architecture+Home

[2] http://www.avispa-project.org

### 5.1.3. *TA4SP*

We have developed, as a second back-end of *AVISPA*, 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. 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 in the rewrite model if it manages to compute an underapproximation 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 Kaochow protocol version 2.

TA4SP handles protocols using operators with algebraic properties. Thanks to a recent quadratic completion algorithm new experimental results have been obtained, for example for the Encrypted Key Exchange protocol (EKE2) using the exponential operator.

Recently, TA4SP was used in [75] to analyse a hierarchy of authentication properties.

## 5.2. Testing Tools

**Participants:** Fabrice Bouquet, Frédéric Dadeau, Philippe Paquelier.

In December 2008, we have started the redevelopment of our original testing tools environment, with two objectives: first, refactoring the existing developments, and, second, providing an open platform aiming at gathering together the various developments, increasing the reusability of components. The resulting platform, named Hydra, is a Eclipse-like platform, based on Plug-ins architecture. Plug-ins can be of five kinds: *parser* is used to analyze source files and build an intermediate format representation of the source; *translator* is used to translate from a format to another or to a specific file; *service* denotes the application itself, i.e. the interface with the user; *library* denotes an internal service that can be used by a service, or by other libraries; *tool*: encapsulates an external tool. The following services have been developed so far:

- BZPAnimator: performs the animation of a BZP model (a B-like intermediate format);
- Angluin: makes it possible to perform a machine learning algorithm (à la Angluin) in order to extract an abstraction of a system behavior;
- UML2SMT: aims at extracting first order logic formulas from the UML Diagrams and OCL code of a UML/OCL model to check them with a SMT solver.

These services involve various libraries (sometimes reusing each other), and rely on several *tool* plug-ins that are: SMTProver (encapsulating Z3 solver), PrologTools (encapsulating CLPS-B solver), Grappa (encapsulating a graph library). The transfer of the existing work on test generation from B abstract machines, JML, statecharts using constraint solving techniques is currently being processed.

## 5.3. Collaborative Tools

**Participants:** Abdessamad Imine, Asma Cherif.

The collaborative tools is a prototype-set to manage collaborative works on shared documents using flexible access control models. These tools have been developed in order to validate and evaluate our approach on combining collaborative edition with optimistic access control.

- **P2PEdit.** This prototype is implemented in Java and supports the collaborative editing of HTML pages and it is deployed on P2P JXTA platform[3]. In our prototype, a user can create a HTML page from scratch by opening a new collaboration group. Other users (peers) may join the group to participate in HTML page editing, as they may leave this group at any time. Each user can

[3] http://www.sun.com/software/jxta/

dynamically add and remove different authorizations for accessing to the shared document according the contribution and the competence of users participating in the group. Using JXTA platform, users exchange their operations in real-time in order to support WYSIWIS (What You See Is What I See) principle. Furthermore, the shared HTML document and its authorization policy are replicated at the local memory of each user. To deal with latency and dynamic access changes, an optimistic access control technique is used where enforcement of authorizations is retroactive.

- **P2PCalendar.** To extend our collaboration and access control models to mobile devices, we implemented a shared calendar on iPhone OS which is decentralized and scalable (i.e. it can be used over both P2P and ad-hoc networks). This application aims to make a collaborative calendar where users can simultaneously modify events (or appointements) and control access on events. The access rights are determined by the owner of an event. The owner decides who is allowed to access the event and what privileges they have. Likewise to our previous tool, the calendar and its authorization policy are replicated at every mobile device.

## 5.4. Others Tools

Several software tools described in previous sections are using tools that we have developed in the past. For instance BZ-TT uses the set constraints solver CLPS. Note that the development of the SMT prover haRVey has been stopped. The successor of haRVey is called veriT and is developed by David Déharbe (UFRN Natal, Brasil) and Pascal Fontaine (Veridis team).

# 6. New Results

## 6.1. Automated Deduction

We develop general techniques which allow us to re-use available tools in order to build a new generation of solvers offering a good trade-off between expressiveness, flexibility, and scalability. We focus on the careful integration of combination techniques and rewriting techniques to design decision procedures for a wide range of verification problems.

### *6.1.1. Decision procedures for data structures combined with theories of arithmetic*

**Participants:** Christophe Ringeissen, Michaël Rusinowitch, Valerio Senni.

We have continued our work on using superposition calculi in connection with combination methods. After our study of the disjoint case [25], we are now focusing on some non-disjoint cases where the shared theories correspond to fragments of arithmetic. In [24], we present some decidability results for the universal fragment of theories modeling data structures and endowed with arithmetic constraints. More precisely, all the theories taken into account extend a theory that constrains the function symbol for the successor. A general decision procedure is obtained, by devising an appropriate calculus based on superposition. Moreover, we derive a decidability result for the combination of the considered theories for data structures and some fragments of arithmetic by applying a general combination schema for theories sharing a common subtheory. The effectiveness of the resulting algorithm is ensured by using the proposed calculus and a careful adaptation of standard methods for reasoning about arithmetic, such as Gauss elimination, Fourier-Motzkin elimination and Groebner bases computation.

### *6.1.2. Extension of algebraic specifications to Java genericity*

**Participants:** Alain Giorgetti, Olga Kouchnarenko, Elena Tushkanova.

The Krakatoa Modeling Language (KML) is a specification language for Java. It is designed to allow algebraic-style specifications, which are more easily discharged by automated theorem provers than program-oriented specifications. A new feature introduced in Java 5 is genericity. We propose [42] extensions to KML for the algebraic specification of generic Java programs. The key features are the introduction of parametricity both for types and for theories and an instantiation relation between theories. Two significant examples illustrate this extension: the specification of the generic method for sorting arrays and the specification of a generic hash map and its use for memoization. We discuss soundness conditions and their verification.

### 6.1.3. Tree Automata and Rewriting

**Participant:** Michaël Rusinowitch.

With Florent Jacquemard (project-team Dahu) we have proposed in [46] a model for XML update primitives of the W3C XQuery Update Facility as parameterized rewriting rules of the form: "insert an unranked tree from a regular tree language $L$ as the first child of a node labeled by $a$". For these rules, we give type inference algorithms, considering types defined by several classes of unranked tree automata. We show that typechecking for arbitrary sequences of XML update primitives can be done in polynomial time when the unranked tree automaton defining the output type is deterministic and complete, and that it is EXPTIME-complete otherwise. We then apply the results to checking the local consistency of a policy, that is, the non-existence of a sequence of authorized update operations starting from a given document that simulates a forbidden update operation.

## 6.2. Security Protocol Verification

The design of cryptographic protocols is error-prone. Without a careful analysis, subtle flaws may be discovered several years after the publication of a protocol, yielding potential harmful attacks. In this context, formal methods have proved their interest for obtaining good security guarantees. Many analysis techniques have been proposed in the litterature [63]. We develop new techniques for richer primitives, wider classes of protocols and higher security guarantees.

### 6.2.1. Modeling complex primitives

**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 [68], we provide a list of such properties and attacks as well as existing formal approaches for analyzing cryptographic protocols under algebraic properties.

Focusing on ground deducibility and static equivalence (checking whether two sequences of messages are indistinguishable to an attacker), we propose a general setting for solving deducibility and indistinguishability for an important class (called monoidal) of these theories. We have also shown that decidability results can be easily combined for any disjoint equational theories: if the deducibility and indistinguishability relations are decidable for two disjoint theories, they are also decidable for their union. These two results are presented in [17].

Encryption “distributing over pairs” is employed in several cryptographic protocols. We have shown that unification is decidable for an equational theory HE specifying such an encryption [12] We have given an algorithm for solving intruder constraints in HE [28] and general intruder constraints in the equational theory ACI [30]. This last result is useful for handling set datastructures and also multiple intruders.

We have defined in [13] a translation from a protocol narration to the sequences of operations to be performed by each protocol role. Unlike previous works, we reduce this compilation process to known decision problems from formal protocol verification. This allows one to define a precise notion of prudent implementation and to reuse results from the literature in order to cover more crypto-primitives. In particular this is a first work showing how to compile protocols parameterised by the algebraic properties of their symbols.

### 6.2.2. Security Properties

**Participants:** Véronique Cortier, Michaël Rusinowitch, Laurent Vigneron.

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 security properties.

Non-repudiation protocols have an important role in many areas where secured transactions with proofs of participation are necessary. Formal methods are clever and without error, therefore using them for verifying such protocols is crucial. In this purpose, in collaboration with F. Klay (France Telecom R&D), we have shown how to partially represent non-repudiation as a combination of authentications, and also defined a new method, based on the handling of the knowledge of protocol participants. This last method has been implemented in the AVISPA Tool, and used for analyzing several protocols. In particular, it has been used with L. Jing (Sun Yat-Sen University, China) for defining and analyzing a non-repudiation protocol for which there is no assumption of existence of resilient channels between the TTP and each protocol participant [22].

Revisiting and extending the NP-complete decision procedure for a bounded number of sessions developped by Hubert Comon-Lundh, we show how to decide several new properties such as the non-existence of key-cycles (required by recent works relating computational and symbolic models), authentication-like properties and the decidability of a significant fragment of protocols with timestamps [16].

Observational equivalence is a crucial notion for specifying security properties such as anonymity or secrecy of a ballot in vote protocols. For instance, observational equivalence can justify that there is no action of an attacker that makes distinguishable two protocol executions with different identities or vote values. For simple processes without branch nor replication observational equivalence can be reduced to checking whether two symbolic constraints (representing honest agents) are equivalent [67]. We have obtained a new proof that symbolic constraints equivalence is decidable for subterm convergent theories [15]. We believe it is simpler than the first one given by M. Baudet [60].

### 6.2.3. Advanced Classes of Protocols

**Participants:** Mathilde Arnaud, Véronique Cortier, Laurent Vigneron.

New classes of protocols are still emerging and not all can be analysed using existing techniques. We study how to cover the emergent families of security protocols.

*Group Protocols.* Although many works have been dedicated to standard protocols, very few address the more challenging class of group protocols. We have investigated group protocol analysis in a synchronous model, that allows the specification of unbounded sets of agents with related behavior. In collaboration with the project-team Madynes, and in the framework of SAFECAST project on secured group communication system design, we have experienced the use of UML and two complementary verification tools [19]: AVISPA enabled us detecting and fixing security flaws; the TURTLE toolkit enabled us saving development time by eliminating design solutions with inappropriate temporal parameters.

*Securing routing Protocols.* The goal of routing protocols is to construct valid routes between distant nodes in the network. If no security is used, it is possible for an attacker to disorganize the network by maliciously interacting with the routing protocols, yielding invalid routes to be build. That is why secure versions of routing protocols are now developed. We have proposed [29] a new model and an associated decision procedure to check whether a routing protocol can ensure that honest nodes only accept valid routes, even if one of the nodes of the network is compromised. This result has been obtained for a bounded number of sessions, adapting constraint solving techniques.

*Security APIs.* In some systems, it is not possible to trust the host machine on which sensitive codes are executed. In that case, security-critical fragments of a program should be executed on some tamper resistant device (TRD), such as a smartcard, USB security token or hardware security module (HSM). The exchanges between the trusted and the untrusted infrastructures are ensured by special kind of API (Application Programming Interface), that are called *security APIs*. We have proposed new techniques for formally analyze APIs.

### 6.2.4. Securely Composing Protocols

**Participants:** Stefan Ciobaca, Véronique Cortier.

Protocols are often built in a modular way. For example, authentication protocols may assume pre-distributed keys or may assume secure channel. However, when an authentication protocol has been proved secure assuming pre-distributed keys, there is absolutely no guarantee that it remains secure when executing a real protocol for distributing the keys. How the security of these protocols can be combined is an important issue that is studied in [38]. More precisely, we show how protocols sharing data can be safely interleaved, provided that they use disjoint primitives or that each common primitive contains some tag identifying each protocol, like e.g. the name of the protocol. As a sub-result, we provide sufficient and simple conditions for composing key distribution protocols with any protocol using secure channels or pre-distributed keys.

### *6.2.5. Soundness of the Dolev-Yao Model*

**Participant:** Véronique Cortier.

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.

A recent line of research consists in identifying cases where 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. We have proposed a survey [18] of the results obtained so far.

### *6.2.6. Safe and Efficient Strategies for Updating Firewall Policies*

**Participants:** Abdessamad Imine, Michaël Rusinowitch.

The large size and complexity of modern networks result in large and complex firewall policies. Two policy editing languages, Type I and Type II, are generally used to update the firewall policies. Due to intervening nature of firewall rules, correct configuration and *deployment* of large policies is a difficult and error-prone task. We have shown that some recently proposed deployment algorithms in the network security contain serious flaws [27]. Then we have defined a notion of safe deployment strategies. We have provided linear algorithms for Type I safe deployment and an approximatively linear and safe algorithm for Type II.

## 6.3. Model-based Verification

We have investigated extensions of regular model-checking to new classes of rewrite relations on trees. We have studied specification and proof of modular imperative programs.

### *6.3.1. Safety Verification Techniques with Regular Fixpoint Computations*

**Participants:** Roméo Courbis, Pierre-Cyrille Héam, Olga Kouchnarenko.

Term rewriting systems are now commonly used as a modelling language for programs or systems. On those rewriting based models, reachability analysis, i.e. proving or disproving that a given term is reachable from a set of input terms, provides an efficient verification technique. Many recent works have shown the relevance of regular approximation techniques to tackle in practice undecidable reachability problems.

We propose in [56], to exploit rewriting approximations developped in [71] for analysing properties of CCS specifications (without renaming). The approach has been implemented and used to verify properties of the Alternating Bit Protocol and of hardware components specifications expressed as CCS processes.

### *6.3.2. Random Generation of Tree Automata*

**Participants:** Pierre-Cyrille Héam, Vincent Hugot, Olga Kouchnarenko.

Developing new algorithms and heuristics raises crucial evaluation issues, as improved worst-case complexity upper-bounds do not always transcribe into clear practical gains. A suite for software performance evaluation can usually gather three types of entries: benchmarks, hard instance and random inputs, that deliver average complexity estimations, for which the catch resides in obtaining a meaningful random distribution (for instance a uniform random distribution).

We presented in [73] a general rejection algorithm that uniformly generates sequential letter-to-letter transducers up to the isomorphism. We tailor this general scheme to randomly generate deterministic tree walking automata and deterministic top-down tree automata. In [21] we extend this approach by providing a new generation feature to fix both the number of states and the number of transitions. The generation is still uniform, up to isomorphism, and can be performed in polynomial time. In [54] we investigate how to generate non-deterministic tree automata with constraints in order to evaluate the performance of algorithms for the emptiness problem. Moreover, we have continued the development of an easy-to-use prototype dedicated to the random generation of recursive data structure for testing [72].

### 6.3.3. Tree Automata with Constraints

**Participants:** Pierre-Cyrille Héam, Vincent Hugot, Olga Kouchnarenko.

Tree automata with constraints are widely used to tackle data base algorithmic problems, particularly to analyse queries over XML documents. The model of Tree Automata with Global Constraints (TAGED) is a model introduced in 2009 for these purposes. The membership problem for TAGED is known to be NP-complete. In [43] an efficient SAT-based approach for this problem is proposed, with very encouraging experimentations.

We are currently working on developing efficient algorithms for the emptiness problem for positive TAGED. In order to evaluate their performances, we have developed in [54] a random generator of hard instances for this problem.

### 6.3.4. Model-Checking Optimistic Replication Algorithms

**Participant:** Abdessamad Imine.

We work with Hanifa Boucheneb (Professor at Ecole Polytechnique de Montréal, Canada) on automatic verification of optimistic replication algorithms supporting collaborative edition. In this work, we propose a symbolic model-checking technique to verify that an Operational Transformation (OT) algorithm ensures replicas convergence [32]. The shared objects are abstracted and their update operations are handled symbolically using difference bound matrices (DBMs) and neither the shared object size nor the update operations parameter sizes are fixed. Our approach provides symbolic counterexamples in case the convergence property is not satisfied. However, we cannot prove automatically that an OT algorithm ensures convergence for an arbitrary number of sites and operations.

### 6.3.5. Towards Regular Model-Checking for Pictures

**Participant:** Alain Giorgetti.

We have participated to the ANR ‘Smart Surface’ project whose aim is the realization of an active surface to automatically position and convey micro-items. This new application has motivated us to study regular model-checking (RMC) for pictures.

Let us recall that the RMC paradigm consists in representing infinite sets of configurations of a system by recognizable languages, and developing meta-transitions which can compute infinite sets of successors in one step. Unfortunately, a necessary property for RMC is missing in the class of recognizable 2D languages, namely decidability of the inclusion problem. This led us to seek sufficient conditions to decide inclusion. We have studied [53] the notion of simulation over the class of two-dimensional On-line Tessellation Automata (2OTA). This class of automata accepts the class of recognizable 2D languages, considered as the natural extension of classical regular word languages to the 2D case. We have proved that simulation over 2OTA implies language inclusion. Even if the existence of a simulation relation between two 2OTA is shown to be an NP-complete problem, this is a useful result since the inclusion problem is undecidable in general in this class of languages. Then we have proved the existence of a unique maximal autosimulation relation in a given 2OTA and the existence of a unique minimal 2OTA which is simulation equivalent to this given 2OTA, both computable in polynomial time.

## 6.4. Model-based Testing

Our research in Model-Based Testing (MBT) aims to extend the coverage of tests. The coverage refers to two artefacts: model and scenario. The test generation uses symbolic animation of models [51] by dedicated constraints or SMT solvers.

### *6.4.1. Automated Test Generation from Behavioral Models*

**Participants:** Fabrice Bouquet, Pierre-Christophe Bué, Kalou Cabrera, Jérome Cantenot, Frédéric Dadeau, Stéphane Debricon, Elizabeta Fourneret, Adrien de Kermadec, Jonathan Lasalle.

We have introduced an original model-based testing approach that takes an UML behavioural view of the system under testing and automatically generates test cases and executable test scripts according to model coverage criteria. We have extended this result to SysML specifications for validating embedded systems [26].

We are working on improving test generation in two directions:

The first direction is based on the preliminary computation of an abstraction of the model. We have experimented two techniques for automatically computing a symbolic transition system representing an abstraction of a behavioral model. First, we use a machine learning algorithm (à la Angluin) that is combined with model animation [35]. Second, we have experimented the use of behavioral decomposition of the model operation to compute the abstraction state, whereas transitions feasibility is computed using constraint solvers [34]. In both cases, the abstraction is used to produce test cases built according to state/transition coverage criteria.

The second direction exploits the evolution of requirements to classify test sequences, and precisely target the parts of the system impacted by this evolution. We have proposed to define the life cycle of a test via three test classes: $(i)$ Regression, used to validate that unimpacted parts of the system did not change, $(ii)$ Evolution, used to validate that impacted parts of the system correctly evolved, and $(iii)$ Stagnation, used to validate that impacted parts of the system did actually evolve. The associated algorithms are under implementation in a dedicated prototype to be used in the SecureChange european project.

### *6.4.2. Scenario-Based Verification and Validation*

**Participants:** Fabrice Bouquet, Pierre-Christophe Bué, Kalou Cabrera, Frédéric Dadeau, Elizabeta Fourneret, Adrien de Kermadec.

Test scenarios represent an abstract test case specification that aims at guiding the model animation in order to produce relevant test cases. Contrary to the previous section, this technique is not fully automated since it requires the user to design the scenario, in addition to the model.

In the context of ANR TASCCC project, we are investigating the automation of test generation from Security Functional Requirements (SFR), as defined in the Common Criteria terminology. SFRs represent security functions that have to be assessed during the validation phase of security products (in the project, the Global Platform, an operating system for last-generation smart cards). To achieve that, we are working on the definition of security property description patterns, to which a given set of SFRs can be related. These properties are used to automatically generate test scenarios that produce model based test cases. The traceability, ensured all along the testing process, makes it possible to provide evidences of the coverage of the SFR by the tests, required by the Common Criteria to reach the highest Evaluation Assurance Levels.

Also, we have experimented the use of scenarios to compute an abstraction of a model [48], [33]. This abstraction can be used in two ways: to evaluate the coverage of test sequences, and to compute test sequences themselves.

In the context of the SecureChange project, we also investigate the evolution of test scenarios. As the system evolves, the model evolves, and the associated test scenarios may also evolve. We are currently extending the tests generation and management of system evolutions to ensure the preservation of the security.

### *6.4.3. Mutation-based Testing of Security Protocols*

**Participants:** Frédéric Dadeau, Pierre-Cyrille Héam.

Verification of security protocols models is an important issue. Nevertheless, the verification reasons on a model of the protocol, and does not consider its concrete implementation. While representing a safe model, the protocol may be incorrectly implemented, leading to security flaws when it is deployed. We have proposed a model-based approach for testing security protocols implementations. This technique relies on the use of mutations of an original protocol, proved to be correct, for injecting realistic errors that may occur during the protocol implementation (e.g. re-use of existing keys, partial checking of received messages, incorrect formatting of sent messages, use of exponential/xor encryption, etc.). Mutations that lead to security flaws are used to build test cases, which are defined as a sequence of messages representing the behavior of the intruder and leads to the leaking of a secret. We have applied our technique on protocols designed in HLPSL, and implemented a protocol mutation tool that performs the mutations. The mutants are then analyzed by the CL-Atse [76] front-end of the AVISPA toolset [57]. Experiments show the relevance of the proposed mutation operators and the efficiency of the CL-Atse tool to conclude on the vulnerability of a protocol and produce an attack trace that can be used as a test case for implementations.

### 6.4.4. Model Validation

**Participants:** Pierre-Christophe Bué, Fabrice Bouquet, Frédéric Dadeau, Adrien de Kermadec.

In model-based testing the model design is a complex activity that falls to the test engineer. The model validation is mainly done by animation to validate the model behavior and check that it corresponds to the informal requirements. We have proposed to define and assess the quality of B models in order to provide an automated feedback on a model by performing systematic checks on its content. We define and classify classes of automatic verification steps that help the modeller in checking whether his model is well-written or not. From a behavioral model, verification conditions are automatically computed and discharged using a dedicated tool. This technique has been adapted to B abstract machines, and is implemented within a tool interfaced with a constraint solver that is able to find counter-examples to invalid verification conditions [39]. In addition, we have designed an abstraction technique that makes it possible to extract, for a behavioral model, a graphical representation as a labeled transition system [34].

### 6.4.5. Combination of Static Analysis and Test Generation

**Participant:** Alain Giorgetti.

We participate to the design of original combinations of static analysis and structural program testing for C program debugging. We have presented a prototype [36] called SANTE (Static ANalysis and TEsting). It calls a static analysis tool (Frama-C) which generates alarms when it cannot ensure the absence of run-time errors. Then these alarms guide a structural test generation tool (PathCrawler) trying to confirm alarms by activating bugs on some test cases. Experiments on real-life software show that this combination can outperform the use of each technique independently.

## 6.5. Verification for Service Oriented Computing

We have investigated several specific verification problems related to the composition of services including security issues and quality of service.

### 6.5.1. Towards An Automatic Analysis of Web Services Security

**Participants:** Tigran Avanesov, Mohamed Anis Mekki, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

Automatic composition of web services is a challenging task. Many works have considered simplified automata models that abstract away from the structure of messages exchanged by the services. For the domain of secured services (using e.g. digital signing or timestamping) we propose a novel approach to automated orchestration of services under security constraints. Given a community of services and a goal service, we reduce the problem of composing the goal from services in the community to a security problem where an intruder should intercept and redirect messages from the service community and a client service till reaching a satisfying state [66] [30]. This work has been pursued in the context of AVANTSSAR and NESSOS FP7 projects.

### *6.5.2. Composition of Web Services*

**Participants:** Christophe Ringeissen, Laurent Vigneron.

In collaboration with Olivier Perrin (Score team) and Eric Monfroy (UTFSM Valparaíso, Chile), we are working on applying constraint programming techniques to the composition problem. Our approach consists in instantiating a given abstract representation of a composite Web service by selecting the most appropriate concrete Web services. This instantiation is performed in a distributed manner by analysing the current request, i.e., the solver of each service is solving some constraints at one level, and it forwards the rest of the request (modified by the local solution) to the next services. When a service cannot build part of the composition, a distributed backtrack mechanism enables to change previous solutions. Our event-based distributed framework is described in [55].

### *6.5.3. Composition of Services with Constraints*

**Participants:** Pierre-Cyrille Héam, Olga Kouchnarenko.

In [31], we focus on the composition of Web services with constraints. The originality of our approach consists in modeling the services by Boolean automata, i.e. finite automata extended with parametric Boolean conditions. We give a theoretical analysis of three service composition problems – the Valuation Decision problem, the Boolean Formula Decision problem, and the Boolean Formula Synthesis problem. New complexity results are established for these problems when considering both simulation-based and trace-based relations between automata. To go further, we have been studying the mediator decision problem.

In addition, the substitutivity problem for component-/service-based systems has been studied when considering extra-functional properties, like QoS. For services modeled by weighted automata, in [44], [20] four notions of simulation-based substitutivity managing QoS aspects are proposed, and related complexity issues on weighted automata are investigated. The substitutivity problem has been shown undecidable in general for bisimulation equivalence, but some decidable classes–important in practice–have been defined.

### *6.5.4. Controlling Access in Distributed Collaborative Editors*

**Participants:** Asma Cherif, Abdessamad Imine.

We propose an access control model where a group of users can define access rights on a set of shared objects [37]. This model has been implemented as a middleware for collaborative editing systems based on logging mechanism where both the shared document and the access control policy are replicated at each collaborating site. It is difficult to manage the interleaving between document updates and policy administration which may lead to security holes. To deal with latency and dynamic access rights, we apply an optimistic access control technique in such a way that enforcement of authorizations is retroactive. A performance analysis shows the algorithm scales. We plan to extend our model to support delegation.

Since our access control model is based on logs to ensure convergence between all copies of shared objects and policies, we propose a garbage collection mechanism in order to reuse this model on mobile devices (e.g. iPhone) [47] with low storage capacities and high communication delays. Our solution consists in capturing a global view of the state of each log through the exchange of garbage messages: when all users have received all operations and thus have the same global view, their logs are cleaned.

# 7. Contracts and Grants with Industry

## 7.1. 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 (national and international) projects or with a new transfer protocol. According to the law of innovation, F. Bouquet is scientific consultant of LEIRIOS Technologies.

## 7.2. European Projects

- AVANTSSAR — *Automated validation of trust and security of service-oriented architectures*. STREP Project funded under 7th FP (Seventh Framework Program) Research area: ICT-2007.1.4 Secure, dependable and trusted infrastructures. The coordinator is the University of Verona (Italy) and Cassis is one of the 10 partners. AVANTSSAR aims to propose a rigorous technology for the formal specification and "Automated VAlidatioN of Trust and Security of Service-oriented ARchitectures". This technology will be automated into an integrated toolset, the AVANTSSAR Validation Platform, tuned on relevant industrial case studies.
- SecureChange[4] is funded under the 7th FP (Seventh Framework Program) Research area: ICT-2007.8.6: ICT forever yours. The project will develop processes and tools that support design techniques for evolution, testing, verification, re-configuration and local analysis of evolving software. Our focus is on mobile devices and homes, which offer both great research challenges and long-term business opportunities. The project is lead by Fabio Massacci (University of Trento, Italy) and it has started in February 2009 for a period of 36 months. Cassis is leader of the 7th workpackage (Testing). The local coordinator is Fabrice Bouquet.
- Nessos is a Network of Excellence on Engineering Secure Future Internet Software Services and Systems in FP7-ICT (starting in October 2010 for a period of 42 months). Nessos has 12 partners and aims at constituting and integrating a long lasting research community on engineering secure software-based services and systems. Partner INRIA is involved through project-teams Arles, Triskell and Cassis. Cassis will focus on developping tools for service security verification and testing tasks.

# 8. Other Grants and Activities

## 8.1. International Grants

- Project INRIA-CONICYT (Chile), CoreWeb — *Constraint Reasoning for the Composition of Web Services*. The coordinators are Eric Monfroy (UTFSM Valparaíso, Chile) and Michaël Rusinowitch.
- Associate Team INRIA (with UTFSM Valparaíso, Chile), VanaWeb — *Hybrid and autonomous constraint solving and applications to composition problems for the Web*. The coordinators are Carlos Castro (UTFSM Valparaíso, Chile) and Christophe Ringeissen. On the french side, VanaWeb also involves the Score team, the project-team Pareo and faculty members from the universities of Angers (Frédéric Saubion) and Caen (Arnaud Lallouet).
- French-Tunisian project on *Security Policies and Configurations of Firewalls: Compilation and Automated Verification*. We collaborate with SupCom Tunis and the INRIA project-team Dahu in the context of STIC-Tunisia.

## 8.2. National Grants

- ARA SETI RAVAJ [5] — *"Rewriting and Approximations for Java Applications Verification"*, duration: 42 months, started on January 2007. The goal of this project is to analyse MIdlets – Java programs designed for mobile devices like cell phones or PDA. In addition to classical proof tools of rewriting, we propose to use approximations of reachable terms. There are three academics partners: the INRIA project-teams Celtique and Pareo, and LIFC/Besançon; and an industrial: France Telecom R&D. The local coordinator is Olga Kouchnarenko.

[4] http://www.securechange.eu

[5] http://www.irisa.fr/lande/genet/RAVAJ/index.html

- ANR SESUR AVOTÉ—*Formal Analysis of Electronic-Voting protocols*, duration: 4 years, started in January 2008. Electronic voting promises the possibility of a convenient, efficient and secure facility for recording and tallying votes. However, the convenience of electronic elections comes with a risk of large-scale fraud. The AVOTÉ project aims at proposing techniques for formally analyzing e-voting protocols. Cassis is the coordinator of the project. Partners are: France Telecom Lannion, LSV Cachan, Verimag Grenoble.
- ANR program "Systèmes interactifs et robotique"— *Smart Surface*, coordinated by AS2M (Automatique et Systèmes Micro-Mécatroniques) department at the FEMTO-ST (Franche-Comté Électronique Mécanique Thermique et Optique - Sciences et Technologies) institute (UMR 6174). This project started in July 2007 for three years. The Cassis participant is Alain Giorgetti.
- ANR DECERT — *Deduction and Certification*, coordinated by Thomas Jensen (IRISA). This project focuses on the design of decision procedures, in particular for fragments of arithmetic, and their integration into larger verification systems, including skeptical proof assistants. Partners are: IRISA Rennes, LRI Orsay, INRIA Sophia, Systerel and CEA. From INRIA Nancy, the teams Veridis and Cassis are involved. This project started in January 2009 for three years.
- ANR TASCCC *Test Automatic basé sur des Scenarios et Critères Communs – Automated Testing based on Scenarios and Common Criteria*, duration: 3 years, starting in December 2009. The project aims at completing the model-based testing process initiated in the POSE project, using scenarios to specify the test cases that have to be generated by model animation. The goal is here to provide an automated mean for generating the scenarios from a given set of properties. The overall objective is to ease the Common Criteria evaluation of secure softwares. Partners: Gemalto (leader), LIG, LIFC, Supelec, Smartesting, and Serma Technologies. The local coordinator is Frédéric Dadeau.
- ANR STREAMS *Solution for Peer-to-peer Real-Time Social Web*, duration: 3 years, starting in October 2010. STREAMS project proposes to design peer-to-peer solutions that offer underlying services required by real-time social web applications and that eliminate the disadvantages of centralised architectures. There exists a tension between sharing data with friends in a social network deployed in an open peer-to-peer network and ensuring privacy. One of the most challenging issues in social applications is how to balance collaboration with access control to shared objects. STREAMS project aims at providing theoretical solutions to these challenges as well as practical experimentations. Partners are: LORIA Score team (leader), INRIA project-teams Regal, Asap, Cassis, and XWiki.
- ANR FREC *Frontiers of recognizability*, duration: 4 years, starting in October 2010. The goal of this project is to be a driving force behind the extension of the algebraic theory of regular languages made possible by recent advances. Four directions will be investigated: tree languages, $\lambda$-terms, automata with counters, algebraic and topological tools. Partners are LABRI (leader), LIAFA (University Paris 7). Pierre-Cyrille Héam is a member of this project, attached to Paris 7 for administrative facilities.
- FCE Vetess [6] — We are working with the university of Haute Alsace, SMARTESTING Technologies and PSA Citroën. The project is labelled by the "Pôle de Compétitivité Véhicule du Futur" and funded by the "Fonds de Compétitivité des Entreprises", an inter-ministry grant. It aims at verifying embedded systems vehicles by automatic model-based tests generation. The duration of the project is 18 months and started in September 2008 ending August 31th 2010. The local coordinator is Fabrice Bouquet.
- DGA RIE Secure Test project, duration: 18 months, started in February 2009. The project provides a specific environment to verify of cryptographic components (hardware or software) with an Model-Based Testing approach. The method help the test team to evaluation DGA to product a test refential. Partners are: DGA CELAR, Smartesting (coordinator), Telecom Bretagne. The local coordinator is Fabrice Bouquet.

[6] http://lifc.univ-fcomte.fr/vetess

- Collaborative Research Initiative INRIA, ARC ACCESS. This project is concerned with the security and access control for Web data exchange, in the context of Web applications and Web services. We aim at defining automatic verification methods for checking properties of access control policies (ACP) for XML, like consistency or secrecy. Partners are: INRIA project-teams Dahu, Mostrare and Cassis.

## 8.3. International Collaborations

- 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 [7]. This cooperation is supported by the France-New-Zealand scientific program.
- In the area of business applications, we have been working on the may-/must semantics of coloured work-flow Petri nets with the Information System group of Professor W. van der Aalst from the Technical University of Eindhoven. This cooperation is supported in part by the NWO scientific program (The Netherlands).

## 8.4. Individual Involvement

*F. Bouquet:* vice-head of LIFC laboratory; PC member of Modevva'10 (Model-Driven Engineering, Verification, And Validation), MBTEST 2010 and QuoMBaT 2010. President of the MCF selection committee of section 27 of UFC. Expert for Luxembourg National Research Fund.

*V. Cortier:* coordinator of the ANR SESUR AVOTÉ (started in January 2008); co-chair of FCS-PrivMod 2010 (Workshop on Foundations of Security and Privacy, affiliated with LICS 2010 and CSF 2010); co-chair of SecCo 2010 (Security in Concurrency), affiliated with ConCur 2010; PC member of FSTTCS 2010 (IARCS Conference on Foundations of Software Technology and Theoretical Computer Science), CCS 2010 (17th ACM Conference on Computer and Communication Security), ESORICS 2010 (15th European Symposium on Research in Computer Security), LICS 2010 (24th IEEE Symposium on Logic in Computer Science), MOVEP 2010 (9th School on MOdelling and VErifying parallel Processes), PLAS 2010 (5th ACM SIGPLAN Workshop on Programming Languages and Analysis for Security), IFIP TCS 2010, track B (6th IFIP International Conference on Theoretical Computer Science); member of the CS (Comité de sélection) for the 2010 CNRS - Versailles University chair, member of the Evaluation Committee of the INRIA since September 2008.

*F. Dadeau:* PC member of the 2nd International Workshop on Constraints in Software Testing, Verification and Analysis (CSTVA'2010), affiliated with ICST'2010. Editorial committee of the Model-Based Testing for Embedded Systems book.

*A. Giorgetti:* Editorial committee member of *Techniques et Science Informatique (TSI)*. PC member of the 1st workshop on hardware and software implementation and control of distributed MEMS (dMEMS'10).

*A. Imine:* PC member of DEXA'2011 (the 22nd International Conference on Database and Expert Systems Applications) and COSI'2010 (Colloque sur l'Optimisation et les Systèmes d'Information).

*O. Kouchnarenko:* director of the research team *VESONTIO* (former TFC) of the *Laboratoire d'informatique de Franche Comté (LIFC)*; PC member of "*International Workshop on Abstractions for Petri Nets and Other Models of Concurrency*", APNOC'10. Director of the "Licence Informatique 2008-2012" in the University of Franche-Comté.

*C. Ringeissen*: PC member of FroCoS'11 (Frontiers of Combining Systems) and IJCAR 2010 (the 5th International Joint Conference on Automated Reasoning).

[7] http://www.cs.waikato.ac.nz/Research/fm/index.html

*M. Rusinowitch:* member of the IFIP Working Group 1.6 (Rewriting), co-organizer of Workshop on Formal Methods for Web Data Trust and Security, Nancy, October 11th 2010. PC member of ASIACCS 2010 (5th ACM Symposium on Information, Computer and Communications Security), STM'10 (6th International Workshop on Security and Trust Management), CRiSIS 2010 (5th International Conference on Risks and Security of Internet and Systems), SecCo'10 (8th International Workshop on Security Issues in Concurrency), SCSS 2010 (Third International Workshop on Symbolic Computation in Software Science), SoICT2010 (Symposium on Information and Communication Technology), SecDay2010 (2010 Grande Région Security and Reliability Day). Member of the selection committees: INRIA Rocquencourt (CR position), Rennes University/INRIA (junior chair), UHP Nancy (Full Professor). Vice-président of Project Committee at INRIA Grand Est since October 2009.

*L. Vigneron:* PC member of UNIF'2010; Member of the FTP steering committee; Member of the IFIP Working Group 1.6 on Rewriting; Webmaster of the site Rewriting Home Page and of the RTA conference Web site.

We are involved in several lectures of the "Master Informatique" of the universities of Nancy. L. Vigneron is in charge of the lectures on *Algorithmic verification* and *Security of communications*. V. Cortier is in charge of the lecture on *Theory of the security*. C. Ringeissen is in charge of the lecture on *Decision procedures and program verification*.

## 8.5. Visits of Foreign Researchers

*Adel Bouhoula* (SupCom Tunis, Tunisie) has visited Cassis (November 29 - December 2) to work on firewalls policies.

*John Mullins* (Ecole Polytechnique de Montréal, Canada) has visited Cassis/LIFC as a Franche-Comté University invited professor to work on substitutivity/composition problems for probabilistic weighted automata (May 31 - July 5).

*Bogdan Warinschi* (University of Bristol, UK) has visited LORIA to work on combination techniques for soundness results of symbolic model (November 22 - 30).

## 8.6. Visits of Team Members

*Olga Kouchnarenko* has visited Natalia Sidorova (Eindhoven Univ. of Technologies, Netherlands) to work on the may-/must-semantics of coloured workflow Petri nets and on their property preservation (July 13 - 27).

*Christophe Ringeissen* and *Laurent Vigneron* have visited Carlos Castro and Eric Monfroy (UTFSM Valparaíso, Chile) to work in the context of the associate team INRIA VanaWeb (January 13 - 22 and October 24 - November 6).

# 9. Dissemination

## 9.1. Committees

*F. Bouquet* is referee for the theses of Hakim Belhoaouri (Paris 6) and Zhe Chen (University Paul Sabatier of Toulouse).

*A. Giorgetti* is examiner for the thesis of Samuel Vidal, LIFL (University of Lille).

*P.-C. Héam* is examiner for the thesis of David Martins, LIFC (University of Franche-Comté).

*O. Kouchnarenko* is referee for the theses of Florence Charreteur-Schadle (Rennes I), Marwa El Houri (University Paul Sabatier of Toulouse), Manuel Garnacho (Grenoble I), and examiner for the theses of Inès Mouahker (University of Nancy 2 and University of Tunis), and Mohamed Faïcal Abouzaid (Ecole Polytechnique of Montréal).

*M. Rusinowitch* is referee for the theses of Lisa Allali (Ecole Polytechnique), Amr Helmy (INP Grenoble), Nizar Kheir (Rennes I), and chair of the thesis committee of Caroline Lavecchia (University of Nancy 2).

*L. Vigneron* is examiner for the thesis of Marwa El Houri (University Paul Sabatier of Toulouse).

## 9.2. Seminars, Workshops, and Conferences

We were invited to give the following talks.

V. CORTIER, Invited talk at VERIFY 2010, 6th International Verification Workshop, Edinburgh, UK, July 20, 2010. Tutorial at MOVEP 2010, Summer school on modeling and verifying parallel processes, July 2, 2010, Aachen, Germany. Lectures at FOSAD 2010, International School on Foundations of Security Analysis and Design, Bertinoro, Italy. September 6-7, 2010. Seminar at Luxembourg, October 5, 2010. Talk in the joint seminar of the Saarbruecken Computer Science Cluster (Universitaet des Saarlandes, DFKI, MPI Informatics, MPI Software Systems, Germany), November 17, 2010.

C. RINGEISSEN, Seminar on Combining Satisfiability Procedures for Unions of Theories Sharing Fragments of Arithmetic, April 30, 2010, MPII Saarbruecken, Germany.

M. RUSINOWITCH, Invited talk at SecRet 2010, 5th International Workshop on Security and Rewriting Techniques, Valencia, Spain, June 18, 2010.

L. VIGNERON, Seminar on “Verification of infinite state systems: application to the analysis of cryptographic protocols”, ENS Lyon, September 21, 2010.

# 10. Bibliography

## Major publications by the team in recent years

[1] M. ABADI, V. CORTIER. *Deciding knowledge in security protocols under equational theories*, in "Theoretical Computer Science", November 2006, vol. 387, n$^{o}$ 1-2, p. 2-32.

[2] A. ARMANDO, D. BASIN, Y. BOICHUT, Y. CHEVALIER, L. COMPAGNA, J. CUELLAR, P. HANKES DRIELSMA, P.-C. HÉAM, O. KOUCHNARENKO, J. MANTOVANI, S. MÖDERSHEIM, D. VON OHEIMB, M. RUSINOWITCH, J. SANTOS SANTIAGO, M. TURUANI, L. VIGANÒ, L. VIGNERON. *The AVISPA Tool for the automated validation of internet security protocols and applications*, in "17th International Conference on Computer Aided Verification, CAV'2005", Edinburgh, Scotland, Lecture Notes in Computer Science, Springer, 2005, vol. 3576, p. 281-285.

[3] 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)", June 2003, vol. 183, n$^{o}$ 2, p. 140–164.

[4] M. BAUDET, V. CORTIER, S. KREMER. *Computationally Sound Implementations of Equational Theories against Passive Adversaries*, in "Information and Computation", April 2009, vol. 207, n$^{o}$ 4, p. 496-520.

[5] Y. BOICHUT, R. COURBIS, P.-C. HEAM, O. KOUCHNARENKO. *Finer is better: Abstraction Refinement for Rewriting Approximations*, in "19th International Conference on Rewriting Techniques and Applications - RTA'2008", Hagenberg, Austria, A. VORONKOV (editor), Lecture Notes in Computer Science, Springer, 2008, vol. 5117, p. 48-62.

[6] 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", August 2004, vol. 6, n$^{o}$ 2, p. 143–157.

[7] Y. CHEVALIER, R. KUESTERS, M. RUSINOWITCH, M. TURUANI. *Complexity results for security protocols with Diffie-Hellman exponentiation and commuting public key encryption*, in "ACM Transactions on Computational Logic (TOCL)", 2008, vol. 9, Article 24.

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

[9] A. Giorgetti, J. Groslambert, J. Julliand, O. Kouchnarenko. *Verification of Class Liveness Properties with Java Modelling Language*, in "IET Software", 2008, vol. 2, n$^{o}$ 6, p. 500-514.

[10] E. Nicolini, C. Ringeissen, M. Rusinowitch. *Combinable Extensions of Abelian Groups*, in "Proc. of 22nd International Conference on Automated Deduction, CADE-22", Montreal, Canada, R. Schmidt (editor), Lecture Notes in Artificial Intelligence, Springer, 2009, vol. 5663, p. 51–66.

## Publications of the year

### Articles in International Peer-Reviewed Journal

[11] T. Abbes, A. Bouhoula, M. Rusinowitch. *Efficient Decision Tree for Protocol Analysis in Intrusion Detection*, in "International Journal of Security and Networks", 2010, http://hal.inria.fr/inria-00528201.

[12] S. Anantharaman, H. Lin, C. Lynch, P. Narendran, M. Rusinowitch. *Unification modulo Homomorphic Encryption*, in "Journal of Automated Reasoning", 2010, vol. (To appear), http://hal.inria.fr/inria-00502850.

[13] Y. Chevalier, M. Rusinowitch. *Compiling and securing cryptographic protocols*, in "Information Processing Letters", 2010, vol. 110, n$^{o}$ 3, p. 116–122, http://hal.inria.fr/inria-00527633.

[14] Y. Chevalier, M. Rusinowitch. *Decidability of Equivalence of Symbolic Derivations*, in "Journal of Automated Reasoning", 2010, http://hal.inria.fr/inria-00527630.

[15] Y. Chevalier, M. Rusinowitch. *Symbolic Protocol Analysis in the Union of Disjoint Intruder Theories: Combining Decision Procedures*, in "Theoretical Computer Science", 2010, vol. 411, n$^{o}$ 10, p. 1261-1282, http://hal.inria.fr/inria-00455290.

[16] H. Comon-Lundh, V. Cortier, E. Zalinescu. *Deciding security properties for cryptographic protocols. Application to key cycles*, in "ACM Transactions on Computational Logic", 2010, vol. 11, n$^{o}$ 2, http://hal.inria.fr/inria-00525775.

[17] V. Cortier, S. Delaune. *Decidability and combination results for two notions of knowledge in security protocols*, in "Journal of Automated Reasoning", 2010, http://hal.inria.fr/inria-00525778.

[18] V. Cortier, S. Kremer, B. Warinschi. *A Survey of Symbolic Methods in Computational Analysis of Cryptographic Systems*, in "Journal of Automated Reasoning", 2010, http://hal.inria.fr/inria-00525776.

[19] P. De Saqui-Sannes, T. Villemur, B. Fontan, S. Mota, M. S. Bouassida, N. Chridi, I. Chrisment, L. Vigneron. *Formal Verification of Secure Group Communications Using AVISPA and TURTLE*, in "Innovations in Systems and Software Engineering", 2010, vol. 6, p. 125-133, http://hal.inria.fr/hal-00447682.

[20] P.-C. Heam, O. Kouchnarenko, J. Voinot. *Component Simulation-based Substitutivity Managing QoS and Composition Issues*, in "Science of Computer Programming", 2010, vol. 75, n$^{o}$ 10, p. 898-917, http://hal.inria.fr/inria-00511466.

[21] P.-C. Heam, C. Nicaud, S. Schmitz. *Parametric Random Generation of Deterministic Tree Automata*, in "Theoretical Computer Science", 2010, vol. 411, p. 3469-3480, http://hal.inria.fr/inria-00511450.

[22] J. Liu, L. Vigneron. *Design and Verification of a Non-Repudiation Protocol Based on Receiver-Side Smart Card*, in "IET Information Security", March 2010, vol. 4, n$^{o}$ 1, p. 15-29, http://hal.inria.fr/inria-00426527/en/.

[23] P.-A. Masson, M.-L. Potet, J. Julliand, R. Tissot, F. Bouquet, B. Legeard, E. Jaffuel, B. Chetali, J. Andronick, A. Haddad. *An Access Control Model Based Testing Approach for Smart Card Applications: Results of the POSÉ Project*, in "Journal of Information Assurance and Security", 2010, vol. 5, n$^{o}$ 1, p. 335-351, http://hal.inria.fr/inria-00533220.

[24] E. Nicolini, C. Ringeissen, M. Rusinowitch. *Combining Satisfiability Procedures for Unions of Theories with a Shared Counting Operator*, in "Fundamenta Informaticae", 2010, vol. 103, http://hal.inria.fr/inria-00526683.

[25] D.-K. Tran, C. Ringeissen, S. Ranise, H. Kirchner. *Combination of Convex Theories: Modularity, Deduction Completeness, and Explanation*, in "Journal of Symbolic Computation", Feb 2010, vol. 45, n$^{o}$ 2, p. 261-286, http://hal.inria.fr/inria-00428583.

### Articles in National Peer-Reviewed Journal

[26] F. Fondement, P.-A. Muller, B. Wittmann, F. Ambert, F. Bouquet, J. Lasalle, E. Oudot, F. Peureux, B. Legeard, M. Alter, C. Scherrer. *VETESS : IDM, Test et SysML*, in "Génie logiciel", 2010, n$^{o}$ 93, p. 43–48, http://hal.inria.fr/inria-00533277.

### International Peer-Reviewed Conference/Proceedings

[27] Z. Ahmed, A. Imine, M. Rusinowitch. *Safe and Efficient Strategies for Updating Firewall Policies*, in "7th International Conference on Trust, Privacy & Security in Digital Business - TrustBus 2010", Espagne Bilbao, Springer, Aug 2010, vol. 6264, p. 45-57, The original publication is available at www.springerlink.com, http://hal.inria.fr/inria-00529077.

[28] S. Anantharaman, H. Lin, C. Lynch, P. Narendran, M. Rusinowitch. *Cap Unification: Application to Protocol Security modulo Homomorphic Encryption*, in "5th ACM Symposium on Information, Computer and Communications Security - ASIACCS 2010", Chine Beijing, ACM, Apr 2010, http://hal.inria.fr/inria-00448703.

[29] M. Arnaud, V. Cortier, S. Delaune. *Modeling and Verifying Ad Hoc Routing Protocols*, in "23rd IEEE Computer Security Foundations Symposium - CSF'10", Royaume-Uni Edinburgh, 2010, http://hal.inria.fr/inria-00525779.

[30] T. Avanesov, Y. Chevalier, M. Rusinowitch, M. Turuani. *Satisfiability of General Intruder Constraints with a Set Constructor*, in "The Fifth International Conference on Risks and Security of Internet and Systems - CRiSIS 2010", Canada Montreal, 2010, http://hal.inria.fr/inria-00531025.

[31] P. Balbiani, F. Cheikh, P.-C. Heam, O. Kouchnarenko. *Composition of services with constraints*, in "Formal Aspects of Component Software", Pays-Bas Eindhoven, Jan 2010, vol. 263, p. pp. 31-46, Rapport de recherche CWI, Amsterdam, NL, SEN-E0902, pp. 99-113, http://hal.inria.fr/hal-00429876.

[32] H. Boucheneb, A. Imine, M. Najem. *Symbolic Model-Checking of Optimistic Replication Algorithms*, in "8th International Conference on Integrated Formal Methods - IFM 2010", France Nancy, Springer Berlin / Heidelberg, Oct 2010, vol. 6396, p. 89-104, The original publication is available at www.springerlink.com, http://hal.inria.fr/inria-00524535.

[33] F. Bouquet, P.-C. Bué, J. Julliand, P.-A. Masson. *Test Generation Based on Abstraction and Test Purposes to Complement Structural Tests*, in "A-MOST'10, 6th int. Workshop on Advances in Model Based Testing, in conjunction with ICST'10", France Paris, 2010, p. 54–61, http://hal.inria.fr/inria-00533281.

[34] P.-C. Bué, F. Dadeau, A. De Kermadec, F. Bouquet. *Building a Test-ready Abstraction of a Behavioral Model using CLP*, in "4th International Conference on Tests and Proofs - TAP 2010", Espagne Malaga, Springer-Verlag, Jul 2010, vol. 6143, p. 167-182, The original publication is available at www.springerlink.com, http://hal.inria.fr/inria-00532608.

[35] P.-C. Bué, F. Dadeau, P.-C. Héam. *Model-Based Testing using Symbolic Animation and Machine Learning*, in "2nd Workshop on Constraints in Software Testing, Verification, and Analysis - CSTVA'2010", France Paris, IEEE Press, Apr 2010, http://hal.inria.fr/inria-00532977.

[36] O. Chebaro, N. Kosmatov, A. Giorgetti, J. Julliand. *Combining Static Analysis and Test Generation for C Program Debugging*, in "4th international conference Tests and proofs - TAP'10", Espagne Malaga, Springer-Verlag, 2010, vol. 6143, p. 94–100, The original publication is available at www.springerlink.com, http://hal.inria.fr/inria-00527877.

[37] A. Cherif, A. Imine, M. Rusinowitch. *Optimistic Access Control for Collaborative Editing Systems*, in "Proceedings of the 2011 ACM Symposium on Applied Computing (SAC)", Taichung, Taiwan, March 21-24 2011, to appear.

[38] S. Ciobaca, V. Cortier. *Protocol composition for arbitrary primitives*, in "23rd IEEE Computer Security Foundations Symposium - CSF'10", Royaume-Uni Edinburgh, 2010, p. 322-336, http://hal.inria.fr/inria-00525781.

[39] A. De Kermadec, F. Dadeau, F. Bouquet. *Assessing the Quality of B Models*, in "SEFM'10 - 8th IEEE International Conference on Software Engineering and Formal Methods", Italie Pisa, Sep 2010, http://hal.inria.fr/inria-00532974.

[40] A. Giorgetti. *Guessing a Conjecture in Enumerative Combinatorics and Proving It with a Computer Algebra System*, in "International workshop on Symbolic Computation in Software Science - SCSS'10", Autriche Linz, 2010, p. 5–18, http://hal.inria.fr/inria-00527883.

[41] A. Giorgetti, A. Hammad, B. Tatibouët. *Using SysML for Smart Surface Modeling*, in "dMEMS'10, 1st workshop on design, control and software implementation for distributed MEMS", France Besançon, IEEE, 2010, p. 100–107, http://hal.inria.fr/inria-00525791.

[42] A. Giorgetti, C. Marché, E. Tushkanova, O. Kouchnarenko. *Specifying Generic Java Programs: two case studies*, in "11th International Workshop on Language Descriptions, Tools, and Applications - LDTA'2010", Chypre Paphos, 2010, p. 92–106, http://hal.inria.fr/inria-00525784.

[43] P.-C. HEAM, V. HUGOT, O. KOUCHNARENKO. *SAT Solvers for Queries over Tree Automata with Constraints*, in "2nd Workshop on Constraints in Software Testing, Verification and Analysis - CSTVA'10, joint to ICST'10", France Paris, 2010, http://hal.inria.fr/inria-00523951.

[44] P.-C. HEAM, O. KOUCHNARENKO, J. VOINOT. *Component Simulation-based Substitutivity Managing QoS Aspects*, in "Formal Aspects On Component Softwre - FACS'08", Espagne Malaga, Jan 2010, vol. 260 (2010), p. 109-123, http://hal.inria.fr/inria-00329909.

[45] A. IMINE. *On Coordinating Collaborative Objects*, in "9th International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLASA)", France Paris, Open Publishing Association, Sep 2010, vol. 30, p. 78-92, http://hal.inria.fr/inria-00529071.

[46] F. JACQUEMARD, M. RUSINOWITCH. *Rewrite-based verification of XML updates*, in "12th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming - PPDP'10", Autriche Hagenberg, ACM, Jul 2010, p. 119-130, http://hal.inria.fr/inria-00529620.

[47] M. D. MECHAOUI, A. CHERIF, A. IMINE, F. BENDELLA. *Log Garbage Collector-based Real Time Collaborative Editor for Mobile Devices*, in "6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)", États-Unis Chicago, Oct 2010, http://hal.inria.fr/inria-00529082.

### National Peer-Reviewed Conference/Proceedings

[48] P.-C. BUÉ, J. JULLIAND, P.-A. MASSON, F. BOUQUET. *Associer des techniques de preuve et de résolution de contraintes pour la construction d'abstractions*, in "10èmes Journées Francophones Internationales sur les Approches Formelles dans l'Assistance au Développement de Logiciels - AFADL 2010", France Poitiers, Jun 2010, p. 11-25, http://hal.inria.fr/inria-00533260.

[49] A. CACIULA, R. COURBIS, V. FELEA, P.-C. HEAM, R. IONESCU. *Une approche parallèle et distribuée pour la complétion d'automates d'arbre*, in "10èmes Journées Francophones Internationales sur les Approches Formelles dans l'Assistance au Développement de Logiciels - AFADL 2010", France Poitiers, Jun 2010, vol. 10, 43, http://hal.inria.fr/hal-00530350.

[50] R. COURBIS, P.-C. HEAM, P. JOURDAN, O. KOUCHNARENKO. *Approximations par réécriture pour deux problèmes indécidables*, in "AFADL", France Poitiers, Jun 2010, vol. 10, 7, http://hal.inria.fr/hal-00530341.

### Scientific Books (or Scientific Book chapters)

[51] F. DADEAU, F. PEUREUX, B. LEGEARD, R. TISSOT, J. JULLIAND, P.-A. MASSON, F. BOUQUET. *Test Generation using Symbolic Animation of Models*, in "Model-Based Testing for Embedded Systems", J. ZANDER, I. SCHIEFERDECKER, P. J. MOSTERMAN (editors), CRC Press, 2010, To be published in 2011, http://hal.inria.fr/inria-00532604.

### Research Reports

[52] T. AVANESOV, Y. CHEVALIER, M. RUSINOWITCH, M. TURUANI. *Satisfiability of General Intruder Constraints with and without a Set Constructor*, INRIA, May 2010, RR-7276, http://hal.inria.fr/inria-00480632.

[53] G. CÉCÉ, A. GIORGETTI. *Simulations for a Class of Two-Dimensional Automata*, INRIA, Oct 2010, RR-7425, http://hal.inria.fr/inria-00527077.

[54] P.-C. Heam, V. Hugot, O. Kouchnarenko. *Random Generation of Positive TAGEDs wrt. the Emptiness Problem*, INRIA, Nov 2010, RR-7441, http://hal.inria.fr/inria-00531350.

[55] E. Monfroy, O. Perrin, C. Ringeissen, L. Vigneron. *A Constraint-based Approach to Web Services Provisioning*, INRIA, Oct 2010, RR-7413, http://hal.inria.fr/inria-00524590.

### Other Publications

[56] R. Courbis. *Rewriting Approximations For Properties Verication Over CCS Specifications*, 2010, http://hal.inria.fr/hal-00530351.

## References in notes

[57] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. Von Oheimb, M. Rusinowitch, J. Santos Santiago, L. Vigano, M. Turuani, L. Vigneron. *The AVISPA Tool for the automated validation of internet security protocols and applications*, in "17th International Conference on Computer Aided Verification - CAV 2005", Lecture Notes in Computer Science, Springer, 2005, vol. 3576, p. 281-285.

[58] C. Arora, M. Turuani. *Validating Integrity for the Ephemerizer's Protocol with CL-Atse*, in "Formal to Practical Security: Papers Issued from the 2005-2008 French-Japanese Collaboration", Lecture Notes in Computer Science, Springer, 2009, vol. 5458, p. 21–32.

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

[60] M. Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires*, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/these-baudet.pdf.

[61] 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, Springer-Verlag, 2001, vol. 2021.

[62] 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", 2004, vol. 34, n$^{o}$ 10, p. 915–948.

[63] Y. Boichut, P.-C. Héam, O. Kouchnarenko. *Vérifier automatiquement les protocoles de sécurité*, in "Techniques de l'ingénieur", October 2007, p. RE95-1–RE95-8.

[64] 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", Springer-Verlag, September 2003, vol. 2805, p. 778–795.

[65] 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, Springer, April 2002, vol. 2280, p. 188–204.

[66] Y. Chevalier, M. A. Mekki, M. Rusinowitch. *Automatic Composition of Services with Security Policies*, in "Web Service Composition and Adaptation Workshop (held in conjunction with SCC/SERVICES-2008)", Honolulu États-Unis, IEEE, 2008, p. 529-537 [*DOI :* 10.1109/SERVICES-1.2008.13].

[67] V. Cortier, S. Delaune. *A method for proving observational equivalence*, in "Proceedings of the 22nd IEEE Computer Security Foundations Symposium (CSF'09)", Port Jefferson, NY, USA, IEEE Computer Society Press, July 2009, p. 266-276.

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

[69] J. Dick, A. Faivre. *Automating the Generation and Sequencing of Test Cases from Model-Based Specifications*, in "FME'93: Industrial-Strength Formal Methods", Lecture Notes in Computer Science, Springer-Verlag, April 1993, vol. 670, p. 268–284.

[70] 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://www.wisdom.weizmann.ac.il/~oded/eg83.html.

[71] P.-C. Heam, O. Kouchnarenko, Y. Boichut. *Tree Automata for Detecting Attacks on Protocols with Algebraic Cryptographic Primitives*, in "Joint Proceedings of the 8th, 9th, and 10th International Workshops on Verification of Infinite-State Systems (INFINITY)", Lisbon, Portugal, Electronic Notes in Theoretical Computer Science, 2009, vol. 239, http://hal.inria.fr/inria-00429356/en/.

[72] P.-C. Heam, C. Nicaud. *Seed: an easy to use random generator of recursive data structures for testing*, 2009, http://hal.inria.fr/inria-00528585/PDF/rr-lsv-2009-15.pdf.

[73] P.-C. Héam, C. Nicaud, S. Schmitz. *Random Generation of Deterministic Tree (Walking) Automata*, in "14th International Conference on Implementation and Application of Automata - CIAA 2009 Implementation and Application of Automata", Sydney, Australia, S. Maneth (editor), Springer-Verlag, 2009, vol. 5642, p. 115–124, http://hal.inria.fr/inria-00408316/en/.

[74] B. Legeard, F. Bouquet, P. Natacha. *Industrialiser le test fonctionnel*, Management des systèmes d'information, Dunod, 2009, http://hal.inria.fr/inria-00430538/en/.

[75] N. Liu, Wen-ye. Zhu, Yue-fei. Zhu. *Security Protocol Analysis Based on Rewriting Approximation*, in ". Second International Symposium on Electronic Commerce and Security, ISECS '09", IEEE, 2009, p. 318-322.

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