

Workshop on Formal and Computational Cryptography  
FCC 2008

Abstracts of talks

June 26, 2008  
Carnegie Mellon University, Pittsburgh, PA, USA

## **Program committee**

Bruno Blanchet, co-chair (CNRS, ENS Paris, INRIA, France)

Ran Canetti (IBM Research, USA)

Anupam Datta, co-chair (Carnegie Mellon University, USA)

Cédric Fournet (MSR, Cambridge, UK)

Ralf Küsters (University of Trier, Germany)

Phillip Rogaway (University of California, Davis, USA)

Andre Scedrov (University of Pennsylvania, USA)

Bogdan Warinschi (University of Bristol, UK)

## **Acknowledgments**

This workshop was sponsored by the ARTIST network of excellence and the ANR project ARA SSIA FormaCrypt.

# Program

## 8:55 Welcome

### 9:00-10:00 Invited talk (chair: Bruno Blanchet)

- *Formal Certification of Code-Based Cryptographic Proofs*  
Gilles Barthe (joint work with Benjamin Grégoire, Romain Janvier, Federico Olmedo, and Santiago Zanella Béguelin) ..... 2

### 10:00-10:30 Computational soundness (chair: Bruno Blanchet)

- *Computational soundness of observational equivalence*  
Hubert Comon-Lundh and Véronique Cortier ..... 4

## 10:30-11:00 Coffee Break

### 11:00-12:30 Languages and compilers (chair: Anupam Datta)

- *A Cryptographic Compiler for Information-Flow Security*  
Cédric Fournet, Gurvan Le Guernic, and Tamara Rezk ..... 6
- *High-Level Programming for E-Cash (work in progress)*  
Pedro Adao, Cédric Fournet, Nataliya Guts, and Francesco Zappa Nardelli ..... 7
- *A Formal Language for Cryptographic Pseudocode*  
Michael Backes, Matthias Berg, and Dominique Unruh ..... 8

## 12:30-14:00 Lunch

### 14:00-16:00 Automatic proofs and proof formalisms (chair: Ralf Küsters)

- *Automated Proofs for Asymmetric Encryption*  
Cristian Ene, Judicaël Courant, Yassine Lakhnech, Marion Daubignard, and Pascal Lafourcade ..... 9
- *Cryptographically Verified Implementations for TLS*  
Karthikeyan Bhargavan, Ricardo Corin, Cédric Fournet, and Eugen Zalinescu ..... 11
- *Task Probabilistic Input/Output Automata as Domains*  
Aaron D. Jaggar, Catherine Meadows, Michael Mislove, and Roberto Segala ..... 12
- *Reasoning about Conditional Probability and Concrete Security in Protocol Proofs (Work in Progress)*  
Anupam Datta, Joseph Halpern, John Mitchell, Riccardo Pucella, and Arnab Roy ..... 13

## 16:00-16:30 Coffee Break

# Formal Certification of Code-Based Cryptographic Proofs<sup>\*</sup>

Gilles Barthe<sup>1,3</sup>, Benjamin Grégoire<sup>2,3</sup>, Romain Janvier, Federico Olmedo<sup>4</sup>,  
and Santiago Zanella Béguelin<sup>2,3</sup>

<sup>1</sup> IMDEA Software, Spain

<sup>2</sup> INRIA Sophia Antipolis - Méditerranée, France

<sup>3</sup> Microsoft Research - INRIA Joint Centre, France

<sup>4</sup> National University of Rosario, Argentina

As cryptographic proofs have become essentially unverifiable, cryptographers have argued in favor of systematically structuring proofs as sequences of games. Code-based techniques form an instance of this approach that takes a code-centric view of games, and that relies on programming language theory to justify steps in the proof—transitions between games. While these techniques contribute to increase confidence in the security of cryptographic systems, code-based proofs involve such a large palette of concepts from different fields that machine-verified proofs seem necessary to achieve the highest degree of confidence. In an inspiring paper, Halevi convincingly argued that a tool assisting in the construction and verification of proofs is necessary to solve the crisis with cryptographic proofs.

CertiCrypt is a framework to construct machine-checked code-based proofs in the Coq proof assistant. CertiCrypt achieves many goals of Halevi’s ideal tool. At the same time, it brings a formal semanticist perspective on the design of the tool, and in particular ensures the highest guarantees with the smallest trusted base. The main characteristics of CertiCrypt are:

*Direct and faithful encoding of code-based techniques.* In order to take advantage of the generality of the code-based game-based approach and to be readily accessible to cryptographers, we have chosen a formalism that is commonly used by cryptographers to describe games. Concretely, the lowest layer of CertiCrypt is a deep embedding in Coq of an imperative programming language with random assignments, structured datatypes, and procedure calls. The language semantics takes into account non-standard features such as complexity of programs, variable usage and calling policies, that are of paramount importance in cryptographic proofs.

*Support for code-based proofs.* Automating the most mundane proof steps and providing powerful libraries to carry advanced proofs steps are equally necessary to ensure an efficient use of any formal tool. We have therefore developed theories of program equivalences, including a theory of contextual equivalence, w.r.t. sets of input and output variable, and a relational logic. These theories are fully formalized in Coq, and are used in conjunction with a set

---

<sup>\*</sup> Most of the work was performed while all the authors were working at INRIA Sophia-Antipolis Méditerranée. The work is partially supported by the ANR Sécurité et Sûreté Informatique SCALP.

of automated tactics for proving the correctness of some program transformations. All tactics are certified, in the sense that they are proved correct with respect to the operational semantics. Transformations fall into three main categories; 1) semantics preserving transformations, including compiler optimizations such as dead code elimination, code motion, constant propagation and common subexpression elimination; 2) transformations based on indistinguishability, i.e. a change that cannot be detected with a non-negligible probability; and 3) transformations based on failure events, where both games behave identically unless a certain *failure* occurs, and it is shown that this failure occurs with negligible probability. The talk shall discuss some specific transformations, and in particular one that consists in moving random assignments from games to oracles; such a transformation is widely used to prove properties in the random oracle model.

*Complete and independently verifiable proofs.* CertiCrypt benefits from being developed on top of the Coq proof assistant to go beyond Halevi’s vision in two respects. First, it supports the construction of full proofs, whereas Halevi suggests to focus only on their “mundane parts”. Second, it permits independent verifiability of proofs by third parties, which is an important motivation behind game-based proofs. Regarding full proofs, CertiCrypt requires that not only the transitions, but also all other reasoning (complexity-theoretic, group-theoretic, probabilistic) is justified formally. Regarding verifiability, CertiCrypt inherits from Coq its ability to provide certificates, or proof objects, that are automatically verifiable with a small trusted core, namely the type-checker of Coq.

The talk shall describe the design of CertiCrypt and its applications to machine-checked proofs of the semantic security of ElGamal and OAEP, and the Full Domain Hash lemma.

# Computational soundness of observational equivalence\*

Hubert Comon-Lundh<sup>†</sup>      Véronique Cortier<sup>‡</sup>

In [7], R. Canetti and J. Herzog consider (composable) security proofs for key exchange protocols. One of the main features of their work is to consider also a security property that is not a trace property: it requires indistinguishability between two versions of the protocol. They first rely on a composition theorem, showing that the security for an arbitrary number of sessions is implied by a one-session security. Then they design symbolic properties, corresponding to the computational ones and show that the symbolic abstraction of the protocol is sound w.r.t. these properties, for one session of the protocol. This allows to automate the security proofs, as described in [6] for instance.

Our work also aims at compositional computational security proofs through a symbolic abstraction. We claim to improve over [7] in the following respects:

1. We consider any indistinguishability property and prove that it is soundly abstracted by observational equivalence. This allows to consider many more security properties, such as for instance anonymity. In addition, this is a uniform way of abstracting properties. We do not need to introduce symbolic functionalities: we simply replace indistinguishability with observational equivalence.
2. We consider an arbitrary number of sessions: processes may be replicated. This is useful since we do not need to prove that one session security implies many-sessions security, while keeping the core of universal composability: observational equivalence of processes implies their security in any environment. In other words, our result allows to prove that a protocol is secure in any environment, without having to prove universal composability.
3. The secrets may be shared at any level: they can be local to a session, shared by one or more participants over sessions or even re-used in different protocols. This is specified at the symbolic level by the scope of the name generation.

To the best of our knowledge, the only general result relating observational equivalence and computational indistinguishability in an active attacker setting is [2], in which, however, cryptographic primitives are not part of the syntax.

We prove our result for symmetric encryption, relying on standard cryptographic assumptions (IND-CPA and INT-CTXT), but the same techniques can be applied to other security primitives such as signatures and public-key encryption. The proof requires the introduction of the concept of *tree soundness* in the case of passive attackers and the use of intermediate structures, which we called *computation trees*. These techniques are general and can be reused in other settings. A complete version of the result with full proofs can be found at [8].

Other related works include results for passive adversaries [1, 5, 12, 11] and for active adversaries, but for dedicated security properties: either trace properties [3, 7, 9, 10] or a special indistinguishability property [4, 7].

---

\*This work has been partially supported by the grants ARA FormaCrypt and ANR AVOTÉ.

<sup>†</sup>ENS Cachan and Research Center for Information Security (RCIS), AIST, Tokyo

<sup>‡</sup>LORIA, CNRS & INRIA project Cassis

## References

- [1] M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In *Foundations of Software Science and Computation Structure (FoSSaCS'06)*, volume 3921 of *LNCS*, pages 398–412, 2006.
- [2] P. Adão and C. Fournet. Cryptographically sound implementations for communicating processes. In *International Colloquium on Algorithms, Languages and Programming (ICALP'06)*, 2006.
- [3] M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable Dolev-Yao style cryptographic library. In *Computer Security Foundations Workshop (CSFW'04)*, 2004.
- [4] M. Backes and B. Pfitzmann. Relating cryptographic und symbolic key secrecy. In *Symp. on Security and Privacy (SSP'05)*, pages 171–182, 2005.
- [5] M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In *Proc. ICALP'05*, volume 3580 of *LNCS*, 2005.
- [6] B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.
- [7] R. Canetti and J. Herzog. Universally composable symbolic analysis of cryptographic protocols. In *Theory of Cryptography Conference (TCC'06)*, 2006.
- [8] H. Comon-Lundh and V. Cortier. Computational soundness of observational equivalence. Research Report 6508, INRIA, 04 2008.
- [9] V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *European Symposium on Programming (ESOP'05)*, volume 3444 of *LNCS*, pages 157–171, 2005.
- [10] P. Laud. Symmetric encryption in automatic analyses for confidentiality against active adversaries. In *Symp. on Security and Privacy (SSP'04)*, pages 71–85, 2004.
- [11] D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Theory of cryptography Conference (TCC 05)*, volume 3378 of *LNCS*, pages 169–187, 2005.
- [12] D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway language of encrypted expressions. *Journal of Computer Security*, 2004.

# A Cryptographic Compiler for Information-Flow Security

Cédric Fournet<sup>2,1</sup>, Gervan le Guernic<sup>1</sup>, and Tamara Rezk<sup>3,1</sup>

<sup>1</sup> MSR-INRIA Joint Centre

<sup>2</sup> Microsoft Research

<sup>3</sup> INRIA

**Abstract.** Our aim is to cryptographically enforce information flow policies in distributed programs that operate at diverse levels of trust. To this end, we are building a compiler from a small imperative language with locality and security annotations down to distributed F# code linked with concrete cryptographic libraries.

- In source programs, security depends on abstract policies for reading and writing the shared memory. These policies enable a simple treatment of confidentiality and integrity properties, and they can often be verified by typing.
- In their implementations, shared memory is unprotected and security depends instead on cryptographic protection. Our compiler inserts adequate encryptions and signatures to enforce confidentiality and integrity, at least at the levels prescribed in the source program. For efficiency reasons, it combines symmetric and asymmetric cryptography, and manages the reuse of keys shared between hosts.

We establish the computational soundness of our implementations against an active adversary that controls parts of the computation and schedules the rest of our compiled code. We rely on a type system that enforces a correct usage of cryptographic primitives in our code. We show that type soundness yields a computational variant of the non-interference property.

## References

1. C. Fournet, G. le Guernic, and T. Rezk. Cryptographic enforcement of information-flow security, 2008. At <http://www.msr-inria.inria.fr/projects/sec/cflow>.
2. C. Fournet and T. Rezk. Cryptographically sound implementations for typed information-flow security. In *35th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'08)*, pages 323–335, San Francisco, USA, Jan. 2008. ACM Press.
3. D. Syme. *F#*, 2005. At <http://research.microsoft.com/fsharp/>.

# High-Level Programming for E-Cash

Pedro Adão<sup>1</sup>, Cédric Fournet<sup>3,2</sup>, Nataliya Guts<sup>2</sup>, and Francesco Zappa Nardelli<sup>4,2</sup>

<sup>1</sup> SQUIG–Instituto de Telecomunicações and IST, TULisbon, Portugal

<sup>2</sup> MSR-INRIA Joint Centre

<sup>3</sup> Microsoft Research

<sup>4</sup> INRIA

**Abstract.** We consider symbolic characterizations of the Compact E-Cash protocol of Camenisch, Hohenberger, and Lysyanskaya [CHL05]. E-cash protocols [Cha82,CFN88] aim at providing robust abstractions for anonymous payment protocols. Properties of interest include, for instance, that users can spend coins anonymously, that users cannot forge coins, and that user should not spend the same coin twice without being eventually caught. These protocols involve sophisticated cryptographic constructions.

Relying on recent work on optimistic value commitment [FGN08], we design a calculus with E-cash primitives. Our calculus has a simple, symbolic semantics; it can be used for programming with E-cash and for reasoning on its properties, while shielding the programmer from its cryptographic implementation.

We consider two variants of the symbolic semantics. An abstract semantics rules out any double spending (by design). A more realistic, intermediate semantics accounts for the possibility of double spending, with reliable detection. We first relate these two semantics, then relate the intermediate semantics to the computational properties of the underlying E-cash protocol.

## References

- [AF06] Pedro Adão and Cédric Fournet. Cryptographically sound implementations for communicating processes. In *ICALP*, volume 4052 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2006.
- [Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.
- [CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 1988.
- [CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 302–321. Springer, 2005.
- [FGN08] Cédric Fournet, Nataliya Guts, and Francesco Zappa Nardelli. A formal implementation of value commitment. *Programming Languages and Systems (ESOP’08)*, volume 4960 of *LNCS*, pages 383–397. Springer, 2008.

# A Formal Language for Cryptographic Pseudocode

Michael Backes<sup>1,2</sup>, Matthias Berg<sup>1</sup>, and Dominique Unruh<sup>1</sup>

<sup>1</sup> Saarland University, Saarbrücken, Germany

<sup>2</sup> MPI-SWS

Game-based cryptographic proofs [9, 3] are typically expressed using pseudocode, which lacks a formal semantics. This leads to ambiguous specifications, hidden mistakes, and wrong proofs. For example, if a subroutine representing an adversary is invoked twice, it might be unclear whether the adversary may keep state between these two invocations. The author of the security definition may explicitly point out these ambiguities and resolve them; this, however, assumes that the author is aware of all other possible interpretations.

We propose a language that is expressive enough to specify all constructs occurring in cryptographic games [1], including probabilistic behaviors and the usage of oracles. From a language perspective, oracles constitute higher-order arguments that are passed to a program. We allow oracles to keep state between their invocations by including ML-style references in the language. Finally, our language supports events, which constitute a common technique in game-based cryptographic proofs for identifying undesirable behavior. This results in a higher-order functional probabilistic language with references and support for events. Nevertheless, the language is simple enough to be accessible to researchers without a strong background on theory of programming languages. The language has been implemented in the proof assistant Isabelle/HOL [7] to enable formal verification. Moreover, we have formalized common game relations such as denotational equivalence, observational equivalence, and computational indistinguishability, and we have conducted first game-based cryptographic proofs in the language. We expect our formalization to culminate in a tool that enables cryptographers to conveniently check the validity of their proofs without having to bother about the details of the language.

The work that comes closest to our framework is CertiCrypt [2], which constitutes a framework for reasoning about game-based cryptographic proofs in the Coq proof assistant [6]. CryptoVerif [4] constitutes an automated tool to support game-based cryptographic proofs. Other approaches for reasoning about game-based cryptographic proofs have been presented in [5, 8].

## References

- [1] M. Backes, M. Berg, and D. Unruh. A formal language for cryptographic pseudocode. Long version available at <http://www.infsec.cs.uni-sb.de/~berg/publications/lang4cp.pdf>, 2008.
- [2] G. Barthe, B. Gregoire, R. Janvier, and S. Zanella Beguelin. Formal certification of code-based cryptographic proofs. IACR ePrint Archive, Aug. 2007. Online available at <http://eprint.iacr.org/2007/314>.
- [3] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT 2006*, volume 4004 of *LNCS*. Springer, 2006. Full version online available at <http://eprint.iacr.org/2004/331.ps>.
- [4] B. Blanchet. A computationally sound mechanized prover for security protocols. In *Proc. 27th IEEE Symposium on Security & Privacy*, pages 140–154, 2006.
- [5] R. Corin and J. den Hartog. A probabilistic hoare-style logic for game-based cryptographic proofs. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2006.
- [6] T. C. development team. The Coq Proof Assistant Reference Manual, 2006. Available at <http://coq.inria.fr>.
- [7] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.
- [8] D. Nowak. A framework for game-based security proofs. IACR Cryptology ePrint Archive 2007/199, 2007. <http://eprint.iacr.org/>.
- [9] V. Shoup. Sequences of games: A tool for taming complexity in security proofs. IACR ePrint Archive, Nov. 2004. Online available at <http://eprint.iacr.org/2004/332.ps>.

# Automated Proofs for Asymmetric Encryption. <sup>\*</sup>

Judicaël Courant, Marion Daubignard, Cristian Ene, Yassine Lakhnech, and  
Pascal Lafourcade

Université Grenoble 1, CNRS, VERIMAG  
firstname.last@imag.fr

We present an automated proof method for analyzing generic asymmetric encryption schemes in the random oracle model (ROM). Generic encryption schemes aim at transforming schemes with weak security properties, such as one-wayness, into schemes with stronger security properties, especially security against chosen ciphertext attacks. Examples of generic encryption schemes are [5, 11, 10, 4, 2, 8, 7, 6]. The paper contains two main contributions. The first one is a compositional Hoare logic for proving IND-CPA-security. That is, we introduce a simple programming language (to specify encryption algorithms that use one-way functions and hash functions) and an assertion language that allows to state invariants and axioms and rules to establish such invariants. Compositionality of the Hoare logic means that the reasoning follows the structure of the program that specifies the encryption oracle. The assertion language consists of three atomic predicates. The first predicate allows us to express that the value of a variable is indistinguishable from a random value even when given the values of a set of variables. The second predicate allows us to state that it is computationally infeasible to compute the value of a variable given the values of a set of variables. Finally, the third predicate allows us to state that the value of a variable has not been submitted to a hash function.

Transforming the Hoare logic into an (incomplete) automated verification procedure is quite standard. Indeed, we can interpret the logic as a set of rules that tell us how to propagate the invariants backwards. We have done this for our logic resulting in a verification procedure implemented in less than 250 lines of CAML. We have been able to automatically verify IND-CPA security of several schemes among which [4, 7, 6]. Our Hoare logic is incomplete for two main reasons. First, the reader should notice that IND-CPA security is an observational equivalence-based property, while with our Hoare logic we establish invariants. Nevertheless, as shown in one of our propositions, we can use our Hoare logic to prove IND-CPA security at the price of completeness. That is, we prove a stronger property than IND-CPA. The second reason, which we think is less important, is that for efficiency reasons some axioms are stronger than needed.

The second contribution of the paper presents a simple criterion for plaintext awareness (PA). Plaintext awareness has been introduced by Bellare and Rogaway in [2]. It has then been refined in [1] such that if an encryption scheme is PA and IND-CPA then it is IND-CCA. Intuitively, PA ensures that an adversary cannot generate a valid cipher without knowing the plaintext, and hence, the decryption oracle is useless for the adversary. The definition of PA is complex

---

<sup>\*</sup> This work was supported by ANR SeSur SCALP, SFINCS and AVOTE.

and proofs of PA are also often quite complex. In this paper, we present a simple syntactic criterion that implies plaintext awareness. Roughly speaking the criterion states that cipher should contain as a sub-string the hash of the plaintext and the random seed. This criterion applies for many schemes such as [4, 6, 7] and easy to check. Although (or maybe because) the criterion is simple, the proof of its correctness is complex.

Putting together these two contributions, we get a proof method for IND-CCA security, that applies for instance to the constructions in [4, 6, 7].

An important feature of our method is that it is not based on a global reasoning and global program transformation as it is the case for the game-based approach [3, 9]. Indeed, both approaches can be considered complementary as the Hoare logic-based one can be considered as aiming at characterizing, by means of predicates, the set of contexts in which the game transformations can be applied safely.

## References

1. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO'98*, pages 26–45, 1998.
2. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT'04*, volume 950 of *LNCS*, pages 92–111, 1994.
3. M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331, 2004.
4. Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS'93*, pages 62–73, 1993.
5. I. Damgard. Towards practical public key systems secure against chosen ciphertext attacks. In *CRYPTO'91*, pages 445–456, 1992.
6. T. Okamoto and D. Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA'01*, pages 159–175, 2001.
7. D. Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In *PKC'00*, pages 129–146, 2000.
8. V. Shoup. Oaep reconsidered. *J. Cryptology*, 15(4):223–249, 2002.
9. V. Shoup. Sequences of games: a tool for taming complexity in security proofs, 2004. URL: <http://eprint.iacr.org/2004/332>.
10. D. Soldera, J. Seberry, and C. Qu. The analysis of zheng-seberry scheme. In *ACISP*, volume 2384 of *LNCS*, pages 159–168, 2002.
11. Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *J. on Selected Areas in Communications*, 11(5):715–724, 1993.

# Cryptographically Verified Implementations for TLS

Karthikeyan Bhargavan  
Microsoft Research

Ricardo Corin  
MSR-INRIA Joint Centre

Cédric Fournet  
Microsoft Research

Eugen Zălinescu  
MSR-INRIA Joint Centre

Recent advances in formal methods and tools enable the automated verification of complex security protocols. However, these tools remain difficult to apply, since verification occurs independently of the development process, rather than during design, prototyping, and testing. We are thus interested in integrating protocol verifiers to these phases, by narrowing the gap between concrete implementations and verified models.

We extract symbolic and computational models from executable code, relying in particular on a new tool for extracting CryptoVerif [1] scripts from ML. We share as much code as possible between implementations and models: they differ mostly in their implementations of core cryptographic libraries, either as concrete code or as encoded assumptions on the adversary.

As a case study, we consider the Transport Layer Security protocol (TLS) [2], one of the most widely deployed communications protocols. We program a small functional implementation of TLS 1.0 in ML. Both client and server code interoperate with mainstream implementations. We obtain a range of positive security results, covering both symbolic and computational cryptographic aspects of the protocol. More details can be found in [3].

**Acknowledgements** We thank Bruno Blanchet and Bogdan Warinschi for helpful discussions on computational verification during this work.

## References

- [1] Bruno Blanchet. Computationally sound mechanized proofs of correspondence assertions. In *20th IEEE Computer Security Foundations Symposium (CSF'07)*, pages 97–111, Venice, Italy, July 2007. IEEE.
- [2] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246 (Proposed Standard), January 1999.
- [3] <http://www.msr-inria.inria.fr/projects/sec/fs2cv/index.html>.

# Task Probabilistic Input/Output Automata as Domains

Aaron D. Jaggar<sup>a,1</sup>, Catherine Meadows<sup>b,2</sup>, Michael Mislove<sup>c,2</sup> and Roberto Segala<sup>d</sup>

<sup>a</sup>DIMACS, Rutgers University, New Brunswick, NJ

<sup>b</sup>Naval Research Laboratory, Washington, DC

<sup>c</sup>Department of Mathematics, Tulane University, New Orleans LA

<sup>d</sup>Dipartimento di Informatica, Università di Verona, Italy

Probabilistic input/output automata were devised by Segala, Lynch and Vaandrager [3] as models of computation that support probabilistic reasoning. In attempting to apply them to reason about crypto-protocols, these models were augmented in [1] with *tasks*, the purpose of which was to limit the power of the adversary in controlling the actions performed by other agents on the system. Task PIOAs have been applied successfully in [1] to model the oblivious transfer protocol, achieving a level of granularity that moves away from the Dolev-Yao model of a universally powerful adversary toward an adversary that is more realistic and more limited in what it can do. However, this and other attempts to combine computational and formal reasoning about crypto-protocols face barriers to being easily digested and understood, in particular in the details, because of the complications that are inherent in making precise the notions needed to model each of these aspects. In addition to this common problem, the Task PIOA approach relies on a mathematical model that, while being accurate and faithful, nonetheless is difficult to absorb. Our work addresses this issue by recasting the Task PIOA model within a more familiar and computationally intuitive framework provided by domain theory.

Domain theory is a mainstay of models of computation. Pioneered by Dana Scott as models of the untyped lambda calculus, these mathematical structures have become the *de facto* approach to modeling computation. Domains are defined using partial orders which reflect information content: the higher an object is in the order, the more computational information it contains. Further, basis elements in a domain correspond to tractable computations, while their limits – expressed as least upper bounds in the informatic order – represent recursive computations that are not finitely computable. Domains also feature a number of constructs that support reasoning about the myriad effects that arise in computation – resumptions, nondeterminism and probabilistic computations, to name a few. The simplicity and intuitive structure of domains allows them to provide models that users often find accessible and fairly easy to manipulate.

In this talk we will describe our on-going effort to use domain theory to model Task PIOAs as they are used by Lynch, Canetti, Segala et al. in [1]. We use a monad on measure spaces whose co-algebras encompass models of Task PIOAs. We use the solution of the associated domain equation to simplify the constructions on probability measures used in [1]. In particular, the **Apply** operator, a crucial construction from [1] that is used to construct the probability measures on executions and traces that are used by Task PIOAs, is presented in a simpler form here, and its relation to schedulers is more easily understood. To achieve this we make use of Martin’s notion of measurement in [2] to deal with the fact that **Apply** is non-monotonic.

Ultimately, the aim of our work is the use domain theory to provide a semantics of Task PIOAs that makes exactly the same identifications and distinctions as the one in [1]. This is work in progress, and the results so far focus on understanding how to capture the essence of the mathematical aspects of the Task PIOA models within a domain-theoretic model. Once complete, this should make the proofs of properties of Task PIOAs easier to understand, and their application to analyzing crypto-protocols more tractable.

[1] Canetti, R., N. Lynch, et al, Using Probabilistic I/O Automata to Analyze an Oblivious Transfer Protocol, preprint MIT-LCS-TR-1001.

[2] Martin, K., The Measurement Process in Domain Theory, LNCS vol. 1853, pages 116–126, 2000.

[3] Segala, R., N. Lynch and F. Vaandrager, Compositionality for Probabilistic Automata, LNCS vol. 2761, pages 208–221, 2003.

---

<sup>1</sup> Support by the National Science Foundation is gratefully acknowledged.

<sup>2</sup> Support by the US Office of Naval Research is gratefully acknowledged.

# Reasoning about Conditional Probability and Concrete Security in Protocol Proofs (Work in Progress)

Anupam Datta<sup>1</sup>, Joseph Y. Halpern<sup>2</sup>, John C. Mitchell<sup>3</sup>, Riccardo Pucella<sup>4</sup>,  
Arnab Roy<sup>3</sup>

<sup>1</sup> CyLab, Carnegie Mellon University

<sup>2</sup> Computer Science Department, Cornell University

<sup>3</sup> Computer Science Department, Stanford University

<sup>4</sup> College of Computer and Information Science, Northeastern University

*Computational Protocol Composition Logic (CPCL)* [2, 3, 9, 10] is a logic for proving asymptotic security properties (authentication and secrecy) of protocols in the computational model. The soundness of its proof system is established by reduction to game-based security conditions for cryptographic primitives (e.g. CMA-security for signatures). Since game-based security conditions are stated using conditional probability, CPCL captures these conditions using a non-standard conditional implication  $\supset$ . In this work, we seek to improve CPCL in two significant ways.

*First*, we clean up the treatment of conditional probability in CPCL by extending prior work in the artificial intelligence community. The interpretation of  $\supset$  in [2] is reminiscent of one of the interpretations of  $\rightarrow$  in conditional logic, where  $\phi \rightarrow \psi$  can be interpreted as “typically, if  $\phi$  then  $\psi$ ” [8]. In the Goldszmidt-Morris-Pearl [5] formulation of  $\epsilon$ -semantics [1, 6], a formula  $\phi \rightarrow \psi$  is evaluated with respect to a sequence  $(Pr_1, Pr_2, \dots)$  of probability measures (*probability sequence*, for short): it is true if, roughly speaking,  $\lim_{n \rightarrow \infty} Pr_n(\psi \mid \phi) = 1$  (where  $Pr_k(\psi \mid \phi)$  is taken to be 1 if  $Pr_k(\phi) = 0$ ). Unfortunately, this formulation is not quite strong enough for cryptographic purposes, where we need the convergence to be faster than any inverse polynomial. In a companion paper [7], we show that we can give first-order conditional logic this interpretation, while still preserving the soundness and completeness of a well-known axiomatization of it [4]. One of the major contributions of this paper is to show how this logic can be used to provide a clean and elegant logic for security properties.

*Second*, we define concrete security semantics for CPCL and develop a proof system that supports high-level, symbolic reasoning about such quantitative security guarantees. As shown in [7], for a large fragment of first-order conditional logic, a high-level “qualitative” proof, which provides only asymptotic guarantees, can be automatically converted to a proof that provides concrete guarantees. We show in this paper how that proof system can be used in conjunction with axioms and rules for reasoning about protocols to establish concrete security properties of a signature-based challenge response protocol.

## References

1. E. Adams. *The Logic of Conditionals*. Reidel, Dordrecht, Netherlands, 1975.
2. A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP '05)*, Lecture Notes in Computer Science, pages 16–29. Springer-Verlag, 2005.
3. A. Datta, A. Derek, J. C. Mitchell, and B. Warinschi. Computationally sound compositional logic for key exchange protocols. In *Proceedings of 19th IEEE Computer Security Foundations Workshop*, pages 321–334. IEEE, 2006.
4. N. Friedman, J. Y. Halpern, and D. Koller. First-order conditional logic for default reasoning revisited. *ACM Trans. on Computational Logic*, 1(2):175–207, 2000.
5. M. Goldszmidt, P. Morris, and J. Pearl. A maximum entropy approach to non-monotonic reasoning. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 15(3):220–232, 1993.
6. M. Goldszmidt and J. Pearl. Rank-based systems: A simple approach to belief revision, belief update and reasoning about evidence and actions. In *Principles of Knowledge Representation and Reasoning: Proc. Third International Conference (KR '92)*, pages 661–672. 1992.
7. J. Y. Halpern. From qualitative to quantitative proofs of security properties using first-order conditional logic. In *AAAI*, 2008. To appear.
8. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.
9. A. Roy, A. Datta, A. Derek, and J. C. Mitchell. Inductive proofs of computational secrecy. In *Computer Security - ESORICS 2007, 12th European Symposium on Research Computer Security, Proceedings*, Lecture Notes in Computer Science. Springer, 2007.
10. A. Roy, A. Datta, and J. C. Mitchell. Formal proofs of cryptographic security of diffie-hellman based protocols. In *Symposium On Trustworthy Global Computing*, Lecture Notes in Computer Science. Springer, 2007.