FormaCrypt: Formal Computational Cryptography

Bruno Blanchet[1], David Pointcheval[1]
Jean Goubault-Larrecq[2], Hubert Comon-Lundh[2]
Stéphanie Delaune[2], Steve Kremer[2]
Véronique Cortier[3], Mathieu Turuani[3]
Martín Abadi[4]

[1]LIENS [2]LSV [3]LORIA [4]UCSC & Microsoft Research

November 2008

ARA SSIA FormaCrypt: participants

- Laboratoire d'Informatique de l'Ecole Normale Supérieure (LIENS)
  - Bruno Blanchet
  - David Monniaux (at VERIMAG since Sept. 1st, 2007)
  - David Pointcheval
- Laboratoire Spécification et Vérification (LSV), ENS Cachan
  - Jean Goubault-Larrecq
  - Mathieu Baudet (at DCSSI since July 1st, 2006)
  - Hubert Comon-Lundh (from August 2007)
  - Stéphanie Delaune (from October 2007)
  - Steve Kremer
  - Laurent Mazaré (October 2006–April 2007)
- Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
  - Véronique Cortier
  - Stéphanie Delaune (January 2007–September 2007)
  - Heinrich Hördegen (Phd thesis defended in December 2007)
  - Mathieu Turuani
  - Bogdan Warinschi (in Bristol since January 2007)
  - Eugen Zalinescu (Phd thesis defended in December 2007)
- Scientific advisor: Martín Abadi

Proofs of cryptographic protocols

There are two main frameworks for analyzing security protocols:

- The Dolev-Yao model: a formal, abstract model.

  The cryptographic primitives are ideal blackboxes.
  The adversary uses only those primitives.

  Proofs can be done automatically.

- The computational model: a realistic model.

  The cryptographic primitives are functions on bit-strings.
  The adversary is a polynomial-time Turing machine.

  Proofs are done manually.

Our goal: bridge the gap between these two frameworks.

Three approaches

We have considered three approaches:

- Direct approach: build an automatic computationally sound prover.
- Intermediate approach: design a computationally sound logic, for reasoning symbolically on protocols.
- Modular approach: obtain computational soundness results, that is, show that security in the formal model implies security in the computational model.

We will obviously compare these approaches on examples ranging from protocols of the literature to more complex, realistic protocols.

CryptoVerif: An automatic computationally sound prover (LIENS, Abstract Interpretation and Cryptography teams)

We have implemented an automatic prover sound in the computational model:

- proves secrecy and correspondence properties.
  Correspondence = if some event has been executed, then other events have been executed before (except in cases of negligible probability).
- handles various cryptographic primitives: MACs (message authentication codes), symmetric encryption, public-key encryption, signatures, hash functions, ...
- works for a parametric number of sessions with an active adversary.
- gives a bound on the probability of an attack (exact security).

Produced proofs

As in Shoup's or Bellare and Rogaway's method, the proof is a sequence of games:

- The prover is given the first game, which represents real protocol, in interaction with an adversary.
- The prover transforms each game into the next one by syntactic transformations or by applying security assumptions on cryptographic primitives.

  The difference of probability between consecutive games is bounded.
- The last game is "ideal": the desired security properties can be read directly on it.

Process calculus for games

Games are formalized in a process calculus:

- It is adapted from the pi calculus.
- The semantics is purely probabilistic (no non-determinism).
- All processes run in polynomial time:
  - polynomial number of copies of processes,
  - length of messages on channels bounded by polynomials.

Indistinguishability as observational equivalence

Two processes (games) Q_1, Q_2 are observationally equivalent when the adversary has a negligible probability of distinguishing them:

$$Q_1 \approx Q_2$$

The adversary is represented by an acceptable evaluation context C (essentially, a process put in parallel with the considered games).

- Observational equivalence is an equivalence relation.
- It is contextual: $Q_1 \approx Q_2$ implies $C[Q_1] \approx C[Q_2]$ where C is any acceptable evaluation context.

Proof technique

We transform a game G_0 into an observationally equivalent one using:

- observational equivalences $L \approx R$ given as axioms and that come from security properties of primitives. These equivalences are used inside a context:
$$G_1 \approx C[L] \approx C[R] \approx G_2$$
- syntactic transformations: simplification, expansion of assignments, ...

We obtain a sequence of games $G_0 \approx G_1 \approx \ldots \approx G_m$, which implies $G_0 \approx G_m$.

If some equivalence or trace property holds with overwhelming probability in G_m, then it also holds with overwhelming probability in G_0.

Proof strategy: advice

- CryptoVerif tries to apply all equivalences given as axioms, which represent security assumptions.

  It transforms the left-hand side into the right-hand side of the equivalence.
- If such a transformation succeeds, the obtained game is then simplified.
- When these transformations fail, they may return syntactic transformations to apply in order to make them succeed, called advised transformations.

  CryptoVerif then applies the advised transformations, and retries the initial transformation.

Input and output of CryptoVerif

CryptoVerif is given as input

- the security assumptions on the cryptographic primitives;
- the initial game, given in a syntax close to the standard notations in cryptography;
- the properties to prove.

CryptoVerif outputs

- the (negligible) probability that each desired property is wrong;
- the sequence of games that leads to the proof;
- a succinct explanation of the transformations performed between games.

The user is allowed (but does not have) to interact with the prover to make it follow a specific sequence of games.

Demo

Demo

Experimental results

CryptoVerif is available at http://www.cryptoverif.ens.fr

We have tested it successfully on many examples:

- protocols of the literature: incorrect and corrected versions of Otway-Rees, Yahalom, Needham-Schroeder shared-key and public-key, Denning-Sacco public key, and Woo-Lam shared-key and public-key;
- the Full Domain Hash signature scheme;
- encryption schemes of [Bellare and Rogaway, CCS'93].
- Kerberos (with A. D. Jaggard, A. Scedrov, and J.-K. Tsay)

It starts being used by others:

- Computationally sound verification of security protocols in F# (Bhargavan et al, FCC'07)
- Study of implementations of TLS (Bhargavan et al, CCS'08)

A computationally sound logic (LORIA)

We have studied the following approach for proving protocols:

1. start from an existing protocol logic, designed in the formal model (here, the Protocol Composition Logic),
2. and adapt it to the computational model.

Advantage of this approach:
proofs that use the logic in the formal model can be adapted to the computational model with only minor corrections.

The Protocol Composition Logic (PCL)

The Protocol Composition Logic allows symbolic reasoning on protocols:

- The protocol is specified in a simple "protocol programming language".
- The logic consists of both
  - logical formulas (including predicates that specify knowledge and actions of participants)
  - and modal formulas, similarly to the Floyd-Hoare logic (if a formula is true at some point and certain actions are executed, then another formula holds afterwards).
- The logic allows compositional reasoning.

Adapting PCL to the computational model

We have adapted the Protocol Composition Logic to the computational model:

- We have given a new probabilistic, polynomial-time semantics to the logic: a formula is true if it holds with asymptotically overwhelming probability.
- We have given a meaning to logical connectives in this semantics.
- We had to extend the logic with new predicates that make sense in the computational model but not in the formal one, for example to express indistinguishability.

A soundness proof has been done for a subset of PCL with positive indistinguishability tests.

Proving more complex properties

We have extended this logic to more complex properties, in particular secrecy of keys.

This result allows the following compositional reasoning:

- we first prove the security of keys established using a key exchange protocol;
- then, we infer the security of a secure channel application that uses these keys.

Formal model: several abstractions

Messages are modeled by terms

- $\{m\}_k$: message m encrypted by k
- $\langle m_1, m_2 \rangle$: pair of m_1 and m_2
- ...

$\longrightarrow$ no collisions:

$$\forall m, m', k, k' \quad \{m\}_k \neq \{m'\}_{k'}, \{\{m\}_k\}_k \neq m, \langle m, m' \rangle \neq \{m\}_k, \ldots$$

Perfect encryption assumption:

Nothing can be learned from $\{m\}_k$ except if k is known.

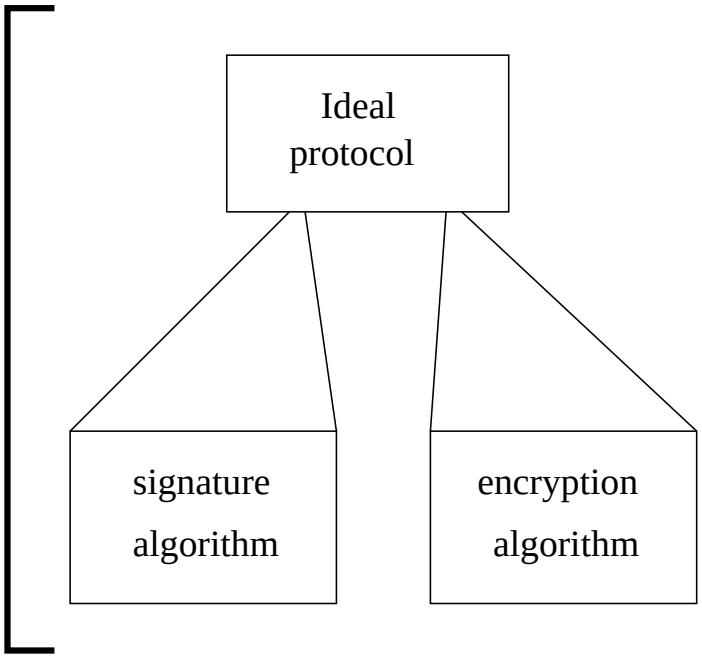
Much easier to analyze automatically

- numerous decidability results
- numerous automatic tools

Goal: soundness of the formal model

Composition of two approaches

Implemented protocol



Formal approach: verification of idealized protocols

Cryptographers: verification of the cryptographic primitives

Main results

Before the project: In the case of asymmetric encryption and signatures, trace properties (e.g. authentication) and secrecy can be safely abstracted symbolically.

During the project:

Active case

- Extension to hash functions and symmetric encryption, providing there is no key-cycles
  $\hookrightarrow$ This is an NP-complete property
- Extension to indistinguishability properties
- Extension to XML documents
- Module CryptoSec added to Avispa
- Safe composition of protocols
- Formalizing computational security of contract signing protocols

Passive case

- Soundness of guessing attacks
- Adding equational theories

Combination result in presence of hash functions (LSV and LORIA)

Example

$A \to B : h(s)$

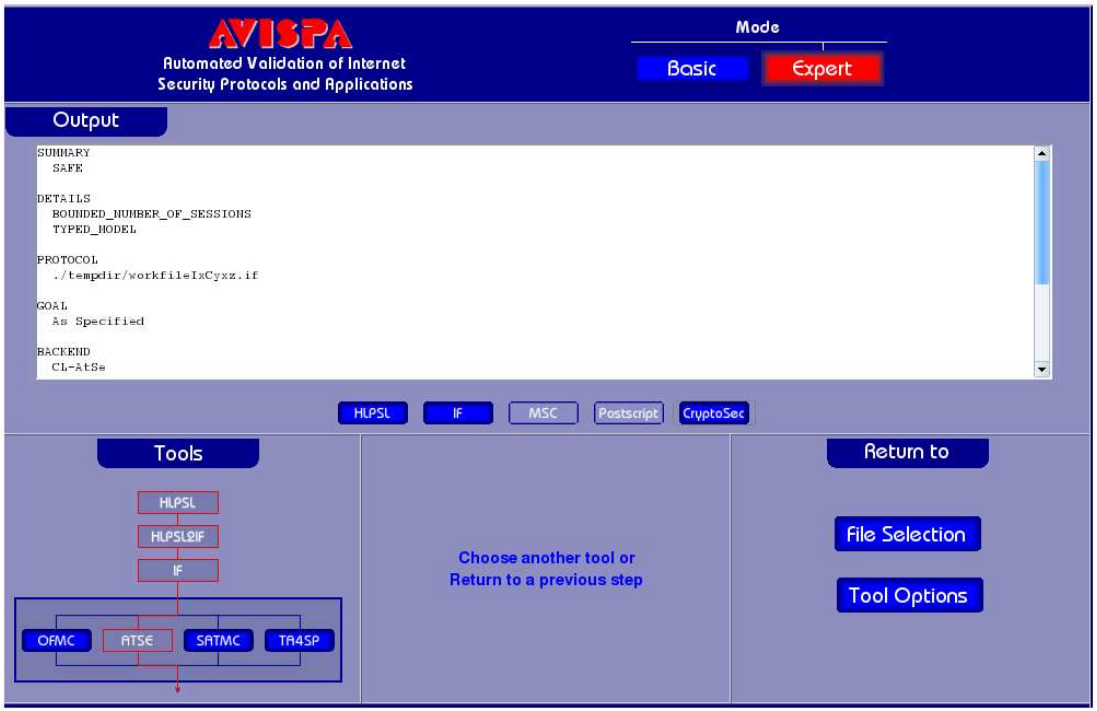
s is symbolically secret but not indistinguishable to an attacker: $h(n_b), n_0, n_1 \to b$

Results:

1. Design of a new formal secrecy property
2. Proof of its soundness and its faithfulness w.r.t. indistinguishability in our new setting:
   - pairing
   - asymmetric encryption
   - hashes (random oracle model)
3. NP-completeness of the secrecy property

Application: CryptoSec

Automatic proof at the cryptographic level
using Avispa, a symbolic theorem prover.



→ 9 protocols proved secure (for authentication or secrecy properties) at the cryptographic level.

Extension to symmetric encryption (LORIA)

Cycles of the form $\{k_1\}_{k_2}$, $\{k_2\}_{k_3}$, $\{k_3\}_{k_1}$ must be forbidden.

As a preliminary result, we obtain that:

> *Deciding if an intruder can create key cycles on keys of honest agents is NP-complete for a bounded number of sessions.*

Extension to indistinguishability-based properties (LSV and LORIA)

Definition (Computational indistinguishability)

$P \approx Q$ if for any adversary $\mathcal{A}$ (that is any PPT Turing machine)
$|\Pr\{r, r'(P(r)\|\mathcal{A}(r')) = 1\}| - |\Pr\{r, r'(Q(r)\|\mathcal{A}(r')) = 1\}|$
is negligible.

Intuitively, an attacker cannot tell the difference between P and Q.

Extension to indistinguishability-based properties (LSV and LORIA)

Definition (Computational indistinguishability)

$P \approx Q$ if for any adversary $\mathcal{A}$ (that is any PPT Turing machine)
$|\Pr\{r, r'(P(r)\|\mathcal{A}(r')) = 1\}| - |\Pr\{r, r'(Q(r)\|\mathcal{A}(r')) = 1\}|$
is negligible.

Intuitively, an attacker cannot tell the difference between P and Q.

There exists a similar symbolic definition!

Definition (observational equivalence)

$P \sim_o Q$ if for any process O, we have $P\|O \sim Q\|O$.

Intuitively, an observer cannot tell the difference between P and Q.

Proving computational indistinguishability symbolically

Observational equivalence is a sound abstraction of computational indistinguishability.

$$P \sim_o Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

- For simple processes
  (A fragment of the applied pi-calculus that captures most security protocols)
- For symmetric encryption implemented using IND-CCA2 schemes

Applications:

- symbolic proof of privacy-like properties
- symbolic proof of anonymity
- symbolic proof of simulatability
- symbolic proof of secrecy (to some extend)
- ...

Guessing attacks (LSV and LORIA)

Guessing attacks: The adversary can guess low entropy values such as passwords and verify them off-line.

Results:

1. Design of a new security property for the interaction between normal and password-based encryption
2. Proof of soundness of static-equivalence (passive case) w.r.t. indistinguishability for:
   - asymmetric encryption
   - symmetric encryption
   - password-based encryption
3. Resistance against off-line guessing attacks composes even for protocols that share the same password

Equational theories (LSV and LORIA)

Equational theories: Cryptographic primitives can be modelled as equational theories, which allows to specify algebraic properties

Results:

1. Generalization of the DDH assumption and soundness result for modular exponentiation in the presence of a passive adversary
2. Soundness result for protocols based on bilinear pairings in the presence of a passive adversary
3. A general framework for reasoning about the soundness of equational theories in the presence of an adaptive adversary: equational theories include modular exponentiation and xor; application to dynamic group key exchange protocols

Case studies

We have made a number of case studies, to apply the techniques we have developped:

- CryptoVerif and AVISPA/CryptoSec have been successfully applied to many protocols of the literature.
- CryptoVerif has been applied to the FDH signature scheme and to encryption schemes of [Bellare, Rogaway, CCS'93]
- CryptoVerif has been applied to Kerberos, with and without its public-key extension PKINIT.
- The modular approach has also been applied to an XML encryption scheme used for implementing access control policies.

Conclusion

- We have investigated three different approaches for bridging the gap between the computational and the formal models of cryptography.
- The project has produced:
  - 4 international journal papers (JACM, TCS, TDSC, FMSD)
  - 2 invited conferences and tutorials (RTA, TGC)
  - 16 international conference papers (CRYPTO, IEEE S&P, CSF(W), CCS, FSTTCS, ...)
  - 8 international workshop papers or presentations (FCC, FCS-ARSPA, WITS, ICS)
  - 3 PhD theses
  - 1 *habilitation à diriger des recherches*
  - 2 tools that provide proofs of protocols in the computational model.

More details, publications, and software available at:
`http://www.di.ens.fr/~blanchet/formacrypt/`

Conclusion

- The project has lead to a series of workshops FCC (Formal and Computational Cryptography):
  - FCC'06, affiliated to ICALP'06 and co-located with CSF'06 (the PC co-chairs, 3 PC members, and 2 talks from FormaCrypt; 50 participants)
  - FCC'07, co-located with CSF'07 (2 PC members and 3 talks from FormaCrypt; 26 participants)
  - FCC'08, affiliated to CSF'08 and co-located with LICS'08 (1 PC co-chair and 1 talk from FormaCrypt; 23 participants)
- A spring school on Computational and Symbolic Proofs of Security is organized in April 2009 in Japan (organizer and several speakers from FormaCrypt).

Conclusion

- There were the following project meetings:
  - March 2006 (ENS Ulm)
  - January 2007 (ENS Cachan)
  - November 2007 (ENS Ulm)
  - A fourth meeting will take place soon.

  These meetings attracted researchers outside the FormaCrypt project, from various European countries.
- This project has brought together two essentially disjoint communities (formal methods and cryptography).
  It has also produced 7 multi-partner publications.

Work planed for the rest of the project

The project is scheduled until July 5, 2009.
We plan to do the following:

- Direct approach: Extend CryptoVerif to Diffie-Hellman using the computational Diffie-Hellman assumption (LIENS)
- Modular approach: Enrich the execution model to capture more protocols (e.g. with loops), add more cryptographic primitives in a modular way (LORIA and LSV)
- Case studies: Compare between CryptoVerif and AVISPA/CryptoSec (LIENS and LORIA)

Follow-up projects

- The AVOTÉ project on the analysis of e-voting protocols (France Telecom R&D, LORIA, LSV, Verimag) started this year and also studies computational aspects, for different security properties.
- A possible project is discussed between the LIENS and the CELAR (Rennes) on user-oriented extensions of CryptoVerif (interface, documentation, ...).