CIL: A Proof System for Computational Indistinguishability

Gilles Barthe\textsuperscript{3}, Marion Daubignard\textsuperscript{2}, Bruce Kapron\textsuperscript{1} and Yassine Lakhnech\textsuperscript{2}

\textsuperscript{1}University of Victoria  
\textsuperscript{2}VERIMAG, Université de Grenoble, CNRS  
\textsuperscript{3}IMDEA, Madrid

Barcelona, Sept. 2009
About me...

- PhD student at Université de Grenoble, since Oct. 2008 under the direction of Pr. Yassine Lakhnech.
- Laboratoire VERIMAG, in Grenoble, FRANCE.
- Team DCS (Distributed Complex Systems)
- Work partially supported by the ANR project SCALP, in cooperation with IMDEA (Madrid), INRIA Sophia-Antipolis, LRI (Paris), CNAM (Paris), ENS Lyon.
Problematics

- Provable security provides guarantees thanks to definitions and proofs, but one scheme = one proof, mainly paper-and-pencil proofs, sometimes unreliable...

- Our long-term goal is to improve the security of cryptographic systems by enabling Computer-Aided Cryptographic Proofs

- Two kinds of existing approaches:
  - indirect: reasoning in the symbolic framework + soundness theorems
  - directly reason in the computational model (e.g. game-based techniques, Hoare logics of limited scope, applied pi-calculus, etc.)

- But the general principles of reasoning remain informal: lack of generic proof systems.
Previous work ([CDELL, CCS’08])

Security proofs for asymmetric encryption schemes

- Three predicates capturing properties of the variables.
- A Hoare logic to propagate these properties.
- Enables to compute some conditions to fulfill to be secure.

Some weaknesses:

- Does not enable conditional reasoning
- Requires to add a new set of rules for each new primitive
- Cannot capture completely the dependencies between variables
Generalities about CIL

- Most security criteria rely on the concept of indistinguishability. Hence our current subgoal: CIL, a system of inference rules to prove indistinguishability.
- Based on computational frames: computational interpretations of the π-calculus frames of [AF,POPL’01], extended with random sampling, adversary calls and oracles.
- Judgments for indistinguishability, negligibility, possibly conditional.
- Reasoning directly in the computational model; additional assumptions can be plugged in, e.g. ROM or OW.
The framework

A cryptographic game is a process of the form:

\( \vec{x}_i \leftarrow \vec{d}_i, \quad c \leftarrow A_1(u_1), \quad r \leftarrow A_2(u_2) \mid I_1/O_1 \cdots I_\ell/O_\ell \)

...consisting in three entities:

- the **frame**: consists in the draws and the computation of the adversary’s inputs.
- a **two-tier adversary**: find-stage \( A_1 \) and guess-stage \( A_2 \), outputting a challenge \( c \) and a final result \( r \).
- the **oracles**: stateful implementations answering the adversary’s queries.

Two dual interpretations: a purely functional semantics, and a more syntactic, pi-calculus-like approach.
Overview of the proof system: 1. the statements

Let $s$ be a frame, $\mathcal{A}$ an adversary, $\mathcal{I}, \mathcal{I}'$ sets of oracles, and let $(s|\mathcal{I})||\mathcal{A}$ denote the interaction of the three entities.

Two kinds of judgments

- $\models s : \epsilon E$ iff for all $\mathcal{A} \in \mathcal{A}$, $\Pr_{x \leftarrow (s|\mathcal{I})||\mathcal{A}}[E \ x] \leq \epsilon$
- $\models s \sim \epsilon t$ iff for all $\mathcal{A} \in \mathcal{A}$,
  
  $|\Pr_{b \leftarrow (s|\mathcal{I})||\mathcal{A}[b = 1] - \Pr_{b \leftarrow (t|\mathcal{I}'||\mathcal{A}[b = 1]| \leq \epsilon$

Remarks:

- Validity extends to sequents $\Gamma \vdash \phi$ in the usual manner.
- Given a set $\Gamma$ of statements, $\Gamma \models \phi$ iff $\models \Gamma$ implies $\models \phi$. 

Barthe, Daubignard, Kapron, Lakhnech

Computational Indistinguishability Logic
Overview of the proof system: 2. the rules

A substantial extension of a logic by Impagliazzo and Kapron to formalize indistinguishability [FOCS’03], CIL only consists in

12 inference rules

Three categories of rules

- basic and interface rules: e.g., capturing that $\sim$ is an equivalence relation, to introduce counting arguments, to transmit negligibility of probability when an event implies another, etc.
- composition rules: to allow substitution, we define a notion of poly-time context and compose it either with a frame or the adversary.
- oracle rules: to capture reasoning like the so-called up-to-bad lemma
Here are, for example, two rules of CIL:

1. The ‘case study’ rule:

   $E \rightarrow s \sim t \quad s : \neg E \quad t : \neg E$

   \[ s \sim t \]  

2. A rule dealing with oracles:

   \[
   s|\mathcal{I} : \epsilon \varphi \forall \land E \quad \mathcal{I} =_{\varphi} \mathcal{I}'
   \]

   \[
   s|\mathcal{I}' : \epsilon \varphi \forall \land E \quad \text{NegOR}\forall
   \]
Using CIL, we have proven:

- Semantic security of encryption schemes:
  - Bellare and Rogaway’s scheme of 93,
  - Pointcheval’s construction at PKC’00,
  - REACT,
  - Hashed El-Gamal in the ROM and standard model,
  - OAEP (IND-CCA security is on-going work)

- Unforgeability of signature schemes: PSS, FDH.

Remark: the level of abstraction of CIL allows it to support proofs of meta-results, e.g. implications between various security criteria.
Others’ contributions in progress

1. CEL, a Computational Equivalence Logic, to capture reasoning performed on equality of distributions;
2. well-advanced formalization in Coq, as a part of the SCALP project,
3. Certicrypt: framework built on top of Coq that allows machine-checked construction and verification of code-based proofs.
Conclusions

- CIL is a generic proof system for indistinguishability that formalizes standard principles of reasoning frequently used in the existing proofs.
- CIL is applicable: several constructions have already been proven secure.
- On the long run, we intend to develop a interfaced tool usable by non-expert Coq users that would provide Coq proofs of schemes and protocols.