Logics for Reasoning about Cryptographic Constructions

Bruce Kapron (U Victoria) (joint work with **Russell Impagliazzo** (UCSD))

Cryptographic Soundnness

- Goal: formal systems with *cryptographically sound* semantics
- Our notion of security: complexity based (provable security)
  - Asymptotic
  - Probabilistic
  - Adversaries are probabilistic poly-time (PPT) TM's
  - *Reduction-based* proofs
- *Computational indistinguishability* is a central notion (PRG, PRG, Secure Enc., CZK...)

Our Approach

- Bottom-up: start with a low-level logic; prove a *master soundness theorem*
- Soundness of more specialized systems can be proved via interpretation

Low level logic

Asymptotics

- Bounded arithmetic (open induction) – use an explicit non-standard value $\mathbf{n}$ which represents the security parameter
- Has a model with all nonstandard values a have $|a|$ polynomial in $\mathbf{n}$.

Low level logic

Complexity

- Two-sorted logic w/ Cobham-style axioms for poly-time functions.

Probability

- Counting terms: $\#(|x| = k)\varphi(x)$

Master soundness theorem

We can show that if

$$\varphi \vdash \psi$$

and for any sequence $\vec{f}$ of poly time functions $\varphi(\vec{f})$ holds asymptotically in $\mathbb{N}$, then $\psi(\vec{f})$ holds asymptotically in $\mathbb{N}$.

Axiomatizing computational indistinguishability

Terms

- let $x \leftarrow \mathsf{rs}(\mathbf{n})$ in t
- let $x \leftarrow \mathsf{rand}(\mathbf{n})$ in t
- Intended interpretation – PPT functions (a.k.a. poly-samplable distribution ensembles.)

Formulas

- $u \approx t$
- Intended interpretation u and t represent computationally indistinguishable ensembles

Rules for $\approx$

$$\frac{u \approx u'}{v\{u/x\} \approx v\{u'/x\}} \text{ (SUB)}$$

- Substituting indistinguishable terms into a poly-time context results in indistinguishable terms (otherwise could compose the distinguisher with the context)
- Notion of substitution ($v\{u/x\}$) requires some syntactic restrictions (keep random bindings at top level).

Rules for $\approx$

$$\frac{\mathsf{let}\ i \leftarrow \mathsf{rand}(p(\mathbf{n}))\ \mathsf{in}\ u(i) \approx \mathsf{let}\ i \leftarrow \mathsf{rand}(p(\mathbf{n}))\ \mathsf{in}\ u(i+1)}{u(0) \approx u(p(\mathbf{n}))} \quad \text{(H-IND)}$$

- Induction-like rule
- Essential for reasoning about iterative constructions (e.g. via hybrid arguments)
- Parameterized by polynomial p

Rules for $\approx$

$$\frac{\vdash Q_1 Q_2 \ldots Q_k (s = t)}{\text{let } b_1 \text{ in } \ldots \text{ let } b_k \text{ in } s \approx \text{let } b_1 \text{ in } \ldots \text{ let } b_k \text{ in } t} \quad \text{(UNIV)}$$

- Q_i – unversal quantification of the form $\forall |x| \leq k$ or $\forall i \leq k$
- b_i – corresponding random binding of the form $x \leftarrow \mathsf{rs}(k)$ or $i \leftarrow \mathsf{rand}(k)$
- Not really a rule: premise requires external reasoning ($\vdash$)

Rules for $\approx$

We also have a (premise-free) rule (EDIT) which captures principles about basic operations on string, e.g.

$$\text{let } \begin{pmatrix} x_1 \leftarrow \mathsf{rs}(k_1) \\ x_2 \leftarrow \mathsf{rs}(k_2) \end{pmatrix} \text{ in } x_1 \circ x_2 \approx \text{let } x \leftarrow \mathsf{rs}(k_1 + k_2) \text{ in } x$$

where $\circ$ denotes string concatenation.

(Note that we could replace $\approx$ by identically distributed here.)

Soundness of the axiomatization

By interpretation, we can prove a soundness result for $\approx$, so that, e.g., if we can derive

$$\mathsf{let}\ x \leftarrow \mathsf{rs}(\mathbf{n})\ \mathsf{in}\ u(x) \approx \mathsf{let}\ y \leftarrow \mathsf{rs}(\mathbf{n})\ \mathsf{in}\ v(y)$$

we may conclude that $\forall A \forall k \exists n_0 \forall n \geq n_0$:

$$\Pr[A(u(X)) = 1] - \Pr[A(v(Y))] = 1 \leq \frac{1}{n^k}$$

where X,Y are chosen uniformly from $\{0,1\}^n$.

E.g. – stretching the output of a PRG

Can formalize: "f is a PRG with stretch 1":

$$\text{let } x \leftarrow \text{rs}(\mathbf{n}) \text{ in } f(x) \approx \text{let } x \leftarrow \text{rs}(\mathbf{n}+1) \text{ in } x$$

Let $b(x) = f(x)_{\{1\}}$, $r(x) = f(x)_{\{2,\ldots,\mathbf{n}+1\}}$, define b' and r' as follows:

$$r'(x,0) = x \qquad b'(x,0) = \epsilon$$

$$r'(x,i+1) = r(r'(x,i)) \qquad b'(x,i+1) = b(r'(x,i)) \circ b'(x,i)$$

Example continued

We want to derive:

$$\text{let } x \leftarrow \text{rs}(\mathbf{n}) \text{ in } b'(x, 2\mathbf{n}) \approx \text{let } x \leftarrow \text{rs}(2\mathbf{n}) \text{ in } x$$

Conclusion (via soundness): b' is a PRG with stretch n.

Note: we could derive this for $p(\mathbf{n})$ for any polynomial p instead of $2\mathbf{n}$.

Example continued

- A completely formalized proof of about 20 lines is possible
- Note that there is virtually no reasoning about probability beyond the painfully obvious (captured by EDIT rule)
- External system ($\vdash$) required only for reasoning about definitions.
- Somewhat trivial result, but more sophisticated proofs have been formalized.

Pseudo-random functions

Standard definition (based on oracle TM's and *random* functions) does not fit easily into our $\approx$ framework – rework definition using indistinguishability of *traces*.

$\mathrm{Tr}(f, h, j)$ – trace generated by adversary h making j queries to f

$\mathrm{RTr}(h, j)$ – trace generated by adversary h making j queries, each new query answered using a uniformly generated string.

f is a PRF if $\forall h \forall j (\mathrm{Tr}(f, h, j) \approx \mathrm{RTr}(h, j))$

Pseudo-random functions

- We can have formalized the construction of PRF's from PRG's
- Very little reasoning about probabilities required, captured by the EDIT rule
- Bulk of the proof involves manipulating defnitions (equational induction) and reasoning about $\approx$
- Low-level system is not used for reasoning about probabilities

Formalized soudness of encryption logics

With PRF's in hand, we can define an "encryption function"

$$\mathcal{E}(k, m) = \text{let } r \leftarrow \text{rs}(\mathbf{n}) \text{ in } (f(k, r) \oplus m) \circ r$$

Recall Abadi-Rogaway *patterns*

$$P ::= x \in \mathbf{Vars} | k \in \mathbf{Keys} | m \in \mathbf{Mess} | \{P\}_k | \square$$

Formalized soundness of encryption logics

Rules for pattern equivalence

If π is any permutation on $\mathbf{Keys}$, $P \equiv P[\pi]$ (EQ1)

If k has *no* occurrence in P, then

$$P[\{P_1\}_k/x_1, \ldots, \{P_r\}_k/x_r] \equiv P[\Box/x_1, \ldots, \Box/x_r] \text{(EQ2)}$$

Formalized soundness of encryption logics

Define

$$\overline{P} = P \quad \overline{(P,Q)} = \langle \overline{P}, \overline{Q} \rangle \quad \overline{\{P\}_k} = \mathcal{E}(k, \overline{P}) \quad \overline{\Box} = \mathsf{rs}(2\mathbf{n})$$

Then $P \equiv Q \Longrightarrow \overline{P} \approx \overline{Q}$.

Proof requires external reasoning (about $\oplus$)

Conclusions and future work

- So far...
  - Axiomatization of $\approx$ which is adequate to formalize many proofs in foundations of cryptography (even beyond the obvious – e.g., involving PRFs)
  - Very general meta-system adequate to formalize a good chunk of the foundations of cryptography

Conclusions and future work

- Possible research directions
  - Develop techniques for automated *checking* of $\approx$ proofs
  - Refine proof-theoretic analysis, e.g., to allow extraction of reductions and/or parameters from proofs
  - Use logical techniques to understand foundational issues in cryptography (e.g., limits of black-box reductions)
  - Give axiomatizations of other basic notions (OWF, CZK)

Example proof details

First use characterization of f

$$\text{let } x \leftarrow \mathsf{rs}(\mathbf{n}) \text{ in } f(x) \approx \text{let } x \leftarrow \mathsf{rs}(n+1) \text{ in } x$$

and SUB with context

$$\text{let } i \leftarrow \mathsf{rand}(\mathbf{n}) \text{ in } z_{\{1\}} \circ b'(z_{\{2\ldots\mathbf{n}\}}, i)$$

to obtain (†):

$$\text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \end{pmatrix} \text{ in } b(x) \circ b'(r(x), i) \approx$$

$$\text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}+1) \end{pmatrix} \text{ in } x_{\{1\}} \circ b'(x_{\{2\ldots\mathbf{n}\}}, i)$$

Example proof details

Recall

$$b'(x,0) = \epsilon \qquad\qquad b'(x,i+1) = b(r'(x,i)) \circ b'(x,i)$$

Can show using equational induction (PV)

$$\forall i < 2\mathbf{n} \forall |x| = \mathbf{n}(b'(x,i+1) = b(x) \circ b'(r(x),i)).$$

So by UNIV we obtain (††):

$$\mathsf{let} \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \end{pmatrix} \mathsf{in}\ b'(x,i+1) \approx$$

$$\mathsf{let} \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \end{pmatrix} \mathsf{in}\ b(x) \circ b'(r(x),i)$$

Example proof details

Define h by $h(z, x, i) = z_{\{1 \ldots 2\mathbf{n}-i\}} \circ b'(x, i)$. We then obtain

$$\text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, i+1) \approx \text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } z_{\{1 \ldots 2\mathbf{n}-(i+1)\}} \circ b'(x, i+1)$$

(by UNIV)

$$\approx \text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } z_{\{1 \ldots 2\mathbf{n}-(i+1)\}} \circ b(x) \circ b'(r(x), i)$$

(by SUB and ††)

$$\approx \text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}+1) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } z_{\{1 \ldots 2\mathbf{n}-(i+1)\}} \circ x_{\{1\}} \circ b'(x_{\{2 \ldots \mathbf{n}\}}$$

(by SUB and †)

$$\approx \text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } z_{\{1 \ldots 2\mathbf{n}-i\}} \circ b'(x, i)$$

(by EDIT and SUB)

$$\approx \text{let } \begin{pmatrix} i \leftarrow \mathsf{rand}(2\mathbf{n}) \\ x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, i)$$

(by UNIV)

Example proof details

Now apply H-IND to obtain

$$\text{let } \begin{pmatrix} x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, 0) \approx \text{let } \begin{pmatrix} x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, 2\mathbf{n})$$

Recall $h(z, x, i) = z_{\{1,\ldots,2\mathbf{n}-i\}} \circ b'(x, i)$, so by UNIV and EDIT we obtain:

$$\text{let } \begin{pmatrix} x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, 0) \approx \text{let } z \leftarrow \mathsf{rs}(2\mathbf{n}) \text{ in } z$$

$$\text{let } \begin{pmatrix} x \leftarrow \mathsf{rs}(\mathbf{n}) \\ z \leftarrow \mathsf{rs}(2\mathbf{n}) \end{pmatrix} \text{ in } h(z, x, 2\mathbf{n}) \approx \text{let } x \leftarrow \mathsf{rs}(\mathbf{n}) \text{ in } b'(x, 2\mathbf{n})$$

and the result follows by transitivity.

PRF details

$$\mathrm{Tr}(f, h, s, 0) = \epsilon$$

$$\mathrm{Tr}(f, h, s, j+1) = \sigma \circ q \circ f(s, q)$$

where $\circ$ denotes string concatentation, $\sigma = \mathrm{Tr}(f, h, s, j)$ and $q = h(\sigma)$.

Note: s is intended to be a (short) random string (length $\mathbf{n}$), assume $f : \{0,1\}^{\mathbf{n}} \rightarrow \{0,1\}^{\mathrm{n}}$.

This approach is not new (compare Goldwasser-Sipser definition of interactive proofs.)

Random function details

$$\mathrm{RTr}(h, r, 0) = \epsilon$$

$$\mathrm{RTr}(h, r, j+1) = \sigma \circ q \circ r^i$$

where

$$\sigma = \mathrm{RTr}(h, r, j)$$

$$q = h(\sigma)$$

$$r^i = \text{ bits } i\mathbf{n}+1 \ldots (i+1)\mathbf{n} \text{ of } r$$

$$i = (\mu l < j)(\sigma^l = q)$$

$$\sigma^l = \text{ bits } 2l\mathbf{n}+1 \ldots (2l+1)\mathbf{n} \text{ of } \sigma$$

In this case, r is intended to be a random string of length

PRF details

$f : \{0,1\}^{\mathrm{n}} \to \{0,1\}^{\mathrm{n}}$ is a PRF if

$\forall h \forall j (\mathsf{let}\ s \leftarrow \mathsf{rs}(\mathrm{n})\ \mathsf{in}\ \mathrm{Tr}(f, h, s, j) \approx \mathsf{let}\ r \leftarrow \mathsf{rs}(j\mathrm{n})\ \mathsf{in}\ \mathrm{R}$