Computational Soundness of Symbolic Security and Implicit Complexity

Bruce Kapron
Computer Science Department
University of Victoria
Victoria, British Columbia

NII Shonan Meeting, November 3-7, 2013

Overview

- We would like to be use secure cryptographic primitives (e.g., block ciphers, hash functions) in schemes and protocols which realize some security functionality
- Problem: how do we validate the correctness of these constructions?
- Two traditional approaches: symbolic and computational
- Can we relate the two?
- Can implicit complexity help?

Symbolic model

- Basic model: Dolev-Yao [Dolev Yao 82]
- Primitives achieve *perfect* security
- Adversaries are in total control of execution and communication
  1. May initiate any number of executions of a protocol in any role with any party
  2. Can intercept and modify any message, or send arbitrary messages to active parties
- Adversaries are nondeterministic – concern is with the *existence* of an attack
- No computational assumptions

A well-known success story

Needham-Schroeder public-key protocol

$$\begin{array}{lllll} 1 & A & \longrightarrow & B & : \{A.N_A\}_{k_B} \\ 2 & B & \longrightarrow & A & : \{N_A.N_B\}_{k_A} \\ 3 & A & \longrightarrow & B & : \{N_A\}_{k_B} \end{array}$$

At the end of this protocol, A and B might assume: (1) they know with whom they have been interacting, (2) they agree on the values N_a and N_b and (3) no one else knows N_a and N_b

Lowe's attack on NSPK

Using a model-checking approach in a Dolev-Yao framework, [Lowe 1996] demonstrated the following *interleaving* attack.
Oscar runs two copies α and β of this protocol concurrently (one as the receiver with A and one as the initiator, impersonating A with B).

$$
\begin{array}{llcll}
\alpha.1 & A & \longrightarrow & O & : \{A.N_A\}_{k_O} \\
\beta.1 & O(A) & \longrightarrow & B & : \{A.N_A\}_{k_B} \\
\beta.2 & B & \longrightarrow & O(A) & : \{N_A.N_B\}_{k_A} \\
\alpha.2 & O & \longrightarrow & A & : \{N_A.N_B\}_{k_A} \\
\alpha.3 & A & \longrightarrow & O & : \{N_B\}_{k_O} \\
\beta.3 & O(A) & \longrightarrow & B & : \{N_B\}_{k_B}
\end{array}
$$

Dolev-Yao model – pros and cons

- Simple symbolic model allows automated reasoning (theorem proving or model checking) – useful for discovering flaws in protocols
- Semantics is not clear – what does it mean when a protocol is shown to be correct?
- Mismatch with computational cryptography – idealized (perfect) primitives, adversaries are computationally unbounded and nondeterministic.

Computational security

- Cryptographic primitives are modeled as PPT algorithms,
- Security holds against poly-time adversaries.
- Security is formulated *probabilistically* – adversaries may have some (small) chance of success
- *Reduction paradigm*: to show a scheme $\mathcal{S}$ built using primitives $P_1, \ldots, P_2$ is secure, show that for any advesary A which breaks $\mathcal{S}$ there is an adversary A' which breaks one of the P_i's
  - *Black-box* reductions: $A' = M^A$ for some poly-time OTM M

Example – Asymptotic CPA security of encryption

- An encryption scheme is a triple $\langle Gen, Enc, Dec \rangle$ where Gen, Enc are PPT functions and Dec is a deterministic poly-time function such that for any $k \in \mathrm{Rng}(Gen)$, and any *message* m, $Dec(k, Enc(k, m)) = m$
- An *adversary* is a pair $A = \langle A_q, A_c \rangle$ where A_q is a poly-time OTM and A_c is PPT

CPA security continued

- Security is defined using the following game, which depends on a *security parameter* n
  1. $k \leftarrow Gen(1^n)$
  2. A_q is given oracle access to $Enc(k, \cdot)$ and 1^n as input and outputs the transcript h of its interaction with the oracle, plus a *challenge pair* m_0, m_1
  3. A_c is given t and $Enc(k, m_b)$ for a random $b \in \{0, 1\}$ and outputs a *guess* b'
- A's *advantage* is $Pr[b = b'] - \frac{1}{2}$
- The scheme is *secure* if there is a *negligible function* ν such that every adversary's advantage is bounded by $\nu(n)$

Computational approach – pros and cons

- Schemes and protocols are formulated in a compuational model
- Security guarantees closely related to security achievable by implementation (*concrete* approach offers quantifiable guarantees)
- Definitions are complex – just defining security of a primitive like encryption requires the use of OTMs
- Proofs are even more complex – involve reductions between (ostensibly) type-2 functions, in a probabilistic setting
- Proof automation is difficult (but not impossible, e.g., [Blanchet 07],[Barthes et. al. 12])

Relating the two views

Goal: Achieving the best of the two worlds.

One possible approach:

- *Computational Soundness:* computational security guarantees from symbolic proofs.
- Typical form: Protocol Π is symbolically secure $\Rightarrow$ generic instantiations of Π (under exactly-defined secure primitives) are computationally secure.

This enables:

- Doing proofs in a symbolic model (without explicitly dealing with complexity-based notions), and
- obtaining computational security from (once and for all) established computational soundness theorems.

Symbolically secure encryption

- Abadi & Rogaway 2001: The first result of this kind. Limited to eavesdropping adversaries and single-message protocols. Many extensions since then in the eavesdropping setting ([Micciancio Warinschi 02], [Herzog 04], . . .)
- [MW 04] – security of trace-based properties (e.g. authentication) against non-adaptive active adversaries, messages cannot contain secret keys
- [Hajiabadi K 13] – extension to adaptive adversaries, reduced restrictions on secret keys in messages

Abadi-Rogaway model

- *Expressions* represent messages built using encryption and simple data constructors, e.g., $\{\{k_1\}_{k_2}.k_3\}_{k_4}$
- Adversarial knowledge is modeled *inductively*: $F_{kr}(E, K)$ denotes the set of keys *recoverable* from E assuming keys in K are already known
- We take the least fixed point of $\lambda K.F_{kr}(E, K)$ to obtain E's *recoverable keys* – all other keys in E are *hidden*
- The *pattern* of E is obtained by replacing subexpressions of the form $\{E'\}_k$, where k is hidden in E, by $\Box$.
- Expressions E, F are *equivalent* ($E \equiv F$) if they have the same pattern, up to renaming of keys

Abadi-Rogaway model

- If we interpret encryption computationally (e.g. by a CPA-secure encryption scheme) then for any n, an expression E has a natural interpretation $[\![E]\!]$ as a distribution over $\{0,1\}^{p(n)}$ for some polynomial p
- Distribution ensembles $X = \{X\}_n$ and $Y = \{Y\}_n$ are *computationally indistinguishable* ($X \approx Y$) if for every n any PPT adversary has negligible in n advantage in distinguishing between a sample from X_n and one from Y_n
- Abadi-Rogaway soundness result (roughly): if E, F are expressions with no *key cycles*, then $E \equiv F \Longrightarrow [\![E]\!] \approx [\![F]\!]$
  - Original result formulated for a more restictive form of encrpytion security

A more foundational approach?

- Some drawbacks of the A-R model – specific to a particular primitive and adversarial model, soundness proof follows the pattern of a standard computational security proof (i.e. reduction)
- Each time we introduce a variation of this logic, a new computational soundness proof will be required
- We will consider a different approach with connections to ICC (at least syntatic modeling of complexity)
- A logical analogue to cryptography based on generic assumptions (OWF $\Rightarrow$ PRG $\Rightarrow$ PRF $\Rightarrow$ CPA-encryption)
- One goal: soundness of A-R style logics via interpretation in a more generic logic

Formalizing computational indistinguishability

- This approach introduced by [Impagliazzo, K 03]
- A distribution ensemble is *samplable* if there is a there is a poly time function which is given n uniform bits of randomness and generates a sample from X_n
- We can just view samplable ensembles as PPT functions, which can be presented by using a standard function algebra (we used Cobham, but more implicit approaches would work as well) with primitives for randomization – $rand(n)$ and $rs(n)$
- Can give axioms and rules for $f \approx g$
- Possible to define basic primitives, e.g., f is a PRG if $f(rs(n)) \approx rs(n+1)$

Formalizing computational soundness

- Can we prove the soundness of A-R by *interpreting* it in the IK system?
- We are working on an approach which will prove soundness for certain encryption schemes based on pseudorandom functions (PRFs)
- Goal one: modeling PRFs (pseudorandom functions) and using them to define encryption in the IK system
- Goal two: proving A-R soundness by interpretation
- Goal three (somewhat orthgonal): formally proving PRG $\Rightarrow$ PRF (mimicking the construction of [Goldreich, Goldwasser, Micali 86])

Modeling PRFs

- Need to model *state* and *interaction* in a function algebra – a natural candidate is BFF_2. We'll give a rough description of how this is done. Let $\mathbb{B} = \{0,1\}^*$.
- An *oracle* (intensional function) is a pair $f = \langle s_f, a_f \rangle$ where $s_f : S_f \times \mathbb{B} \to S_f$ and $a_f : S_f \times \mathbb{B} \to \mathbb{B}$
- We assume that elements of S_A just consist of A's randomness and its query history
- An *adversary* is a triple $A = \langle q_A, s_A, e_A \rangle$ where $q_A : S_A \to \mathbb{B}$, $s_A : S_A \times \mathbb{B} \to \mathbb{B}$ and $e_A : S_A \rightharpoonup \mathbb{B}$

Modeling interaction

- For $\sigma \in S_A$, $\tau \in S_f$, $Step(A, f, \sigma, \tau)$ equals $\langle \sigma', \tau' \rangle$ where $\sigma' = s_A(\sigma, a)$, $\langle \tau', a \rangle = f(\tau, q)$ and $q = q_A(\sigma)$.
- We can now use feasible iteration to define $Step^*$ so that if A is polytime in its input, then for sufficiently large n, $e_A(Step_1^*(A, f, \sigma_0(1^n), \tau_0(1^n), 1^n))$ is identically distributed to $A^f(1^n)$
- This will allow us to define *indistinguishability* of intensional functions $f \sim g$ – can "lift" axiomatiztion of $\approx$ to one for $\sim$

Defining PRFs in this setting

Consider the intensional function ρ whose state consists of a sequence of pairs of elements of $\mathbb{B}$ corresponding to the queries that it has made. ρ is defined as follows: suppose $\sigma = (\langle q_1, a_1 \rangle, \ldots, \langle q_k, a_k \rangle)$. If there is some $j \leq k$ with $q = q_j$, then

$$\rho(\sigma, q) = \langle \sigma, a_i \rangle$$

where $i = (\mu j \leq k)(q = q_j)$. Otherwise,

$$\rho(\sigma, q) = r \;\leftarrow\; rs(n).\langle \sigma^\frown \langle q, r \rangle, r \rangle$$

Then for a length preserving f (i.e.$|f(x)| = |x|$,) f is a PRF if $f \sim \rho$.

Conclusions and future work

- There are now a variety of computationally sound symbolic systems for reasoning about security (in the A-R style. There are many other approaches that we haven't even mentioned.)
- Generic logics for computational indistinguishability could provide a more basic framework for reasoning about security – logics for specific primitives or security models proved sound by *interpretation*
- Can model, e.g., PRFs
- To do: finish up Goals 2 and 3