Probabilistic Polynomial-Time Process Calculus for Security Protocol Analysis

J. Mitchell, A. Ramanathan, A. Scedrov, V. Teague
P. Lincoln, P. Mateus, M. Mitchell

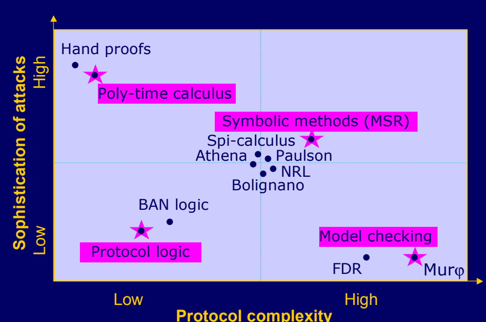
Standard analysis methods

- Finite-state analysis
- Dolev-Yao model
  - Symbolic search of protocol runs
  - Proofs of correctness in formal logic
- Consider probability and complexity
  - More realistic intruder model
  - Interaction between protocol and cryptography

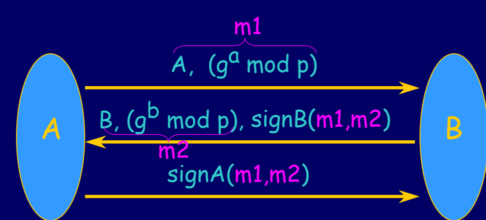
Easier

Harder

Protocol analysis spectrum



IKE subprotocol from IPSEC



Result: A and B share secret g^{ab} mod p

Analysis involves probability, modular exponentiation, digital signatures, communication networks, …

Equivalence-based specification

- Real protocol
  - The protocol we want to use
  - Expressed precisely in some formalism
- Idealized protocol
  - May use unrealistic mechanisms (e.g., private channels)
  - Defines the behavior we want from real protocol
  - Expressed precisely in same formalism
- Specification
  - Real protocol indistinguishable from ideal protocol
  - Beaver '91, Goldwasser-Levin '90, Micali-Rogaway '91
  - Depends on some characterization of observability
- Achieves compositionality

Compositionality (intuition)

- Crypto primitives
  - Ciphertext indistinguishable from noise
    $\Rightarrow$ encryption secure in all protocols
- Protocols
  - Protocol indistinguishable from ideal key distribution
    $\Rightarrow$ protocol secure in all systems that rely on secure key distributions

Compositionality

- Intuitively, if:
  - Q securely realizes I ,
  - R securely realizes J,
  - R, J use I as a component,
- then
  R{Q/I} securely realizes J
- Fits well with process calculus because ≈ is a congruence
  - $Q \approx I \Rightarrow C[Q] \approx C[I]$
  - contexts constructed from R, J, simulators

Language Approach

Roscoe '95, Schneider '96, Abadi-Gordon'97

- Write protocol in process calculus
  - Dolev-Yao model
- Express security using observational equivalence
  - Standard relation from programming language theory
    $P \approx Q$ iff for all contexts $C[\ ]$, same observations about $C[P]$ and $C[Q]$
  - Inherently compositional
  - Context (environment) represents adversary
- Use proof rules for ≈ to prove security
  - Protocol is secure if no adversary can distinguish it from some idealized version of the protocol
    Great general idea; application is complicated

Aspect of compositionality

- Property of observational equiv

$$\frac{A \approx B \qquad C \approx D}{A|C \approx B|D}$$

similarly for other process forms

The proof is easy

$$\frac{A \approx B \qquad C \approx D}{A|C \approx B|D}$$

- Recall definition
  $P \approx Q$ iff for all contexts $C[\ ]$, same observations about $C[P]$ and $C[Q]$
- Assume
  - $A \approx B \Rightarrow \forall C[\ ],\ C[A] \sim C[B]$
- Therefore
  - For any $C[\ ]$, let $C'[\bullet] = C[\bullet \mid D]$
  - By assumption, $C'[A] \sim C'[B]$
  - Which means that $A|D \approx B|D$
- By similar reasoning
  - Can show $A|C \approx A|D$
  - Therefore $A|C \approx A|D \approx B|D$

Probabilistic Poly-time Analysis

- Add probability, complexity
- Probabilistic polynomial-time process calc
  - Protocols use probabilistic primitives
    - Key generation, nonce, probabilistic encryption, ...
  - Adversary may be probabilistic
- Express protocol and spec in calculus
- Security using observational equivalence
  - Use probabilistic form of process equivalence

Pseudo-random number generators

- Sequence generated from random seed
  P_n: let $b = n^k$*-bit sequence generated from* n *random bits*
  in PUBLIC ⟨b⟩ end
- Truly random sequence
  Q_n: let b = *sequence of* n^k *random bits*
  in PUBLIC ⟨b⟩ end
- P is crypto strong pseudo-random number generator
  $P \approx Q$
  Equivalence is asymptotic in security parameter n

Secrecy for Challenge-Response

- Protocol P
  - $A \to B$: $\{ i \}_K$
  - $B \to A$: $\{ f(i) \}_K$
- "Obviously" secret protocol Q
  - $A \to B$: $\{ random\_number \}_K$
  - $B \to A$: $\{ random\_number \}_K$

Secrecy for Challenge-Response

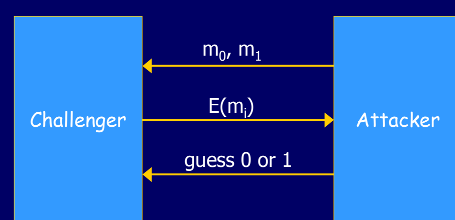
- Protocol P
  - $A \to B$: $\{ i \}_K$
  - $B \to A$: $\{ f(i) \}_K$
- "Obviously" secret protocol Q
  - $A \to B$: $\{ random\_number \}_K$
  - $B \to A$: $\{ random\_number \}_K$
- Analysis: $P \approx Q$ reduces to crypto condition related to *non-malleability* [Dolev, Dwork, Naor]
  - Fails for "plain old" RSA if $f(i) = 2i$

Non-malleability:
Given only a ciphertext, it is difficult to generate a different ciphertext so that the respective plaintexts are related

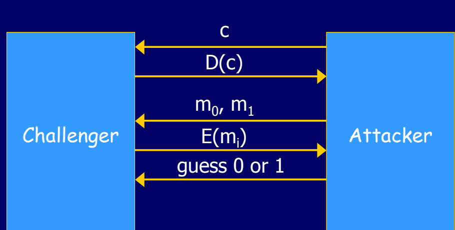
Security of encryption schemes

- Passive adversary
  - Semantic security
  - Indistinguishability
- Chosen ciphertext attacks (CCA1)
  - Adversary can ask for decryption before receiving a challenge ciphertext
- Chosen ciphertext attacks (CCA2)
  - Adversary can ask for decryption before *and after* receiving a challenge ciphertext

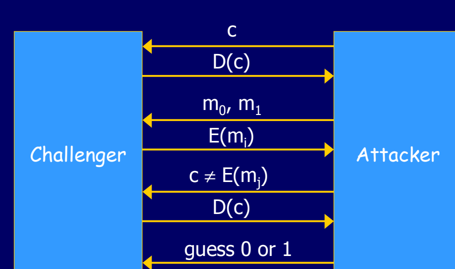
Passive Adversary



Chosen ciphertext CCA1



Chosen ciphertext CCA2



Specification with Authentication

- Protocol P
  - $A \to B$: $\{ \text{random } i \}_K$
  - $B \to A$: $\{ f(i) \}_K$
  - $A \to B$: "OK" if f(i) received
- "Obviously" authenticating protocol Q
  - $A \to B$: $\underbrace{\{ \text{random } i \}_K}_{\text{public channel}} \; \underbrace{}_{\text{private channel}}$
  - $B \to A$: $\underbrace{\{ \text{random } j \}_K}_{\text{public channel}} \; \underbrace{i , j}_{\text{private channel}}$
  - $A \to B$: "OK" if private i, j match public msgs

Methodology

- Define general system
  - Process calculus
  - Probabilistic semantics
  - Asymptotic observational equivalence
- Apply to protocols
  - Protocols have specific form
  - "Attacker" is context of specific form

Nondeterminism vs encryption

- Alice encrypts msg and sends to Bob
  - $A \to B$: $\{ msg \}_K$
- Adversary uses nondeterminism
  - Process E_0 $\quad c\langle 0\rangle \mid c\langle 0\rangle \mid \ldots \mid c\langle 0\rangle$
  - Process E_1 $\quad c\langle 1\rangle \mid c\langle 1\rangle \mid \ldots \mid c\langle 1\rangle$
  - Process E
    - $c(b_1).c(b_2)\ldots c(b_n).\mathrm{decrypt}(b_1 b_2 \ldots b_n, msg)$

In reality, at most 2^{-n} chance to guess n-bit key

Related work

- Canetti; B. Pfitzmann, Waidner, Backes
  - Interactive Turing machines
  - General framework for crypto properties
  - Protocol ***simulates*** an ideal setting
  - Universally composable security
- Abadi, Rogaway, Jürjens; Herzog; Warinschi
  - Toward transfer principles between formal Dolev-Yao model and computational model

Technical Challenges

- Language for prob. poly-time functions
  - Extend work of Cobham, Bellantoni, Cook, Hofmann
- Replace nondeterminism with probability
  - Otherwise adversary is too strong ...
- Define probabilistic equivalence
  - Related to poly-time statistical tests ...
- Proof rules for probabilistic equivalence
  - Use the proof system to derive protocol properties

Syntax

Expressions have size poly in |n|

- Bounded π-calculus with integer terms

$P ::= 0$	
$\mid \; c_{q(\|n\|)} \langle T \rangle$	send up to q(|n|) bits
$\mid \; c_{q(\|n\|)} (x).\, P$	receive
$\mid \; \upsilon c_{q(\|n\|)} .\, P$	private channel
$\mid \; [T=T]\, P$	test
$\mid \; P \mid P$	parallel composition
$\mid \; !_{q(\|n\|)} .\, P$	bounded replication

Terms may contain symbol n; channel width and replication bounded by poly in |n|

Probabilistic Semantics

- Basic idea
  - Alternate between terms and processes
    - Probabilistic evaluation of terms (incl. rand)
    - Probabilistic scheduling of parallel processes
- Two evaluation phases
  - Outer term evaluation
    - Evaluate all exposed terms, evaluate tests
  - Communication
    - Match send and receive
    - Probabilistic if multiple send-receive pairs

Scheduling

- Outer term evaluation
  - Evaluate all exposed terms in parallel
  - Multiply probabilities
- Communication
  - E(P) = set of eligible subprocesses
  - S(P) = set of schedulable pairs
  - Prioritize – private communication first
  - Probabilistic poly-time computable scheduler that makes progress

Example

- Process
  - c⟨rand+1⟩ | c(x).d⟨x+1⟩ | d⟨2⟩ | d(y). e⟨x+1⟩
- Outer evaluation
  - c⟨1⟩ | c(x).d⟨x+1⟩ | d⟨2⟩ | d(y). e⟨x+1⟩
  - c⟨2⟩ | c(x).d⟨x+1⟩ | d⟨2⟩ | d(y). e⟨x+1⟩

  Each prob $\frac{1}{2}$
- Communication
  - c⟨1⟩ | c(x).d⟨x+1⟩ | d⟨2⟩ | d(y). e⟨x+1⟩

  Choose according to probabilistic scheduler

Complexity results

- Polynomial time
  - For each closed process expression P, there is a polynomial q(x) such that
    - For all n
    - For all probabilistic polynomial-time schedulers

    eval of P halts in time q(|n|)

Complexity: Intuition

- Bound on number of communications
  - Count total number of inputs, multiplying by q(|n|) to account for $!_{q(|n|)}$. P
- Bound on term evaluation
  - Closed T evaluated in time $q_T(|n|)$
- Bound on time for *each* comm step
  - Example: c⟨m⟩ | c(x).P → [m/x]P
  - Substitution bounded by orig length of P
    - Size of number m is bounded
    - Previous steps preserve # occurr of x in P

Problem:

How to define process equivalence?

- Intuition
  - | Prob{ C[P] → "yes" } - Prob{ C[Q] → "yes" } | < ε
- Difficulty
  - How do we choose ε?
    - Less than 1/2, 1/4, … ? (not equiv relation)
    - Vanishingly small ? As a function of what?
- Solution
  - Use security parameter
    - Protocol is family $\{ P_n \}_{n>0}$ indexed by key length
  - Asymptotic form of process equivalence

Probabilistic Observational Equiv

- Asymptotic equivalence within f

  Process, context families $\{P_n\}_{n>0}$ $\{Q_n\}_{n>0}$ $\{C_n\}_{n>0}$

  $P \approx_f Q$ if $\forall$ contexts $C[\ ]$. $\forall$ obs v. $\exists n_0$. $\forall n > n_0$.
  $| Prob[C_n[P_n] \to v] - Prob[C_n[Q_n] \to v] | < f(n)$

- Asymptotically polynomially indistinguishable

  $P \approx Q$ if $P \approx_f Q$ for every polynomial $f(n) = 1/p(n)$

Final def'n gives robust equivalence relation

One way to get equivalences

- Labeled transition system
  - Evaluate process is a "maximally benevolent context"
  - Allows process read any input on a public channel or send output even if no matching input exists in process
  - Label with numbers "resembling probabilities"
- Bisimulation relation
  - If $P \sim Q$ and $P \xrightarrow{r} P'$, then exists Q' with $Q \xrightarrow{r} Q'$ and $P' \sim Q'$, and vice versa
- Strong form of prob equivalence
  - But enough to get started …

    [van Glabbeek - Smolka - Steffen]

Provable equivalences

- Assume scheduler is stable under bisimulation

- $P \sim Q \ \Rightarrow\ C[P] \sim C[Q]$
- $P \sim Q \ \Rightarrow\ P \approx Q$
- $P \mid (Q \mid R) \approx (P \mid Q) \mid R$
- $P \mid Q \approx Q \mid P$
- $P \mid 0 \approx P$

Provable equivalences

- $P \approx \upsilon\, c.\ (\ c\langle T\rangle \mid c(x).P)$ $\qquad x \notin FV(P)$
- $P\{a/x\} \approx \upsilon\, c.\ (\ c\langle a\rangle \mid c(x).P)$

  if bandwidth of c large enough
- $P \approx 0$ if no public channels in P
- $P \approx Q \ \Rightarrow\ P\{d/c\} \approx Q\{d/c\}$

  c , d same bandwidth, d fresh
- $c\langle T\rangle \approx c\langle T'\rangle$

  if $Prob[T \to a] = Prob[T' \to a]$ all a

Connections with modern crypto

- Cryptosystem consists of three parts
  - Key generation
  - Encryption (often probabilistic)
  - Decryption
- Many forms of security
  - Semantic security, non-malleability, chosen-ciphertext security, …
  - Formal derivation of semantic security of ElGamal from DDH and *vice versa*
- Common conditions use prob. games

Decision Diffie-Hellman DDH

- Standard crypto benchmark
- n security parameter (*e.g.,* key length)

  G_n cyclic group of prime order p,

  length of p roughly n ,

  g generator of G_n
- For random a, b, c $\in \{0, \ldots, p-1\}$

  $\langle g^a, g^b, g^{ab} \rangle \approx \langle g^a, g^b, g^c \rangle$

ElGamal cryptosystem

- n security parameter (*e.g.,* key length)
  G_n cyclic group of prime order p ,
  length of p roughly n , g generator of G_n
- Keys
  - public $\langle g , y \rangle$, private $\langle g , x \rangle$ s.t. $y = g^x$
- Encryption of $m \in G_n$
  - for random $k \in \{0, . . . , p-1\}$ outputs $\langle g^k , m\, y^k \rangle$
- Decryption of $\langle v, w \rangle$ is $w\, (v^x)^{-1}$
  - For $v = g^k$, $w = m\, y^k$ get
    $w\, (v^x)^{-1} = m\, y^k / g^{kx} = m\, g^{xk} / g^{kx} = m$

Semantic security

- Known equivalent:
  indistinguishability of encryptions
  - adversary can't tell from the traffic which of the two chosen messages has been encrypted
  - ElGamal:
    $\langle 1^n , g^k , m\, y^k \rangle \approx \langle 1^n , g^{k'} , m'\, y^{k'} \rangle$
- In case of ElGamal known to be equivalent to DDH
- *Formally derivable using the proof rules*

Current State of Project

- Compositional framework for protocol analysis
  - Determine crypto requirements of protocols
  - Precise definition of crypto primitives
- Probabilistic ptime language
- Process framework
  - Replace nondeterminism with rand
  - Equivalence based on ptime statistical tests
- Methods for establishing equivalence
  - Probabilistic simulation technique
- Emulation and compositionality
- Examples:
  Decision Diffie-Hellman, ElGamal, Bellare-Rogaway,
  Oblivious Transfer, Computational Zero Knowledge, …