

Protocol security

◆ Cryptographic Protocol

- Program distributed over network
- Use cryptography to achieve goal

◆ Attacker

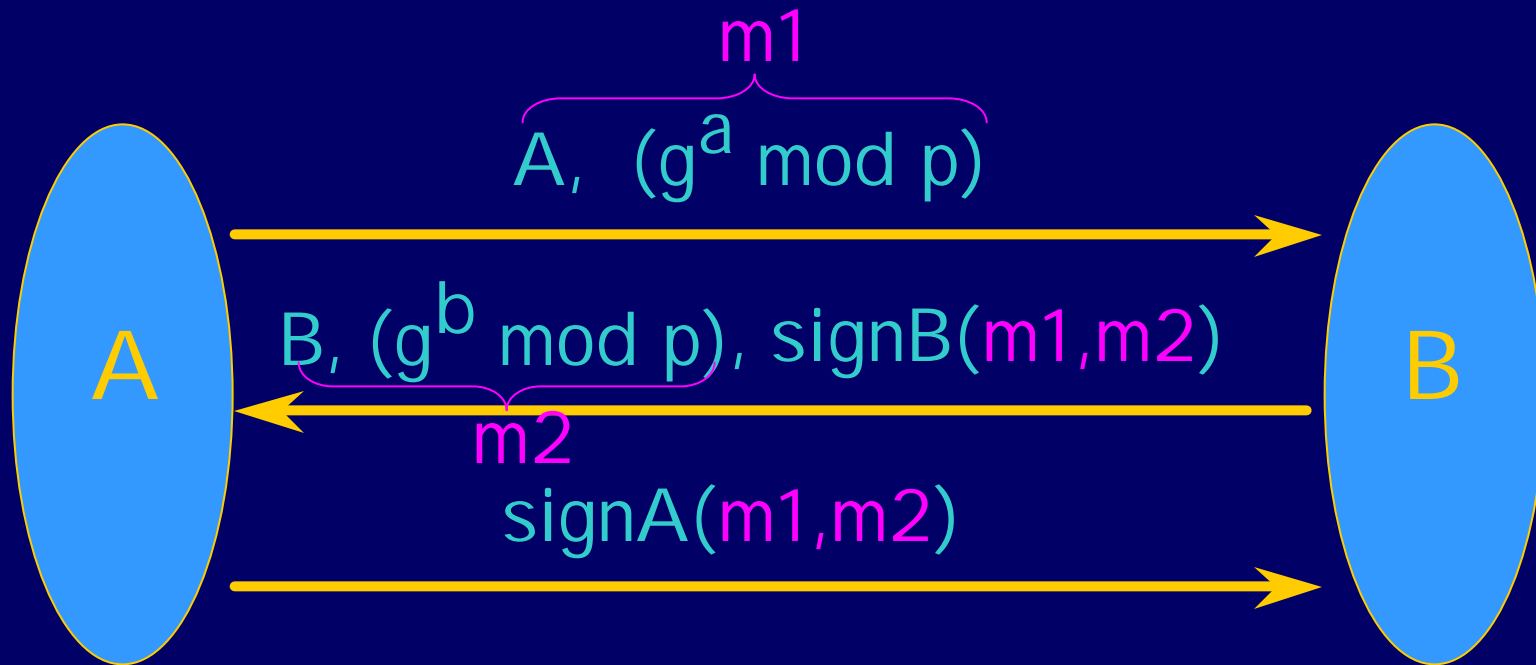
- Intercept, replace, remember messages
- Guess random numbers, some computation

◆ Correctness

- Attacker cannot learn protected secret or cause incorrect conclusion

IKE subprotocol from IPSEC

www.cba.hawaii.edu/~dave/secure/secure.html



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

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

Compositionality

◆ Confidentiality

- $A \rightarrow B$: $\text{encrypt}_{K_B}(\text{msg})$

◆ Authentication

- $A \rightarrow B$: $\text{sign}_{K_A}(\text{msg})$

◆ Composition

- $A \rightarrow B$: $\text{encrypt}_{K_B}(\text{msg}), \text{sign}_{K_A}(\text{msg})$
- Broken! $\text{sign}_{K_A}(\text{msg})$ can leak info abt. msg
- Right way: $\text{encrypt}_{K_B}(\text{msg}), \text{sign}_{K_A}(\text{cipher})$

Standard analysis methods

◆ Model-checking (finite state analysis) Easier

◆ Automated theorem provers

- Symbolic search of protocol runs
- Correctness proofs in formal logic (Dolev-Yao)

◆ Computational model

- Consider probability and complexity
 - More realistic intruder model
 - Interaction between protocol and cryptography

Harder

Outline

◆ Security protocols

➔ Research goals

◆ Specific process calculus

- Probabilistic semantics & complexity
- Asymptotic equivalence & bisimulation
- Equational proof system
- Examples
 - Computational indistinguishability
 - Decision Diffie-Hellman & ElGamal encryption

Language approach

- ◆ 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 \approx to prove security
 - Protocol is secure if no adversary can distinguish it from some idealized version of the protocol

Probabilistic poly-time process calculus

- ◆ Probabilistic polynomial-time execution model
- ◆ Specify security via equivalence to “ideal” protocol
- ◆ Also state cryptographic assumptions via equivalences
- ◆ Leads to new proof system
 - Equational reasoning
 - Based on probabilistic bisimulation, asymptotic equivalence
- ◆ Connections with modern crypto
 - Characterize computational indistinguishability
 - Formal derivation of semantic security from computational assumption DDH (both stated as equations) and vice versa (indistinguishability of encryptions)

Outline

◆ Security protocols

◆ Research goals

Specific process calculus

- ➔ Probabilistic semantics & complexity
- Asymptotic equivalence & bisimulation
 - Equational proof system
 - Examples
 - Computational indistinguishability
 - Decision Diffie-Hellman & ElGamal encryption

Syntax

Expressions have size
poly in $|n|$

◆ Bounded CCS with integer terms

$P ::= 0$	
$\text{out}(c_{q(n)}, T). P$	send up to $q(n)$ bits
$\text{in}(c_{q(n)}, X). P$	receive
$\nu c_{q(n)}. (P)$	private channel
$[T=T] P$	test
$P \mid P$	parallel composition
$!_{q(n)}. P$	bounded replication

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

Evaluation

◆ Reduction

- Evaluate unguarded terms and matches
- Local computation embodied in terms

◆ Scheduling

- Probabilistically pick a type of action

◆ Communication

- Pick a particular action of the chosen type uniformly at random
- During an actual run only pick input/output actions.

Nondeterminism vs probabilism

- ◆ Alice encrypts msg and sends to Bob

$A \rightarrow B: \{ \text{msg} \}_K$

- ◆ Adversary uses nondeterminism

Process E_0 $\text{out}(c,0) \mid \dots \mid \text{out}(c,0)$

Process E_1 $\text{out}(c,1) \mid \dots \mid \text{out}(c,1)$

Process E

$\text{in}(c, b_1) \dots \text{in}(c, b_n) . \text{out}(d, b_1 b_2 \dots b_n, \text{msg})$

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

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 schedulerseval of P halts in time $q(|n|)$

Outline

- ◆ Security protocols
- ◆ Research goals
- ◆ Specific process calculus
 - Probabilistic semantics & complexity
- ➔ Asymptotic equivalence & bisimulation
 - Equational proof system
 - Examples
 - Computational indistinguishability
 - Decision Diffie-Hellman & ElGamal encryption

How to define process equivalence?

◆ Intuition

- $|\text{Prob}\{C[P] \rightarrow o\} - \text{Prob}\{C[Q] \rightarrow o\}| < \varepsilon$

◆ Difficulty

- How do we choose ε ?
 - Less than $1/2, 1/4, \dots$? (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

$P \approx Q$ if for all polynomials p , observables $\varepsilon < 1/p(n)$

One way to get equivalences

◆ Labeled transition system

- Allow process to send any output, read any input
- Label with numbers “resembling probabilities”

◆ Probabilistic bisimulation relation

- Relation \sim on processes
- If $P \sim Q$ and $P \xrightarrow{r} P'$, then exists Q' with $Q \xrightarrow{r} Q'$ and $P' \sim Q'$, and vice versa
- Reactive form of bisimulation (scheduling)
- van Glabbeek, Smolka, Steffen '95

Outline

- ◆ Security protocols
- ◆ Research goals
- ◆ Specific process calculus
 - Probabilistic semantics & complexity
 - Asymptotic equivalence & bisimulation
- ➔ Equational proof system
 - Examples
 - Computational indistinguishability
 - Decision Diffie-Hellman & ElGamal encryption

Provable equivalences

- Assume scheduler is stable under bisimulation

$$\blacklozenge P \sim Q \Rightarrow C[P] \sim C[Q]$$

$$\blacklozenge P \sim Q \Rightarrow P \approx Q$$

$$\blacklozenge P \mid (Q \mid R) \approx (P \mid Q) \mid R$$

$$\blacklozenge P \mid Q \approx Q \mid P$$

$$\blacklozenge P \mid 0 \approx P$$

Provable equivalences

$$\blacklozenge P \approx \nu c. (\text{out}(c, T) \mid \text{in}(c, x).P) \quad x \notin \text{FV}(P)$$

$$\blacklozenge P\{a/x\} \approx \nu c. (\text{out}(c, a) \mid \text{in}(c, x).P)$$

bandwidth of c large enough

$$\blacklozenge P \approx 0 \quad \text{if no public channels in } P$$

$$\blacklozenge P \approx Q \quad \Rightarrow \quad P\{d/c\} \approx Q\{d/c\}$$

c, d same bandwidth, d fresh

$$\blacklozenge \text{out}(c, T) \approx \text{out}(c, T')$$

$$\text{Prob}[T \rightarrow a] = \text{Prob}[T' \rightarrow a] \quad \text{all } a$$

Outline

- ◆ Security protocols
- ◆ Research goals
- ◆ Specific process calculus
 - Probabilistic semantics & complexity
 - Asymptotic equivalence & bisimulation
 - Equational proof system
 - Examples
 - ➔ Computational indistinguishability
 - Decision Diffie-Hellman & ElGamal encryption

Computational indistinguishability

- ◆ $T(i,n), T'(i,n)$ terms in the calculus
 - T, T' represent uniform prob. poly-time function ensembles $f_i, g_i : \{0,1\}^i \rightarrow \{0,1\}^{q(|n|)}$
- ◆ $\text{out}(c,T) \approx \text{out}(c,T')$ says exactly that the function ensembles f_i, g_i are *indistinguishable by prob. poly-time statistical tests*
- ◆ Yao '82: fundamental notion in crypto

Outline

- ◆ Security protocols
- ◆ Research goals
- ◆ Specific process calculus
 - Probabilistic semantics & complexity
 - Asymptotic equivalence & bisimulation
 - Equational proof system
 - Examples
 - Computational indistinguishability
 - ➔ Decision Diffie-Hellman & ElGamal encryption

Connections with modern crypto

- ◆ Ciphersystem consists of three parts
 - Key generation
 - Encryption (often probabilistic)
 - Decryption
 - Formal derivation of semantic security of ElGamal from DDH and *vice versa*
 - Well known fact in crypto [Tsiounis & Yung '98]

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 $k \in \{0, \dots, p-1\}$ 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 [Tsiounis-Yung]

◆ *Formally derivable using the proof rules*

Decision Diffie-Hellman (DDH)

- ◆ Standard crypto assumption
- ◆ 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 $a, b, c \in \{0, \dots, p-1\}$
 $\langle g^a, g^b, g^{ab} \rangle \approx \langle g^a, g^b, g^c \rangle$

DDH implies sem. sec. of ElGamal

- ◆ Start with $\langle g^a, g^b, g^{ab} \rangle \approx \langle g^a, g^b, g^c \rangle$
(random a, b, c)
- ◆ Build up statement of sem. sec. from this.
 - $\text{in}(c, \langle x, y \rangle). \text{out}(c, \langle g^r, x.g^{rx} \rangle) \approx$
 $\text{in}(c, \langle x, y \rangle). \text{out}(c, \langle g^r, y.g^{rx} \rangle)$
- ◆ The proof consists of
 - Structural transformations
 - E.g., $\text{out}(c, T(r); r \text{ random}) \approx \text{out}(c, U(r))$ (any r) implies
 $\text{in}(c, x). \text{out}(c, T(x)) \approx \text{in}(c, x). \text{out}(c, U(x))$
 - Domain-specific axioms
 - E.g., $\text{out}(c, \langle g^a, g^b, g^{ab} \rangle) \approx \text{out}(c, \langle g^a, g^b, g^c \rangle)$ implies
 $\text{out}(c, \langle g^a, g^b, Mg^{ab} \rangle) \approx \text{out}(c, \langle g^a, g^b, Mg^c \rangle)$ (any M)

Sem. sec. of ElGamal implies DDH

- ◆ Harder direction. Compositionality of \approx makes 'building up' easier than breaking down.
- ◆ Want to go from
$$\text{in}(c, \langle x, y \rangle). \text{out}(c, \langle g^r, x.g^{rx} \rangle) \approx \text{in}(c, \langle x, y \rangle). \text{out}(c, \langle g^r, y.g^{ry} \rangle)$$
to
$$\langle g^x, g^r, g^{rx} \rangle \approx \langle g^x, g^r, g^c \rangle$$
- ◆ Proof idea: if $x = 1$, then we essentially have DDH.
- ◆ The proof 'constructs' a DDH tuple by
 - Hiding all public channels except the output challenge
 - Setting a message to 1
- ◆ Need structural rule equating a process with the term simulating the process
 - We use special case where process only has one public output

Current State of Project

- ◆ Compositional framework for protocol analysis
 - Precise language for studying security protocols
 - Replace nondeterminism with probability
 - Equivalence based on ptime statistical tests
- ◆ Probabilistic ptime language
- ◆ Methods for establishing equivalence
 - Probabilistic bisimulation technique
- ◆ Notion of compositionality
- ◆ Examples
 - Decision Diffie-Hellman, semantic security, ElGamal encryption, computational indistinguishability

Conclusion

◆ Future work

- Simplify semantics
- Weaken bisimulation technique to generate asymptotic equivalences
- Apply to more complex protocols
 - Bellare-Rogaway, Oblivious Transfer, Computational Zero Knowledge, ...
- Studying various models of compositionality for security protocols (WITS '04)
 - Canetti (ITMs), Pfitzmann-Waidner (IOAs)

