Probabilistic Polynomial-Time Process Calculus for Security Protocol Analysis

John Mitchell

Stanford University

P. Lincoln, M. Mitchell,

A. Ramanathan, A. Scedrov , V. Teague

Outline

u Some discussion of protocols

u Goals for process calculus

u Specific process calculus

  ¥ Probabilistic semantics

  ¥ Complexity — probabilistic poly time

  ¥ Asymptotic equivalence

  ¥ Pseudo-random number generators

  ¥ Equational properties and challenges

Protocol Security

u Cryptographic Protocol

¥ Program distributed over network

¥ Use cryptography to achieve goal

u Attacker

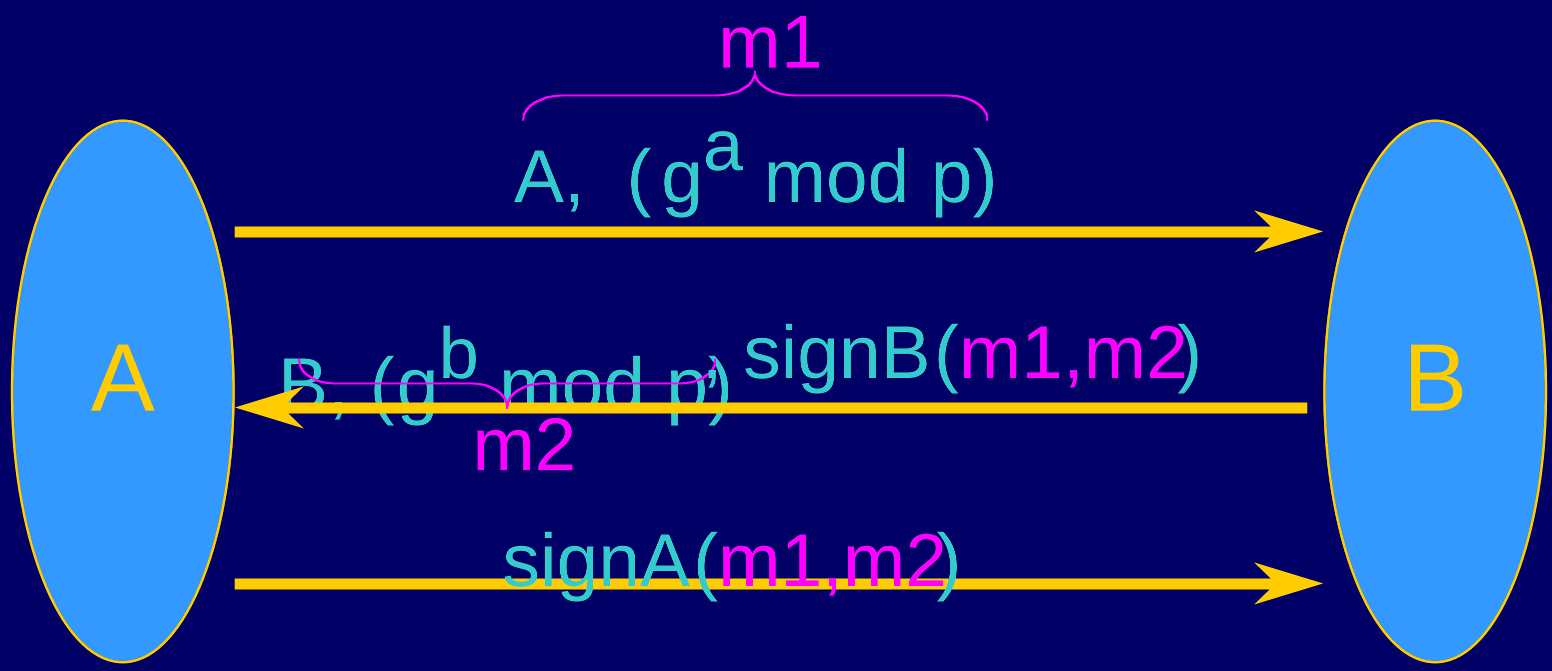
¥ Intercept, replace, remember messages

¥ Guess random numbers, do computation

u Correctness

¥ Attacker cannot learn protected secret
or cause incorrect protocol completion

IKE subprotocol from IPSEC



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

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

Simpler: Challenge-Response

u Alice wants to know Bob is listening

¥ Send ÒfreshÓ number n, Bob returns f(n)

¥ Use encryption to avoid forgery

u Protocol

¥ Alice ⟶ Bob: $\{ \text{nonce} \}_K$

¥ Bob ⟶ Alice: $\{ \text{nonce} * 5 \}_K$

u Can Alice be sure that

– Message is from Bob?

– Message is in response to one Alice sent?

Important Modeling Decisions

- How powerful is the adversary?
  - Simple replay of previous messag es
  - Decompose, reassemble and resend
  - Statistical analysis, timing attacks, ...
- How much detail in model o f crypto?
  - Assume perfect cryptography
  - Include algebraic properties
    - encr(x*y) = encr(x) * encr (y) for

      RSA encrypt(k,msg) = m sg^{k} mod N

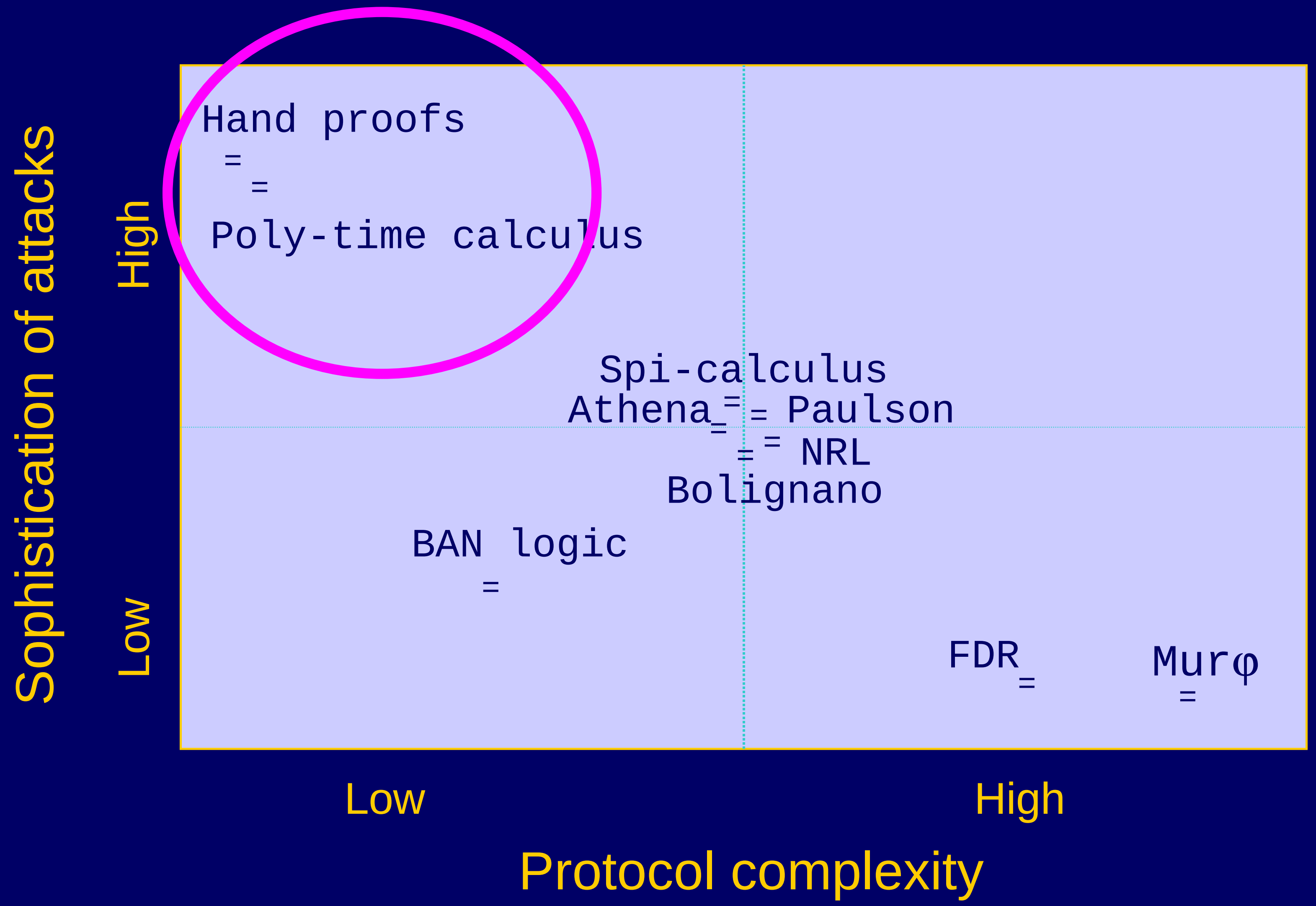
Standard analysis methods

- Finite-state analysis
- Logic based models
  - Symbolic search of protocol runs
  - Proofs of correctness in formal logic
- Consider probability and comp lexity
  - More realistic intruder model
  - Interaction between protocol and cryptography

Easy

Hard

Comparison



Sophistication of attacks
High
Low
Hand proofs
Poly-time calculus
Spi-calculus
Athena
Paulson
NRL
Bolignano
BAN logic
FDR
Murφ
Low
High
Protocol complexity

Outline

- u Some discussion of protoco ls
- Goals for process calcu lus
- u Specific process calc ulus
  - ¥ Probabilistic semantics
  - ¥ Complexity — probabilistic poly time
  - ¥ Asymptotic equivalence
  - ¥ Pseudo-random number generators
  - ¥ Equational properties and challenges

Language Approach [Abadi, Gordon]

- Write protocol in process calculus
- Express security using observational equivalence
  - Standard relation from programming language theory

    P ≈ Q iff for all contexts C[ ], same observations about C[P] and C[Q]
  - 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

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

Secrecy for Challenge-Response

- Protocol P

  $A \rightarrow B$: $\{ i \}_K$

  $B \rightarrow A$: $\{ f(i) \}_K$

- ÒObviouslyÓ secret protocol Q

  $A \rightarrow B$: { random_number }$_K$

  $B \rightarrow A$: { random_ number }$_K$

- Analysis: $P \approx Q$ reduces to crypto condition related to *non-malleability* [Dolev, Dwork, Naor]

  – Fails for RSA encryption if $f(i) = 2i$

Specification with Authentication

- Protocol P

  A → B: { random i } $_K$

  B → A: { f(i) } $_K$

  A → B: ÒOKÓ if f(i) r eceived

- ÒObviouslyÓ authenticating protoco l Q

  A → B: { random i } $_K$ public channel private channe l

  B → A: { random j } $_K$ i , j public channel private channe l

  A → B: ÒOKÓ if private i, j match public msgs

Nondeterminism vs encryption

u Alice encrypts msg and sends to Bob

¥ $A \rightarrow B$: $\{ msg \}_K$

u Adversary uses nondeterminism

¥ Process E_0 $c\langle 0\rangle \mid c\langle 0\rangle \mid$ É$\mid c\langle 0\rangle$

¥ Process E_1 $c\langle 1\rangle \mid c\langle 1\rangle \mid$ É$\mid c\langle 1\rangle$

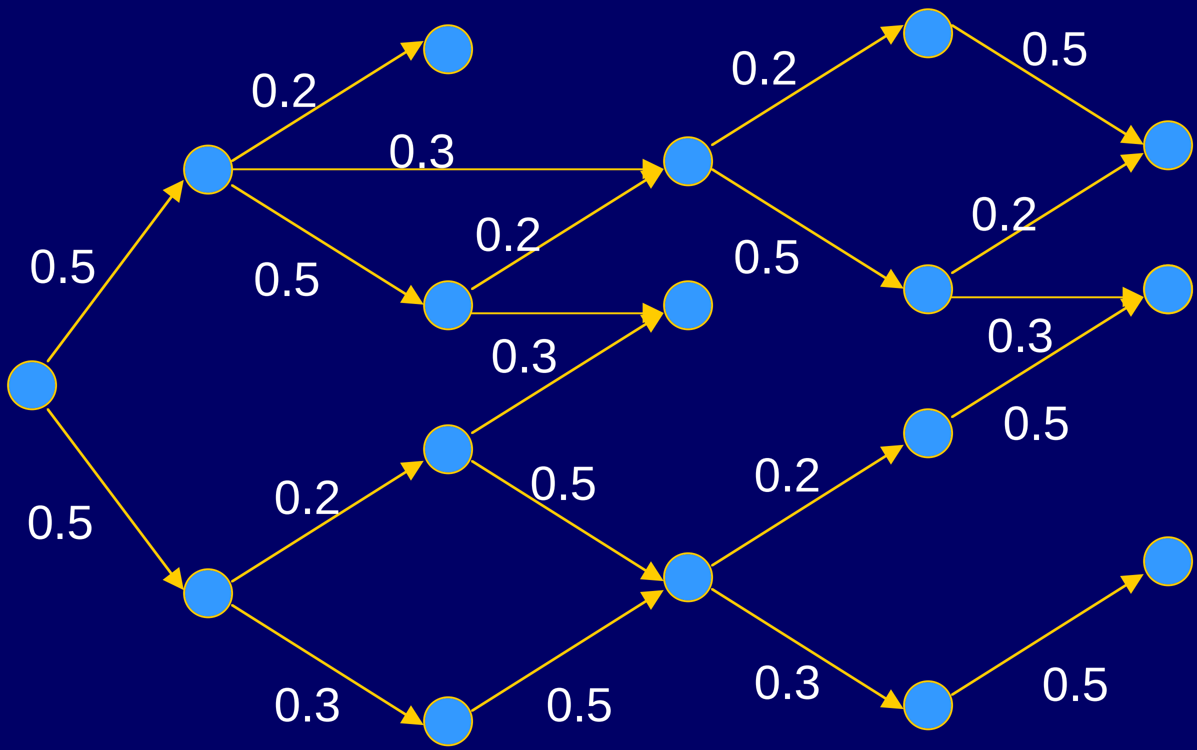
¥ Process E

$c(b_1).c(b_2)...c(b_n).decrypt(b_1 b_2 ... b_n, msg)$

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

Semantics

Probabilistic Semantics



Prove initial results for arbitrary scheduler

Methodology

u Define general system

- ¥ Process calculus
- ¥ Probabilistic semantics
- ¥ Asymptotic observational equival ence

u Apply to protocols

- ¥ Protocols have specific form
- ¥ ÒAttackerÓs context of specific form
  - — Induces coarser obser vational equivalence

This talk: general calculus and properties

Outline

u Some discussion of protoco ls

u Goals for process calcul us

Specific process cal culus

¥ Probabilistic semantics

¥ Complexity — probabilistic poly time

¥ Asymptotic equivalence

¥ Pseudo-random number generators

¥ Equational properties and challenges

Technical Challenges

- Language for prob. poly-time functions
  - Extend work of Cobham, Cook,Hofmann
- Replace nondeterminism with probability
  - Otherwise adver sary is too stro ng ...
- Define probabilistic equivalence
  - Related to poly-time statistical tests ...

Syntax

- Bounded π-calculus with integer terms

$P ::= 0$

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

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

Probabilistic Semantics

u Basic idea

- ¥ Alternate between terms and processes
  - Probabilistic evaluation of terms (incl. rand)
  - Probabilistic scheduling of parallel processes

u Two evaluation phases

- ¥ Outer term evaluation
  - Evaluate all exposed terms, evaluate tes ts
- ¥ Communication
  - Match send and receive

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
  - Choose highest-priority communication with uniform (or other) probability

Example

u Process

¥ c⟨rand+1⟩ | c(x).d ⟨x+1⟩ | d⟨ 2⟩ | d(y). e ⟨x+1⟩

u 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 _

u Communication

¥ c⟨1⟩ | c(x).d ⟨x+1⟩ | d⟨ 2⟩ | d(y). e ⟨x+1⟩

Choose according to probabilistic scheduler

Example (again)

c⟨rand+1⟩ | c(x).d ⟨x+1⟩ | d⟨ 2⟩ | d(y). e ⟨x+1⟩

Outer Eval

Each with prob 0.5

c⟨2⟩ | c(x).d ⟨x+1⟩ | d ⟨2⟩ | d(y). e ⟨x+1⟩

c⟨1⟩ | c(x).d ⟨x+1⟩ | d ⟨2⟩ | d(y). e ⟨x+1⟩

Comm Step

Choose according to probabilistic scheduler

Complexity results

- Polynomial time
  - For each process P, there is a poly q (x) such that
    - For all n
    - For all probabilistic schedulers
    - All minimal evaluation contexts C[ ]

    eval of C[P] halts in time q(|n|+|C[]|)

  - Minimal evaluation context

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: $c\langle m\rangle \mid c(x).P \rightarrow [m/x]P$
  - Substitution bounded by orig length of P
    - Size of number m is bounded

Outline

- Some discussion of protoco ls
- Application of process ca lculus
- Specific process calc ulus
  - Probabilistic semantics
  - Complexity — probabilistic poly time
  - Asymptotic equivalence
  - Pseudo-random number generators
  - Equational properties and challenges

Problem:

How to define process equivalence?

u Intuition

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

u Difficulty

¥ How do we choose ε?

— Less than 1/ 2, 1/4, É ? (not equiv relation)

— Vanishingly s mall ? As a fun ction of what?

u Solution

¥ Use security parameter

— Protocol is family $\{ P_n \}_{n>0}$ indexed by key le ngth

¥ Asymptotic form of proces s equivalence

Probabilistic Observational Equiv

- Asymptotic equivalence withi n f

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

  $P \approx_f Q$ if $\forall$ contexts C[ ]. $\forall$ obs v. $\exists n_0 . \forall n > n_0 .$

  $$| \mathrm{Prob}[C_n[P_n] \rightarrow v] - \mathrm{Prob}[C_n[Q_n] \rightarrow v] | < f(n)$$

- Asymptotically polynomial ly indistinguishable

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

Final def'n gives robust equivalence relation

Outline

u Some discussion of protoco ls

u Application of process ca lculus

u Specific process calc ulus

¥ Probabilistic semantics

¥ Complexity — probabilistic poly time

¥ Asymptotic equivalence

¥ Pseudo-random number generators

¥ Equational properties and challenges

Compare with standard crypto

- Sequence generated from random seed

  P_n: let b = n^k-*bit sequence generated from* n *random bits*

  in PUBLIC $\langle b \rangle$ end

- Truly random sequence

  Q_n: let b = *sequence of* n^k *random bits*

  in PUBLIC $\langle b \rangle$ end

- P is crypto strong pseudo-random generator

  $P \approx Q$

  Equivalence is asymptotic in security parameter n

Desired equivalences

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

u $P \mid Q \approx Q \mid P$

u $P \mid 0 \approx P$

u $P \approx Q \;\Rightarrow\; C[P] \approx C[Q]$

u $P \approx \upsilon\, c.\,(\, c<1> \mid c(x).P)$ $\quad x \notin FV(P)$

Warning: hard to get all of these É

How to establish equivalence

- u Labeled transition system
  - ¥ Allow process to send any output, r ead any input
  - ¥ Label with numbers Òresembling probabilitiesÓ
- u Simulation relation $\lesssim$
  - ¥ Relation $\lesssim$ on processes
  - ¥ If P $\lesssim$ Q and P $\xrightarrow{r}$ PÕ then exists Q Õ with Q $\xrightarrow{r}$ QÕ and PÕ $\lesssim$ QÕ
- u Weak form of prob equivalence
  - ¥ But enough to get started É

Hold for uniform scheduler

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

Problem

u Want this equivalence

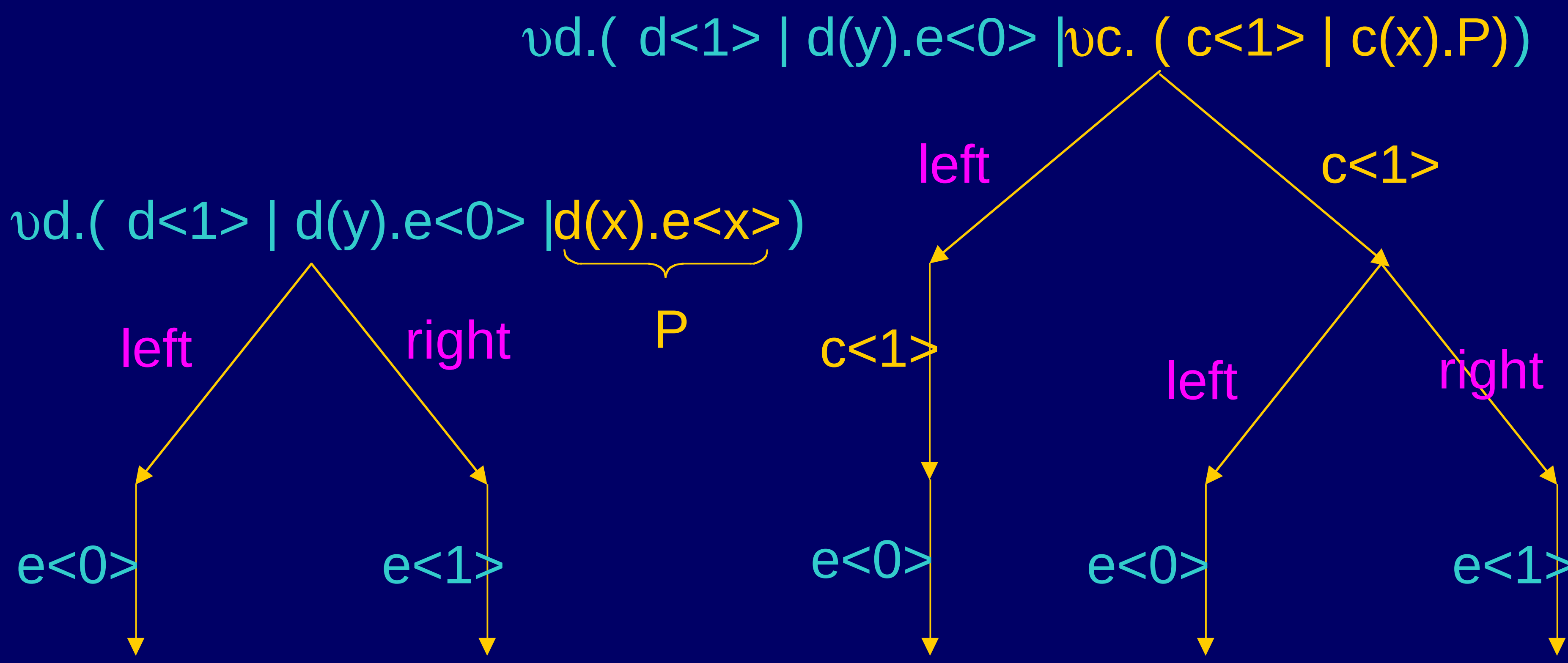
¥ P ≈ υc. (c<1> | c(x).P)   x ∉ FV(P)

u Fails for general calculus, general ≈

¥ P = d(x).e<x>

¥ C[ ] = υd.(d<1> | d(y).e<0> | [ ])

Comparison



Even prioritizing private channels, equivalence fails

Paradox

- u Two processors connect by network
- u Each does private actions
- u Unrealistic interaction
  - ¥ Private coin flip in Beijing does not influence coin flip in Washington

Solutions

u Modify scheduler

¥ Process private channels left-to-right

¥ Each channel: random send-receive p air

u Restrict syntax of protocol, attack

¥ C[P] = C[υc. (c<1> | c(x).P)]

for all contexts C[ ] that

— do not share private channels

— do not bind channel names used in [ ]

Modification of schedule r more reasonable for *protocols*

Current State of Project

u Framework for protocol analysis

¥ Determine crypto requireme nts of protocols

¥ Precise definition of crypto primitives

u Probabilistic ptime language

u Process framework

¥ Replace nondeterminism with rand

¥ Equivalence based on ptime statistical tests

u Methods for establishing equivalence

¥ Develop probabilistic simulation technique

u Examples: Diffie-Hellman, Bellare-Rogaway É

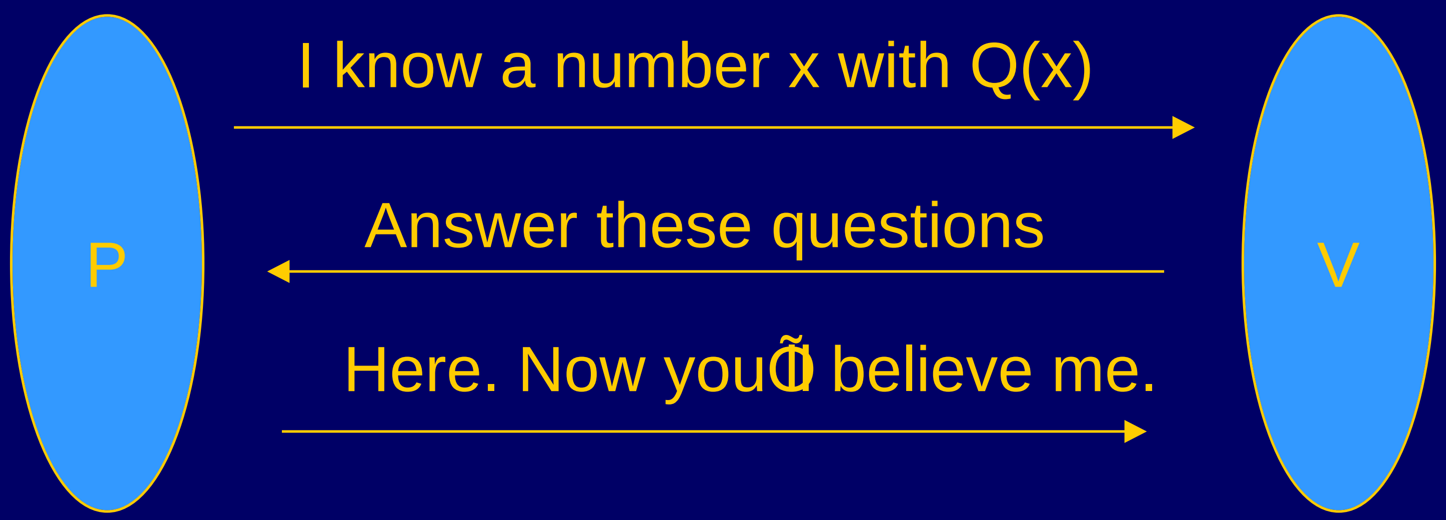
Compositionality

- Property of observational equiv

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

similarly for other process forms

Zero-Knowledge Protocol



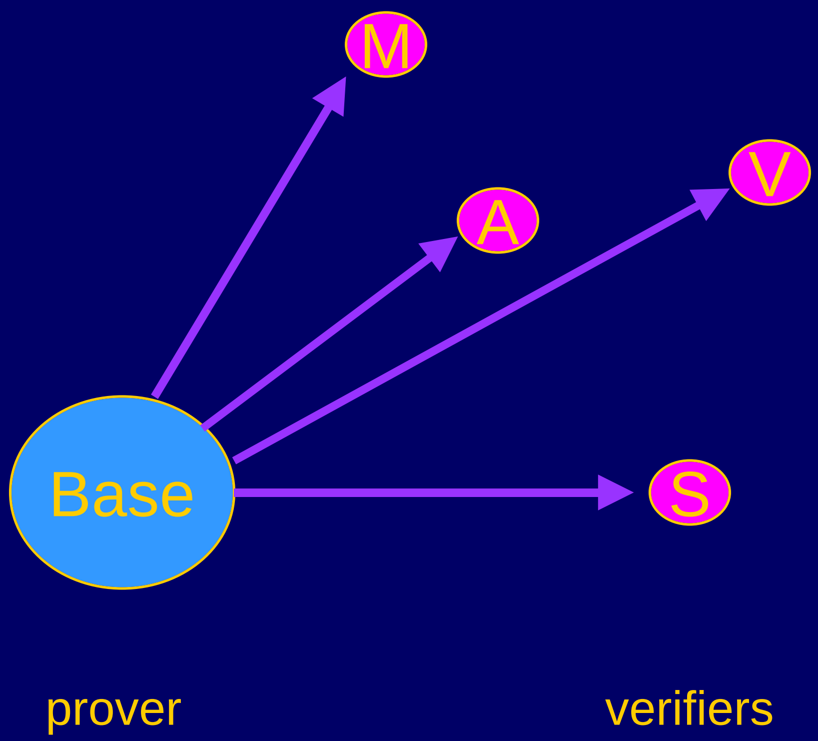
u Witness protection program

¥ Q(x) iff ∃ w. P(x,w)

¥ Prove ∃ w. P(x,w) without revealing w

Identify Friend or Foe

- Sequential
  - One conversation at a time
- Concurrent
  - Base station proves identity concurrently



Are concurrent sessions still zero-k ?