Security Protocols and Trust

A Tutorial

Joshua D Guttman

guttman@mitre.org

http://www.ccs.neu.edu/home/guttman

Coauthors: F. Javier Thayer
Jonathan C. Herzog
Lenore D. Zuck

Thanks to support from: National Security Agency
MITRE-Sponsored Research

Main Topics for Today

Cryptographic Protocol Analysis

- How to find attacks on protocols
- How to prove protocols correct

Cryptographic Protocol Design

- Crafting protocol goals for limited trust
- Engineering protocols to meet goals

Protocols and Trust Management

- Protocol analysis tells what happened
- Trust management explains how protocol actions are embedded within real world activities
  - What have I committed myself to in a run?
  - How must I trust my peers to complete a run?

The Dolev-Yao Problem

Abstract from details of cryptography

- Assume cryptographic implementation "perfect"
- Consider structural properties of protocol

Abstraction focuses attention on

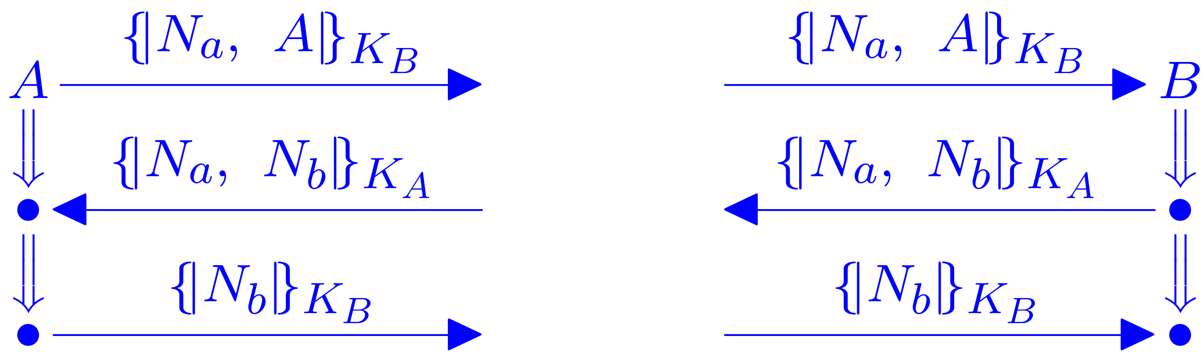
- A kind of protocol flaw
- A class of security goal
  (absence of flaws of this kind)

Suggests modeling for protocols and their security goals

Today's purpose: Describe how to

- Discover flaws (of this kind)
- Prove no flaws exist
- Design protocols without flaws

Needham-Schroeder



K_A, K_B	Public (asymmetric) keys of A, B
N_a, N_b	Nonces, one-time random bitstrings
$\{\!|t|\!\}_K$	Encryption of t with K
$N_a \oplus N_b$	New shared secret

Essence of Cryptography (for this talk)

Public key cryptography: algorithm using
two related values, one private, the other public

- Encryption: Public key makes ciphertext,
  only private key owner can decrypt
- Signature: Private key makes ciphertext,
  anyone can verify signature with public key

A's public key: K_A A's private key: K_A^{-1}

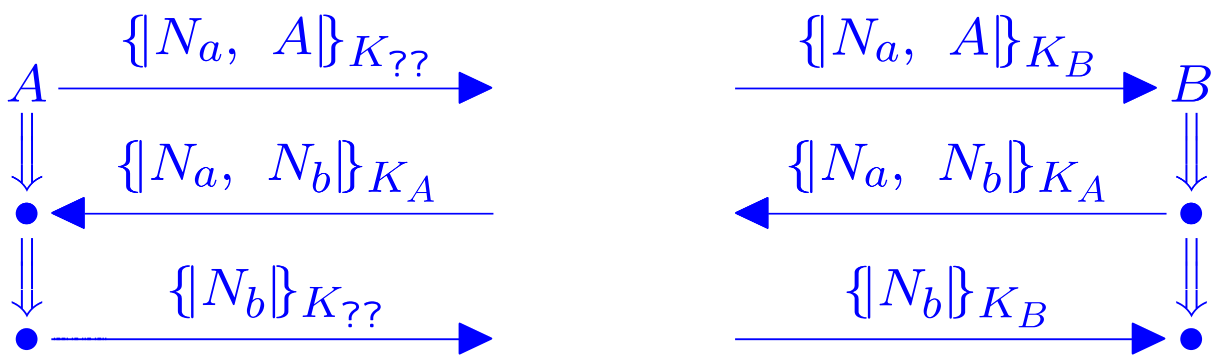
Symmetric key cryptography: algorithm using
a single value, shared as a secret between sender, receiver

- Same key makes ciphertext, extracts plaintext

$K = K^{-1}$

Needham-Schroeder: How does it work?

Assume A's private key K_A^{-1} uncompromised

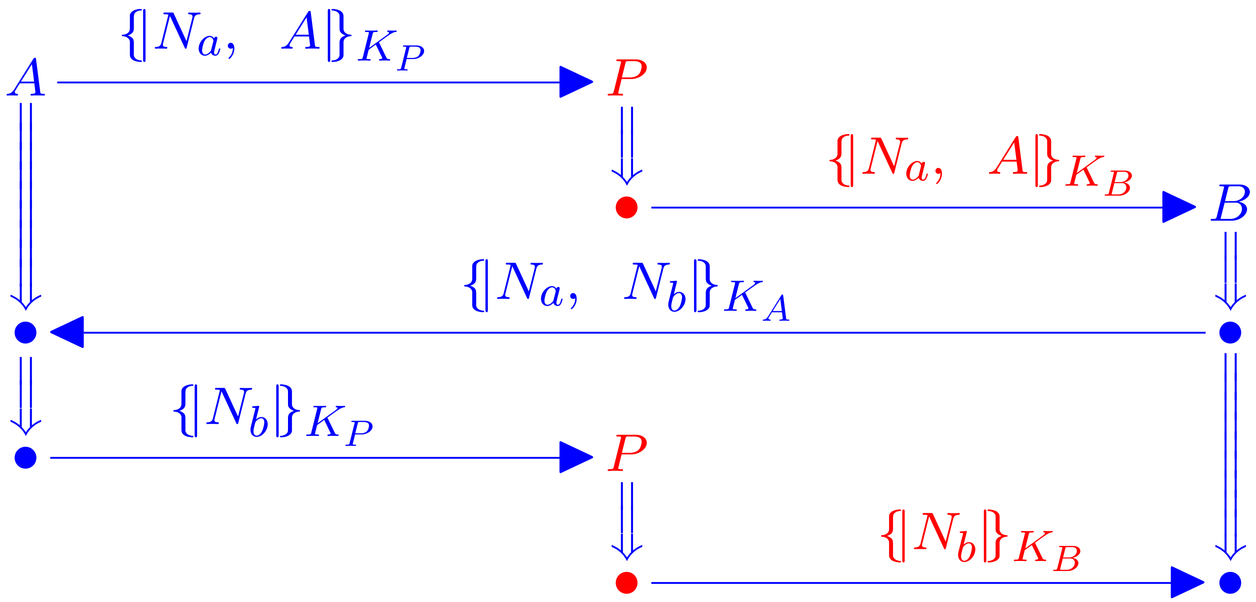


K_A, K_B	Public (asymmetric) keys of A, B
N_a, N_b	Nonces, one-time random bitstrings
$\{\!\|t\|\!\}_K$	Encryption of t with K
$N_a \oplus N_b$	New shared secret

Whoops

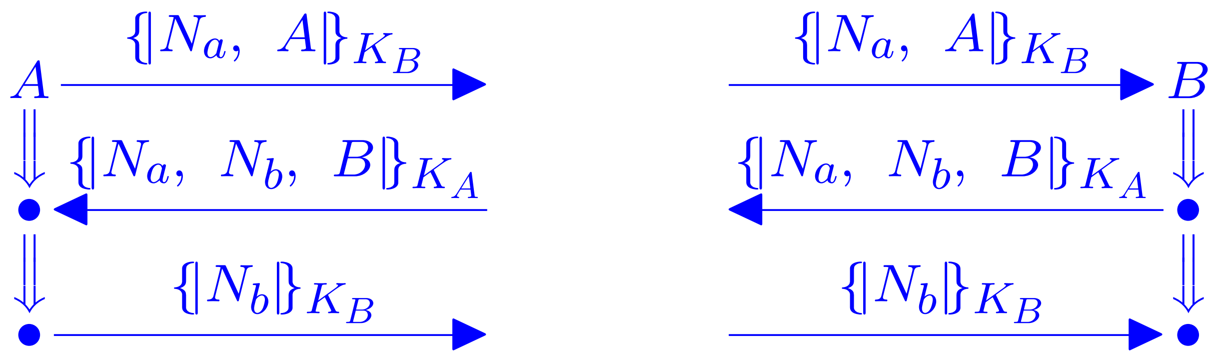
Needham-Schroeder Failure

If ?? $= P$,



(Gavin Lowe)

Needham-Schroeder-Lowe



K_A, K_B	Public (asymmetric) keys of A, B
N_a, N_b	Nonces, one-time random bitstrings
$\{\!\|t\|\!\}_K$	Encryption of t with K
$N_a \oplus N_b$	New shared secret

Protocol Executions are Bundles

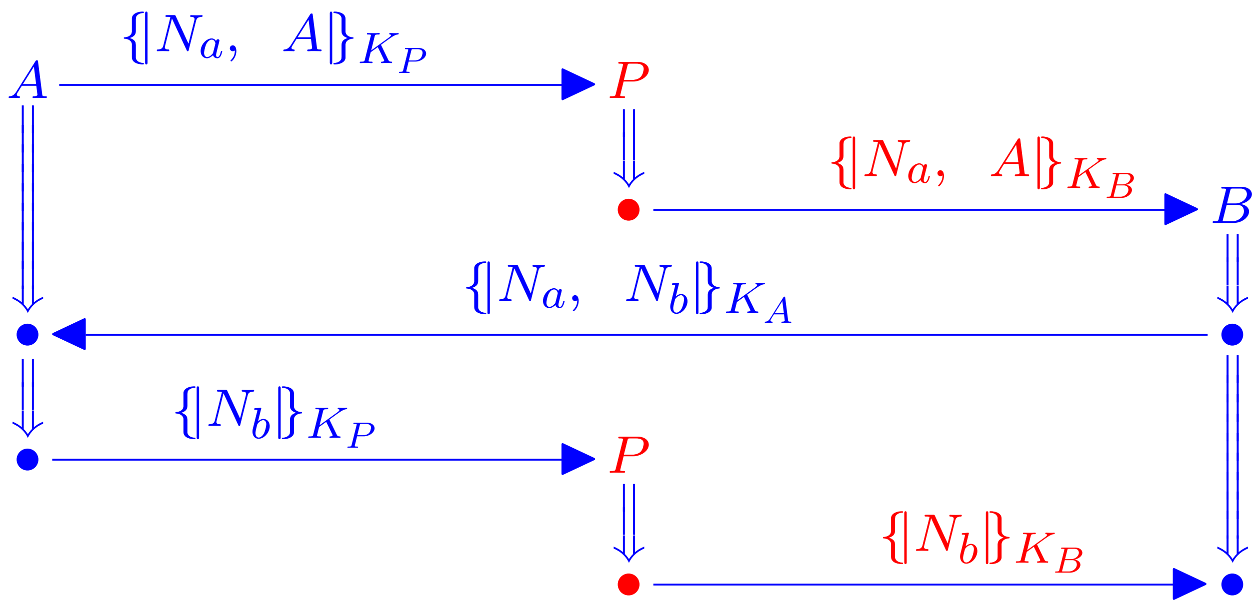
Send, receive events on strands called "nodes"

- Positive for send
- Negative for receive

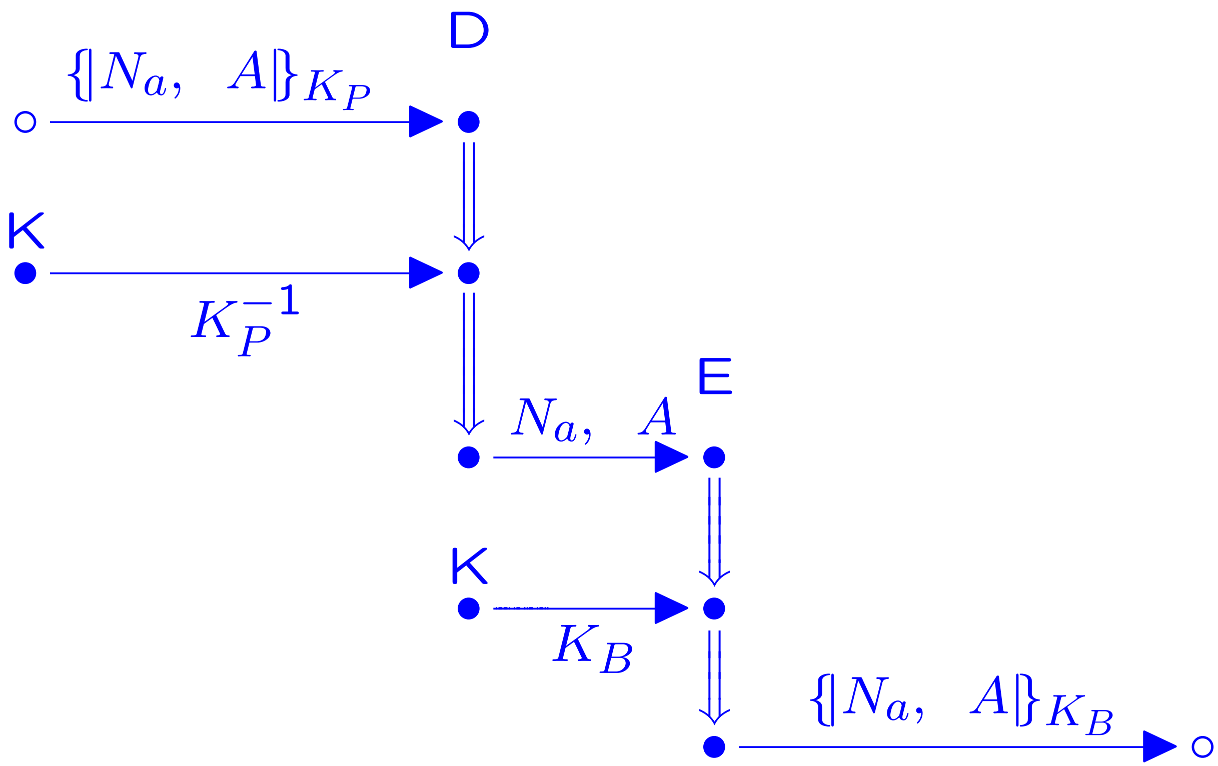
Bundle $\mathcal{B}$: Finite graph of nodes and edges representing causally well-founded execution; Edges are arrows $\rightarrow$, $\Rightarrow$

- For every reception $-t$ in $\mathcal{B}$, there's a unique transmission $+t$ where
  $+t \rightarrow -t$
- When nodes $n_i \Rightarrow n_{i+1}$ on same strand, if n_{i+1} in $\mathcal{B}$, then n_i in $\mathcal{B}$
- $\mathcal{B}$ is acyclic

A Bundle

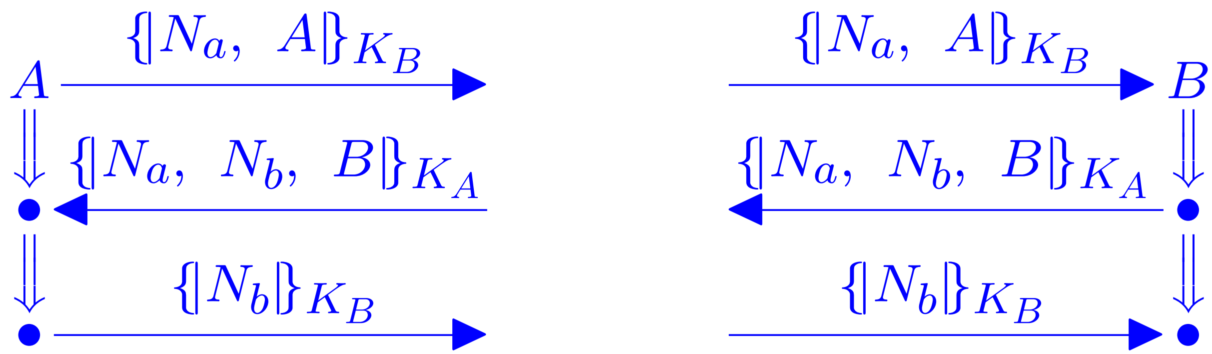


NS Attack: Adversary Activity



Bundles built from adversary strands
and regular strands

Regular Strands for NSL

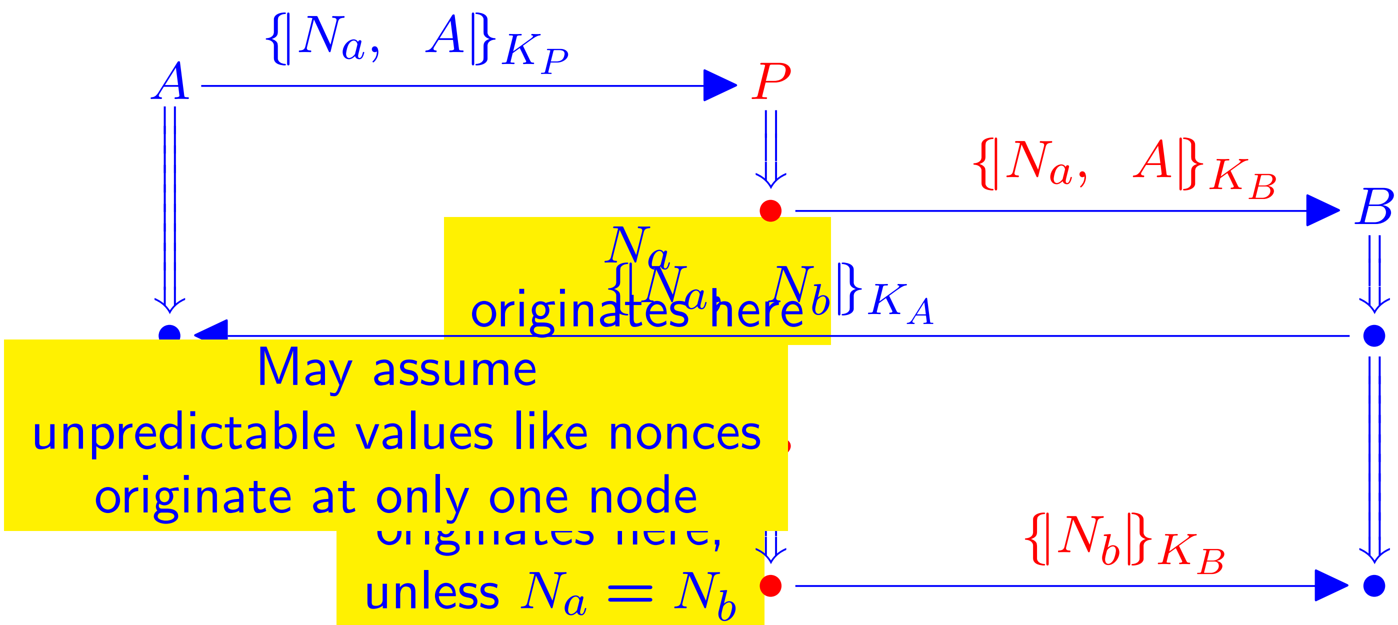


NSInit$[A, B, N_a, N_b]$

NSResp$[A, B, N_a, N_b]$

A protocol is a finite set of parametric strands, called the roles of the protocol

Origination



t originates at n if

- n positive
- t is a subterm of term transmitted: $t \sqsubset \text{term}(n)$
- $t \not\sqsubset \text{term}(m)$ if $m \Rightarrow^{+} n$

Subterms and Origination

Subterm relation $\sqsubset$
least transitive, reflexive relation with

$$g \sqsubset g, \quad h$$
$$h \sqsubset g, \quad h$$
$$h \sqsubset \{\!|h|\!\}_K$$

May assume uncompromised
private long-term keys
originate nowhere:
"Safe" keys

Note: $K \not\sqsubset \{\!|h|\!\}_K$ unless $K \sqsubset h$

Represents *contents* of message, not how it's constructed

t originates at n_1 means

n_1 is a transmission $(+)$
$t \sqsubset \text{term}(n_1)$
if $n_0 \Rightarrow \cdots \Rightarrow n_1$, then $t \not\sqsubset \text{term}(n_0)$

Unique origination, non-origination formalize probabilistic assumptions

- Unique origination expresses nonce properly chosen
- Non-origination expresses long-term key uncompromised (reason for defn of subterm)

A Secrecy Goal

Suppose:

- Bundle $\mathcal{B}$ contains a strand Resp$[A, B, N_a, N_b]$
- K_A^{-1}, K_B^{-1} non-originating
- N_b originates uniquely in $\mathcal{B}$

Then:

- There is no node $n \in \mathcal{B}$ with term$(n) = N_b$

Form: $\forall$. This is false for NS, true for NSL

To prove secrecy:	(1)		Non-originating values are safe
	(2)	If	a originates, but on regular strand, always inside $\{\!|\ldots a \ldots|\!\}_K$ with K^{-1} safe
		then	a also safe

(1),(2) inductively define Safe (relative to $\mathcal{B}$)

An Authentication Goal

Suppose:

- Bundle $\mathcal{B}$ contains a strand Resp$[A, B, N_a, N_b]$
- K_A^{-1} non-originating
- N_b originates uniquely in $\mathcal{B}$
- $N_b \neq N_a$

Then:

- There is a strand Init$[A, B, N_a, N_b]$ in $\mathcal{B}$

Authentication: correspondence assertions (of form $\forall\exists$)
This is false for NS: Only have

$$\text{Init}[A, X, N_a, N_b] \text{ in } \mathcal{B}$$

for some X

Precedence within a Bundle

Bundle precedence ordering $\preceq_{\mathcal{B}}$

$n \preceq_{\mathcal{B}} n'$ means sequence of 0 or more arrows $\rightarrow$, $\Rightarrow$ lead from n to n'

$\preceq_{\mathcal{B}}$ is a partial order by acyclicity

$\preceq_{\mathcal{B}}$ is well-founded by finiteness

Bundle induction: Every non-empty subset of $\mathcal{B}$ has $\preceq_{\mathcal{B}}$-minimal members

Reasoning about protocols combines

- Bundle induction
- Induction on message structure

Occurring Within

S is a set of terms

a occurs only within S in t means

- in abstract syntax tree of t
  every branch leading to a through subterms
  traverses some $t_0 \in S$ before reaching it

a occurs outside S in t means

- $a \sqsubset t$ but
  a does not occur only within S in t

S offers export protection means

- $t_0 \in S$ implies
  t_0 has form $\{\!|h|\!\}_K$ where $K^{-1} \in \mathsf{Safe}$

Only regular strands get a out through export protection

Outgoing Authentication Test

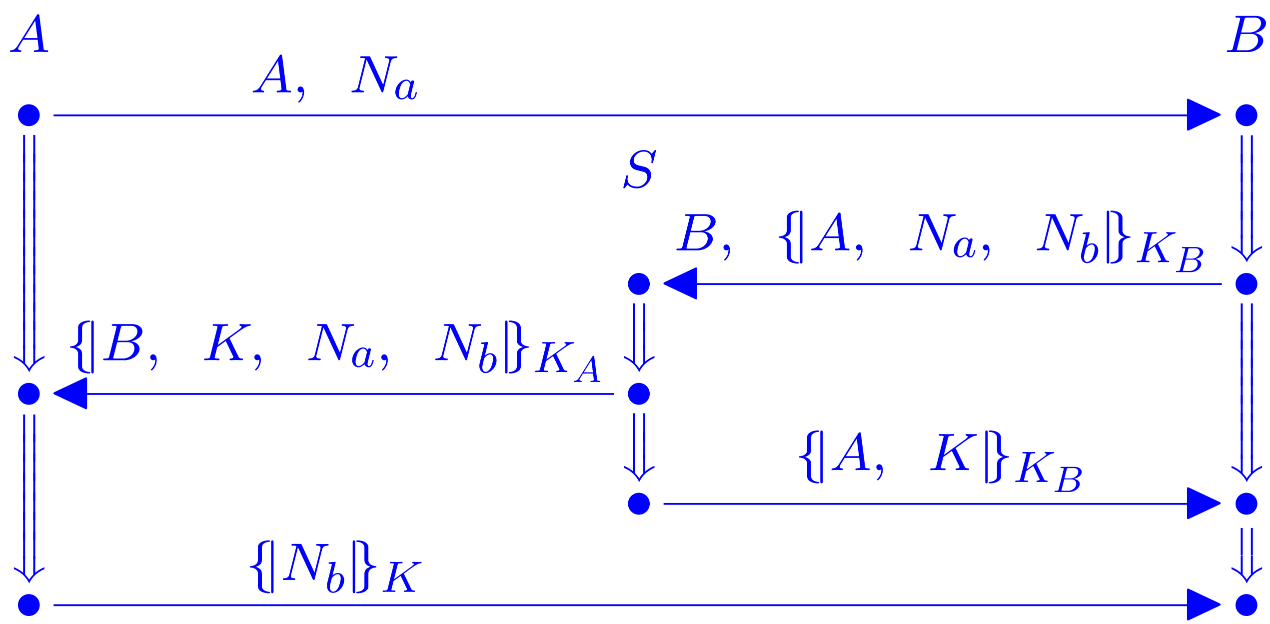
$$\begin{tikzcd}
m_0 \arrow[r, dashed, "a \text{ occurs only within } S"] \arrow[d, Rightarrow] & n_0 \arrow[d, Rightarrow] \\
m_1 & n_1 \arrow[l, dashed, "a \text{ occurs outside } S"']
\end{tikzcd}$$

Assume	a originates uniquely at m_0
	a occurs only within S in term(m_0)
	a occurs outside S in term(m_1)
	S offers export protection
Conclude	nodes n_0, n_1 exist in $\mathcal{B}$ and are regular
	a occurs outside S in n_1
	$m_0 \prec n_0 \prec n_1 \prec m_1$

Useful for proving recency

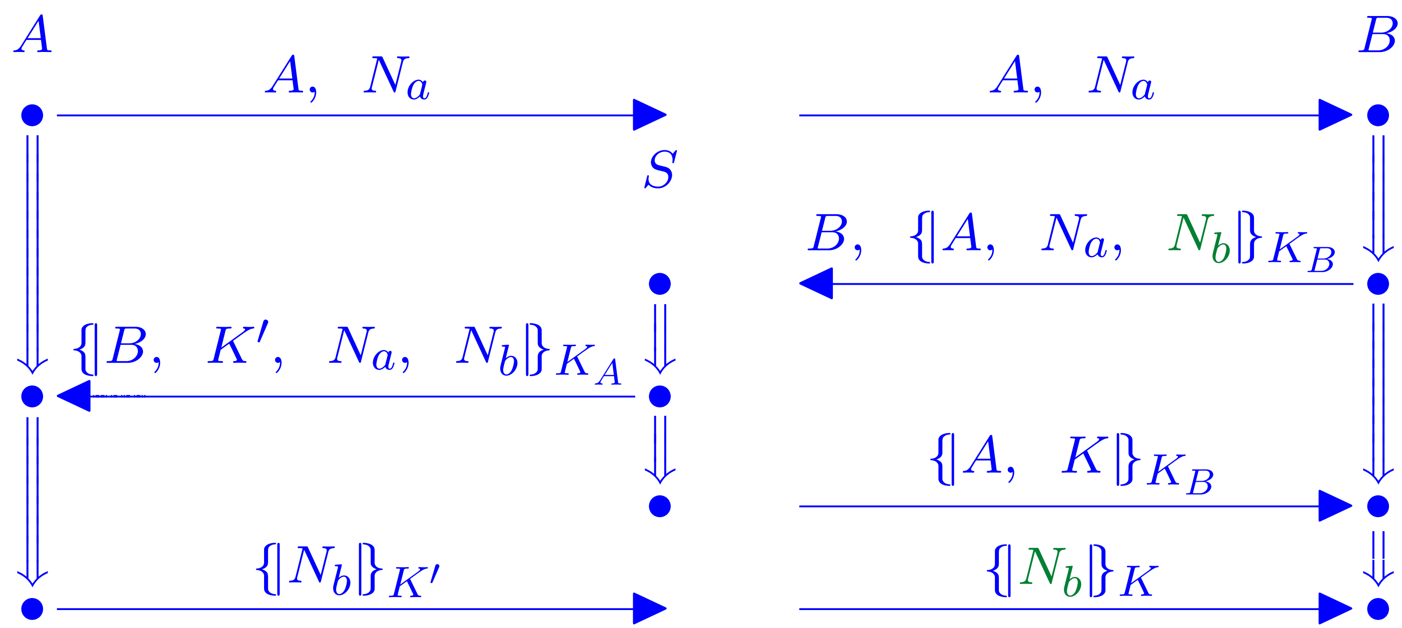
Useful because typically few regular candidates for n_0, n_1

An Example: Yahalom's Protocol



Slightly modified: $\{\!|A,\ K|\!\}_{K_B}$ not forwarded via A

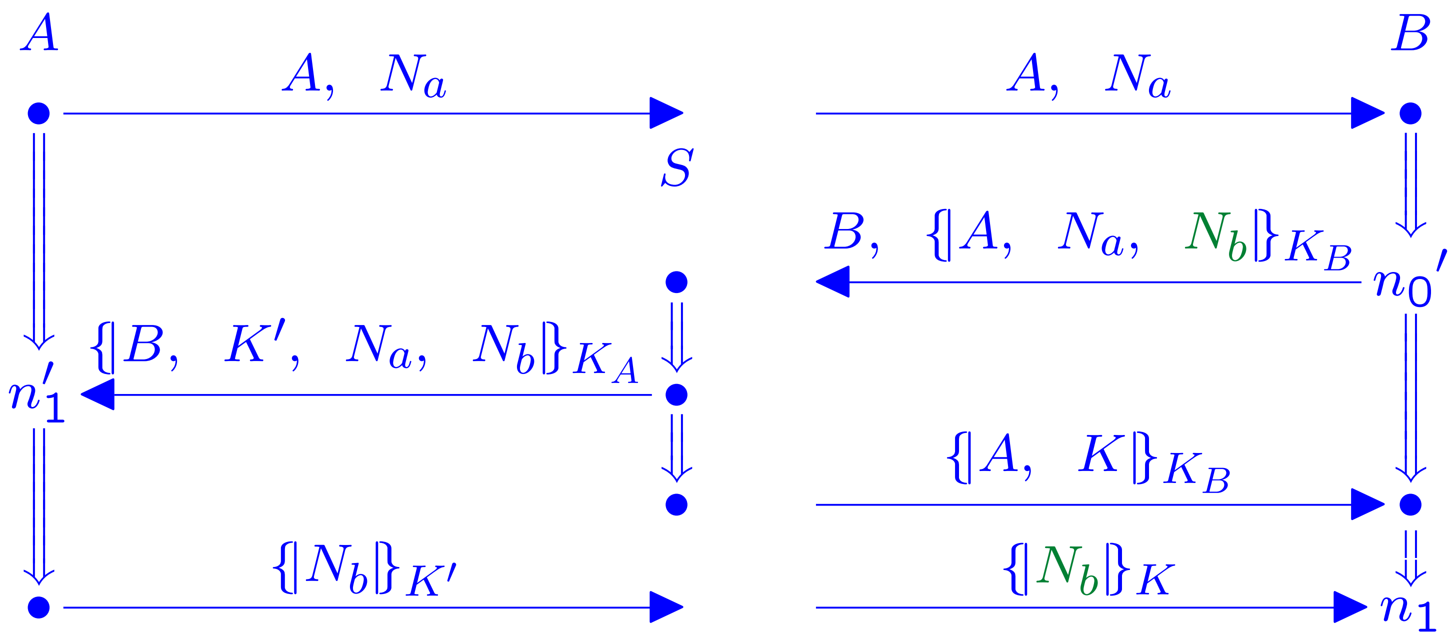
Yahalom Responder's Guarantee: Idea



Does $K' = K$?

Otherwise, must be another transforming edge,
but no regular strand can transform $\{\!|N_b|\!\}_{K'}$ into $\{\!|N_b|\!\}_K$

Yahalom Responder's Guarantee



$$S_1 = \{\{\!|B, \ K', \ N_a, \ N_b|\!\}_{K_A} : K' \text{ is a key}\} \cup \{\{\!|A, \ N_a, \ N_b|\!\}_{K_B}\}$$

$$S_2 = \{\{\!|A, \ N_a, \ N_b|\!\}_{K_B}\}$$

Either $K = K'$ or $K \neq K'$

Import Protection

S offers import protection means

- $t_0 \in S$ implies
  t_0 has form $\{\!|h|\!\}_K$ where $K \in \mathsf{Safe}$

Only regular strands get
a in through import protection

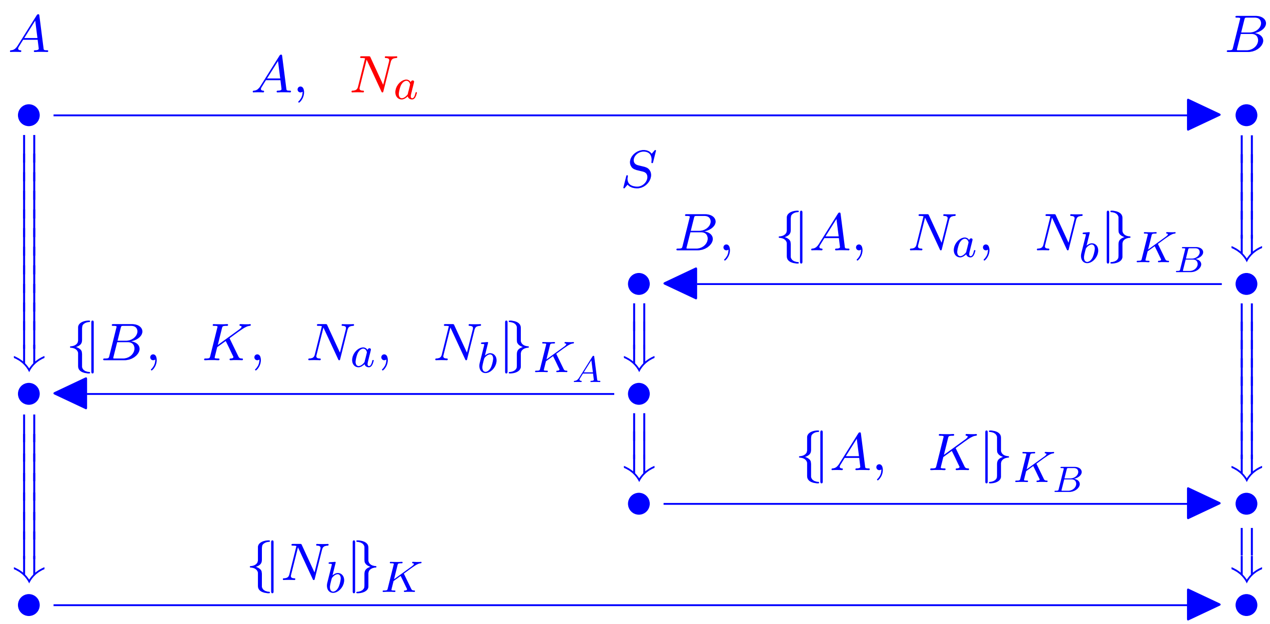
Incoming Tests

$$
\begin{tikzcd}[column sep=huge, row sep=large]
m_0 \arrow[r, dashed, "a \text{ originates uniquely}"] \arrow[d, Rightarrow] & n_0 \arrow[d, Rightarrow] \\
m_1 & n_1 \arrow[l, dashed, "\{\!|h|\!\}_K \quad K \in \mathsf{Safe} \quad \{\!|h|\!\}_K"']
\end{tikzcd}
$$

Assume	$S = \{\{\!|h|\!\}_K\}$ offers import protection
Conclude	n_1 exists in $\mathcal{B}$ and is regular

If also	$a \sqsubset h$ originates uniquely at m_0
and	$\{\!|h|\!\}_K \not\sqsubset \mathsf{term}(m_0)$
then	$m_0 \prec n_0 \Rightarrow^+ n_1 \prec m_1$

Yahalom Initiator Guarantee



The Protocol Design Problem

Specific real-world tasks interweave

- Authentication
- Access control or trust determination
- Agreement on data (request or reply)

Desirable to be able to hand craft a protocol for task

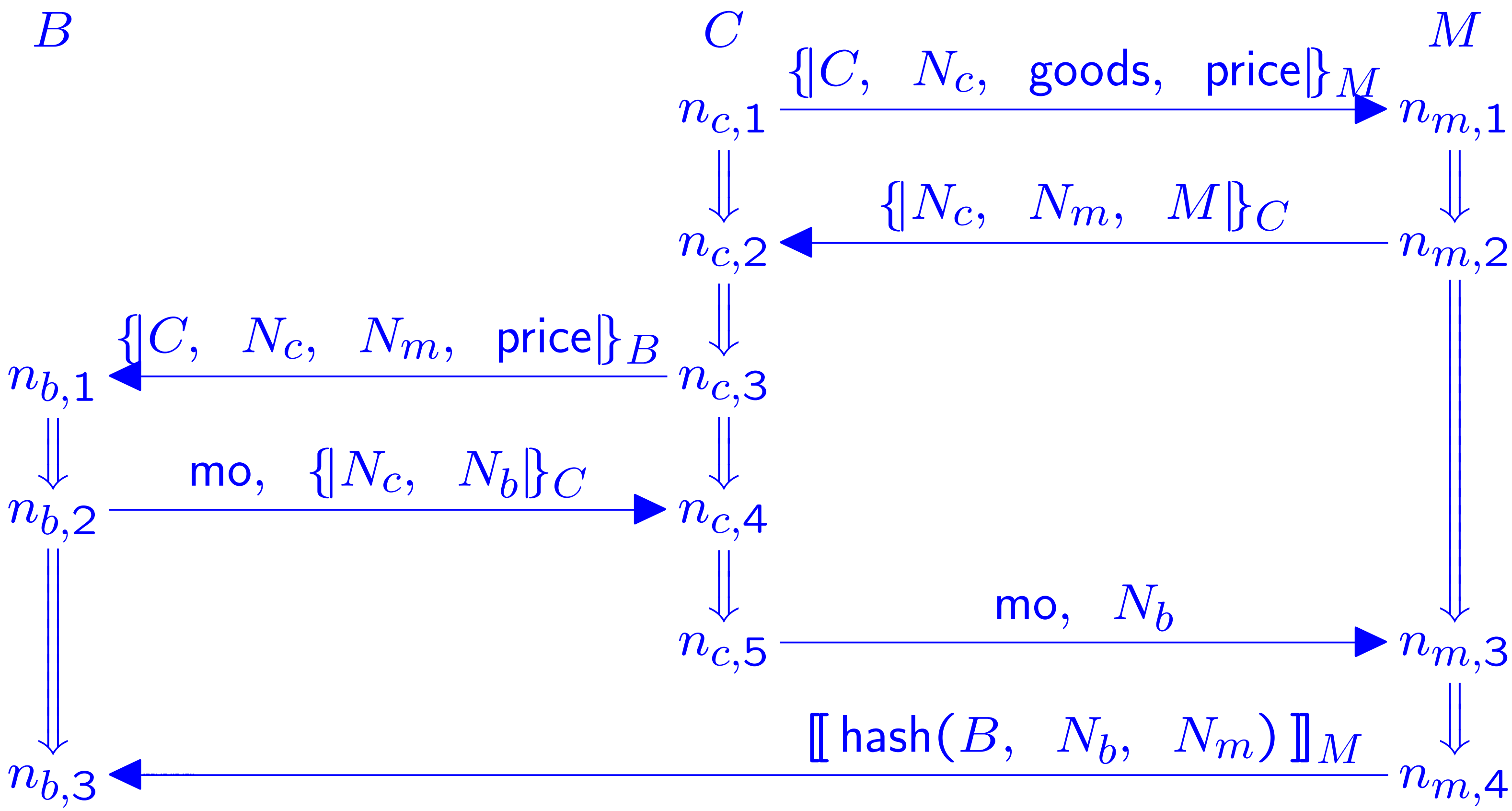
An Example: Electronic Purchase with a Money Order

- Participants: Customer, Merchant, Bank
- C, M have accounts at B
- C will get money order, B puts "hold" on account
- B transfers funds when M redeems money order

Security goals

- C, M mutual authentication, agree on B, price, goods
- Confidentiality for parameters
- B learns M only if transaction completes, does not learn goods

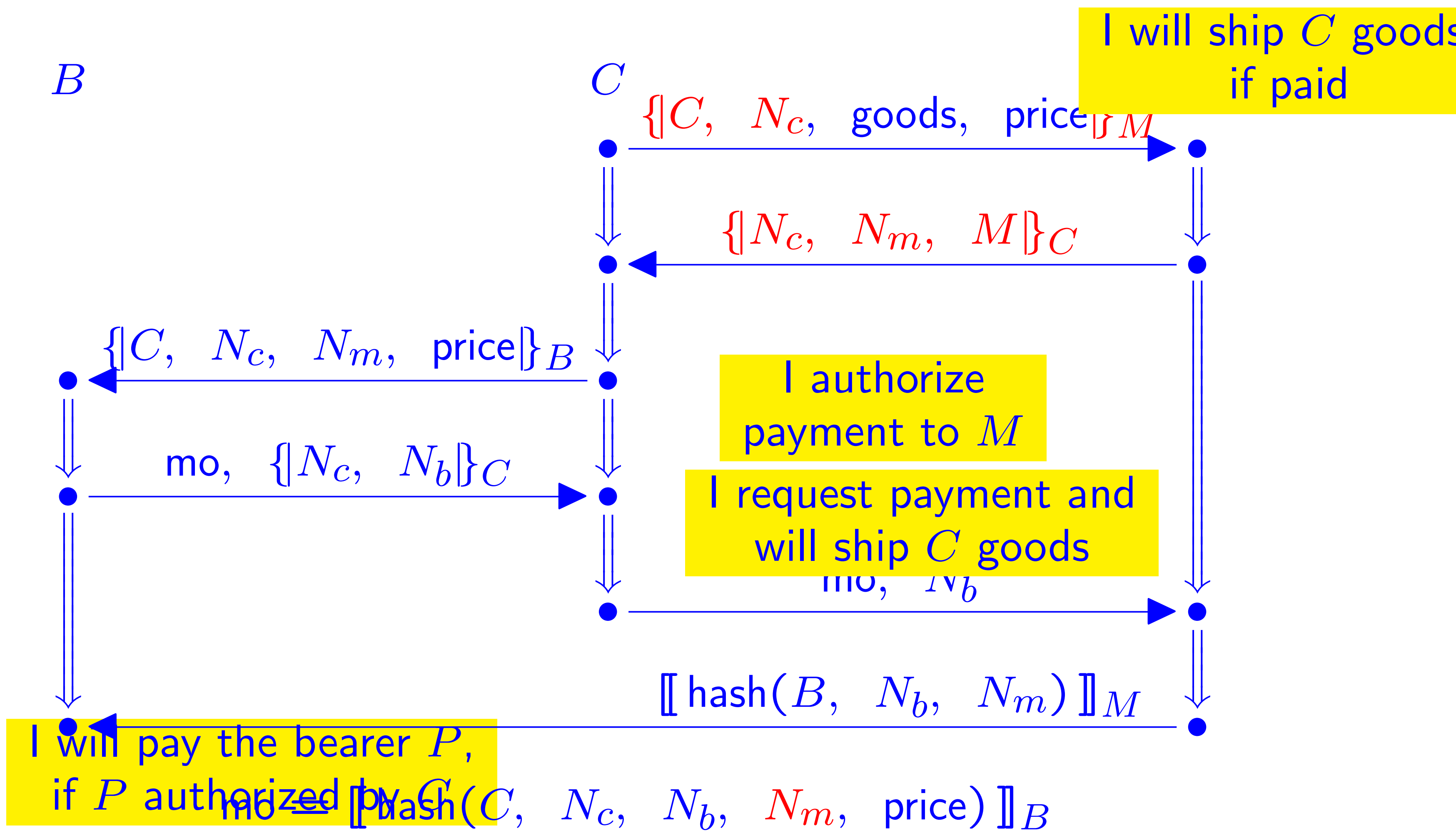
A Solution: EPMO



Electronic Purchase using Money Order

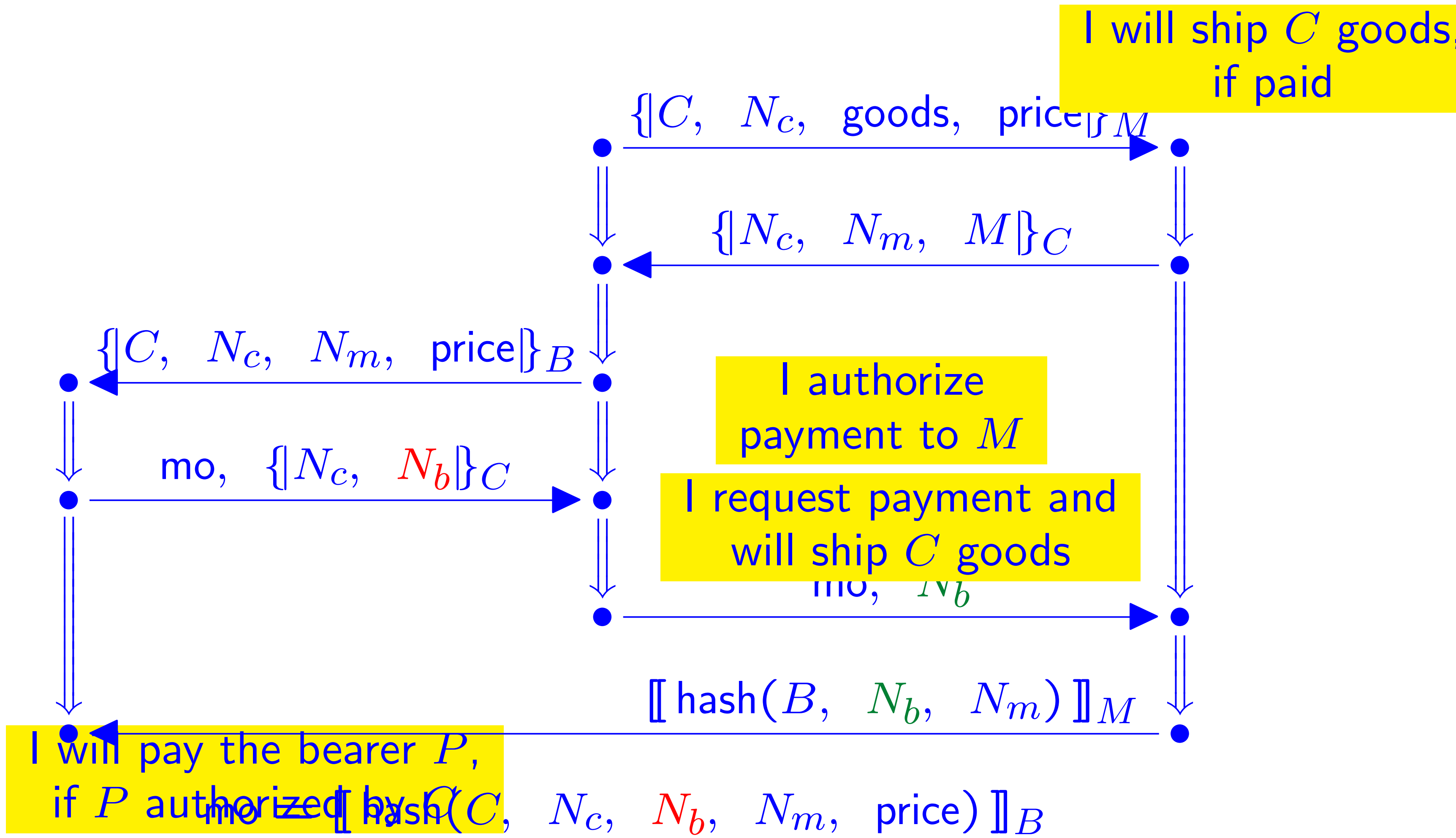
$\mathsf{mo} = [\![\,\mathsf{hash}(C,\ N_c,\ N_b,\ N_m,\ \mathsf{price})\,]\!]_B$

EPMO and Needham-Schroeder-Lowe



Outgoing tests achieve agreement between C and M

EPMO and the Bank



Outgoing test authenticates C to B
Incoming tests authenticate M, B

Protocol Design

Incoming and outgoing tests are a strong heuristic

- Suggest design for special-purpose protocols
- Lead to provably correct results
- Rapid, well-constrained design process

Trust and Protocols

Reason about real world consequences of cryptographic protocols

- Capitalize on methods for protocol analysis and design

Examples:

- Distributed access control
  - Principals cooperate to share resources selectively
  - As formulated via trust management logic
- Electronic retail commerce
  - When is customer committed to paying?
  - When is merchant committed to shipping?
  - Whose word did you depend on when deciding?

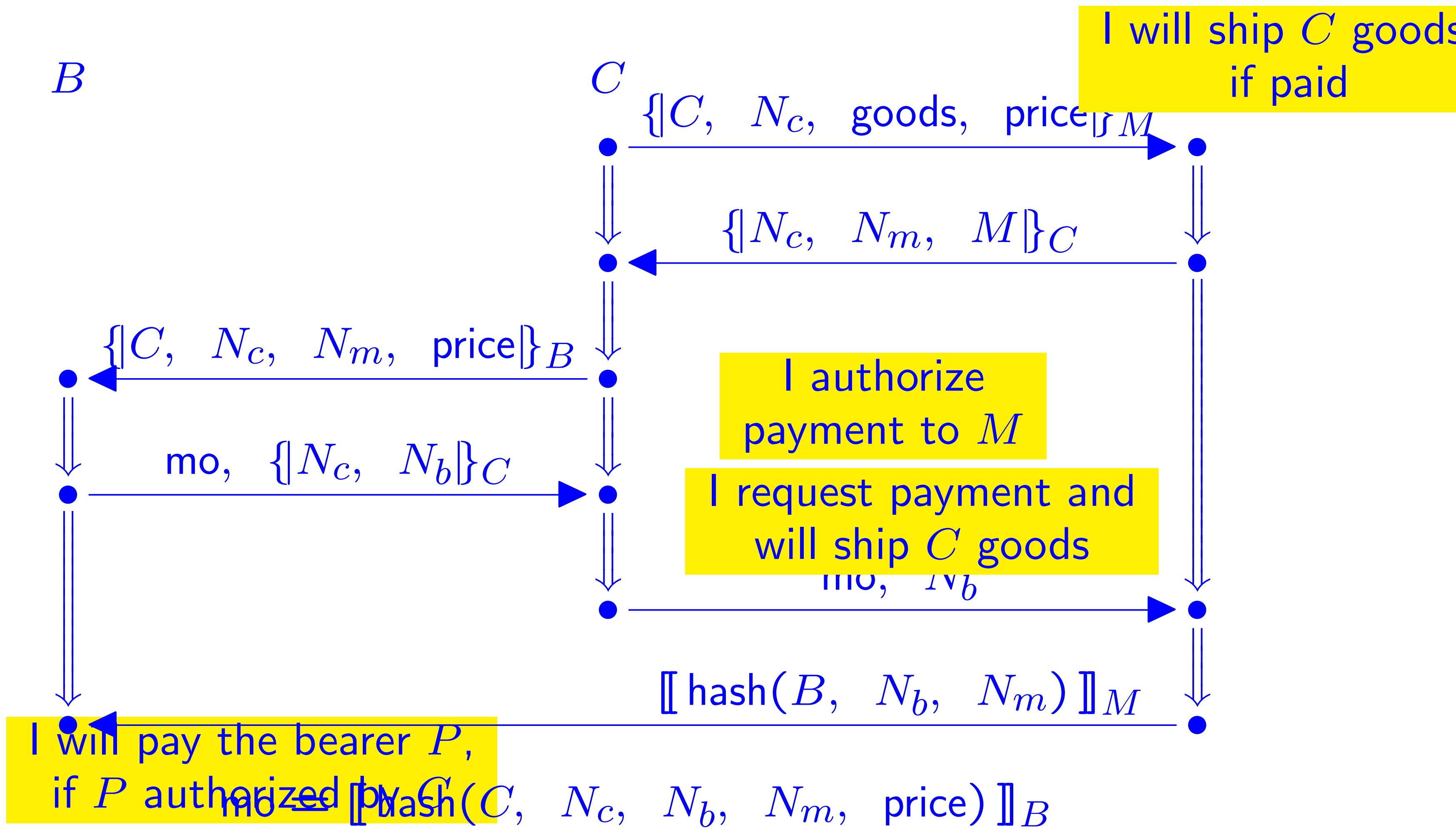
control access
(or actions) via distribu
logical deduction

Example: EPMO

Remainder of talk: Enrich strand space framework with formulas from a trust management logic

- Formulas for message transmissions are guaranteed by sender
- Formulas for message receipt are assumptions the receiver relies on

EPMO: Commitments on sends



Trust management and protocols

Each principal P

- Reasons locally in Th_P
- Derives guarantee before transmitting message
- Relies on assertions of others as premises

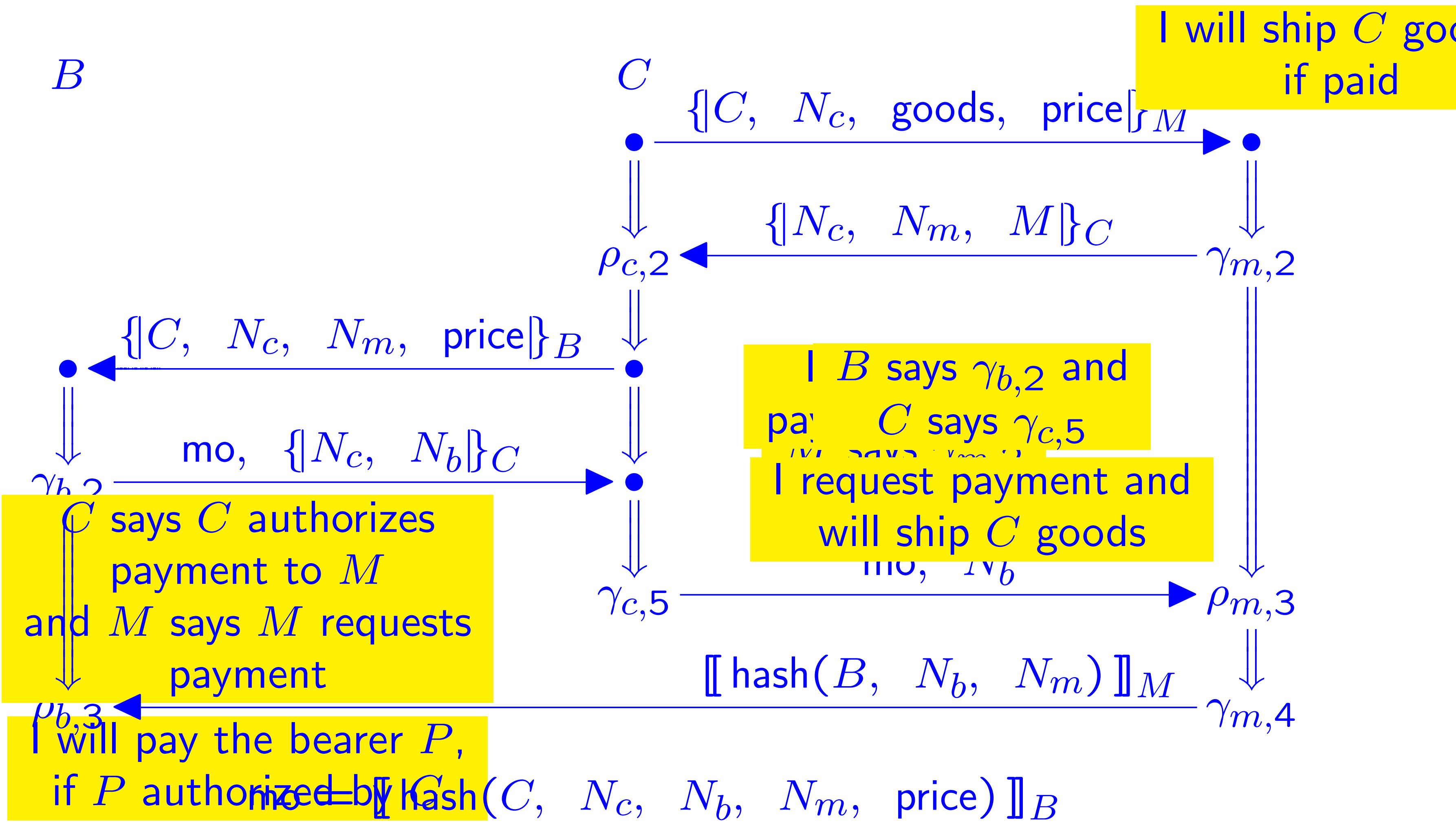
Premises: formulas associated with message receptions

- Specifies what recipient may rely on, e.g.
  "B says 'I will transfer funds if authorized' "
- Provides local representation of remote guarantee
- Th_P determines whether ϕ follows from P' says ϕ

Role of protocol

- When I rely on you having asserted a formula,
  then you did guarantee that assertion
- Coordination mechanism for rely/guarantees
- Sound protocol: "relies" always backed by "guarantees"

EPMO: Rely/Guarantee Formulas



Contrast: Earlier Work

The BAN tradition

- Messages are formulas or formulas idealize messages
- Who asserted the formulas?
- Who drew consequences from formulas?

Embedding formulas explicitly inside messages

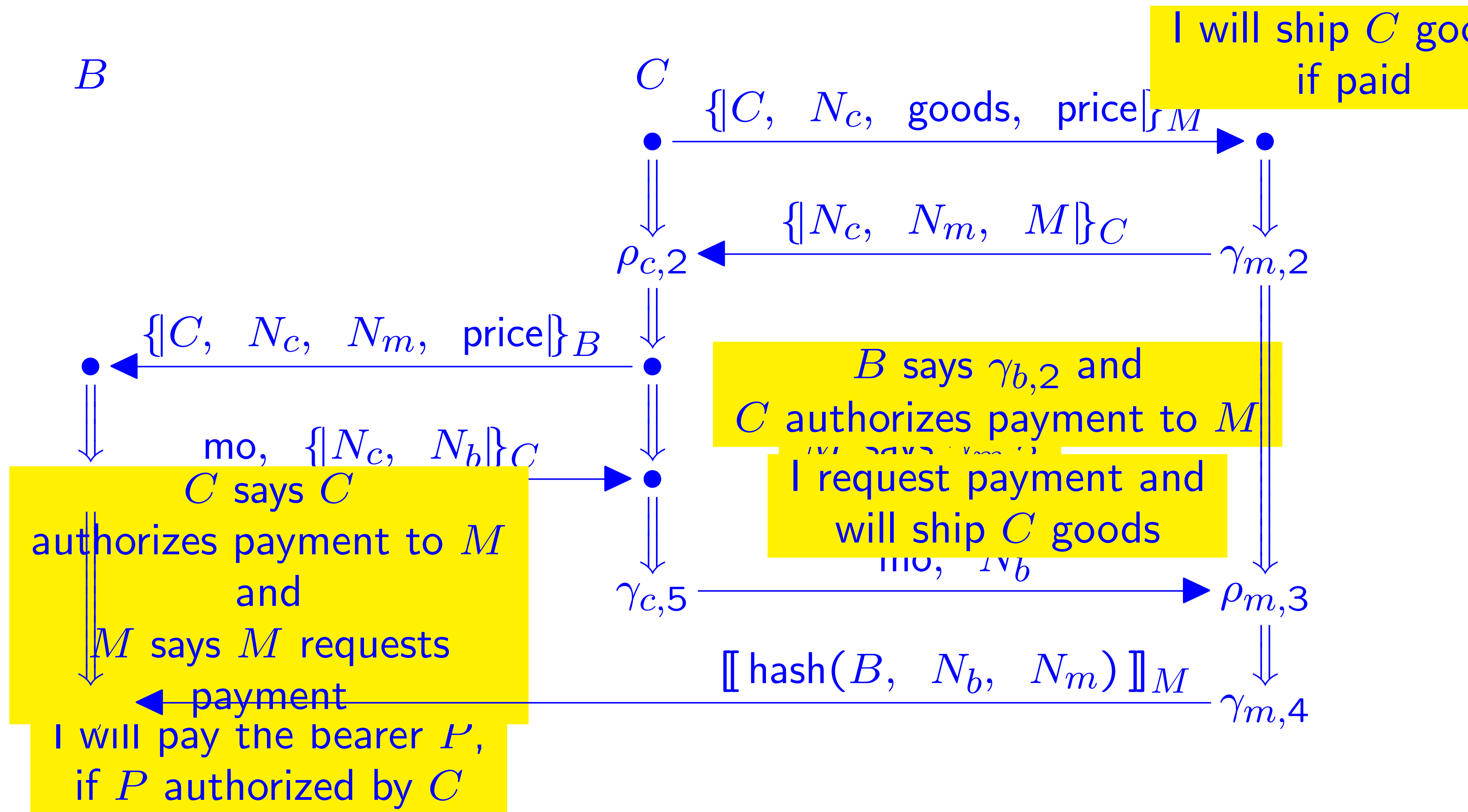
starts
with LAWB

- Main view of logical trust mgt
- Formulas parsed out of certificates
- Problem of partial information?

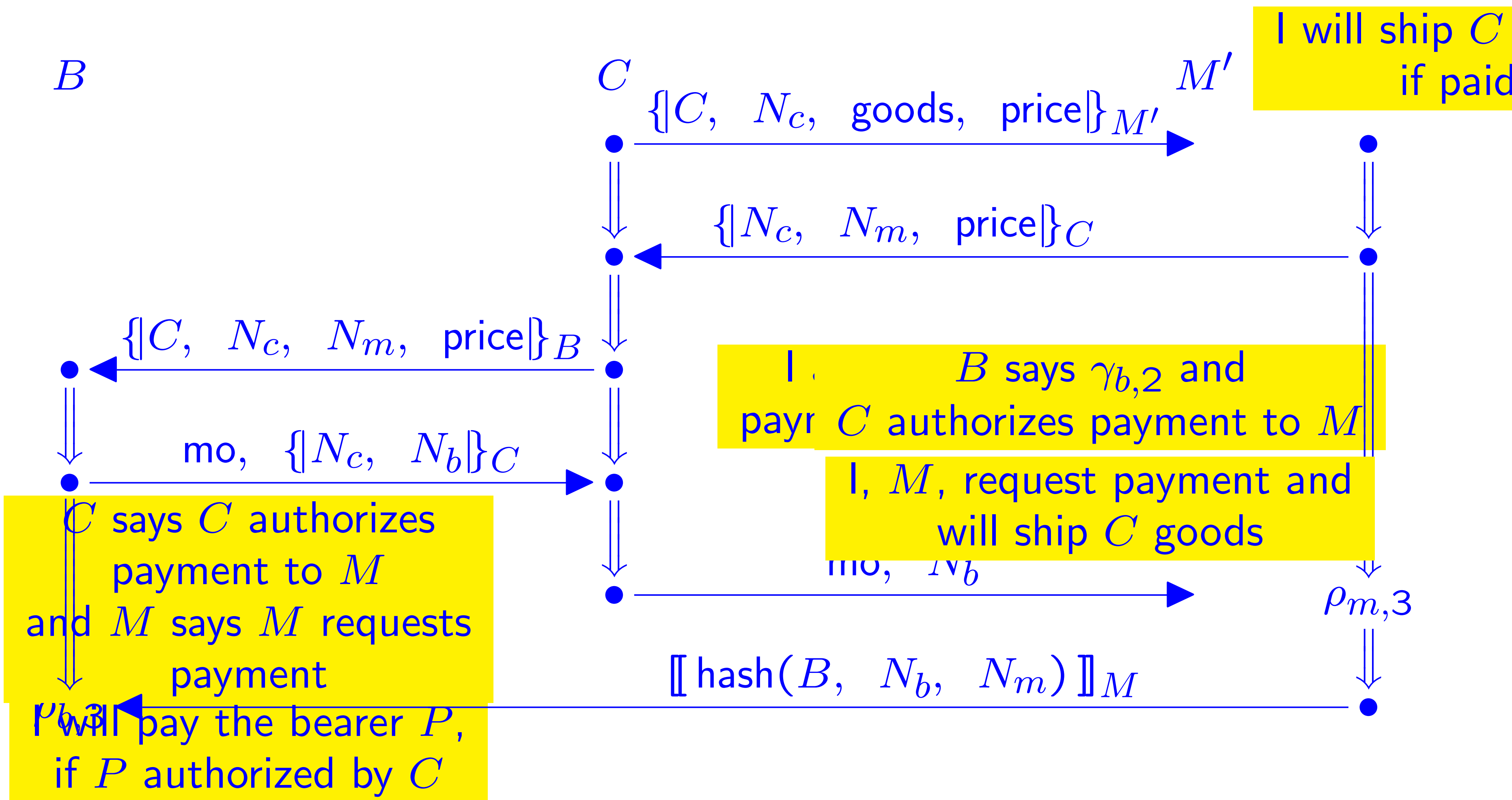
Our view: Formulas part of transmission/reception, not msg

- Compatible with many insights of earlier views
- Independent method to determine what events happened
- Clarity about who makes assertions, who infers consequences
- Partial information easy to handle
- Rigorous notion of soundness

EPMO Weakened



Lowe-style attack



Soundness

Let Π be an annotated protocol, i.e.

- A set of roles (parametrized behaviors)
  - A role is a sequence of transmissions/receptions (nodes)
- For each transmission node n, a guarantee γ_n
- For each reception n, a rely formula ρ_n
- The principal active on node n is prin(n)

γ_n, ρ_n may refer to message ingredients

Π is sound if, for all executions $\mathcal{B}$, and message receptions $n \in \mathcal{B}$

$$\{\text{prin}(m) \text{ says } \gamma_m \colon m \prec_{\mathcal{B}} n\} \quad \longrightarrow_{\mathcal{L}} \quad \rho_n$$

where $\longrightarrow_{\mathcal{L}}$ is the consequence relation of the underlying logic

Soundness follows from authentication properties

- Authentication tests a good tool
- Recency easy to incorporate

One case of soundness

$$\rho_{m,3} = \begin{array}{l} B \text{ says } \gamma_{b,2} \\ \text{and } C \text{ says } \gamma_{c,5} \end{array}$$

Suppose $n_{m,3} \in \mathcal{B}$
where $m \in \mathsf{Merchant}[B, C, M, p, g, N_c, N_m, N_b]$
necessary keys uncompromised, nonces u.o.

Then $n_{b,2}, n_{c,5} \in \mathcal{B}$ for some
$b \in \mathsf{Bank}[B, C, *, p, N_c, N_m, N_b]$ and
$c \in \mathsf{Customer}[B, C, M, p, g, N_c, N_m, N_b]$

Moreover, $n_{m,1} \preceq_{\mathcal{B}} n_{b,2}$ and $n_{m,1} \preceq_{\mathcal{B}} n_{c,5}$

Same form as an authentication result with recency
In weakened EPMO, only know

$$c \in \mathsf{Customer}[B, C, X, p, g, N_c, N_m, N_b]$$

Four Tenets of Logical Trust Management

1. Principal theories: Each principal P holds a theory Th_P;
   P derives conclusions using Th_P
   - May rely on formulas P' says ψ as additional premises
   - P says ϕ only when P derives ϕ
2. Trust in others: "P trusts P' for a subject ψ" means
   - P says $((P'$ says $\psi) \supset \psi)$
3. Syntactic authority: Certain formulas, e.g.
   - P says ϕ
   - P authorizes ϕ

   are true whenever P utters them
4. Access control via deduction: P may control resource r;
   P takes action $\phi(r, P')$ on behalf of P' when P derives
   - P' requests $\phi(r, P')$
   - P' deserves $\phi(r, P')$

Trust Management in Strand Spaces

Combining trust management with nonce-based protocols

- Trust and commitment in e-commerce

Key idea: Annotate positive nodes with guarantees, negative nodes with rely formulas

- This localizes trust management reasoning
- Each principal reasons in local theory
- Soundness ensures every rely was guaranteed

Strand spaces and authentication tests: Strong method for

- Discovering protocol flaws
- Proving protocols correct
- Shaping protocol design

Trust engineering via cryptographic protocols

Permissible Bundles

Let $\mathcal{B}$ a bundle; let each P hold theory Th_P

$\mathcal{B}$ is permissible if

$$\{\rho_m : m \Rightarrow^+ n\} \longrightarrow_{\mathsf{Th}_P} \gamma_n$$

for each positive,
regular $n \in \mathcal{B}$

Means, every principal derives guarantee before sending each message

- permissible is vertical (strand-by-strand)
- sound is horizontal (cross-strand)

What trust is needed in permissible bundles of a sound protocol?

For which P' and ψ must P accept

$$P \text{ says } ((P' \text{ says } \psi) \supset \psi)$$

Trust Mgt Reasoning for EPMO, 1: Bank

$$\begin{array}{llll}
\gamma_{b,2} & \forall P_M & \textbf{if} & C \text{ authorizes transfer}(B, \text{price}, P_M, N_m), \\
 & & \textbf{and} & P_M \text{ requests transfer}(B, \text{price}, P_M, N_m), \\
 & & \textbf{then} & \text{transfer}(B, \text{price}, P_M, N_m).
\end{array}$$

$$\begin{array}{lll}
\rho_{b,3} & & C \text{ says } C \text{ authorizes transfer}(B, \text{price}, M, N_m), \\
 & \textbf{and} & M \text{ says } M \text{ requests transfer}(B, \text{price}, M, N_m).
\end{array}$$

Universal quantifier $\forall P_M$ expresses "payable to bearer"

After node $n_{b,3}$, B can deduce

$$\text{transfer}(B, \text{price}, P_M, N_m)$$

Uses syntactic authority (authorizes, requests) but not trust

Trust Mgt Reasoning for EPMO, 2: Merchant

$\gamma_{m,2}$	$\forall P_B$	**if**	$\mathsf{transfer}(P_B, \mathsf{price}, M, N_m)$,
		then	$\mathsf{ship}(M, \mathsf{goods}, C)$.
$\rho_{m,3}$			B says $\gamma_{b,2}$,
		and	C says $\gamma_{c,5}$.
$\gamma_{m,4}$			M requests $\mathsf{transfer}(B, \mathsf{price}, M, N_m)$,
		and	$\mathsf{ship}(M, \mathsf{goods}, C)$.

After node $n_{m,3}$, can M can deduce $\mathsf{ship}(M, \mathsf{goods}, C)$?
Yes, if M requests transfer and accepts

$$B \text{ says } \gamma_{b,2} \quad \text{implies} \quad \gamma_{b,2}$$

i.e. M trusts B to transfer the funds as promised

$\gamma_{b,2}$	$\forall P_M$	**if**	C authorizes $\mathsf{transfer}(B, \mathsf{price}, P_M, N_m)$,
		and	P_M requests $\mathsf{transfer}(B, \mathsf{price}, P_M, N_m)$,
		then	$\mathsf{transfer}(B, \mathsf{price}, P_M, N_m)$.

Trust Mgt Formulas for EPMO, 3: Customer

Customer:

$\rho_{c,2}$	M says $\gamma_{m,2}$.
$\rho_{c,4}$	B says $\gamma_{b,2}$.
$\gamma_{c,5}$	C authorizes transfer$(B, \text{price}, M, N_m)$.

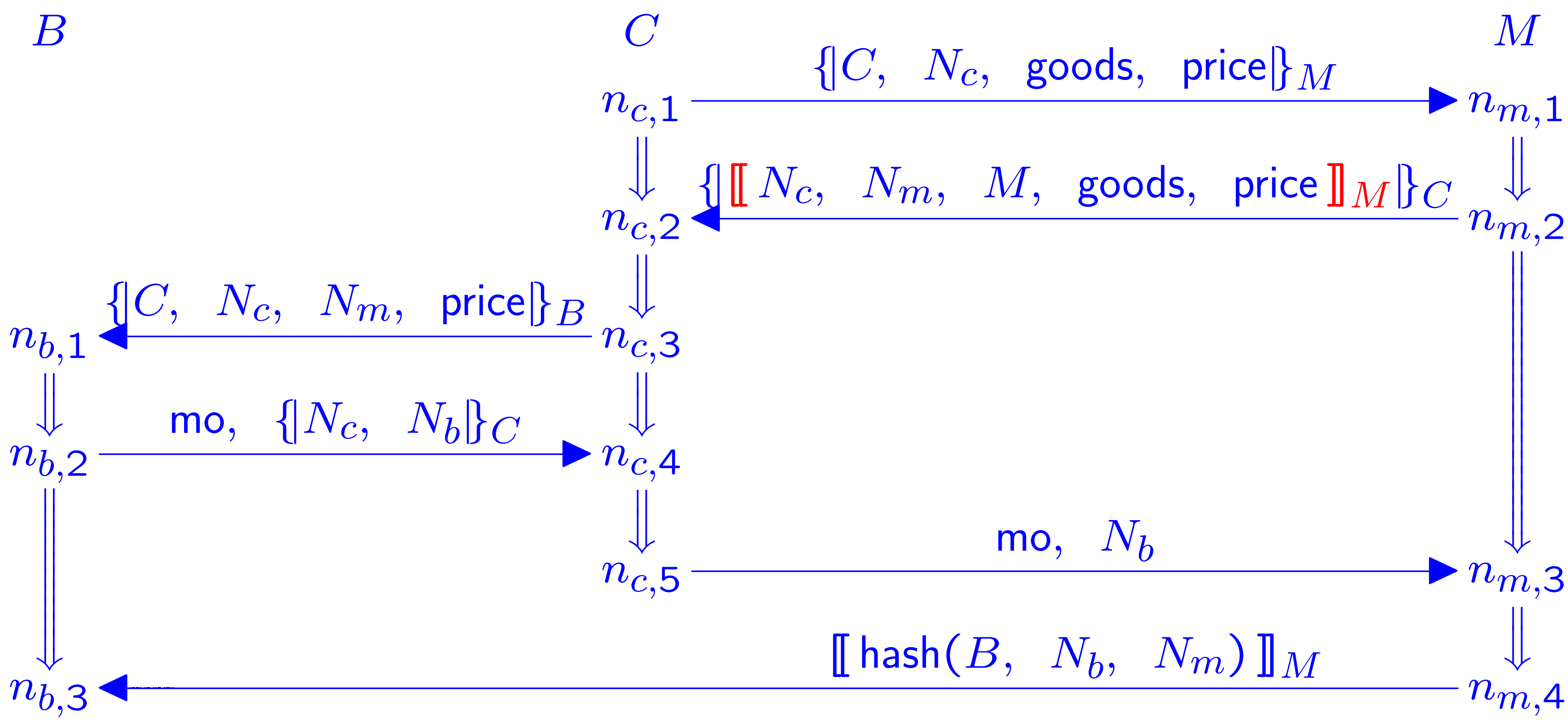
Decision to assert $\gamma_{c,5}$ depends on C's trust in M:

$$M \text{ says } \gamma_{m,2} \quad \text{implies} \quad \gamma_{m,2}$$

and C's trust in B:

$$B \text{ says } \gamma_{b,2} \quad \text{implies} \quad \gamma_{b,2}$$

A Signed Alternate: SEPMO



Signed Electronic Purchase using Money Order

$\text{mo} = [\![\text{hash}(C, N_c, N_b, N_m, \text{price})]\!]_B$