Trust Engineering with Cryptographic Protocols

Joshua Guttman
Javier Thayer

10 February 2004

Supported by the National Security Agency
and the MITRE-Sponsored Research program.

Goal of this Line of Work

Develop methods for reasoning about cryptographic protocols as used with real world consequences

Examples:

- Electronic retail commerce
  - When is customer committed to paying?
  - When is merchant committed to shipping?
  - Whose word did you depend on when deciding?
- Distributed access control
  - As formulated via trust management
- Electronic finance, etc.

Enrich strand space framework with

- Guaranteed formulas on message transmission nodes
- Rely formulas (assumptions) on reception nodes

where the formulas belong to some trust management logic

Goals of Today's Talk

Explain underlying ideas by example

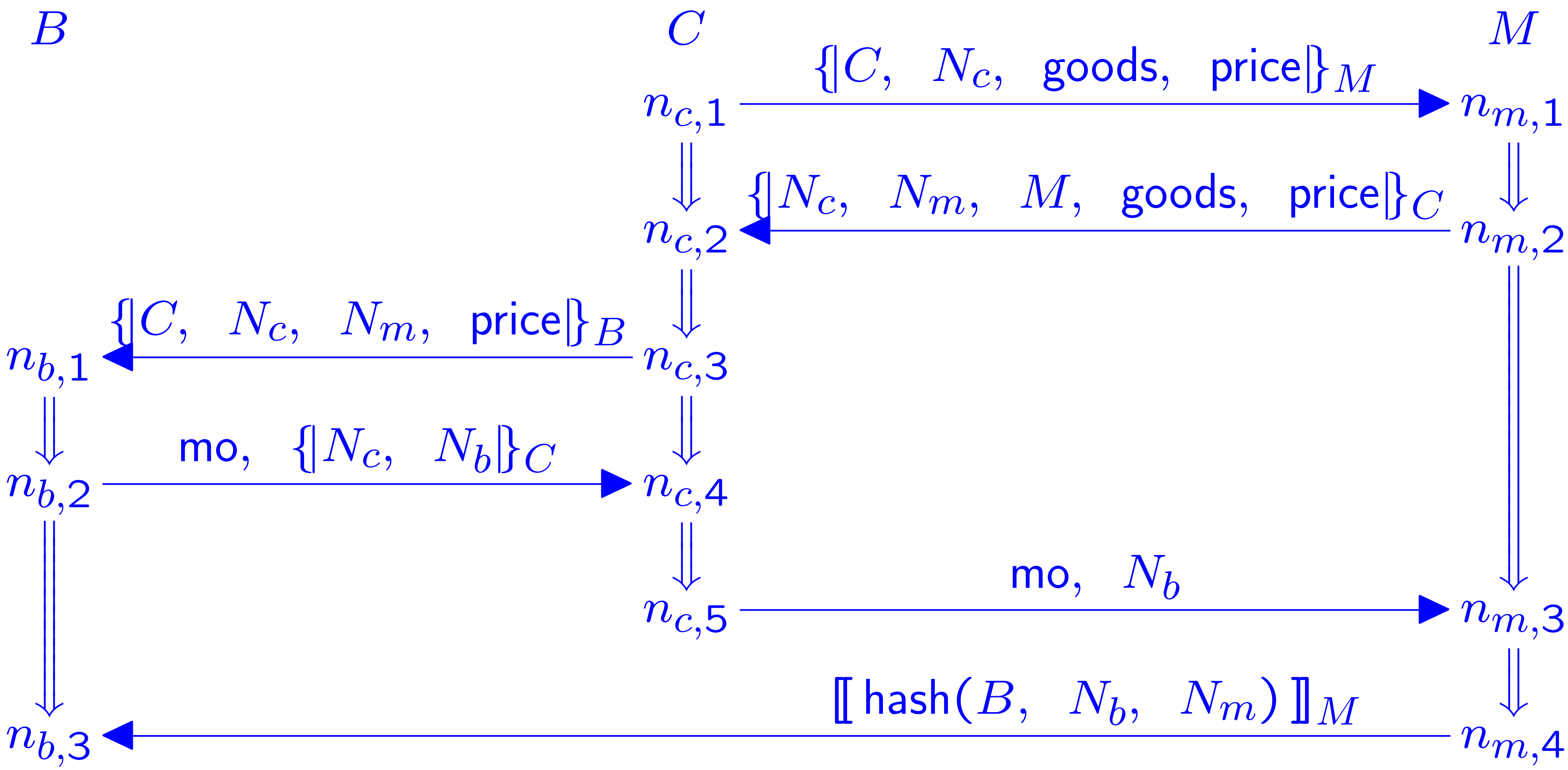
Explore the "trust support" of each role R of a protocol

- Describes degree of trust R may require,
  trusting others to be right
- Depends on shape of this execution
- If only finitely many shapes possible,
  trust support for role R is a single formula

Indicate how to find the shapes of a protocol

- Generate sets of regular strands $\mathbb{A}$
- No other regular strands needed
  - If these regular strands $\mathbb{A}$ belong to any bundle,
    they belong to some bundle with
    no regular strands other than $\mathbb{A}$

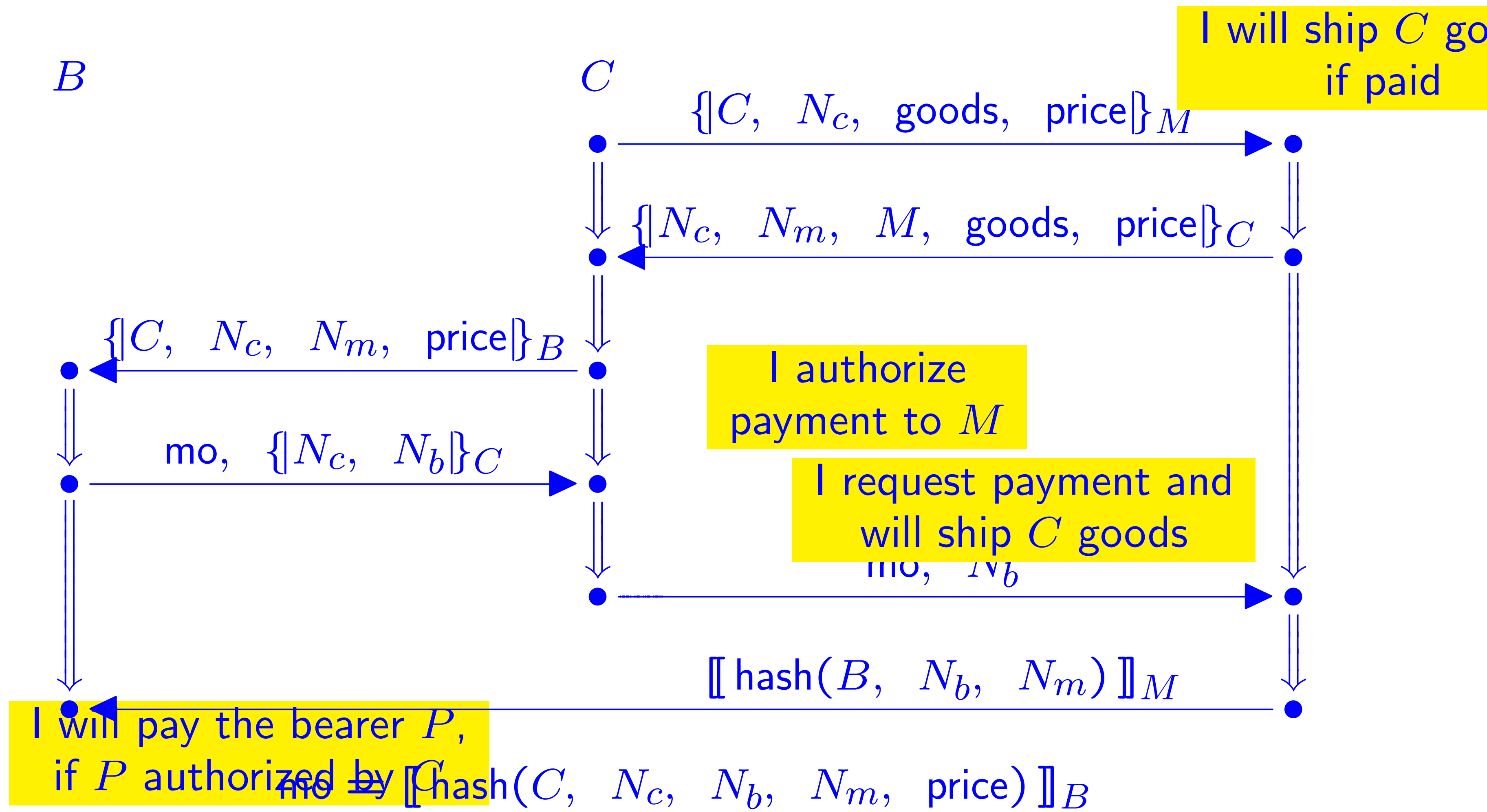
An Example: EPMO



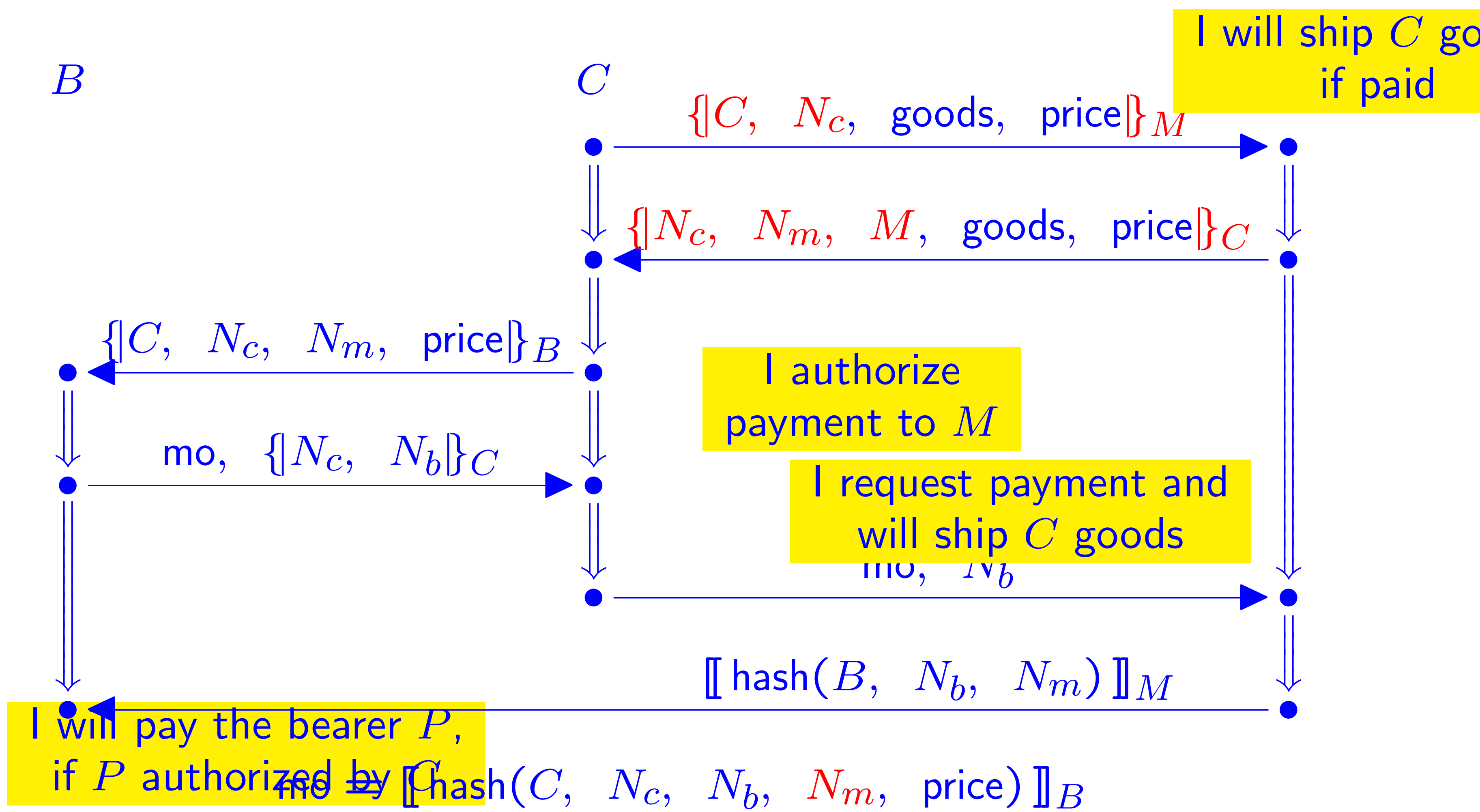
Electronic Purchase using Money Order

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

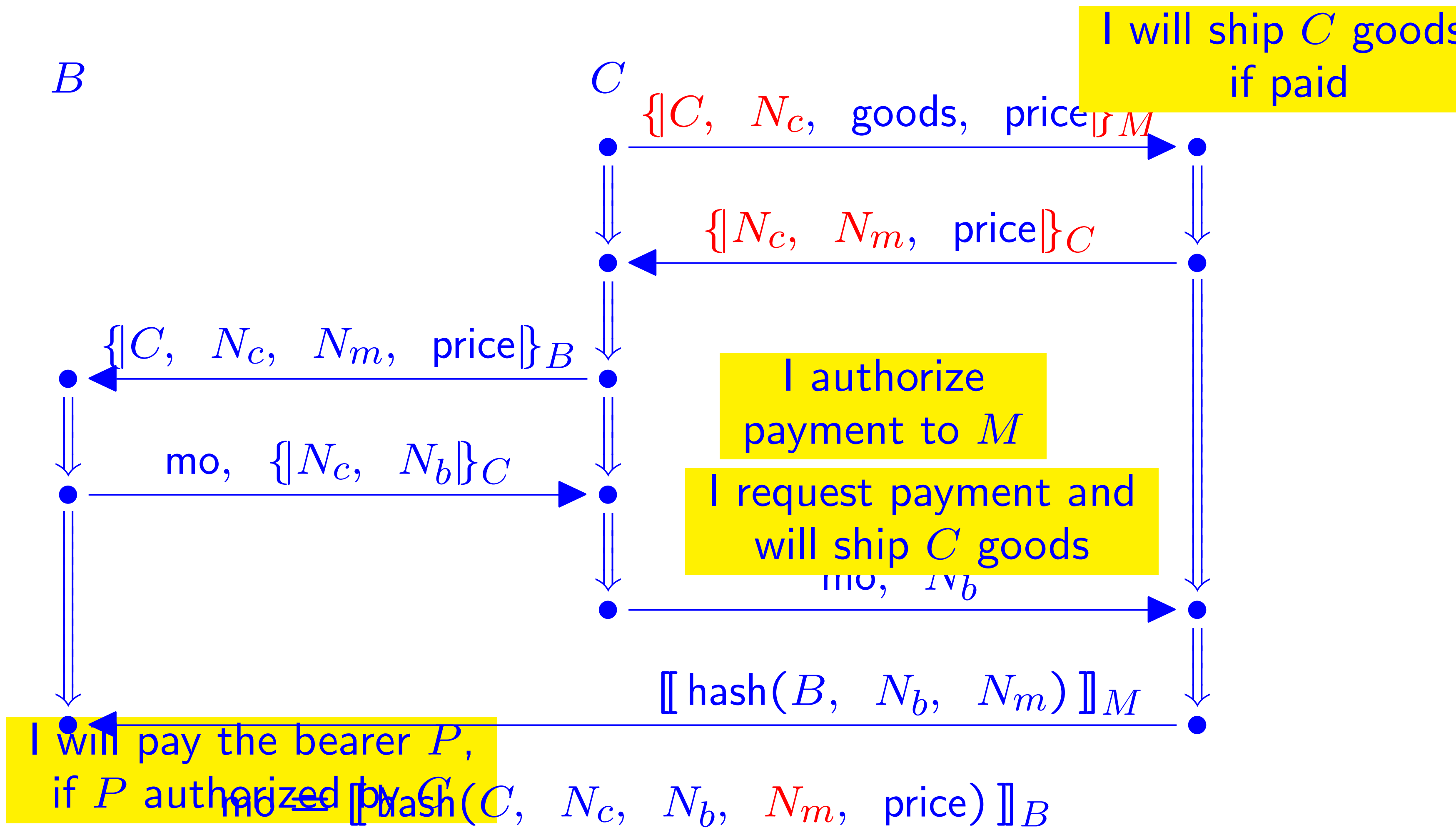
EPMO: Commitments on sends



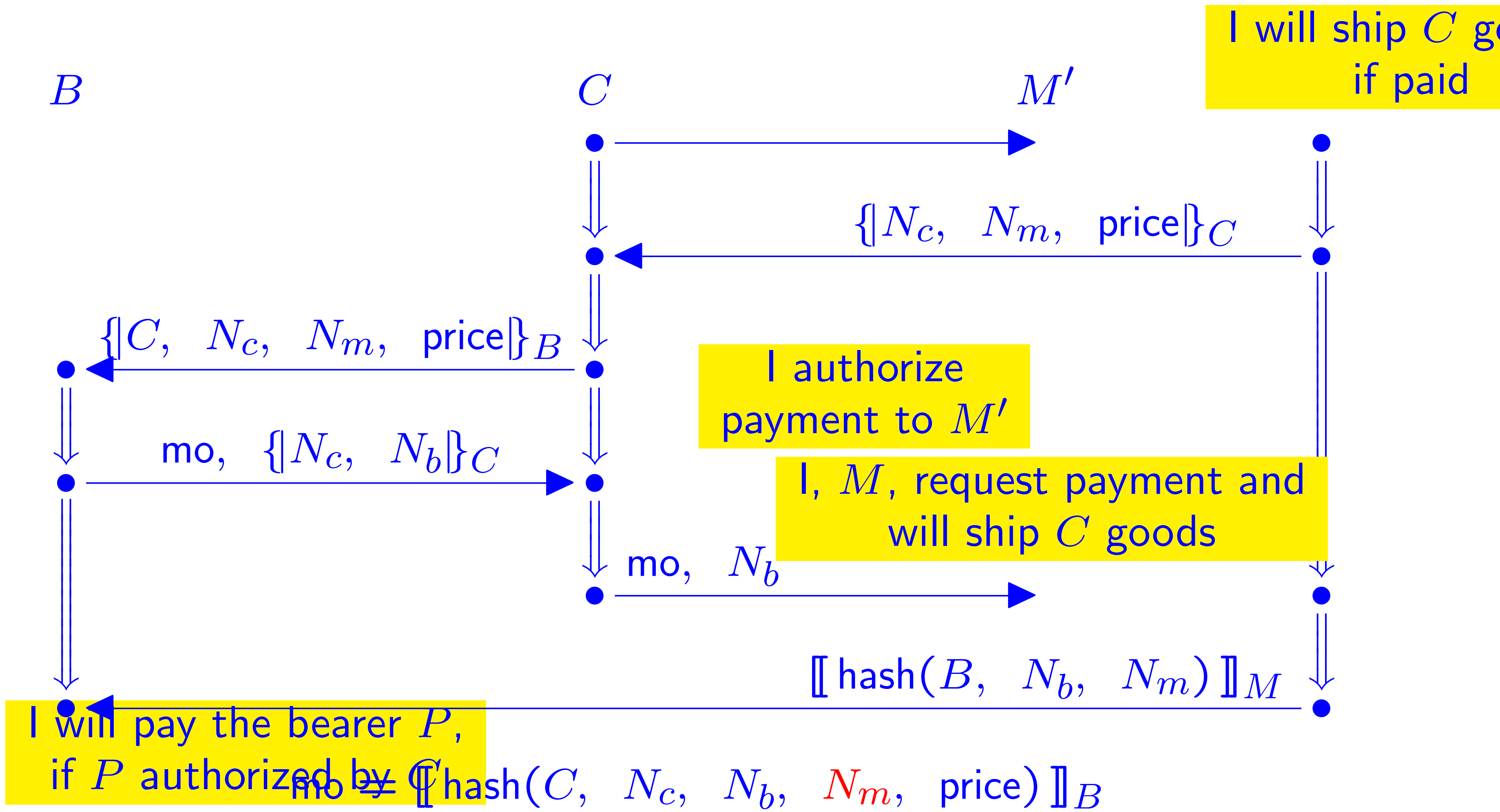
EPMO and Needham-Schroeder-Lowe



Weakened EPMO



Lowe-style attack



Trust management and protocols

Strategy: Each principal P

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

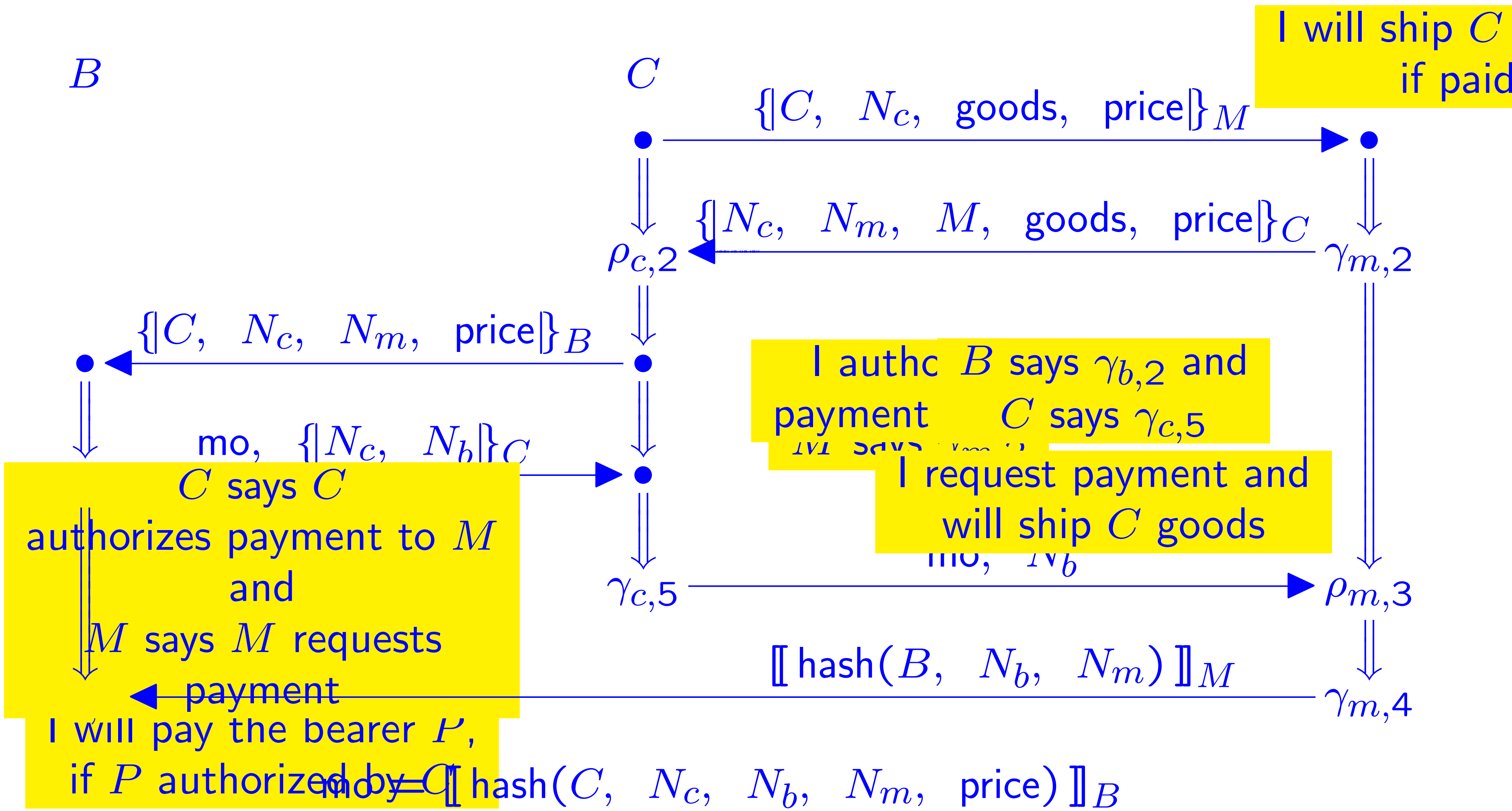
Also need formulas on negative nodes

- Specifies what recipient may rely on
- Provides local representation of remote guarantee

Role of protocol

- When I rely on you having made a guarantee,
  then you did make that guarantee
- Coordination mechanism for rely/guarantees
- Sound protocol: one where
  "relies" always backed by "guarantees"

EPMO: Rely/Guarantee Formulas



Soundness

Let Π be an annotated protocol, i.e.

- A set of parametric strands, called the roles
  - prin(n) the principal active on node n
- For each positive node n, a guarantee γ_n
- For each negative node n, a rely formula ρ_n

γ_n, ρ_n may share parameters with strand

Π sound for bundles $\mathcal{B} \in \mathbb{B}$ if for all negative $n \in \mathcal{B}$,

$$\Gamma \longrightarrow_{\mathcal{L}} \rho_n$$

where

$$\Gamma = \{\text{prin}(m) \text{ says } \gamma_m \colon m \prec_{\mathcal{B}} n\}$$

and $\longrightarrow_{\mathcal{L}}$ 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 $m \in \text{Merchant}[B, C, M, p, g, N_c, N_m, N_b]$
necessary keys uncompromised, nonces u.o.

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

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

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

Four Tenets of Logical Trust Management

1. Syntactic authority: Certain formulas, e.g.
   - P says ϕ
   - P authorizes ϕ

   are true whenever P utters them
2. 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
3. Trust in others: "P trusts P' for a subject ψ" means
   - P says $((P'$ says $\psi) \supset \psi)$
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 $\phi(r, P')$

Permissible Bundles

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

$\mathcal{B}$ is permissible if

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

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

$\gamma_{b,2}$	$\forall P_M$	**if**	C authorizes transfer$(B, \text{price}, P_M, N_m)$,
		and	P_M requests transfer$(B, \text{price}, P_M, N_m)$,
		then	transfer$(B, \text{price}, P_M, N_m)$.
$\rho_{b,3}$			C says C authorizes transfer$(B, \text{price}, M, N_m)$,
		and	M says M requests transfer$(B, \text{price}, M, N_m)$.

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

After node $n_{m,3}$, can M can deduce ship(M, 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 transfer$(B, \text{price}, P_M, N_m)$,
		and	P_M requests transfer$(B, \text{price}, P_M, N_m)$,
		then	transfer$(B, \text{price}, P_M, N_m)$.

Pattern of Reasoning We Used

Suppose $m \Rightarrow^{+} m'$ with m negative and m' positive

Premise ρ_m of the form: prin(n) says γ_n
P uses Th$_P$ to decide whether to trust prin(n) for γ_n

prin(n) says γ_n implies γ_n

Really, constraint on Th$_P$

Where this succeeds, reason from Th$_P$ plus formulas γ_n

- Try to infer $\gamma_{m'}$
- If this succeeds, send message on m'

Non-Machiavellian reasoning:

- prin(n) says γ_n yields γ_n
  or nothing

prin(m') trusts prin(n) for γ_n
but maybe prin(n) relied on someone else?

- prin(n) responsible for deriving γ_n

Trusting Peers

Non-Machiavellian

Let $\mathcal{B}$ be permissible for a sound protocol with

$n \in \mathcal{B}$ positive, regular
$P = \text{prin}(n)$
$S = \text{rely}_n \subset \{m \colon m \prec_{\mathcal{B}} n \text{ and } m \text{ positive, regular}\}$

Th$_P$ establishes

(check claims) $\quad \bigwedge_{m \in S}(\text{prin}(m) \text{ says } \gamma_m)$ implies $\bigwedge_{m \in S} \gamma_m$

(make progress) $\quad \bigwedge_{m \in S} \gamma_m$ implies γ_n

Trust reasoning

- Trust evaluation
- Trust extension: Define cf$(n) =$
  - prin(n) says $\bigwedge_{m \in S} \gamma_m$ implies γ_n

Trust extension: for all $n \in \mathcal{B}$, γ_n is true, just in case
for all $m \in \mathcal{B}$, cf(m) is true

Trust Engineering

Protocol designer gives principal P two degrees of freedom

(1) When prin(m) says γ_m, does Th$_P$ derive γ_m?

(2) When does Th$_P$ derive cf(n)?

In (1), decision is a function of

- prin(m)
- protocol parameters occurring in γ_m

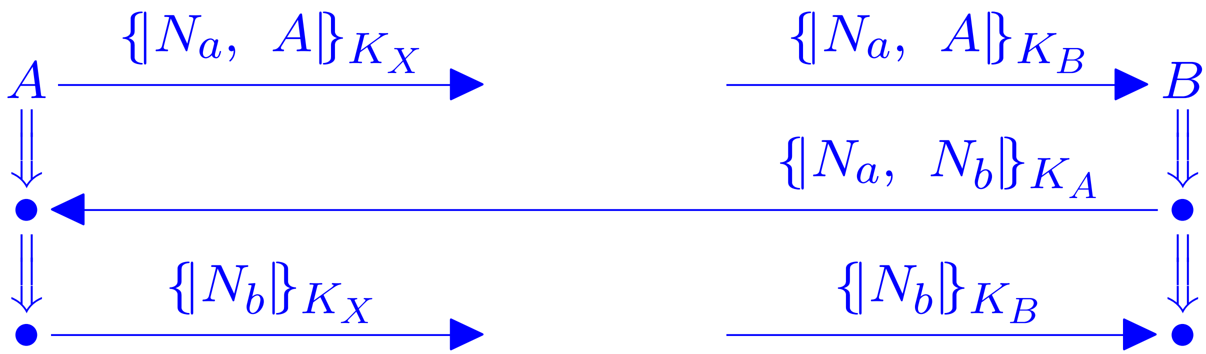
In (2), decision is a function of

- parameters in cf(n)

But this assumes a known set of regular nodes

- What if protocol has several shapes of bundle?

Some Protocols Have A Single Shape



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

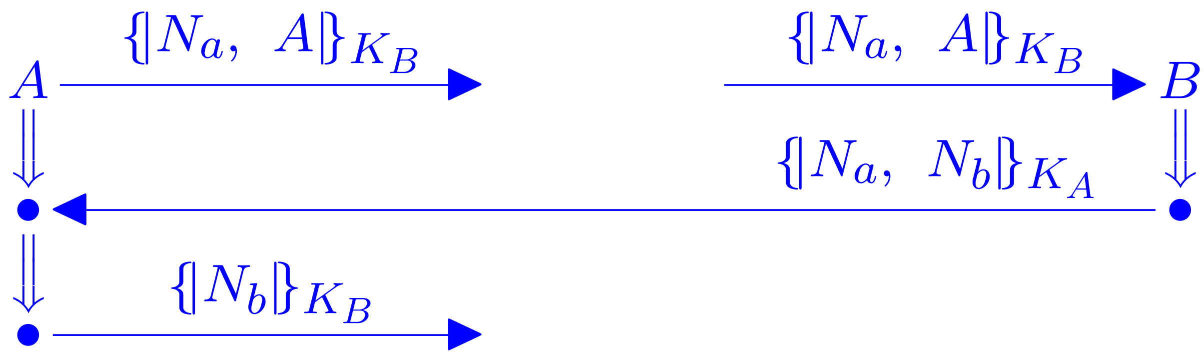
NSResp$[A, B, N_a, N_b]$

for every $\mathbb{A}$ containing lower right node

assuming K_A, K_B non-originating,

N_a, N_b uniquely originating

More or Less



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

for every $\mathbb{A}$ dominated by lower left node

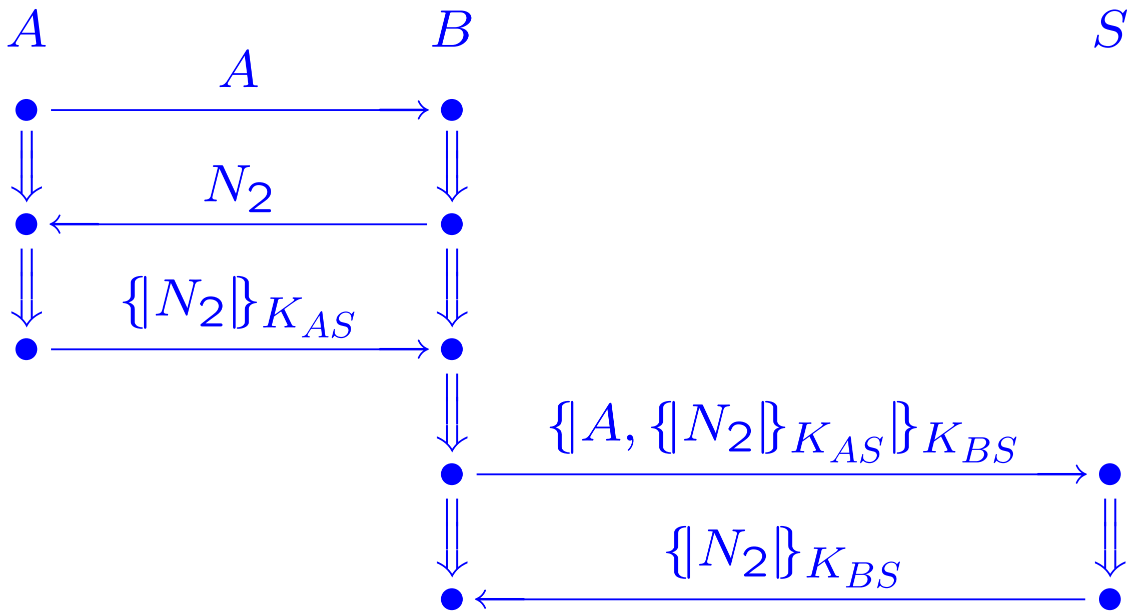
assuming K_A, K_B non-originating,

N_a, N_b uniquely originating

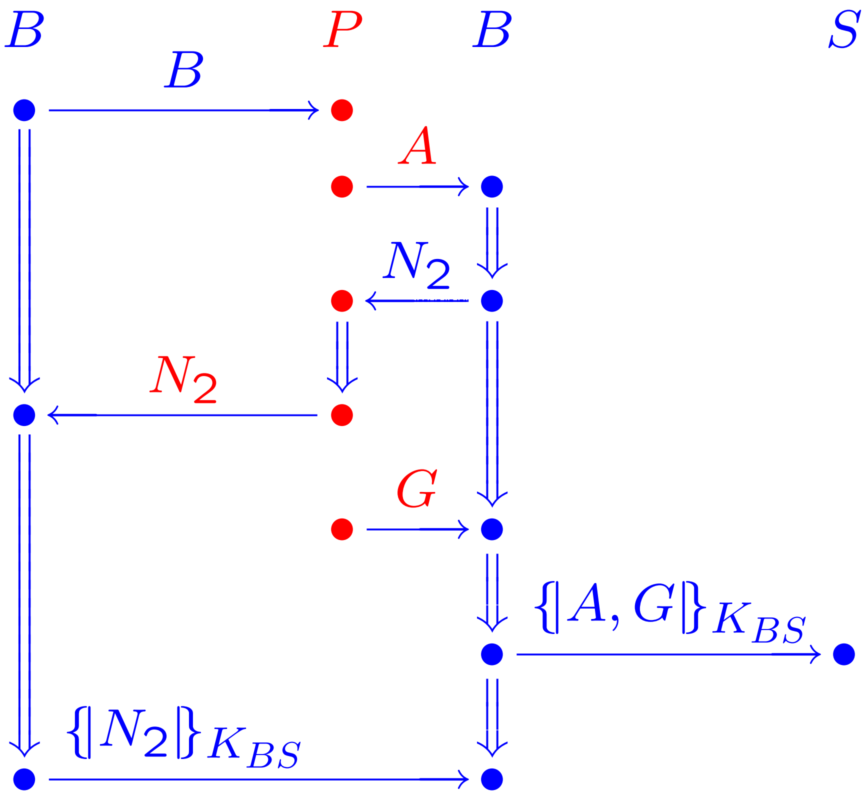
Other Protocols Have Multiple Shapes

Otway-Rees if $A = B$ possible

Woo-Lam



Woo-Lam Infiltrated



The Shapes of a Protocol

Definition: A shape for Π, R is a

- A skeleton $\mathbb{A}$ i.e. set of regular strands with $\preceq$
  such that there's $\mathcal{B}$ for Π with just those strands
  and last node of R-strand is maximal in $\mathbb{A}$

A shape catalog for Π, R is

- A set $\mathcal{S}$ of shapes such that
  Every bundle is equivalent to an instance of just one $\mathbb{A} \in \mathcal{S}$

Shape catalog for NS is singleton:

$$\begin{tikzcd}
A \arrow[r, "\{\!|N_a,\ A|\!\}_{K_X}"] \arrow[d, Rightarrow] & {} & \preceq & {} \arrow[r, "\{\!|N_a,\ A|\!\}_{K_B}"] & B \arrow[d, Rightarrow] \\
\bullet \arrow[d, Rightarrow] & {} \arrow[l, "\{\!|N_a,\ N_b|\!\}_{K_A}"'] & \succeq & {} & \bullet \arrow[l, "\{\!|N_a,\ N_b|\!\}_{K_A}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "\{\!|N_b|\!\}_{K_X}"] & {} & \preceq & {} \arrow[r, "\{\!|N_b|\!\}_{K_B}"] & \bullet
\end{tikzcd}$$

Outgoing Authentication Test

$$\begin{tikzcd} m_0 \arrow[rr, dashed, "K^{-1} \in \mathsf{S} \qquad a \sqsubset \{\!|h|\!\}_K"] \arrow[d, Rightarrow] & & n_0 \arrow[d, Rightarrow] \\ m_1 & & n_1 \arrow[ll, dashed, "a \sqsubset \mathsf{term}(n_1) \qquad a \sqsubset t'"'] \end{tikzcd}$$

"Regular" means uncompromised,
i.e. not the penetrator

Assume	$\{\!\|h\|\!\}_K \not\sqsubset \mathsf{term}(m_1)$
	a created freshly at m_0,
	a contained only in $\{\!\|h\|\!\}_K$
Conclude	nodes n_0, n_1 exist in $\mathcal{B}$ and are regular
	$\{\!\|h\|\!\}_K \not\sqsubset t'$
	$m_0 \prec n_0 \prec n_1 \prec m_1$

NSL: Responder's Outgoing Test

$$
\begin{tikzcd}
{} \arrow[r, "\{\!|N_a,\ A|\!\}_{K_B}"] & B \arrow[d, Rightarrow] \\
{} & m_0 \arrow[l, "\{\!|N_a,\ N_b,\ B|\!\}_{K_A}"'] \arrow[d, Rightarrow] \\
{} \arrow[r, "\{\!|N_b|\!\}_{K_B}"] & m_1
\end{tikzcd}
$$

This is an outgoing test

"Test edge" is $\{\!|N_a,\ N_b,\ B|\!\}_{K_A} \Longrightarrow \{\!|N_b|\!\}_{K_B}$

What regular strand can transform $\{\!|N_a,\ N_b,\ B|\!\}_{K_A}$?

Matching Transforming Edges

What edges can transform $\{\!|N'_a,\ {\color{green}N'_b},\ B'|\!\}_{K'_A}$?

$$
\begin{tikzcd}
A \arrow[r, "\{\!|N_a,\ A|\!\}_{K_B}"] \arrow[d, Rightarrow] & {} \\
\bullet \arrow[d, Rightarrow] & {} \arrow[l, "\{\!|N_a,\ N_b,\ B|\!\}_{K_A}"'] \\
\bullet \arrow[r, "\{\!|N_b|\!\}_{K_B}"] & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
{} \arrow[r, "\{\!|N_a,\ A|\!\}_{K_B}"] & B \arrow[d, Rightarrow] \\
{} & \bullet \arrow[l, "\{\!|N_a,\ N_b,\ B|\!\}_{K_A}"'] \arrow[d, Rightarrow] \\
{} \arrow[r, "\{\!|N_b|\!\}_{K_B}"] & \bullet
\end{tikzcd}
$$

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

A Few Refinements

Test nodes need not be on same edge

$+\{\!|N_a,\ N_b,\ B|\!\}_{K_A} \Longrightarrow -\{\!|N_b|\!\}_{K_B}$
could be
$+\{\!|N_a,\ N_b,\ B|\!\}_{K_A} \preceq -\{\!|N_b|\!\}_{K_B}$

Test value N_b need not originate on m_0

- m_0 must precede all red forms of N_b

Transforming edge must precede some regular node containing N_b

Hence, outgoing test may be used repeatedly

Incoming Test

Symmetrically,

$$
\begin{tikzcd}
n_0 \arrow[r, dashed, "\star a \sqsubset \mathsf{term}(n_0)"] \arrow[d, Rightarrow] & {} \arrow[r, dashed] & \bullet \arrow[d, Rightarrow] \\
n_1 & {} \arrow[l, dashed, "a \sqsubset \{\!|h|\!\}_K \sqsubset \mathsf{term}(n_1) \quad K \in \mathsf{S}"'] & \bullet \arrow[l, dashed, "\{\!|h|\!\}_K"']
\end{tikzcd}
$$

"Unsolicited" tests

Regular

$$
\begin{tikzcd}
\circ & \bullet \arrow[l, dashed, "\{\!|h|\!\}_K \quad K \in \mathsf{S}"']
\end{tikzcd}
$$

Key Safety

We assume K_A initially uncompromised

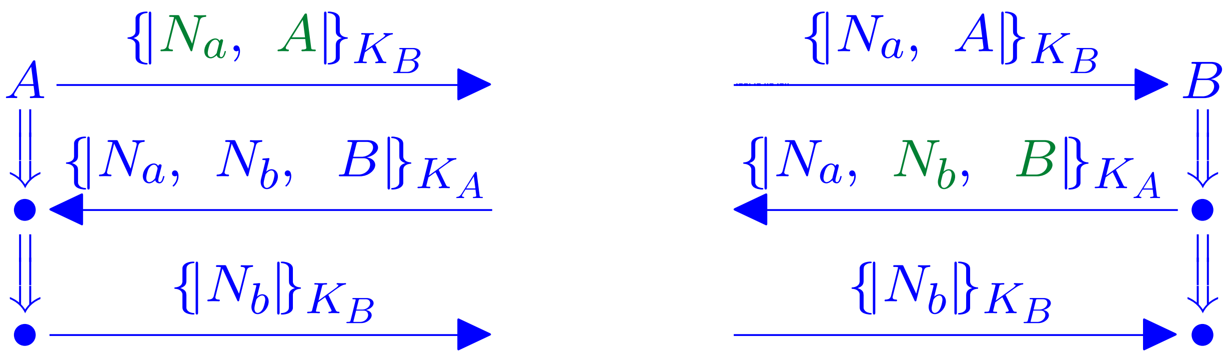
K_A never can be compromised via protocol
since it's never transmitted, only used

A key K with this property is *safe* (written $K \in \mathsf{S}$)

- Recursively, also safe if transmitted
  only when protected by encryption with safe keys

Theorem: $K \in \mathsf{S}$ implies K never disclosed to penetrator;
never available for penetrator encrypt or decrypt

Automation: Primary occurrences



Test edge: Primary occurrence of nonce or key,
followed by secondary occurrence in new form

Algorithm

Enumerate safe values
starting with keys assumed initially uncompromised

For each test edge, repeatedly search for transforming edges

- Take cases when multiple candidates

For new values such as session keys
check safety

- Assumption: servers generate uniquely originating keys
  distinct from long term keys

New values may lead to new tests

Some questions

(Soundness) Is every result of this algorithm a shape?

(Completeness) Is every shape eventually generated?

(Termination) Is there a reasonable class of protocols for which this algorithm terminates?

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:

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