Combining (and Refining) Protocols using the Authentication Tests

Joshua D. Guttman

The MITRE Corporation

Protocol Exchange

Supported by the MITRE-Sponsored Research Program

Combining protocols: The problem

- A protocol Π_1 may
  - achieve a goal if used in isolation
  - fail the goal if used in combination with protocol Π_2

Combining protocols: The problem

- A protocol Π_1 may
  - achieve a goal if used in isolation
  - fail the goal if used in combination with protocol Π_2
- How to analyze Π_1 once and infer set of safe combiners Π_2

  Safety: All goals achieved alone preserved in combination

Combining protocols: The problem

- A protocol Π_1 may
  - achieve a goal if used in isolation
  - fail the goal if used in combination with protocol Π_2
- How to analyze Π_1 once and infer set of safe combiners Π_2

  Safety: All goals achieved alone preserved in combination
- Advantages:

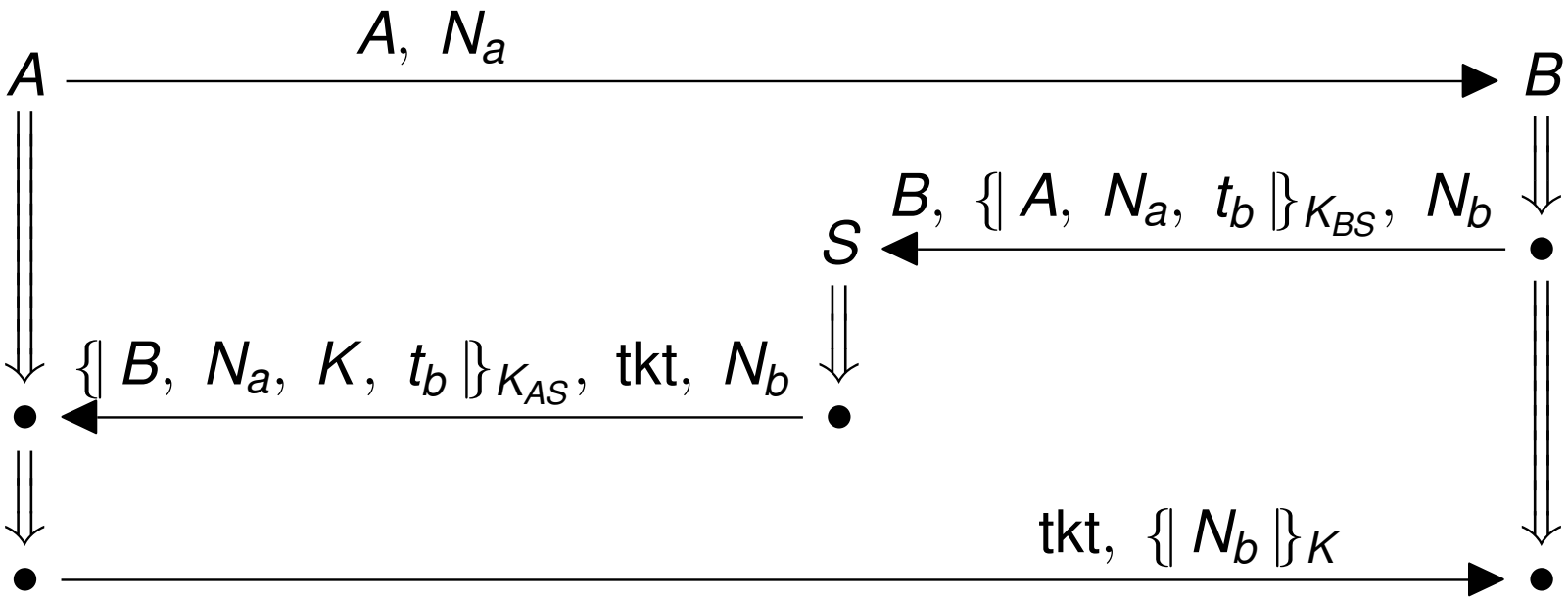
  Efficiency: Analyze small protocols, not big protocols

  Locality: Separate local correctness from preservation

  Flexibility: To add goals, just prove 'em locally

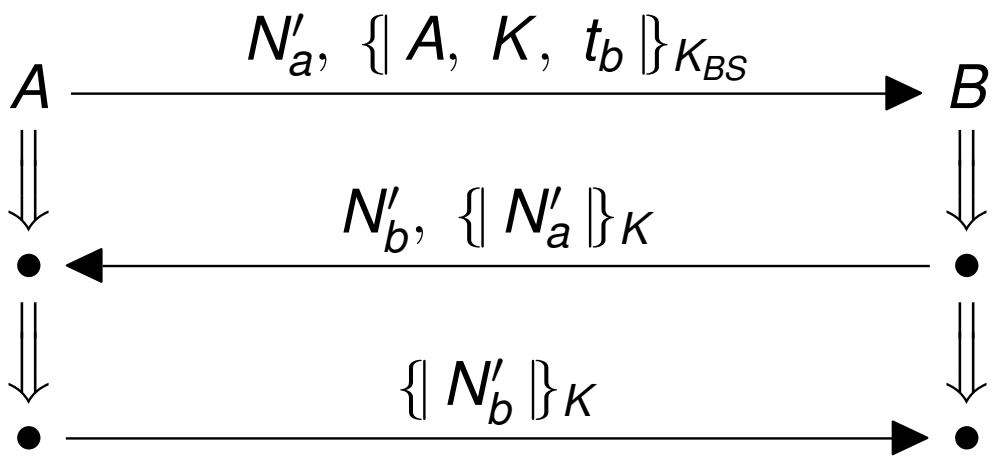
  Refinement: Method applicable to stepwise protocol design

Neuman-Stubblebine Primary: Π_1

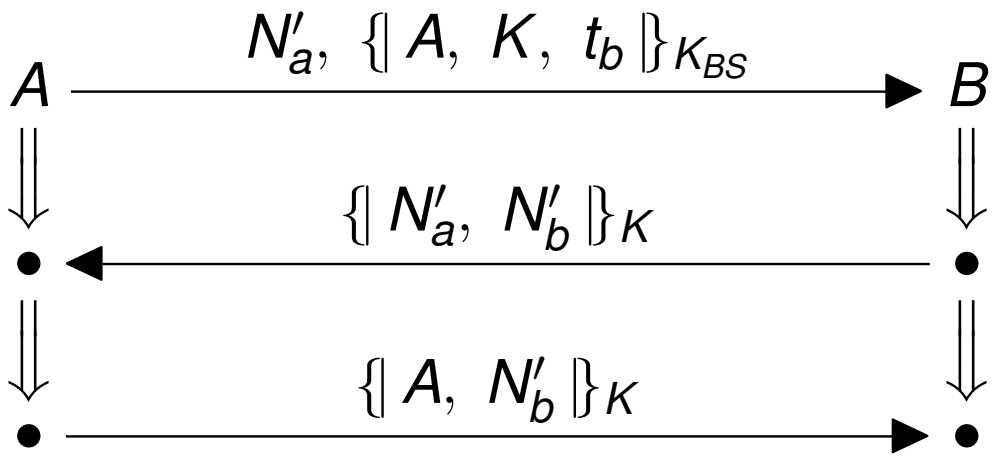


$$\text{tkt} = \{\!|\, A,\ K,\ t_b \,|\!\}_{K_{BS}}$$

Neuman-Stubblebine Secondary: Π_2



Neuman-Stubblebine Secondary Corrected: Π'_2



Syntactic criteria for independence

$\Pi_1 \longrightarrow \Pi_2$

When	some $s_1 \in \Pi_1$ transmits $\{\!\|\ \ldots a \ldots\ \|\!\}_K$
and	some $s_2 \in \Pi_2$ receives it,
then	s_2 never re-transmits a in any new form

$\Pi_2 \longrightarrow \Pi_1$

When	some $s_2 \in \Pi_2$ transmits $\{\!\|\ t\ \|\!\}_K$
and	some $s_1 \in \Pi_1$ receives it,
then	s_2 previously received $\{\!\|\ t\ \|\!\}_K$ in the same form

Protocol independence

- Does Π_2 create new ways to solve challenges Π_1 uses?
  - If not, how can it undermine Π_1's goals?
- Challenge/solution terminology from authentication test idea

Outgoing authentication test

Suppose nonce or session key a originates freshly at event n_0, and

> *a occurs at n_0 only within*
> *a set S of encryptions $\{\!|\ \cdots a \cdots\ |\!\}_K$*

but later at event n_1, a occurs outside those encryptions.

Then either:

> *a regular (honest) participant transforms a*

or else

> *K^{-1}, the decryption key for one of the Ks,*
> *is compromised.*

Outgoing authentication test

Suppose nonce or session key a originates freshly at event n_0, and

> *a occurs at n_0 only within*
> *a set S of encryptions $\{\!|\ \cdots a \cdots\ |\!\}_K$*

but later at event n_1, a occurs outside those encryptions.

A test transformation from n_0 to n_1

Then either:

> *a regular (honest) participant transforms a*

or else

> *K^{-1}, the decryption key for one of the Ks,*
> *is compromised.*

Solutions: regular transformation or compromise

Incoming authentication test

Suppose an encrypted message $m = \{\!|\, t_0 \,|\!\}_{K_0}$

> *m occurs outside a set S of encryptions* $\{\!|\, \cdots m \cdots \,|\!\}_K$ *at event* n_1.

Then either:

> *a regular participant transforms m*

or else

> K^{-1}, *the decryption key for one of the Ks, is compromised*

or else

> *the encryption key* K_0 *is compromised.*

Incoming authentication test

Suppose an encrypted message $m = \{\!|\, t_0 \,|\!\}_{K_0}$

> *m occurs outside a set S of encryptions $\{\!|\, \cdots m \cdots \,|\!\}_K$ at event n_1.*

test transformation

Then either:

> *a regular participant transforms m*

or else

> *K^{-1}, the decryption key for one of the Ks, is compromised*

or else

> *the encryption key K_0 is compromised.*

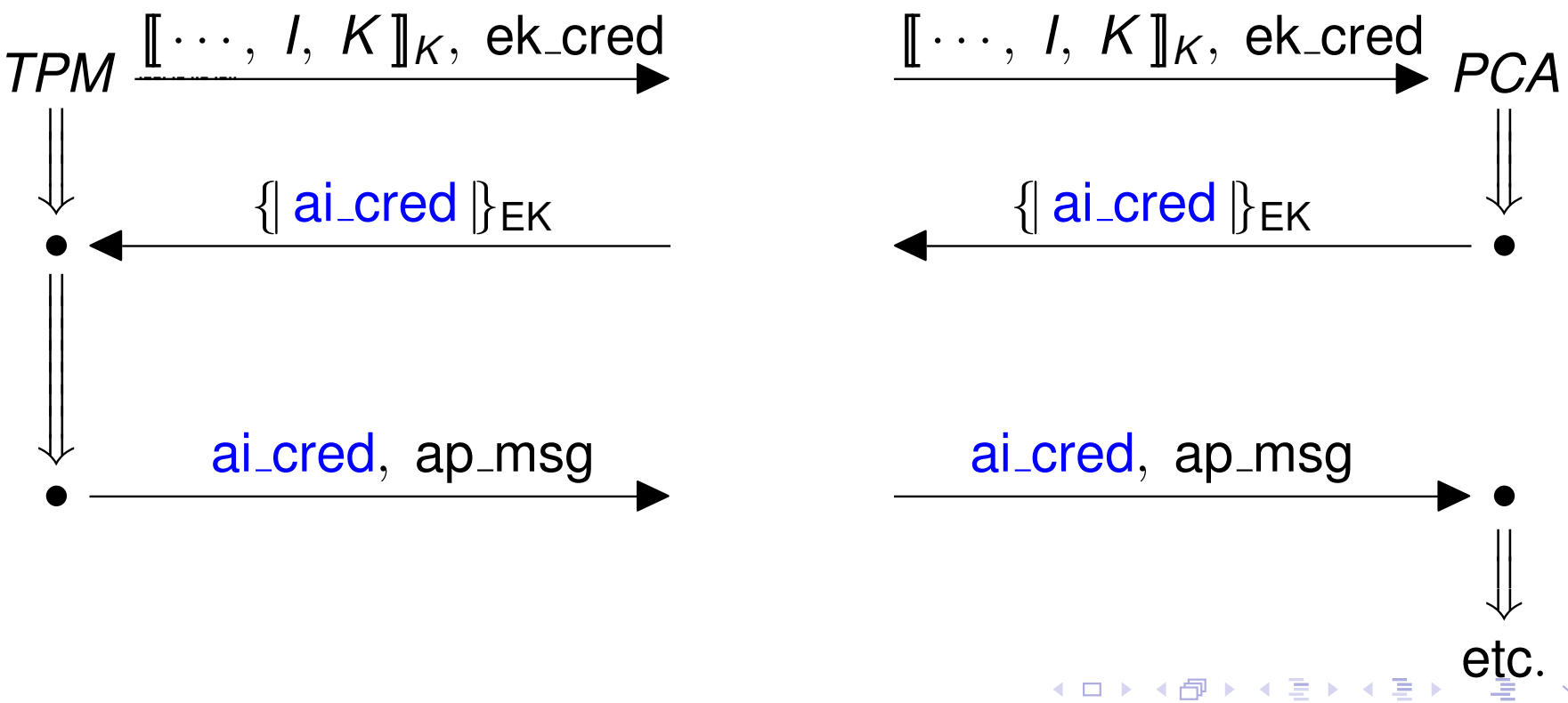
solutions, of three kinds

Transforming a cryptographic value

Old TCG Anonymous Identity protocol

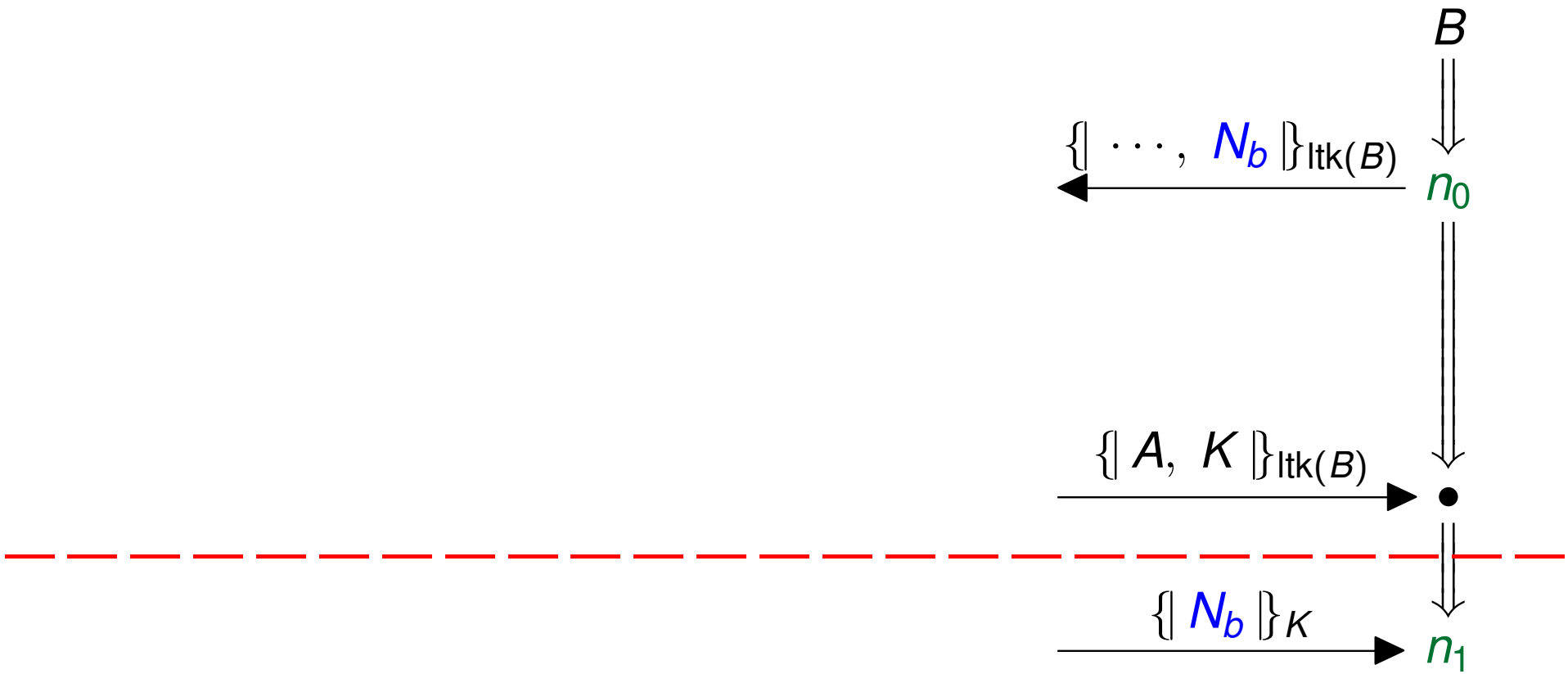
$$\text{ek\_cred} = [\![\cdots, \text{EK}]\!]_{\text{MFG}} \qquad \text{ai\_cred} = [\![\cdots, I, K]\!]_{\text{PCA}}$$

$$\text{ap\_msg} = [\![\cdots]\!]_K$$



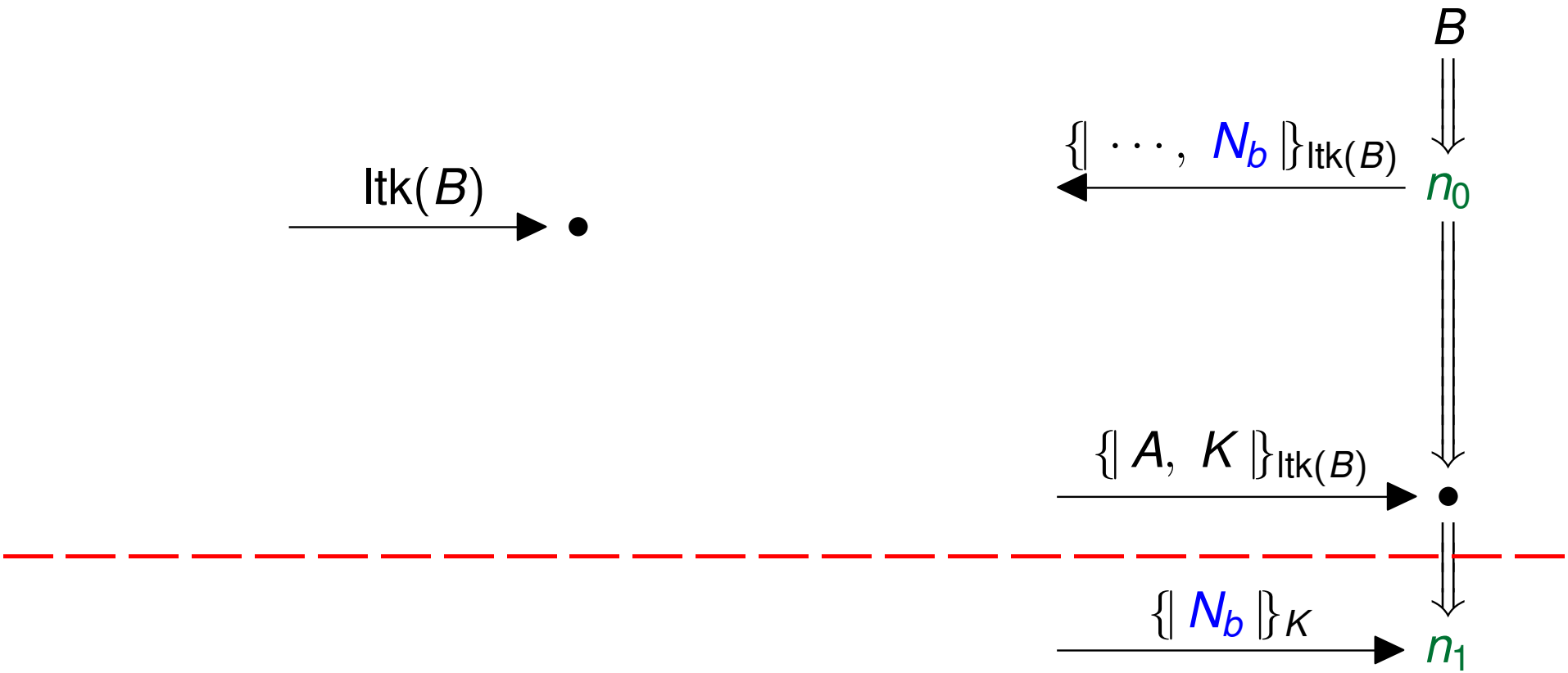
N_b escapes $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$ and $\{\!|\ \cdots,\ K,\ N_b\ |\!\}_{\mathrm{ltk}(A)}$

The test from n_0 to n_1



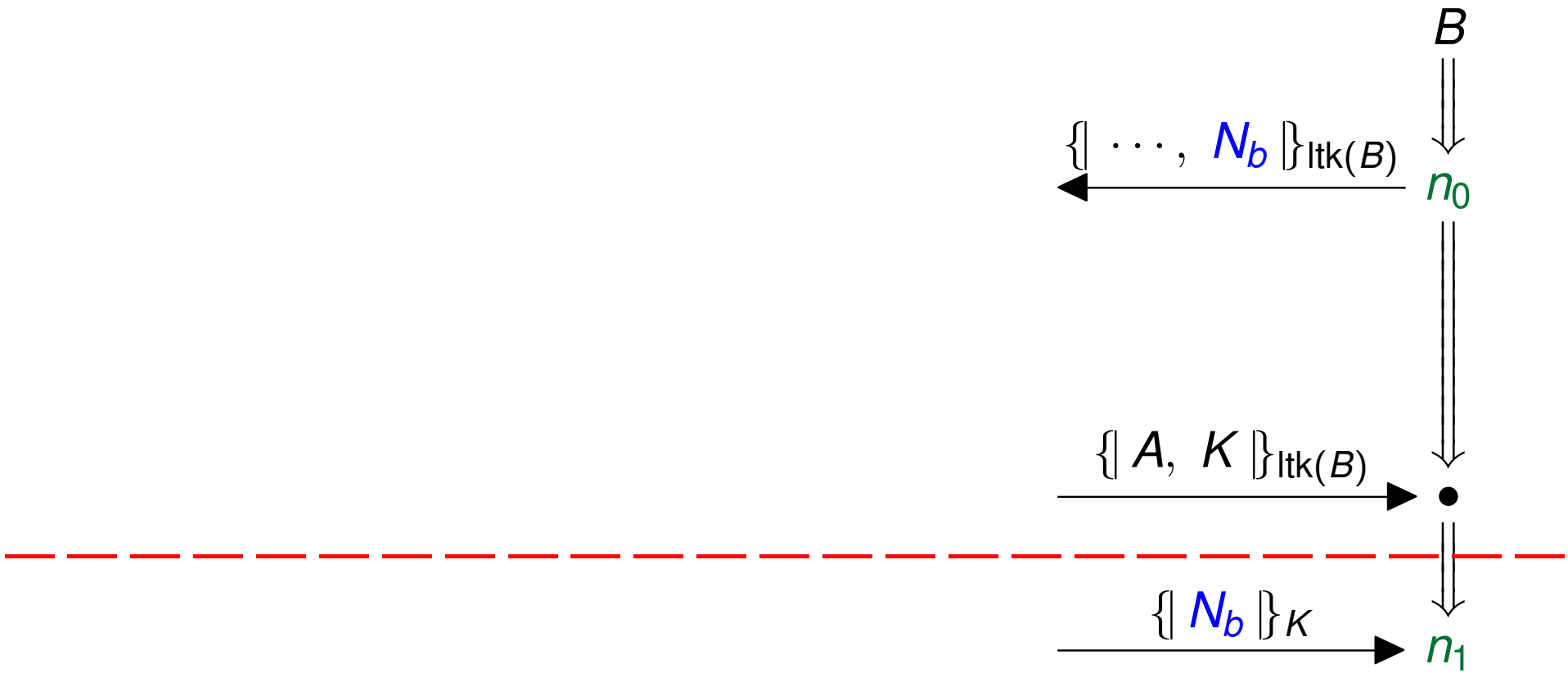
N_b escapes $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$ and $\{\!|\ \cdots,\ K,\ N_b\ |\!\}_{\mathrm{ltk}(A)}$

One solution



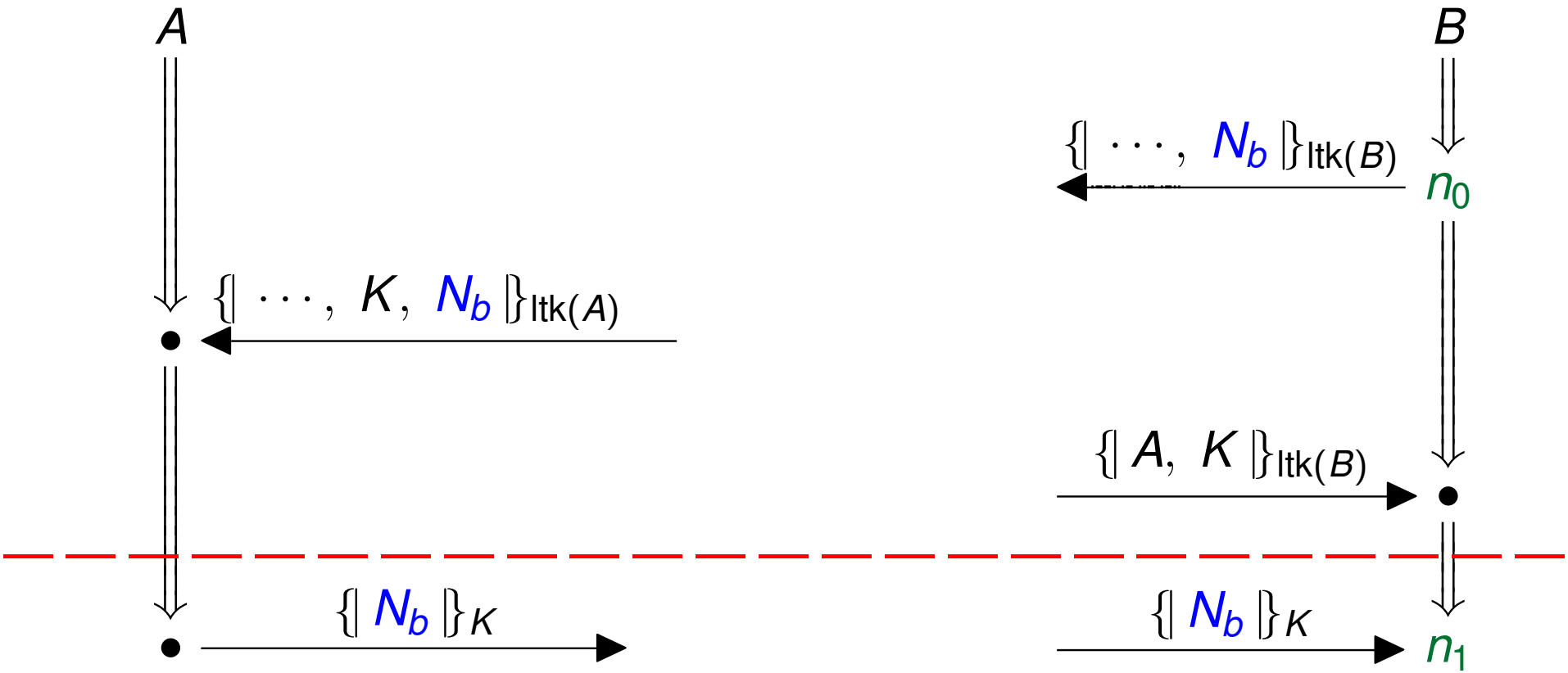
N_b escapes $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$ and $\{\!|\ \cdots,\ K,\ N_b\ |\!\}_{\mathrm{ltk}(A)}$

Same test



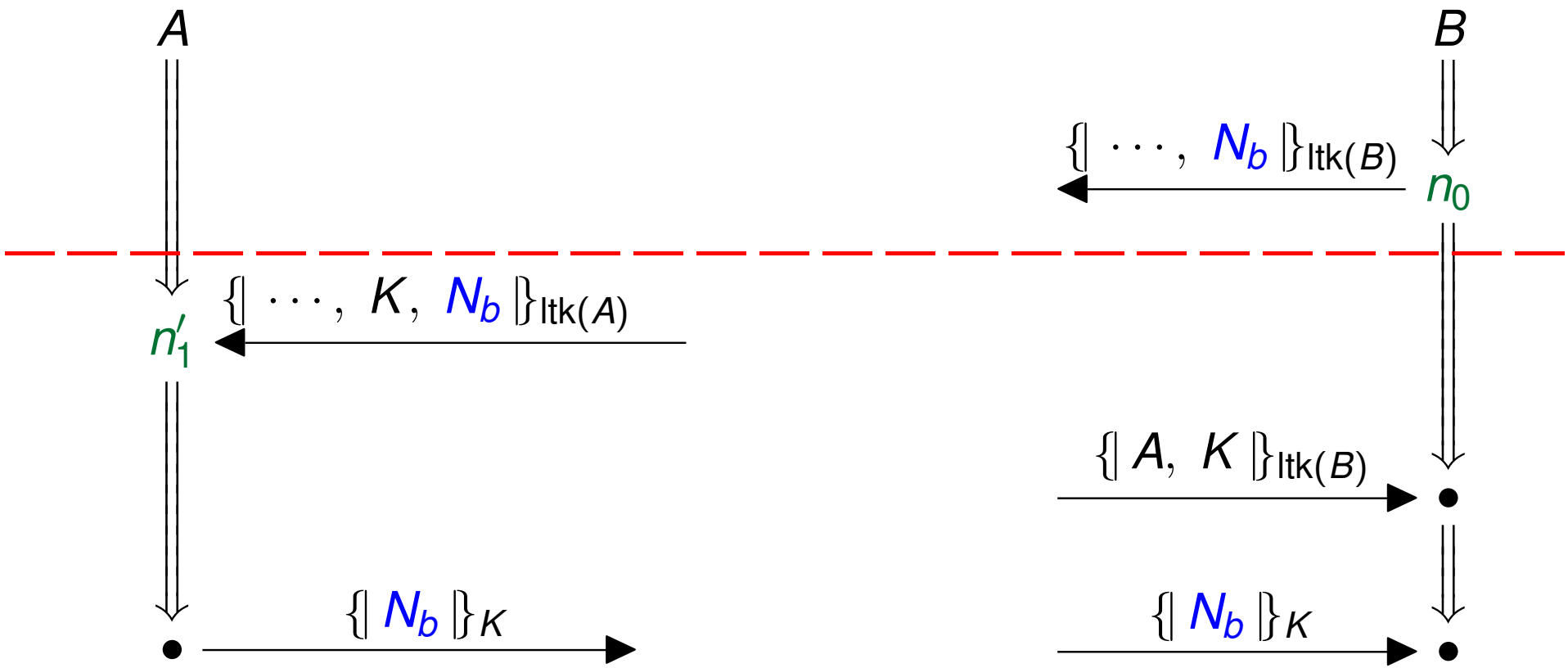
N_b escapes $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$ and $\{\!|\ \cdots,\ K,\ N_b\ |\!\}_{\mathrm{ltk}(A)}$

Another solution



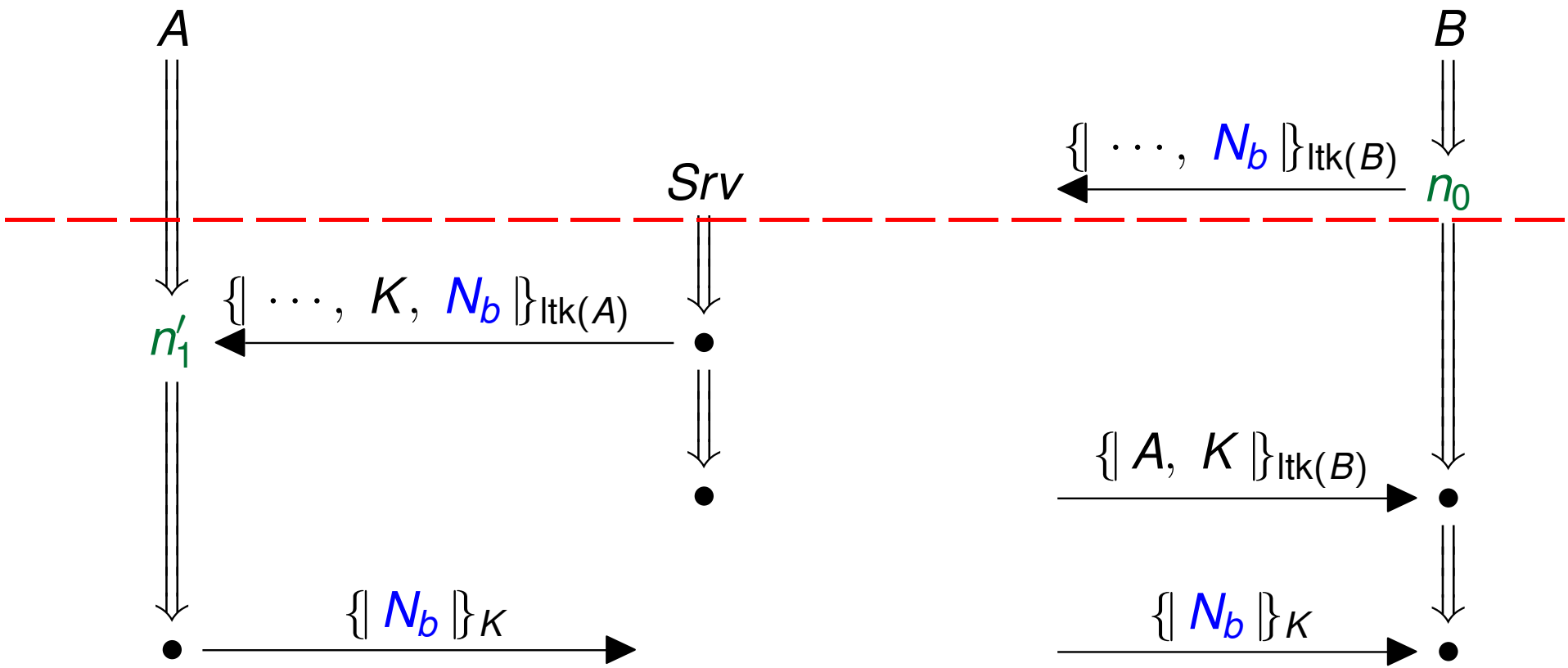
N_b escapes from $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$

Another test, from n_0 to n'_1



N_b escapes from $\{\!|\ \cdots,\ N_b\ |\!\}_{\mathrm{ltk}(B)}$

One solution



Form of the authentication test principles

Every test transformation has a solution

Soundness and completeness

Soundness:

In every possible execution, every test has a solution

Completeness:

Where every test has a solution, there is a possible execution

(Execution="bundle" $\mathcal{B}$;
tests and solutions exist in "skeletons" $\mathbb{A}$;
when some $\mathcal{B} = \mathbb{A} +$ adversary behavior,
then $\mathbb{A}$ is realized)

Restricting to Primary Protocol

Suppose for every realized $\mathbb{A}$,

$$\mathbb{A} \restriction \Pi_1 \text{ is also realized} \qquad *$$

- Then

  If $\mathbb{A}$ is a counterexample to a goal
  then so is $\mathbb{A} \restriction \Pi_1$
  in which no Π_2 behavior occurred

- Sufficient condition for $*$:

  For every authentication test in Π_1
  every solution in $\Pi_1 \cup \Pi_2$ lies in Π_1

Restricting to Primary Protocol

Suppose for every realized $\mathbb{A}$,

$$\mathbb{A} \restriction \Pi_1 \text{ is also realized} \qquad *$$

- Then

  If $\mathbb{A}$ is a counterexample to a goal
  then so is $\mathbb{A} \restriction \Pi_1$
  in which no Π_2 behavior occurred

- Sufficient condition for $*$:

  For every authentication test in Π_1
  every solution in $\Pi_1 \cup \Pi_2$ lies in Π_1

- Sufficiency is a consequence of completeness

Syntactic criteria for independence

$\Pi_1 \longrightarrow \Pi_2$

When	some $s_1 \in \Pi_1$ transmits $\{\!\|\ \ldots a \ldots\ \|\!\}_K$
and	some $s_2 \in \Pi_2$ receives it,
then	s_2 never re-transmits a in any new form

$\Pi_2 \longrightarrow \Pi_1$

When	some $s_2 \in \Pi_2$ transmits $\{\!\|\ t\ \|\!\}_K$
and	some $s_1 \in \Pi_1$ receives it,
then	s_2 previously received $\{\!\|\ t\ \|\!\}_K$ in the same form

Restriction: Definition

$\mathbb{A} \restriction \Pi_1 = \mathbb{A}'$ where

1. $\mathsf{nodes}(\mathbb{A}') =$

$$\{n \in \mathsf{nodes}(\mathbb{A}) : n \text{ lies on some } s \in \Pi_1\}$$

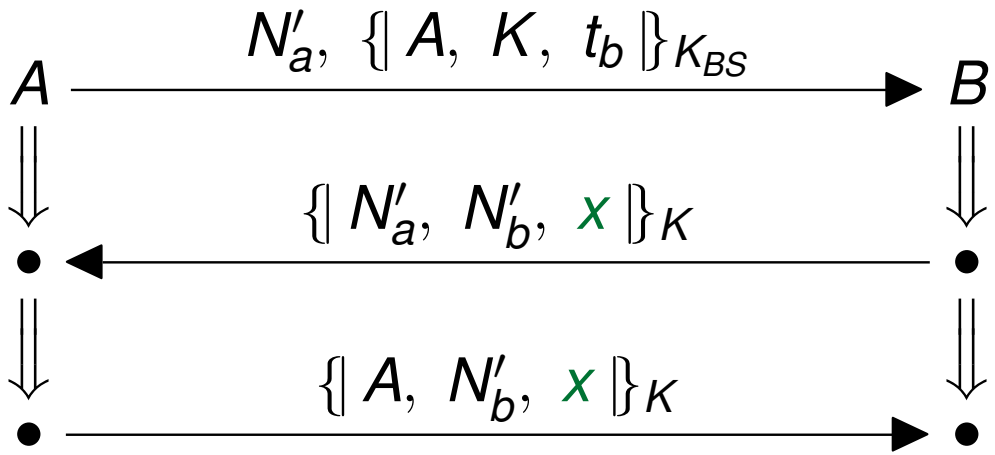
2. $n_0 \preceq_{\mathbb{A}'} n_1$ iff $n_0, n_1 \in \mathbb{A}'$ and $n_0 \preceq_{\mathbb{A}} n_1$
3. $\mathsf{non}_{\mathbb{A}'} =$

$$\mathsf{non}_{\mathbb{A}} \cap \{K : K \text{ used for encryption in some } n \in \mathsf{nodes}(\mathbb{A}')\}$$

4. $\mathsf{unique}_{\mathbb{A}'} =$

$$\mathsf{unique}_{\mathbb{A}} \cap \{a : a \text{ originates on some } n \in \mathsf{nodes}(\mathbb{A}')\}$$

Limitation of this criterion



More inclusive criterion

Consider roles $r \in \Pi_1, \Pi_2$

$\Pi_1 \longrightarrow \Pi_2$

When	some $r_1 \in \Pi_1$ transmits t_1 with subterm $\{\!\lvert \ldots a \ldots \rvert\!\}_K = t_1'$
and	some $r_2 \in \Pi_2$ receives a msg with subterm unifying under α,
then	r_2 never re-transmits $a \cdot \alpha$ outside $t_1' \cdot \alpha$

$\Pi_2 \longrightarrow \Pi_1$

When	some $r_2 \in \Pi_2$ transmits t_1 with subterm $\{\!\lvert t \rvert\!\}_K$
and	some $s_1 \in \Pi_1$ receives a msg with subterm unifying under α,
then	s_2 previously received $\{\!\lvert t \rvert\!\}_K$ in the same form

Formalizing completeness

Where every test has a solution, there is a possible execution

- Decisions required for formalization
  - What sort of thing satisfies/fails the authentication test principles?
  - What sort of thing is a possible execution?

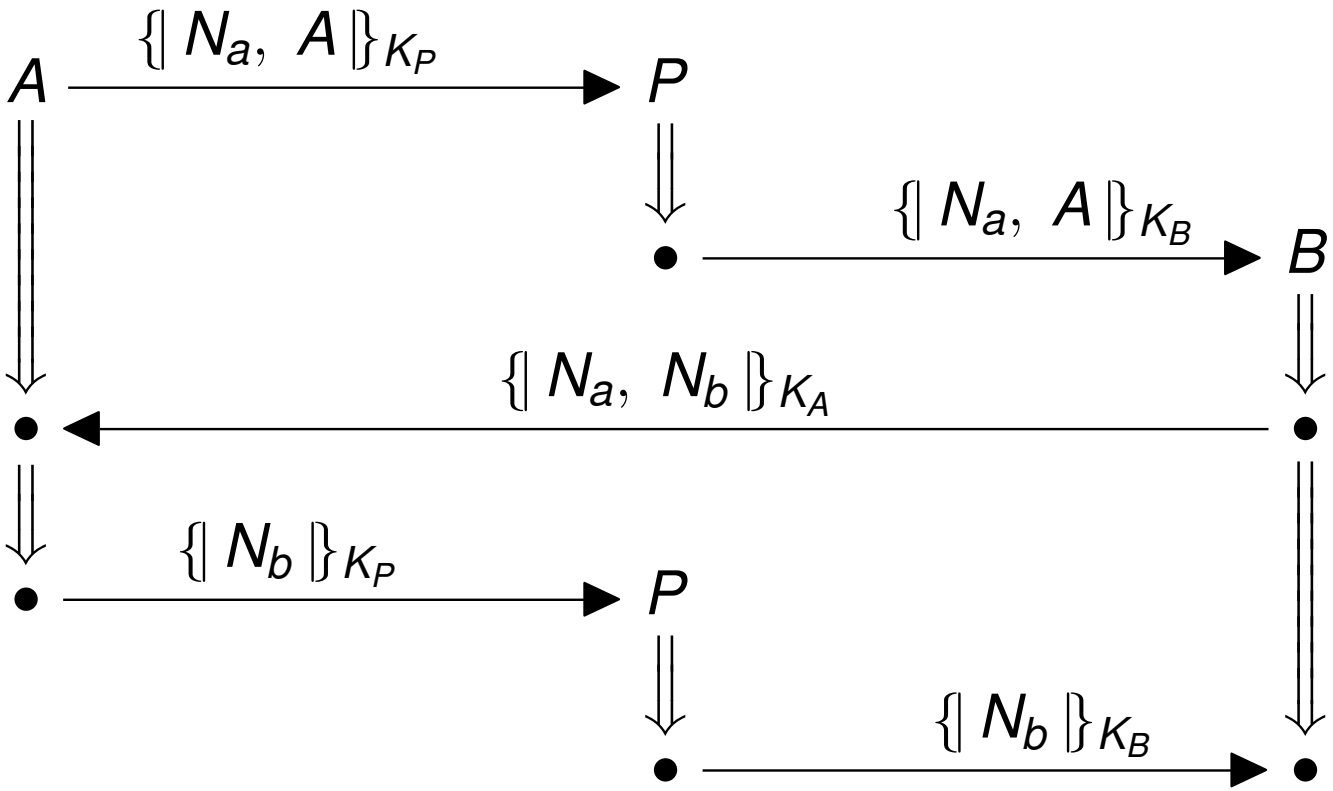
Formalizing completeness

Where every test has a solution, there is a possible execution

- Decisions required for formalization
  - What sort of thing satisfies/fails the authentication test principles?
  - What sort of thing is a possible execution?
    bundles

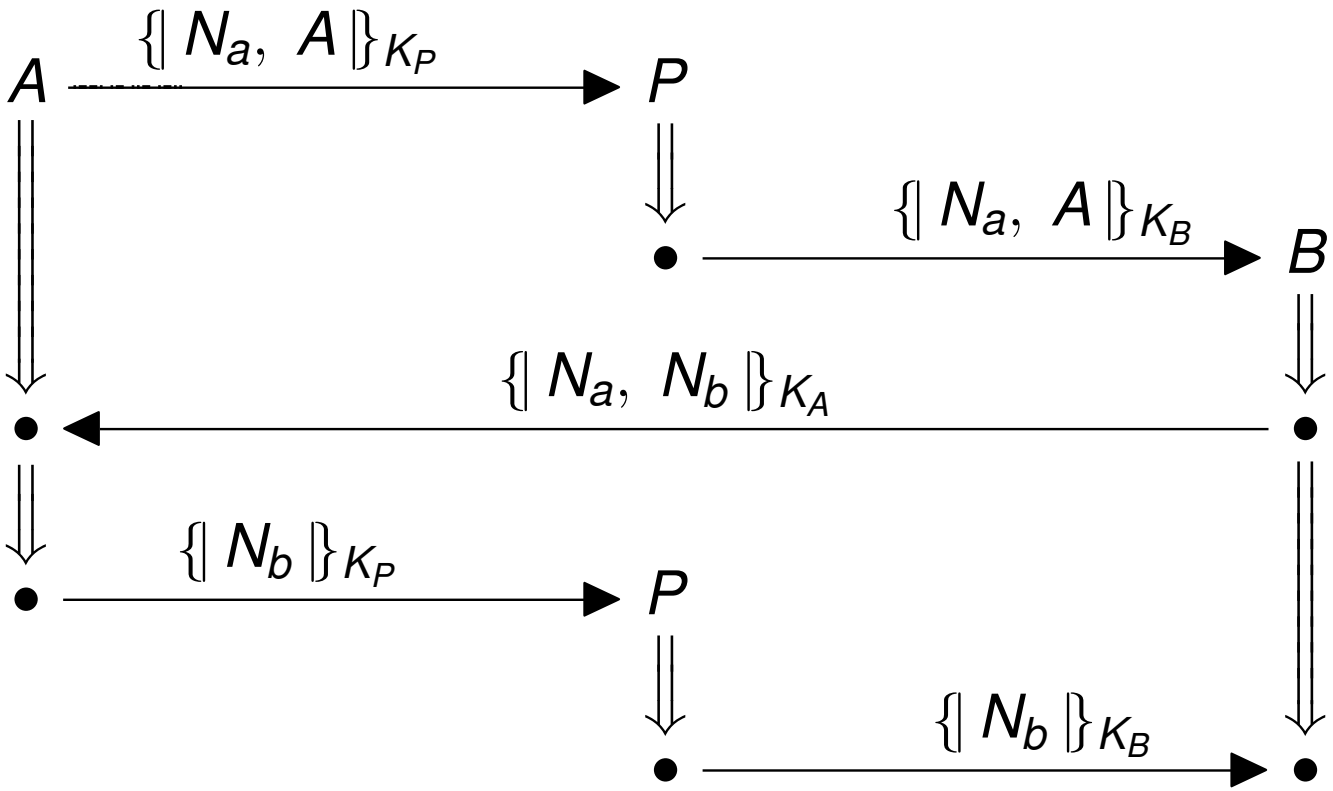
“Execution” means bundle

Example: A Bundle $\mathbb{B}$ for Needham-Schroeder



“Execution” means bundle

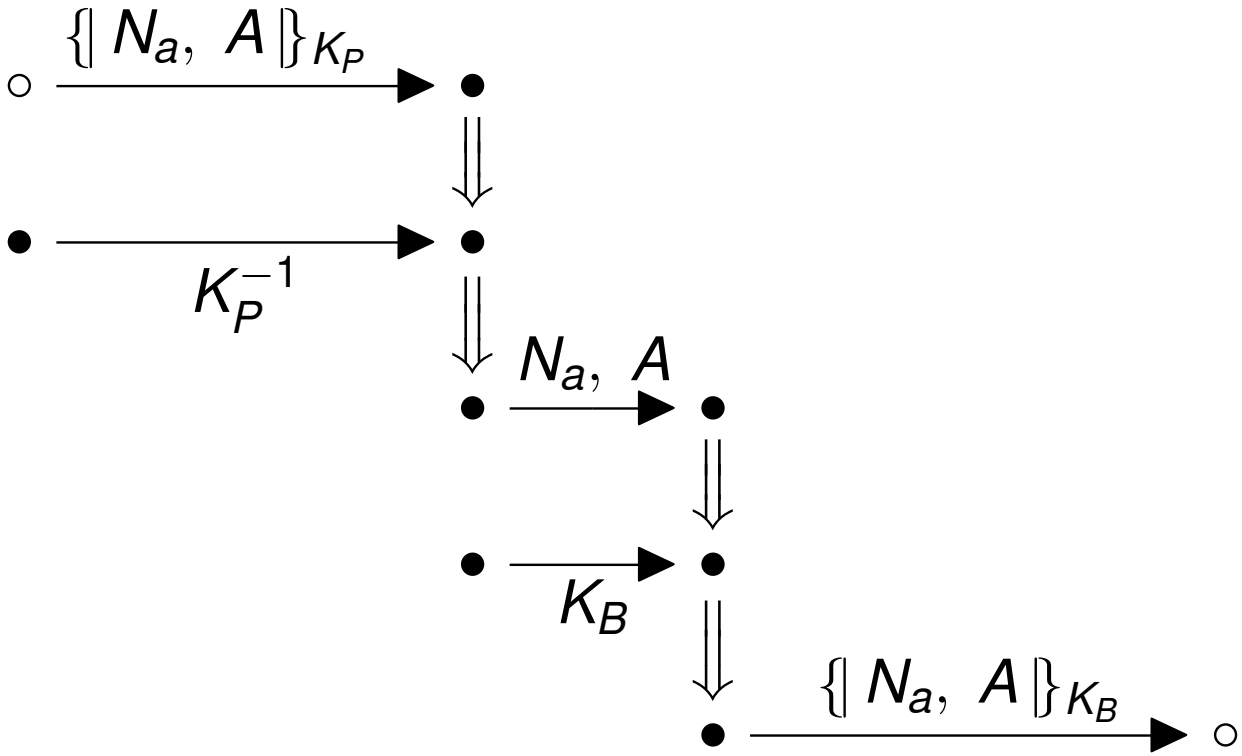
Example: A Bundle $\mathbb{B}$ for Needham-Schroeder



Vertical columns are strands

“Execution” means bundle

NS penetrator web



“Execution” means bundle

Definition of bundle

- Causally well founded directed graph made of
  - Instances of the protocol role behaviors
  - Adversary behaviors
- “Causally well founded” means:
  - Every reception node has one incoming message arrow
    - ★ Every message received was sent
  - Closed backward along strands
    - ★ If later event occurred, earlier event occurred too
  - Graph is finite and acyclic

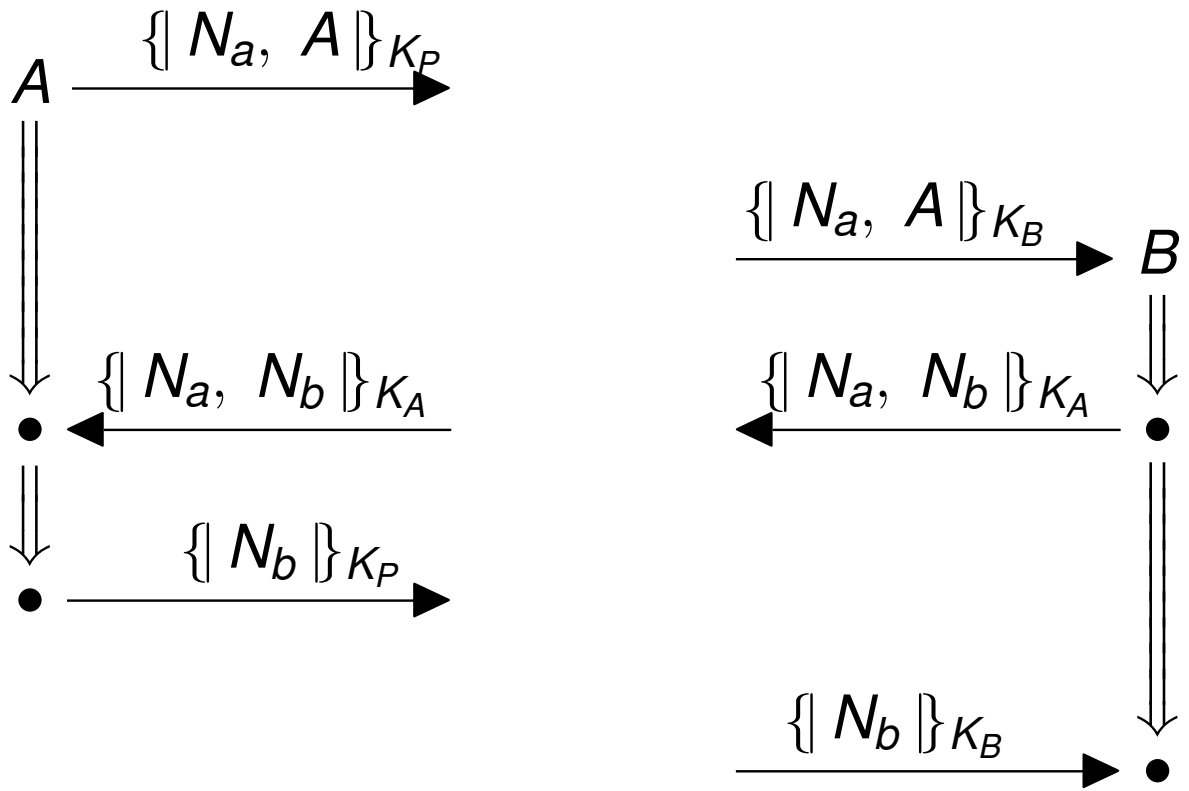
Formalizing completeness

Where every test has a solution, there is a possible execution

- Decisions required for formalization
  - What sort of thing satisfies/fails the authentication test principles?
    skeletons
  - What sort of thing is a possible execution?
    bundles

Example: skeleton $\mathbb{A}$

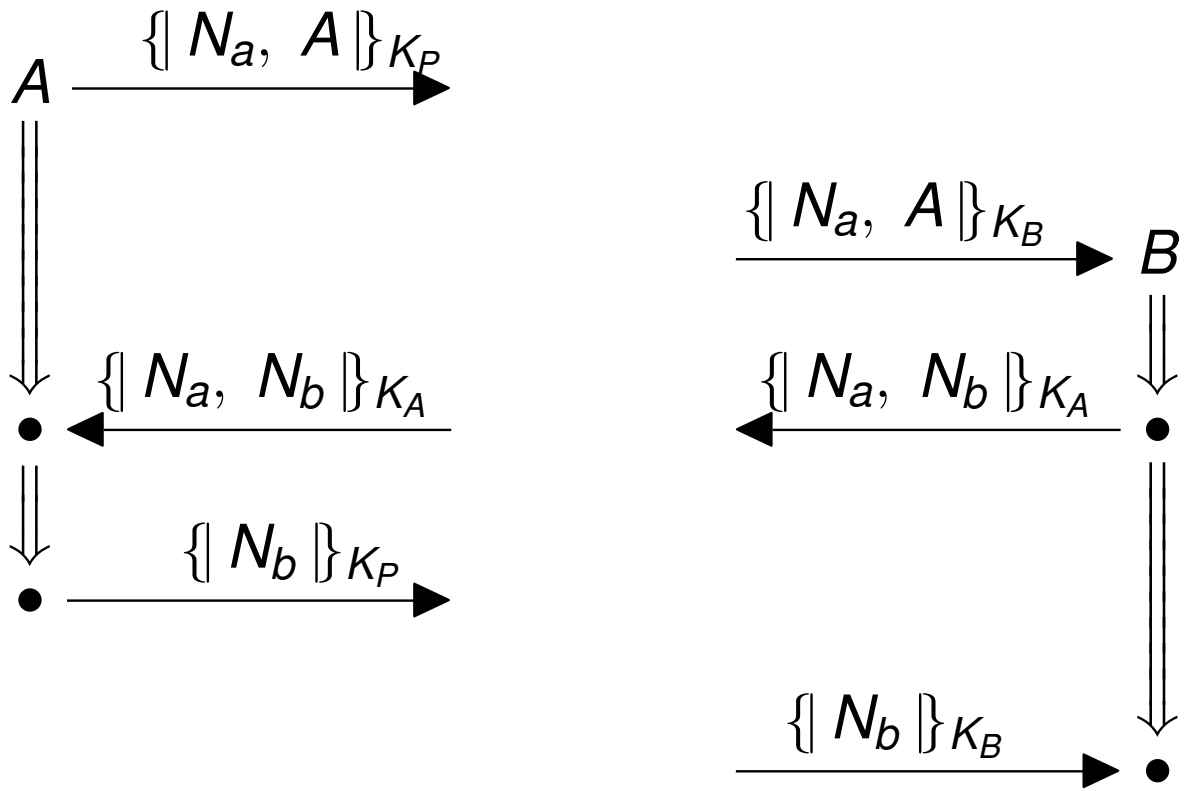
Bundle $\mathbb{B}$ minus adversary behavior



N_b fresh, K_A^{-1} uncompromised

Example: skeleton $\mathbb{A}$

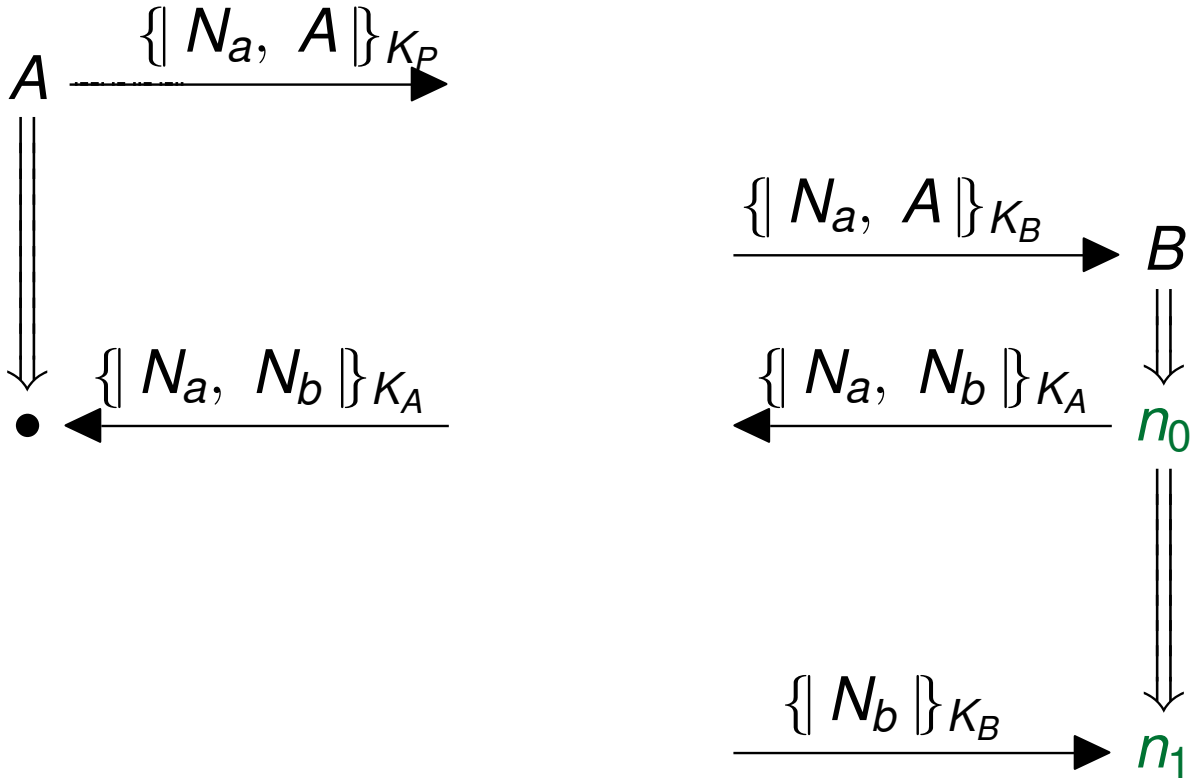
Bundle $\mathbb{B}$ minus adversary behavior



N_b fresh, K_A^{-1} uncompromised

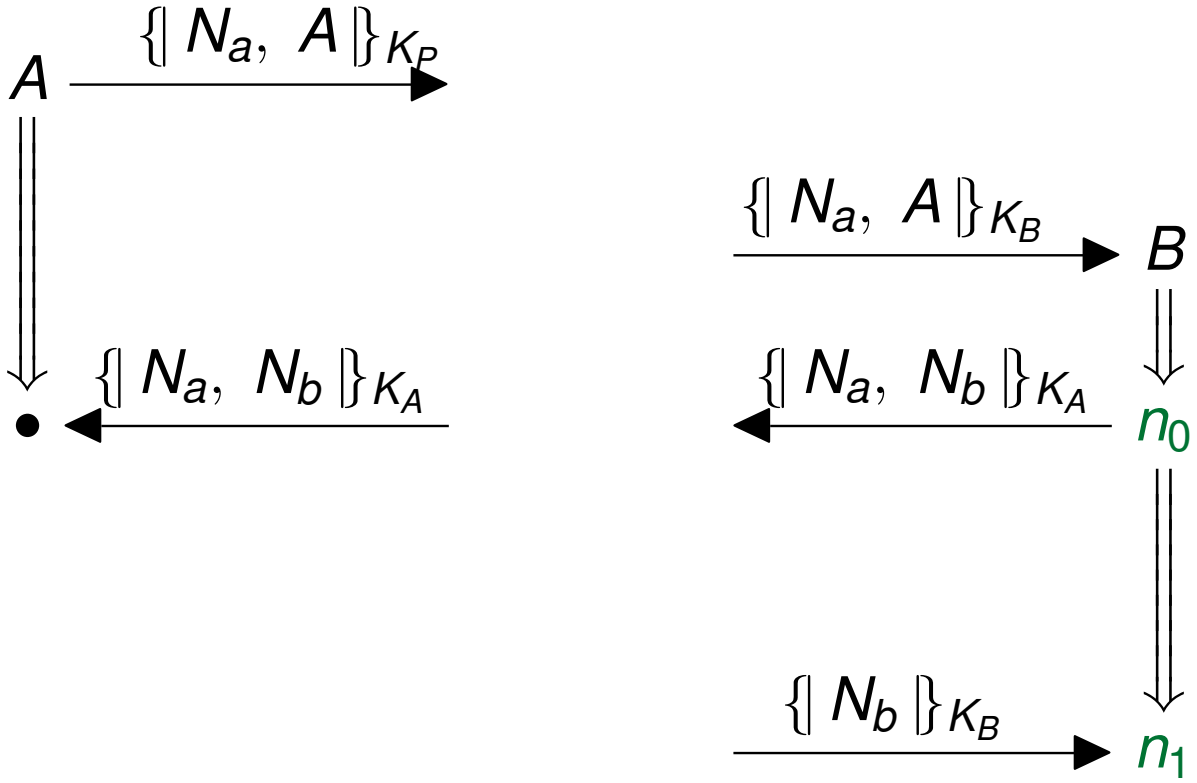
$\mathbb{A} = \text{skeleton}(\mathbb{B})$

Example: Unrealized skeleton $\mathbb{A}'$



N_b fresh, K_A^{-1} uncompromised

Example: Unrealized skeleton $\mathbb{A}'$



N_b fresh, K_A^{-1} uncompromised
For every bundle $\mathbb{B}'$, $\mathbb{A}' \neq \text{skeleton}(\mathbb{B}')$

Skeletons: Definition

A skeleton $\mathbb{A}$ is a 4-tuple

- Set of regular events, nodes($\mathbb{A}$)
- A "before" partial ordering $\preceq_{\mathbb{A}}$ on nodes($\mathbb{A}$)
- Set of values assumed fresh, $\text{unique}_{\mathbb{A}}$
- Set of keys assumed never transmitted, $\text{non}_{\mathbb{A}}$

Skeletons: Definition

A skeleton $\mathbb{A}$ is a 4-tuple

- Set of regular events, nodes($\mathbb{A}$)
- A "before" partial ordering $\preceq_{\mathbb{A}}$ on nodes($\mathbb{A}$)
- Set of values assumed fresh, $\text{unique}_{\mathbb{A}}$
- Set of keys assumed never transmitted, $\text{non}_{\mathbb{A}}$

$\mathbb{A}$ is realized if

$$\mathbb{A} = \text{skeleton}(\mathbb{B})$$

for any bundle $\mathbb{B}$

Completeness of the authentication tests

For all skeletons $\mathbb{A}$, if

$\mathbb{A}$ *satisfies the authentication tests*

then

$\mathbb{A}$ *is realized*

Completeness of the authentication tests

For all skeletons $\mathbb{A}$, if

$\mathbb{A}$ *satisfies the authentication tests*

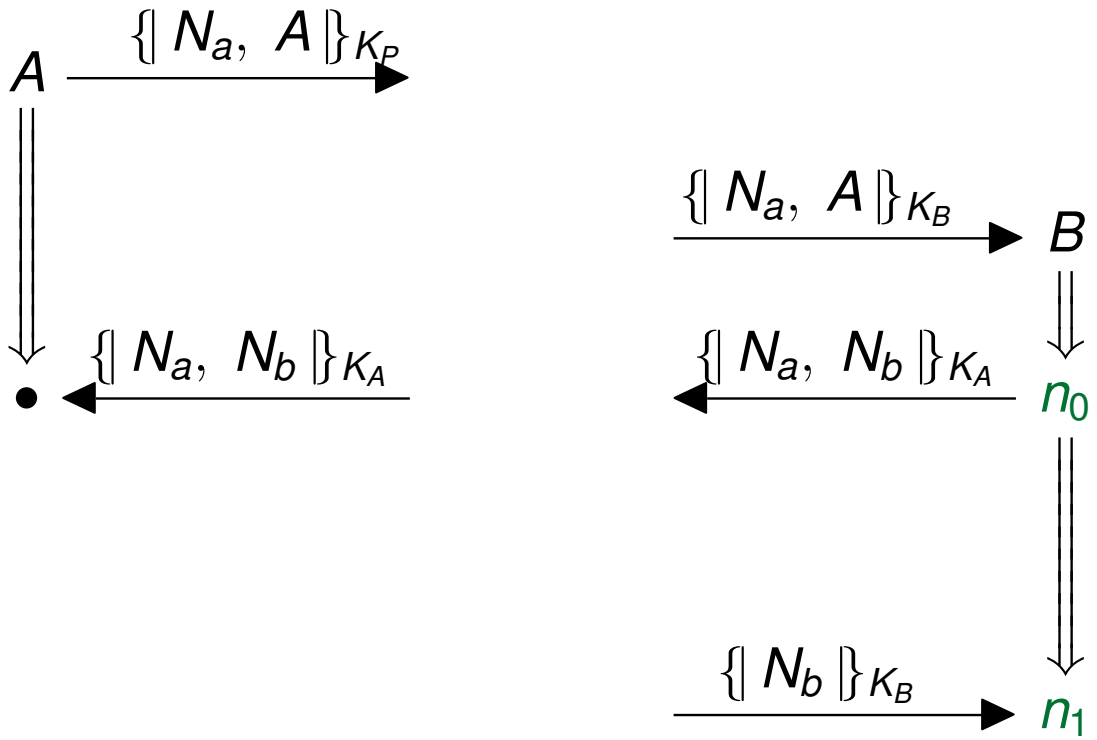
then

$\mathbb{A}$ *is realized*

There is a bundle $\mathbb{B}$ *such that* $\mathbb{A} = \mathsf{skeleton}(\mathbb{B})$

Outgoing test not satisfied

N_b fresh, K_A^{-1} uncompromised



Outgoing test from n_0 to n_1 has no solution

Completeness: Proof idea

- Must show, for all $n_1 \in \mathbb{A}$

  If there's no unsolved test for n_1
  then msg(n_1) is derivable from earlier nodes

- Derivation normal form

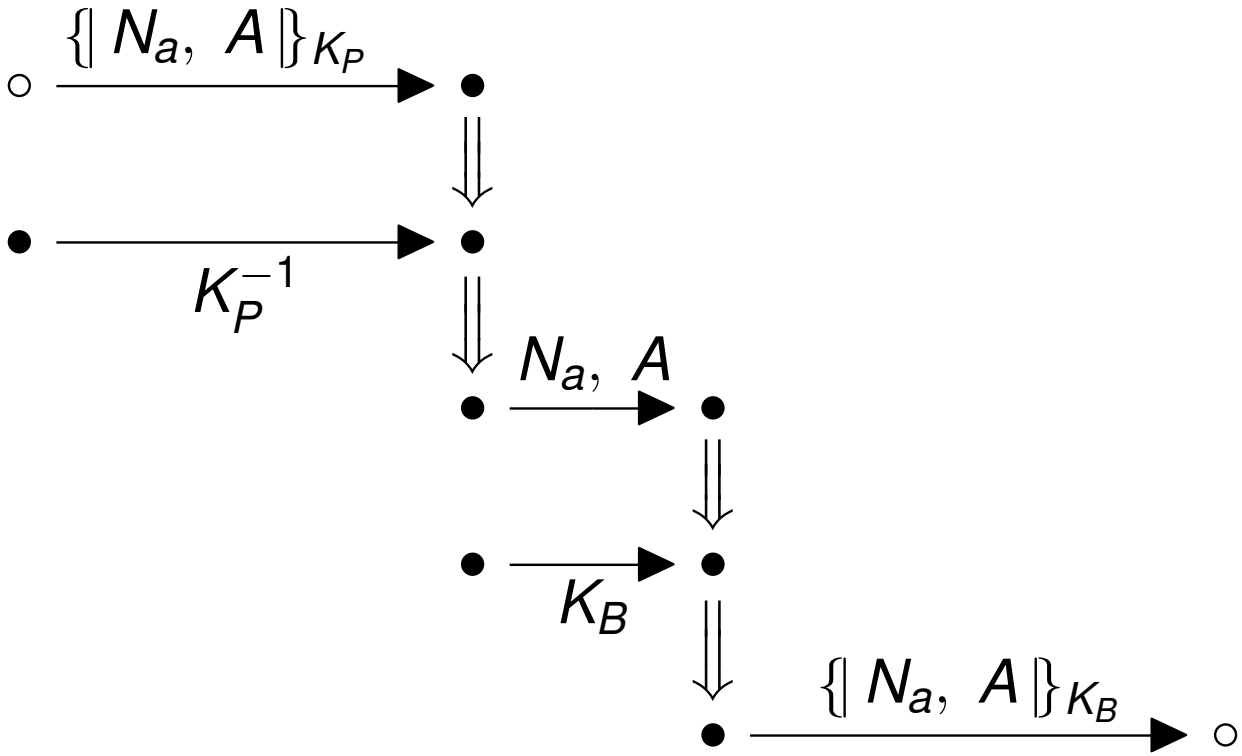
  De-structure then construct

- Structural induction on

$$\text{msg}(n_1), \quad \{\text{msg}(m) : m \prec_{\mathbb{A}} n\}$$

Decryption precedes encryption

NS penetrator web



Completeness: Proof idea

- Must show, for all $n_1 \in \mathbb{A}$

  If there's no unsolved test for n_1
  then msg(n_1) is derivable from earlier nodes

- Derivation normal form

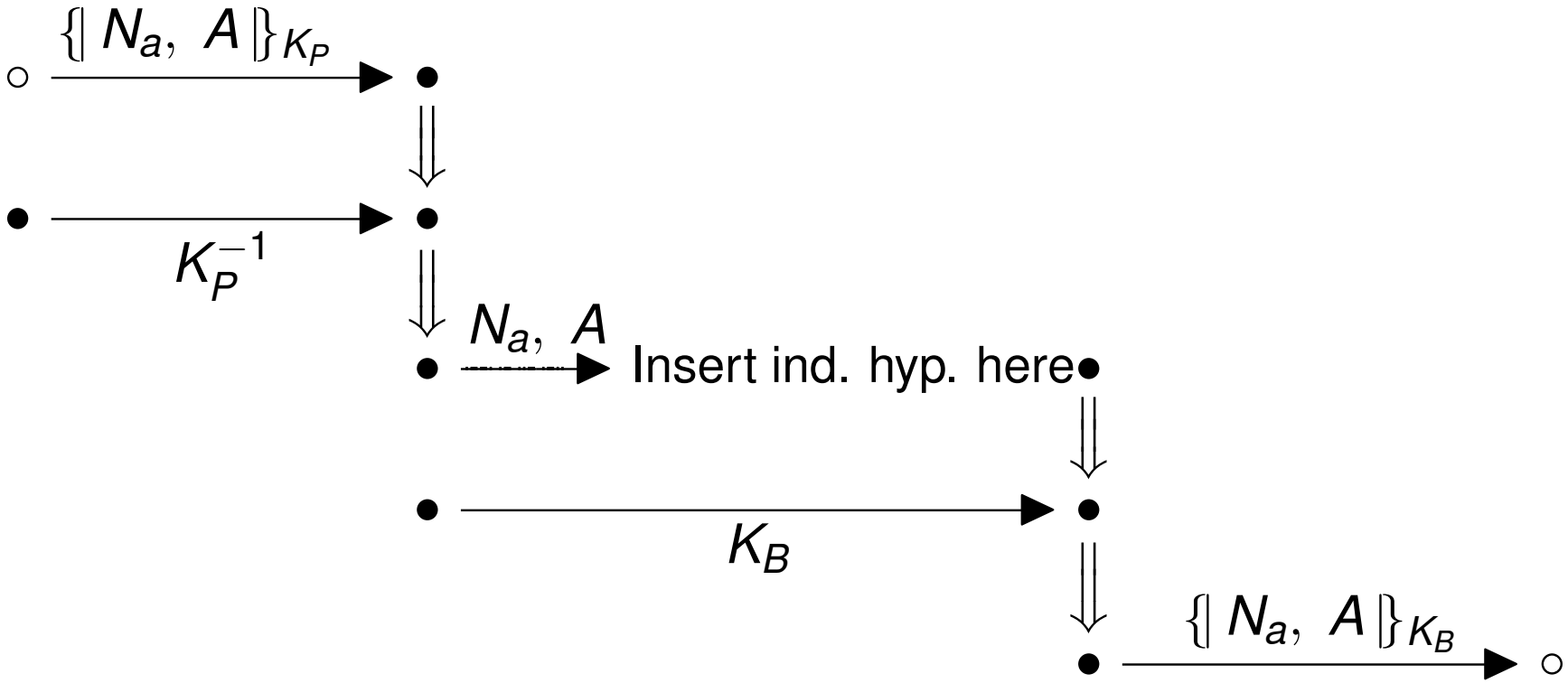
  De-structure then construct

- Structural induction on

$$\text{msg}(n_1), \quad \{\text{msg}(m) : m \prec_{\mathbb{A}} n\}$$

Inductive step

NS penetrator web



Mechanizing the authentication tests: CPSA

- Cryptographic Protocol Shapes Analyzer
  - Shapes: Minimal, essentially different, realized skeletons
- CPSA starts with a small skeleton $\mathbb{A}_0$
  - For each unsolved authentication test, it explores every solution
  - When exploration leads to $\mathbb{A}_1$ where every test has a solution, then $\mathbb{A}_1$ is a shape
- Completeness theorem guarantees:

  CPSA finds all the shapes

Protocol Composition and Refinement

- When protocols Π_1, Π_2 used in combination
  - Can a security goal fail for $\Pi_1 \cup \Pi_2$, if it holds for Π_1 alone?
  - No, unless

    Π_2 *provides a new solution to some test in* Π_1

Protocol Composition and Refinement

- When protocols Π_1, Π_2 used in combination
  - Can a security goal fail for $\Pi_1 \cup \Pi_2$, if it holds for Π_1 alone?
  - No, unless

    Π_2 *provides a new solution to some test in* Π_1

- When Π_2 refines protocol Π_1 by adding information
  - Can Π_2 falsify a security goal that holds for Π_1?
  - No, unless for some $\tau_2 = (t, S, n_1)$ in Π_2,

    τ_2 *refines a test* τ_1 *in* Π_1,
    but some solution for τ_2
    does not refine any solution for τ_1 *in* Π_1

Protocol Composition and Refinement

- When protocols Π_1, Π_2 used in combination
  - Can a security goal fail for $\Pi_1 \cup \Pi_2$, if it holds for Π_1 alone?
  - No, unless

    Π_2 *provides a new solution to some test in* Π_1

- When Π_2 refines protocol Π_1 by adding information
  - Can Π_2 falsify a security goal that holds for Π_1?
  - No, unless for some $\tau_2 = (t, S, n_1)$ in Π_2,

    τ_2 *refines a test* τ_1 *in* Π_1,
    but some solution for τ_2
    does not refine any solution for τ_1 *in* Π_1

- Authentication tests clarify protocol design