Relating Strands and Multiset Rewriting For Security Protocol Analysis

Iliano Cervesato

Nancy Durgin, Patrick Lincoln

John Mitchell, Andre Scedrov

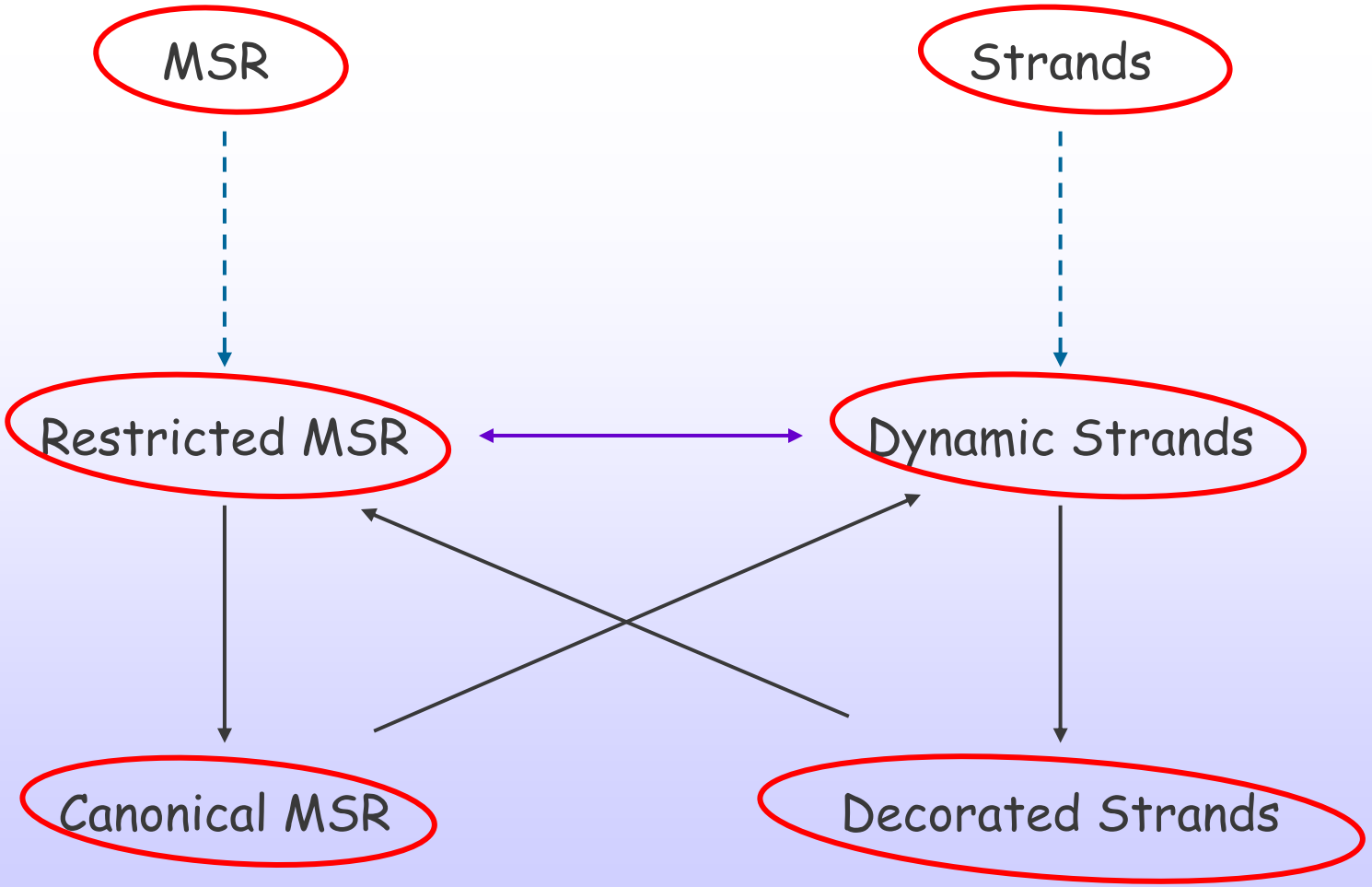
Representing Security Protocols

Several recent proposal based on the Dolev-Yao model:

- Strand spaces
- Multiset rewriting
- Spi-calculus, …

How are they related?

Roadmap


MSR
Strands
Restricted MSR
Dynamic Strands
Canonical MSR
Decorated Strands

Running Example

Needham-Schroeder Protocol

$$A \rightarrow B: \{N_A, A\}_{KB}$$

$$B \rightarrow A: \{N_A, N_B\}_{KA}$$

$$A \rightarrow B: \{N_B\}_{KB}$$

MSR

- Executable specification language
- Adapts multiset rewriting with ∃
  - Solid logical foundation
  - Ties with linear logic and process algebra
- Flexible and fully precise
- Follows the Dolev-Yao model

Multiset rewriting ...

- Multiset: set with repetitions allowed
- Rewrite rule:

$$r: N_1 \rightarrow N_2$$

- Application

$$M_1 \xrightarrow{r} M_2$$

$$\underbrace{M', N_1} \xrightarrow{r} \underbrace{M', N_2}$$

- Multi-step transition, reachability

... with existentials

- msets of 1st-order atomic formulas
- Rules:

$$r: F(\underline{x}) \rightarrow \exists \underline{n}.\ G(\underline{x},\underline{n})$$

- Application

$$\underbrace{M_1} \xrightarrow{r} \underbrace{M_2}$$

$$\underbrace{M', F(\underline{t})} \xrightarrow{r} \underbrace{M', G(\underline{t},\underline{c})}$$

$\underline{c}$ not in M_1

MSR predicates

- N(m) — Network messages
- I(m) — Intruder info.
- $A_i(t_1,\ldots,t_{ni})$ — Role states
- *Pr*, *PrvK*, *PubK*, … — Persistent info.

Protocol Theories

- Initialization rules
- For each role
  - 1 role generation rule
  - n execution rules

MSR → Restricted MSR

- Assume initialization has already happened
- Initial info: Π

? No initialization in strands

NS: MSR rules for Alice

$$\pi_{A0}(A) \rightarrow A_0(A), \pi_{A0}(A)$$

$$A_0(A), \pi_{A1}(B) \rightarrow \exists N_A.\ A_1(A,B,N_A), N(\{N_A,A\}_{KB}), \pi_{A1}(B)$$

$$A_1(A,B,N_A), N(\{N_A,N_B\}_{KA}) \rightarrow A_2(A,B,N_A,N_B)$$

$$A_2(A,B,N_A,N_B) \rightarrow A_3(A,B,N_A,N_B), N(\{N_B\}_{KB})$$

where $\pi_{A0}(A) = Pr(A), PrvK(A,K_A^{-1})$

$\pi_{A1}(B) = Pr(B), PubK(B,K_B)$

NS: MSR rules for Bob

$$\pi_{B0}(B) \rightarrow B_0(B), \pi_{B0}(B)$$

$$B_0(A), \pi_{B1}(A), N(\{N_A, A\}_{KB}) \rightarrow B_1(A, B, N_A), \pi_{B1}(A)$$

$$B_1(A, B, N_A) \rightarrow \exists N_B.\ B_2(A, B, N_A, N_B), N(\{N_A, N_B\}_{KA})$$

$$B_2(A, B, N_A, N_B), N(\{N_B\}_{KB}) \rightarrow B_3(A, B, N_A, N_B)$$

where $\pi_{B0}(B) = Pr(B), PrvK(B, K_B^{-1})$

$\pi_{B1}(A) = Pr(A), PubK(A, K_A)$

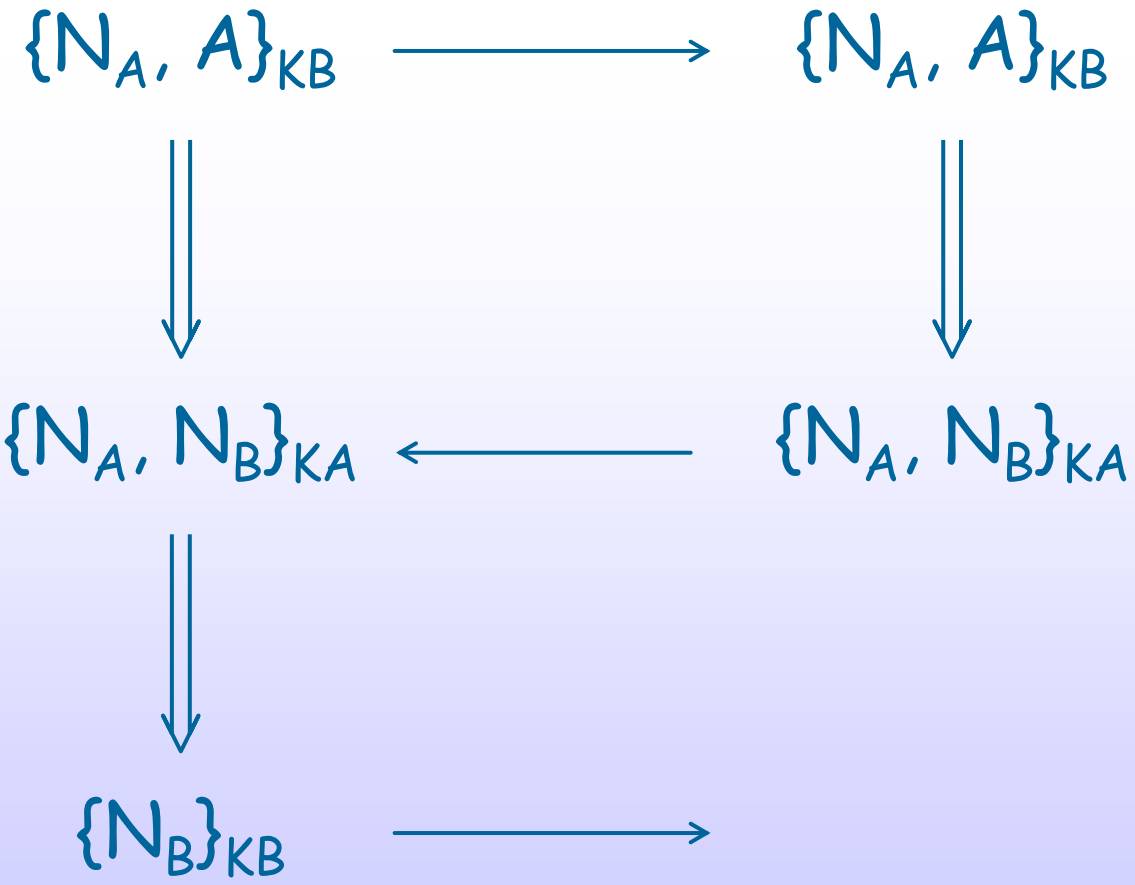
MSR Intruder

- Implement the Dolev-Yao model
  - Decryption/Encryption
  - Decomposition/composition
  - Nonce generation
  - …
- Expressed within the language

Strands

- Graphical representation of execution
- Designed for after-the-fact analysis
- Very simple
- Follow the Dolev-Yao model
- Related to
  - Lamport's causality
  - Mazurkiewicz's traces

NS: A Bundle



Penetrator Strands

- Implement the Dolev-Yao model
  - Decryption/Encryption
  - Decomposition/composition
  - Nonce generation
  - …
- Expressed within the language

Strands ⟶ Dyn. Strands

? Support executable specifications

- Specification language
  - Parametric strands
- Execution capabilities
  - Configurations
  - Transitions

Parametric strands

- Strands are instances of *roles*
- Parameters: instantiable information
- Constraints:
  - Nonces
  - Persistent info.

NS: Parametric Strand for *Alice*

Alice (A, B, N_A, N_B) :

N_A Fresh, $\pi_A(A,B)$

$\{N_A, A\}_{KB} \longrightarrow$

$\Downarrow$

$\{N_A, N_B\}_{KA} \longleftarrow$

$\Downarrow$

$\{N_B\}_{KB} \longrightarrow$

where

$\pi(A,B) = Pr(A), PrvK(A,K_A^{-1}), Pr(B), PubK(B,K_B)$

NS: Parametric Strand for *Bob*

Bob (A,B,N_A,N_B) :

N_B Fresh, $\pi_B(A,B)$

$\longrightarrow$ $\{N_A, A\}_{KB}$

$\Downarrow$

$\longleftarrow$ $\{N_A, N_B\}_{KA}$

$\Downarrow$

$\longrightarrow$ $\{N_B\}_{KB}$

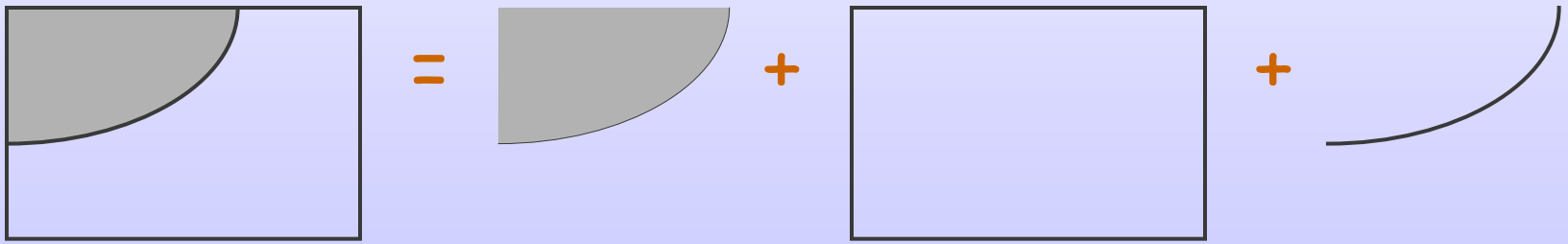
where

$\pi(A,B) =$ $Pr(B)$, $PrvK(B,K_B^{-1})$,
$Pr(A)$, $PubK(A,K_A)$

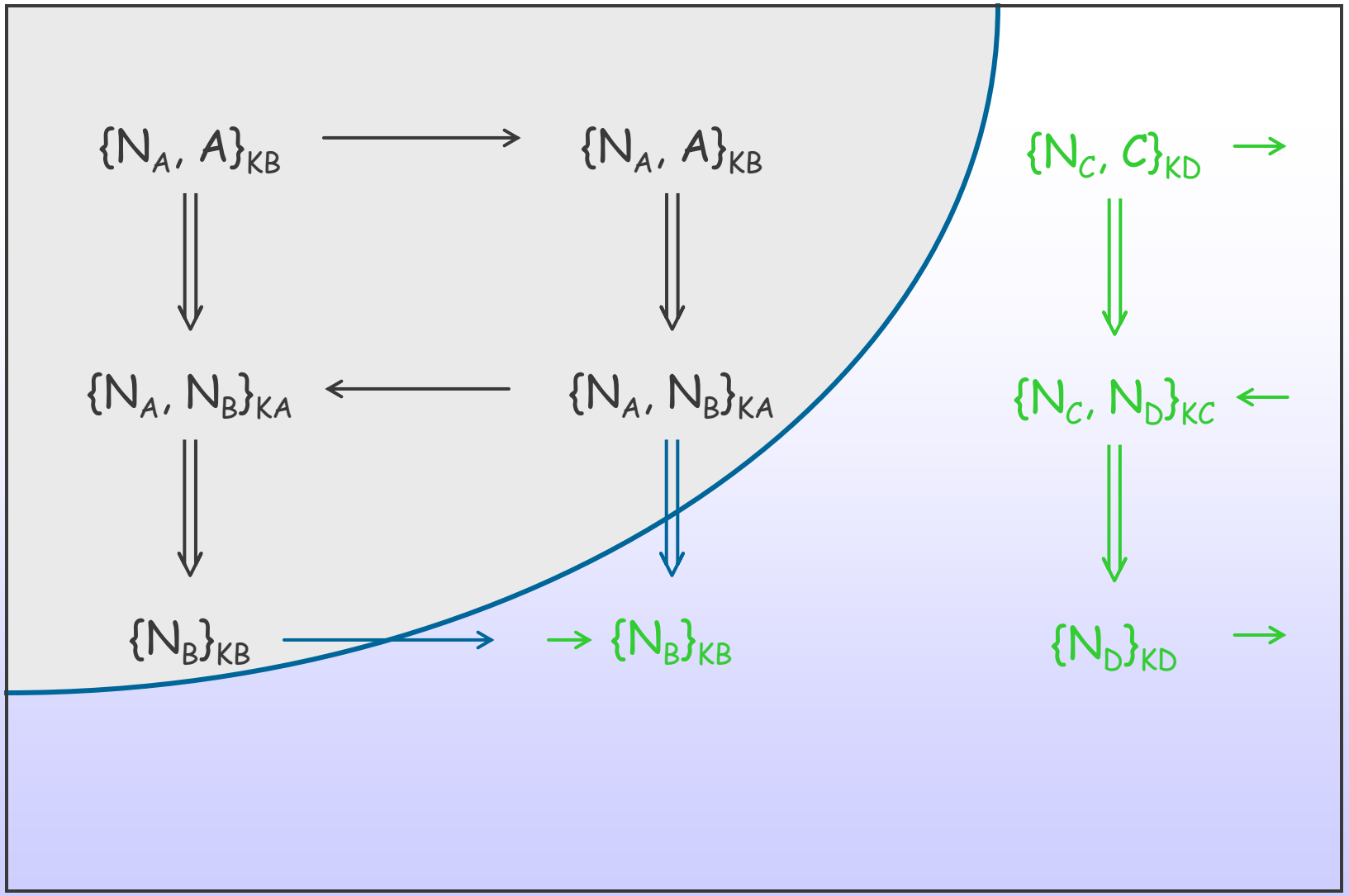
Configurations

? Capture possible next actions

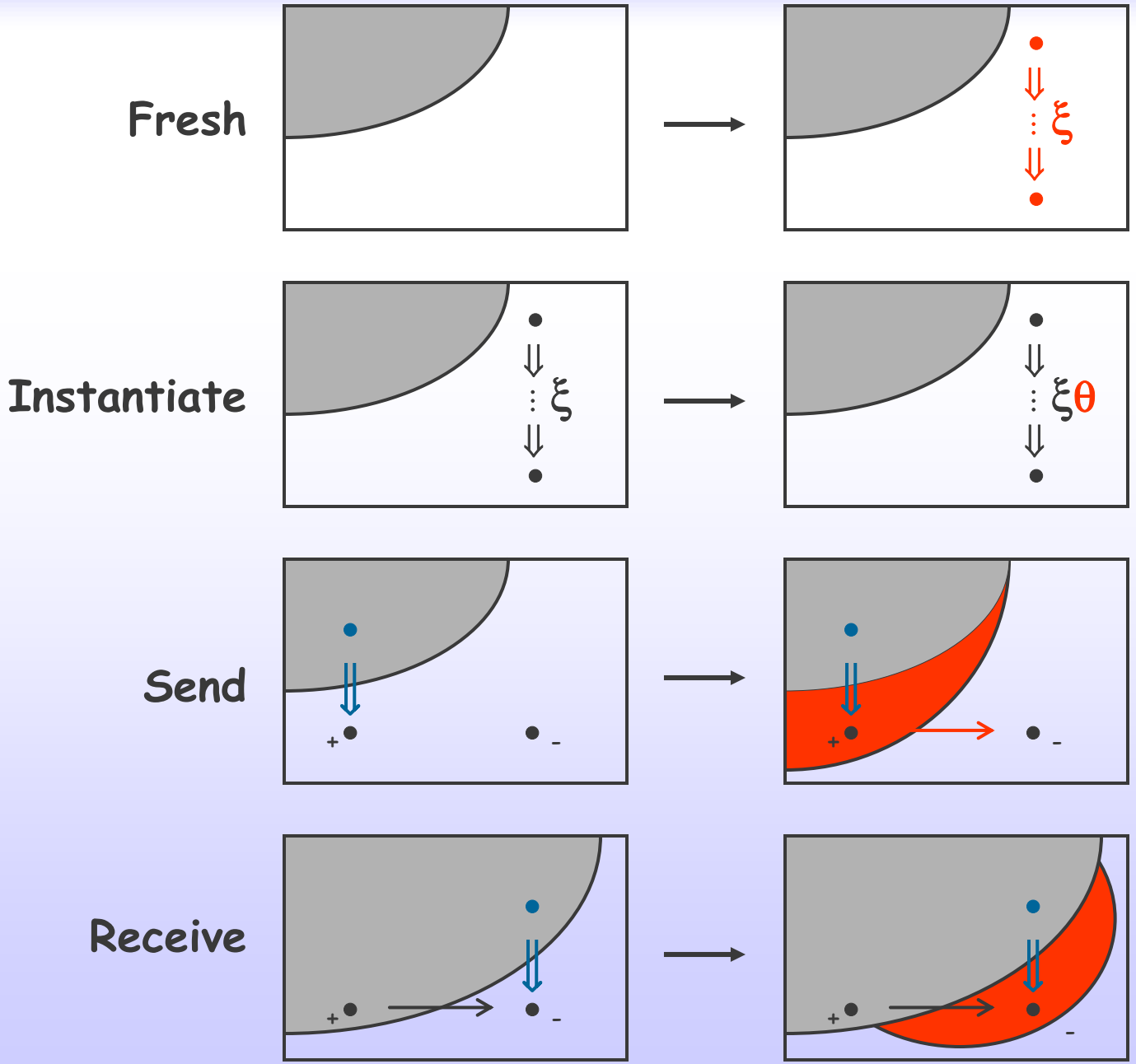
- Extension : bundle + remaining actions
- Configuration : bundle + extension
- Fringe : crossing arrows



NS: Configuration

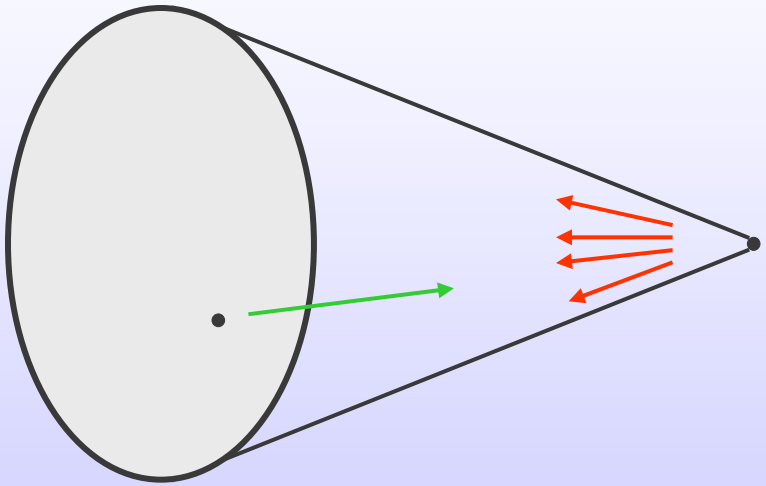


Strand Transitions



Bundles vs. Transition Sequences

- 1 bundle ⇒ O(n!) transition sequences
- 1 transition sequence ⇒ 1 bundle



- Bundles represent execution more compactly

Restr. MSR ⟶ Can. MSR

- Merge role gen. with 1st exec. rule
- Choose nonces upfront
- Guess persistent info. upfront

Conversion to canonical form preserves reachability

NS: Canonical MSR rules for Alice

$$\pi_A(A,B) \rightarrow \exists N_A.\ A_1(A,B,N_A),\ N(\{N_A,A\}_{KB}),\ \pi_A(A,B)$$

$$A_1(A,B,N_A),\ N(\{N_A,N_B\}_{KA}) \rightarrow A_2(A,B,N_A,N_B)$$

$$A_2(A,B,N_A,N_B) \rightarrow A_3(A,B,N_A,N_B),\ N(\{N_B\}_{KB})$$

where $\pi_A(A,B) = Pr(A),\ PrvK(A,K_A^{-1})$
$Pr(B),\ PubK(B,K_B)$

Can. MSR ⟶ Dyn. Strands

- Rules ⇒ nodes
- Role state predicates ⇒ arrows
- Nonces, persistent info. ⇒ constraints

- Configuration ⇐ state

Reachable states

⇓

Reachable configurations

NS: MSR $\xrightarrow{(1)}$ Strands

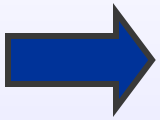
$\pi_A(A,B)$

$\rightarrow \exists N_A.$

$A_1(A,B,N_A),$

$N(\{N_A,A\}_{KB}),$

$\pi_A(A,B)$



Alice (A,B,N_A,N_B) :

N_A Fresh, $\pi_A(A,B)$

$\{N_A, A\}_{KB} \longrightarrow$

where $\pi(A,B) = Pr(A), PrvK(A,K_A^{-1}), Pr(B), PubK(B,K_B)$

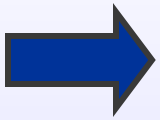
NS: MSR $\xrightarrow[(2)]{}$ Strands

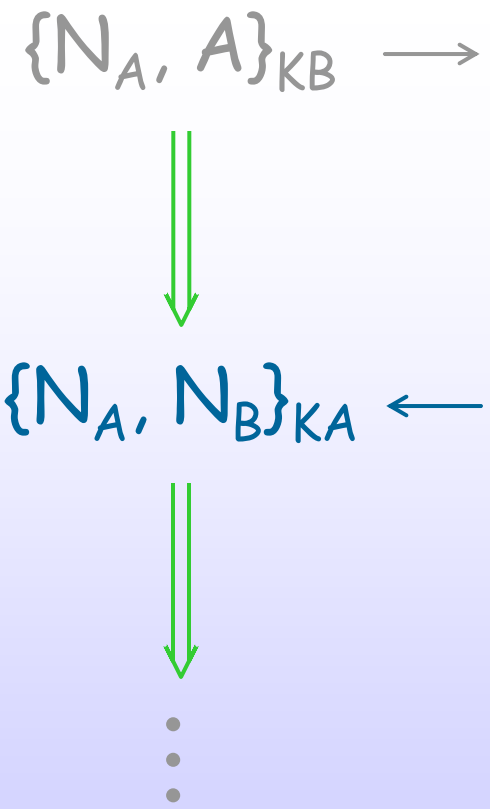
$A_1(A,B,N_A)$,

$N(\{N_A,N_B\}_{KA})$

$\rightarrow$

$A_2(A,B,N_A,N_B)$





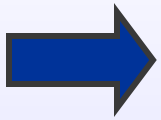
NS: MSR $\xrightarrow{(3)}$ Strands

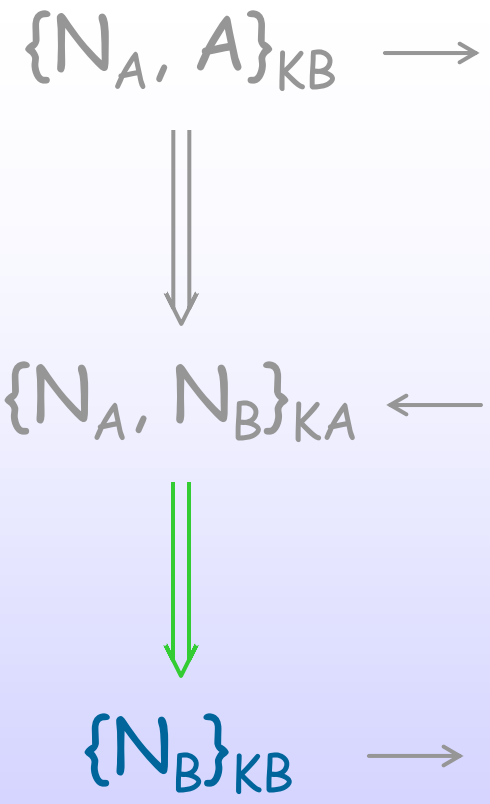
$A_2(A,B,N_A,N_B)$

$\rightarrow$

$A_3(A,B,N_A,N_B)$,

$N(\{N_B\}_{KB})$



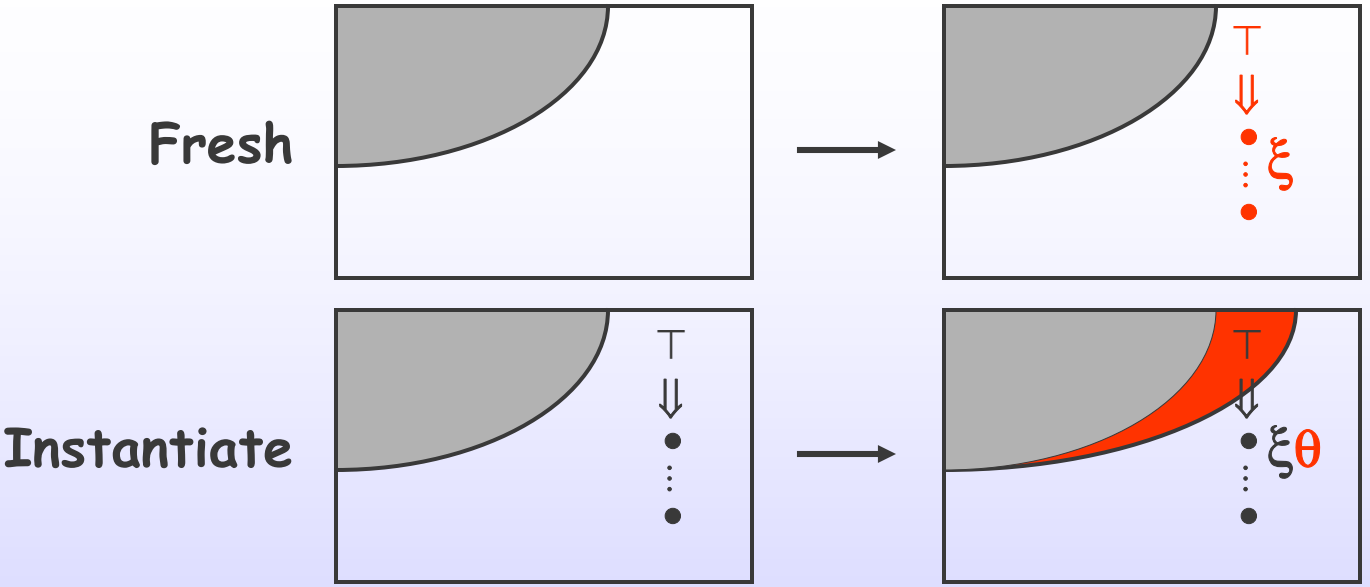


Dyn. Strands $\xrightarrow{(1)}$ Dec. Strands

- Add initial ($\top$) and final node ($\bot$)
- Add labels $A_i(t_1,\ldots,t_{ni})$ to arrows $t_1,\ldots,t_{ni}$ from
  - Constraints
  - Arguments of A_{i-1}

Dyn. Strands → (2) Dec. Strands

- Transitions

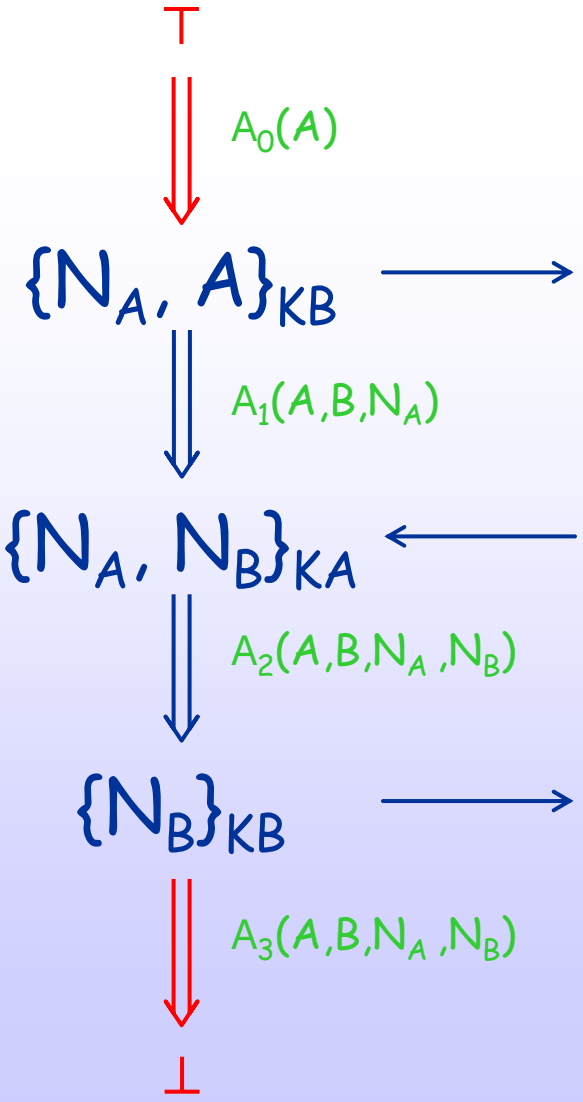


Decoration preserves reachability

NS: Decorated Strand for *Alice*

Alice (A,B,N_A,N_B) :

N_A Fresh, $\pi_A(A,B)$



where

$\pi(A,B) = Pr(A), PrvK(A,K_A^{-1}),$
$Pr(B), PubK(B,K_B)$

Dec. Strands ⟶ Restr. MSR

- Labels ⇒ role state predicates
- Events ⇒ network messages
- Constraints ⇒ nonces, persistent info.

- State ⇐ fringe

Reachable configurations

⇓

Reachable states

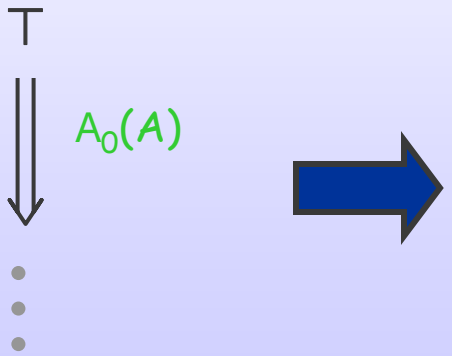
NS: Strands $\xrightarrow{(1)}$ MSR

Alice (A, B, N_A, N_B)

N_A Fresh, $\pi_A(A,B)$

where

$\pi(A,B) =$ $Pr(A)$, $PrvK(A, K_A^{-1})$, $Pr(B)$, $PubK(B, K_B)$



$\pi_{A0}(A) \rightarrow A_0(A), \pi_{A0}(A)$

where $\pi_{A0}(A) = Pr(A), PrvK(A, K_A^{-1})$

NS: Strands $\xrightarrow{(2)}$ MSR

Alice (A,B,N_A,N_B)

N_A Fresh, $\pi_A(A,B)$

where

$\pi(A,B) =$ $Pr(A)$, $PrvK(A,K_A^{-1})$, $Pr(B)$, $PubK(B,K_B)$

$A_0(A)$

$\{N_A, A\}_{KB} \rightarrow$

$A_1(A,B,N_A)$

$A_0(A), \pi_{A1}(B)$

$\rightarrow$

$\exists N_A.\ A_1(A,B,N_A),$
$N(\{N_A,A\}_{KB}), \pi_{A1}(B)$

where $\pi_{A1}(B) = Pr(B), PubK(B,K_B)$

NS: Strands $\xrightarrow{(3)}$ MSR

Alice (A,B,N_A,N_B)

N_A Fresh, $\pi_A(A,B)$

where

$\pi(A,B) =$ $Pr(A)$, $PrvK(A,K_A^{-1})$, $Pr(B)$, $PubK(B,K_B)$

$A_1(A,B,N_A)$

$\{N_A, N_B\}_{KA}$ ←

$A_2(A,B,N_A,N_B)$

$A_1(A,B,N_A)$, $N(\{N_A,N_B\}_{KA})$

→

$A_2(A,B,N_A,N_B)$

What did we learn?

- Substantial equivalence of
  - MSR
  - Strands
- Strands as executable spec. language
  - Parametric strands
  - Configurations, transitions
- Computational traces
  - Bundles
  - Transition sequences