

# MSR 3.0:

## The Logical Meeting Point of Multiset Rewriting and Process Algebra



*Iliano Cervesato*

*iliano@itd.nrl.navy.mil*

*ITT Industries, inc @ NRL Washington, DC*

*<http://www.cs.stanford.edu/~iliano>*

... 1<sup>st</sup> slide of my CSFW'00 talk ...



### Representing Security Protocols

Several recent proposal based on the Dolev-Yao model:

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

How are they related?

Relating Strands and Multiset Rewriting for Security Protocols

- Since then
  - MSR  $\Leftrightarrow$  linear logic  $\Leftrightarrow$  strands
  - MSR 2.0
  - MSR  $\Leftrightarrow$  process algebra

# MSR vs. PA

## MSR

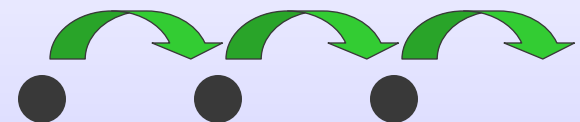
- NRL Prot. Analyzer, CAPSL/CIL, Paulson's approach, ...

## and Process Algebra

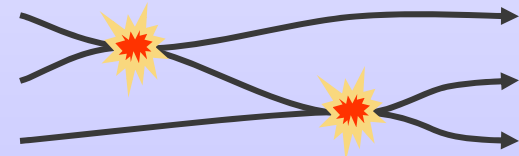
- Strand spaces, spi-calculus, other process-based lang.

operate in very different ways:

- State transitions



- Contact evolution



# Representing Protocols

- MSR  $\left[ \begin{array}{l} n \rightarrow a_1, n' \\ n'', a_1 \rightarrow a_2, n''' \\ \dots \end{array} \right]$

- $a_i$  pass control/data to the next rule

- PA  $\underline{n}.n'.\underline{n''}.n''' \dots .0$

- Control is implicit

## NS: MSR rules for Alice

$$\begin{aligned} \pi_{A0}(A) &\rightarrow A_0(A), \pi_{A0}(A) \\ A_0(A), \pi_{A1}(B) &\rightarrow \exists N_A. A_1(A, B, N_A), N(\{N_A, A\}_{KB}), \pi_{A1}(B) \\ A_1(A, B, N_A), N(\{N_A, N_B\}_{KA}) &\rightarrow A_2(A, B, N_A, N_B) \\ A_2(A, B, N_A, N_B) &\rightarrow A_3(A, B, N_A, N_B), N(\{N_B\}_{KB}) \end{aligned}$$

where  $\pi_{A0}(A) = Pr(A), PrvK(A, K_A^{-1})$   
 $\pi_{A1}(B) = Pr(B), PubK(B, K_B)$

Relating Strands and Multiset Rewriting for Security Protocols

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

Relating Strands and Multiset Rewriting for Security Protocols

# During Translation

- MSR  $\rightarrow$  PA
  - Use  $a_i$  to piece process together
  - Besides that, very easy
- PA  $\rightarrow$  MSR
  - Synthesize  $a_i$ 
    - Not trivial for parameters
  - Come up with state



# What Makes Encoding Hard?

Two activities

- Move between formalisms
- Move between paradigms

Analogy: translate Lisp to C

- Turn S-Expressions to structures
- Turn recursion into iteration

... but C supports recursion ...





# Extending MSR

Idea: devise an extension of MSR that brings it closer to PA

## Benefits

- Simplifies translation (a lot)
- Internalizes paradigm shift
  - Independent from target formalism
  - Easier to understand
  - In-house optimizations

# MSR 3.0

- ... or higher-order MSR

- $\omega$ -multisets

$$w ::= . \mid a, w \mid w \rightarrow w$$

- Computation


$$\frac{}{u, v, (u \rightarrow w) \rightarrow v, w}$$






# PA to MSR 1

- PA to MSR 3

 a.b.c.d.0  
 $a \rightarrow b, (c \rightarrow d)$

- MSR 3 to MSR 1

  $a \rightarrow b, x$   
 $x, c \rightarrow d$

- Done completely within MSR
- Done once and for all
- Opportunity for optimization (FO setting)
  - Study of memory denial-of-service



# MSR 3 to PA

- Easy but not as trivial
- Care is required
  - If we want a somewhat invertible translation





# The Rest of the Story

... but there is more to PA

- $\parallel, !, +, v, \dots$

- There is more to MSR 3

- MSR 3 is linear logic in disguise

- ... more radically so than MSR 1