MSR 3.0:
The Logical Meeting Point of Multiset
Rewriting and Process Algebra

Iliano Cervesato
ilitano@itd.nrl.navy.mil

ITT Industries, inc @ NRL Washington, DC

http://www.cs.stanford.edu/~iliano

CS Department, UMBC
February 27-28, 2003
Representing Security Protocols

Several recent proposal based on the Dolev-Yao model:
- Strand spaces
- Multiset rewriting
- Spi-calculus, ...

How are they related?

Since then
- MSR ⇔ linear logic ⇔ strands
- MSR 2.0
- MSR ⇔ 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**

  \[
  \begin{align*}
  n \rightarrow & a_1, n' \\
  n'', a_1 \rightarrow & a_2, n'' \\
  & \ldots
  \end{align*}
  \]

  - \(a_i\) pass control/data to the next rule

- **PA**

  \[
  n.n'.n''.n'''. \ldots .0
  \]

  - Control is implicit

**NS: MSR rules for Alice**

\[
\begin{align*}
\pi_{A_0}(A) & \rightarrow A_1(A), \pi_{A_0}(A) \\
A_0(A), \pi_{A_1}(B) & \rightarrow \exists N_A. A_0(A,B,N_A), N((N_A,A)_{KB}), \pi_{A_1}(B) \\
A_0(A,B,N_A), N((N_A,N_A)_{KA}) & \rightarrow A_1(A,B,N_A,N_B) \\
A_0(A,B,N_A,N_B) & \rightarrow A_2(A,B,N_A,N_B,N_B), N((N_B)_{KB})
\end{align*}
\]

where

\[
\begin{align*}
\pi_{A_0}(A) &= Pr(A), PrvK(A,K_A^{-1}) \\
\pi_{A_1}(B) &= Pr(B), PubK(B,K_B)
\end{align*}
\]

**NS: Parametric Strand for Alice**

Alice \((A,B,N_A,N_B)\):

\[
\begin{align*}
N_A & \text{ Fresh}, \pi_{A_1}(A,B) \\
\{N_A, A\}_{KB} & \rightarrow \\
\{N_A, N_B\}_{KA} & \leftarrow \\
\{N_B\}_{KB} & \rightarrow
\end{align*}
\]

where

\[
\begin{align*}
\pi(A,B) &= Pr(A), PrvK(A,K_A^{-1}) \\
\pi(B) &= Pr(B), PubK(B,K_B)
\end{align*}
\]

**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
• ... or higher-order MSR

• $\omega$-multisets

\[
\begin{align*}
w & ::= . \mid a,w \mid w \rightarrow w
\end{align*}
\]

• Computation

\[
\begin{align*}
u,v,(u \rightarrow w) & \rightarrow v,w
\end{align*}
\]
PA to MSR 1

- PA to MSR 3
  - a.b.c.d.0
  - a → b, (c → d)

- MSR 3 to MSR 1
  - 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
  - \(||, !, +, \lor, \ldots|

• There is more to MSR 3
  - MSR 3 is linear logic in disguise
  - ... more radically so than MSR 1