Ensuring Security through Model Checking in a Logical Environment

Marco Bozzano

DISI, University of Genova, Italy

(Presentation by Giorgio Delzanno)

A Multiset Rewriting Specification Logic

- A network of communicating processes, with internal state
- *Intra-object* communication
- *Inter-object* communication by *rendez-vous*
- *Locality* of rewriting
- *Parametric* and *open* specifications
- *Universal quantification* to model *fresh* name generation [Cervesato et al.'99]

Summary of the Talk

- A presentation of the logics $\mathrm{MSR}_\forall$,
  multiset rewriting with universal quantification
- An assertional language for security properties
- *Parameterized* verification
- A *backward* verification procedure
- Case study: the Needham-Schroeder authentication protocol

The Needham-Schroeder Protocol

A public-key authentication protocol

$$
\begin{array}{lcl}
A \rightarrow S & : & A, B \\
S \rightarrow A & : & \{K_b, B\}_{K_s^{-1}} \\
\mathbf{A} \rightarrow \mathbf{B} & : & \{\mathbf{N_a}, \mathbf{A}\}_{\mathbf{K_b}} \\
B \rightarrow S & : & B, A \\
S \rightarrow B & : & \{K_a, A\}_{K_s^{-1}} \\
\mathbf{B} \rightarrow \mathbf{A} & : & \{\mathbf{N_a}, \mathbf{N_b}\}_{\mathbf{K_a}} \\
\mathbf{A} \rightarrow \mathbf{B} & : & \{\mathbf{N_b}\}_{\mathbf{K_b}}
\end{array}
$$

The logic $\mathrm{MSR}_\forall$ - Atomic Formulas

Atomic formulas:

$$\mathtt{p}(\mathtt{t}_1, \ldots, \mathtt{t}_\mathtt{k})$$

where $\mathtt{p}$ is a predicate symbol and $\mathtt{t}_1, \ldots, \mathtt{t}_\mathtt{k}$ are terms

Principals:	`pr(identifier,internal_state)`
Messages:	`n(pubk(identifier),message_content)`
Intruder Memory:	`m(information)`

Example:

$$\mathtt{pr}(\mathtt{alice}, \mathtt{init})$$

$$\mathtt{n}(\mathtt{pubk}(\mathtt{bob}), \mathtt{mess}(\mathtt{na}, \mathtt{alice}))$$

$$\mathtt{m}(\mathtt{alice})$$

The logic $\mathrm{MSR}_\forall$ - Configurations

Configuration = multiset of ground atomic formulas (principals + messages + intruder memory)

Example

$$\mathtt{pr}(\mathtt{alice}, \mathtt{init}) \mid \mathtt{pr}(\mathtt{bob}, \mathtt{init})$$

$$\mathtt{pr}(\mathtt{alice}, \mathtt{step1}(\mathtt{na}, \mathtt{bob})) \mid \mathtt{pr}(\mathtt{bob}, \mathtt{init}) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{bob}), \mathtt{mess}(\mathtt{na}, \mathtt{alice}))$$

The logic $\mathrm{MSR}_\forall$ - Rules

Rules:

$$\underbrace{\mathtt{M}_1 \mid \ldots \mid \mathtt{M}_n}_{multiset} \longrightarrow \underbrace{\forall \mathtt{X}_1 \ldots \mathtt{X}_k.}_{quantif.variables} \quad \underbrace{(\mathtt{M}'_1 \mid \ldots \mid \mathtt{M}'_m)}_{multiset}$$

where $\mathtt{M}_\mathtt{i}$ are (non-ground) atomic formulas

Universal quantification used to generate *fresh values*

Example: *nonce* generation in security protocols

$$\underbrace{\mathtt{A}_1 \mid \ldots \mid \mathtt{A}_n}_{old\ princ.\ \&\ mess.} \longrightarrow \underbrace{\forall \mathtt{N}.}_{new\ nonce} \quad \underbrace{(\mathtt{B}_1 \mid \ldots \mid \mathtt{B}_m)}_{new\ princ.\ \&\ mess.}$$

The logic $\mathrm{MSR}_\forall$ - Rewriting

Rule:

$$\mathtt{pr}(\mathtt{A},\mathtt{init}) \mid \mathtt{pr}(\mathtt{B},\mathtt{init}) \longrightarrow \forall \mathtt{NA}.\ (\mathtt{pr}(\mathtt{B},\mathtt{init}) \mid \mathtt{pr}(\mathtt{A},\mathtt{step1}(\mathtt{NA},\mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}),\mathtt{mess}(\mathtt{NA},\mathtt{A})))$$

Rule Instance:

$$\mathtt{pr}(\mathtt{alice},\mathtt{init}) \mid \mathtt{pr}(\mathtt{bob},\mathtt{init}) \longrightarrow (\mathtt{pr}(\mathtt{bob},\mathtt{init}) \mid \mathtt{pr}(\mathtt{alice},\mathtt{step1}(\mathtt{na},\mathtt{bob})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{bob}),\mathtt{mess}(\mathtt{na},\mathtt{alice})))$$

where $\mathtt{na}$ is a newly generated *nonce*

One-step Rewriting:

$$\mathtt{pr}(\mathtt{alice},\mathtt{init}) \mid \mathtt{pr}(\mathtt{bob},\mathtt{init}) \Rightarrow \mathtt{pr}(\mathtt{bob},\mathtt{init}) \mid \mathtt{pr}(\mathtt{alice},\mathtt{step1}(\mathtt{na},\mathtt{bob})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{bob}),\mathtt{mess}(\mathtt{na},\mathtt{alice}))$$

Encoding Needham-Schroeder

p_1) $\mathtt{pr}(\mathtt{A}, \mathtt{init}) \mid \mathtt{pr}(\mathtt{B}, \mathtt{init}) \longrightarrow$

$\mathtt{pr}(\mathtt{B}, \mathtt{init}) \mid \forall \mathtt{NA}.\ (\mathtt{pr}(\mathtt{A}, \mathtt{step1}(\mathtt{NA}, \mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NA}, \mathtt{A})))$

p_2) $\mathtt{pr}(\mathtt{B}, \mathtt{init}) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NA}, \mathtt{A})) \longrightarrow$

$\forall \mathtt{NB}.\ (\mathtt{pr}(\mathtt{B}, \mathtt{step2}(\mathtt{NA}, \mathtt{NB}, \mathtt{A})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{A}), \mathtt{mess}(\mathtt{NA}, \mathtt{NB})))$

p_3) $\mathtt{pr}(\mathtt{A}, \mathtt{step1}(\mathtt{NA}, \mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{A}), \mathtt{mess}(\mathtt{NA}, \mathtt{NB})) \longrightarrow$

$\mathtt{pr}(\mathtt{A}, \mathtt{step3}(\mathtt{NA}, \mathtt{NB}, \mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NB}))$

p_4) $\mathtt{pr}(\mathtt{B}, \mathtt{step2}(\mathtt{NA}, \mathtt{NB}, \mathtt{A})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NB})) \longrightarrow$

$\mathtt{pr}(\mathtt{B}, \mathtt{step4}(\mathtt{NA}, \mathtt{NB}, \mathtt{A}))$

r_1) $\epsilon \longrightarrow \forall \mathtt{ID}.\ (\mathtt{pr}(\mathtt{ID}, \mathtt{init}))$ *(principal creation)*

r_2) $\mathtt{pr}(\mathtt{Z}, \mathtt{S}) \longrightarrow \mathtt{pr}(\mathtt{Z}, \mathtt{S}) \mid \mathtt{pr}(\mathtt{Z}, \mathtt{init})$ *(role creation)*

Why Backward Verification?

Initial States: $Init$

Unsafe States: $Unsafe$

Safety: $\begin{cases} Post^*(Init) \cap Unsafe = \emptyset & \text{(\textbf{forward} approach)} \\ \boxed{Pre^*(Unsafe) \cap Init = \emptyset} & \text{(\textbf{backward} approach)} \end{cases}$

Backward Evaluation:

- $Unsafe$ described by means of **minimality violations**
- Symbolic representation for $Unsafe$
- An assertional language for security violations

An Assertional Language

Unsafe States = Configurations:

$$\mathtt{p_1}(\mathtt{t_{11}}, \ldots, \mathtt{t_{1k_1}}) \mid \ldots \mid \mathtt{p_n}(\mathtt{t_{n1}}, \ldots, \mathtt{t_{nk_n}})$$

Example: Needham-Schroeder

$$\mathbf{U} \equiv \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NA})$$

Denotation of Constrained Configurations:

$$[\![S]\!] = Up\,(Inst\,(S))$$

$$[\![\mathbf{U}]\!] = \{\ \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{na}, \mathtt{nb}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{na}) \mid \mathtt{M_1} \mid \ \ldots \ \mid \mathtt{M_j}$$
$$\text{s.t. } \mathtt{M_1}, \ldots, \mathtt{M_j} \text{ are ground atomic formulas } \}$$

Automated Verification

Init: symbolic representation of initial states

Unsafe: symbolic representation of unsafe states

Pre: symbolic predecessor operator (based on multiset unification)

$\sqsubseteq_m$**:** containment test for sets of symbolic configurations
(used to cut redundant multisets and to test fixpoint)

Procedure Symbolic_Verification:

$$
\begin{array}{l}
\textbf{input} : \textbf{Init},\ \textbf{Unsafe} \\
Old\ :=\ \emptyset;\ \ New\ :=\ \textbf{Unsafe}; \\
\quad \textbf{repeat} \\
\quad\quad Old\ :=\ Old\ \cup\ New; \\
\quad\quad New\ :=\ \textbf{Pre}\,(New); \\
\quad \textbf{until}\ Old\ \sqsubseteq_m New; \\
\textbf{check}\ \text{if}\ [\![Old]\!] \cap [\![\textbf{Init}]\!] = \emptyset
\end{array}
$$

The Dolev-Yao Intruder

An attacker can:

- arbitrarily **intercept** any message
- **block** message transmission
- **replay** any intercepted message at any later time
- **decompose** any message encrypted with a known key
- **store** decrypted message components
- randomly **compose** new messages from stored components
- **send** messages to any principals using their public keys
- store principal identifiers
- **generate** and store new values

Intruder Theory

$i_1)$ $\mathtt{n(pubk(A), M)} \longrightarrow \mathtt{intercept(n(pubk(A), M))}$ *(interception)*

$i_2)$ $\mathtt{intercept(n(pubk(A), M))} \longrightarrow$ " "

$\quad\quad \mathtt{intercept(n(pubk(A), M))} \mid$

$\quad\quad \mathtt{n(pubk(A), M)}$

$i_3)$ $\mathtt{n(pubk(intruder), M)} \longrightarrow \mathtt{dec(M)}$ *(decomposition)*

$i_4)$ $\mathtt{dec(mess(M))} \longrightarrow \mathtt{m(M)}$ " "

$i_5)$ $\mathtt{dec(mess(M, N))} \longrightarrow \mathtt{m(M)} \mid \mathtt{m(N)}$ " "

$i_6)$ $\mathtt{m(M)} \longrightarrow \mathtt{m(M)} \mid \mathtt{comp(mess(M))}$ *(composition)*

$i_7)$ $\mathtt{m(M)} \mid \mathtt{m(N)} \longrightarrow \mathtt{m(M)} \mid \mathtt{m(N)} \mid \mathtt{comp(mess(M, N))}$ " "

$i_8)$ $\mathtt{comp(C)} \longrightarrow \mathtt{n(pubk(Z), C)}$ *(send)*

$i_9)$ $\mathtt{pr(Z, S)} \longrightarrow \mathtt{pr(Z, S)} \mid \mathtt{m(Z)}$ *(id. memoization)*

$i_{10})$ $\epsilon \longrightarrow \forall \mathtt{N}.\ (\mathtt{m(N)})$ *(value generation)*

Verification of Needham-Schroeder

Strong Correctness:

$$\begin{array}{rcl} \mathbf{S} & = & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NA}) \\ & & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NB}) \\ & & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NA}) \\ & & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NB}) \end{array}$$

Attack (Lowe'95) (NS Protocol is **not** safe with respect to **S**):

$$\begin{array}{lllll} 1. & alice \rightarrow intruder & : & \{N_a, alice\}_{K_{intruder}} \\ 2. & intruder \rightarrow bob & : & \{N_a, alice\}_{K_{bob}} \\ 3. & bob \rightarrow alice & : & \{N_a, N_b\}_{K_{alice}} \\ 4. & alice \rightarrow intruder & : & \{N_b\}_{K_{intruder}} \\ 5. & intruder \rightarrow bob & : & \{N_b\}_{K_{bob}} \end{array}$$

Verification of Needham-Schroeder

Weak Correctness:

$$\begin{aligned}\mathbf{W} \quad = \quad & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \\ & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NA}) \\ & \\ & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \\ & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NB})\end{aligned}$$

Invariant strengthening: (for efficiency)

$$\mathbf{W} \quad + \quad \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step1}, \mathtt{NA}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NA})$$

Correctness: NS is safe with respect to $\mathbf{W}$

Needham-Schroeder Protocol (Corrected)

$$\begin{array}{rcl} A \rightarrow B & : & \{N_a, A\}_{K_b} \\ B \rightarrow A & : & \{\boxed{\mathrm{B}}, N_a, N_b\}_{K_a} \\ A \rightarrow B & : & \{N_b\}_{K_b} \end{array}$$

$p_2)$ $\mathtt{pr}(\mathtt{B}, \mathtt{init}) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NA}, \mathtt{A})) \longrightarrow$

$\forall \mathtt{NB}.\ (\mathtt{pr}(\mathtt{B}, \mathtt{step2}(\mathtt{NA}, \mathtt{NB}, \mathtt{A})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{A}), \mathtt{mess}(\boxed{\mathrm{B}}, \mathtt{NA}, \mathtt{NB})))$

$p_3)$ $\mathtt{pr}(\mathtt{A}, \mathtt{step1}(\mathtt{NA}, \mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{A}), \mathtt{mess}(\boxed{\mathrm{B}}, \mathtt{NA}, \mathtt{NB})) \longrightarrow$

$\mathtt{pr}(\mathtt{A}, \mathtt{step3}(\mathtt{NA}, \mathtt{NB}, \mathtt{B})) \mid \mathtt{n}(\mathtt{pubk}(\mathtt{B}), \mathtt{mess}(\mathtt{NB}))$

Verification of (Corrected) Needham-Schroeder

Strong Correctness:

$$\begin{aligned}\mathbf{S} \quad = \quad & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NA}) \\ & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step3}, \mathtt{NA}, \mathtt{NB}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NB}) \\ & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NA}) \\ & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step4}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NB})\end{aligned}$$

Invariant strengthening: (for efficiency)

$$\begin{aligned}\mathbf{S} \quad + \quad & \mathtt{pr}(\mathtt{alice}, \mathtt{s}(\mathtt{step1}, \mathtt{NA}, \mathtt{bob})) \mid \mathtt{m}(\mathtt{NA}) \\ & \mathtt{pr}(\mathtt{bob}, \mathtt{s}(\mathtt{step2}, \mathtt{NA}, \mathtt{NB}, \mathtt{alice})) \mid \mathtt{m}(\mathtt{NB})\end{aligned}$$

Correctness: Corrected NS is safe with respect to **S**

Experimental Results

	Without invariant strength.	With invariant strength.
Incorrect NS (strong correctness)	`ATTACK (60)`	–
Incorrect NS (weak correctness)	`CORRECT (600)`	`CORRECT (70)`
Corrected NS (strong correctness)	`CORRECT (1140)`	`CORRECT (35)`

(Execution time, in seconds, on a Pentium III 450MhZ,
Linux 2.2.13-0.9, SML of NJ V.110.0.7)

Future Work on Security

Protocol Translation from usual notation (possibly automatic)

Theory Optimization: specialized intruder theory (*folding/unfolding*)

Invariant Strengthening: support for automatic invariant generation

Constraint solver to manage protocols with *timestamps*

An Optimized Implementation

Typed Multiset Rewriting

Termination Issues