Complete Symbolic Bisimilarity for an Extended Spi Calculus

Raphael Reischuk

based on work of
Johannes Borgström et al.

January 23, 2009
Outline

1. Introduction
   - The Big Picture
   - Notation

2. Symbolic Semantics
   - Introduction
   - Symbolic Operational Semantics
   - Problems & Solutions

3. Symbolic Bisimilarity
   - Constraint System
   - Hedges
want to formalize and prove correctness properties for cryptographic protocols

using extended spi calculus [AG97] with equality = and Boolean guards ¬, ∧, ∨, arbitrary constructors, and destructors

need equivalence between certain processes

- secrecy: \( \forall m, n : P(m) \approx P(n) \)
- authenticity: \( \forall m : P(m) \approx P_{spec}(m) \)
- vote privacy: 
  \[
  \forall a, b, v_1, v_2 : P(a, v_1) \mid P(b, v_2) \approx P(a, v_2) \mid P(b, v_1)
  \]

where \( \approx \) denotes indistinguishability (in all contexts)

quantification over all contexts captures the intuition of an unknown attacker

\( \rightarrow \) contextual equivalence
proofs by hand are hard [AG97] (quantification over infinitely many process contexts)

labelled bisimilarity ($\pi$-calculus) is too strong as it distinguishes the processes $P(0)$ and $P(1)$ with

$$P(M) \overset{\text{def}}{=} (\nu K) \overline{c} \langle \text{enc}_K(M) \rangle$$

solution: take into account the knowledge of the environment

possibility to receive arbitrary messages yields infinite number of concrete transitions

solution: use symbolic semantics (no substitution of received messages for input variables) and successively add constraints
Notation

- names $N$
  - $a, b, c, k, l, m, n$
- variables $V$
  - $x, y, z$
- expressions $E$
  - $F ::= u \mid f(\tilde{F}) \mid g(\tilde{F})$ (f: constructor, g: destructor)
- messages $M$
  - $M ::= u \mid f(\tilde{F})$ ($\tilde{F}$: sequence of expressions)
- processes
  - $P ::= 0 \mid F(x).P \mid \overline{F}\langle F\rangle.P \mid (\nu a)P \mid \phi P \mid \ldots$
- constraints
  - $\phi ::= true \mid [F = F] \mid [F : N] \mid \phi \land \phi \mid \neg \phi$
Symbolic Semantics

• problem: infinite branching on input
  • consider process $P \overset{\text{def}}{=} c(x).P'$
  • could reduce via transition $P \xrightarrow{c(K)} P'$
  • could reduce via transition $P \xrightarrow{c(\text{enc}_K(m))} P'$
  • could reduce via transition $P \xrightarrow{c(\text{pair}(a,b))} P'$

• solution:
  • use symbolic semantics
  • successively add constraints

• this talk [Bor08]
  • symbolic semantics for spi calculus
  • corresponding symbolic bisimilarity
Symbolic Semantics

- idea: record necessary conditions for transitions (no checking)
- symbolic transition
  \[ P \xrightarrow{\mu_s, \phi} P' \]
  
  action \( \mu_s \in \{(\nu \tilde{c}) \tau, (\nu \tilde{c}) G(x), (\nu \tilde{c}) \overline{G} \langle F \rangle\} \), constraint \( \phi \)

- add requirement (e.g. that channels are names) to transition constraint

- goal: symbolic semantics should be complete and sound w.r.t. concrete one
Symbolic Semantics - Some Rules

- **Cin** \[ G(x).P \xrightarrow{a(x)} P \] if \( e(G) = a \)

- **Sin** \[ G(x).P \xrightarrow{G(x)} P \] \([G : N] \]

- **Sout** \[ \overline{G} \langle F \rangle .P \xrightarrow{\overline{G} \langle F \rangle} P \] \([G : N] \land [F : M] \]

- **Scom**

\[
\begin{align*}
P & \xrightarrow{(\nu \tilde{b}) \overline{G} \langle F \rangle} P' & Q & \xrightarrow{(\nu \tilde{c}) G'(x)} Q'
\end{align*}
\]

\[
\phi_P & \quad \phi_Q
\]

\[
P \mid Q \xrightarrow{(\nu \tilde{b} \tilde{c}) \tau} P' \mid Q' \{ F / x \}
\]

(where \( e(\cdot) \) denotes evaluation)
Symbolic Semantics - Some Rules (ctd)

- **Sres**

\[
P \xrightarrow{\mu_s} P' \quad \frac{\phi}{P \xrightarrow{\phi} (\nu a) P' } \quad \text{if } a \not\in (n(\mu_s) \cup n(\phi))
\]

- **Sguard**

\[
P \xrightarrow{\mu_s} P' \quad \frac{\phi}{P' \xrightarrow{\phi} (\phi' P \xrightarrow{\mu_s} P')}
\]

- **Sopen**

\[
P \xrightarrow{\mu_s} P' \quad \frac{\phi}{(\nu a) P \xrightarrow{\phi} (\nu a) P' } \quad \text{if } a \in (fn(\mu_s) \cup n(\phi)) \text{ and } a \not\in bn(\mu_s)
\]
Example 1

- $P := (\nu b) \overline{a} \langle \pi_1(a.b) \rangle . P'$ for some $P'$
- concrete: $P \xrightarrow{\overline{a}\langle a \rangle} (\nu b) P'$
- symbolic: $P \xrightarrow{(\nu b) \overline{a} \langle \pi_1(a.b) \rangle} [a:N] \land [\pi_1(a.b):M] P'$ new: destructors succeed
- problem: processes differ in restriction of names!
Transition Relation $\rightarrow_s$

- **solution**: add restrictions back to resulting process

\[
\frac{P \xrightarrow{(\nu b) F(x)} P'}{\phi}
\]

- **CDin**

\[
\frac{P \xrightarrow{(\nu b) F(x)} s (\nu \tilde{b}) P'}{\phi}
\]

if $\{\tilde{b}\} \cap en(e_a(F)) = \emptyset$

- **CDout**

\[
\frac{P \xrightarrow{(\nu b) \overline{F} \langle G \rangle} P'}{\phi}
\]

as above with $\{\tilde{c}\} = \{\tilde{b}\} \setminus en(e_a(G))$

- **example again**: $P \xrightarrow{(\nu b) \overline{a} \langle \pi_1(a.b) \rangle} (\nu b) P'$

$[a : N] \land [\pi_1(a.b) : M]$

extruded names: $en(a) = \{a\}$  $en(x) = \emptyset$  $en(\overline{g}(\tilde{G})) = \emptyset$  $en(f(\tilde{G})) = \bigcup_i en(G_i)$

abstract evaluation: $e_a(\cdot)$ extends evaluation $e(\cdot)$ to the entire set of expressions
symbolic semantics allow for the communication of non-message terms:

- $Q := \overline{a}\langle\pi_1(x)\rangle | a(y).\overline{a}\langle y \rangle$
- symbolic: $Q \xrightarrow{\tau} 0 | \overline{a}\langle\pi_1(x)\rangle$
- constraint $\phi$ may not be fulfilled
- substitution $\rho := \{(a.a)/x\}$ enables a transition
- concrete: $Q \xrightarrow{\tau} 0 | \overline{a}\langle a \rangle$
- symbolic: $Q \xrightarrow{\tau}_{\phi \rho} 0 | \overline{a}\langle\pi_1(a.a)\rangle$
- problem: processes differ!
solution: define relation accordingly
Let $\succ_a$ be a relation on expressions, guards and processes satisfying

- $F \succ_a M$ whenever $e(F) = M$
- $(\nu a)(\nu b)P \succ_a (\nu b)(\nu a)P$
- $(\nu a)(P \mid Q) \succ_a ((\nu a)P) \mid Q$ if $a \notin fn(Q)$
- $(\nu a)(P \mid Q) \succ_a P \mid ((\nu a)Q)$ if $a \notin fn(P)$

Example:
- $(0 \mid \bar{a}\langle\pi_1((a.a))\rangle) \succ_a (0 \mid \bar{a}\langle a\rangle)$

The relation $\succ_a$ is a labelled bisimulation.
Symbolic Semantics is Complete and Sound

- Transition relation $\rightarrow_s$ yields a symbolic operational semantics that is complete and sound w.r.t. the concrete one (modulo $\succ_a$).
- Finitely branching (concrete one is infinitely branching).
- problem: infinite number of solutions
- idea: find a finite set of representative solutions
- using decompositions of symbolic environments (usually small)
Hedges

- Hedges are similar to frame-theory pairs (first talk [Mar98])
- Part of symbolic environments
  - **Hedge** \( h \subseteq E \times E \)
    represents the knowledge of an attacker as pairs of expressions considered equivalent
  - **Synthesis** \( S(h) \subseteq E \times E \)
    set of message pairs obtained by applying constructors to \( h \)
  - **Analysis** \( A(h) \subseteq E \times E \)
    set of message pairs obtained by applying destructors to \( h \)
- spi: need timing information
Main Result

without proof

- contextual equivalence
  - sound
  - sound

- strong hedged bisimilarity
  - complete sound
  - complete sound

- concrete semantics
  - complete sound

- symbolic semantics
  - complete sound

Raphael Reischuk
Complete Symbolic Bisimilarity for an Extended Spi Calculus
Thank you for the attention.

Thanks to Catalin and Matteo.


