Some Help for the Project

Ralf Sasse, Christoph Sprenger

Institute of Information Security
ETH Zurich

FMSEC Project Help, v.1
Apr 6, 2016

Guarded Formulas

All property formulas in Tamarin must be guarded.

Definition (Guarded formula)

A formula φ is guarded if all its quantified subformulas are of the forms:

$$\forall \overline{x}.\ F(\overline{z})@i \Rightarrow \psi \quad \exists \overline{x}.\ F(\overline{z})@i \wedge \psi \quad \text{(and special cases: } (\forall|\exists)\overline{x}.\ F(\overline{z})@i\text{)}$$

where F is a fact and $\overline{x}$ and $\overline{z}$ are vectors of variables such that $\overline{x} \subseteq \overline{z} \cup \{i\}$, i.e., all bound variables appear in the fact formula $F(\overline{z})@i$.

Guarded Formulas

All property formulas in Tamarin must be guarded.

Definition (Guarded formula)

A formula φ is guarded if all its quantified subformulas are of the forms:

$$\forall \overline{x}.\ F(\overline{z})@i \Rightarrow \psi \quad \exists \overline{x}.\ F(\overline{z})@i \wedge \psi \quad \text{(and special cases: } (\forall|\exists)\overline{x}.\ F(\overline{z})@i\text{)}$$

where F is a fact and $\overline{x}$ and $\overline{z}$ are vectors of variables such that $\overline{x} \subseteq \overline{z} \cup \{i\}$, i.e., all bound variables appear in the fact formula $F(\overline{z})@i$.

Example

Not guarded:

$$\exists Id\ i.\ Create(A, Id, 'I')@i \vee Create(B, Id, 'R')@i$$

Guarded equivalents:

$$(\exists Id\ i.\ Create(A, Id, 'I')@i \wedge T) \vee (\exists Id\ i.\ Create(B, Id, 'R')@i \wedge T)$$
$$(\exists Id\ i.\ Create(A, Id, 'I')@i) \vee (\exists Id\ i.\ Create(B, Id, 'R')@i)$$

Claim and Honesty Facts

Example (Honesty Facts in Security Properties)

Secrecy:

$$\begin{aligned}&\forall A\ M\ i.\ \mathit{Secret}(A, M)@i\\ &\quad\Rightarrow (\neg(\exists j. K(M)@j) \lor (\exists X\ j.\ \mathit{Rev}(X)@j \land \mathit{Honest}(X)@i))\end{aligned}$$

Non-injective agreement:

$$\begin{aligned}&\forall A\ B\ M\ i.\ \mathit{Commit}(A, B, \langle `I`, `R`, M\rangle)@i\\ &\quad\Rightarrow ((\exists j.\ \mathit{Running}(B, A, \langle `I`, `R`, M\rangle)@j)\\ &\qquad\lor(\exists X\ j.\ \mathit{Rev}(X)@j \land \mathit{Honest}(X)@i))\end{aligned}$$

Claim and Honesty Facts

Example (Honesty Facts in Security Properties)

Secrecy:

$$\forall A\ M\ i.\ \mathit{Secret}(A, M)@i$$
$$\Rightarrow (\neg(\exists j.K(M)@j) \lor (\exists X\ j.\ \mathit{Rev}(X)@j \land \mathit{Honest}(X)@i))$$

Non-injective agreement:

$$\forall A\ B\ M\ i.\ \mathit{Commit}(A, B, \langle `I`, `R`, M\rangle)@i$$
$$\Rightarrow ((\exists j.\ \mathit{Running}(B, A, \langle `I`, `R`, M\rangle)@j)$$
$$\lor(\exists X\ j.\ \mathit{Rev}(X)@j \land \mathit{Honest}(X)@i))$$

- The honesty facts *Honest*(X) label the same rule ($@i$) as the main claim fact (e.g., *Secret*, *Commit*).
- The properties hold (i.e., secrecy of M resp. existence of a *Running* fact) unless an agent that is expected to be honest is compromised in the trace.

Roles and Agents in Agreement

Example (Non-injective agreement of initiator with responder)

$$\begin{aligned} &\forall A\ B\ M\ i.\ \mathit{Commit}(A, B, \langle ‘I‘, ‘R‘, M\rangle)@i \\ &\quad \Rightarrow ((\exists j.\ \mathit{Running}(B, A, \langle ‘I‘, ‘R‘, M\rangle)@j) \\ &\quad\quad \vee(\exists X\ j.\ \mathit{Rev}(X)@j \wedge \mathit{Honest}(X)@i)) \end{aligned}$$

- Order of $‘I‘$ and $‘R‘$ fixed, meaning that the agent (A) in the initiator role agrees with the agent (B) in the responder role (on M).
- Order of agents A and B instantiating the initiator and responder roles is swapped.
- Idea is that the first agent name is the one “executing” the claim.

Executability Lemmas

- Executability lemmas are so-called existential properties.
- These show the existence of some protocol trace satisfying the formula ...
- ... instead of the usual case where all traces must satisfy the formula.

Example (Executabilty in Tamarin)

Insert the keyword **exists-trace** between the lemma name and the formula.

lemma executablility: **exists-trace**
"...(formula φ)..."

"There exists a trace that reaches the end of the protocol (expressed by φ)."

Syntax Issues: Type Annotations

- You must mark index variables with a hash ($\#$) in quantifications.
- This is not done on our slides to avoid notational clutter.

Example (Secrecy)

$$
\begin{aligned}
&\forall A\ M\ \#i.\ \mathit{Secret}(A, M)@i \\
&\quad \Rightarrow (\neg(\exists \#j. K(M)@j) \lor (\exists X\ \#j.\ \mathit{Rev}(X)@j \land \mathit{Honest}(X)@i))
\end{aligned}
$$

Syntax Issues: Type Annotations

- You must mark index variables with a hash (#) in quantifications.
- This is not done on our slides to avoid notational clutter.

Example (Secrecy)

$$\begin{aligned}&\forall A\; M\; \#i.\ \mathit{Secret}(A, M)@i\\ &\quad\Rightarrow (\neg(\exists \#j. K(M)@j) \vee (\exists X\; \#j.\ \mathit{Rev}(X)@j \wedge \mathit{Honest}(X)@i))\end{aligned}$$

In rewrite rules:

- You must mark all occurrences of a fresh name with a tilde (e.g., ~k) or no occurrence. A similar remark holds for agent names (e.g., $\$A$)
- A variable that occurs only on the right-hand side of a rule must be marked public, i.e., carry a \$ annotation (e.g. $\mathit{Fr}(sk) \rightarrow !\mathit{Ltk}(\$A, sk)$).
- Generally, you should not annotate elements of messages received in *In* facts with types as this would reduce the scope of the analysis.

Warning Messages

- No warnings are allowed in hand-in version!
- Warnings give good information what is wrong, e.g.:
  - ★ Mismatch of types: Use of $\$A$ and A in same rule
  - ★ Using one fact name with different arities
  - ★ Guardedness problem in formula

Warning Messages

- No warnings are allowed in hand-in version!
- Warnings give good information what is wrong, e.g.:
  - ★ Mismatch of types: Use of $\$A$ and A in same rule
  - ★ Using one fact name with different arities
  - ★ Guardedness problem in formula

Tamarin offers strict mode to stop such trouble early:

- Add command-line parameter: `--quit-on-warning`