ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Security Protocol Verification Using the Tamarin Prover

Michael Schläpfer
Institute of Information Security
May 22, 2013

Motivation

$$
\begin{array}{lcll}
I & \rightarrow & R & : \quad \{ni, I\}_{pk(R)} \\
I & \leftarrow & R & : \quad \{ni, nr\}_{pk(I)} \\
I & \rightarrow & R & : \quad \{nr\}_{pk(R)}
\end{array}
$$

Figure: NSPK.

Outline

Preliminaries

The Tamarin Prover

Case Studies

Conclusions

Linear Logic

- Logic of resources
- Resources consumable
- Persistent fact $!A$ vs. linear fact B

Example

Assume a coin will buy you one candy bar. We might express this as $coin \implies candy$. In classical logic, from A and $A \implies B$ we can conclude $A \wedge B$. The linear implication $coin \multimap candy$ allows to consume the *coin*.

Multisets

- A multiset (or bag) is a 2-tuple (S, m) where S is a set and $m : S \to \mathbb{N}_{\geq 1}$
- Usual operations on multisets are denoted by superscript $.^b$ (e.g., $\cup^b$)

Example

$\{a, a, a, b, b, c\}$ is a multiset over set $S = \{a, b, c\}$ with $m(a) = 3, m(b) = 2, m(c) = 1$.

Multiset Term Rewriting

- A rewriting theory R consists of rewriting rules $l \rightarrow r$
- The symbol $\rightarrow$ indicates that an expression matching the left side can be rewritten to the one of the right side
- Tamarin uses labeled multiset rewriting rules. A labeled multiset rewriting rule is a triple (l, a, r), denoted by $[l] - [a] -> [r]$

Examples

$\neg\neg A \rightarrow A$ represents a rule for double negative elimination in logic.
$[A, A, B] - [] -> [C, D, D, E]$ is a multiset rewriting rule in Tamarin syntax.

Outline

Preliminaries

The Tamarin Prover

Case Studies

Conclusions

Overview

- Security protocol model based on multiset term rewriting (protocol and adversary)
- User specific equational theories
- Specification language for security properties / goals
- Constraint solving algorithm for falsification and verification of security protocols
- Running implementation

Transition System States

- The states of the transition system are modeled as finite multisets of facts
- The adversary's knowledge, freshness information, and messages on the network are encoded using a fixed set of fact symbols
- Protocol states are modeled using the remaining fact symbols

Transition Rules

Three types of rules:

- A rule for the generation of fresh names
- Message deduction rules representing the adversary's capability to receive messages from and send messages to the protocol
- Rules specifying the protocol and the adversary's capabilities

Adversary Rules

The message deduction rules $\mathcal{MD}$ are defined in Equation 1.

$$\begin{aligned}\mathcal{MD} = &\{\,[\mathrm{Out}(x)] - [] \to [!\mathrm{K}(x)],\ [!\mathrm{K}(x)] - [!\mathrm{K}(x)] \to [\mathrm{In}(x)]\,\} \\ &\cup \{\,[] - [] \to [!\mathrm{K}(x : \mathit{pub})],\ [\mathrm{Fr}(x : \mathit{fresh})] - [] \to [!\mathrm{K}(x : \mathit{fresh})]\,\} \\ &\cup \left\{\,[!\mathrm{K}(f(x_1, \ldots x_k))] - [] \to [!\mathrm{K}(f(x_1, \ldots x_k)) \mid f \in \Sigma^k]\,\right\}\end{aligned} \tag{1}$$

The !K fact in the action in the second rule is used to observe the messages sent by the adversary in a trace and is used to specify secrecy properties. The second line rules represent the adversary's capabilities to learn public and freshly generated names. The last rule allows the adversary to apply any function in Σ^k to known messages.

Protocol Rules Example

$\mathcal{A} \rightarrow D : t$

The platform $\mathcal{A}$ forwards the message t received from the insecure channel to the smart-card reader D using an insecure channel:

$$\mathrm{A\_1} = [\mathrm{Recv}_{\mathrm{Insec}}(H, A, t), \mathrm{State}(A, \mathrm{A\_0}, \emptyset)] \\ -[\mathrm{Learn}(A, t)]-> \\ [\mathrm{Send}_{\mathrm{Insec}}(A, D, t), \mathrm{State}(A, \mathrm{A\_1}, t)]$$

$D \bullet\!\!\rightarrow\!\!\bullet H : t$

The smart-card reader D forwards t, received from an insecure channel to H, using a secure channel, i.e., its own display:

$$\mathrm{D\_1} = [\mathrm{Recv}_{\mathrm{Insec}}(A, D, t), \mathrm{State}(D, \mathrm{D\_0}, \langle H, pin, ltkD \rangle)] \\ -[\mathrm{Learn}(D, t)]-> \\ [\mathrm{Send}_{\mathrm{Sec}}(D, H, t), \mathrm{State}(D, \mathrm{D\_1}, \langle H, pin, ltkD, t \rangle)]$$

Transition Relation

The labeled transition relation $\rightarrow_P \subseteq \mathcal{G}^b \times \mathcal{P}(\mathcal{G}) \times \mathcal{G}^b$ for a protocol P is defined as:

$$\frac{[l]\text{-}[a]\text{->}[r] \in \mathit{ginsts}(P \cup \mathcal{MD} \cup \mathcal{CH} \cup \{\textsc{Fresh}\}) \quad \mathit{lfacts}(l) \subseteq^b S \quad \mathit{pfacts}(l) \subseteq \mathit{set}(S)}{S \xrightarrow{\mathit{set}(a)}_P ((S \setminus^b \mathit{lfacts}(l)) \cup^b \mathit{mset}(r))}, \qquad (2)$$

where $\mathit{lfacts}(l)$ and $\mathit{pfacts}(l)$ are the multisets of all linear and persistent facts in l respectively. The transition rewrites the current state with a ground instance of a protocol, message deduction, or fresh rule.

Traces of a Protocol

The set of traces of a protocol P is defined as follows:

$$\begin{aligned} traces(P) = \{ & [A_1, \ldots, A_n] \\ & | \exists S_1, \ldots, S_n \in \mathcal{G}^b.\emptyset^b \xrightarrow{A_1}_P \cdots \xrightarrow{A_n}_P S_n \\ & \wedge \forall i \neq j. \forall x. (S_{i+i} \setminus^b S_i) = \{\mathrm{Fr}(x)\}^b \Rightarrow \\ & (S_{j+1} \setminus^b S_j) \neq \{\mathrm{Fr}(x)\}^b \} \end{aligned} \tag{3}$$

The second conjunct ensures that each instance of the FRESH rule is used at most once in a trace. Each consumer of a Fr fact therefore obtains a different fresh name. Transitions labeled with $\emptyset$ are silent. We therefore define the observable trace $\overline{tr}$ of a trace tr as the subsequence of all non-silent actions in tr.

Traces of a Protocol

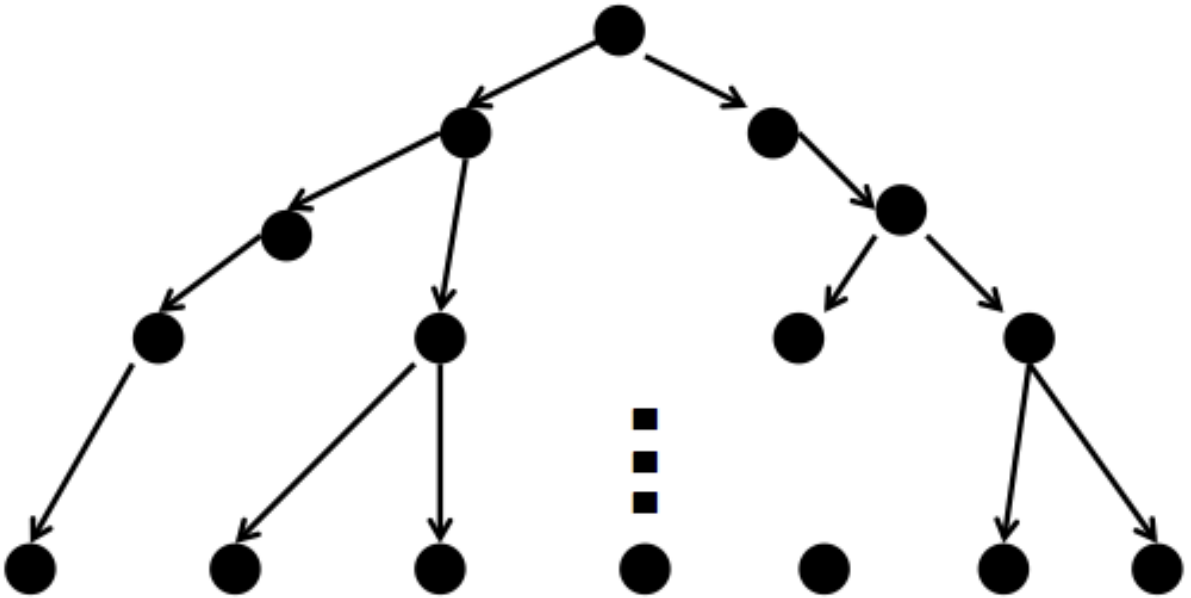

Figure: Traces of a protocol.

Traces of a Protocol

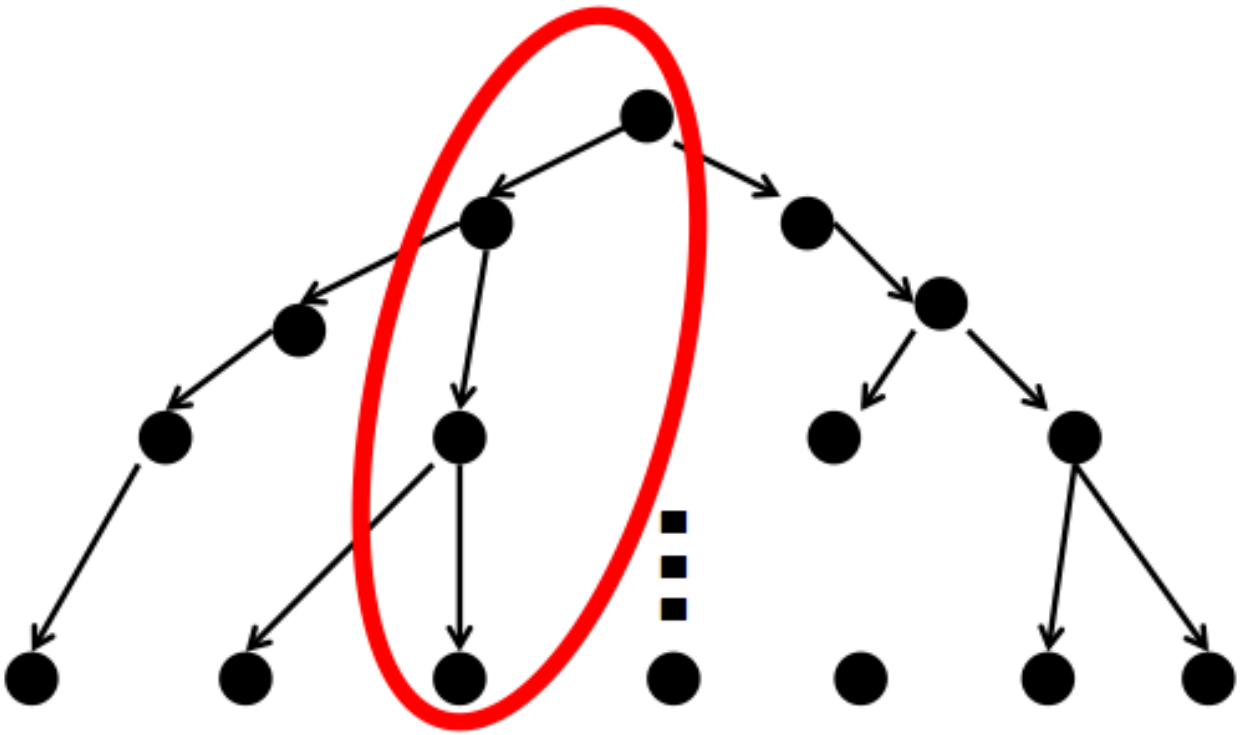

Figure: A specific trace of a protocol.

Traces of a Protocol

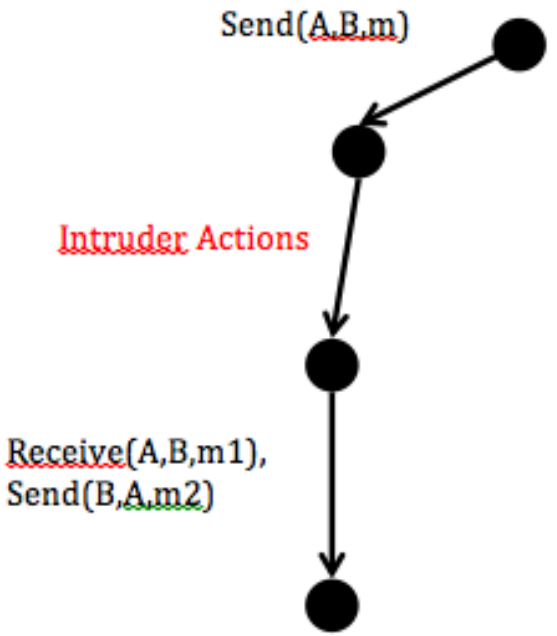

Figure: A specific trace of a protocol.

Security Goals: Secrecy

A term t is confidentially transmitted from an honest agent A to an honest agent B in a protocol P iff for all traces $tr \in traces(P)$ and time-points $i, j \in idx(tr)$, the following is satisfied:

$$\neg(\exists\, A, B, t, i, j : \mathrm{Secret}(A, B, t) \in tr_i \wedge \mathrm{K}(t) \in tr_j).$$

Security Goals: Authenticity

Aliveness: A term t is authentically transmitted from honest A to honest B in a protocol P iff for all traces $tr \in traces(P)$ and time-points $i, j \in idx(tr)$, the following is true:

$$\forall A, B', m, j : \text{Receive}(A, B', m) \in tr_j$$
$$\Longrightarrow \exists A', B, i : \text{Send}(A', B, m) \in tr_i \wedge i < j.$$

Non-injective agreement: A term t is authentically transmitted from honest A to honest B in a protocol P iff for all traces $tr \in traces(P)$ and time-points $i, j \in idx(tr)$, the following is true:

$$\forall A, B, t, j : \text{Receive}(A, B, t) \in tr_j$$
$$\Longrightarrow \exists B', i : \text{Send}(A, B', t) \in tr_i \wedge i < j.$$

Outline

Preliminaries

The Tamarin Prover

Case Studies

Conclusions

Needham-Schroeder Revisited

$$
\begin{array}{lcll}
I & \rightarrow & R & : \quad \{'1', ni, I\}_{pk(R)} \\
I & \leftarrow & R & : \quad \{'2', ni, nr\}_{pk(I)} \\
I & \rightarrow & R & : \quad \{'3', nr\}_{pk(R)}
\end{array}
$$

Figure: NSPK.

Smartcard-based Transaction Authentication

$$
\begin{array}{lcll}
H & \rightarrow & \mathcal{A} & : \quad t \\
\mathcal{A} & \rightarrow & D & : \quad t \\
D & \bullet\!\!\rightarrow & H & : \quad t \\
H & \bullet\!\!\rightarrow & D & : \quad pin \\
D & \rightarrow & \mathcal{A} & : \quad \{t\}_{sk(D)} \\
\mathcal{A} & \rightarrow & S & : \quad \langle t, \{t\}_{sk(D)} \rangle
\end{array}
$$

Figure: Transaction authentication protocol.

Outline

Preliminaries

The Tamarin Prover

Case Studies

Conclusions

Conclusions

- Intuitive protocol specification language
- Highly extensible
- Built-in DH equational theory
- Efficient tool support
- No bisimulation, i.e., no strong secrecy verification (yet)
- Protocol verification still “an art”
- Basis for more specific models (human behavior, channel abstraction, etc.)

Sources, Binaries and Further Reading

For more information, papers, sources, and binaries of the Tamarin prover visit:

`http://www.infsec.ethz.ch/research/software/tamarin`