A Framework for Security Analysis with Team Automata

Marinella Petrocchi

Istituto di Informatica e Telematica
National Research Council
IIT-CNR
Pisa, Italy

Tuesday 8 June 2004
DIMACS

with Maurice ter Beek and Gabriele Lenzini
(ISTI-CNR, Italy) (U. Twente, Netherlands)
Outline

Team Automata (TA):

origins, foundations, and examples

TA applied to security analysis:

origins and inspiration
an insecure communication scenario
Generalized Non Deducibility on Compositions (GNDC) – from process algebras to TA
compositional result for the insecure scenario

Case study: integrity of EMSS protocol

Conclusions and future work
Origins of TA

Ellis informally introduced TA at ACM GROUP’97 (Team Automata for Groupware Systems) as an extension of the I/O automata (IOA) of Lynch & Tuttle, namely:

- TA are not required to be input-enabled
- TA may synchronize on output actions
- no fixed method of composition for TA

Series of papers and Ph.D. thesis of ter Beek show that the usefulness of TA is not limited to modeling groupware, but:

extends to modeling collaboration in reactive, distributed systems in general!
Foundations of TA

- model logical architecture of system design
- abstract from concrete data and actions
- describe behavior in terms of
  - state-action diagram (automaton)
  - role of actions (input, output, internal)
  - synchronizations (simultaneous execution of shared actions)
- crux: automata composition!

+ flexible (role of actions, choice of transitions)
+ scalable (modular construction, iteration)
+ extendible (time, probabilities, priorities)
+ verifiable (automata-theoretic results)

- no tool (yet)
Example TA over Component Automata

$C_1$: \( a, b \) external actions

$C_2$: 

\[ q_1 \xrightarrow{b} q_0 \xrightarrow{a} q'_1 \] 
\[ q_2 \xrightarrow{a} q'_2 \] 

$\Rightarrow$ TA $T^\text{free} \& T^\text{ai}$ over the composable system \( \{C_1, C_2\} \) defined by choosing their transitions!

$T^\text{free}$:

$T^\text{ai}$:

$T^\text{ai} = \big|\big| \{C_1, C_2\} = \text{composition like that of IOA}$

$\Rightarrow$ every TA is a component automaton!
TA Applied to Security Analysis

ter Beek et al. first applied TA to security at ECSCW’01

(*Team Automata for Spatial Access Control*)

by specifying and analyzing a variety of access control strategies

Inspired by Lynch’ approach to use IOA for specifying and analyzing (cryptographic) communication protocols at CSFW’99

(*I/O Automaton Models and Proofs for Shared-Key Communication Systems*)

we started to apply TA in the same direction at WISP’03

(*Team Automata for Security Analysis of Multicast/Broadcast Communication*)

which meanwhile has been extended and led to

(*A Framework for Security Analysis with Team Automata*)
An Insecure Communication Scenario

An informal description of TA by their interactions:

\[ \{\text{Reveal}\} \text{ assertions} \]

\[ \{\text{Pub}\} \rightarrow \{\text{Pub}'\} \]

\[ \text{send/receive} \]

\[ \{\text{Reveal}'\} \text{ assertions} \]

\[ \text{insecure channel} \]

\[ \text{initiator} - \sum_{\text{com}}^S \text{ to communicate with } \mathcal{T}_{IC} \]

\[ \text{responder} - \sum_{\text{com}}^R \text{ to communicate with } \mathcal{T}_{IC} \]

\[ \text{intruder} - \sum_{\text{com}}^I \text{ to communicate with } \mathcal{T}_{IC} \]

\[ \sum_{\text{com}}^S \cap \sum_{\text{com}}^R \cap \sum_{\text{com}}^I = \emptyset \]

\[ \sum_{\text{com}}^P = \sum_{\text{com}}^S \cup \sum_{\text{com}}^R \]

\[ \mathcal{T}_P = \text{hid}e_{\sum_{\text{com}}^P}(\||\{\mathcal{T}_S, \mathcal{T}_R, \mathcal{T}_{IC}\}) \]

secure and

\[ \mathcal{T}_I = \text{hid}e_{\sum_{\text{com}}^I}(\||\{\mathcal{T}_P, \mathcal{T}_X\}) \]

insecure scenario
Generalized Non Deducibility on Compositions (GNDC)

\[ P \in GND C_{\leq}^{\alpha(P)} \iff (P \parallel Top_{C}^{\phi}) \setminus C \leq \alpha(P) \]

\( P \) – term of a process algebra, modeling a system running in isolation

\( \leq \) – behavioral relation (trace inclusion)

\( \alpha(P) \) – the expected (correct) behavior of \( P \)

\( Top_{C}^{\phi} \) – term modeling the most general intruder

\( \phi \) – the (bounded) initial knowledge of \( Top_{C}^{\phi} \)

\( C \) – channels used by \( Top_{C}^{\phi} \) to interact with \( P \)

\( \parallel \) – parallel composition operator

\((\_ \parallel \_ \setminus C)\) – restriction to communication over channels other than \( C \)
GNDC in Terms of TA

\[ T_P \in GND\subseteq^{\alpha(T_P)} \iff \forall C \\exists C \\forall \{T_P, Top^\phi_C\} \subseteq \alpha(T_P) \]

\( T_P \) – TA modeling secure communication scenario

\( \subseteq \) – behavioral inclusion (set of traces/language)

\( \alpha(T_P) \) – the expected (correct) behavior of \( T_P \)

\( Top^\phi_C \) – TA modeling the most general intruder

\( \phi \) – the (bounded) initial knowledge of \( Top^\phi_C \)

\( C \) – actions used by \( Top^\phi_C \) to interact with \( T_P \)

\( \|\| \{T_P, Top^\phi_C\} \) – (as before) composition like IOA

\( hide_C(T) \) – (as before) hides external actions \( C \) (as internal actions) of a TA \( T \)

\( O^C_T \) – observational behavior of a TA \( T \)

(w.r.t. actions not in \( C \))
Compositionality

Compositional reasoning, useful for

– identifying sub-problems and separately treated them

– evaluating (security) properties over sub-components

– asserting the properties validity over the whole system (e.g., using theorems about automata composition)

– other...

We decompose the insecure communication scenario, and...

**Result**: the observational behaviour of the overall system is the “shuffle” of the observational behaviours of the sub-components!
Compositional Result for Insecure Scenario

Recall: $\Sigma_{com}^P = \text{all public send/receive actions}$

Let $T_1 = \text{hide}_{\Sigma_{com}^P} (\|\| \{T_S, T_{IC}\})$

and $T_2 = \text{hide}_{\Sigma_{com}^P} (\|\| \{T_R, T_{IC}\})$

**Theorem:** if $T_1 \in \text{GNDC}_{\Sigma_1}^{O^C_{T_1}}$ and $T_2 \in \text{GNDC}_{\Sigma_2}^{O^C_{T_2}}$, then

\[
\|\| \{T_1, T_2\} \in \text{GNDC}_{\Sigma_1\cup\Sigma_2}^{\{O^C_{T_1}, O^C_{T_2}\}}
\]

$\|\| \{\Sigma_1, \Sigma_2\} \{L_1, L_2\} - \text{full synchronized shuffle of language } L_i \text{ over alphabet } \Sigma_i$

**Example:** if $L_1 = \{abc\} \subseteq \Sigma_1 = \{a, b, c\}$ and $L_2 = \{cd\} \subseteq \Sigma_2 = \{c, d\}$, then $abc \ \Sigma_1 \| \Sigma_2 \ cd = \{abcd\}$

(i.e. words must synchronize on $\Sigma_1 \cap \Sigma_2 = \{c\}$)

shuffle/free interleaving: \{abcd, acbcd, cdabc, ...\}
Case Study: Integrity of EMSS Protocol

\[
S \xrightarrow{P_0} \{R_n \mid n \geq 1\} \quad P_0 = \langle m_0, \emptyset, \emptyset \rangle
\]
\[
S \xrightarrow{P_1} \{R_n \mid n \geq 1\} \quad P_1 = \langle m_1, h(P_0), \emptyset \rangle
\]
\[
S \xrightarrow{P_i} \{R_n \mid n \geq 1\} \quad P_i = \langle m_i, h(P_{i-1}), h(P_{i-2}) \rangle \quad 2 \leq i \leq \text{last}
\]
\[
S \xrightarrow{P_{\text{sign}}} \{R_n \mid n \geq 1\} \quad P_{\text{sign}} = \langle \{h(P_{\text{last}}), h(P_{\text{last-1}})\}_{sk(s)} \rangle
\]

- modeling sender and receiver as TA $\mathcal{T}_S$, $\mathcal{T}_R$
- embed $\mathcal{T}_S$, $\mathcal{T}_R$ in the insecure communication scenario
- defining integrity as the ability of $\mathcal{T}_R$ to accept a message $m_i$ only as the ith message sent by $\mathcal{T}_S$
- evaluating the property over two subcomponents
- applying compositionality

$\Rightarrow$ allowed us to prove that integrity is guaranteed in the EMSS protocol!
Conclusions and Future Work

What has been done:

Security analysis with TA by
- defining an insecure communication scenario
- reformulating GNDC in terms of TA
- formulating some effective compositional analysis strategies

What we would like to do:

- extend the analysis to other security properties
- try to automate the currently manual specification and verification of properties
- promote TA for security analysis! :)

Questions & suggestions are welcome!
Component Automaton

\[ C = (Q, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, I) \]

- \( Q \) set of states
- \( \Sigma = \Sigma_{inp} \cup \Sigma_{out} \cup \Sigma_{int} \) alphabet (a partition!)
- \( \delta \subseteq Q \times \Sigma \times Q \) transition relation \( q \xrightarrow{a} q' \)
- \( I \subseteq Q \) set of initial states \( (q, q') \in \delta_a \)

\[ \Sigma_{inp} \text{ input actions} \]
\[ \Sigma_{out} \text{ output actions} \]
\[ \{ \Sigma_{ext} \text{ externally observable} \]  
\[ \Sigma_{int} \text{ internal actions cannot be observed} \]

Composable System

A set \( S = \{ C_1, \ldots, C_n \} \) of component automata is a composable system if \( \forall i \in \{1, \ldots, n\} \):

\[ \Sigma_{i, int} \cap \bigcup_{j \in \{1, \ldots, n\} \setminus \{i\}} \Sigma_j = \emptyset \]
Complete Transition Space

The complete transition space of \( a \in \Sigma = \bigcup_{i \in \{1, \ldots, n\}} (\Sigma_{i,\text{inp}} \cup \Sigma_{i,\text{out}} \cup \Sigma_{i,\text{int}}) \) in \( \mathcal{S} \) is

\[
\Delta_a(\mathcal{S}) = \{(q, q') \in \prod_{i \in \{1, \ldots, n\}} Q_i \times \prod_{i \in \{1, \ldots, n\}} Q_i \mid \exists j \in \{1, \ldots, n\} : (\text{proj}_j(q), a, \text{proj}_j(q')) \in \delta_j \wedge \\
\forall i \in \{1, \ldots, n\} : (\text{proj}_i(q), a, \text{proj}_i(q')) \in \delta_i \lor \text{proj}_i(q) = \text{proj}_i(q') \}
\]

\( \Rightarrow \) in every team transition at least 1 component acts according to its transition relation

\( \Rightarrow \) all other components either join or are idle
⇒ the choices of team transition relations $\delta_a$, $\forall a \in \Sigma$, define a specific TA!
Team Automaton

\[ T = ( \prod_{i \in \{1, \ldots, n\}} Q_i, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, \prod_{i \in \{1, \ldots, n\}} I_i) \]

is a TA composed over composable system \( S \) if

\[
\begin{align*}
\Sigma_{int} &= \bigcup_{i \in \{1, \ldots, n\}} \Sigma_{i, int} \\
\Sigma_{out} &= \bigcup_{i \in \{1, \ldots, n\}} \Sigma_{i, out} \\
\Sigma_{inp} &= (\bigcup_{i \in \{1, \ldots, n\}} \Sigma_{i, inp}) \setminus \Sigma_{out}
\end{align*}
\]

\( \delta \subseteq \prod_{i \in \{1, \ldots, n\}} Q_i \times \Sigma \times \prod_{i \in \{1, \ldots, n\}} Q_i \) such that

\[ \forall a \in \Sigma \quad \delta_a \subseteq \Delta_a(S) \quad \text{and} \quad \delta_a = \Delta_a(S) \text{ if } a \in \Sigma_{int} \]

\[ \Rightarrow \text{ every TA is a component automaton!} \]