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 !

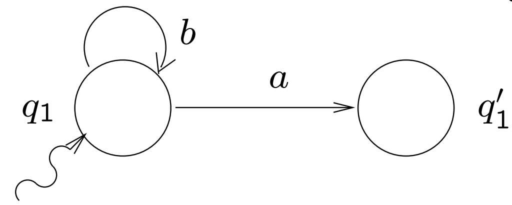
+ flexible (role of actions, choice of transitions)
+ scalable (modular construction, iteration)
+ extendible (time, probabilities, priorities)
+ verifiable (automata-theoretic results)

– no tool (yet)

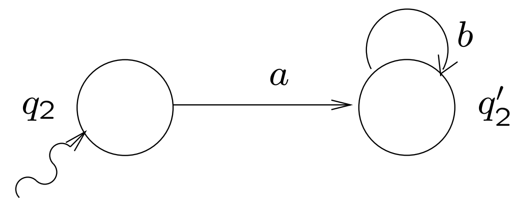
Example TA over Component Automata

$\mathcal{C}_1$:

a, b external actions

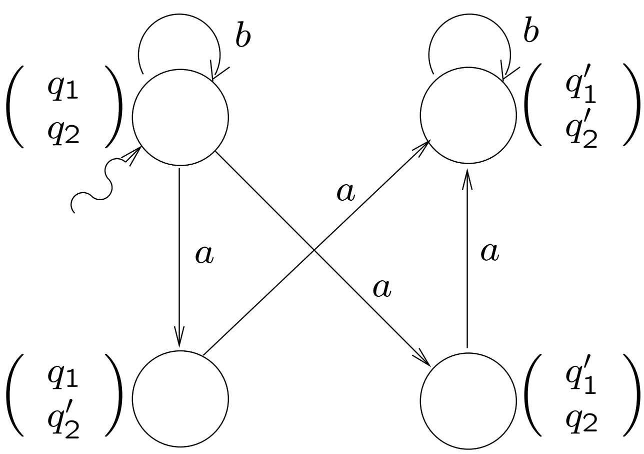


$\mathcal{C}_2$:

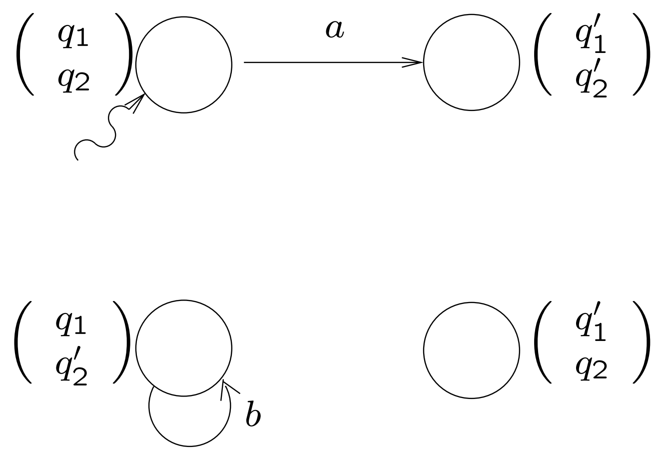


$\Rightarrow$ TA $\mathcal{T}^{free}$ & $\mathcal{T}^{ai}$ over the composable system $\{\mathcal{C}_1, \mathcal{C}_2\}$ defined by choosing their transitions !

$\mathcal{T}^{free}$:



$\mathcal{T}^{ai}$:



$\mathcal{T}^{ai} = \,|||\, \{\mathcal{C}_1, \mathcal{C}_2\}$ = 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

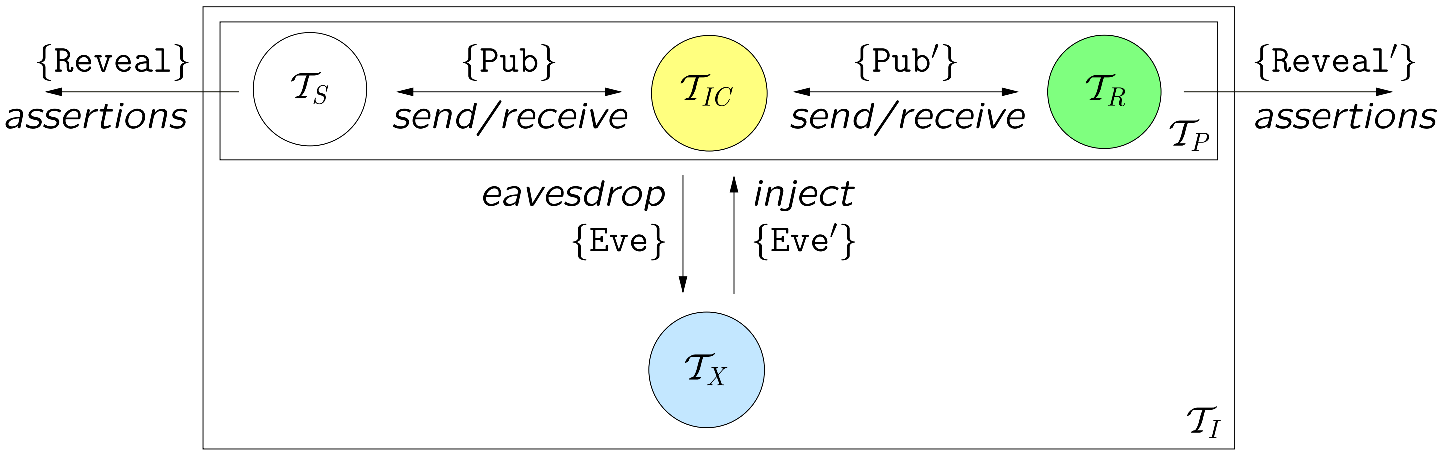
(*Team Automata for Security Analysis of Multicast/Broadcast Communication*)

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:



$\mathcal{T}_{\mathcal{IC}}$ – insecure channel
$\mathcal{T}_{\mathcal{S}}$ – initiator – Σ^S_{com} to communicate with $\mathcal{T}_{\mathcal{IC}}$
$\mathcal{T}_{\mathcal{R}}$ – responder – Σ^R_{com} to communicate with $\mathcal{T}_{\mathcal{IC}}$
$\mathcal{T}_{\mathcal{X}}$ – intruder – Σ^I_{com} to communicate with $\mathcal{T}_{\mathcal{IC}}$

$\Sigma^S_{com} \cap \Sigma^R_{com} \cap \Sigma^I_{com} = \varnothing \qquad \Sigma^P_{com} = \Sigma^S_{com} \cup \Sigma^R_{com}$

$\mathcal{T}_P = \text{hide}_{\Sigma^P_{com}}(\,|||\,\{\mathcal{T}_S, \mathcal{T}_R, \mathcal{T}_{IC}\})$ secure and

$\mathcal{T}_I = \text{hide}_{\Sigma^I_{com}}(\,|||\,\{\mathcal{T}_P, \mathcal{T}_X\})$ insecure scenario

Generalized Non Deducibility on Compositions (GNDC)

$P \in GNDC_{\leq}^{\alpha(P)}$ <u>iff</u> $(P \parallel \mathit{Top}_C^{\phi})\backslash C \leq \alpha(P)$

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^{ϕ} – term modeling the most general intruder

ϕ – the (bounded) initial knowledge of Top_C^{ϕ}

C – channels used by Top_C^{ϕ} to interact with P

$\parallel$ – parallel composition operator

$(\_ \parallel \_)\backslash C$ – restriction to communication over channels other than C

GNDC in Terms of TA

$\mathcal{T}_P \in GNDC_{\subseteq}^{\alpha(\mathcal{T}_P)}$ <u>iff</u> $\mathbf{O}^{C}_{\mathsf{hide}_C(\,|||\,\{\mathcal{T}_P, \mathit{Top}^{\phi}_{C}\})} \subseteq \alpha(\mathcal{T}_P)$

$\mathcal{T}_P$ – TA modeling secure communication scenario

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

$\alpha(\mathcal{T}_P)$ – the expected (correct) behavior of $\mathcal{T}_P$

Top^{ϕ}_{C} – TA modeling the most general intruder

ϕ – the (bounded) initial knowledge of Top^{ϕ}_{C}

C – actions used by Top^{ϕ}_{C} to interact with $\mathcal{T}_P$

$|||\,\{\mathcal{T}_P, \mathit{Top}^{\phi}_{C}\}$ – (as before) composition like IOA

$\mathsf{hide}_C(\mathcal{T})$ – (as before) hides external actions C (as internal actions) of a TA $\mathcal{T}$

$\mathbf{O}^{C}_{\mathcal{T}}$ – observational behavior of a TA $\mathcal{T}$ (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: Σ^P_{com} = all public send/receive actions

Let $\mathcal{T}_1 = \text{hide}_{\Sigma^P_{com}}(\,|||\,\{\mathcal{T}_S, \mathcal{T}_{IC}\})$

and $\mathcal{T}_2 = \text{hide}_{\Sigma^P_{com}}(\,|||\,\{\mathcal{T}_R, \mathcal{T}_{IC}\})$

Theorem: if $\mathcal{T}_1 \in GNDC^{\mathbf{O}^C_{\mathcal{T}_1}}_{\subseteq}$ and $\mathcal{T}_2 \in GNDC^{\mathbf{O}^C_{\mathcal{T}_2}}_{\subseteq}$, then

$$|||\,\{\mathcal{T}_1, \mathcal{T}_2\} \in GNDC^{\underline{||}_{\{\Sigma^{\mathcal{T}_1}, \Sigma^{\mathcal{T}_2}\}}\ \{\mathbf{O}^C_{\mathcal{T}_1}, \mathbf{O}^C_{\mathcal{T}_2}\}}_{\subseteq}$$

$\underline{||}_{\{\Sigma_1, \Sigma_2\}}\ \{L_1, L_2\}$ – *full synchronized shuffle of* language L_i over alphabet Σ_i

Example: if $L_1 = \{abc\} \subseteq \Sigma_1 = \{a, b, c\}$ and $L_2 = \{cd\} \subseteq \Sigma_2 = \{c, d\}$, then $abc\ {}_{\Sigma_1}\underline{||}_{\Sigma_2}\ cd = \{abcd\}$ (i.e. words must synchronize on $\Sigma_1 \cap \Sigma_2 = \{c\}$)

shuffle/free interleaving: $\{abccd, acbcd, cdabc, \ldots\}$

Case Study: Integrity of EMSS Protocol

$$
\begin{array}{llll}
S \xrightarrow{P_0} \{R_n \mid n \geq 1\} & P_0 & = \langle m_0, \varnothing, \varnothing \rangle & \\
S \xrightarrow{P_1} \{R_n \mid n \geq 1\} & P_1 & = \langle m_1, h(P_0), \varnothing \rangle & \\
S \xrightarrow{P_i} \{R_n \mid n \geq 1\} & P_i & = \langle m_i, h(P_{i-1}), h(P_{i-2}) \rangle & 2 \leq i \leq \mathit{last} \\
S \xrightarrow{P_{\mathit{sign}}} \{R_n \mid n \geq 1\} & P_{\mathit{sign}} & = \langle \{h(P_{\mathit{last}}), h(P_{\mathit{last}-1})\}_{sk(S)} \rangle &
\end{array}
$$

- 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 to accept a message m_i only as the *i*th message sent by $\mathcal{T}_S$

- evaluating the property over two subcomponents

- applying compositionality

⇒ 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

$$\mathcal{C} = (Q, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, I)$$

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* $\quad q \xrightarrow{a} q'$
$I \subseteq Q$ set of *initial states* $\quad (q, q') \in \delta_a$

$$\left.\begin{array}{l}\Sigma_{inp} \textit{ input actions} \\ \Sigma_{out} \textit{ output actions}\end{array}\right\} \Sigma_{ext} \text{ externally observable}$$

Σ_{int} *internal actions* — cannot be observed

Composable System

a set $\mathcal{S} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ of component automata is a *composable system* if $\forall\, i \in \{1, \ldots, n\}$:

$$\Sigma_{i,int} \cap \bigcup_{j \in \{1,\ldots,n\} \setminus \{i\}} \Sigma_j = \varnothing$$

Complete Transition Space

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

$$\Delta_a(\mathcal{S}) = \{(q, q') \in \prod_{i \in \{1,\ldots,n\}} Q_i \times \prod_{i \in \{1,\ldots,n\}} Q_i \mid$$

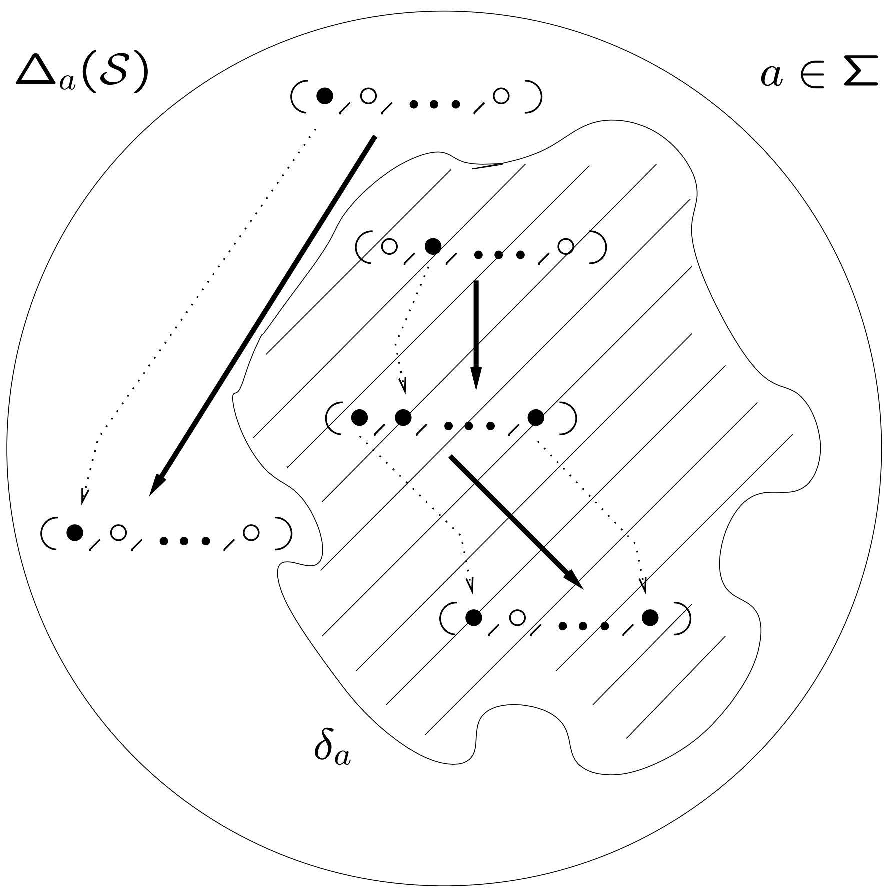
$$\exists j \in \{1, \ldots, n\} : \ (\mathbf{proj}_j(q), a, \mathbf{proj}_j(q')) \in \delta_j \ \wedge$$

$$\forall i \in \{1, \ldots, n\} : \ (\mathbf{proj}_i(q), a, \mathbf{proj}_i(q')) \in \delta_i \ \vee \ \mathbf{proj}_i(q) = \mathbf{proj}_i(q')\}$$

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

$\Rightarrow$ all other components either join or are idle

Transition Space of TA



$\Rightarrow$ the choices of team transition relations δ_a, $\forall a \in \Sigma$, define a specific TA !

Team Automaton

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

is <u>a</u> *TA composed over* composable system $\mathcal{S}$ if

$$\left.\begin{array}{l} \Sigma_{int} = \bigcup_{i \in \{1,\ldots,n\}} \Sigma_{i,int} \\ \Sigma_{out} = \bigcup_{i \in \{1,\ldots,n\}} \Sigma_{i,out} \\ \Sigma_{inp} = (\bigcup_{i \in \{1,\ldots,n\}} \Sigma_{i,inp}) \setminus \Sigma_{out} \end{array}\right\} = \Sigma$$

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

$\forall a \in \Sigma$ $\quad \delta_a \subseteq \Delta_a(\mathcal{S})$
and $\delta_a = \Delta_a(\mathcal{S})$ if $a \in \Sigma_{int}$

$\Rightarrow$ every TA is a component automaton !