A Calculus for Trust Management

Vladimiro Sassone

University of Sussex, UK

GC 2004: MyThS/MIKADO/DART Meeting

Venice 16.06.04

with M. Carbone and M. Nielsen

Trust and Trust Management

Trust: What is it?

- Think of the usual human-like notion...

Trust and Trust Management

Trust: What is it?

- Think of the usual human-like notion...
- ...but on a *global computing* scale.

Trust and Trust Management

Trust: What is it?

- Think of the usual human-like notion...
- ...but on a *global computing* scale.

Trust Management: Fundamental aspects?

1. Trust is gathered by individuals from personal experiences;
2. Trust is shared by communities, e.g. to form "reputation systems";

Trust and Trust Management

Trust: What is it?

- Think of the usual human-like notion...
- ...but on a *global computing* scale.

Trust Management: Fundamental aspects?

1. Trust is gathered by individuals from personal experiences;
2. Trust is shared by communities, e.g. to form "reputation systems";

Which means:

- Principals act according to "policies" upon consulting "trust tables," and "update" these constantly according to the outcome of transactions.

The Framework

$$a\{\ P\ \}_{\alpha}\ |\ N$$

It consists of:

- The Principal's name
- The Principal's program
- The Principal's policy
- The rest of the network

The Framework

$$a\{\ P\ \}_{\alpha} \mid N$$

It consists of:

- The Principal's name
- The Principal's program
- The Principal's policy
- The rest of the network

- $\phi :: b \cdot c\langle n \rangle$: if a can prove ϕ according to α, it will grant n to b along c. E.g.

$$x \cdot \texttt{print}(y)\,.\,Access(x, ColorPrinter) :: colPr \cdot \texttt{print}\langle y \rangle$$

The Framework

$$a\{\ P\ \}_{\alpha} \mid N$$

It consists of:

- The Principal's name
- The Principal's program
- The Principal's policy
- The rest of the network

- $\phi :: b \cdot c\langle n \rangle$: if a can prove ϕ according to α, it will grant n to b along c. E.g.

$$x \cdot \texttt{print}(y)\,.\,\mathit{Access}(x, \mathit{ColorPrinter}) :: \mathit{colPr} \cdot \texttt{print}\langle y \rangle$$

- $b \cdot c(y)\,.\,P$: Receive y from b along c, and record the observation in policy α.

The Interaction Rule

Interaction

$$\frac{\beta \vdash \phi \quad \alpha' = \alpha\,\mathsf{upd}(\mathsf{b} \cdot \mathsf{c} \triangleright \tilde{\mathsf{m}}) \quad \mathsf{b} : \tilde{\mathsf{m}}\ \texttt{match}\ \mathsf{p} : \tilde{\mathsf{x}} = \sigma}{a\{\, p \cdot c(\tilde{x}).P \,\}_\alpha \mid b\{\, \phi :: a \cdot c\langle \tilde{m} \rangle.Q \,\}_\beta \to a\{\, P\sigma \,\}_{\alpha'} \mid b\{\, Q \,\}_\beta}$$

The logic

$\mathsf{Val} = \mathsf{P} + \mathsf{N}$.

$\overline{\mathsf{Val}} = P \times \mathsf{Val}^+$: observations $(p, ch, mess)$.

Definition

Fix a signature Σ augmented with:

- constants $\overline{\mathsf{Val}}$;
- $upd : s \times \overline{\mathsf{Val}} \to s$ (s distinguished sort).

Definition

A message structure S, Op is a term algebra for the Σ above. Let $\mathcal{R}$ be a set of predicate symbols.

Let π be a set of Horn clauses $L \leftarrow L_1, \ldots L_k$ over such S and $\mathcal{R}$.

Principal's policies α is of the form $(\pi, \#)$, for $\# \in S$.

The calculus

Definition

$$
\begin{array}{rlr}
N, M ::= & \epsilon & \text{(empty)} \\
& \mid N \mid N & \text{(net-par)} \\
& \mid a\{\, P \,\}_{\alpha} & \text{(principal)} \\
& \mid\mid (\nu n)\, N & \text{(new-net)}
\end{array}
\qquad
\begin{array}{rlr}
P, Q ::= & \mathbf{0} & \text{(null)} \\
& \mid Z & \text{(sub)} \\
& \mid P \mid P & \text{(par)} \\
& \mid (\nu n)\, P & \text{(new)} \\
& \mid\, !P & \text{(bang)}
\end{array}
$$

$$
\begin{array}{rlr}
Z ::= & p \cdot u(\tilde{v})\,.\,P & \text{(output)} \\
& \mid \phi :: p \cdot u\langle \tilde{v} \rangle\,.\,P & \text{(input)} \\
& \mid Z + Z & \text{(sum)}
\end{array}
\qquad
\begin{array}{rlr}
\phi ::= & L(\tilde{I}) \quad L \in \mathcal{P} & \text{(null)}
\end{array}
$$

Example: A print server

Basic predicate $Access(x, y)$, for x a principal and $y \in \{Color, BW\}$.

$$\text{Site policy } \pi : \{\ x \cdot - \rhd \texttt{junk} < 3 \rightarrow Access(x, Color),$$
$$x \cdot - \rhd \texttt{junk} < 6 \rightarrow Access(x, BW)\}$$

where $x \cdot - \rhd \texttt{junk}$ counts the occurrences of `junk` messages.

Example: A print server

Basic predicate $Access(x, y)$, for x a principal and $y \in \{Color, BW\}$.

$$\text{Site policy } \pi : \{\ x \cdot - \rhd \texttt{junk} < 3 \rightarrow Access(x, Color),$$
$$x \cdot - \rhd \texttt{junk} < 6 \rightarrow Access(x, BW)\}$$

where $x \cdot - \rhd \texttt{junk}$ counts the occurrences of $\texttt{junk}$ messages.

Let a, the print server, and b be principals with resp. protocols:

$$P = !x \cdot \texttt{printCol}(y) \,.\, Access(x, Color) :: printer \cdot \texttt{printCol}\langle y \rangle \mid$$
$$!x \cdot \texttt{printBW}(y) \,.\, Access(x, BW) :: printer \cdot \texttt{printBW}\langle y \rangle$$

Example: A print server

Basic predicate $Access(x, y)$, for x a principal and $y \in \{Color, BW\}$.

$$\text{Site policy } \pi : \{\ x \cdot - \rhd \texttt{junk} < 3 \rightarrow Access(x, Color),$$
$$x \cdot - \rhd \texttt{junk} < 6 \rightarrow Access(x, BW)\}$$

where $x \cdot - \rhd \texttt{junk}$ counts the occurrences of `junk` messages.

Let a, the print server, and b be principals with resp. protocols:

$$P = !x \cdot \texttt{printCol}(y) \,.\, Access(x, Color) :: printer \cdot \texttt{printCol}\langle y \rangle \mid$$
$$!x \cdot \texttt{printBW}(y) \,.\, Access(x, BW) :: printer \cdot \texttt{printBW}\langle y \rangle$$

$$Q = a \cdot \texttt{printCol}\langle \texttt{junk} \rangle \,.\, a \cdot \texttt{printBW}\langle \texttt{junk} \rangle \,.\, a \cdot \texttt{printCol}\langle \texttt{junk} \rangle$$
$$\mid a \cdot \texttt{printCol}\langle \texttt{doc} \rangle$$

Consider $N = a\{\ P\ \}_{(\pi, \emptyset)} \mid b\{\ Q\ \}_{\alpha}$.

Example: A bank recommendation system

Interpret messages as recommendations.

Assume message structure is list of last k recommendations for each user. Let's consider the protocol

$$P = !x \cdot \mathtt{mg}(y) \,.\, \mathtt{Grant}(x, y) :: x \cdot \mathtt{mg}\langle\rangle \,.\, x \cdot \mathtt{pay}(y) \mid$$
$$!\mathtt{ITAbank} \cdot \mathtt{rec}(x, y)$$

Policy for principal *UKBank*:

$$\pi = \{\mathtt{ITAbank} \cdot \mathtt{rec} \rhd (x, \mathtt{Bad}) + x \cdot \mathtt{pay} \rhd \mathtt{no} = 0 \rightarrow \mathtt{Grant}(x, y)\}$$

which checks if the sum of messages from $\mathtt{ITAbank}$ of type $(x, \mathtt{Bad})$ and from x of type $\mathtt{no}$ is zero.

Mortgage allowed whenever there is not bad observed or bad recommended behaviour.

Results

A nice cluster of bisimulations I don't have time to tell you about.