

Secure Contexts for Confidential Data

A. Bossi, D. Macedonio, C. Piazza, and S. Rossi

Department of Computer Science
University Ca' Foscari of Venezia

{bossi,mace,piazza,srossi}@dsi.unive.it

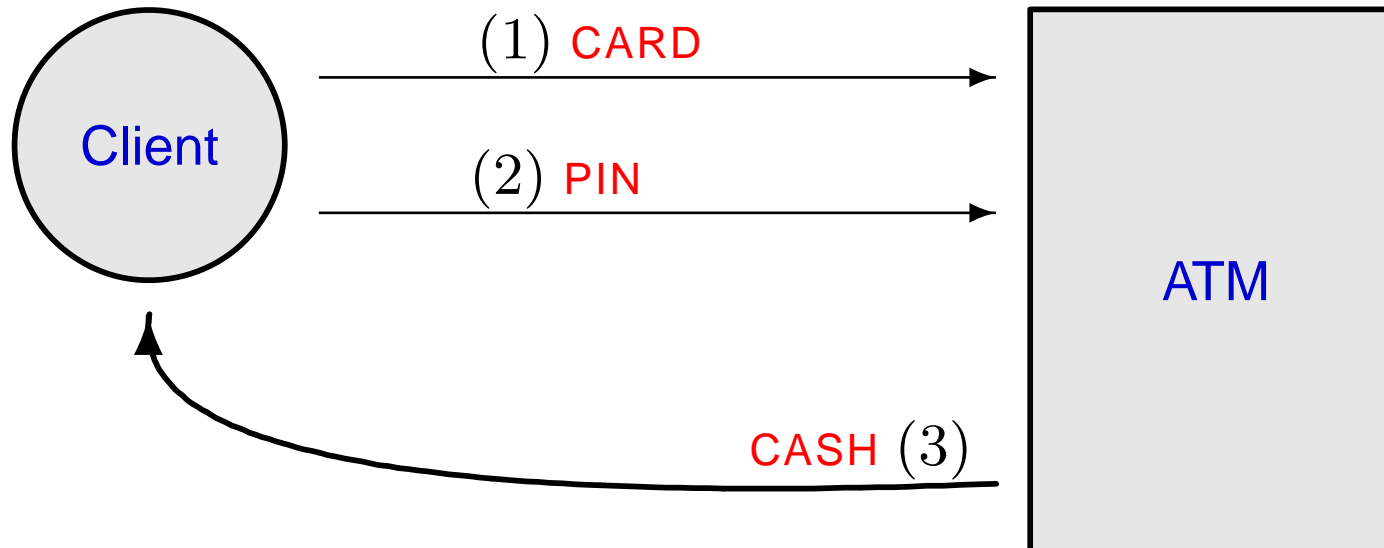
CSFW 2003, June 30 - July 2, 2003, Asilomar

Protect Confidential Data in a Multilevel System

- ▷ **Information Flow Security** aims at guaranteeing that no **high** level (**confidential**) information is revealed to users at **low** level, even in the presence of any possible **malicious process**
- ▷ **Non-Interference** [Goguen-Meseguer'82]: **information does not flow** from high to low if the **high behavior** has **no effect** on what **low** level can **observe**
- ▷ Such a requirement could be **inadequate** when we have some knowledge about the **context** in which the process is going to run (e.g., not all the **malicious processes** are admissible)

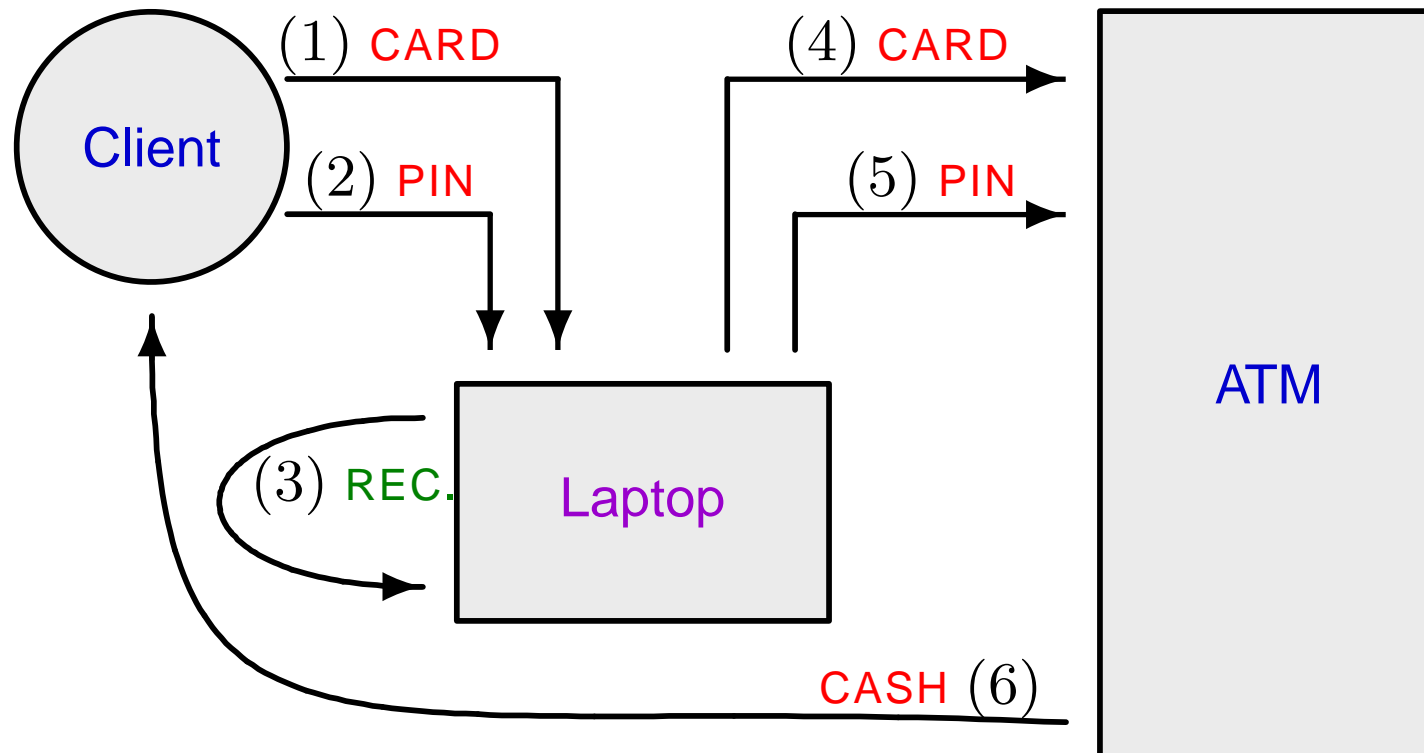
The ATM example - 1

A **client** uses his cash card in an **Automatic Teller Machine** to make a **withdrawal** from his account



The **withdrawal** is “**secure**” in this environment

The ATM example - 2



The **withdrawal** is **not secure** in this environment

Plan of the Talk

- ▷ The SPA language: syntax and semantics
- ▷ The notion of Secure Contexts
- ▷ Two perspectives
- ▷ Secure Contexts using Bisimulation
- ▷ Secure Contexts using Trace Equivalence

The SPA syntax

T	$::=$	0	<i>empty process</i>
		Z	<i>variable</i>
		$a.T$	<i>prefix</i>
		$T + T$	<i>non-det. choice</i>
		$T \mid T$	<i>parallel composition</i>
		$T \setminus v$	<i>restriction</i>
		$T[f]$	<i>relabelling</i>
		$recZ.T$	<i>recursion</i>

- ▷ H high actions and L low actions
- ▷ Contexts C : terms in which free variables can occur.
- ▷ Processes E : terms without free variables

The SPA semantics

Semantics given through transition relations

Input		Output	
	$a.E \xrightarrow{a} E$		$a.E \xrightarrow{\bar{a}} E$
Parallel	$E_1 \xrightarrow{a} E'_1$		$E_1 \xrightarrow{a} E'_1 \quad E_2 \xrightarrow{\bar{a}} E'_2$
	$E_1 E_2 \xrightarrow{a} E'_1 E_2$		$E_1 E_2 \xrightarrow{\tau} E'_1 E'_2$

Two processes are **equivalent** if they are **weak bisimilar**: $E \approx F$

Secure Contexts

\sim observational equivalence, used to equate two processes

\cdot_l low level view which determines

E_l : low level behavior of the process E

\sim_l : low level equivalence ($E \sim_l F$ stands for $E_l \sim F_l$)

\mathcal{C} class of contexts, \mathcal{P} class of processes, and X a variable.

\mathcal{C} is secure for \mathcal{P} with respect to X if

$$\forall C[X] \in \mathcal{C}, \forall E \in \mathcal{P}, \quad C[E] \sim_l C[E_l]$$

A low level user cannot discern whether \mathcal{C} is interacting with E or E_l

Two Perspectives

Consider one process E and one context $C[X]$

$$C[E] \sim_l C[E_l]$$

can be interpreted in two way:

- ▷ **Security for the process:** C is not able to reveal any high level information of E , since it reveals only the information that is revealed by the interaction with E_l
- ▷ **Security for the context:** E is not able to reveal any high information of C , since it reveals the same information which can be revealed by E_l

Example: Security for the Process

Let E be a Java applet which allows WHOLESALER's customers to get the **price-list**, while the rest of the world can only see the **product-list**

$$E = \text{PWD_SHOPKEEPER}.\text{PRICES} + \text{PRODUCTS}$$

Possible machines on which E is going to run:

▷ $X | \overline{\text{PWD_SHOPKEEPER}}.\mathbf{0}$

▷ $\text{PWD_HIGH}.(X | \overline{\text{PWD_SHOPKEEPER}}.\mathbf{0}) + \text{PWD_LOW}.X$

Example: Security for the Context

MR EARNER has on his own machine C some files containing the **information** about his investments. He bought a program E which **checks** on the stock market, reads the files and determines whether the investments are profitable, and, if necessary, **checks** again on the stock market, for better opportunities.

$$C = X | \overline{\text{GOOD}}.0 \quad \text{or} \\ X | \overline{\text{BAD.SUGGESTIONS}}.0$$

$$E = \text{CHECK}.(\text{GOOD}.0 + \\ \text{BAD.CHECK}.\overline{\text{SUGGESTIONS}}.0)$$

Instance 1: Weak Bisimulation

$$E \xRightarrow{\hat{a}} E' = E(\xrightarrow{\tau})^* \xrightarrow{a} (\xrightarrow{\tau})^* E'$$

$$E \xRightarrow{\hat{\tau}} E' = E(\xrightarrow{\tau})^*$$

Weak Bisimulation

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ over SPA processes such that if $(E, F) \in \mathcal{S}$ then:

$E \xrightarrow{a} E'$ implies $F \xRightarrow{\hat{a}} F'$ and $(E', F') \in \mathcal{S}$

$F \xrightarrow{a} F'$ implies $E \xRightarrow{\hat{a}} E'$ and $(E', F') \in \mathcal{S}$

$E \approx_B F$ if there exists a weak bisimulation \mathcal{S} containing (E, F)

Instance 1: Weak Bisimulation and Restriction

- ▷ $E \sim F$ iff $E \approx_B F$
- ▷ E_l is $E \setminus H$

\mathcal{C} is secure for \mathcal{P} with respect to X iff

$$\forall C[X] \in \mathcal{C}, \forall E \in \mathcal{P}, C[E] \setminus H \approx_B C[E \setminus H] \setminus H$$

The ATM example - 3

- ▷ Client:

$$E = \overline{\text{CARD}}.\overline{\text{PIN}}.\text{CASH}.\mathbf{0}$$

- ▷ Automatic Teller Machine:

$$ATM = \text{CARD}.\text{PIN}.\overline{\text{CASH}}.\mathbf{0}$$

- ▷ Laptop:

$$Lap = \text{CARD}.\text{PIN}.\text{RECORD}.\overline{\text{CARD}}.\overline{\text{PIN}}.\mathbf{0}$$

- ▷ Isolated System:

$$C_1[X] = X|ATM$$

- ▷ System with Laptop:

$$C_2[X] = X|Lap|ATM$$

The ATM example - 4

C_1 is **secure** for E since:

$$\begin{aligned}
 C_1[E] \setminus H &= (E | ATM) \setminus H \\
 &= (\overline{CARD.PIN.CASH.0} | CARD.PIN.\overline{CASH.0}) \setminus H \\
 &\approx_B \tau.\tau.\tau.0
 \end{aligned}$$

$$\begin{aligned}
 C_1[E \setminus H] \setminus H &= (E \setminus H | ATM) \setminus H \\
 &= ((\overline{CARD.PIN.CASH.0}) \setminus H | CARD.PIN.\overline{CASH.0}) \setminus H \\
 &\approx_B 0
 \end{aligned}$$

The ATM example - 3

C_2 is **not secure** for E since:

$$\begin{aligned}
 C_2[E] \setminus H &= (E \mid ATM \mid Lap) \setminus H \\
 &= (\overline{CARD.PIN.CASH.0} \mid \overline{CARD.PIN.CASH.0} \mid \\
 &\quad \overline{CARD.PIN.RECORD.CARD.PIN.0}) \setminus H \\
 &\approx_B \tau.\tau.\tau.0 + \tau.\tau.RECORD.\tau.\tau.0
 \end{aligned}$$

$$\begin{aligned}
 C_2[E \setminus H] \setminus H &= (E \setminus H \mid ATM \mid Lap) \setminus H \\
 &\approx_B (0 \mid \overline{CARD.PIN.CASH.0} \mid \\
 &\quad \overline{CARD.PIN.RECORD.CARD.PIN.0}) \setminus H \\
 &\approx_B 0
 \end{aligned}$$

An Instance of our First Instance

The **BNDC** property [Focardi-Gorrieri'94], which formalizes the notion of **Non-Interference** [Goguen-Meseguer'82], can be seen as **an instance of our security notion**

▷ Definition: $E \in \text{BNDC}$ iff $\forall \Pi \in \mathcal{E}_H, E \setminus H \approx_B (E|\Pi) \setminus H$

▷ Characterization: $E \in \text{BNDC}$ iff it is **secure**

in all contexts $C[X] \equiv X|\Pi, \Pi \in \mathcal{E}_H$

Classes of Secure Contexts

A class of contexts which are **secure for all the processes** is the minimum class \mathcal{C}_s such that:

- ▷ if $F \in \mathcal{E}$, then $F \in \mathcal{C}_s$
- ▷ if Y is a variable, then $Y \in \mathcal{C}_s$
- ▷ if $C_i \in \mathcal{C}_s$ for all $i \in I$, then $\sum_{i \in I} a_i \cdot C_i \in \mathcal{C}_s$
- ▷ if $C \in \mathcal{C}_s$, then $C \setminus v \in \mathcal{C}_s$
- ▷ if $C \in \mathcal{C}_s$, then $C[f] \in \mathcal{C}_s$
- ▷ if $C \in \mathcal{C}_s$, then $recY.C \in \mathcal{C}_s$

No parallel compositions:

X is secure for \mathcal{E} , but $X|X$ is not secure for $h.l.\mathbf{0} + \bar{h}.\mathbf{0}$

Secure Contexts for P_BNDC processes

P_BNDC [Focardi-Rossi'02]

$E \in \text{P_BNDC}$ iff $\forall E'$ reachable from E , $E' \in \text{BNDC}$

$C[X]$ is a P_BNDC-secure with respect to X if

- ▷ for all $E \in \text{P_BNDC}$ it holds that $C[E] \in \text{P_BNDC}$
- ▷ it is secure for P_BNDC with respect to X

Parallel compositions:

If C and D are P_BNDC-secure, then $C|D$ is P_BNDC-secure

The Wholesaler example - 2

We had the Java applet

$$E = \text{PWD_SHOPKEEPER.PRICES.0} + \text{PRODUCTS.0}$$

Which is not secure in the context

$$C = X | \overline{\text{PWD_SHOPKEEPER.0}}$$

WHOLESALE knows that people usually store passwords. It changes the applet giving in output an **encrypted price-list**

$$E' = \text{PWD_SHOPKEEPER.ENC_PRICES.0} + \text{PRODUCTS.0}$$

The Wholesaler example - 3

Unfortunately, E' is not secure in C

$$C[E'] \setminus H = \tau.0 + \text{PRODUCTS}.0$$

$$C[E' \setminus H] \setminus H = \text{PRODUCTS}.0$$

The two processes are **not weak bisimilar**

Bisimulation seems **too demanding** in this case

Note that $C[E'] \setminus H$ and $C[E' \setminus H] \setminus H$ are **trace equivalent**

Instance 2: Trace Equivalence

$$E \xrightarrow{a_1 \dots a_n} E' = E(\tau)^* \xrightarrow{a_1} (\tau)^* \dots (\tau)^* \xrightarrow{a_n} (\tau)^* E'$$

Trace

For any process $E \in \mathcal{E}$ the set of traces associates with E is

$$Tr(E) = \{a_1 \dots a_n \in \mathcal{L}^* : \exists E', E \xrightarrow{a_1 \dots a_n} E'\}$$

Trace Equivalence

Two processes are trace equivalent if they have the same traces, i.e.

$$E \approx_T F \quad \text{iff} \quad Tr(E) = Tr(F)$$

Instance 2: Trace Equivalence and Restriction

▷ $E \sim_l F$ iff $E \setminus H \approx_T F \setminus H$

▷ E_l is $E \setminus H$

\mathcal{C} is secure for \mathcal{P} with respect to X iff

$$\forall C[X] \in \mathcal{C}, \forall E \in \mathcal{P}, C[E] \setminus H \approx_T C[E \setminus H] \setminus H$$

A special instance: **NDC** [Focardi-Gorrieri'95]

▷ Definition: $E \in \text{NDC}$ iff $\forall \Pi \in \mathcal{E}_H, E \setminus H \approx_T (E|\Pi) \setminus H$

▷ Characterization: $E \in \text{NDC}$ iff it is **secure**

for all contexts $C[X] \equiv X|\Pi, \Pi \in \mathcal{E}_H$

Conclusions

- ▷ We presented a **general notion of security**
- ▷ The notion of **secure context** for a process is parametric, i.e.,
 - ▷ it can be used to **restrict** the set of possible **attackers**
(e.g., if some level passwords cannot be guessed)
 - ▷ it allows to **enlarge** the set of possible **attackers**
(SPA operators can be combined in the contexts construction)
- ▷ We studied two instances: **bisimulation** and **trace equivalence**
- ▷ We showed that **BNDC** and **NDC** are instances of our notion