A weakest precondition approach to active attacks analysis

Musard Balliu   Isabella Mastroeni

School of Computer Science and Communication
Royal Institute of Technology (KTH)
Stockholm, Sweden

Dipartimento di Informatica
Università di Verona
Italy

Dublin, June 15th, 2009

Security Background

Goal: Protect data confidentiality from malicious attackers.

System data:

- H stands for private, unmodifiable
- L stands for public, modifiable

Standard Non Interference

Aims to protect private inputs. (H $\not\rightsquigarrow$ L)

$$\forall l \in \mathbb{V}^{\mathrm{L}}, \forall h_1, h_2 \in \mathbb{V}^{\mathrm{H}}. [\![P]\!](h_1, l)^{\mathrm{L}} = [\![P]\!](h_2, l)^{\mathrm{L}}$$

PROBLEM

$\Downarrow$

Real systems release private information intentionally.

Security Background

Goal: Protect data confidentiality from malicious attackers.

System data:

- H stands for private, unmodifiable
- L stands for public, modifiable

Standard Non Interference

Aims to protect private inputs. ($\mathtt{H} \not\rightsquigarrow \mathtt{L}$)

$$\forall l \in \mathbb{V}^{\mathtt{L}}, \forall h_1, h_2 \in \mathbb{V}^{\mathtt{H}}. [\![P]\!](h_1, l)^{\mathtt{L}} = [\![P]\!](h_2, l)^{\mathtt{L}}$$

PROBLEM

$\Downarrow$

Real systems release private information intentionally.

Security Background

Goal: Protect data confidentiality from malicious attackers.

System data:

- H stands for private, unmodifiable
- L stands for public, modifiable

Standard Non Interference

Aims to protect private inputs. (H $\not\rightsquigarrow$ L)

$$\forall l \in \mathbb{V}^{\mathtt{L}}, \forall h_1, h_2 \in \mathbb{V}^{\mathtt{H}}. [\![P]\!](h_1, l)^{\mathtt{L}} = [\![P]\!](h_2, l)^{\mathtt{L}}$$

PROBLEM

$\Downarrow$

Real systems release private information intentionally.

Security Background

Goal: Protect data confidentiality from malicious attackers.

Solution

$\Downarrow$

Declassified Non Interference

$\phi(H)$: declassified private property ($\phi(H) \rightsquigarrow$ L)

$$\forall l \in \mathbb{V}^{\mathtt{L}}, \forall h_1, h_2 \in \mathbb{V}^{\mathtt{H}}.$$
$$\phi(h_1) = \phi(h_2) \;\Rightarrow\; [\![P]\!](h_1, l)^{\mathtt{L}} = [\![P]\!](h_2, l)^{\mathtt{L}}$$

No property stronger than $\phi(H)$ can be disclosed.

[Myers and Liskov 1997, Sabelfeld and Myers 2003]

Robustness [Myers et al. 2004]

Goal: Active attacks vs Passive attacks power.

- Additional integrity level.
- Active attackers: Can modify data in fixed points called holes $[\bullet]$.
- Security type: LL, LH, HL and HH (confidentiality, integrity)

$$c[\bullet] ::= \textbf{skip} \mid x := e \mid c_1; c_2 \mid \textbf{if } e \textbf{ then } c_1 \textbf{ else } c_2 \mid \textbf{while } e \textbf{ do } c \mid [\bullet]$$

- Fair attacks: Programs on LL variables.

Robustness

$P[\bullet]$ is robust if no active fair attack can disclose more private information than a passive attacker.

Robustness [Myers et al. 2004]

Goal: Active attacks vs Passive attacks power.

- Additional integrity level.
- Active attackers: Can modify data in fixed points called holes $[\bullet]$.
- Security type: LL, LH, HL and HH (confidentiality, integrity)

$$c[\bullet] ::= \textbf{skip} \mid x := e \mid c_1; c_2 \mid \textbf{if } e \textbf{ then } c_1 \textbf{ else } c_2 \mid \\ \textbf{while } e \textbf{ do } c \mid [\bullet]$$

- Fair attacks: Programs on LL variables.

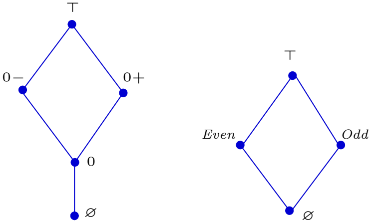
Robustness

$P[\bullet]$ is robust if no active fair attack can disclose more private information than a passive attacker.

Abstract Interpretation [Cousot and Cousot '77,'79]

Abstract Interpretation:

A general theory of sound approximation of program semantics.



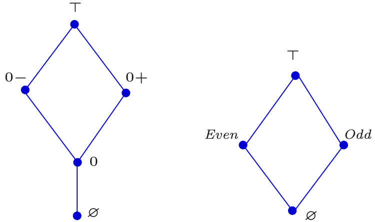
$sum(x, y) \stackrel{\text{def}}{=} x + y \quad \rightsquigarrow$

- $sum^*(+, +) = +$
- $sum^*(-, -) = -$
- $sum^*(+, -) = \top$
- $sum^*(even, even) = even$
- $sum^*(odd, odd) = even$
- $sum^*(even, odd) = odd$

Abstract Interpretation [Cousot and Cousot '77,'79]

Abstract Interpretation:

A general theory of sound approximation of program semantics.



$sum(x, y) \stackrel{\text{def}}{=} x + y \quad \rightsquigarrow$

- $sum^*(+, +) = +$
- $sum^*(-, -) = -$
- $sum^*(+, -) = \top$
- $sum^*(even, even) = even$
- $sum^*(odd, odd) = even$
- $sum^*(even, odd) = odd$

Declassification by *Wlp* [Banerjee et al. 2007]

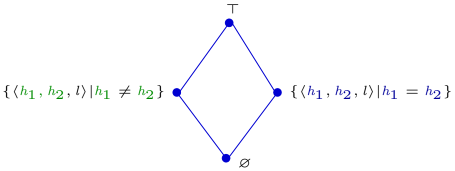
Wlp:

Greatest set of input states leading to a given output observation.

$$P \stackrel{\text{def}}{=} \textbf{if } (h_1 = h_2) \textbf{ then } l := 0; \textbf{ else } l := 1;$$

$$\mathit{Wlp}(P, l = a) = (h_1 = h_2 \wedge a = 0) \vee (h_1 \neq h_2 \wedge a = 1)$$

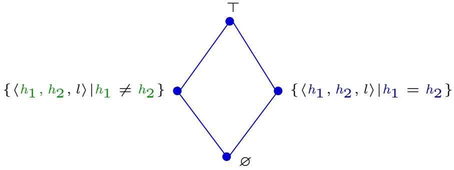
$\Downarrow$ Maximal information released



Declassification by *Wlp* [Banerjee et al. 2007]

Wlp:

Greatest set of input states leading to a given output observation.



From non-interference point of view

$$h_1 = 0, h_2 = 0,\ l = 0 \rightsquigarrow l = 0$$

$$h_1 = 1, h_2 = 0,\ l = 0 \rightsquigarrow l = 1$$

Maximal release by active attackers

Goal:

Compute the maximal information disclosed by active attackers.

$\Rightarrow$ Unfair attacks: Programs on LL and HL variables.

$P ::= l := h; [\bullet];$ with variables $h : \mathrm{HH}$, $l : \mathrm{LL}$ and $k : \mathrm{HL}$.

- $Wlp(l := h; [skip], \{l = a\}) = \{h = a\}$
- $Wlp(l := h; [l := k], \{l = a\}) = \{k = a\}$
- $Wlp(l := h; [l := l + k], \{l = a\}) = \{h + k = a\}$

- Active attackers $\Rightarrow$ Semantic transformation.
- Different attacks $\Rightarrow$ Different information release.

Active attacks can be potentially infinite!

Maximal release by active attackers

Goal:

Compute the maximal information disclosed by active attackers.

$\Rightarrow$ Unfair attacks: Programs on LL and HL variables.

$P ::= l := h; [\bullet];$ with variables h : HH, l : LL and k : HL.

- $\mathit{Wlp}\big(l := h; [\mathit{skip}], \{l = a\}\big) = \{h = a\}$
- $\mathit{Wlp}\big(l := h; [l := k], \{l = a\}\big) = \{k = a\}$
- $\mathit{Wlp}\big(l := h; [l := l + k], \{l = a\}\big) = \{h + k = a\}$

- Active attackers $\Rightarrow$ Semantic transformation.
- Different attacks $\Rightarrow$ Different information release.

Active attacks can be potentially infinite!

Maximal release by active attackers

Goal:
Compute the maximal information disclosed by active attackers.

$\Rightarrow$ Unfair attacks: Programs on $\mathtt{LL}$ and $\mathtt{HL}$ variables.

$P ::= l := h; [\bullet];$ with variables $h : \mathtt{HH}$, $l : \mathtt{LL}$ and $k : \mathtt{HL}$.

- $\mathit{Wlp}\big(l := h; [\mathit{skip}], \{l = a\}\big) = \{h = a\}$
- $\mathit{Wlp}\big(l := h; [l := k], \{l = a\}\big) = \{k = a\}$
- $\mathit{Wlp}\big(l := h; [l := l + k], \{l = a\}\big) = \{h + k = a\}$

- Active attackers $\Rightarrow$ Semantic transformation.
- Different attacks $\Rightarrow$ Different information release.

Active attacks can be potentially infinite!

Parametric attacks

Active attack $\equiv$ function on LL and HL variables.

- Extend the *Wlp* computation parametric on $f(\vec{l})$.
- Analyze the final formula containing f as parameter.

Back to the example

Consider the above example. Represent the possible unfair attacks in $[\bullet]$ with $\langle l, k\rangle := \langle f(l,k), g(l,k)\rangle$.

$$\{f(h,k) = a\}$$
$$l := h;$$
$$\{f(l,k) = a\}$$
$$[\langle l, k\rangle := \langle f(l,k), g(l,k)\rangle;]$$
$$\{l = a\}$$

$\Rightarrow \{f(h,k) = a\}$: f "measures" the information of h and k.

Parametric attacks

Active attack $\equiv$ function on `LL` and `HL` variables.

- Extend the *Wlp* computation parametric on $f(\vec{l})$.
- Analyze the final formula containing f as parameter.

Back to the example

Consider the above example. Represent the possible unfair attacks in $[\bullet]$ with $\langle l, k\rangle := \langle f(l,k), g(l,k)\rangle$.

$$\{f(h,k) = a\}$$
$$l := h;$$
$$\{f(l,k) = a\}$$
$$[\langle l, k\rangle := \langle f(l,k), g(l,k)\rangle;]$$
$$\{l = a\}$$

$\Rightarrow \{f(h,k) = a\}$: f "measures" the information of h and k.

I/O Analysis

$$\{h_2 \ \textit{mod}\ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ \textit{mod}\ 2 = a\}$$
$$h_2 := h_2 \ \textit{mod}\ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ mod \ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod \ 2 = a\}$$
$$h_2 := h_2 \ mod \ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_2 := h_2 \ \mathit{mod}\ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_2 := h_2 \ \mathit{mod}\ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ mod \ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod \ 2 = a\}$$
$$h_2 := h_2 \ mod \ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_2 := h_2 \ \mathit{mod}\ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

I/O Analysis

$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_1 := h_2;$$
$$\{h_2 \ \mathit{mod}\ 2 = a\}$$
$$h_2 := h_2 \ \mathit{mod}\ 2;$$
$$\{h_2 = a\}$$
$$l_1 := h_2;$$
$$\{l_1 = a\}$$
$$h_2 := h_1;$$
$$\{l_1 = a\}$$
$$l_2 := h_2;$$
$$\{l_1 = a\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \; mod \; 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \; mod \; 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \; mod \; 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \ mod\ 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod\ 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \ mod\ 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \ mod\ 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod\ 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \ mod\ 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \; mod \; 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \; mod \; 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \; mod \; 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2\ mod\ 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$

$$h_1 := h_2;$$

$$\{h_2\ mod\ 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$

$$h_2 := h_2\ mod\ 2;$$

$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$

$$l_1 := h_2;$$

$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$

$$h_2 := h_1;$$

$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$

$$l_2 := h_2;$$

$$\{l_1 = a \wedge [l_2 = b]\}$$

$$l_2 := l_1;$$

$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \ mod \ 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod \ 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \ mod \ 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Trace Analysis [Mastroeni and Banerjee 2008]

$$\{h_2 \ mod\ 2 = a \wedge h_2 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_1 := h_2;$$
$$\{h_2 \ mod\ 2 = a \wedge h_1 = b \wedge l_2 = c \wedge l_1 = d\}$$
$$h_2 := h_2 \ mod\ 2;$$
$$\{h_2 = a \wedge h_1 = b \wedge l_2 = c \wedge [l_1 = d]\}$$
$$l_1 := h_2;$$
$$\{l_1 = a \wedge h_1 = b \wedge l_2 = c\}$$
$$h_2 := h_1;$$
$$\{l_1 = a \wedge h_2 = b \wedge [l_2 = c]\}$$
$$l_2 := h_2;$$
$$\{l_1 = a \wedge [l_2 = b]\}$$
$$l_2 := l_1;$$
$$\{l_1 = l_2 = a\}$$

Maximal release on traces

If $Holes \subseteq Obs$ then all fair attacks are $\vec{l} := \vec{c}$.

$$\Downarrow$$

Compute the maximal private information disclosed independently of the active attacker!

$$\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land d = 0\}$$
$$l := 0;$$
$$\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land [l = d]\}$$
$$[l := c;]$$
$$\{((h > 0 \land l = a) \lor (h \leq 0 \land a = 0)) \land [l = b]\}$$
if $(h > 0)$ **then skip else** $l := 0;$
$$\{l = a\}$$

Maximal release on traces

If $\textit{Holes} \subseteq \textit{Obs}$ then all fair attacks are $\vec{l} := \vec{c}$.

$\Downarrow$

Compute the maximal private information disclosed independently of the active attacker!

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge d = 0\}$$
$$l := 0;$$
$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge [l = d]\}$$
$$[l := c;]$$
$$\{((h > 0 \wedge l = a) \vee (h \leq 0 \wedge a = 0)) \wedge [l = b]\}$$
$$\textbf{if } (h > 0) \textbf{ then skip else } l := 0;$$
$$\{l = a\}$$

Maximal release on traces

If $Holes \subseteq Obs$ then all fair attacks are $\vec{l} := \vec{c}$.

$\Downarrow$

Compute the maximal private information disclosed independently of the active attacker!

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge d = 0\}$$

$$l := 0;$$

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge [l = d]\}$$

$$[l := c;]$$

$$\{((h > 0 \wedge l = a) \vee (h \leq 0 \wedge a = 0)) \wedge [l = b]\}$$

if $(h > 0)$ **then skip else** $l := 0;$

$$\{l = a\}$$

Maximal release on traces

If $Holes \subseteq Obs$ then all fair attacks are $\vec{l} := \vec{c}$.

$\Downarrow$

Compute the maximal private information disclosed independently of the active attacker!

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge d = 0\}$$

$$l := 0;$$

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge [l = d]\}$$

$$[l := c;]$$

$$\{((h > 0 \wedge l = a) \vee (h \leq 0 \wedge a = 0)) \wedge [l = b]\}$$

$$\textbf{if } (h > 0) \textbf{ then skip else } l := 0;$$

$$\{l = a\}$$

Maximal release on traces

If $Holes \subseteq Obs$ then all fair attacks are $\vec{l} := \vec{c}$.

$$\Downarrow$$

Compute the maximal private information disclosed independently of the active attacker!

$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge d = 0\}$$
$$l := 0;$$
$$\{((h > 0 \wedge c = a) \vee (h \leq 0 \wedge a = 0)) \wedge c = b \wedge [l = d]\}$$
$$[l := c;]$$
$$\{((h > 0 \wedge l = a) \vee (h \leq 0 \wedge a = 0)) \wedge [l = b]\}$$
$$\textbf{if } (h > 0) \textbf{ then skip else } l := 0;$$
$$\{l = a\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

The program (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$k := h;$$
$$[\bullet]$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Passive attacker (h : HH and l, k : LL)

$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[\textbf{skip}]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Passive attacker (h : HH and l, k : LL)

$$\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}$$
$$k := h;$$
$$\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}$$
$$[\textbf{skip}]$$
$$\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \land k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Passive attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[\textbf{skip}]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Passive attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[\textbf{skip}]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Unsuccessful active attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := c_1; k := c_2;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Unsuccessful active attacker (h : HH and l, k : LL)

$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := c_1; k := c_2;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Unsuccessful active attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(c_1 = 0 \wedge a = 0 \wedge b = 0) \vee (c_1 \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := c_1; k := c_2;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Successful active attacker (h : HH and l, k : LL)

$$\{(h = 0 \wedge a = 0 \wedge b = 0) \vee (h \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(k = 0 \wedge a = 0 \wedge b = 0) \vee (k \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := k;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Successful active attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(h = 0 \wedge a = 0 \wedge b = 0) \vee (h \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(k = 0 \wedge a = 0 \wedge b = 0) \vee (k \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := k;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

Enforcing robustness

Goal: Enforce robust programs independently of the attack.

Successful active attacker (h : $\mathtt{HH}$ and l, k : $\mathtt{LL}$)

$$\{(h = 0 \wedge a = 0 \wedge b = 0) \vee (h \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$k := h;$$
$$\{(k = 0 \wedge a = 0 \wedge b = 0) \vee (k \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$[l := k;]$$
$$\{(l = 0 \wedge a = 0 \wedge b = 0) \vee (l \neq 0 \wedge a = 1 \wedge b = 1)\}$$
$$\textbf{if } (l = 0) \textbf{ then } (l := 0; k := 0) \textbf{ else } (l := 1; k := 1);$$
$$\{l = a \wedge k = b\}$$

A sufficient condition

Let $P = P_2[\bullet]P_1$ be a program and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap (\mathtt{LL} \cup \mathtt{HL}) = \varnothing \Rightarrow P \text{ robust wrt unfair attacks}}$$

Example $l : \mathtt{LL}$, $h : \mathtt{HH}$ and $k : \mathtt{HL}$

$$P ::= \left[\begin{array}{l} l := h + l; [\bullet]; l := 1; k := h; \\ \textbf{while } (h > 0) \textbf{ do } (l := l - 1; l := h); \end{array}\right.$$

$$\begin{array}{rc} l, k \notin & \{(h \leq 0 \wedge a = 1) \vee (h > 0 \wedge a = 0)\} \\ & l := 1; k := h; \\ & \{(h \leq 0 \wedge l = a) \vee (h > 0 \wedge a = 0)\} \\ & \textbf{while } (h > 0) \textbf{ do } (l := l - 1; l := h); \\ & \{l = a\} \end{array}$$

A sufficient condition

Let $P = P_2[\bullet]P_1$ be a program and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap (\mathtt{LL} \cup \mathtt{HL}) = \varnothing \Rightarrow P \text{ robust wrt unfair attacks}}$$

Example $l : \mathtt{LL}$, $h : \mathtt{HH}$ and $k : \mathtt{HL}$

$$P ::= \left[\begin{array}{l} l := h + l; [\bullet]; l := 1; k := h; \\ \textbf{while } (h > 0) \textbf{ do } (l := l - 1; l := h); \end{array}\right.$$

$$l, k \notin \quad \{(h \leq 0 \wedge a = 1) \vee (h > 0 \wedge a = 0)\}$$

$$l := 1; k := h;$$

$$\{(h \leq 0 \wedge l = a) \vee (h > 0 \wedge a = 0)\}$$

$$\textbf{while } (h > 0) \textbf{ do } (l := l - 1; l := h);$$

$$\{l = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$

$$k := h_1 + h_2;$$

$$[skip;]$$

$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

$$k := h_3 \ mod\ 2;$$

$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$

$$l := h_3;$$

$$\{k = a \wedge [l = b]\}$$

$$l := k;$$

$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ mod \ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$

$$k := h_1 + h_2;$$

$$[skip;]$$

$$\{h_3 \ mod \ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

$$k := h_3 \ mod \ 2;$$

$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$

$$l := h_3;$$

$$\{k = a \wedge [l = b]\}$$

$$l := k;$$

$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ mod \ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$

$$k := h_1 + h_2;$$

$$[skip;]$$

$$\{h_3 \ mod \ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

$$k := h_3 \ mod \ 2;$$

$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$

$$l := h_3;$$

$$\{k = a \wedge [l = b]\}$$

$$l := k;$$

$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$
$$k := h_1 + h_2;$$
$$[skip;]$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$
$$k := h_3 \ \mathit{mod}\ 2;$$
$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$
$$l := h_3;$$
$$\{k = a \wedge [l = b]\}$$
$$l := k;$$
$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$

$$k := h_1 + h_2;$$

$$[skip;]$$

$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

$$k := h_3 \ mod\ 2;$$

$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$

$$l := h_3;$$

$$\{k = a \wedge [l = b]\}$$

$$l := k;$$

$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = d\}$$
$$k := h_1 + h_2;$$
$$[skip;]$$
$$\{h_3 \ mod\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$
$$k := h_3 \ mod\ 2;$$
$$\{k = a \wedge h_3 = b \wedge [l = c]\}$$
$$l := h_3;$$
$$\{k = a \wedge [l = b]\}$$
$$l := k;$$
$$\{l = k = a\}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$P ::= k := h_1 + h_2; [\bullet]; k := h_3 \ \mathit{mod}\ 2; l := h_3; l := k;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = e\}$$

$$k := h_1 + h_2;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge d = d_1 \wedge [k = e]\}$$

$$[k := d_1;]$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

Let $P = P_2[\bullet]P_1$, $\mathit{Holes} \subseteq \mathit{Obs}$ and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap \mathtt{LL} = \varnothing \Rightarrow P \text{ robust wrt fair attacks}}$$

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$P ::= k := h_1 + h_2; [\bullet]; k := h_3 \ \mathit{mod}\ 2; l := h_3; l := k;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = e\}$$
$$k := h_1 + h_2;$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge d = d_1 \wedge [k = e]\}$$
$$[k := d_1;]$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

Let $P = P_2[\bullet]P_1$, $\mathit{Holes} \subseteq \mathit{Obs}$ and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap \mathtt{LL} = \varnothing \Rightarrow P \text{ robust wrt fair attacks}}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$P ::= k := h_1 + h_2; [\bullet]; k := h_3 \ \mathit{mod}\ 2; l := h_3; l := k;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = e\}$$
$$k := h_1 + h_2;$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge d = d_1 \wedge [k = e]\}$$
$$[k := d_1;]$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

Let $P = P_2[\bullet]P_1$, $\mathit{Holes} \subseteq \mathit{Obs}$ and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap \mathrm{LL} = \varnothing \Rightarrow P \text{ robust wrt fair attacks}}$$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$P ::= k := h_1 + h_2; [\bullet]; k := h_3 \ \mathit{mod}\ 2; l := h_3; l := k;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = e\}$$
$$k := h_1 + h_2;$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge d = d_1 \wedge [k = e]\}$$
$$[k := d_1;]$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

Let $P = P_2[\bullet]P_1$, $\mathit{Holes} \subseteq \mathit{Obs}$ and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$\boxed{\mathcal{FV}(\Phi) \cap \mathrm{LL} = \varnothing \Rightarrow P \text{ robust wrt fair attacks}}$

A robustness condition on traces

Example (l : LH, k : LL and h_1, h_2, h_3 : HH)

$$P ::= k := h_1 + h_2; [\bullet]; k := h_3 \ \mathit{mod}\ 2; l := h_3; l := k;$$

$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge h_1 + h_2 = e\}$$
$$k := h_1 + h_2;$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge d = d_1 \wedge [k = e]\}$$
$$[k := d_1;]$$
$$\{h_3 \ \mathit{mod}\ 2 = a \wedge h_3 = b \wedge l = c \wedge [k = d]\}$$

Let $P = P_2[\bullet]P_1$, $\mathit{Holes} \subseteq \mathit{Obs}$ and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$.

$$\boxed{\mathcal{FV}(\Phi) \cap \mathtt{LL} = \varnothing \Rightarrow P \text{ robust wrt fair attacks}}$$

Relative Robustness

Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $Var(\mathcal{A}) \subseteq \mathtt{LL} \cup \mathtt{HL}$.

$P[\bullet]$ relatively robust

$\Updownarrow$

$\forall \vec{a} \in \mathcal{A}$, $P[\vec{a}]$ does not release more than $P[\overrightarrow{skip}]$.

Robustness wrt unfair ($\supseteq$ fair)

$P ::= l := h; [\bullet];$ with variables $h : \mathtt{HH}$, $l : \mathtt{LL}$ and $k : \mathtt{HL}$.

- $\mathit{Wlp}(l := h; [skip], \{l = a\}) = \{h = a\}$
- $\mathit{Wlp}(l := h; [l := k], \{l = a\}) = \{k = a\}$
- $\mathit{Wlp}(l := h; [l := l + k], \{l = a\}) = \{h + k = a\}$

Relative Robustness

Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $Var(\mathcal{A}) \subseteq \mathtt{LL} \cup \mathtt{HL}$.

$$P[\bullet] \text{ relatively robust}$$
$$\Updownarrow$$
$$\forall \vec{a} \in \mathcal{A},\ P[\vec{a}] \text{ does not release more than } P[\overrightarrow{skip}].$$

Robustness wrt unfair ($\supseteq$ fair)

$P ::= l := h; [\bullet];$ with variables $h : \mathtt{HH}$, $l : \mathtt{LL}$ and $k : \mathtt{HL}$.

- $Wlp\big(l := h; [skip], \{l = a\}\big) = \{h = a\}$
- $Wlp\big(l := h; [l := k], \{l = a\}\big) = \{k = a\}$
- $Wlp\big(l := h; [l := l + k], \{l = a\}\big) = \{h + k = a\}$

Relative Robustness

Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $Var(\mathcal{A}) \subseteq \mathtt{LL} \cup \mathtt{HL}$.

$$P[\bullet] \text{ relatively robust}$$
$$\Updownarrow$$
$$\forall \vec{a} \in \mathcal{A},\ P[\vec{a}] \text{ does not release more than } P[\overrightarrow{skip}].$$

Robustness wrt fair

$P ::= l := h; [\bullet];$ with variables $h : \mathtt{HH}$, $l : \mathtt{LL}$ and $k : \mathtt{HL}$.

- $Wlp\big(l := h; [skip], \{l = a\}\big) = \{h = a\}$
- $Wlp\big(l := h; [l := l + 1], \{l = a\}\big) = \{h = a - 1\}$

Relative Robustness

Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $Var(\mathcal{A}) \subseteq \mathtt{LL} \cup \mathtt{HL}$.

$$P[\bullet] \text{ relatively robust}$$
$$\Updownarrow$$
$$\forall \vec{a} \in \mathcal{A},\ P[\vec{a}] \text{ does not release more than } P[\overrightarrow{skip}].$$

Proposition

Let $P = P_2[\bullet]P_1$ be a program and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$. P is relatively robust wrt the attacks in $\mathcal{A}$ if

$$\mathcal{FV}(\Phi) \cap \mathit{Var}(\mathcal{A}) = \varnothing$$

$$\begin{array}{c} \mathcal{FV}(\Phi) \cap X = \varnothing \\ \swarrow \quad \swarrow \quad \cdots \quad \searrow \\ (\mathtt{LL} \cup \mathtt{HL}) \qquad \mathtt{LL} \qquad \cdots \qquad Var(\mathcal{A}) \end{array}$$

Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

$\Rightarrow \qquad P := P_2[\bullet]P_1 \wedge \Phi = \mathit{Wlp}(P_1, \Phi_0)$

$\Rightarrow \qquad \mathcal{V} = \{x | (x : \mathtt{LL} \vee x : \mathtt{HL}) \wedge x \notin \mathcal{FV}(\Phi)\}$

$$\Downarrow$$

P is relatively robust wrt $\{a | \mathit{Var}(a) \subseteq \mathcal{V}\}$

Example $h_1, h_2 : \mathtt{HH}$, $l_1, l_3 : \mathtt{LL}$ and $l_2 : \mathtt{HL}$

$$P_1$$

$$[\bullet]$$

$$\{(h_2 > 0 \wedge h_1 \ \mathit{mod}\ 2 = a \wedge b = 0) \vee (h_2 \leq 0 \wedge l_2 = b = a)\}$$

if $(h_2 > 0)$ **then** $l_1 := h_1 \ \mathit{mod}\ 2; l_3 := 0$ **else** $l_3 := l_2; l_1 := l_3$

$$\{l_1 = a \wedge l_3 = b\}$$

P is relatively robust wrt attacks on l_1 and l_3!

Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

$\Rightarrow \qquad P := P_2[\bullet]P_1 \wedge \Phi = \mathit{Wlp}(P_1, \Phi_0)$

$\Rightarrow \qquad \mathcal{V} = \{x | (x : \mathtt{LL} \vee x : \mathtt{HL}) \wedge x \notin \mathcal{FV}(\Phi)\}$

$$\Downarrow$$

P is relatively robust wrt $\{a | \mathit{Var}(a) \subseteq \mathcal{V}\}$

Example $h_1, h_2 : \mathtt{HH}$, $l_1, l_3 : \mathtt{LL}$ and $l_2 : \mathtt{HL}$

$$P_1$$

$$[\bullet]$$

$$\{(h_2 > 0 \wedge h_1 \;\mathit{mod}\; 2 = a \wedge b = 0) \vee (h_2 \leq 0 \wedge l_2 = b = a)\}$$

$$\textbf{if } (h_2 > 0) \textbf{ then } l_1 := h_1 \;\mathit{mod}\; 2; l_3 := 0 \textbf{ else } l_3 := l_2; l_1 := l_3$$

$$\{l_1 = a \wedge l_3 = b\}$$

P is relatively robust wrt attacks on l_1 and l_3!

Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

$\Rightarrow \quad P := P_2[\bullet]P_1 \wedge \Phi = \mathit{Wlp}(P_1, \Phi_0)$

$\Rightarrow \quad \mathcal{V} = \{x | (x : \mathtt{LL} \vee x : \mathtt{HL}) \wedge x \notin \mathcal{FV}(\Phi)\}$

$$\Downarrow$$

P is relatively robust wrt $\{a | \mathit{Var}(a) \subseteq \mathcal{V}\}$

Example $h_1, h_2 : \mathtt{HH}$, $l_1, l_3 : \mathtt{LL}$ and $l_2 : \mathtt{HL}$

$$P_1$$

$$[\bullet]$$

$$\{(h_2 > 0 \wedge h_1 \ \mathit{mod}\ 2 = a \wedge b = 0) \vee (h_2 \leq 0 \wedge l_2 = b = a)\}$$

$$\textbf{if } (h_2 > 0) \textbf{ then } l_1 := h_1 \ \mathit{mod}\ 2; l_3 := 0 \textbf{ else } l_3 := l_2; l_1 := l_3$$

$$\{l_1 = a \wedge l_3 = b\}$$

P is relatively robust wrt attacks on l_1 and l_3!

Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

$\Rightarrow \qquad P := P_2[\bullet]P_1 \wedge \Phi = \mathit{Wlp}(P_1, \Phi_0)$

$\Rightarrow \qquad \mathcal{V} = \{x | (x : \mathtt{LL} \vee x : \mathtt{HL}) \wedge x \notin \mathcal{FV}(\Phi)\}$

$$\Downarrow$$

P is relatively robust wrt $\{a | \mathit{Var}(a) \subseteq \mathcal{V}\}$

Example $h_1, h_2 : \mathtt{HH}$, $l_1, l_3 : \mathtt{LL}$ and $l_2 : \mathtt{HL}$

$$P_1$$

$$[\bullet]$$

$$\{(h_2 > 0 \wedge h_1 \ \mathit{mod}\ 2 = a \wedge b = 0) \vee (h_2 \leq 0 \wedge l_2 = b = a)\}$$

$$\textbf{if } (h_2 > 0) \textbf{ then } l_1 := h_1 \ \mathit{mod}\ 2; l_3 := 0 \textbf{ else } l_3 := l_2; l_1 := l_3$$

$$\{l_1 = a \wedge l_3 = b\}$$

P is relatively robust wrt attacks on l_1 and l_3!

Relative VS Decentralized Robustness

Decentralized Robustness: Principals distrusting each other.

- Analysis and Attacker: Fixes which data principal p believes the attacker q can read or write.
- Robustness: Must hold for all pairs p, q with power $\langle R_{p\to q}, W_{p\leftarrow q}\rangle$

Relative Robustness: Fixed principals p and q.

- Static confidentiality levels $\mathcal{C}_{p\to q}$ and integrity levels $\mathcal{I}_{p\leftarrow q}$.
- If $\mathcal{I}_{p\leftarrow q}(x) = \mathtt{L}$ p believes that q can modify x.

$P = P_2[\bullet]P_1$ be a program and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$. P satisfies decentralized robustness wrt the principals p, q if we have that

$$\mathcal{FV}(\Phi) \cap (\mathrm{LL} \cup \mathrm{HL})_{p\to q} = \varnothing$$

where $(\mathrm{LL} \cup \mathrm{HL})_{p\to q} \stackrel{\text{def}}{=} \left\{\, x \;\middle|\; \mathcal{I}_{p\to q}(x) = \mathtt{L} \,\right\}$

Relative VS Decentralized Robustness

Decentralized Robustness: Principals distrusting each other.

- Analysis and Attacker: Fixes which data principal p believes the attacker q can read or write.
- Robustness: Must hold for all pairs p, q with power $\langle R_{p \to q}, W_{p \leftarrow q} \rangle$

Relative Robustness: Fixed principals p and q.

- Static confidentiality levels $\mathcal{C}_{p \to q}$ and integrity levels $\mathcal{I}_{p \leftarrow q}$.
- If $\mathcal{I}_{p \leftarrow q}(x) = \mathtt{L}$ p believes that q can modify x.

$P = P_2[\bullet]P_1$ be a program and $\Phi = \mathit{Wlp}(P_1, \Phi_0)$. P satisfies decentralized robustness wrt the principals p, q if we have that

$$\mathcal{FV}(\Phi) \cap (\mathtt{LL} \cup \mathtt{HL})_{p \to q} = \varnothing$$

where $(\mathtt{LL} \cup \mathtt{HL})_{p \to q} \stackrel{\text{def}}{=} \left\{ x \,\middle|\, \mathcal{I}_{p \to q}(x) = \mathtt{L} \right\}$

Conclusions:

Conclusions

- Robustness in language-based security.
- Maximal information released by active attackers.
- Condition to check robust programs.
- Considerations for both I/O and trace semantics.

Future work

- An algorithm for static certification of robust programs.
- Extend this work to deal with abstract active attackers.
- Extend this work to concurrent attackers or other attacker models.
- Relation between relative robustness and decentralized robustness.

Conclusions:

Conclusions

- Robustness in language-based security.
- Maximal information released by active attackers.
- Condition to check robust programs.
- Considerations for both I/O and trace semantics.

Future work

- An algorithm for static certification of robust programs.
- Extend this work to deal with abstract active attackers.
- Extend this work to concurrent attackers or other attacker models.
- Relation between relative robustness and decentralized robustness.

THANK YOU!

Relative robustness dependent on the attack

Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $|\mathcal{A}| \lnsim \omega$.

$$\Downarrow$$

- Compute the maximal information disclosed for all attacks.
  - $\Rightarrow$ Requires a finite number of tests.
- Compare with the passive attacker.
  - $\Rightarrow$ Check robustness in a finite number of tests.

Example: Holes inside conditionals or loops

Consider the program P

$$P ::= \left[\begin{array}{ll} k := h \ mod\ 3; & \\ \textbf{if } (h \ mod\ 2 = 0) & \textbf{then}[\bullet]; l := 0; k := l \\ & \textbf{else } l := 1; \end{array} \right.$$

where $h : \texttt{HH}$, $l : \texttt{LL}$ and $k : \texttt{LL}$.

$$\left\{ \begin{array}{c} (h \ mod\ 2 = 0 \wedge a = 0 \wedge b = 0) \vee \\ (h \ mod\ 2 \neq 0 \wedge a = 1 \wedge k = b) \end{array} \right\}$$

$$\textbf{if } (h \ mod\ 2 = 0) \textbf{ then } [\bullet]; l := 0; k := l \textbf{ else } l := 1;$$

$$\{l = a \wedge k = b\}$$