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. ($H \not\Rightarrow L$)

\[
\forall l \in V^L, \forall h_1, h_2 \in V^H. [P](h_1, l)^L = [P](h_2, l)^L
\]

**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. \((H \not
leftrightarrow L)\)

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

PROBLEM

↓

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. ($H \not\leftrightarrow L$)

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

PROBLEM
↓
Real systems release private information intentionally.
Security Background

**Goal:** Protect data confidentiality from malicious attackers.

**Solution**

\[ \phi(H) : \text{declassified private property (} \phi(H) \rightsquigarrow L \) \]

\[
\forall l \in V^L, \forall h_1, h_2 \in V^H.
\phi(h_1) = \phi(h_2) \Rightarrow [P](h_1, l)^L = [P](h_2, l)^L
\]

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

[Myers and Liskov 1997, Sabelfeld and Myers 2003]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

**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)

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

- **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.
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)

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

- 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: A general theory of sound approximation of program semantics.

Let $\text{sum}(x, y) \overset{\text{def}}{=} x + y$.

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

Musard Balliu, Isabella Mastroeni

A weakest precondition approach to active attacks analysis
Abstract Interpretation: A general theory of sound approximation of program semantics.

Musard Balliu, Isabella Mastroeni

\(\text{sum}(x, y) \overset{\text{def}}{=} x + y\) 

- \(\text{sum}^*(+, +) = +\)
- \(\text{sum}^*(-, -) = -\)
- \(\text{sum}^*(+, -) = \top\)

- \(\text{sum}^*(\text{even}, \text{even}) = \text{even}\)
- \(\text{sum}^*(\text{odd}, \text{odd}) = \text{even}\)
- \(\text{sum}^*(\text{even}, \text{odd}) = \text{odd}\)
Declassification by $Wlp$ [Banerjee et al. 2007]

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

$$P \overset{\text{def}}{=} \begin{cases} \text{if } (h_1 = h_2) \text{ then } l := 0; & \text{else } l := 1; \\ Wlp(P, l = a) = (h_1 = h_2 \land a = 0) \lor (h_1 \neq h_2 \land a = 1) \end{cases}$$

\[\Downarrow\]
Maximal information released

A weakest precondition approach to active attacks analysis
Wlp:
Greatest set of input states leading to a given output observation.

\[
\{\langle h_1, h_2, l \rangle | h_1 \neq h_2 \} \cup \{\langle h_1, h_2, l \rangle | h_1 = h_2 \} \cup \emptyset
\]

From non-interference point of view

\[
h_1 = 0, h_2 = 0, l = 0 \implies l = 0
\]

\[
h_1 = 1, h_2 = 0, l = 0 \implies l = 1
\]
Maximal release by active attackers

Goal:
Compute the maximal information disclosed by active attackers.

⇒ Unfair attacks: Programs on LL and HL variables.

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

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

- Active attackers ⇒ Semantic transformation.
- Different attacks ⇒ Different information release.

Active attacks can be potentially infinite!
Goal:
Compute the maximal information disclosed by active attackers.

⇒ Unfair attacks: Programs on LL and HL variables.

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

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

- Active attackers ⇒ Semantic transformation.
- Different attacks ⇒ Different information release.

Active attacks can be potentially infinite!
Goal:
Compute the maximal information disclosed by active attackers.

⇒ Unfair attacks: Programs on LL and HL variables.

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

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

- Active attackers ⇒ Semantic transformation.
- Different attacks ⇒ Different information release.

Active attacks can be potentially infinite!
Active attack $\equiv$ function on $\text{LL}$ and $\text{HL}$ variables.

- Extend the $\text{Wlp}$ 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 $\langle l, k \rangle := \langle f(l, k), g(l, k) \rangle$.

\[
\begin{align*}
\{f(h, k) = a\} \\
l := h; \\
\{f(l, k) = a\} \\
\langle l, k \rangle := \langle f(l, k), g(l, k) \rangle; \\
\{l = a\}
\end{align*}
\]

$\Rightarrow \{f(h, k) = a\}$: $f$ “measures” the information of $h$ and $k$. 
Active attack ≡ function on LL and HL variables.

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

Back to the example

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

\[
\begin{align*}
\{ f(h, k) &= a \} \\
\quad & l := h; \\
\{ f(l, k) &= a \} \\
\text{[} & \langle l, k \rangle := \langle f(l, k), g(l, k) \rangle;] \\
\{ l &= a \}
\end{align*}
\]

$\Rightarrow \{ f(h, k) = a \}$: $f$ “measures” the information of $h$ and $k$. 
I/O Analysis

A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

\begin{align*}
\{ h_2 \text{ mod } 2 = a \} \\
&\begin{aligned}
h_1 &:= h_2; \\
\{ h_2 \text{ mod } 2 = a \} \\
&\begin{aligned}
h_2 &:= h_2 \text{ 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 \}
\end{aligned}
\end{aligned}
\end{align*}
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

\[ \begin{align*}
\{ h_2 \mod 2 = a \} \\
h_1 & := h_2; \\
\{ h_2 \mod 2 = a \} \\
h_2 & := h_2 \mod 2; \\
\{ h_2 = a \} \\
l_1 & := h_2; \\
\{ l_1 = a \} \\
h_2 & := h_1; \\
\{ l_1 = a \} \\
l_2 & := h_2; \\
\{ l_1 = a \} \\
l_2 & := l_1; \\
\{ l_1 = l_2 = a \}
\end{align*} \]
I/O Analysis

A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

\[
\begin{align*}
\{h_2 \mod 2 = a\} \\
& h_1 := h_2; \\
\{h_2 \mod 2 = a\} \\
& h_2 := h_2 \mod 2; \\
& \{h_2 = a\} \\
& l_1 := h_2; \\
& \{l_1 = a\} \\
& h_2 := h_1; \\
& \{l_1 = a\} \\
& l_2 := h_2; \\
& \{l_1 = a\} \\
& l_2 := l_1; \\
& \{l_1 = l_2 = a\}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

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 \}
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

I/O Analysis

\[
\begin{align*}
\{ h_2 \mod 2 = a \} \\
& h_1 := h_2; \\
\{ h_2 \mod 2 = a \} \\
& h_2 := h_2 \mod 2; \\
& \{ h_2 = a \} \\
& l_1 := h_2; \\
& \{ l_1 = a \} \\
& h_2 := h_1; \\
& \{ l_1 = a \} \\
& l_2 := h_2; \\
& \{ l_1 = a \} \\
& l_2 := l_1; \\
& \{ l_1 = l_2 = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

\[
\begin{align*}
\{ h_2 \mod 2 = a \} \\
h_1 &:= h_2; \\
\{ h_2 \mod 2 = a \} \\
h_2 &:= h_2 \mod 2; \\
\{ h_2 = a \} \\
l_1 &:= h_2; \\
\{ l_1 = a \} \\
h_2 &:= h_1; \\
\{ l_1 = a \} \\
l_2 &:= h_2; \\
\{ l_1 = a \} \\
l_2 &:= l_1; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
I/O Analysis

A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

\[
\begin{align*}
\{ h_2 \ mod \ 2 &= a \} \\
& h_1 := h_2; \\
\{ h_2 \ mod \ 2 &= a \} \\
& h_2 := h_2 \ mod \ 2; \\
& \{ h_2 = a \} \\
& l_1 := h_2; \\
& \{ l_1 = a \} \\
& h_2 := h_1; \\
& \{ l_1 = a \} \\
& l_2 := h_2; \\
& \{ l_1 = a \} \\
& l_2 := l_1; \\
& \{ l_1 = l_2 = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
\quad h_1 := h_2; \\
\{ h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
\quad h_2 := h_2 \mod 2; \\
\{ h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
\quad l_1 := h_2; \\
\{ l_1 = a \land h_1 = b \land l_2 = c \} \\
\quad h_2 := h_1; \\
\{ l_1 = a \land h_2 = b \land [l_2 = c] \} \\
\quad l_2 := h_2; \\
\{ l_1 = a \land [l_2 = b] \} \\
\quad l_2 := l_1; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
   & h_1 := h_2 ; \\
\{ h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
   & h_2 := h_2 \mod 2 ; \\
\{ h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
   & l_1 := h_2 ; \\
\{ l_1 = a \land h_1 = b \land l_2 = c \} \\
   & h_2 := h_1 ; \\
\{ l_1 = a \land h_2 = b \land [l_2 = c] \} \\
   & l_2 := h_2 ; \\
\{ l_1 = a \land [l_2 = b] \} \\
   & l_2 := l_1 ; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
h_1 &:= h_2; \\
\{ h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
h_2 &:= h_2 \mod 2; \\
\{ h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
l_1 &:= h_2; \\
\{ l_1 = a \land h_1 = b \land l_2 = c \} \\
h_2 &:= h_1; \\
\{ l_1 = a \land h_2 = b \land [l_2 = c] \} \\
l_2 &:= h_2; \\
\{ l_1 = a \land [l_2 = b] \} \\
l_2 &:= l_1; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

Trace Analysis [Mastroeni and Banerjee 2008]

\[\{h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d\}\]

\[h_1 := h_2;\]

\[\{h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d\}\]

\[h_2 := h_2 \mod 2;\]

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

\[l_1 := h_2;\]

\[\{l_1 = a \land h_1 = b \land l_2 = c\}\]

\[h_2 := h_1;\]

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

\[l_2 := h_2;\]

\[\{l_1 = a \land [l_2 = b]\}\]

\[l_2 := l_1;\]

\[\{l_1 = l_2 = a\}\]
Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ & h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
& h_1 := h_2; \\
\{ & h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
& h_2 := h_2 \mod 2; \\
\{ & h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
& l_1 := h_2; \\
\{ & l_1 = a \land h_1 = b \land l_2 = c \} \\
& h_2 := h_1; \\
\{ & l_1 = a \land h_2 = b \land [l_2 = c] \} \\
& l_2 := h_2; \\
\{ & l_1 = a \land [l_2 = b] \} \\
& l_2 := l_1; \\
\{ & l_1 = l_2 = a \} 
\end{align*}
\]

Musard Balliu, Isabella Mastroeni

A weakest precondition approach to active attacks analysis
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ h_2 \mod 2 = a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
& h_1 := h_2; \\
\{ h_2 \mod 2 = a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
& h_2 := h_2 \mod 2; \\
\{ h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
& l_1 := h_2; \\
\{ l_1 = a \land h_1 = b \land l_2 = c \} \\
& h_2 := h_1; \\
\{ l_1 = a \land h_2 = b \land [l_2 = c] \} \\
& l_2 := h_2; \\
\{ l_1 = a \land [l_2 = b] \} \\
& l_2 := l_1; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

Trace Analysis [Mastroeni and Banerjee 2008]

\[
\begin{align*}
\{ h_2 \mod 2 &= a \land h_2 = b \land l_2 = c \land l_1 = d \} \\
h_1 &:= h_2; \\
\{ h_2 \mod 2 &= a \land h_1 = b \land l_2 = c \land l_1 = d \} \\
h_2 &:= h_2 \mod 2; \\
\{ h_2 = a \land h_1 = b \land l_2 = c \land [l_1 = d] \} \\
l_1 &:= h_2; \\
\{ l_1 = a \land h_1 = b \land l_2 = c \} \\
h_2 &:= h_1; \\
\{ l_1 = a \land h_2 = b \land [l_2 = c] \} \\
l_2 &:= h_2; \\
\{ l_1 = a \land [l_2 = b] \} \\
l_2 &:= l_1; \\
\{ l_1 = l_2 = a \}
\end{align*}
\]
Maximal release on traces

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

⇓

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

\[
\{(h > 0 \land c = a) \lor (h \leq 0 \land a = 0) \land [l = d]\}
\]
\[
l := 0;
\]
\[
\{(h > 0 \land c = a) \lor (h \leq 0 \land a = 0) \land [l = d]\}
\]
\[
[l := c;]
\]
\[
\{(h > 0 \land l = a) \lor (h \leq 0 \land a = 0) \land [l = b]\}
\]
\[
\text{if } (h > 0) \text{ then skip else } l := 0;
\]
\[
\{l = a\} \]
Maximal release on traces

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

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

\[
\begin{align*}
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land d = 0\} \\
\quad l := 0; \\
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land [l = d]\} \\
\quad [l := c]; \\
\{((h > 0 \land l = a) \lor (h \leq 0 \land a = 0)) \land [l = b]\} \\
\quad \textbf{if} (h > 0) \textbf{ then skip else } l := 0; \\
\{l = a\}
\end{align*}
\]
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!

\[
\begin{align*}
\{ & ((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land d = 0 \\
& l := 0; \\
& (((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land [l = d])
\end{align*}
\]

\[
\begin{align*}
& [l := c]; \\
& \{ ((h > 0 \land l = a) \lor (h \leq 0 \land a = 0)) \land [l = b] \\
& \textbf{if } (h > 0) \text{ then skip else } l := 0; \\
& \{ l = a \}
\end{align*}
\]
If $\text{Holes} \subseteq \text{Obs}$ then all fair attacks are $\vec{l} := \vec{c}$.

$\downarrow$

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

\[
\begin{align*}
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land d = 0\} \\
\quad l := 0; \\
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land \lbrack l = d \rbrack\} \\
\quad [l := c;] \\
\{((h > 0 \land l = a) \lor (h \leq 0 \land a = 0)) \land \lbrack l = b \rbrack\} \\
\text{if } (h > 0) \text{ then skip else } l := 0; \\
\{l = a\}
\end{align*}
\]
If $\text{Holes} \subseteq \text{Obs}$ then all fair attacks are $\vec{l} := \vec{c}$.

\[ \downarrow \]

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

\[
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land d = 0\}
\]
\[ l := 0; \]
\[
\{((h > 0 \land c = a) \lor (h \leq 0 \land a = 0)) \land c = b \land [l = d]\}
\]
\[ [l := c;] \]
\[
\{((h > 0 \land l = a) \lor (h \leq 0 \land a = 0)) \land [l = b]\}
\]
\[ \text{if } (h > 0) \text{ then skip else } l := 0; \]
\[
\{l = a\} \]
Goal: Enforce robust programs independently of the attack.

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

\[
k := h; \\
[\bullet] \\
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1);
\]
**Goal:** Enforce robust programs independently of the attack.

**Passive attacker** \((h : HH \text{ and } l, k : LL)\)

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

\[
k := h;
\]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

\[\text{[skip]}\]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

\[
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1);
\]

\[
\{(l = a \land k = b)\}
\]
Goal: Enforce robust programs independently of the attack.

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

\[
\begin{align*}
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\} \\
\quad k := h; \{\text{skip}\} \\
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\} \\
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1); \\
\quad \{l = a \land k = b\}
\end{align*}
\]
Enforcing robustness

Goal: Enforce robust programs independently of the attack.

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

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

\[k := h;\]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

[skip]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

if \((l = 0)\) then \((l := 0; k := 0)\) else \((l := 1; k := 1)\);

\[
\{l = a \land k = b\} \]
Enforcing robustness

**Goal:** Enforce robust programs independently of the attack.

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

\[
\begin{align*}
&\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\} \\
&k := h; \\
&\{l = 0 \land a = 0 \land b = 0\} \lor (l \neq 0 \land a = 1 \land b = 1)\} \\
&[skip] \\
&(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\} \\
&\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1); \\
&\{l = a \land k = b\}
\end{align*}
\]
Goal: Enforce robust programs independently of the attack.

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

\[
\{(c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1)\}
\]

\[
k := h;
\]

\[
\{(c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1)\}
\]

\[
[l := c_1; k := c_2;]
\]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

\[
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1);
\]

\[
\{l = a \land k = b\}
Enforcing robustness

**Goal:** Enforce robust programs independently of the attack.

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

\[
\begin{align*}
\{(c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1)\} & \quad k := h; \\
\{(c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1)\} & \quad [l := c_1; k := c_2;] \\
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\} & \quad \text{if } (l = 0) \text{ then } (l := 0; k := 0) \quad \text{else } (l := 1; k := 1); \\
\{l = a \land k = b\}
\end{align*}
\]
Goal: Enforce robust programs independently of the attack.

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

\[
\{ (c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1) \}
\]
\[
k := h;
\]
\[
\{ (c_1 = 0 \land a = 0 \land b = 0) \lor (c_1 \neq 0 \land a = 1 \land b = 1) \}
\]
\[
[l := c_1; k := c_2;]
\]
\[
\{ (l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1) \}
\]
\[
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1);
\]
\[
\{ l = a \land k = b \} \]
Enforcing robustness

**Goal:** Enforce robust programs independently of the attack.

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

\[
\begin{align*}
\{ (h = 0 \land a = 0 \land b = 0) \lor (h \neq 0 \land a = 1 \land b = 1) \} & \\
k := h; & \\
\{ (k = 0 \land a = 0 \land b = 0) \lor (k \neq 0 \land a = 1 \land b = 1) \} & \\
[l := k;] & \\
\{ (l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1) \} & \\
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1); & \\
\{ l = a \land k = b \}
\end{align*}
\]
Goal: Enforce robust programs independently of the attack.

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

\[
\{(h = 0 \land a = 0 \land b = 0) \lor (h \neq 0 \land a = 1 \land b = 1)\}
\]

\[
k := h;
\]

\[
\{(k = 0 \land a = 0 \land b = 0) \lor (k \neq 0 \land a = 1 \land b = 1)\}
\]

\[
[l := k;]
\]

\[
\{(l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1)\}
\]

if \((l = 0)\) then \((l := 0; k := 0)\) else \((l := 1; k := 1)\);

\[
\{l = a \land k = b\}
**Goal:** Enforce robust programs independently of the attack.

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

\[
\begin{align*}
\{ & (h = 0 \land a = 0 \land b = 0) \lor (h \neq 0 \land a = 1 \land b = 1) \\
& k := h; \\
& (k = 0 \land a = 0 \land b = 0) \lor (k \neq 0 \land a = 1 \land b = 1) \\
& [l := k;] \\
& (l = 0 \land a = 0 \land b = 0) \lor (l \neq 0 \land a = 1 \land b = 1) \\
\text{if } (l = 0) \text{ then } (l := 0; k := 0) \text{ else } (l := 1; k := 1); \\
& \{l = a \land k = b\}
\end{align*}
\]
Let $P = P_2[\bullet]P_1$ be a program and $\Phi = Wlp(P_1, \Phi_0)$.

$$\mathcal{F}\mathcal{V}(\Phi) \cap (LL \cup HL) = \emptyset \Rightarrow P \text{ robust wrt unfair attacks}$$

**Example** $l : LL$, $h : HH$ and $k : HL$

$$P ::= \begin{cases} l := h + l; [\bullet]; l := 1; k := h; \\
\text{while } (h > 0) \text{ do } (l := l - 1; l := h); \end{cases}$$

$$l, k \notin \{(h \leq 0 \land a = 1) \lor (h > 0 \land a = 0)\}$$

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

$$\{(h \leq 0 \land l = a) \lor (h > 0 \land a = 0)\}$$

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

$$\{l = a\}$$
A sufficient condition

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

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

Example $l : LL$, $h : HH$ and $k : HL$

$$P ::= \begin{cases} l := h + l; [\bullet]; l := 1; k := h; \\
\text{while} \ (h > 0) \ \text{do} \ (l := l - 1; l := h); \end{cases}$$

$$l, k \notin \{(h \leq 0 \land a = 1) \lor (h > 0 \land a = 0)\}$$

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

$$\{(h \leq 0 \land l = a) \lor (h > 0 \land a = 0)\}$$

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

$$\{l = a\}$$
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

\[
\begin{align*}
\{ & h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d \} \\
& k := h_1 + h_2; \\
& \{ \text{ skip; } \} \\
\{ & h_3 \mod 2 = a \land h_3 = b \land l = c \land [k = d] \} \\
& k := h_3 \mod 2; \\
& \{ k = a \land h_3 = b \land [l = c] \} \\
& l := h_3; \\
& \{ k = a \land [l = b] \} \\
& l := k; \\
& \{ l = k = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

\[
\{h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d\} \\
\begin{align*}
  k & := h_1 + h_2; \\
  \text{[skip;]} \\
\end{align*}
\]

\[
\{h_3 \mod 2 = a \land h_3 = b \land l = c \land [k = d]\} \\
\begin{align*}
  k & := h_3 \mod 2; \\
\end{align*}
\]

\[
\{k = a \land h_3 = b \land [l = c]\} \\
\begin{align*}
  l & := h_3; \\
\end{align*}
\]

\[
\{k = a \land [l = b]\} \\
\begin{align*}
  l & := k; \\
\end{align*}
\]

\[
\{l = k = a\} \\
\begin{align*}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

\[
\begin{align*}
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d \} \\
\quad k := h_1 + h_2; \\
\quad [skip;] \\
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land [k = d] \} \\
\quad k := h_3 \mod 2; \\
\quad \{ k = a \land h_3 = b \land [l = c] \} \\
\quad l := h_3; \\
\quad \{ k = a \land [l = b] \} \\
\quad l := k; \\
\quad \{ l = k = a \}
\end{align*}
\]
### Example ($l : \text{LH}$, $k : \text{LL}$ and $h_1, h_2, h_3 : \text{HH}$)

\[
\begin{align*}
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d \} \\
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land [k = d] \} \\
\end{align*}
\]

\[
\begin{align*}
& k := h_1 + h_2; \\
& [\text{skip;}]; \\
& k := h_3 \mod 2; \\
& \{ k = a \land h_3 = b \land [l = c] \} \\
& l := h_3; \\
& \{ k = a \land [l = b] \} \\
& l := k; \\
& \{ l = k = a \}
\end{align*}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

\[
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d \}
\]

\[
k := h_1 + h_2;
\]

\[
[\text{skip;}]\]

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

\[
k := h_3 \mod 2;
\]

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

\[
l := h_3;
\]

\[
\{ k = a \land [l = b] \}
\]

\[
l := k;
\]

\[
\{ l = k = a \} \]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

\begin{align*}
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = d \} \\
& \quad \vdash k := h_1 + h_2; \\
& \quad \left[ \text{skip}; \right] \\
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land [k = d] \} \\
& \quad \vdash k := h_3 \mod 2; \\
\{ k = a \land h_3 = b \land [l = c] \} \\
& \quad \vdash l := h_3; \\
\{ k = a \land [l = b] \} \\
& \quad \vdash l := k; \\
\{ l = k = a \} 
\end{align*}
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

**A robustness condition on traces**

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

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

\[
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = e \}
\]

\[
k := h_1 + h_2;
\]

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

\[
[k := d_1;]
\]

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

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

\[
\mathcal{FV}(\Phi) \cap LL = \emptyset \Rightarrow P \text{ robust wrt fair attacks}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

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

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

\[
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = e \}
\]

\[
k := h_1 + h_2;
\]

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

\[
[k := d_1;]
\]

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

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

\[
\mathcal{FV}(\Phi) \cap LL = \emptyset \Rightarrow P \text{ robust wrt fair attacks}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

Example \((l : \text{LH}, \ k : \text{LL} \text{ and } h_1, h_2, h_3 : \text{HH})\)

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

\[
\{ h_3 \mod 2 = a \land h_3 = b \land l = c \land h_1 + h_2 = e \}
\]

\[
k := h_1 + h_2;
\]

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

\[
[k := d_1;]
\]

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

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

\[
\mathcal{FV}(\Phi) \cap \text{LL} = \emptyset \Rightarrow P \text{ robust wrt fair attacks}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

Example (l : LH, k : LL and h₁, h₂, h₃ : HH)

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

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

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

\[
\mathcal{FV}(\Phi) \cap LL = \emptyset \Rightarrow P \text{ robust wrt fair attacks}
\]
A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

A robustness condition on traces

Example \((l : \text{LH}, k : \text{LL} \text{ and } h_1, h_2, h_3 : \text{HH})\)

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

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

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

\[
\mathcal{FV}(\Phi) \cap \text{LL} = \emptyset \Rightarrow P \text{ robust wrt \textit{fair} attacks}
\]
Relative Robustness

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

$P[\bullet]$ relatively robust

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

Robustness wrt unfair ($\supseteq$ fair)

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

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

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

$P[\bullet]$ relatively robust

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

Robustness wrt unfair ($\supseteq$ fair)

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

- $Wlp(l := h; [\text{skip}], \{l = a\}) \models \{h = a\}$
- $Wlp(l := h; [l := k], \{l = a\}) \models \{k = a\}$
- $Wlp(l := h; [l := l + k], \{l = a\}) \models \{h + k = a\}$
Let $P[\bullet]$ be a program and $A$ a set of attacks so that $Var(A) \subseteq LL \cup HL$.

$P[\bullet]$ relatively robust

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

**Robustness wrt fair**

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

- $Wlp(l := h; [\overrightarrow{skip}], \{l = a\}) \models \{h = a\}$
- $Wlp(l := h; [l := l + 1], \{l = a\}) \models \{h = a - 1\}$
Let $P[\bullet]$ be a program and $\mathcal{A}$ a set of attacks so that $\text{Var}(\mathcal{A}) \subseteq \text{LL} \cup \text{HL}$.

$P[\bullet]$ relatively robust

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

**Proposition**

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

$$\mathcal{FV}(\Phi) \cap \text{Var}(\mathcal{A}) = \emptyset$$

$$\mathcal{FV}(\Phi) \cap X = \emptyset$$

(LL $\cup$ HL) LL $\cdots$ Var($\mathcal{A}$)
Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

⇒ $P := P_2[\bullet]P_1 \land \Phi = Wlp(P_1, \Phi_0)$
⇒ $\mathcal{V} = \{x|(x:LL \lor x:HL) \land x \notin \mathcal{FV}(\Phi)\}$

⇓

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

Example $h_1, h_2 : HH$, $l_1, l_3 : LL$ and $l_2 : HL$

<table>
<thead>
<tr>
<th>$P_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[\bullet]$</td>
</tr>
<tr>
<td>{$(h_2 &gt; 0 \land h_1 \mod 2 = a \land b = 0) \lor (h_2 \leq 0 \land l_2 = b = a)$}</td>
</tr>
</tbody>
</table>

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

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

$P$ is relatively robust wrt attacks on $l_1$ and $l_3$!
Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

$\Rightarrow P := P_2[\bullet]P_1 \land \Phi = WlP(P_1, \Phi_0)$

$\Rightarrow \mathcal{V} = \{x|(x : LL \lor x : HL) \land x \not\in \mathcal{FV}(\Phi)\}$

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

Example $h_1, h_2 : HH, l_1, l_3 : LL$ and $l_2 : HL$

$P_1$

$[\bullet]$

\{(h_2 > 0 \land h_1 \mod 2 = a \land b = 0) \lor (h_2 \leq 0 \land l_2 = b = a)\}$

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

\{l_1 = a \land 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.

\[ \Rightarrow P := P_2[\bullet] P_1 \land \Phi = WlP(P_1, \Phi_0) \]
\[ \Rightarrow V = \{ x \mid (x : LL \lor x : HL) \land x \not\in FV(\Phi) \} \]
\[ \Downarrow \]

\[ P \text{ is relatively robust wrt } \{ a \mid \text{Var}(a) \subseteq V \} \]

Example \( h_1, h_2 : HH, l_1, l_3 : LL \text{ and } l_2 : HL \)

\[
\begin{align*}
P_1[\bullet] \\
\{ (h_2 > 0 \land h_1 \mod 2 = a \land b = 0) \lor (h_2 \leq 0 \land l_2 = b = a) \} \\
\text{if } (h_2 > 0) \text{ then } l_1 := h_1 \mod 2; l_3 := 0 \text{ else } l_3 := l_2; l_1 := l_3 \\
\{ l_1 = a \land l_3 = b \}
\end{align*}
\]

\[ P \text{ is relatively robust wrt attacks on } l_1 \text{ and } l_3! \]
Certifying (relative) robustness

Express the sufficient condition in the opposite direction.

\[ P := P_2[\bullet] P_1 \land \Phi = Wlp(P_1, \Phi_0) \]

\[ V = \{ x \ | (x : LL \lor x : HL) \land x \notin \mathcal{FV}(\Phi) \} \]

\[ \Downarrow \]

\[ P \text{ is relatively robust wrt } \{ a \ | \ Var(a) \subseteq V \} \]

Example \( h_1, h_2 : HH, l_1, l_3 : LL \) and \( l_2 : HL \)

\[
\begin{align*}
P_1[\bullet]
\{(h_2 > 0 \land h_1 \mod 2 = a \land b = 0) \lor (h_2 \leq 0 \land l_2 = b = a)\}
\text{if } (h_2 > 0) \text{ then } l_1 := h_1 \mod 2; l_3 := 0 \text{ else } l_3 := l_2; l_1 := l_3
\end{align*}
\]

\[ \{l_1 = a \land l_3 = b\} \]

\[ P \text{ is relatively robust wrt attacks on } l_1 \text{ and } l_3! \]
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 $\langle R_{p \rightarrow q}, W_{p \leftarrow q} \rangle$

**Relative Robustness:** Fixed principals $p$ and $q$.
- Static confidentiality levels $C_{p \rightarrow q}$ and integrity levels $I_{p \leftarrow q}$.
- If $I_{p \leftarrow q}(x) = L$, $p$ believes that $q$ can modify $x$.

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

$$\mathcal{FV}(\Phi) \cap (\text{LL} \cup \text{HL})_{p \rightarrow q} = \emptyset$$

where $(\text{LL} \cup \text{HL})_{p \rightarrow q} \overset{\text{def}}{=} \{ x \mid I_{p \rightarrow q}(x) = L \}$
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 $\langle R_{p\rightarrow q}, W_{p\leftarrow q} \rangle$

Relative Robustness: Fixed principals $p$ and $q$.

- Static confidentiality levels $C_{p\rightarrow q}$ and integrity levels $I_{p\leftarrow q}$.
- If $I_{p\leftarrow q}(x) = L$, $p$ believes that $q$ can modify $x$.

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

$$\mathcal{FV}(\Phi) \cap (LL \cup HL)_{p\rightarrow q} = \emptyset$$

where $(LL \cup HL)_{p\rightarrow q} \overset{\text{def}}{=} \{ x \mid I_{p\rightarrow q}(x) = L \}$
Conclusions:

A weakest precondition approach to active attacks analysis

Musard Balliu, Isabella Mastroeni

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:

- 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 $|\mathcal{A}| \leq \omega$.

$\downarrow$

- 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 $P$

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

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

$$\begin{cases} (h \mod 2 = 0 \land a = 0 \land b = 0) \lor \\ (h \mod 2 \neq 0 \land a = 1 \land k = b) \end{cases}$$

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

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