The Security π-calculus and Non-interference

M. Hennessy, University of Sussex

- Background
- The Security π-calculus
- Types
- Behavioural Equivalences
- Non-Interference Results

Work in progress by EU Gobal Computing projects Mikado/ Myths

Background

- Control of information flow in systems

  cf. Denning, Goguen, Mesegeur

- **Integrity:** No High-to-Low information flow:
  High (security) level users should not be able to send high-level information to Low level users.
  (No Trojan horses)

- **Non-interference:** Formal property of systems which ensures their integrity.

High-to-Low Information Flow

Explicit: **H** sends high-level data (my visa no) to **L**

Implicit: **H** sends low-level data to **L**

H, L could have prearranged interpretation:

- 0 - Boss is in town
- 1 - Boss is away

Implicit: **H** may rendez-vous with **L**

- **H** turns up - Boss is away
- **H** absent - Boss is in town

How to Avoid H-to-L Information Flow

H can not send any data to **L**

Q?: What kind of data can **L** send to **H** ?
Q?: How can rendez-vous's be managed ?

More General Q?: How can we SPECIFY behaviour of system which will ensure no H-to-L information flow ?

ANSWER: Codify using Types cf. Volpano et al.

A system is **safe** if it can be typechecked

Safe Systems

- How do we prove safe systems contain no H-to-L information flow?
- Introduce Interference-Freeness: Formal verifiable concept, which informally implies no H-to-L information flow
- Main Theorem: S is typeable (using my type system) implies S is interference-free

Interference-Freeness

Requirements:

- concept of High-level process (specified using Type system)
- concept of behavioural equivalence $\simeq_\sigma$, relativised to security levels σ, $(= \mathsf{bot}, \ldots, \mathsf{top})$

Definition: $\boxed{\mathsf{S}}$ is **Interference-Free** if

$$\boxed{\mathsf{S}} \mid H \simeq_{\mathsf{bot}} \boxed{\mathsf{S}} \mid K$$

for all High-level processes H, K.

Remainder of Talk

- Language: π-calculus
- Types: input/output types, relativised to security levels
- Behavioural Equivalences $\simeq_\sigma$: based on testing

The Security π-calculus (asynchronous)

channels = resources = read once variables

- $u?(x : \mathrm{T})\, P$ - patterned input on channel u
  to resource u
- $u!\langle v \rangle$ - polyadic output on channel u
  from resource u
  v a tuple of values - may be **channels**
- $P \mid Q$ - concurrent code
- if $u = v$ then P else Q - value testing
- (new $n : \mathrm{T}$) P - generation of new names
- $*P$, **0** - iteration and termination

Reduction Semantics

Same as ever (for π hackers):

(com) $a!\langle v\rangle \mid a?(x:\mathrm{A})\,P \mapsto (P[{}^{v}/_{x}])$

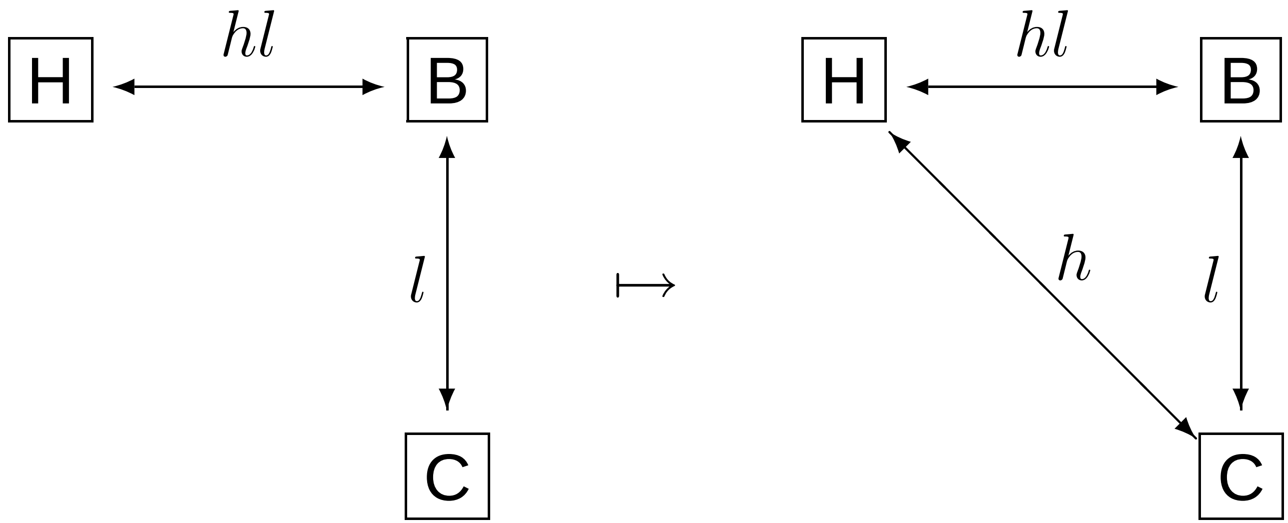
(str) $$\frac{P \equiv Q,\;\; P \mapsto P',\;\; P' \equiv Q'}{Q \mapsto Q'}$$

(cong) $$\frac{P \mapsto P'}{P \mid Q \mapsto P' \mid Q}$$

(etc.)

Reduction Semantics

Dynamic creation of communication links:



Interface between **H** and **L** processes must be managed

Security Levels

A (static) complete lattice of security levels, *SL*.

- bot: lowest security level
  - the great unwashed
  - processes arriving off the web
  - processes at this level offer no security
- top: highest security level
  - the chosen few
  - processes owned by superuser on local machine
- bot $\leq$ *moderate* $\leq$ top:
  - processes originating on local area network
  - processes which have demonstrated some reliability

SL may have an arbitrary complicated structure.

Types - graded read/write capabilities

Type$_\sigma$: Type for values accessible at security level σ

$$\{\mathsf{w}_\sigma\langle \mathrm{A}\rangle,\ \mathsf{r}_{\rho_1}\langle \mathrm{B}_1\rangle,\ \mathsf{r}_{\rho_2}\langle \mathrm{B}_2\rangle,\ \ldots \mathsf{r}_{\rho_k}\langle \mathrm{B}_k\rangle\}$$

provided

- $\sigma \leq \rho_i$ - no write ups
- $\mathrm{A} \in$ *Type*$_\sigma$, $\mathrm{B}_i \in$ *Type*$_{\rho_i}$
- A *subtype* B_i

Example:

- Yes: $\{\mathsf{w}_{\mathsf{bot}}\langle \mathbf{int}\rangle,\ \mathsf{r}_{\mathsf{bot}}\langle \mathbf{int}\rangle,\ \mathsf{r}_{\mathsf{top}}\langle \mathbf{int}\rangle\}$ - multi-level type
- No: $\{\mathsf{w}_{\mathsf{top}}\langle \mathbf{int}\rangle,\ \mathsf{r}_{\mathsf{bot}}\langle \mathbf{int}\rangle,\ \mathsf{r}_{\mathsf{top}}\langle \mathbf{int}\rangle\}$
- No: $\{\mathsf{w}_{\mathsf{bot}}\langle \mathsf{w}_{\mathsf{bot}}\langle \mathbf{int}\rangle\rangle,\ \mathsf{r}_{\mathsf{bot}}\langle \ldots\rangle,\ \mathsf{r}_{\mathsf{top}}\langle \mathsf{w}_{\mathsf{bot}}\langle \mathbf{int}\rangle\rangle\}$

Typing Systems non-stop

Type Environment $\Gamma = u_1 : \mathrm{A}_1, \ldots \ldots u_k : \mathrm{A}_k$

- $\Gamma \vdash P$ - P well -typed wrt Γ, **ignoring security levels**
- $\Gamma \vdash_{\sigma} P$ - P well-typed, using **at most** level σ resources
- $\Gamma \vdash^{\sigma} P$ - P well-typed, using **at least** level σ resources
- $\Gamma \vdash_{r\sigma} P$ - $\ldots$, **reading** from **at most** level σ resources
- $\Gamma \vdash^{w\sigma} P$ - $\ldots$, **writing** to **at least** level σ
- $\ldots\ldots$

Thm: Subject Reduction: $\Delta \Vdash P$ and $P \longmapsto^* Q$ implies $\Delta \Vdash Q$

Type Inference

(LT-IN)
$$\frac{\Gamma, X : \mathrm{A} \vdash_\sigma P \qquad \Gamma \vdash u : \mathsf{r}_\delta\langle \mathrm{A} \rangle}{\Gamma \vdash_\sigma u?(X : \mathrm{A})\, P} \quad \delta \preceq \sigma$$

(LT-OUT)
$$\frac{\Gamma \vdash v : \mathrm{A} \qquad \Gamma \vdash u : \mathsf{w}_\delta\langle \mathrm{A} \rangle}{\Gamma \vdash_\sigma u!\langle v \rangle} \quad \delta \preceq \sigma$$

(T-EQ)
$$\frac{\Gamma \vdash u : \mathrm{A}, v : \mathrm{B} \qquad \Gamma \Vdash Q \qquad \Gamma \sqcap \{u : \mathrm{B}, v : \mathrm{A}\} \Vdash P}{\Gamma \Vdash \mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q}$$

(T-NEW)
$$\frac{\Gamma, a : \mathrm{A} \Vdash P}{\Gamma \Vdash (\mathsf{new}\, a : \mathrm{A})\ P}$$

Examples

$$\boxed{\mathsf{H}} \Leftarrow \mathbf{lh}?(x)\, x!\langle 3pm\rangle$$

$$\boxed{\mathsf{L}} \Leftarrow \mathbf{lh}!\langle cvt\rangle \;\; cvt?(i) \;\; broadcast(i)$$

If $\mathbf{lh}$ is $\mathsf{w}_{\mathsf{bot}}\langle\ldots\rangle,\ \mathsf{r}_{\mathsf{top}}\langle\ldots\rangle$ then $\Gamma \not\models L \mid H$

$L \mid H$ contains information flow

$$\mathit{TrH} \Leftarrow \mathbf{h}?(x)\,.\mathsf{if}\ x = \mathit{boss}\ \mathsf{then}\ \mathsf{tr}_1!\langle\rangle\ \ \mathsf{else}\ \mathsf{tr}_2!\langle\rangle$$

If $\mathbf{h}$ high, tr_i low, then *TrH* can not be High-level

TrH represents a trojan horse

Safe Systems at last

Definition: $\boxed{\mathsf{S}}$ is Γ-safe if $\Gamma \vdash_{r\mathsf{bot}} S$

They can only read from low-level channels

Claim: If $\boxed{\mathsf{S}}$ is Γ-safe then

$$\boxed{\mathsf{S}} \mid H \simeq_{\mathsf{bot}} \boxed{\mathsf{S}} \mid K \qquad \text{informal}$$

for all High-level processes $H,\ K$.

Definition: H is a High-level process if $\Gamma \vdash^{w\mathsf{top}} H$

They can only write to high-level channels

Behavioural Equivalences

Idea: $\boxed{\text{S}} \simeq_\sigma \boxed{\text{U}}$ at level σ, if no observer running at level at most σ can not distinguish between S and U.

- An observation of S by O is a sequence $O \mid S \mapsto O_1 \mid S_1 \ldots \mapsto O_n \mid S_n \mapsto \ldots$
- Successful if some O_k, can report success
- S **may** O if there is some successful observation of S by O
- S **must** O if every observation of S by O is successful

Definition: $\Gamma \triangleright_\sigma \boxed{\text{S}} \simeq_{may} \boxed{\text{U}}$ if for every $\Gamma \vdash_\sigma O$,
S **may** O if and only if U **may** O

$\Gamma \triangleright_\sigma \boxed{\text{S}} \simeq_{must} \boxed{\text{U}}$ if . . . . . .

Non-Interference

Idea: S is interference-free if low-level observers/users can not detect the presence/absence of high-level users in S.

Definition: S is mayIntFree if

$$\Gamma \triangleright_{\mathsf{bot}} S \mid H \simeq_{may} S \mid K$$

for all High-level process $H,\ K$

NonInterference for Free:

Thm: If S is Γ-safe ($\Gamma \vdash_{r\mathsf{bot}} S$) then S is mayIntFree

Examples

Assume $H,\ K$ high-level ($\Gamma \vDash^{w\mathsf{top}} H,\ K$)
S safe ($\Gamma \vdash_{r\mathsf{bot}} S$)

$$\boxed{\mathsf{H}} = \mathbf{h}?(x)\,\mathsf{if}\ x = \textit{boss}\ \mathsf{then}\ \mathsf{tr}_1!\langle\rangle\ \ \mathsf{else}\ \mathsf{tr}_2!\langle\rangle$$

$$\boxed{\mathsf{K}} = \mathbf{h}?(x)\,\mathsf{tr}_1!\langle\rangle$$

$\Gamma \rhd_{\mathsf{bot}} S \mid H \simeq_{may} S \mid K$ because write on tr_i must be high

$$\boxed{\mathsf{H}} = \mathbf{h}?(x)\,\mathsf{if}\ x = \textit{boss}\ \mathsf{then}\ \mathsf{tr}_1?()\ \ \mathsf{else}\ \mathsf{tr}_2?()$$

$$\boxed{\mathsf{K}} = \mathbf{h}?(x)\,\mathsf{tr}_1?()$$

$\Gamma \rhd_{\mathsf{bot}} S \mid H \simeq_{may} S \mid K$ because communication is asynchronous

Example: Multi-level types

Γ maps ml to $\{\mathsf{w}_{\mathsf{bot}}\langle\ldots\rangle, \mathsf{r}_{\mathsf{bot}}\langle\ldots\rangle, \mathsf{r}_{\mathsf{top}}\langle\ldots\rangle\}$ multi-level type

$\boxed{\mathsf{S}} = \mathsf{ml}!\langle a\rangle \mid \mathsf{ml}?(x)\ x!\langle\rangle$

$\Gamma \rhd_{\mathsf{bot}} S \mid H \simeq_{may} S \mid K$ because S is safe

BUT: $\Gamma \rhd_{\mathsf{bot}} S \mid H \not\simeq_{must} S \mid K$
eg with $H = \mathbf{0}$ and $K = \mathsf{ml}?(x : \mathrm{B})\,\mathbf{0}$
observer $a?()\ \omega!\langle\rangle$ sees a difference

Thm: Suppose Γ uses only single-level types.

If S is Γ-safe then it is mustIntFree

Wrap up

Thesis: Potential H-to-L information flow in concurrent systems can be detected by type systems

Questions:

- How difficult is type inference?
- How restrictive is the type system?
- Can types be extended to distributed systems?

Technical Details: Sussex technical reports