Type-Based Security for Mobile Computing

Nobuko Yoshida

Department of Computing
Imperial College London
Type-Based Security for Mobile Computing

- The open characteristic of mobile computing introduces fundamental security issues.

- Example: E-commerce mobile agent
  - **Integrity (Access Control)**: the agent may modify sensitive information
  - **Privacy (Secrecy)**: the agent may transfer credit card numbers to the public channels
  - **Availability (Liveness)**: the agent may enter into an infinite loop, consuming unbounded local resources.
The $\pi$-Calculus [1989–]

Shared Point

Name Passing as Pointer Passing

Identity of Agent
The \( \pi \)-Calculus [1989–]

- Shared Point
- Name Passing as Pointer Passing
- Identity of Agent

Diagram:
- Tom
- Lisa
- Bill
- \( a \)
- New
The π-Calculus

\[ a(x).P | \bar{a}v \rightarrow P[v/x] \]
Higher-Order π-Calculus

[Sangiorgi 93]

CML, Facile, LLlinda, ...

\( \forall v \ (\lambda x.Q)v \rightarrow Q[v/x] \)

\( \forall v \bar{a}\langle P\rangle | a(x). \text{run} X \)

\[ \rightarrow \text{run } \bar{P} \rightarrow P \]

where

\[ \text{thunk} \quad r_P^\downarrow = \lambda(). P \]

\[ \text{run} \quad = \lambda X. X() \]
Aims of Types/Typechecking

- Using Types to control the effects of Mobile Code / Processes.
- Host refuses to execute incoming code unless it conforms to predetermined access policy.
History of Types in HOPi/π

HOPi

93 HOPi [Sangiorgi PhD]

98 Fine-Grained Types [Yoshida and Hennessy]

Other Mobility Types

• Seal Calculus
• M-Calculus
• Safe-Boxed Ambient

92 Polyadic π
93 IO-Subtyping
Linear Typings
Polymorphism
Causality-based Typings

89 π-Calculus

01 Fully Abstraction
• PCF [TCAL 01]
• λ→+x [LICS 01]
• System F [FoSSacs 01]
• Control [CW04]

02 Secure Info Flow
[ESOP 01, FoSSacs 02, POPL 02]

03 Channel Dependency
Existential Types

⇒ Integration with Linearity

⇒ Secure Info Role-Based Access Control
Problem

\[ c(X: \texttt{proc}). \text{run } X \]

where \[ \texttt{proc} = \text{unit} \rightarrow \text{proc} \]

\[ \text{Any Process is welcome} \]

\[ \text{We are very different} \]

P \[ \bar{a}(X). \bar{b}(X) \]

Q \[ b(X). \bar{a}(X) \]

R \[ d(\text{string}) \]
Assigning Types to Processes

\[ \Gamma \vdash P : [\Delta] \]

channel environment
\[ \square \]
INTERFACE

Example

\[ P \vdash a(x). \overline{b}(x) : [a : \{\tau\}^1, b : \{\tau\}^0] \]

\[ Q \vdash b(x). \overline{a}(x) : [a : \{\tau\}^0, b : \{\tau\}^1] \]

\[ R \vdash \text{d}<\text{string}> : [d : \text{(string)}^0] \]

\[ C(X : \{a : \{\tau\}^1, b : \{\tau\}^0\}). \text{run} X \]

O O O O O O O
History of Types in $\text{HO}\pi\pi$

- $\text{HO}\pi\pi$
  - [Sangiorgi PhD]

- $\pi$
  - 89: $\pi$-Calculus
  - 92: Polyadic $\pi$
  - 93: IO-Subtyping
  - 93: Linear Typings
  - 93: Polymorphism
  - 93: Causality-based Typings

- 98: Fine-Grained Types
  - [Yoshida and Hennessy]

- 00: Other Mobility Types
  - Seal Calculus
  - M-Calculus
  - Safe-Boxed Ambient

- 02: Secure Info Flow
  - PCF [TCAL 01]
  - $\lambda\rightarrow+\times$ [LICS 01]
  - System F [FoSSacs 01]
  - Control [CW 04]

- 03: Channel Dependency
  - Existential Types

$\Rightarrow$ Integration with Linearity

$\Rightarrow$ Application Secure Info Role-Based Access Control
Main Theorems

Subject Reduction
\[ \Gamma \vdash P : \Delta, \ P \rightarrow P' \Rightarrow \Gamma \vdash P' : \Delta \]

Type Safety
\[ \Gamma \vdash P : \left[\Delta\right] \Rightarrow P \xrightarrow{\text{err}}\]

where \( P \xrightarrow{\text{err}} \) means
\( P \) can use at most resources in \( \Delta \)

Consequence:
\[ a(x : \left[\Delta'\right]).P \mid \overline{a}(R) \xrightarrow{\text{err}} \]
if \( \Gamma \not\vdash R : \left[\Delta\right] \)
The Project Aim

- to develop a general and uniform framework for type systems of mobile programs using process calculi.

- (Base Language) $\Rightarrow$ Liveness and Integrity
  - An integrated framework using typed $\pi$

- (Mobility) $\Rightarrow$ Access Control
  - A fine-grained typing system for HO$\pi$

- (Application) $\Rightarrow$ Language Design Discipline
  - A secure distributed multi-threaded Java and a proof of its correctness

- **Novelty** directly impacting language/system design from the initial stage
The $\pi$-Calculus

$$a(x).P | \bar{a}v \rightarrow P\{v/x\}$$
Conclusion

➢ Timeliness and Impacts

W3C Choreography Working Group www.w3c.org

➢ Many themes for ambitious RAs and PhDs, involving both theory and software design and implementation

➢ Contact

➢ Nobuko Yoshida yoshida@doc.ic.ac.uk
  Home Page: www.doc.ic.ac.uk/~yoshida

➢ Kohei Honda kohei@dcs.qmul.ac.uk
  Home Page: www.dcs.qmul.ac.uk/~kohei