The Higher-Order, Call-by-Value Applied Pi-Calculus

Nobuyuki Sato
Eijiro Sumii
(Tohoku University)

Agenda of the talk

- Informal overview of the work
- A little technical details
- A little more technical details
- Conclusion

Main Result

A bisimulation proof technique
for higher-order process calculus
with cryptographic primitives

- Can be used for proving security properties of concurrent systems that send/receive programs using encryption/decryption

Motivation

Higher-order cryptographic systems are now ubiquitous

- Web-based e-mail clients (e.g. Gmail)
- Software update systems (e.g. Windows Update)

Higher-order: transmitting programs themselves

⇒ Security is even more important than in first-order systems

- Cryptography is essential

Problem

The theory of higher-order cryptographic computation is underdeveloped

- **Little work for the combination of higher-order processes and cryptographic primitives**
  - Cf. Higher-order pi-calculus (no cryptography), spi-calculus (first-order), ...

A Challenge of Higher-Order Cryptographic Processes

- Consider the process $P = \overline{c}\langle Q\rangle$ where $Q = \overline{c}\langle encrypt(m,k)\rangle$
  - $\overline{c}\langle\ \rangle$ denotes output to the network c
  - Assume c is public and k is secret
- Does P leak m?
  1. Yes, because the attacker can receive Q from c and <u>extract</u> m
  2. No, if m is encrypted <u>before</u> Q is sent to c

Observations

- Computation (e.g. encryption) and computed values (e.g. ciphertext) must be distinguished
- The attacker should be able to decompose transmitted processes (but not computed values)

(Recall the previous example $P = \overline{c}\langle Q \rangle$ where $Q = \overline{c}\langle encrypt(m,k) \rangle$)

Solution

- Syntactically distinguish computation (e.g. encrypt(m,k)) and computed values (e.g. ^encrypt(m,k))
- Extend the calculus with a primitive to decompose transmitted processes:

  match P as x in Q

  (bind x to the decomposed abstract syntax tree of P and execute Q)
  - Computed values can not be decomposed

Examples

$\overline{c}\langle\ \overline{c}\langle \mathrm{encrypt}(m,k)\rangle\ \rangle$ |

c(X).match X as y in R

→ match $\overline{c}\langle \mathrm{encrypt}(m,k)\rangle$ as y in R

→ [Out(Nam c,Enc(Nam m,Nam k))/y]R

$\overline{c}\langle\ \overline{c}\langle \hat{}\mathrm{encrypt}(m,k)\rangle\ \rangle$ |

c(X).match X as y in R

→ match $\overline{c}\langle \hat{}\mathrm{encrypt}(m,k)\rangle$ as y in R

→ [Out(Nam c,Val ^encrypt(m,k))/y]R

Next Challenge

How do we reason about higher-order cryptographic processes?

- Traditional techniques (bisimulations, in particular) do not apply
  - Most of them are first-order
  - Normal bisimulations [Sangiorgi 92] are unsound for process decomposition
    - Because they only transmit "triggers" (i.e. pointers to processes)

Solution

Adopt environmental bisimulations

- Devised for λ-calculus with encryption [Sumii-Pierce 04]
- Adapted for various languages [Sumii-Pierce, Koutavas-Wand, ...]
  - Including higher-order pi-calculus [Sangiorgi-Kobayashi-Sumii 07]

Idea of Environmental Bisimulations

- Traditional (i.e. non-environmental) bisimulation $P \sim P'$ means:

  P and P' behave the same
  under any observer process

- Environmental bisimulation $P \sim_E P'$ means:

  P and P' behave the same
  under any observer process
  <u>that uses any elements (V,V') of E</u>

  - E is a binary relation on values
    that represents the observer's <u>knowledge</u>
    (called an <u>environment</u>)

Agenda of the talk

- Informal overview of the work
- A little technical details
- A little more technical details
- Conclusion

Our Environmental Bisimulations (1/3)

Binary relation X on processes,
indexed by environments E,
is an <u>environmental simulation</u>
if $P\ X_E\ P'$ implies:

1. If P reduces to Q, then
   P' reduces to some Q'
   such that $Q\ X_E\ Q'$
2. If P outputs V and becomes Q, then
   P' outputs some V' and becomes some Q'
   such that $Q\ X_{E\cup\{(V,V')\}}\ Q'$

(cont.)

Our Environmental Bisimulations (2/3)

X is an environmental simulation
if P X_E P' implies:

3. For any V and V' composed from E,
if P inputs V and becomes Q, then
P' inputs V' and becomes some Q'
such that Q X_E Q'
   - "Composed from" means
   for some context C and $(V_1, V_1'), \ldots, (V_n, V_n') \in E$,
   $V = C[V_1, \ldots, V_n]$ and $V' = C[V_1', \ldots, V_n']$
4. P|Q X_E P'|Q' for any $(Q, Q') \in E$

(cont.)

Our Environmental Bisimulations (3/3)

X is an environmental simulation
if P X_E P' implies:

5. P $X_{E\cup\{(V,V')\}}$ P' if V and V' can be computed from E (by decomposition or first-order computation)

   E.g. suppose:

   E = {(k,k'), (^encrypt(V,k),^encrypt(V',k'))}

   Then (V,V') can be computed from E by the first-order context:

   $$C = \text{decrypt}([]_2,[]_1)$$

6. E preserves equality

Main Theorem

The <u>largest</u> environmental bisimulation (with appropriate E) coincides with reduction-closed barbed equivalence

- It exists because the generating function is monotone [Tarski 55]

- The $\subseteq$ direction is proved via a context closure property of environmental bisimulations
- The $\supseteq$ direction is proved by coinduction

Agenda of the talk

- Informal overview of the work
- A little technical details
- A little more technical details
- Conclusion

Our Calculus: Syntax of Terms

$M ::=$	terms
V	values
x	variables
$M(M_1,\ldots,M_n)$	computations
$V ::=$	values
a	names
f	function symbols
$\hat{}f(V_1,\ldots,V_n)$	computed values
$`P$	transmitted processes
$`M$	transmitted terms

Syntax of Processes

P ::=	processes
0	inaction
$M(x).P$	input
$\overline{M}\langle N\rangle.P$	output
$P\|Q$	parallel composition
$!P$	replication
$\nu x.P$	restriction
run(M)	execution
if $M=N$ then P else Q	conditional
match M as x in P	decomposition

Labeled Transition Semantics

- Parameterized by semantics of terms
  - Defined by (strongly normalizing and confluent) term rewriting system
- Key rules:

$\overline{c}\langle M\rangle.P \xrightarrow{\overline{c}\langle V\rangle} P$

if M rewrites to V ("call-by-value")

$\mathrm{run}(\grave{}P) \xrightarrow{\tau} P$ (important!)

$\mathrm{match}\ \grave{}P\ \mathrm{as}\ x\ \mathrm{in}\ Q \xrightarrow{\tau} [M/x]Q$

where M is decomposed AST of P

Examples (Revisited)

$\overline{c}$⟨ ` $\overline{c}$⟨encrypt(m,k)⟩ ⟩ |
c(X).match X as y in R
→ match ` $\overline{c}$⟨encrypt(m,k)⟩ as y in R
→ [Out(Nam c,Enc(Nam m,Nam k))/y]R

$\overline{c}$⟨ ` $\overline{c}$⟨^encrypt(m,k)⟩ ⟩ |
c(X).match X as y in R
→ match ` $\overline{c}$⟨^encrypt(m,k)⟩ as y in R
→ [Out(Nam c,Val ^encrypt(m,k))/y]R

Bisimulation Example

$P = \overline{c}\langle$ \` $\overline{c}\langle$^encrypt(3,k)$\rangle$ $\rangle$ and
$P' = \overline{c}\langle$ \` $\overline{c}\langle$^encrypt(7,k)$\rangle$ $\rangle$
are bisimilar

Proof outline: Take X as follows (so $P\ X_E\ P'$)

X = { (E, C[^encrypt(3,k)], C[^encrypt(7,k)]) | k not free in C }

E = { (D[^encrypt(3,k)], D[^encrypt(7,k)]) | k not free in D }

and prove it to be an env. bisim.
(by case analysis on C and D)

Non-Bisimulation Example

$P = \overline{c}\langle \ `\ \overline{c}\langle encrypt(3,k)\rangle \ \rangle$ and
$P' = \overline{c}\langle \ `\ \overline{c}\langle encrypt(7,k)\rangle \ \rangle$ are
not bisimilar

Proof outline:

If $P\ X_E\ P'$ for some env. bisim. X and E, then by output we get $0\ X_{E'}\ 0$ with $(`\ \overline{c}\langle encrypt(3,k)\rangle, `\ \overline{c}\langle encrypt(7,k)\rangle) \in E'$.

Since (3,7) can be computed from E' by decomposition, we get $0\ X_{E''}\ 0$ with $(3,7) \in E''$, which violates integer equality.

Simplification by Up-To Context Technique

Problem:

Many environmental bisimulations include all processes/values of the forms $C[V_1,\ldots,V_n]$ and $C[V_1',\ldots,V_n']$ for some $(V_1,V_1'),\ldots,(V_n,V_n')$

Solution:

A "smaller" version of environmental bisimulations, where processes/values of the forms $C[V_1,\ldots,V_n]$ and $C[V_1',\ldots,V_n']$ can be omitted if $(V_1,V_1'),\ldots,(V_n,V_n')$ are included

Example of Environmental Bisimulation Up-To Context

Consider again:

$$P = \overline{c}\langle\ \grave{}\ \overline{c}\langle\hat{}\mathrm{encrypt}(3,k)\rangle\ \ \rangle$$

$$P' = \overline{c}\langle\ \grave{}\ \overline{c}\langle\hat{}\mathrm{encrypt}(7,k)\rangle\ \ \rangle$$

Then

$$Y = \{\ (E,\ P,\ P')\ \}$$

is an environmental bisimulation up-to context, where:

$$E = \{(c,c), (\hat{}\mathrm{encrypt}(3,k), \hat{}\mathrm{encrypt}(7,k))\}$$

In the paper

- Formal definitions of the calculus and our environmental bisimulations (and the up-to context technique)
- Soundness and completeness proofs (i.e. proof of coincidence with reduction-closed barbed equivalence)
- More sophisticated examples
  - Software distribution system
  - Online e-mail client

Agenda of the talk

- Informal overview of the work
- A little technical details
- A little more technical details
- Conclusion

Conclusions

- Higher-order cryptographic processes are non-trivial
  - Previous theories do not apply (higher-order pi-calculus, spi-calculus, …)
- Environmental bisimulations "scale" well to such sophisticated calculi
  - Including the present one

- Future work: automation, extension, simplification, …