Noninterference

Philip W. L. Fong

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

CPSC 601.40 (Fall 2016)
BLP Reconsidered

- BLP depends on the correct attribution of accesses as observations or alterations.
  - It correctly controls information flow only when the attribution is correct and complete.
Example: Timing Channel

- Suppose 2 users are in a, say, BLP system.
- User 1, who has higher clearance, dominates the CPU when she chooses to run a process.
- They agree to transmit a single bit of information in regular intervals.
  - Bit 1: user 1 runs a process, thereby dominating the CPU, causing user 2 to be denied CPU access.
  - Bit 0: user 1 doesn’t run anything, thereby freeing the CPU for user 2.
The “ss-property” and “*-property” do not properly capture the essence of information flow.

- They are tied to particular mechanism for controlling information flow.

Information flow should be defined independently of particular access control mechanisms.
Outline

1. Machine Model
2. Noninterference Security
3. Views
4. The Unwinding Theorem
5. Sketch of Completeness Proof
Machine Model

Noninterference Security

Views

The Unwinding Theorem

Sketch of Completeness Proof

Outline

1. Machine Model

2. Noninterference Security

3. Views

4. The Unwinding Theorem

5. Sketch of Completeness Proof
A system (or machine) is composed of

- a set $S$ of states,
- an initial state $s_0 \in S$
- a set $A$ of actions
- a set $O$ of outputs
- a function $\text{step} : S \times A \rightarrow S$
- a function $\text{output} : S \times A \rightarrow O$

Akin to a Mealy machine.

- deterministic
- input-total
A **system** (or **machine**) is composed of
- a set $S$ of **states**,
- an **initial state** $s_0 \in S$
- a set $A$ of **actions**
- a set $O$ of **outputs**
- a function $\text{step} : S \times A \rightarrow S$
- a function $\text{output} : S \times A \rightarrow O$

Akin to a Mealy machine.
- deterministic
- input-total
Shorthands

- $\text{run}(s, \alpha)$
  - state obtained by applying action sequence $\alpha$ to state $s$
- $\text{do}(\alpha) = \text{run}(s_0, \alpha)$
- $\text{test}(\alpha, a) = \text{output}(\text{do}(\alpha), a)$
Outline

1. Machine Model
2. Noninterference Security
3. Views
4. The Unwinding Theorem
5. Sketch of Completeness Proof
Domains

- a set $D$ of security domains
  - subjects sharing the same security properties belong to the same domain
- a function $\text{dom} : A \rightarrow D$
A security policy is a reflexive relation $\rightsquigarrow$

- $u \rightsquigarrow v$: “$u$ may interfere with $v$”
  - information may flow from $u$ to $v$
- $u \not\rightsquigarrow v$: “$u$ must not interfere with $v$”
  - information must not flow from $u$ to $v$
Noninterference and Purging

- \( \text{purge}(\alpha, v) \)
  - Subsequence of \( \alpha \) with actions \( a \) for which \( \text{dom}(a) \leadsto v \)
  - Actions \( b \) for which \( \text{dom}(b) \nleq v \) are “purged”.

- A system is **secure** for policy \( \leadsto \) iff

\[
\text{test}(\alpha, a) = \text{test}(\text{purge}(\alpha, \text{dom}(a)), a)
\]
Intuition

- An action $a$ “interferes” with an output if its elimination changes the output.
- An action $b$ does not “interfere” with an output if the output remains the same after $b$ is eliminated.
- Suppose $u \not\rightarrow v$. Then, an observer in domain $v$ cannot distinguish the two possibilities below:
  1. Something interesting occurred in domain $u$.
  2. Nothing occurred in domain $u$.

Because both explanations are consistent with the observation available in domain $v$, a rational observer is forced to wonder: “maybe nothing has happened?”
Outline

1. Machine Model
2. Noninterference Security
3. Views
4. The Unwinding Theorem
5. Sketch of Completeness Proof
A system $M$ is **view-partitioned** if, for each domain $u \in D$, there is an equivalence relation $u \sim$ on $S$.

The relations $u \sim$ are **output-consistent** if

$$s^{dom(a)} \sim t \Rightarrow output(s, a) = output(t, a)$$

Intuitively, $u \sim$ induces equivalence classes of states that are **observationally equivalent**.
Another Path to Noninterference

Lemma

Let $\rightsquigarrow$ be a policy and $M$ a view-partitioned, output consistent system such that,

$$
\text{do}(\alpha) \cup \text{do}(\text{purge}(\alpha, u))
$$

Then $M$ is secure for $\rightsquigarrow$. 
Some states are indistinguishable from the perspective of a subject living in a certain domain.

From such a view, the machine has “fewer” states than it actually has.

Output consistency grounds the notion of observational equivalence with the output semantics of the machine.
Motivation

- Noninterference is a global property of the system.
  - Hard to establish!
- Want a “local” verification condition similar to the preservation property in, say, the Basic Security Theorem in BLP.
Let $M$ be a view-partitioned system and $\sim$ a policy. We say that $M$ is \textit{step consistent} if

$$s \sim u \implies step(s, a) \sim u \implies step(t, a)$$
Let $M$ be a view-partitioned system and $\rightsquigarrow$ a policy. We say that $M$ **locally respects** $\rightsquigarrow$ if

$$\text{dom}(a) \not\owns u \Rightarrow s \overset{u}{\sim} \text{step}(s, a)$$
Let \( \rightsquigarrow \) be a policy and \( M \) a view-partitioned system. \( M \) is said to satisfy the **Unwinding Condition** iff the following hold:

1. \( M \) is output consistent
2. \( M \) is step consistent
3. \( M \) locally respects \( \rightsquigarrow \)
Due to its “local” nature, the Unwinding Condition can serve as a verification condition for Noninterference.

Need to show that the test is **sound** and **complete**.

- **Soundness**: unwinding $\Rightarrow$ noninterference
  - so we won’t draw the wrong conclusion
- **Completeness**: noninterference $\Rightarrow$ unwinding
  - so we won’t miss out genuinely secure systems
Soundness of Unwinding

**Theorem (Soundness)**

Let $\rightsquigarrow$ be a policy and $M$ a view-partitioned system. If $M$ satisfies the unwinding condition then $M$ is secure for $\rightsquigarrow$.

- This theorem is also known as the **Unwinding Theorem**.
Theorem (Completeness)

Suppose system $M$ is secure for policy $\rightsquigarrow$. Then there is a family of equivalence relations $\sim_u$ that satisfy the unwinding condition.
Outline

1. Machine Model
2. Noninterference Security
3. Views
4. The Unwinding Theorem
5. Sketch of Completeness Proof
Construction

- Given \( \rightsquigarrow \) and \( M \), construct the following view-partitioning relation:

\[
\begin{align*}
    s \sim^u t & \overset{\text{def}}{=} (\forall \alpha \in A^* . \forall b \in A . \ \
    \text{dom}(b) = u \Rightarrow \
    \text{output}(\text{run}(s, \alpha), b) = \text{output}(\text{run}(t, \alpha), b))
\end{align*}
\]

- Intuitively, \( s \sim^u t \) if all future observations made in domain \( u \) are identical no matter which of \( s \) or \( t \) we start with.

\[
\begin{array}{c}
    s \sim^u t \\
    \text{output}(s', b) \\
    \sim \\
    \text{output}(t', b)
\end{array}
\]
Take $\alpha$ to be the empty string, we have:

$$s^{\text{dom}(b)} \sim t \Rightarrow \text{output}(s, b) = \text{output}(t, b)$$

$$s \cdots \text{output}(s', b)$$

$$\text{dom}(b) \sim$$

$$\text{output}(t', b)$$
**Step Consistency**

- **Want:** $s \sim t \Rightarrow \text{step}(s, a) \sim \text{step}(t, a)$.
- Suppose the $u$-outputs generated from executions initiated at $s$ is identical to the $u$-outputs generated from execution initiated at $t$.

\[
\begin{align*}
  s \xrightarrow{\alpha} & s' \quad \cdots \quad \text{output}(s', b) \\
  \phantom{\sim} & \phantom{\sim} \\
  \text{dom}(b) \sim & \phantom{\sim} \\
  \phantom{\sim} & \phantom{\sim} \\
  t \xrightarrow{\alpha} & t' \quad \cdots \quad \text{output}(t', b)
\end{align*}
\]

Then it is obvious that, after taking one step of execution, future $u$-outputs are still identical.

\[
\begin{align*}
  s \xrightarrow{a} & s'' \xrightarrow{\alpha} s' \quad \cdots \quad \text{output}(s', b) \\
  \phantom{\sim} & \phantom{\sim} \\
  \text{dom}(b) \sim & \phantom{\sim} \\
  \phantom{\sim} & \phantom{\sim} \\
  t \xrightarrow{a} & t'' \xrightarrow{\alpha} t' \quad \cdots \quad \text{output}(t', b)
\end{align*}
\]
Local Respect for $\rightsquigarrow$

Suppose $\text{dom}(a) \not\rightsquigarrow \text{dom}(b)$.

\[
\begin{align*}
\text{s}_0 & \rightsquigarrow \gamma \rightarrow \text{s} \\
\text{s} & \rightsquigarrow \alpha \rightarrow \text{s}' \\
\text{s}' & \rightsquigarrow \alpha \rightarrow \text{s}'' & \text{output}(\text{s}'', \text{b})
\end{align*}
\]

Then

\[
\begin{align*}
\text{output}(\text{s}'', \text{b}) &= \text{test}(\gamma \cdot \text{a} \cdot \alpha, \text{b}) \\
&= \text{test}(\text{purge}(\gamma \cdot \text{a} \cdot \alpha, \text{dom}(\text{b})), \text{b}) \\
&= \text{test}(\text{purge}(\gamma \cdot \alpha, \text{dom}(\text{b})), \text{b}) \\
&= \text{test}(\gamma \cdot \alpha, \text{b}) \\
&= \text{output}(\text{s'''}, \text{b})
\end{align*}
\]
Point To Remember

Intuition

Two states are observationally equivalent with respect to domain $u$ if their future outputs (i.e., computational effects) are identical when observed from domain $u$. 
Access Control Interpretations

- Model of an access control system similar to BLP.
  - “No read up” and “No write down”
- Shows that the model satisfies noninterference.
- Shows that transitive policies can always be expressed as MLS-style policies.
Methodological Lessons

- Separation of policy and mechanism
- Observational equivalence
- The need for unwinding as a verification condition
  - Soundness is a “must”.
  - Completeness is nice to have.