Towards Semantics for Provenance Security

Stephen Chong
Harvard University
TaPP ‘09
Provenance security

- Some data are sensitive
  - Must ensure provenance does not reveal sensitive data
    - E.g., “John participated in medical study S” reveals “John has disease D”
Provenance security

- Some data are sensitive
  - Must ensure provenance does not reveal sensitive data
    - E.g., “John participated in medical study S” reveals “John has disease D”

- Some provenance is sensitive
  - Must ensure output does not reveal sensitive provenance
    - E.g., Workshop referee reports should not contain name/email of referee
  - Must ensure provenance does not reveal sensitive provenance
    - E.g., If student in Disciplinary Hearing, then student’s advisor must attend.
      “Prof. Smith participated as an Advisor” may reveal “John participated as respondent”
Provenance security

- Some data are sensitive
  - Must ensure provenance does not reveal sensitive data
    - E.g., “John participated in medical study S” reveals “John has disease D”
- Some provenance is sensitive
  - Must ensure output does not reveal sensitive provenance
    - E.g., Workshop referee reports should not contain name/email of referee
  - Must ensure provenance does not reveal sensitive provenance
    - E.g., If student in Disciplinary Hearing, then student’s advisor must attend. “Prof. Smith participated as an Advisor” may reveal “John participated as respondent”

- How do we know if we have security right?
  - Complex interaction between information security and provenance
  - Not well-understood
Semantics for provenance security

Goal:
- precise, useful, intuitive definitions of provenance security
- understand provenance security
- principles and mechanisms to apply in practice

This work: Formal definitions for provenance security
- public data does not reveal sensitive provenance
- public provenance does not reveal sensitive provenance
- public provenance does not reveal sensitive data
- (public data does not reveal sensitive data)
Semantics for provenance security

Goal:
- precise, useful, intuitive definitions of provenance security
- understand provenance security
- principles and mechanisms to apply in practice

This work: Formal definitions for provenance security
- public data does not reveal sensitive provenance
- public provenance does not reveal sensitive provenance
- public provenance does not reveal sensitive data
- (public data does not reveal sensitive data)
Language model

- Simple language-based model (based on Cheney, Acar, Ahmed [2008])
- Program $c$ has input locations, produces single output
  
  $\langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v$

E.g.,

$\langle l_1=3, l_2=5, l_3=7 ; x = l_1; \text{if (x) then } l_2 \text{ else } l_3 \rangle \Rightarrow 5$
**Language model**

- Simple language-based model (based on Cheney, Acar, Ahmed [2008])
- Program $c$ has input locations, produces single output
  - $\langle l_1=v_1, \ldots, l_n=v_n \ ; \ c \rangle \Rightarrow v$
- Provenance $T$ describes execution
  - $\langle l_1=v_1, \ldots, l_n=v_n \ ; \ c \rangle \Rightarrow v \models T$

E.g.,
$\langle l_1=3, l_2=5, l_3=7 \ ; \ x = l_1; \text{if } (x) \text{ then } l_2 \text{ else } l_3 \rangle \Rightarrow 5$

$\models x=l_1; \text{cond}(x,\text{true},l_2)$
Language model

- Simple language-based model (based on Cheney, Acar, Ahmed [2008])
- Program c has input locations, produces single output
  - \( \langle l_1=\nu_1, \ldots, l_n=\nu_n ; \ c \rangle \Rightarrow \nu \)
- Provenance \( T \) describes execution
  - \( \langle l_1=\nu_1, \ldots, l_n=\nu_n ; \ c \rangle \Rightarrow \nu \ \vdash \ T \)
- Partial provenance: allow parts of \( T \) to be elided

E.g.,
\[ \langle l_1=3, l_2=5, l_3=7 ; \ x = l_1; \text{if (x) then l}_2 \text{ else l}_3 \rangle \Rightarrow 5 \]
\[ \vdash x=l_1; \text{cond}(x,\text{true},l_2) \]
Language model

- Simple language-based model (based on Cheney, Acar, Ahmed [2008])

- Program $c$ has input locations, produces single output
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$

- Provenance $T$ describes execution
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \models T$

- Partial provenance: allow parts of $T$ to be elided

E.g.,

$\langle l_1=3, l_2=5, l_3=7 \ ; \ x=l_1; \ \text{if} \ (x) \ \text{then} \ l_2 \ \text{else} \ l_3 \rangle \Rightarrow 5$

$\models x=l_1 ; \ \text{cond}(x, \text{true}, \star)$
Language model

- **Simple language-based model** (based on Cheney, Acar, Ahmed [2008])
- Program $c$ has input locations, produces single output
  - $\langle l_1=v_1, \ldots, l_n=v_n \rangle; c \Rightarrow v$
- Provenance $T$ describes execution
  - $\langle l_1=v_1, \ldots, l_n=v_n \rangle; c \Rightarrow v \models T$
- Partial provenance: allow parts of $T$ to be elided

E.g.,

$\langle l_1=3, l_2=5, l_3=7 \rangle; x = l_1; \text{if} \ (x) \ \text{then} \ l_2 \ \text{else} \ l_3 \Rightarrow 5$

$\models x = l_1; \ cond(x,\star,\star)$
Language model

- Simple language-based model (based on Cheney, Acar, Ahmed [2008])
- Program $c$ has input locations, produces single output
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \; c \rangle \Rightarrow v$
- Provenance $T$ describes execution
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \; c \rangle \Rightarrow v \models T$
- Partial provenance: allow parts of $T$ to be elided

E.g.,
$\langle l_1=3,l_2=5,l_3=7 \; ; \; x = l_1; \text{if (x) then } l_2 \text{ else } l_3 \rangle \Rightarrow 5$

$\models x=l_1 \; \star$
Security policies

- Each input location has security policy for data and provenance
  - e.g., $\Gamma(l_1) = LL$, $\Gamma(l_2) = LH$, $\Gamma(l_3) = HH$

Data security:
- H : High security (secret)
- L : Low security (public)

Provenance security:
- H : High provenance (secret)
- L : Low provenance (public)
Security policies

- Each input location has security policy for data and provenance
  - e.g., $\Gamma(l_1) = LL$ \hspace{1cm} $\Gamma(l_2) = LH$ \hspace{1cm} $\Gamma(l_3) = HH$

- User knows low security inputs, and is given output and partial provenance trace
  - User should not learn high security data
  - User should not learn which high provenance locations involved in computation

What (partial) provenance can we give to user?
First attempt

- We think $T$ is secure for execution
  
  $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ if:

  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \models T$ and

- $T$ does not contain any high provenance locations.
First attempt

We think $T$ is secure for execution

\[ \langle l_1=v_1, \ldots, l_n=v_n \; ; \; c \rangle \Rightarrow v \text{ if:} \]

1. $\langle l_1=v_1, \ldots, l_n=v_n \; ; \; c \rangle \Rightarrow v \models T$ and
2. $T$ does not contain any high provenance locations.

E.g.,

\[ \langle \ldots ; \; \text{if } (l_1) \; \text{then} \; l_2 + l_3 \; \text{else} \; l_4 + l_5 \rangle \Rightarrow 5 \models \text{cond}(l_1, \text{true}, l_2 + l_3) \]

\[ \Gamma(l_1) = \text{HL} \]
\[ \Gamma(l_2) = \text{HH} \]
\[ \Gamma(l_3) = \text{HL} \]
\[ \Gamma(l_4) = \text{HH} \]
\[ \Gamma(l_5) = \text{HL} \]
First attempt

We think $T$ is secure for execution

$$\langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v \text{ if:}$$

- $\langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v \equiv T$ and

- $T$ does not contain any high provenance locations.

E.g.,

$$\langle \ldots ; \text{if } (l_1) \text{ then } l_2+l_3 \text{ else } l_4+l_5 \rangle \Rightarrow 5 \equiv \text{cond}(l_1, \text{true}, \star+l_3)$$

$$\Gamma(l_1) = \text{HL}$$
$$\Gamma(l_2) = \text{HH}$$
$$\Gamma(l_3) = \text{HL}$$
$$\Gamma(l_4) = \text{HH}$$
$$\Gamma(l_5) = \text{HL}$$

Provenance security

- $T$ satisfies **provenance security** for execution
  \[\langle l_1=v_1, \ldots, l_n=v_n \rangle \implies v \text{ if:}\]
  - $\langle l_1=v_1, \ldots, l_n=v_n \rangle \implies v \equiv T$ and
  - for any high provenance $l_i$, there is an execution
    \[\langle l_1=w_1, \ldots, l_n=w_n \rangle \implies v\text{ such that}\]
    - if $l_j$ is low security then $v_j = w_j$ and
    - $\langle l_1=w_1, \ldots, l_n=w_n \rangle \implies v \equiv T$ and
    - $l_i$ involved in $\langle l_1=v_1, \ldots, l_n=v_n \rangle \implies v \iff$
      \[l_i \text{ not involved in } \langle l_1=w_1, \ldots, l_n=w_n \rangle \implies v\]
Provenance security

- $T$ satisfies **provenance security** for execution

$$\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$$

if:

$$\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \equiv T \text{ and}$$

for any high provenance $l_i$, there is an execution

$$\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$$

such that

if $l_j$ is low security then $v_j = w_j$

and

$$\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v \equiv T \text{ and}$$

$l_i$ involved in \(\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v\) iff

$l_i$ not involved in \(\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v\)

Looks the same
Provenance security

- $T$ satisfies **provenance security** for execution
  $$\langle l_1=v_1, \ldots, l_n=v_n; \ c \rangle \Rightarrow v$$ if:
  - $$\langle l_1=v_1, \ldots, l_n=v_n; \ c \rangle \Rightarrow v \models T$$ and
  - for any high provenance $l_i$, there is an execution
    $$\langle l_1=w_1, \ldots, l_n=w_n; \ c \rangle \Rightarrow v$$ such that
      - if $l_j$ is low security then $v_j = w_j$ and
      - $$\langle l_1=w_1, \ldots, l_n=w_n; \ c \rangle \Rightarrow v \models T$$ and
      - $l_i$ involved in $$\langle l_1=v_1, \ldots, l_n=v_n; \ c \rangle \Rightarrow v$$ iff
        - $l_i$ not involved in $$\langle l_1=w_1, \ldots, l_n=w_n; \ c \rangle \Rightarrow v$$

Neither output $v$ nor provenance $T$ reveal which high provenance input locations were used.

Provenance security

- $T$ satisfies **provenance security** for execution
  
  $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ if:
  
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \iff T$ and
  - for any high provenance $l_i$, there is an execution $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$ such that
    
    - if $l_i$ is low security then $v_i=w_i$ and
    - $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v \iff T$ and
    - $l_i$ involved in $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ iff
      
      $l_i$ not involved in $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$

**E.g.,**

$\langle \ldots ; \text{ if } (l_1) \text{ then } l_2 \ + l_3 \text{ else } l_4 \ + l_5 \rangle \Rightarrow 5 \iff$

$$\begin{align*}
\Gamma(l_1) &= \text{HL} \\
\Gamma(l_2) &= \text{HH} \\
\Gamma(l_3) &= \text{HL} \\
\Gamma(l_4) &= \text{HH} \\
\Gamma(l_5) &= \text{HL}
\end{align*}$$
Provenance security

- $T$ satisfies **provenance security** for execution
  
  $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ if:
  
  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \neq T$ and
  
  for any high provenance $l_i$ there is an execution
  
  $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$ such that
  
  - if $l_i$ is low security then $v_i = w_i$ and
  
  - $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v \neq T$ and
  
  - $l_i$ involved in $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ iff
  
  - $l_i$ not involved in $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$

E.g.,

$\langle \ldots ; \text{ if } (l_1) \text{ then } l_2 + l_3 \text{ else } l_4 + l_5 \rangle \Rightarrow 5 \neq \text{cond}(l_1, true, l_2 + l_3)$

$\Gamma(l_1) = HL$

$\Gamma(l_2) = HH \quad \Gamma(l_3) = HL$

$\Gamma(l_4) = HH \quad \Gamma(l_5) = HL$
T satisfies provenance security for execution
\[ \langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v \text{ if:} \]
- \[ \langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v = T \text{ and} \]
- for any high provenance \( l_i \), there is an execution
  \[ \langle l_1=w_1, \ldots, l_n=w_n ; c \rangle \Rightarrow v \] such that
    - if \( l_i \) is low security then \( v_i = w_i \) and
    - \[ \langle l_1=w_1, \ldots, l_n=w_n ; c \rangle \Rightarrow v = T \] and
    - \( l_i \) involved in \[ \langle l_1=v_1, \ldots, l_n=v_n ; c \rangle \Rightarrow v \] iff
      \( l_i \) not involved in \[ \langle l_1=w_1, \ldots, l_n=w_n ; c \rangle \Rightarrow v \]

E.g.,
\[ \langle \ldots ; \text{ if } (l_1) \text{ then } l_2 + l_3 \text{ else } l_4 + l_5 \rangle \Rightarrow 5 \equiv \text{cond}(l_1, \text{true}, \star + l_3) \]
\[ \Gamma(l_1) = \text{HL} \]
\[ \Gamma(l_2) = \text{HH} \quad \Gamma(l_3) = \text{HL} \]
\[ \Gamma(l_4) = \text{HH} \quad \Gamma(l_5) = \text{HL} \]

**Provenance security**

- $T$ satisfies **provenance security** for execution
  
  $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ if:

  - $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v \neq T$ and
  - for any high provenance $l_i$, there is an execution
    $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$ such that

    - if $l_i$ is low security then $v_i=w_i$ and
    - $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v = T$ and
    - $l_i$ involved in $\langle l_1=v_1, \ldots, l_n=v_n ; \ c \rangle \Rightarrow v$ iff

      $l_i$ not involved in $\langle l_1=w_1, \ldots, l_n=w_n ; \ c \rangle \Rightarrow v$

**E.g.,**

$\langle \ldots ; \text{if } (l_1) \text{ then } l_2 + l_3 \text{ else } l_4 + l_5 \rangle \Rightarrow 5 \neq \text{cond}(l_1, \text{true}, \star)$

\[
\Gamma(l_1) = \text{HL} \\
\Gamma(l_2) = \text{HH} \\
\Gamma(l_3) = \text{HL} \\
\Gamma(l_4) = \text{HH} \\
\Gamma(l_5) = \text{HL}
\]
Provenance security

- $T$ satisfies **provenance security** for execution
  
  \[
  \langle l_1=v_1, \ldots, l_n=v_n \ ; \ c \rangle \Rightarrow v 
  \]

  if:

  - $\langle l_1=v_1, \ldots, l_n=v_n \ ; \ c \rangle \Rightarrow v \neq T$ and
  - for any high provenance $l_i$ there is an execution
    \[
    \langle l_1=w_1, \ldots, l_n=w_n \ ; \ c \rangle \Rightarrow v
    \]
    such that

    - if $l_i$ is low security then $v_i=w_i$ and
    - $\langle l_1=w_1, \ldots, l_n=w_n \ ; \ c \rangle \Rightarrow v \neq T$ and
    - $l_i$ involved in $\langle l_1=v_1, \ldots, l_n=v_n \ ; \ c \rangle \Rightarrow v$ iff
      
      - $l_i$ not involved in $\langle l_1=w_1, \ldots, l_n=w_n \ ; \ c \rangle \Rightarrow v$

E.g.,

\[
\langle \ldots ; \text{ if } (l_1) \text{ then } l_2 + l_3 \text{ else } l_4 + l_5 \rangle \Rightarrow 5 \equiv \text{cond}(l_1, \star, \star)
\]

\[
\begin{align*}
\Gamma(l_1) &= \text{HL} \\
\Gamma(l_2) &= \text{HH} \\
\Gamma(l_3) &= \text{HL} \\
\Gamma(l_4) &= \text{HH} \\
\Gamma(l_5) &= \text{HL}
\end{align*}
\]
Conclusion

- Need to understand provenance security, and interactions with data security
- This work: Formal definitions for provenance security
  - public data does not reveal sensitive provenance
  - public provenance does not reveal sensitive provenance
  - public provenance does not reveal sensitive data
- Practical implications:
  - determining access control for provenance
  - consistency of security policies for data and provenance
- Future work:
  - Moving from the T towards the P of TaPP