Data provenance in a distributed calculus
Motivation

- (Meta)data is almost entirely neglected in the process calculi literature
- Track data provenance both for its important applications and as an challenging exercise in modelling (meta)data. We aim at simplicity:
  - data annotations representing provenance
  - structure, interpretation and management of provenance information
  - provenance tracking
- Provenance-based security (aspects: trust + data confidentiality and privacy)
  - Example: photography competition
- The overall ambition is to underpin practical development, like trust-policy languages and protocols, and provenance-middleware
Provenance model

Annotated data

\( v : K \)
Provenance model

Annotated data

Annotated value

\( v : K \)
Provenance model

Annotated data

Annotated value

Value

Actual data

$\nu : \mathcal{K}$
Provenance model

Annotated data

Annotated value

Value

Actual data

Provenance

Meta information describing the origin of the value
Provenance model
Structure and interpretation of provenance

\[ \nu : \varepsilon ; a!k_1 ; b ? k_2 ; b!(\varepsilon; c!k_3, b ? k_4) ; \ldots \]
Provenance model
Structure and interpretation of provenance

\[ v : \varepsilon ; a!\kappa_1 ; b ? \kappa_2 ; b!(\varepsilon ; c!\kappa_3 , b ? \kappa_4 ) ; \ldots \]
Provenance model
Structure and interpretation of provenance

Value

Provenance

\[ \nu : \varepsilon ; a! \kappa_1 ; b ? \kappa_2 ; b!(\varepsilon ; c! \kappa_3 , b ? \kappa_4 ) ; \ldots \]

“Operations” that were performed on the value. They record the principals that “influenced” the value and how.
Provenance model
Structure and interpretation of provenance

$\varepsilon$ (empty provenance)
denotes value $v$ originated here
Provenance model
Structure and interpretation of provenance

\( \epsilon \) (empty provenance)
denotes value \( v \) originated here

It was sent by \( a \) on a channel with provenance \( \kappa_1 \)
Provenance model
Structure and interpretation of provenance

\( v : \varepsilon ; a!\kappa_1 ; b?\kappa_2 \)

\( \varepsilon \) (empty provenance)
denotes value \( v \) originated here

It was sent by \( a \) on a channel with provenance \( \kappa_1 \)

Was then received by \( b \) on a channel with provenance \( \kappa_2 \)
Provenance model
Structure and interpretation of provenance

\[ \nu : \varepsilon ; a! \kappa_1 ; b? \kappa_2 ; b!(\varepsilon; c! \kappa_3, b? \kappa_4) ; \ldots \]

It was sent by \( a \) on a channel with provenance \( \kappa_1 \)

Was then received by \( b \) on a channel with provenance \( \kappa_2 \)

\( \varepsilon \) (empty provenance) denotes value \( \nu \) originated here

And then sent by \( b \) on a channel that \( b \) received from \( c \)…
Confidentiality in provenance systems

- Data may be public, yet its provenance confidential, or vice versa
- Principals who may access data are not necessarily the same as those who may access its provenance
- In general, fine grained access control over provenance “histories” is needed as different parts of it have different sensitivity

Security requirements of data ≠ Security requirements of its provenance
Hiding provenance trees

Example: photography competition

c: Contestant
a: Administrator
j: Judge
Hiding provenance trees
Example: photography competition

c: Contestant
a: Administrator
j: Judge
Hiding provenance trees
Example: photography competition

c: Contestant
a: Administrator
j: Judge
Hiding provenance trees
Example: photography competition

entry : ε; c!κ
entry : ε; c!κ; a?κ'; a!κ'
entry : ε; c!κ; a?κ'; a!κ'; j?κ"; j!κ"" 

score : ε; j!κ"

c: Contestant
a: Administrator
j: Judge
Hiding provenance trees
Example: photography competition

c: Contestant
a: Administrator
j: Judge
Hiding provenance trees

Example: photography competition

c

entry : ε; c!κ_s

entry : ε; c!κ_s; a?κ'; a!κ'

entry : ε; c!κ_s; a?κ'; a!κ'; j?κ''; j!κ''; a?κ''; a!κ''

score : ε; j!κ''; a?κ''; a!κ''

j

entry : ε; c!κ_s; a?κ'; a!κ'

entry : ε; c!κ_s; a?κ'; a!κ'; j?κ''; j!κ''

score : ε; j!κ''

a

entry : ε; c!κ_s; a?κ'; a!κ'; j?κ''; j!κ''; a?κ''; a!κ''

score : ε; j!κ''
Hiding provenance trees

Example: photography competition

entry: ε; c!κ
score: ε; j!κ

entry: ε; c!κ; a?κ'; a!κ'

entry: ε; c!κ; a?κ'; j?κ''; j!κ''; a?κ'; a!κ'

entry: ε; c!κ; a?κ'; a!κ'; j?κ''; j!κ''

score: ε; j!κ''

Hidden from j
Hiding provenance trees
Example: photography competition

**Entry:** c

**Entry:** j

**Entry:** a

**Score:** c

**Score:** j

**Score:** a

**Hidden from j:**
- c
- a

**Hidden from c:**
- j
- a
Confidentiality in provenance systems
a promising approach

✦ One value, multiple **views**

✦ Different principals have different views of the same provenance list based on their privileges

\[
\text{entry: } \varepsilon; c!\kappa_s; a ?\kappa'_s; a!\kappa'_r; j ?\kappa''_r; j!\kappa''_n; a ?\kappa'_n; a!\kappa'_m
\]
Confidentiality in provenance systems
a promising approach

✧ One value, multiple views

✧ Different principals have different views of the same provenance list based on their privileges

entry: $\varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$
Confidentiality in provenance systems
a promising approach

✧ One value, multiple views

✧ Different principals have different views of the same provenance list based on their privileges

\[ \text{entry} : \varepsilon; c! \kappa_s; a ? \kappa'_s; a! \kappa'_r; j ? \kappa''_r; j! \kappa''_n; a ? \kappa'_n; a! \kappa'_m \]
Confidentiality in provenance systems
a promising approach

✦ One value, multiple views

✦ Different principals have different views of the same provenance list based on their privileges

entry: $\varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$
Confidentiality in provenance systems
a promising approach

✦ One value, multiple views

✦ Different principals have different views of the same provenance list based on their privileges

entry: ε; c!κ; a?κ'; a!κ'; j?κ''; j!κ''; a?κ''; a!κ''