**790-132: Principled Security** **Spring 2017**

# A Method for Verifying Privacy-Type Properties

*Lecturer: Lee Barnett* *Date: March 30*

## 1 Introduction

- Motivation: need to be able to formally verify privacy protocols.
- Goal: focus on two properties which stipulate that a user can:
  (a) make multiple uses of a service without others being able to link them together (*unlinkability*),
  (b) use a service without disclosing their identity (*anonymity*).
- But (a) and (b) are not definable as traces properties, so typically formulate them as equivalence relations. Problem: these are hard to check automatically (they do not scale well).
- Approach: devise sufficient (easy to check) conditions which imply (a) and (b) hold, for a large class of 2-party protocols

## 2 Model

- Security protocols are modeled using a process calculus:
  - participants as processes,
  - communication between participants as elements of a term algebra.

### 2.1 Term algebra

- $\mathcal{T}(F, A)$ – freely generated algebraic structure over set $A$ and signature $F$ (i.e., the initial $F$-algebra).
- $\Sigma = \Sigma_c \sqcup \Sigma_d$ – signature; $\Sigma_c$ of *constructors*, $\Sigma_d$ of *destructors*
- $\mathcal{N}$ – set of *names*; $\mathcal{X}$, $\mathcal{W}$ – sets of *variables* ($\mathcal{X} \cap \mathcal{W} = \emptyset$)
- *Constructor term* $u \in \mathcal{T}(\Sigma_c, \mathcal{N} \cup \mathcal{X})$ is a *message* if $u$ is ground
- $\mathcal{T}(\Sigma_c, \mathcal{N} \cup \mathcal{X})$ subject to an equational theory $E$ (i.e., congruence $=_E$; gen.'d by eq.'s over $\mathcal{T}(\Sigma_c, \mathcal{X})$)
- *Computation relation* $\Downarrow \subset \mathcal{T}(\Sigma, \mathcal{N}) \times \mathcal{T}(\Sigma_c, \mathcal{N})$
  - rel. each ground term to at most one (wrt $E$) message (if ground term $t$ isn't rel. to any message then say *computation fails*, write $t \not\Downarrow$)
  - Define $\Downarrow$ via a rewrite relation $\to$ (which is confluent and terminating wrt $E$).
- Example: let $\Sigma = \{(\text{enc}, 2), (\text{dec}, 2), (\langle\ \rangle, 2), (\pi_1, 1), (\pi_2, 1), (\oplus, 2), (0, 0), (\text{eq}, 2), (\text{ok}, 0)\}$
  - $\Sigma_c = \{\text{enc}, \langle\ \rangle, \oplus, 0, \text{ok}\}$, $\Sigma_d = \Sigma \setminus \Sigma_c$

- $x \oplus 0 = x$, $x \oplus x = 0$, $(x \oplus y) \oplus z = x \oplus (y \oplus z)$, $x \oplus y = y \oplus x$
- $\mathrm{dec}(\mathrm{enc}(x, y), y) \to x$, $\mathrm{eq}(x, x) \to \mathrm{ok}$, $\pi_i(\langle x_1, x_2\rangle) \to x_i$

- Split $\Sigma$ into $\Sigma_{\mathrm{pub}}$, $\Sigma_{\mathrm{priv}}$. Attacker builds messages applying $f \in \Sigma_{\mathrm{pub}}$ to terms avail. through $\mathcal{W}$
- I.e. attacker computations are terms of $\mathcal{T}(\Sigma_{\mathrm{pub}}, \mathcal{W})$ called *recipes*

### 2.2 Process calculus

- $\mathcal{C}$ – set of (public) communication channels
- Syntax for processes: $P, Q := 0 \mid \mathtt{in}(c, x).P \mid \mathtt{out}(c, u).P \mid \mathtt{let}\ \overline{x} = \overline{v}\ \mathtt{in}\ P\ \mathtt{else}\ Q \mid P|Q \mid !P \mid \nu n.P$
  - $\mathtt{let}$ construction: if ($\exists$ messages $\overline{u}$ s.t. $\overline{v} \Downarrow \overline{u}$) then $P[\overline{u}/\overline{x}]$ executes, otherwise $Q$ executes
- (Operational) semantics for processes: labeled transition system over *configurations* $K = (\mathcal{P}; \phi)$:
  - $\mathcal{P}$ – multiset of ground processes (null processes implicitly removed)
  - $\phi = \{w_1 \mapsto u_1, \ldots, w_n \mapsto u_n\}$ –*frame*, represents messages known by attacker
- $\xrightarrow{\alpha}$ – transition relation, rules are fairly intuitive. $\xrightarrow{\alpha_1 \ldots \alpha_n}$ – transitive closure of $\xrightarrow{\alpha}$.
- Example: RFID protocol. Using example term algebra above, $P := \nu k.\ (\nu n_I.P_I \mid \nu n_R.P_R)$, where:
  - $P_I = \mathtt{out}(c_I, n_I).\mathtt{in}(c_I, x_1).\mathtt{let}\ \ x_2, x_3 = \mathrm{eq}(n_i, \pi_1(u)), \pi_2(u)\ \ \mathtt{in}\ \mathtt{out}(c_I, \mathrm{enc}(\langle x_3, n_I\rangle, k))$
  - $P_R = \mathtt{in}(c_R, y_1).\mathtt{out}(c_R, \mathrm{enc}(\langle y_1, n_R\rangle, k)).\mathtt{in}(c_R, y_2).\mathtt{let}\ \ y_3 = \mathrm{eq}(y_2, \mathrm{enc}(\langle n_R, y_1\rangle, k))\ \ \mathtt{in}\ \ 0$
- Normal execution of one session of protocol: $P \xrightarrow{\mathrm{tr}} (\emptyset; \phi_0)$, where:
  - $\mathrm{tr} = \tau.\tau.\tau.\tau.\mathtt{out}(c_I, w_1).\mathtt{in}(c_R, w_1).\mathtt{out}(c_R, w_2).\mathtt{in}(c_I, w_2).\tau_{\mathrm{then}}.\mathtt{out}(c_I, w_3).\mathtt{in}(c_R, w_3).\tau_{\mathrm{then}}$
  - $\phi_0 = \{w_1 \mapsto n'_I, w_2 \mapsto \mathrm{enc}(\langle n'_I, n'_R\rangle, k'), w_3 \mapsto \mathrm{enc}(\langle n'_R, n'_I\rangle, k')\}$
- static equivalence $\phi \sim \phi'$ between frames
- trace equivalence $K \approx K'$ between configurations

## 3 Protocols & properties

- Consider 2-party protocols, two roles: *initiator* and *responder*
- Initiator is a ground process $P_I ::= 0 \mid l\mathtt{:out}(c, u).P_R$ (where $l \in \mathcal{L}$ is a syntactic label)
- Responder is $P_R ::= 0 \mid \mathtt{in}(c, y).\mathtt{let}\ \overline{x} = \overline{v}\ \mathtt{in}\ P_I \mid \mathtt{in}(c, y).\mathtt{let}\ \overline{x} = \overline{v}\ \mathtt{in}\ P_I\ \mathtt{else}\ l\mathtt{:out}(c', u')$
- $\Pi = (\overline{k}, \overline{n_I}, \overline{n_R}, \mathcal{I}, \mathcal{R})$ – protocol
- $\mathcal{M}_\Pi :=!\nu\overline{k}.(!\nu\overline{n_I}.\mathcal{I} \mid !\nu\overline{n_R}.\mathcal{R})$ – rep. arbitrary number of agents, arbitrary number of sessions
- $\mathcal{S}_\Pi :=!\nu\overline{k}.(\nu\overline{n_I}.\mathcal{I} \mid \nu\overline{n_R}.\mathcal{R})$ – rep. arbitrary number of agents, at most one session each

### 3.1 Unlinkability

- $\Pi$ preserves unlinkability wrt $\mathcal{I}$ and $\mathcal{R}$ if $\mathcal{M}_\Pi \approx \mathcal{S}_\Pi$

### 3.2 Anonymity

- $\overline{id} \subseteq \overline{k}$ – set of identities
- $\mathcal{M}_\Pi^{\text{id}} := \mathcal{M}_\Pi \mid \nu\overline{k}.(!\nu\overline{n_I}.\mathcal{I}_0 \mid !\nu\overline{n}_R.\mathcal{R}_0)$ –process where $\mathcal{I}_0, \mathcal{R}_0$– new agents, disclosed their identity
- $\Pi$ preserves anonymity wrt $\overline{id}$ if $\mathcal{M}_\Pi \approx \mathcal{M}_\Pi^{\text{id}}$.

## 4 Two conditions

- $A(\overline{k}, \overline{n})$ – *annotation* $(A \in \{I, R\})$
- $\tau$, $\alpha[a]$ –annotated action, $P[a]$ –annotated process
- Annotated semantics for processes:
  - Agents in the multiset of processes– each annotated by its identity
  - Actions (other than $\tau$)– each annotated with the identity of the agent responsible for it

### 4.1 Frame opacity

- In any execution, outputs are indistinguishable from randomness
- Define $[\cdot]^{\text{ideal}} : \mathcal{T}(\Sigma_c, \mathcal{N}) \to \mathcal{T}(\Sigma_t, \{\Box\})$ by
  - $[u]^{\text{ideal}} = f([u_1]^{\text{ideal}}, \dots, [u_n]^{\text{ideal}})$ if $u =_E f(u_1, \dots, u_n)$ for $f \in \Sigma_t$, or
  - $[u]^{\text{ideal}} = \Box$ otherwise;
  - $[u]^{\text{ideal}} = [v]^{\text{ideal}}$ whenever $u =_E v$.
- $[u]^{\text{nonce}}$ – the set $\text{inst}([u]^{\text{ideal}})$ of all *concretizations* of $[u]^{\text{ideal}}$.
- Condition: $\Pi$ ensures frame-opacity if, for any $(\mathcal{M}_\Pi^{\text{id}}; \emptyset) \xrightarrow{\text{ta}} (Q; \emptyset)$:
  - $\exists \psi \in [\phi]^{\text{nonce}}$ s.t. $\phi \sim \psi$, and
  - $\forall w_i, w_j \in \text{dom}(\phi)$ with the same label, $[w_i\phi]^{\text{ideal}} = [w_j\phi]^{\text{ideal}}$.

### 4.2 Well-authentication

- A conditional `let` $\overline{x} = \overline{v}$ `in` $P$ `else` $Q$ is *safe* if $\overline{v} \in \mathcal{T}(\Sigma_{\text{pub}}, \{x_1, \dots, x_n\} \cup \{u_1, \dots, u_n\})$
- Agents $A_1(\overline{k_1}, \overline{n_1})$, $A_2(\overline{k_2}, \overline{n_2})$ are *associated* in $(\text{ta}, \phi)$ if
  - $((A_1 \neq A_2)$ and $\overline{k_1} = \overline{k_2})$;
  - the interaction ta between them is honest for $\phi$.
- $\Pi$ is well-authenticating if, for any $(\mathcal{M}_\Pi^{\text{id}}; \emptyset) \xrightarrow{\text{ta}.\tau_{\text{then}}[A(\overline{k}, \overline{n_1})]} (\mathcal{P}; \phi)$, either
  - the last action was a safe conditional, or
  - $\exists A', \overline{n_2}$ s.t. $A(\overline{k}, \overline{n_1}), A'(\overline{k}, \overline{n_2})$ are associated in $(\text{ta}, \phi)$ and $A'(\overline{k}, \overline{n_2})$ not assoc. to anything else

## 5 Main result

- $\Pi$ – protocol with identity names $\overline{id} \subseteq \overline{k}$.
- **$\Pi$ is w.-a. and ensures frame opacity $\Rightarrow$ $\Pi$ ensures unlinkability and anonymity wrt $\overline{id}$**

## References

[1] L. Hirschi, D. Baelde, and S. Delaune, "A Method for Verifying Privacy-Type Properties: The Unbounded Case," *IEEE Symposium on Security and Privacy*, 2016, pp. 564–581.