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

• $T(F,A)$ – freely generated algebraic structure over set $A$ and signature $F$ (i.e., the initial $F$-algebra).
• $\Sigma = \Sigma_c \cup \Sigma_d$ – signature; $\Sigma_c$ of constructors, $\Sigma_d$ of destructors
• $N$ – set of names; $\mathcal{X}, \mathcal{W}$ – sets of variables ($\mathcal{X} \cap \mathcal{W} = \emptyset$)
• Constructor term $u \in T(\Sigma_c, N \cup \mathcal{X})$ is a message if $u$ is ground
• $T(\Sigma_c, N \cup \mathcal{X})$ subject to an equational theory $E$ (i.e., congruence $=_E$; gen.’d by eq.’s over $T(\Sigma_c, \mathcal{X})$)
• Computation relation $\Downarrow \subset T(\Sigma, N) \times T(\Sigma_c, N)$
  – 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 $\rightarrow$ (which is confluent and terminating wrt $E$).
• Example: let $\Sigma = \{(\text{enc}, 2), (\text{dec}, 2), (\text{leaf}, 2), (\pi_1, 1), (\pi_2, 1), (\oplus, 2), (0, 0), (eq, 2), (ok, 0)\}$
  – $\Sigma_c = \{\text{enc}, ( ), \oplus, 0, \text{ok}\}$, $\Sigma_d = \Sigma \setminus \Sigma_c$
\[ x \oplus 0 = x, \quad x \oplus -x = 0, \quad (x \oplus y) \oplus z = x \oplus (y \oplus z), \quad x \oplus y = y \oplus x \]
\[ \text{dec(enc}(x,y), y) \rightarrow x, \quad \text{eq}(x, x) \rightarrow \text{ok}, \quad \pi_i((x_1, x_2)) \rightarrow x_i \]

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

### 2.2 Process calculus

- \( \mathcal{C} \) – set of (public) communication channels

- Syntax for processes: \( P, Q := 0 | \text{in}(c, x).P | \text{out}(c, u).P | \text{let } \overline{x} = \overline{y} \text{ in } P \text{ else } Q \text{ | } P \mid Q \mid !P \mid \nu n.P \)

- let construction: if \( \exists \) messages \( \overline{w} \text{ s.t. } \overline{x} \downarrow \overline{w} \) then \( P[\overline{x} \mapsto \overline{w}] \) executes, otherwise \( Q \) executes

- (Operational) semantics for processes: labeled transition system over configurations \( K = (P; \phi) \):

  - \( \mathcal{P} \) – 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, \( P := \nu k. \,(\nu n_1.P_l | \nu n_R.P_R) \), where:

  - \( P_l = \text{out}(c_l, n_I).\text{in}(c_l, x_1).\text{let } x_2, x_3 = \text{eq}(n_I, \pi_1(u)), \pi_2(u) \text{ in } \text{out}(c_l, \text{enc}((x_3, n_I), k)) \)
  - \( P_R = \text{in}(c_R, y_1).\text{out}(c_R, \text{enc}((y_1, n_R), k)).\text{in}(c_R, y_2).\text{let } y_3 = \text{eq}(y_2, \text{enc}(n_R, y_1), k) \text{ in } 0 \)

- Normal execution of one session of protocol: \( P \xrightarrow{\text{tr}} (\emptyset; \phi_0) \), where:

  - \( \text{tr} = \tau.\tau.\tau.\tau.\text{out}(c_l, w_1).\text{in}(c_R, w_1).\text{out}(c_R, w_2).\text{in}(c_l, w_2).\tau_{\text{then}}.\text{out}(c_l, w_3).\text{in}(c_R, w_3).\tau_{\text{then}} \)
  - \( \phi_0 = \{w_1 \mapsto n'_I, w_2 \mapsto \text{enc}(n'_I, n'_R), k'), w_3 \mapsto \text{enc}(n'_R, n'_I), k') \} \)

- static equivalence \( \phi \sim \phi' \) between frames
- trace equivalence \( K \approx K' \) between configurations

### 3 Protocols & properties

- Consider 2-party protocols, two roles: \textit{initiator} and \textit{responder}

- Initiator is a ground process \( P_I := 0 \mid l:\text{out}(c, u).P_R \) (where \( l \in \mathcal{C} \) is a syntactic label)

- Responder is \( P_R := 0 \mid \text{in}(c, y).\text{let } \overline{x} = \overline{y} \text{ in } P_I \mid \text{in}(c, y).\text{let } \overline{x} = \overline{y} \text{ in } P_I \text{ else } l:\text{out}(c', u') \)

- \( \Pi = (\mathcal{R}, \nu n_I, \nu n_R, I, \mathcal{R}) \) – protocol

- \( M_{\Pi} := !\nu \mathcal{R}.(\nu n_I, I, R) \mid !\nu n_R, R \) – rep. arbitrary number of agents, arbitrary number of sessions

- \( S_{\Pi} := !\nu \mathcal{R}.(\nu n_I, I, R) \mid \nu n_R, R \) – rep. arbitrary number of agents, at most one session each

### 3.1 Unlinkability

- \( \Pi \) preserves unlinkability wrt \( I \) and \( R \) if \( M_{\Pi} \approx S_{\Pi} \)
3.2 Anonymity

- \( \overline{id} \subseteq \overline{k} \) – set of identities
- \( M^\Pi_\Pi := M_\Pi \mid \nu \overline{\pi}_I, I_0 \mid \nu \overline{\pi}_R, R_0 \) – process where \( I_0, R_0 \) – new agents, disclosed their identity
- \( \Pi \) preserves anonymity wrt \( \overline{id} \) if \( M^\Pi \approx M^\Pi_\Pi \).

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 \( [\cdot]^{\text{ideal}} : \mathcal{T}(\Sigma_c, \mathcal{N}) \rightarrow \mathcal{T}(\Sigma_t, \{\square\}) \) by
  - \( [u]^{\text{ideal}} = f([u_1]^{\text{ideal}}, \ldots, [u_n]^{\text{ideal}}) \) if \( u = E f(u_1, \ldots, u_n) \) for \( f \in \Sigma_t \), or
  - \( [u]^{\text{ideal}} = \square \) otherwise;
  - \( [u]^{\text{ideal}} = [v]^{\text{ideal}} \) whenever \( u = E v. \)
- \( [u]^{\text{nonce}} \) – the set inst(\( [u]^{\text{ideal}} \)) of all concretizations of \( [u]^{\text{ideal}} \).
- Condition: \( \Pi \) ensures frame-opacity if, for any \( (M^\Pi_\Pi; \emptyset) \xrightarrow{ta} (Q; \emptyset) \):
  - \( \exists \psi \in [\phi]^{\text{nonce}} \) 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 \( \overline{v} \in \mathcal{T}(\Sigma_{\text{pub}}, \{x_1, \ldots, x_n\} \cup \{u_1, \ldots, u_n\}) \)
- Agents \( A_1(\overline{k}_1, \overline{\pi}_1), A_2(\overline{k}_2, \overline{\pi}_2) \) are associated in \( (ta, \phi) \) 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 \( (M^\Pi_\Pi; \emptyset) \xrightarrow{ta, \tau\text{then}[A(\overline{E}, \overline{\pi})]} (P; \phi) \), either
  - the last action was a safe conditional, or
  - \( \exists A', \overline{\pi}_2 \) s.t. \( A(\overline{k}, \overline{\pi}_1), A'(\overline{k}, \overline{\pi}_2) \) are associated in \( (ta, \phi) \) and \( A'(\overline{k}, \overline{\pi}_2) \) not assoc. to anything else.
5 Main result

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

References