DTU

DIGISEC Seminar - September 9, 2021

Laouen Fernet - Research Assistant in Formal Methods - lpkf@dtu.dk

Deciding a Fragment of (α, β)-Privacy

Automated reasoning about privacy in security protocols

Automated reasoning about privacy in security protocols

Produce a program

Automated reasoning about privacy in security protocols

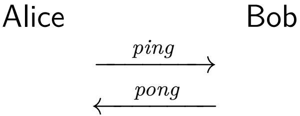
Produce a program

Control information

Automated reasoning about privacy in security protocols

Produce a program

Control information



Privacy

Relevant in many fields, a security goal of its own:

- Electronic voting, digital health information, mobile payments...
- Distributed systems in general.
- More than just secrecy.

Privacy

Relevant in many fields, a security goal of its own:

- Electronic voting, digital health information, mobile payments...
- Distributed systems in general.
- More than just secrecy.

De facto standard = indistinguishability

- Given two possible worlds, can they be distinguished?
- Automated verification is difficult.
- Specification of goals is not intuitive.
- There is no guarantee that every privacy aspect has been covered.

Novel approach

(α, β)-privacy[1] = logical approach with many advantages:

- declarative and intuitive
- recast privacy as a reachability problem[2]
- decidable fragments: possibility for automated verification

[1] Mödersheim S., Viganò L.: Alpha-Beta Privacy. ACM Trans. Priv. Secur. 22(1), 1–35 (2019).

[2] Gondron, S., Mödersheim, S., Viganò, L.: Privacy as Reachability. Tech. rep., DTU (2021), `http://www2.compute.dtu.dk/~samo/abg.pdf`

Table of Contents

Table of Contents

Grammar

$$\begin{aligned}
\langle \textit{Term} \rangle \quad &::= \langle \textit{Variable} \rangle \mid \langle \textit{Function} \rangle(\langle \textit{Term} \rangle, \ldots, \langle \textit{Term} \rangle) \\
\langle \textit{Formula} \rangle \quad &::= \langle \textit{Term} \rangle = \langle \textit{Term} \rangle \\
&\mid \quad \langle \textit{Relation} \rangle(\langle \textit{Term} \rangle, \ldots, \langle \textit{Term} \rangle) \\
&\mid \quad \neg \langle \textit{Formula} \rangle \\
&\mid \quad \langle \textit{Formula} \rangle \wedge \langle \textit{Formula} \rangle \\
&\mid \quad \exists \langle \textit{Variable} \rangle . \langle \textit{Formula} \rangle
\end{aligned}$$

Table of Contents

Frame

Frames encode the knowledge of messages based on the protocol specification.

$$F = \{\!|\, \mathsf{l}_1 \mapsto t_1, \ldots, \mathsf{l}_k \mapsto t_k \,|\!\}$$

l_i: distinguished constant called *label*

t_i: term that does not contain any l_i

Example: $F = \{\!|\, \mathsf{l}_1 \mapsto ping, \mathsf{l}_2 \mapsto pong \,|\!\}$

Recipes

Frames allow to reason about actions taken and not simply messages themselves.

Set of *recipes* = least set that contains $\mathsf{l}_1, \ldots, \mathsf{l}_k$ and that is closed under cryptographic operators

$F\{\!|\, r \,|\!\}$: application of recipe r to frame F

Static equivalence

F_1 and F_2 with the same domain are *statically equivalent*, written $F_1 \sim F_2$, if the intruder cannot distinguish them.

$$\forall (r_1, r_2), F_1\{\!| r_1 |\!\} \approx F_1\{\!| r_2 |\!\} \iff F_2\{\!| r_1 |\!\} \approx F_2\{\!| r_2 |\!\}$$

Static equivalence

$\mathcal{F}_1$ and $\mathcal{F}_2$ with the same domain are *statically equivalent*, written $\mathcal{F}_1 \sim \mathcal{F}_2$, if the intruder cannot distinguish them.

$$\forall (r_1, r_2), \mathcal{F}_1\{\!|\, r_1 \,|\!\} \approx \mathcal{F}_1\{\!|\, r_2 \,|\!\} \iff \mathcal{F}_2\{\!|\, r_1 \,|\!\} \approx \mathcal{F}_2\{\!|\, r_2 \,|\!\}$$

Example:

$$\mathcal{F}_1 = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, t_1), \mathsf{l}_2 \mapsto k, \mathsf{l}_3 \mapsto t_1 \,|\!\}$$
$$\mathcal{F}_2 = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, t_2), \mathsf{l}_2 \mapsto k, \mathsf{l}_3 \mapsto t_2 \,|\!\}$$

$\mathcal{F}_1 \sim \mathcal{F}_2$ because, even though $t_1 \not\approx t_2$, there is no way to distinguish the frames.

Table of Contents

Idea

Formula α: high-level information which is voluntarily disclosed, based on Σ_0

Σ_0 contains only non-technical information

Formula β: includes the technical information, e.g., cryptographic messages exchanged during the execution of the protocol

$\Sigma_0 \subsetneq \Sigma$

Violation of privacy: logically deriving information from β that does not follow from α alone

Table of Contents

Message-analysis problem

Substitution θ: a model of α (interpretation of symbols making the formula true)

Example: $\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$

Message-analysis problem

Substitution θ: a model of α (interpretation of symbols making the formula true)

Example: $\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$

$struct = \{\!| \mathsf{l}_1 \mapsto t_1, \ldots, \mathsf{l}_k \mapsto t_k |\!\}$: the specification of the protocol (structural knowledge)

$concr = \theta(struct)$: one execution of the protocol (concrete knowledge)

Message-analysis problem

Substitution θ: a model of α (interpretation of symbols making the formula true)

Example: $\alpha \equiv x, y, z \in \{0, 1\} \land x + y + z = 1 \qquad \theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$

$struct = \{\!| \, \mathsf{l}_1 \mapsto t_1, \ldots, \mathsf{l}_k \mapsto t_k \, |\!\}$: the specification of the protocol (structural knowledge)

$concr = \theta(struct)$: one execution of the protocol (concrete knowledge)

$\beta \equiv MsgAna(\alpha, struct, \theta)$: knowledge of $\alpha, struct, concr$ and $struct \sim concr$

Table of Contents

Destructors

Destructors are functions used to decrypt (= decompose) terms. Modern cryptographic primitives allow to check if decryption works.

Example: $t = \mathsf{scrypt}(k, x)$
$\longrightarrow$ can decrypt with k
$\longrightarrow$ cannot decrypt with k'

Destructor theory

- $\Sigma_{pub} \subseteq \Sigma_f$: public functions
- E: algebraic equations of the form $\mathsf{destr}(k, \mathsf{constr}(t_1, \ldots, t_n)) = t_i$ (where $i \in \{1, \ldots, n\}, fv(k) \subseteq fv(t_1, \ldots, t_n)$) and symbols in E are disjoint from Σ_0)

Example cryptographic operators

Constructors	Destructors	Properties
pub, priv		
crypt	dcrypt	$\mathsf{dcrypt}(\mathsf{priv}(s), \mathsf{crypt}(\mathsf{pub}(s), r, t)) = t$
sign	retrieve	$\mathsf{retrieve}(\mathsf{pub}(s), \mathsf{sign}(\mathsf{priv}(s), t)) = t$
scrypt	dscrypt	$\mathsf{dscrypt}(k, \mathsf{scrypt}(k, t)) = t$
pair	$\mathsf{proj}_1, \mathsf{proj}_2$	$\mathsf{proj}_1(\mathsf{pair}(t_1, t_2)) = t_1$ $\mathsf{proj}_2(\mathsf{pair}(t_1, t_2)) = t_2$
h		

Table: Example set Σ_{op}

Destructor theory

Example: $t = \mathsf{scrypt}(k, x)$

$\longrightarrow \mathsf{dscrypt}(k, \mathsf{scrypt}(k, x)) \approx x$

$\longrightarrow \mathsf{dscrypt}(k', \mathsf{scrypt}(k, x)) \approx \mathsf{error}$

Table of Contents

Frame with shorthands

Frames with shorthands extend the previous definition of frames.

$$F' = \{\!| \, \mathsf{l}_1 \mapsto t_1, \ldots, \mathsf{l}_k \mapsto t_k, \mathsf{m}_1 \mapsto s_1, \ldots, \mathsf{m}_n \mapsto s_n \, |\!\}$$

- $F = \{\!| \, \mathsf{l}_1 \mapsto t_1, \ldots, \mathsf{l}_k \mapsto t_k \, |\!\}$: frame
- m_j: recipes over the l_i
- $F\{\!| \, \mathsf{m}_j \, |\!\} \approx s_j$
- $\mathsf{m}_1 \mapsto s_1, \ldots, \mathsf{m}_n \mapsto s_n$: shorthands

Frame with shorthands

Example:

$$F = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k,t), \mathsf{l}_2 \mapsto k \,|\!\}$$
$$F' = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k,t), \mathsf{l}_2 \mapsto k, \mathsf{m}_1 \mapsto t \,|\!\}$$

$$\mathsf{m}_1 = \mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1)$$

Frame with shorthands

Example:

$$F = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k,t), \mathsf{l}_2 \mapsto k \,|\!\}$$
$$F' = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k,t), \mathsf{l}_2 \mapsto k, \mathsf{m}_1 \mapsto t \,|\!\}$$

$$\mathsf{m}_1 = \mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1)$$

$$F\{\!|\, \mathsf{m}_1 \,|\!\} = F\{\!|\, \mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1) \,|\!\} = \mathsf{dscrypt}(k, \mathsf{scrypt}(k,t))$$

Frame with shorthands

Example:

$$\mathcal{F} = \{\!\mid \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, t), \mathsf{l}_2 \mapsto k \mid\!\}$$
$$\mathcal{F}' = \{\!\mid \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, t), \mathsf{l}_2 \mapsto k, \mathsf{m}_1 \mapsto t \mid\!\}$$

$$\mathsf{m}_1 = \mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1)$$

$$\mathcal{F}\{\!\mid \mathsf{m}_1 \mid\!\} = \mathcal{F}\{\!\mid \mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1) \mid\!\} = \mathsf{dscrypt}(k, \mathsf{scrypt}(k, t)) \approx t$$

Table of Contents

Illustration

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$

$$struct = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) \,|\!\}$$

$$concr = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0) \,|\!\}$$

$$\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \beta \equiv MsgAna(\alpha, struct, \theta)$$

Intruder deduction:

Illustration

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z)\ |\!\}$$
$$concr = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0)\ |\!\}$$

$$\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \beta \equiv MsgAna(\alpha, struct, \theta)$$

Intruder deduction:

1. $concr\{\!|\ \mathsf{l}_1\ |\!\} \approx concr\{\!|\ \mathsf{l}_3\ |\!\}$: two messages are equal at the concrete level

Illustration

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$

$$struct = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z)\ |\!\}$$

$$concr = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0)\ |\!\}$$

$$\alpha \equiv x, y, z \in \{0, 1\} \land x + y + z = 1 \qquad \beta \equiv MsgAna(\alpha, struct, \theta)$$

Intruder deduction:

1. $concr\{\!|\ \mathsf{l}_1\ |\!\} \approx concr\{\!|\ \mathsf{l}_3\ |\!\}$: two messages are equal at the concrete level
2. $struct\{\!|\ \mathsf{l}_1\ |\!\} \approx struct\{\!|\ \mathsf{l}_3\ |\!\}$: they must also be equal at the structural level

Illustration

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) \,|\!\}$$
$$concr = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0) \,|\!\}$$

$$\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \beta \equiv MsgAna(\alpha, struct, \theta)$$

Intruder deduction:

1. $concr\{\!|\, \mathsf{l}_1 \,|\!\} \approx concr\{\!|\, \mathsf{l}_3 \,|\!\}$: two messages are equal at the concrete level
2. $struct\{\!|\, \mathsf{l}_1 \,|\!\} \approx struct\{\!|\, \mathsf{l}_3 \,|\!\}$: they must also be equal at the structural level
3. $x = z$: violation of privacy!

Table of Contents

Composition in a structural frame

Three methods to compose a term:

1. Try to use labels with a corresponding substitution.
2. If the term is a variable, use the true value $\theta(x)$ (a constant) as a recipe with the substitution $[x \mapsto \theta(x)]$.
3. If the top-level is a public function, try to compose all arguments (and combine the substitutions).

Composition in a structural frame

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z)\ |\!\}$$
$$concr = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0)\ |\!\}$$

Composition in a structural frame

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$

$$struct = \{\!| \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) |\!\}$$

$$concr = \{\!| \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0) |\!\}$$

$$composeUnder(\theta, struct, \mathsf{scrypt}(k, x)) =$$

Composition in a structural frame

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z)\ |\!\}$$
$$concr = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0)\ |\!\}$$

$$composeUnder(\theta, struct, \mathsf{scrypt}(k, x)) = \{(\mathsf{l}_1, \varepsilon),$$

Composition in a structural frame

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!| \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) |\!\}$$
$$concr = \{\!| \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0) |\!\}$$

$$composeUnder(\theta, struct, \mathsf{scrypt}(k, x)) = \{(\mathsf{l}_1, \varepsilon), (\mathsf{l}_2, [x \mapsto y]),$$

Composition in a structural frame

Example:

$$\theta = [x \mapsto \mathsf{0}, y \mapsto \mathsf{1}, z \mapsto \mathsf{0}]$$

$$struct = \{\!\mid \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) \mid\!\}$$

$$concr = \{\!\mid \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, \mathsf{0}), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, \mathsf{1}), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, \mathsf{0}) \mid\!\}$$

$$composeUnder(\theta, struct, \mathsf{scrypt}(k, x)) = \{(\mathsf{l}_1, \varepsilon), (\mathsf{l}_2, [x \mapsto y]), (\mathsf{l}_3, [x \mapsto z])\}$$

Composition in a structural frame

Example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z) \,|\!\}$$
$$concr = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0) \,|\!\}$$

$$composeUnder(\theta, struct, \mathsf{scrypt}(k, x)) = \{(\mathsf{l}_1, \varepsilon), (\mathsf{l}_2, [x \mapsto y]), (\mathsf{l}_3, [x \mapsto z])\}$$

The intruder knows three ways to compose $\mathsf{scrypt}(k, x)$ (substitution = constraints for the recipe to work).

Recipes may generate different terms in $concr = \theta(struct)$!

Table of Contents

Idea

- We know how to find recipes with *composition only*.
- We want to *all* generable terms using only composition.

$\longrightarrow$ Thus we need to perform *analysis steps*: decrypt messages, open signed messages, deserialize etc.

Analysis of a structural frame

Analysis steps: we check if we can decrypt terms in the frame.

Analysis of a structural frame

Analysis steps: we check if we can decrypt terms in the frame.

- If the decryption fails in $concr$, i.e., the key cannot be composed, no new terms can be added. But if the key can be composed in $struct$, we can exclude some models.

Analysis of a structural frame

Analysis steps: we check if we can decrypt terms in the frame.

- If the decryption fails in $concr$, i.e., the key cannot be composed, no new terms can be added. But if the key can be composed in $struct$, we can exclude some models.
- If the decryption is successful in $concr$, then it is also successful in $struct$ and we can define recipes for the new terms.
  $\longrightarrow$ We add shorthands!

Analysis of a structural frame

Analysis steps: we check if we can decrypt terms in the frame.

- If the decryption fails in $concr$, i.e., the key cannot be composed, no new terms can be added. But if the key can be composed in $struct$, we can exclude some models.
- If the decryption is successful in $concr$, then it is also successful in $struct$ and we can define recipes for the new terms.
  $\longrightarrow$ We add shorthands!
- Repeat until no more new terms can be added.

Analysis of a structural frame

Example:

$$\theta = [x \mapsto \mathsf{k}_1, y \mapsto \mathsf{a}, z \mapsto \mathsf{k}_1]$$
$$struct = \{\!\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(x, y), \mathsf{l}_2 \mapsto z\ |\!\!\}$$

Analysis of a structural frame

Example:

$$\theta = [x \mapsto \mathsf{k}_1, y \mapsto \mathsf{a}, z \mapsto \mathsf{k}_1]$$
$$struct = \{\!|\, \mathsf{l}_1 \mapsto \mathsf{scrypt}(x, y), \mathsf{l}_2 \mapsto z \,|\!\}$$

The analysis adds the shorthand $\mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1) \mapsto y$ because the decryption is successful in $concr$. $\longrightarrow x = z$.

Analysis of a structural frame

Example:

$$\theta = [x \mapsto \mathsf{k}_1, y \mapsto \mathsf{a}, z \mapsto \mathsf{k}_1]$$
$$struct = \{\!| \, \mathsf{l}_1 \mapsto \mathsf{scrypt}(x, y), \mathsf{l}_2 \mapsto z \, |\!\}$$

The analysis adds the shorthand $\mathsf{dscrypt}(\mathsf{l}_2, \mathsf{l}_1) \mapsto y$ because the decryption is successful in $concr$. $\longrightarrow x = z$.

For the model $\theta' = [x \mapsto \mathsf{k}_1, y \mapsto \mathsf{a}, z \mapsto \mathsf{k}_2]$, the decryption fails. $\longrightarrow x \neq z$.

Table of Contents

Relations between variables

1. Try to compose terms in $concr$ in different ways:
   - Pairs of recipes must generate the same term in $struct$ because $concr \sim struct$. $\longrightarrow$ Find equalities ($x = t \wedge y = t' \ldots$).

Relations between variables

1. Try to compose terms in $concr$ in different ways:
   - Pairs of recipes must generate the same term in $struct$ because $concr \sim struct$. $\longrightarrow$ Find equalities ($x = t \wedge y = t' \ldots$).
2. Try to compose terms in $struct$ in different ways:
   - Check pairs (label, recipe).
   - If they generate the same term in $concr$, nothing to deduce (it comes from $concr \sim struct$ and has been found previously).
   - If they generate different terms in $concr$, some models can be excluded. $\longrightarrow$ Find inequalities ($x \neq t \vee y \neq 0 \ldots$)

Relations between variables

1. Try to compose terms in $concr$ in different ways:
   - Pairs of recipes must generate the same term in $struct$ because $concr \sim struct$. $\longrightarrow$ Find equalities ($x = t \wedge y = t' \ldots$).
2. Try to compose terms in $struct$ in different ways:
   - Check pairs (label, recipe).
   - If they generate the same term in $concr$, nothing to deduce (it comes from $concr \sim struct$ and has been found previously).
   - If they generate different terms in $concr$, some models can be excluded. $\longrightarrow$ Find inequalities ($x \neq t \vee y \neq 0 \ldots$)
3. $\phi \equiv$ conjunction of equalities and inequalities (= relations between variables)

Relations between variables

Recall the illustration example:

$$\theta = [x \mapsto 0, y \mapsto 1, z \mapsto 0]$$
$$struct = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, x), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, y), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, z)\ |\!\}$$
$$concr = \{\!|\ \mathsf{l}_1 \mapsto \mathsf{scrypt}(k, 0), \mathsf{l}_2 \mapsto \mathsf{scrypt}(k, 1), \mathsf{l}_3 \mapsto \mathsf{scrypt}(k, 0)\ |\!\}$$

$$\alpha \equiv x, y, z \in \{0, 1\} \wedge x + y + z = 1 \qquad \beta \equiv MsgAna(\alpha, struct, \theta)$$

With our decision procedure:

1. We analyze $struct$.
2. We generate $\phi \equiv x = z \wedge x \neq y$.
3. We find that $\alpha \not\models \phi$: violation of privacy!

Relations between variables

- ϕ is enough to decide privacy: we only need to check whether $\alpha \models \phi$.
- If $\alpha \models \phi$: the protocol respects privacy and we have a proof.
- If $\alpha \not\models \phi$: the protocol is not secure and we have a witness.

Table of Contents

Current research

Focus on automation:

- ☒ Design a decision procedure to verify privacy goals for decidable fragments.[3]
- ☒ Develop a proof-of-concept tool in Haskell.
- ☐ Support more general theories (e.g., commutativity, exponentiation).
- ☐ Model and verify real-world protocols.
- ☐ Improve tool support.

[3] Fernet L., Mödersheim S.: Deciding a Fragment of (α, β)-Privacy. STM 2021, LNCS.