Security Models
Lecture 4
Active Intruder

Pascal Lafourcade

2020-2021

Outline of Today

Outline of Today

1. Unification Notions
   - Terms and Messages
   - Unification
2. Active Intruder: Security Problem

Outline of Today

Outline of Today

Outline of Today

Outline

Arity

Definition

- $\mathcal{F}$ is a finite set
- *Arity* is a mapping from $\mathcal{F}$ into $\mathbb{N}$
- $(\mathcal{F}, Arity)$ is a **ranked alphabet** or **signature** denoted Σ

- The **arity** of a symbol $f \in \mathcal{F}$ is $Arity(f)$
- The set of symbols of arity p is denoted by $\mathcal{F}_p$.
- Elements of arity 0, 1, ... p are respectively called constants, unary, ... p-ary symbols.

Example

Example

Let $\mathcal{F} = \{\mathsf{enc}, \mathsf{pair}, \mathsf{k}_1, \mathsf{k}_2, 0, 1\}$

$Arity(\mathsf{enc}) = Arity(\mathsf{pair}) = 2$
$Arity(\mathsf{k}_1) = Arity(\mathsf{k}_2) = Arity(0) = Arity(1) = 0$

We also denote $\mathcal{F} = \{\mathsf{enc}/2, \mathsf{pair}/2, \mathsf{k}_1/0, \mathsf{k}_2/0, 0/0, 1/0\}$

Terms

Let $\mathcal{X}$ be a set of symbols called **variables**.

Definition

The set $\mathcal{T}(\mathcal{F}, \mathcal{X})$ of **terms** over the ranked alphabet $\mathcal{F}$ and the set of variables $\mathcal{X}$ is the smallest set defined by:

- $\mathcal{F}_0 \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$
- $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$
- if $p \geq 1$, $f \in \mathcal{F}_p$ and $t_1, \ldots, t_p \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, then $f(t_1, \ldots, t_p) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$.

- If $\mathcal{X} = \emptyset$ then $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is also written $\mathcal{T}(\mathcal{F})$. Terms in $\mathcal{T}(\mathcal{F})$ are called **ground terms**.
- A term in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is **linear** if each variable occurs at most once in t.

Example

Example

Let $\mathcal{F} = \{\mathsf{enc}/2, \mathsf{pair}/2, \mathsf{k}_1/0, \mathsf{k}_2/0, 0/0, 1/0\}$ and $\mathcal{X} = \{x, y, z\}$

$\mathsf{pair}(x, 1)$, $\mathsf{enc}(\mathsf{pair}(y, z), \mathsf{k}_1)$ and $\mathsf{enc}(0, \mathsf{k}_1)$ are terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$

$\mathsf{pair}(0, 1)$, $\mathsf{enc}(0, \mathsf{k}_1)$ are terms in $\mathcal{T}(\mathcal{F})$, i.e., ground terms

We also denote $\{\_\}_\_$ for $\mathsf{enc}(\_, \_)$ and $\langle \_, \_ \rangle$ for $\mathsf{pair}(\_, \_)$.

Substitution

Definition

- A **substitution** (respectively a **ground substitution**) σ is a mapping from $\mathcal{X}$ into $\mathcal{T}(\mathcal{F}, \mathcal{X})$ (respectively into $\mathcal{T}(\mathcal{F})$) where there are only finitely many variables not mapped to themselves.
- Substitutions can be extended to $\mathcal{T}(\mathcal{F}, \mathcal{X})$ in such a way that $\forall f \in \mathcal{F}_n, \forall t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$:

$$\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n)).$$

The **domain** of a substitution σ is the subset of variables $x \in \mathcal{X}$ such that $\sigma(x) \neq x$.

Example:

Let $\sigma = \{x \leftarrow N_A, y \leftarrow \{\langle N_A, N_B \rangle\}_{k_B}\}$ and $t = \langle x, \langle y, \langle x, x \rangle\rangle\rangle$.

Then,

$$\sigma(t) = \langle N_A, \langle \{\langle N_A, N_B \rangle\}_{k_B}, \langle N_A, N_A \rangle\rangle\rangle$$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:
$s = a \quad t = X$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:
$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$ No unifier

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \quad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$ No unifier

$s = p(a, X) \quad t = p(Y, b)$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:
$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$
$s = a \quad t = p(X)$ No unifier
$s = p(a, X) \quad t = p(Y, b) \qquad \sigma = \{X \leftarrow b; Y \leftarrow a\}$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$ No unifier

$s = p(a, X) \quad t = p(Y, b) \qquad \sigma = \{X \leftarrow b; Y \leftarrow a\}$

$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$ No unifier

$s = p(a, X) \quad t = p(Y, b) \qquad \sigma = \{X \leftarrow b; Y \leftarrow a\}$

$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$

$\qquad \sigma = \{X \leftarrow a; Y \leftarrow g(Z)\}$

Unification

Definition

Two t and s are unifiable if there exists a substitution σ such that $\sigma(s) = \sigma(t)$

Examples:

$s = a \quad t = X \qquad \sigma = \{X \leftarrow a\}$

$s = a \quad t = p(X)$ No unifier

$s = p(a, X) \quad t = p(Y, b) \qquad \sigma = \{X \leftarrow b; Y \leftarrow a\}$

$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$

$\sigma = \{X \leftarrow a; Y \leftarrow g(Z)\}$ or $\sigma = \{X \leftarrow a; Y \leftarrow g(b); Z \leftarrow b\}$

Most General Unifier

Definition

The most general unification between two terms s and t, denoted by $mgu(s,t)$ if: $\forall\sigma$ such that $s\sigma = t\sigma, \exists\theta$ such that $\sigma = mgu(s,t)\theta$

Most General Unifier

Definition

The most general unification between two terms s and t, denoted by $mgu(s, t)$ if: $\forall \sigma$ such that $s\sigma = t\sigma, \exists \theta$ such that $\sigma = mgu(s, t)\theta$

Example:

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

Most General Unifier

Definition

The most general unification between two terms s and t, denoted by $mgu(s,t)$ if: $\forall\sigma$ such that $s\sigma = t\sigma, \exists\theta$ such that $\sigma = mgu(s,t)\theta$

Example:

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

$\sigma_1 = \{X \leftarrow a; Y \leftarrow g(Z)\}$ $\sigma_2 = \{X \leftarrow a; Y \leftarrow g(b); Z \leftarrow b\}$

Most General Unifier

Definition

The most general unification between two terms s and t, denoted by $mgu(s, t)$ if: $\forall\sigma$ such that $s\sigma = t\sigma, \exists\theta$ such that $\sigma = mgu(s, t)\theta$

Example:

$$s = p(f(X), g(Z)) \quad t = p(f(a), Y)$$

$\sigma_1 = \{X \leftarrow a; Y \leftarrow g(Z)\}$ $\sigma_2 = \{X \leftarrow a; Y \leftarrow g(b); Z \leftarrow b\}$

$\theta = \{z \mapsto b\}$, $\sigma_2 = \sigma_1\theta$

Goal

Design an algorithm that for a given unification problem $s =^{?} t$

- returns an mgu of s and t if they are unifiable.
- reports failure otherwise.

Naive Algorithm

Write down two terms and set markers at the beginning of the terms. Then:

1. Move the markers simultaneously, one symbol at a time, until both move off the end of the term (success), or until they point to two different symbols;
2. If the two symbols are both non-variables, then fail; otherwise, one is a variable (call it x) and the other one is the first symbol of a subterm (call it t):
   - If x occurs in t, then fail;
   - Otherwise, replace x everywhere by t (including in the solution), write down "$x \leftarrow t$" as a part of the solution, and return to 1.

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(x, g(a), g(z))$

$f(g(y), g(y), g(g(x)))$

$\sigma = \{x \leftarrow g(y)\}$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(g(y), g(a), g(z))$

$f(g(y), g(y), g(g(g(y))))$

$\sigma = \{x \leftarrow g(y)\}$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(g(y), g(a), g(z))$

$f(g(y), g(y), g(g(g(y))))$

$\sigma = \{x \leftarrow g(y)\}$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(g(a), g(a), g(z))$

$f(g(a), g(a), g(g(g(a))))$

$\sigma = \{x \leftarrow g(a), y \leftarrow a\}$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(g(a), g(a), g(z))$

$f(g(a), g(a), g(g(g(a))))$

$\sigma = \{x \leftarrow g(a), y \leftarrow a\}$

Example: $f(x, g(a), g(z)) =^? f(g(y), g(y), g(g(x)))$

$f(g(a), g(a), g(g(g(a))))$

$f(g(a), g(a), g(g(g(a))))$

$\sigma = \{x \leftarrow g(a), y \leftarrow a, z \leftarrow g(g(a))\}$

Questions

1. Correctness:
   - Does the algorithm always terminate?
   - Does it always produce an mgu for two unifiable terms, and fail for non-unifiable terms?
   - Do these answers depend on the order of operations?
2. Complexity:
   - How much space does this take, and how much time?
3. Extension with equational theory, e.g., $ab = ba$.

Syntactic Unification is Unitary

Theorem (Robinson)

Without equational theory there exists an unique mgu for syntactic unification (modulo renaming). Unification is called unitary.

Herbrand, Martelli, Montanari, Plotkin, Robinson, Huet, Knuth, Bendix, Siekman, Baader.

Outline

Active Intruder with bounded number of sessions

- Theoriticaly: decidable
- Interesting practically:
  - Find flaws
  - Usually attacks use few sessions !

Dolev-Yao Deduction System

Deduction System : $T_0 \vdash^? s$

(A) $$\frac{u \in T_0}{T_0 \vdash u}$$

(UL) $$\frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash u}$$

(P) $$\frac{T_0 \vdash u \qquad T_0 \vdash v}{T_0 \vdash \langle u, v \rangle}$$

(UR) $$\frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash v}$$

(C) $$\frac{T_0 \vdash u \qquad T_0 \vdash v}{T_0 \vdash \{u\}_v}$$

(D) $$\frac{T_0 \vdash \{u\}_v \qquad T_0 \vdash v}{T_0 \vdash u}$$

Model: actions, roles and protocol

Definition (Action)

An action is a couple $(recv(u), send(v))$ such that
$u \in \mathcal{T}(\mathcal{F},\mathcal{X}) \cup \{init\}$, $v \in \mathcal{T}(\mathcal{F},\mathcal{X}) \cup \{stop\}$. Denoted $(u \to v)$.

Model: actions, roles and protocol

Definition (Action)

An action is a couple $(recv(u), send(v))$ such that
$u \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \cup \{init\}$, $v \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \cup \{stop\}$. Denoted $(u \to v)$.

Example

First and last actions of Needham Schroeder

- $(init, X_b \to \{N_a, A\}_{pk(X_b)})$
- $(\{N_b\}_{pk(B)} \to stop)$

Model: actions, roles and protocol

Definition (Role)

A role is a finite sequence of actions:

$$(u_1 \to v_1), \ldots, (u_n \to v_n)$$

such that $vars(v_i) \subseteq \bigcup_{1 \leq j \leq i} vars(u_j)$.

Model: actions, roles and protocol

Definition (Role)

A role is a finite sequence of actions:

$$(u_1 \to v_1), \ldots, (u_n \to v_n)$$

such that $vars(v_i) \subseteq \bigcup_{1 \leq j \leq i} vars(u_j)$.

Definition (Protocol)

A protocol P is a finite set of roles: $P = \{R_1, \ldots, R_k\}$

1st Example:

Example (Needham-schroeder)

$$\begin{array}{llcll} 1. & A & \to & B & : \{N_a, A\}_{pk(B)} \\ 2. & B & \to & A & : \{N_a, N_b\}_{pk(A)} \\ 3. & A & \to & B & : \{N_b\}_{pk(B)} \end{array}$$

Write down each agent's role description, this A talks with anybody.

$$\begin{array}{rl} R_A = & (init, X_b \to \{N_a, A\}_{pk(X_b)}), \\ & (\{N_a, x_{N_b}\}_{pk(A)} \to \{x_{Nb}\}_{pk(X_b)}), \end{array}$$

$$\begin{array}{rl} R_B = & (\{x_{N_a}, x_A\}_{pk(B)} \to \{x_{N_a}, N_b\}_{pk(x_A)}) \\ & (\{N_b\}_{pk(B)} \to stop) \end{array}$$

Scyther Notation

```
A:   const Na: Nonce;
     var Nb: Nonce;

     send(A,B, {Na,A}pk(B));
     recv(B,A, {Na,Nb}pk(A));
     send(A,B, {Nb}pk(B));

B:   const Nb: Nonce;
     var Na: Nonce;

     recv(A,B,{Na,A}pk(B));
     send(B,A,{Na,Nb}pk(A));
     recv(A,B,{Nb}pk(B));
```

Exercise

Denning-Sacco Protocol

$$
\begin{array}{lll}
1. & A \to S: & \langle A, B\rangle \\
2. & S \to A: & \{\langle\langle B, N_{AB}\rangle, \langle N_s, \{\langle N_{AB}, \langle A, N_s\rangle\rangle\}_{K_{BS}}\rangle\rangle\}_{K_{AS}} \\
3. & A \to B: & \{\langle N_{AB}, \langle A, N_s\rangle\rangle\}_{K_{BS}} \\
4. & B \to A: & \{S_{AB}\}_{N_{AB}}
\end{array}
$$

$P_{DS} = \{R_A, R_B, R_S\}$ models one session of A, B and S.

Exercise

Denning-Sacco Protocol

$$
\begin{array}{lll}
1. & A \to S: & \langle A, B\rangle \\
2. & S \to A: & \{\langle\langle B, N_{AB}\rangle, \langle N_s, \{\langle N_{AB}, \langle A, N_s\rangle\rangle\}_{K_{BS}}\rangle\rangle\}_{K_{AS}} \\
3. & A \to B: & \{\langle N_{AB}, \langle A, N_s\rangle\rangle\}_{K_{BS}} \\
4. & B \to A: & \{S_{AB}\}_{N_{AB}}
\end{array}
$$

$P_{DS} = \{R_A, R_B, R_S\}$ models one session of A, B and S.

$$
\begin{aligned}
R_A = \quad & (init, X_B \to \langle A, X_B\rangle), \\
& (\{\langle\langle X_B, x_{N_{AB}}\rangle, \langle x_{N_S}, z_A\rangle\rangle\}_{K_{AS}} \to z_A), \\
& (\{w_A\}_{x_{N_{AB}}} \to stop)
\end{aligned}
$$

$$
R_B = \quad (\{y_{N_{AB}}, \langle X_A, y_{N_S}\rangle\rangle\}_{K_{BS}} \to \{S_{AB}\}_{y_{N_{AB}}})
$$

$$
R_S = \quad (\langle X_A, X_B\rangle \to \{\langle X_B, N_{AB}, \langle N_S, \{\langle N_{AB}, \langle X_A, N_S\rangle\rangle\}_{K_{BS}}\rangle\rangle\}_{K_{AS}})
$$

Semantics

Definition (States)

- T is a set of ground terms (intruder knowledge)
- P a protocol

A state is a couple (T, P)

Definition (Transtion)

Is a relation between states $(T, P) \rightarrow^{\sigma} (T', P')$

- $P = \bigcup_i^k R_i$, take an i : $R_i = (u_i \rightarrow v_i)$
- Possible σ : $T \vdash u_i\sigma \quad (dom(\sigma) = vars(u_i))$
- Update intruder knowledge : $T' = T \cup \{v_i\sigma\}$
- Update Protocol $\forall j \neq i, R_j \in P', P' = (P \setminus \{R_i\}) \cup R_j\sigma$

Example

Example

Simple Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y\rangle \to \langle \{y\}_k, x\rangle), (z \to \langle x, \langle y, z\rangle\rangle)$.

- $(T, P) \to^{\sigma} (T \cup \{\langle \{b\}_k, a\rangle\}, \{(z \to \langle a, \langle b, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow b\}$

Example

Example

Simple Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y\rangle \rightarrow \langle \{y\}_k, x\rangle), (z \rightarrow \langle x, \langle y, z\rangle\rangle)$.

- $(T, P) \rightarrow^{\sigma} (T \cup \{\langle \{b\}_k, a\rangle\}, \{(z \rightarrow \langle a, \langle b, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow b\}$
- $(T, P) \rightarrow^{\sigma} (T \cup \{\langle \{\{a\}_{k_I}\}_k, a\rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_{k_I}, z\rangle\rangle)\}$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_{k_I}\}$

Example

Example

Simple Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y\rangle \to \langle\{y\}_k, x\rangle), (z \to \langle x, \langle y, z\rangle\rangle)$.

- $(T, P) \to^{\sigma} (T \cup \{\langle\{b\}_k, a\rangle\}, \{(z \to \langle a, \langle b, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow b\}$
- $(T, P) \to^{\sigma} (T \cup \{\langle\{\{a\}_{k_I}\}_k, a\rangle\}, \{(z \to \langle a, \langle\{a\}_{k_I}, z\rangle\rangle)\}$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_{k_I}\}$
- $(T, P) \not\to^{\sigma} (T \cup \{\langle\{\{a\}_k\}_k, a\rangle\}, \{(z \to \langle a, \langle\{a\}_k, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_k\}$

Example

Example

Simple Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y\rangle \rightarrow \langle \{y\}_k, x\rangle), (z \rightarrow \langle x, \langle y, z\rangle\rangle)$.

- $(T, P) \rightarrow^\sigma (T \cup \{\langle \{b\}_k, a\rangle\}, \{(z \rightarrow \langle a, \langle b, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow b\}$
- $(T, P) \rightarrow^\sigma (T \cup \{\langle \{\{a\}_{k_I}\}_k, a\rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_{k_I}, z\rangle\rangle)\}$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_{k_I}\}$
- $(T, P) \not\rightarrow^\sigma (T \cup \{\langle \{\{a\}_k\}_k, a\rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_k, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_k\}$

Each branch has a finite depth (protocol are finite),

Example

Example

Simple Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y\rangle \to \langle \{y\}_k, x\rangle), (z \to \langle x, \langle y, z\rangle\rangle)$.

- $(T, P) \to^\sigma (T \cup \{\langle \{b\}_k, a\rangle\}, \{(z \to \langle a, \langle b, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow b\}$
- $(T, P) \to^\sigma (T \cup \{\langle \{\{a\}_{k_I}\}_k, a\rangle\}, \{(z \to \langle a, \langle \{a\}_{k_I}, z\rangle\rangle)\}$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_{k_I}\}$
- $(T, P) \not\to^\sigma (T \cup \{\langle \{\{a\}_k\}_k, a\rangle\}, \{(z \to \langle a, \langle \{a\}_k, z\rangle\rangle)\})$
  $\sigma = \{x \leftarrow a, y \leftarrow \{a\}_k\}$

Each branch has a finite depth (protocol are finite),
but possibly a infinite branching (infinite number of terms).

Preservation of the secrecy

Definition (Secrecy)

Let T_1 be a ground set of terms (Initial knowledge of the intruder).
A protocol P does not preserve the secrecy of a ground term s for T_1 if there exists a state (T', P'), such that

- $T' \vdash s$
- $(T_1, P) \rightarrow^* (T', P')$

where $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$.

If there does not exist a such state (T', P') we say that P preserves the secrecy of s for the initial intruder knowledge T_1.

Interleaving

Definition (Partial Order $<_p$)

A protocol P define a partial order $<_P$ on actions of P, s.t

$$(u_i \to v_i) <_P (u_j \to v_j)$$

if $R \in P$, $R = (u_1 \to v_1) \ldots (u_i \to v_i) \ldots (u_j \to v_j) \ldots (u_n \to v_n)$ $(1 \leq i \leq j \leq n)$.

Interleaving

Definition (Partial Order $<_p$)

A protocol P define a partial order $<_P$ on actions of P, s.t

$$(u_i \to v_i) <_P (u_j \to v_j)$$

if $R \in P$, $R = (u_1 \to v_1) \dots (u_i \to v_i) \dots (u_j \to v_j) \dots (u_n \to v_n)$ $(1 \leq i \leq j \leq n)$.

Definition (Execution Order $<_E$)

An execution order $<_E$ of P is a total order on the subset A of actions of P, compatible with $<_P$ and stable by predecessor, i.e.

$$\text{if } b \in A \text{ et } a <_P b \text{ then } a \in A \text{ and } a <_E b$$

It corresponds to an interleaving of roles.

Secrecy

Definition (Secrecy over $<_E$)

Let an execution order $<_E$ of P. We assume that

$$(u_1 \to v_1) <_E \ldots <_E (u_n \to v_n)$$

$<_E$ does not preserve the secrecy of s, given T_1 if there exists $\sigma_1, \ldots, \sigma_n$ such that

$$(T_1, P) \to (T_1 \cup \{v_1\sigma_1\}, P_1) \to \ldots \to (T_1 \cup \{v_1\sigma_1, \ldots, v_n\sigma_n\}, P_n)$$

and $T_1 \cup \{v_1\sigma_1, \ldots, v_n\sigma_n\} \vdash s$.

Outline

Constraints System

Symbolic representation of execution tree by constraints system.

Definition (Constraints System)

A constraint is an expression $T \Vdash u$
where T is a set of terms and u a term.

A constraints system C is a finite set of constraints $\cup_{1 \le i \le n} T_i \Vdash u_i$ such that

- $T_i \subseteq T_{i+1} \qquad (1 \le i \le n)$
- if $T_i \Vdash u_i \in C$ and $x \in vars(T_i)$ then
  $T_j = min\{T' \mid T' \Vdash v \in C, x \in vars(v)\}$ exists and $j < i$

A substitution σ is a solution of C if $T\sigma \vdash u\sigma$ for all $T \Vdash u \in C$.

We denote by $\perp$ a constraints system unsatisfiable.

From Protocols to Constraints system

Let P a protocol, $<_E$ an execution order of P and s a secret term.

$$(u_1 \rightarrow v_1) <_E (u_2 \rightarrow v_2) <_E \ldots <_E (u_n \rightarrow v_n)$$

We associate C:

$$\begin{array}{rcl}
T_1 & \Vdash & u_1 \\
T_2 = T_1 \cup \{v_1\} & \Vdash & u_2 \\
 & \vdots & \\
T_n = T_{n-1} \cup \{v_{n-1}\} & \Vdash & u_n \\
T_{n+1} = T_n \cup \{v_n\} & \Vdash & s
\end{array}$$

We show that C has a solution iff $<_E$ does not preserve the secret of the term s.

Exercises

Exercise 1

$$
\begin{array}{rl}
A \rightarrow B: & \langle A, N_A \rangle \\
B \rightarrow A: & \{\langle N_A, N_B \rangle\}_{K_{ab}} \\
A \rightarrow B: & N_B \\
B \rightarrow A: & \{\langle K, N_B \rangle\}_{K_{ab}} \\
A \rightarrow B: & \{s\}_K
\end{array}
$$

Intruder knows only identities of A and B.

- Give role specification of this protocol of an instance of execution between A and B.
- Give a constraint system associated to this protocol between A and B.

Solution

$$\begin{array}{ll} A \to B: & \langle A, N_A \rangle \\ B \to A: & \{\langle N_A, N_B \rangle\}_{K_{ab}} \\ A \to B: & N_B \\ B \to A: & \{\langle K, N_B \rangle\}_{K_{ab}} \\ A \to B: & \{s\}_K \end{array}$$

$T_1 =$
$\{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}_K, init, stop\}$

Roles

$$\begin{array}{rl} R_A = & (init \to \langle A, N_A \rangle), \\ & (\{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \to X_{N_B}), \\ & (\{\langle X_K, X_{N_B} \rangle\}_{K_{(A,X_B)}} \to \{s\}_{X_K}) \end{array}$$

$$\begin{array}{rl} R_B = & (\langle X_A, X_{N_A} \rangle \to \{\langle X_{N_A}, N_B \rangle\}_{K_{(X_A,B)}}) \\ & (N_B \to \{\langle K, N_B \rangle\}_{K_{(X_A,B)}}), \\ & (\{X_s\}_K \to stop) \end{array}$$

Solution

$$\begin{array}{ll} A \to B: & \langle A, N_A \rangle \\ B \to A: & \{\langle N_A, N_B \rangle\}_{K_{ab}} \\ A \to B: & N_B \\ B \to A: & \{\langle K, N_B \rangle\}_{K_{ab}} \\ A \to B: & \{s\}_K \end{array}$$

$T_1 =$
$\{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}_K, init, stop\}$

Constraint System

$$\begin{array}{lll} T_1 & \Vdash & init \\ T_2 = T_1 \cup \{\langle A, N_A \rangle\} & \Vdash & \langle X_A, X_{N_A} \rangle \\ T_3 = T_2 \cup \{\{\langle X_{N_A}, N_B \rangle\}_{K_{(X_A,B)}}\} & \Vdash & \{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\ T_4 = T_3 \cup \{X_{N_B}\} & \Vdash & N_B \\ T_5 = T_4 \cup \{\{\langle K, N_B \rangle\}_{K_{(X_A,B)}}\} & \Vdash & \{\langle X_K, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\ T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash & \{X_s\}_K \\ T_7 = T_6 \cup \{stop\} & \Vdash & s \end{array}$$

Resolution of Constraints systems

Definition (Rules of simplification: $C \rightsquigarrow_\sigma C'$)

R_1	$C \cup \{T \Vdash u\}$	$\rightsquigarrow$	C	if $T \cup \{x \mid T' \Vdash x \in C, T' \subset T\} \vdash u$
R_2	$C \cup \{T \Vdash u\}$	$\rightsquigarrow_\sigma$	$C\sigma \cup \{T\sigma \Vdash u\sigma\}$	$\sigma = mgu(t, u), t \in st(T)$, t, u no variables
R_3	$C \cup \{T \Vdash u\}$	$\rightsquigarrow_\sigma$	$C\sigma \cup \{T\sigma \Vdash u\sigma\}$	$\sigma = mgu(t_1, t_2), t_1, t_2 \in st(T)$, t_1, t_2 no variables
R_4	$C \cup \{T \Vdash \{u\}_v\}$	$\rightsquigarrow$	$C \cup \{T \Vdash u, T \Vdash v\}$	
R_5	$C \cup \{T \Vdash \langle u, v\rangle\}$	$\rightsquigarrow$	$C \cup \{T \Vdash u, T \Vdash v\}$	
R_6	$C \cup \{T \Vdash u\}$	$\rightsquigarrow$	$\perp$	if $T = \emptyset$ or $var(T) = var(u) = \emptyset$ and $T \not\vdash u$

Properties of simplification rules

Lemma (Preservation)

Simplification rules transform a constraints system into a constraints system.

Properties of simplification rules

Lemma (Preservation)

Simplification rules transform a constraints system into a constraints system.

Lemma (Correctness)

If $C \leadsto_\sigma C'$ *then if* θ *is a solution of* C', $\sigma\theta$ *is also a solution of* C.

Properties of simplification rules

Lemma (Preservation)

Simplification rules transform a constraints system into a constraints system.

Lemma (Correctness)

If $C \rightsquigarrow_\sigma C'$ *then if* θ *is a solution of* C', $\sigma\theta$ *is also a solution of* C.

Lemma (Termination)

Simplification rules always terminate: There does not exist infinite chain $C \rightsquigarrow_{\sigma_1} C_1 \rightsquigarrow_{\sigma_2} C_2 \rightsquigarrow_{\sigma_3} \ldots$.

Properties

Definition (Solved Form)

A constraints system C is in solved form if $C = \perp$ or if each constraint is of the following form $T \Vdash x$ where x is a variable $T \neq \emptyset$.

Lemma

All constraints systems in solved form different of $\perp$ has at least one solution.

Properties

Definition (Solved Form)

A constraints system C is in solved form if $C = \perp$ or if each constraint is of the following form $T \Vdash x$ where x is a variable $T \neq \emptyset$.

Lemma

All constraints systems in solved form different of $\perp$ has at least one solution.

Lemma (Completeness)

If C is a constraint system not in solved form and if σ is a solution of C then there exists θ, τ such that $C \rightsquigarrow_\theta C'$, $\sigma = \theta\tau$ and τ is a solution of C'.

Decidability

Theorem

Preservation of the secrecy for protocol with bounded number of sessions is decidable.

- Guess an interleaving and build constraints system associated.
- Using previous lemma C has a solution iff there exists C' in solved form such that $C' \neq \perp$ and $C \rightsquigarrow_\tau C'$
- Using termination lemma to conclude.

We also can show that the problem is in co-NP.

Exercises

Exercise 1

$$\begin{array}{rl} A \rightarrow B: & \langle A, N_A \rangle \\ B \rightarrow A: & \{\langle N_A, N_B \rangle\}_{K_{ab}} \\ A \rightarrow B: & N_B \\ B \rightarrow A: & \{\langle K, N_B \rangle\}_{K_{ab}} \\ A \rightarrow B: & \{s\}_K \end{array}$$

Intruder knows only identities of A and B.

- Use simplification rules to transform the system in solved form.
- There exists an easy attack, can you find it ?

Solution

$T_1 = \{A, B, \langle A, N_A\rangle, \{\langle N_A, N_B\rangle\}_{K_{ab}}, N_B, \{\langle K, N_B\rangle\}_{K_{ab}}, \{s\}_K, init, stop\}$

$$
\begin{array}{lllll}
C_1 & T_1 & & \Vdash & init \\
C_2 & T_2 = T_1 \cup \{\langle A, N_A\rangle\} & & \Vdash & \langle X_A, X_{N_A}\rangle \\
C_3 & T_3 = T_2 \cup \{\{\langle X_{N_A}, N_B\rangle\}_{K_{(X_A,B)}}\} & & \Vdash & \{\langle N_A, X_{N_B}\rangle\}_{K_{(A,X_B)}} \\
C_4 & T_4 = T_3 \cup \{X_{N_B}\} & & \Vdash & N_B \\
C_5 & T_5 = T_4 \cup \{\{\langle K, N_B\rangle\}_{K_{(X_A,B)}}\} & & \Vdash & \{\langle X_K, X_{N_B}\rangle\}_{K_{(A,X_B)}} \\
C_6 & T_6 = T_5 \cup \{\{s\}_{X_K}\} & & \Vdash & \{X_s\}_K \\
C_7 & T_7 = T_6 \cup \{stop\} & & \Vdash & s
\end{array}
$$

Road book

Interleaving: $(u_1^A, v_1^A)(u_1^B, v_1^B)(u_2^A, v_2^A)(u_2^B, v_2^B)(u_3^A, v_3^A)(u_3^B, v_3^B)$

$$
\begin{array}{lllll}
R_2 & C \cup \{T \Vdash u\} & \rightsquigarrow_\sigma & C\sigma \cup \{T\sigma \Vdash u\sigma\} & \sigma = mgu(t,u), t \in st(T), \\
 & & & & t, u \text{ no variables}
\end{array}
$$

- Apply nothing on C_1, already in resolved form.
- Apply R_2 on C_2 give $\sigma_1 = \{X_{N_A} \leftarrow N_A, X_A \leftarrow A\}$ and R_1

Solution

$$T_1 = \{A, B, \langle A, N_A\rangle, \{\langle N_A, N_B\rangle\}_{K_{ab}}, N_B, \{\langle K, N_B\rangle\}_{K_{ab}}, \{s\}_K, init, stop\}$$

$$\begin{array}{lllll}
C_3\sigma_1 & T_3 = T_2 \cup \{\{\langle N_A, N_B\rangle\}_{K_{(A,B)}}\} & \Vdash & \{\langle N_A, X_{N_B}\rangle\}_{K_{(A,X_B)}} \\
C_4\sigma_1 & T_4 = T_3 \cup \{X_{N_B}\} & \Vdash & N_B \\
C_5\sigma_1 & T_5 = T_4 \cup \{\{\langle K, N_B\rangle\}_{K_{(A,B)}}\} & \Vdash & \{\langle X_K, X_{N_B}\rangle\}_{K_{(A,X_B)}} \\
C_6\sigma_1 & T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash & \{X_s\}_K \\
C_7\sigma_1 & T_7 = T_6 \cup \{stop\} & \Vdash & s
\end{array}$$

Road book $\sigma_1 = \{X_{N_A} \leftarrow N_A, X_A \leftarrow A\}$

- Apply R_2 on C_3 gives $\sigma_2 = \{X_{N_B} \leftarrow N_B, X_B \leftarrow B\}$ (or N_A) and R_1

Solution

$$T_1 = \{A, B, \langle A, N_A\rangle, \{\langle N_A, N_B\rangle\}_{K_{ab}}, N_B, \{\langle K, N_B\rangle\}_{K_{ab}}, \{s\}_K, init, stop\}$$

$$\begin{array}{lll} C_5\sigma_1\sigma_2 \quad T_5 = T_4 \cup \{\{\langle K, N_B\rangle\}_{K_{(A,B)}}\} & \Vdash & \{\langle X_K, N_B\rangle\}_{K_{(A,B)}} \\ C_6\sigma_1\sigma_2 \quad T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash & \{X_s\}_K \\ C_7\sigma_1\sigma_2 \quad T_7 = T_6 \cup \{stop\} & \Vdash & s \end{array}$$

Road book $\sigma_1 = \{X_{N_A} \leftarrow N_A, X_A \leftarrow A\}$ $\sigma_2 = \{X_{N_B} \leftarrow N_B, X_B \leftarrow B\}$

- Apply R_2 on $C_5\sigma_1\sigma_2$ give $\sigma_3 = \{X_K \leftarrow N_A\}$
- Apply R_2, on $\sigma_1\sigma_2\sigma_3 C_6$ give $\sigma_4 = \{X_S \leftarrow s\}$

Solution

$$
\begin{array}{lll}
1 & A \to B: & \langle A, N_A \rangle \\
2 & B \to A: & \{\langle N_A, N_B \rangle\}_{K_{ab}} \\
3 & A \to B: & N_B \\
4 & B \to A: & \{\langle K, N_B \rangle\}_{K_{ab}} \\
5 & A \to B: & \{s\}_K
\end{array}
$$

The resolution of constraint system gives the following attack: Send 2nd message $\{\langle N_A, N_B \rangle\}_{K_{ab}}$ instead of the 4th message $\{\langle K, N_B \rangle\}_{K_{ab}}$. Hence A will replay $\{s\}_{N_A}$ because intruder knows N_A

Exercises

Exercise 2

$$\begin{array}{rl} A \rightarrow B : & \{\langle A, K \rangle\}_{K_{ab}} \\ B \rightarrow A : & \{s\}_{K_{ab}} \end{array}$$

Intruder knows only identities of A and B. Show that the secret data s is preserved by one single session between A and B.

Solution

$$\begin{array}{rl} A \rightarrow B: & \{\langle A, K\rangle\}_{K_{ab}} \\ B \rightarrow A: & \{s\}_{K_{ab}} \end{array}$$

$T_1 = \{A, B, \{\langle A, K\rangle\}_{K_{ab}}, \{s\}_{K_{ab}}\}$

Constraint System

$$\begin{array}{lll} C_1 & T_1 & \Vdash \; init \\ C_2 & T_2 = T_1 \cup \{\{\langle A, X_K\rangle\}_{K_{ab}}\} & \Vdash \; \{\langle A, X_K\rangle\}_{K_{ab}} \\ C_3 & T_3 = T_2 \cup \{\{s\}_{X_{K_{ab}}}\} & \Vdash \; \{s\}_{X_{K_{ab}}} \\ C_4 & T_4 = T_3 \cup \{stop\} & \Vdash \; s \end{array}$$

Solution

$$\begin{array}{lll} C_1 & T_1 & \Vdash \quad init \\ C_2 & T_2 = T_1 \cup \{\{\langle A, X_K \rangle\}_{K_{ab}}\} & \Vdash \quad \{\langle A, X_K \rangle\}_{K_{ab}} \\ C_3 & T_3 = T_2 \cup \{\{s\}_{X_{K_{ab}}}\} & \Vdash \quad \{s\}_{X_{K_{ab}}} \\ C_4 & T_4 = T_3 \cup \{stop\} & \Vdash \quad s \end{array}$$

$T_1 = \{A, B, \{\langle A, K \rangle\}_{K_{ab}}, \{s\}_{K_{ab}}\}$

Road book

- Apply nothing or R_4 or R_5 and R_2 on C_1 give $\sigma_0 = \{X_K \leftarrow K, X_{K_{ab}} \leftarrow K_{ab}\}$

Each time you meet a solved form of the form $\perp$ with R_6.

Outline

NP-hardness

Theorem

Decide if a protocol P does not preserve the secrecy of a ground term s from an initial knowledge T_1 is NP-difficult.

Recall 3-SAT Problem

Definition

Input: set of propositional variables $\{x_1, \ldots, x_n\}$ and a conjunction of clauses with 3 literals.

$$f(\vec{x}) = \bigwedge_{1 \leq i \leq l} (x_{i,1}^{\epsilon_{i,1}} \vee x_{i,2}^{\epsilon_{i,2}} \vee x_{i,3}^{\epsilon_{i,3}})$$

where $\epsilon_{i,j} \in \{+, -\}$ and $x^+ = x, x^- = \neg x$.
Question : Does exist a valuation V of $\{x_1, \ldots, x_n\}$, such that $V(f(\vec{x})) = \top$.

Theorem

3-SAT problem is NP-complete.

NP-difficulty

We build a protocol such that an intruder can deduce s iff $f(\vec{x})$ is satisfaisable.

$$g(x_{i,j}^{\epsilon_{i,j}}) = \begin{cases} x_{i,j} & \text{if } \epsilon_{i,j} = + \\ \{x_{i,j}\}_K & \text{if } \epsilon_{i,j} = - \end{cases}$$
$$\forall 1 \le i \le l : \; f_i(\vec{x}) = \langle g(x_{i,1}^{\epsilon_{i,1}}), \langle g(x_{i,2}^{\epsilon_{i,2}}), g(x_{i,3}^{\epsilon_{i,3}})\rangle\rangle$$

We suppose Initial intruder knowledge is $\{\bot, \top\}$.

$$\begin{array}{rl}
A: & \langle x_1, \langle \ldots, x_n\rangle\rangle \to \{\langle f_1(\vec{x}), \langle f_2(\vec{x}), \langle \ldots, \langle f_n(\vec{x}), end\rangle \ldots\rangle\rangle\}_p \\
\forall 1 \le i \le l: & \\
B_i: & \{\langle\langle \top, \langle x, y\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
\overline{B}_i: & \{\langle\langle \{\bot\}_K, \langle x, y\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
C_i: & \{\langle\langle x, \langle \top, y\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
\overline{C}_i: & \{\langle\langle x, \langle \{\bot\}_K, y\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
D_i: & \{\langle\langle x, \langle y, \top\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
\overline{D}_i: & \{\langle\langle x, \langle y, \{\bot\}_K\rangle\rangle, z\rangle\}_p \to \{z\}_p \\
E: & \{end\}_p \to s
\end{array}$$

Outline

Summary

Today

- Active Intruder
- Bounded Number of Sessions
- NP-Hardness
- Tools

Next Time

- Playing with Tools:
    - Scyther
    - Avispa: OFMC, Cl-Atse, SATMC, TA4SP
    - Proverif

Thank you for your attention

Questions ?