Special topics in Logic and Security I

Master Year II, Sem. I, 2022-2023

Ioana Leuştean
FMI, UB

References

- Cremers, C. J. F. (2006). Scyther : semantics and verification of security protocols Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR614943
- Cremers C. and Mauw S. Operational Semantics and Verification of Security Protocols. Springer, 2012.

Operational semantics

The operational semantics of a security protocol P is a labelled transition system

$$(State, RunEvent, \rightarrow, st_0(P))$$

- $State = \mathcal{P}(RunTerm) \times \mathcal{P}(Run)$ where $Run = Inst \times RoleEvent^*$
- $st_0(P) = \langle\!\langle AKN_0(P), \emptyset \rangle\!\rangle$ where $AKN_0(P)$ is the initial adversary knowledge
- $RunEvent = Inst \times (RoleEvent \cup \{create(R) \mid R \in Role\}$
- The *transition system* has four rules, one for each of the events:

  create, *send*, *recv*, *claim*

Operational semantics: transitions

$$(State, RunEvent, \rightarrow, st_0(P))$$

$$[create_P]\ \frac{R \in dom(P) \quad ((\theta, \rho, \emptyset), s) \in runsof(P, R) \quad \theta \notin runsIDs(F)}{\langle\!\langle AKN, F \rangle\!\rangle \xrightarrow{((\theta,\rho,\emptyset),create(R))} \langle\!\langle AKN, F \cup \{((\theta, \rho, \emptyset), s)\} \rangle\!\rangle}$$

Recall that

$runIds(F) = \{\theta \mid ((\theta, \rho, \sigma), s) \in F$ for some $\rho, \sigma, s\}$ and

$F \subseteq Run = Inst \times RoleEvent^*$.

Operational semantics : transitions

$$(State, RunEvent, \rightarrow, st_0(P))$$

$$[send]\ \frac{e = send_l(R_1, R_2, m) \quad (inst, [e] \cdot s) \in F}{\langle\!\langle AKN, F \rangle\!\rangle \xrightarrow{(inst,e)} \langle\!\langle AKN \cup \{inst(m)\}, F \setminus \{(inst, [e] \cdot s)\} \cup \{(inst, s)\} \rangle\!\rangle}$$

$$[claim]\ \frac{e = claim_l(R, c, t) \quad (inst, [e] \cdot s) \in F}{\langle\!\langle AKN, F \rangle\!\rangle \xrightarrow{(inst,e)} \langle\!\langle AKN, F \setminus \{(inst, [e] \cdot s)\} \cup \{(inst, s)\} \rangle\!\rangle}$$

Operational semantics : transitions

$$(State, RunEvent, \rightarrow, st_0(P))$$

$$[recv]\ \frac{e = recv_l(R_1, R_2, pt) \ \ AKN \vdash m \ \ (inst, [e] \cdot s) \in F \ \ Match(inst, pt, m, inst')}{\langle\!\langle AKN, F \rangle\!\rangle \xrightarrow{(inst', e)} \langle\!\langle AKN, F \setminus \{(inst, [e] \cdot s)\} \cup \{(inst', s)\} \rangle\!\rangle}$$

Recall that, for $pt \in RoleTerm$ and $m \in RunTerm$, $Match(inst, pt, m, inst')$ holds if the incoming message m is matched with the pattern pt and the instantiation $inst'$ is $inst$ extended with the new assignments, i.e.
$inst = (\theta, \rho, \sigma)$, $inst' = (\theta, \rho, \sigma')$, $inst'(pt) = m$,
$dom(\sigma') = dom(\sigma) \cup vars(pt)$, $\sigma \subseteq \sigma'$,
$\sigma'(v) \in type(v)$ for any $v \in dom(\sigma')$,
where $vars(pt)$ is the set of variables from Var which appear in pt.

Operational semantics: traces

$$(State, RunEvent, \rightarrow, st_0(P))$$

- Execution:
  $[st_0, \alpha_1, st_1, \alpha_2, \ldots, \alpha_n, st_n]$ where $\alpha_i \in RunEvent$ și $st_i = \langle\langle AKN_i, F_i \rangle\rangle$

- Knowing the initial state we define the execution using traces $[\alpha_1, \alpha_2, \ldots, \alpha_n]$.

  Given a protocol P, we define *traces*(P) as the set of the finite traces of the labelled transition system $(State, RunEvent, \rightarrow, st_0(P))$ associated to P.

Example: trace for the Needham-Schroeder protocol

Let t be the following trace:

$[((1, \rho, \emptyset), create(i))$,

$((1, \rho, \emptyset), send_1(i, r, \{\!| \; ni, i \; |\!\}_{pk(r)})),$

$((2, \rho, \emptyset), create(r)),$

$((2, \rho, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!| \; W, i \; |\!\}_{pk(r)})),$

$((2, \rho, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!| \; W, nr \; |\!\}_{pk(i)})),$

$((1, \rho, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!| \; ni, V \; |\!\}_{pk(i)})),$

$((1, \rho, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!| \; V \; |\!\}_{pk(r)})),$

$((1, \rho, \{V \mapsto nr^{\#2}\}), claim_4(i, synch)),$

$((2, \rho, \emptyset), recv_3(i, r, \{\!| \; nr \; |\!\}_{pk(r)})),$

$((2, \rho, \emptyset), claim_5(r, synch))]$

Example: trace for the Needham-Schroeder protocol

$$[create_P]\ \frac{R \in dom(P) \quad ((\theta,\rho,\emptyset),s) \in runsof(P,R) \quad \theta \notin runsIDs(F)}{\langle\langle AKN, F\rangle\rangle \xrightarrow{((\theta,\rho,\emptyset),create(R))} \langle\langle AKN, F \cup \{((\theta,\rho,\emptyset),s)\}\rangle\rangle}$$

The initial state is
$st_0(NS) = \langle\langle AKN_0(NS), \emptyset\rangle\rangle$ where
$AKN_0(NS) = AdversaryFresh \ \cup \ Agent \cup$
$\{pk(A) \mid A \in Agent\} \cup \{sk(A) \mid A \in Agent_C\}$

For $\rho = \{i \mapsto A, r \mapsto B\}$, the first transition on t is

$$st_0(NS,t) = \langle\langle AKN_0(NS), \emptyset\rangle\rangle \xrightarrow{((1,\rho,\emptyset),create(i))} st_1(NS,t)$$

where

$st_1(NS,t) = \langle\langle AKN_0(NS), \{((1,\rho,\emptyset),s_1)\}\rangle\rangle$

$$\begin{aligned} s_1 = \quad & [send_1(i,r,\{\!|\, ni, i \,|\!\}_{pk(r)}), recv_2(r,i,\{\!|\, ni, V \,|\!\}_{pk(i)}),\\ & send_3(i,r,\{\!|\, V \,|\!\}_{pk(r)}), claim_4(i, synch)] \end{aligned}$$

Example: trace for the Needham-Schroeder protocol

$$[send]\ \frac{e = send_l(R_1, R_2, m) \quad (inst, [e] \cdot s) \in F}{\langle\!\langle AKN, F \rangle\!\rangle \xrightarrow{(inst,e)} \langle\!\langle AKN \cup \{inst(m)\}, F \setminus \{(inst, [e] \cdot s)\} \cup \{(inst, s)\} \rangle\!\rangle}$$

For $\rho = \{i \mapsto A, r \mapsto B\}$, the second transition on t is

$$st_1(NS, t) = \langle\!\langle AKN_0(NS), \{((1, \rho, \emptyset), s_1)\} \rangle\!\rangle \xrightarrow{((1,\rho,\emptyset), send_1(i, r, \{\!|\ ni, i\ |\!\}_{pk(r)}))} st_2(NS, t)$$

where

$st_2(NS, t) = \langle\!\langle AKN_0(NS) \cup \{\{\!|\ ni^{\#1}, A\ |\!\}_{pk(B)}\}, \{((1, \rho, \emptyset), s_2)\} \rangle\!\rangle$ and

$s_2 = [recv_2(r, i, \{\!|\ ni, V\ |\!\}_{pk(i)}), send_3(i, r, \{\!|\ V\ |\!\}_{pk(r)}), claim_4(i, synch)]$

Analysing security properties

Security properties

In our formalism, security properties are defined by (local) *claim* events. This means that an agent has a local view based on the messages he receives and the protocol should offer guarantee that the agent can be sure about certain properties.

In the sequel we shall analyze *secrecy*, which means that certain information is not revealed to an adversary.

Recall that the set of agents is partitioned in honest and corrupted agents $Agent = Agent_H \cup Agent_C$. For an instantiation $(\theta, \rho, \sigma) \in Inst$ we define the predicate $honest(\theta, \rho, \sigma)$ which is true if the roles are instantiated with honest agents, i.e. $honest(\theta, \rho, \sigma)$ iff $range(\rho) \subseteq Agent_H$

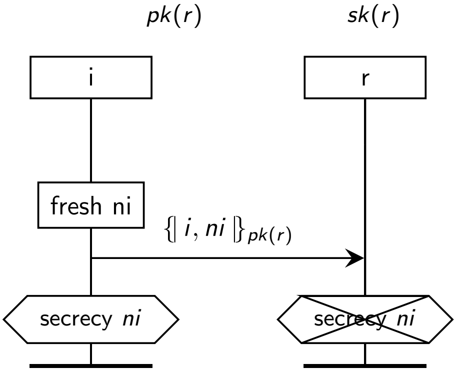
Formally, for a protocol P and a role R, the secrecy claim $\gamma = claim_l(R, secret, rt)$ is correct if

$$\begin{gathered} \text{for any } t \in traces(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t \\ honest((\theta, \rho, \sigma)) \text{ implies } AKN(t) \nvdash (\theta, \rho, \sigma)(rt), \end{gathered}$$

where $AKN([\alpha_1, \ldots, \alpha_n]) = AKN_n$.

Security properties: secrecy

Example: OSS Protocol (One-Sided Secrecy)



The OSS protocol: secrecy

$$OSS(i) = (\{i, r, ni, pk(r)\}, [send_1(i, r, \{\!| i, ni |\!\}_{pk(r)}), claim_2(i, secret, ni)])$$

$$OSS(r) = (\{i, r, sk(r)\}, [recv_1(i, r, \{\!| i, W |\!\}_{pk(r)}), claim_3(r, secret, W)])$$

Assume $\theta \in RID$ and $\rho = \{i \mapsto A, r \mapsto B\}$ such that $A, B \in Agent_H$.

Consider the following trace:

$$\begin{aligned} t = \; & [((\theta, \rho, \emptyset), create(r)), \\ & ((\theta, \rho, \{W \mapsto ne\}), recv_1(i, r, \{\!| i, W |\!\}_{pk(r)})) \\ & ((\theta, \rho, \{W \mapsto ne\}), claim_3(r, secret, W))] \end{aligned}$$

where $ne \in AdversaryFresh \subseteq AKN_0(P)$.

The OSS protocol: secrecy

$$OSS(i) = \begin{array}{l}(\{i, r, ni, pk(r)\},\\ [send_1(i, r, \{\!|\ i, ni\ |\!\}_{pk(r)}),\\ \ claim_2(i, secret, ni)])\end{array} \qquad OSS(r) = \begin{array}{l}(\{i, r, sk(r)\},\\ [recv_1(i, r, \{\!|\ i, W\ |\!\}_{pk(r)}),\\ \ claim_3(r, secret, W)])\end{array}$$

Assume $\theta \in RID$ and $\rho = \{i \mapsto A, r \mapsto B\}$ such that $A, B \in Agent_H$.

Consider the following trace:

$$t = \begin{array}{l}[((\theta, \rho, \emptyset), create(r)),\\ ((\theta, \rho, \{W \mapsto ne\}), recv_1(i, r, \{\!|\ i, W\ |\!\}_{pk(r)}))\\ \ ((\theta, \rho, \{W \mapsto ne\}), claim_3(r, secret, W))]\end{array}$$

where $ne \in AdversaryFresh \subseteq AKN_0(P)$.

If $\gamma = claim_3(r, secret, W)$ then $(\theta, \rho, \{W \mapsto ne\}), \gamma) \in t$ and $honest(\theta, \rho, \{W \mapsto ne\})$ but $AKN(t) \vdash ne = (\theta, \rho, \{W \mapsto ne\})(W)$

Consequently, the *secrecy* claim of the **responder** r does not hold.

However, the *secrecy* claim of the **initiator** role holds!

The OSS protocol: secrecy

$$OSS(i) = (\{i, r, ni, pk(r)\}, [send_1(i, r, \{\!| i, ni |\!\}_{pk(r)}), claim_2(i, secret, ni)])$$

$$OSS(r) = (\{i, r, sk(r)\}, [recv_1(i, r, \{\!| i, W |\!\}_{pk(r)}), claim_3(r, secret, W)])$$

We sketch a proof that $\zeta = claim_2(i, secret, ni)$ holds.

Assume that $t = [\alpha_1, \dots, \alpha_n] \in traces(OSS)$ and $inst = (\theta, \rho, \sigma) \in Inst$ such that $(inst, \zeta) \in t$ and $honest(inst)$.

(*) We assume that $AKN(t) \vdash inst(ni)$.

Hence there is the least $k < n$ such that $AKN_k \nvdash inst(ni)$ and $AKN_{k+1} \vdash inst(ni)$. We noticed that the only deduction rule that enriches the adversay knowledge is

[*send*], so
$AKN_{k+1} = AKN_k \cup \{inst(\{\!| i, ni |\!\}_{pk(r)})\} = AKN_k \cup \{\{\!| \rho(i), ni^{\#\theta} |\!\}_{pk(\rho(r))}\}$.

It follows that, $AKN_k \cup \{\{\!| \rho(i), ni^{\#\theta} |\!\}_{pk(\rho(r))}\} \vdash ni^{\#\theta}$. According to the deduction system on terms this is possible only if $sk(\rho(r))$ belongs to the adversary knowledge, but this is impossible since all agents are honest. Consequently, the assumption (*) is false and we proved by contradiction that the *secrecy* claim of the **initiator** role holds.

Security properties: authentication

Recall that our security properties are defined by (local) *claim* events.

In the sequel we shall analyze *aliveness*, which is a form of *authentication*. Our goal is to establish that a certain agent is "alive".

Let P be a protocol with the roles R and R'. We assume that R executes the claim $\gamma = \mathit{claim}_l(R, \mathit{alive}, R')$, which is correct if whenever R' is honest, he executed an event (action).

Authentication: aliveness

Recall that

$$\begin{array}{rcl} RoleEvent_R & ::= & send_{Label}(R, Role, RoleTerm) \\ & & \mid recv_{Label}(Role, R, RoleTerm) \\ & & \mid claim_{Label}(R, Claim[, RoleTerm]) \end{array}$$

where R is the role the event belongs to.

$RoleEvent = \bigcup\{RoleEvent_R \mid R \in Role\}$
$RunEvent = Inst \times (RoleEvent \cup \{create(R) \mid R \in Role\}$

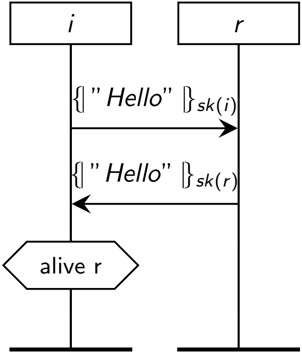
We define
$role : RoleEvent \rightarrow Role$, $role =$ the role the event belongs to
$actor : Inst \times RoleEvent \rightarrow Agent$, $actor((\theta, \rho, \sigma), re) = \rho(role(re))$

The claim $\gamma = claim_l(R, alive, R')$ is correct if, whenever R' is executed by an honest agent, this agent performed an event (action). Formally, γ is correct if

$$\text{for any } t \in traces(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$honest((\theta, \rho, \sigma)) \text{ implies there exists } ev \in t \text{ such that } actor(ev) = \rho(R').$$

Authentication: aliveness

We consider the following protocol:



The claim $\gamma = \mathit{claim}_l(i, \mathit{alive}, r)$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$\mathit{honest}((\theta, \rho, \sigma)) \text{ implies there exists } \mathit{ev} \in t \text{ such that } \mathit{actor}(\mathit{ev}) = \rho(r).$$

Authentication: aliveness

$i \longrightarrow i : \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(i)}$
$r \longrightarrow i : \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(r)}$

We sketch the proof that $\gamma = claim_l(i, alive, r)$ holds.

Assume that t is a trace and $((\theta, \rho, \sigma), \gamma)$ is a label such that $A = \rho(i)$ and $B = \rho(r)$ are honest agents.
Since A played his role correctly, there should be a label $((\theta, \rho, \sigma), recev_2(i, r, \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(r)})$, which means that A received a message encrypted with the secret key of B. Since an adversary cannot know the secret key of B, we infer that B actually sent the message $\{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(B)}$, so there should be a label $ev = ((\theta', \rho', \sigma'), sent_l(R_1, R_2, \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(R_1)}) \in t$ such that $\rho'(R_1) = B$.Consequently, $actor(ev) = \rho'(R_1) = B = \rho(r)$, and the claim is proved.

Authentication: aliveness

$i \longrightarrow r : \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(i)}$
$r \longrightarrow i : \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(r)}$

In the above proof for $\gamma = claim_I(i, alive, r)$ we found an event (action) $ev = ((\theta', \rho', \sigma'), sent_I(R_1, R_2, \{\!|\, \text{"}Hello\text{"} \,|\!\}_{sk(R_1)}) \in t$ such that $actor(ev) = \rho(r)$.

Note that:

- we don't know what roles played R_1 and R_2, so we don't know if R_1 played the correct role (the receiver),
- we don't know *when* the event took place, i.e., there is no relation between the runs θ and θ'.

The above property *alive* is a very weak form of authentication, so it will be called *weak-alive* and stronger versions of *aliveness* will be defined.

Authentication: Weak Aliveness

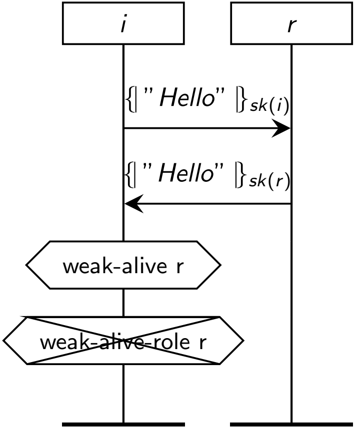
- Weak Aliveness:
  the claim $\gamma = \mathit{claim}_l(R, \mathit{weak\text{-}alive}, R')$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$\mathit{honest}((\theta, \rho, \sigma)) \text{ implies there exists } \mathit{ev} \in t \text{ such that } \mathit{actor}(\mathit{ev}) = \rho(R').$$

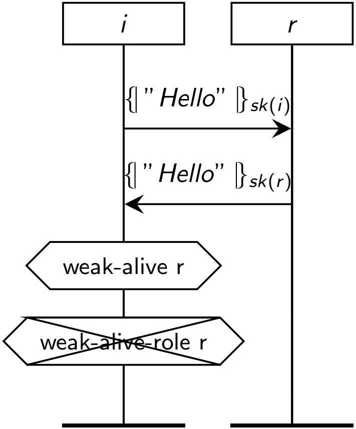
- Weak Aliveness in the Correct Role:
  the claim $\gamma = \mathit{claim}_l(R, \mathit{weak\text{-}alive\text{-}role}, R')$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$\mathit{honest}((\theta, \rho, \sigma)) \text{ implies there exists } \mathit{ev} \in t \text{ such that}$$
$$\mathit{actor}(\mathit{ev}) = \rho(R') \text{ and } \mathit{role}(\mathit{ev}) = R'.$$

Authentication: Weak Aliveness



Authentication: Weak Aliveness



An attack on the *weak-alive-role r* claim:

$$A \longrightarrow B : \{\!|\, "Hello" \,|\!\}_{sk(A)}$$
$$B \longrightarrow E : \{\!|\, "Hello" \,|\!\}_{sk(B)}$$
$$E \longrightarrow A : \{\!|\, "Hello" \,|\!\}_{sk(B)}$$

A thinks that B (correctly) played the receiver role, but B played the initiator role (in a seession with E).

Authentication: Recent Aliveness

Let $<_t$ be the order of the run events (labels) in the trace t.

- Recent Aliveness:
  the claim $\gamma = \mathit{claim}_l(R, \mathit{recent\text{-}alive}, R')$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$\mathit{honest}((\theta, \rho, \sigma)) \text{ implies there exists } \mathit{ev}, \mathit{ev}' \in t \text{ such that}$$
$$\mathit{actor}(\mathit{ev}) = \rho(R'),\ \mathit{runidof}(\mathit{ev}') = \mathit{runidof}(\mathit{inst}) \text{ and } \mathit{ev}' <_t \mathit{ev} <_t (\mathit{inst}, \gamma).$$

- Recent Aliveness in the Correct Role:
  the claim $\gamma = \mathit{claim}_l(R, \mathit{recent\text{-}alive\text{-}role}, R')$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t$$
$$\mathit{honest}((\theta, \rho, \sigma)) \text{ implies there exists } \mathit{ev}, \mathit{ev}' \in t \text{ such that}$$
$$\mathit{actor}(\mathit{ev}) = \rho(R'),\ \mathit{role}(\mathit{ev}) = R',$$
$$\mathit{runidof}(\mathit{ev}') = \mathit{runidof}(\mathit{inst}) \text{ and } \mathit{ev}' <_t \mathit{ev} <_t (\mathit{inst}, \gamma).$$

Examples in the seminar!

Thank you!

References

- Cremers, C. J. F. (2006). Scyther : semantics and verification of security protocols Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR614943
- Cremers C. and Mauw S. Operational Semantics and Verification of Security Protocols. Springer, 2012.