Special topics in Logic and Security I

Master Year II, Sem. I, 2022-2023

Ioana Leuștean
FMI, UB

References

1 Cremers, C. J. F. (2006). Scyther : semantics and verification of security protocols Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR614943

2 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: 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.

Security properties

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

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.

The security properties are defined as properties over traces.

Security properties: Secrecy and Aliveness

- The *secrecy* claim $\gamma = \mathit{claim}_l(R, \mathit{secret}, rt)$ holds if the information rt is not revealed to an adversary. Formally, for a protocol P and a role R, the secrecy claim $\gamma = \mathit{claim}_l(R, \mathit{secret}, rt)$ is correct if

$$\begin{array}{c}\text{for any } t \in \mathit{traces}(P) \text{ and any } ((\theta, \rho, \sigma), \gamma) \in t \\ \mathit{honest}((\theta, \rho, \sigma)) \text{ implies } \mathit{AKN}(t) \not\vdash (\theta, \rho, \sigma)(rt),\end{array}$$

  where $\mathit{AKN}([\alpha_1, \ldots, \alpha_n]) = \mathit{AKN}_n$.

- The *aliveness* claim $\gamma = \mathit{claim}_l(R, \mathit{alive}, R')$ is correct if, whenever the role R' is executed by an honest agent, this agent performed an event (action). Formally, γ is correct if

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

  Recall that we can define and study different forms of aliveness.

Authentication: Synchronization

Note that *aliveness* is a form of authentication that only requires that the communication partner executes some event, without any requirements on the messages he sends or receives.

Consequently, the Needham-Schroeder protocol (NSPK) satisfies the aliveness properties. However, we know that the protocol is not "secure", since the "man in the middle" attack is possible:

For analyzing this vulnerability a stronger form of *authentication is defined*: *synchornisation*. Informally, synchronisation states that the actual exchange of messages appears in the trace exactly as required in the description of the protocol.

In order to formally define synchronization, we introduce the *event order* and the *cast function*.

Event Order

For a protocol P we define:

- the Role Event Order is the total order $\epsilon_1 <_R \cdots <_R \epsilon_n$ where R is a role with $P(R) = (kn, [\epsilon_1, \ldots, \epsilon_n])$.
- the Communication Relation $\dashrightarrow \subseteq \mathit{RoleEvent} \times \mathit{RoleEvent}$ is defined by

  $\epsilon_1 \dashrightarrow \epsilon_2$ if and only if
  there exist $l \in \mathit{Label}$, $R, R' \in \mathit{Role}$, $rt_1, rt_2 \in \mathit{RoleTerm}$
  such that $\epsilon_1 = \mathit{send}_l(R, R', rt_1)$ and $\epsilon_2 = \mathit{recev}_l(R', R, rt_2)$
- the Protocol Order is

$$\prec_P = \left(\dashrightarrow \cup \bigcup_{R \in Role} <_R \right)^+$$

Protocol Order $\prec_P \subseteq \mathit{RoleEvent} \times \mathit{RoleEvent}$

For the NSPK protocol

$$
\begin{array}{lll}
NS(i) = & (\{i, r, ni, sk(i), pk(i), pk(r)\}, & \\
& [send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}), & \\
& recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}), & \\
& send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}), & \\
& claim_4(i, synch)]) &
\end{array}
\qquad
\begin{array}{lll}
NS(r) = & (\{i, r, nr, sk(r), pk(r), pk(i)\}, & \\
& [recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}), & \\
& send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}), & \\
& recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}), & \\
& claim_5(r, synch)]) &
\end{array}
$$

the event order is:

$$
\begin{array}{ccc}
send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}) & \prec_{NS} & recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}) \\
\curlywedge_{NS} & & \curlywedge_{NS} \\
recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}) & \succ_{NS} & send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}) \\
\curlywedge_{NS} & & \curlywedge_{NS} \\
send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}) & \prec_{NS} & recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}) \\
\curlywedge_{NS} & & \curlywedge_{NS} \\
claim_4(i, synch) & & claim_5(r, synch)
\end{array}
$$

Security properties: cast function

"As a protocol description can consist of a number of roles, which can be instantiated any number of times, we need some way to express which run is (supposedly) communicating with which other runs.

Therefore, we introduce the notion of a cast, borrowing intuition from a theatre play that is performed several times.

The cast for a particular performance of the play relates activity in the performance to particular roles. Likewise, the concrete activities in a protocol instance at the trace level, are assigned to the roles in the protocol. In the theater case, in different performances an actor can play different roles. Also, the same role can be taken up, for different performances of the play, by different actors. Likewise, run events associated with different roles may belong to the same agent and run events that are instances for the same role may belong to different agents.

For our purposes the coupling of roles and run events is more important, than the coupling of roles and actors."
[1, 3.3.2]

Security properties: cast function

For a protocol P a trace $t \in \mathit{traces}(P)$ a partial function

$$\Gamma : \mathit{RunEvent} \times \mathit{Role} \rightharpoonup \mathit{RID}$$

is a *cast* function if for all $cv = (\mathit{inst}, c) \in \mathit{RunEvent}$ with c a *claim* role event, the following properties hold:

- $\Gamma(cv, \mathit{role}(cv)) = \mathit{runidof}(cv)$
  (i.e. the run identifier of the analyzed claim event is executing the corresponding role)

  and

- $\Gamma(cv, R) = \theta$ if and only if
  for any $ev \in t$, $\mathit{runidof}(ev) = \theta$ implies $\mathit{role}(ev) = R$
  (i.e. the run identifier θ associated to a certain role is actually executing that role).

Example: trace for the Needham-Schroeder protocol

Let t be the following trace:

$[((1, \rho, \emptyset), create(i))$,
$((1, \rho, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$((2, \rho, \emptyset), create(r))$,
$((2, \rho, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$((2, \rho, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$((1, \rho, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$((1, \rho, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$((1, \rho, \{V \mapsto nr^{\#2}\}), claim_4(i, synch))$,
$((2, \rho, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$((2, \rho, \{W \mapsto ni^{\#1}\}), claim_5(r, synch))]$

For $cv = ((1, \rho, \{V \mapsto nr^{\#2}\}), claim_4(i, synch))$, a cast function is $\Gamma(cv, i) = 1$ and $\Gamma(cv, r) = 2$.

Authentication: Synchronization

For a protocol P the claim $\gamma = \mathit{claim}_l(R, \mathit{synch})$ is correct if

for any $t \in \mathit{traces}(P)$ there exists a cast function Γ such that

for any $(\mathit{inst}, \gamma) \in t$, if $\mathit{honest}(\mathit{inst})$ then

for any $\epsilon_1, \epsilon_2 \in \mathit{RoleEvent}$, $\epsilon_1 \dashrightarrow \epsilon_2$ and $\epsilon_2 \prec_P \gamma$
(i.e. for all communications occurring before the claim event in the protocol specification)

the corresponding run events

- occur in the right order in the trace,
- are executed in the runs defined by the the cast function Γ,
- have the same contents.

Authentication: Synchronization

For a protocol P the claim $\gamma = \mathit{claim}_l(R, \mathit{synch})$ is correct if

$$\text{for any } t \in \mathit{traces}(P) \text{ there exists a cast function } \Gamma \text{ such that}$$
$$\text{for any } (\mathit{inst}, \gamma) \in t, \text{ if } \mathit{honest}(\mathit{inst}) \text{ then}$$
$$\text{for any } \epsilon_1,\ \epsilon_2 \in \mathit{RoleEvent},\ \epsilon_1 \dashrightarrow \epsilon_2 \text{ and } \epsilon_2 \prec_P \gamma \text{ imply}$$
$$\text{there exists } \mathit{inst}_1, \mathit{inst}_2 \text{ such that}$$

$$(\mathit{inst}_1, \epsilon_1) <_t (\mathit{inst}_2, \epsilon_2) <_t (\mathit{inst}, \gamma) \text{ and}$$

$$\mathit{runidof}(\mathit{inst}_1) = \Gamma((\mathit{inst}, \gamma), \mathit{role}(\epsilon_1)) \text{ and}$$
$$\mathit{runidof}(\mathit{inst}_2) = \Gamma((\mathit{inst}, \gamma), \mathit{role}(\epsilon_2)) \text{ and}$$

$$\mathit{cont}((\mathit{inst}_1, \epsilon_1)) = \mathit{cont}((\mathit{inst}_2, \epsilon_2))$$

where *cont* is the content extraction function for *send* and *receive* events:

$$\mathit{cont}(\mathit{inst}, \mathit{send}_l(R, R', m)) = \mathit{inst}(R, R', m) \text{ and}$$
$$\mathit{cont}(\mathit{inst}, \mathit{recv}_l(R', R, m)) = \mathit{inst}(R', R, m)$$

Security properties: Synchronization

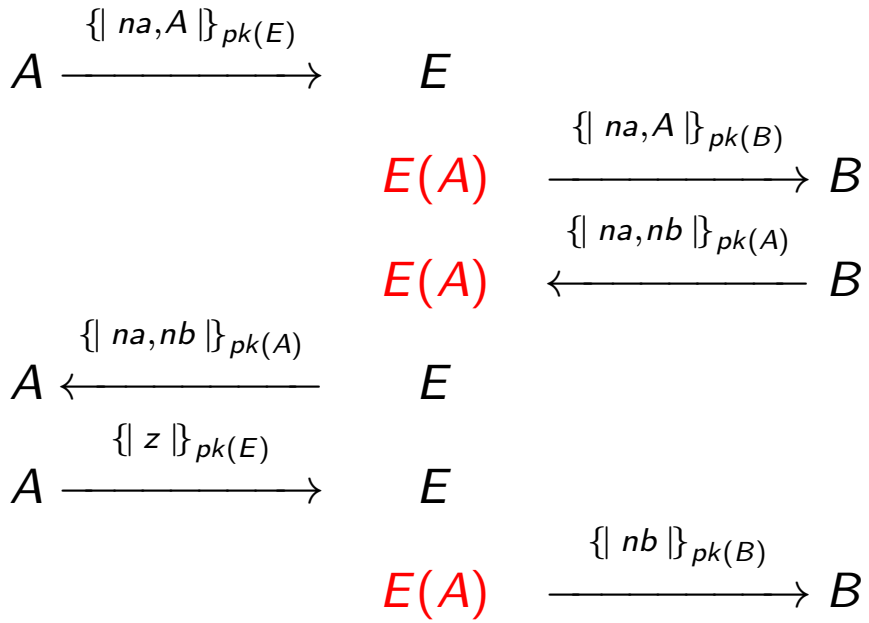
For a protocol P the claim $\gamma = claim_l(R, synch)$ is correct if

$$\begin{array}{c}
\text{for any } t \in traces(P) \text{ there exists a cast function } \Gamma \text{ such that} \\
\text{for any } (inst, \gamma) \in t, \text{ if } honest(inst) \text{ then} \\
\text{for any } \epsilon_1,\ \epsilon_2 \in RoleEvent,\ \epsilon_1 \dashrightarrow \epsilon_2 \text{ and } \epsilon_2 \prec_P \gamma \text{ imply} \\
\text{there exists } inst_1, inst_2 \text{ such that} \\
(inst_1, \epsilon_1) <_t (inst_2, \epsilon_2) <_t (inst, \gamma) \text{ and} \\
runidof(inst_1) = \Gamma((inst, \gamma), role(\epsilon_1)) \text{ and} \\
runidof(inst_2) = \Gamma((inst, \gamma), role(\epsilon_2)) \text{ and} \\
cont((inst_1, \epsilon_1)) = cont((inst_2, \epsilon_2))
\end{array}$$

Example:
We prove that the claim $\gamma = claim_5(R, synch)$ does not hold in the Needham-Schroeder protocol.

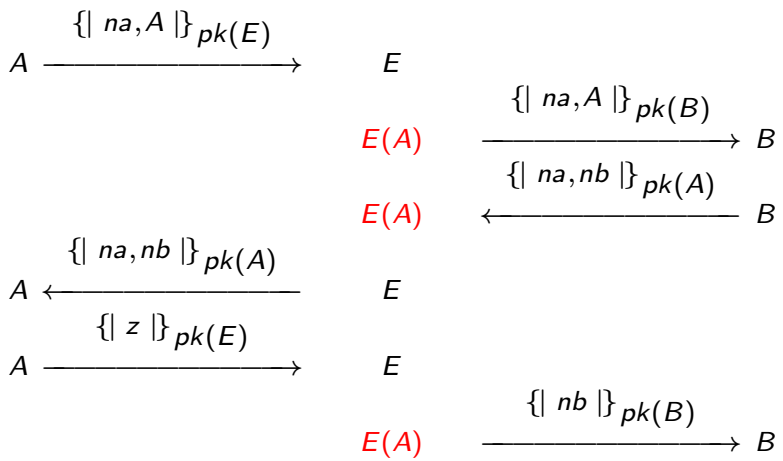
$$\begin{array}{lll}
NS(i) = & (\{i, r, ni, sk(i), pk(i), pk(r)\}, & NS(r) = \quad (\{i, r, nr, sk(r), pk(r), pk(i)\}, \\
& [send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}), & \quad [recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}), \\
& recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}), & \quad send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}), \\
& send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}), & \quad recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}), \\
& claim_4(i, synch)]) & \quad claim_5(r, synch)])
\end{array}$$

The "man in the middle" attack on NSPK



Obviously, $\{A, B\}$ are honest agents, E is a corrupted agent. We have to define a trace representing this attack.

The "man in the middle" attack on NSPK



Define $\rho_1 = \{i \mapsto A, r \mapsto E\}$ and $\rho_2 = \{i \mapsto A, r \mapsto B\}$, where $\{A, B\}$ are honest agents, E is a corrupted agent. A trace for the "man in the middle" attack is:

$[((1, \rho_1, \emptyset), create(i))$,
$((1, \rho_1, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \emptyset), create(r))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), claim_5(r, synch))]$

The "man in the middle" attack on NSPK

$\rho_1 = \{i \mapsto A, r \mapsto E\}$
$\rho_2 = \{i \mapsto A, r \mapsto B\}$

Let mt be the trace

$[((1, \rho_1, \emptyset), create(i))$,
$((1, \rho_1, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \emptyset), create(r))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), claim_5(r, synch))]$

We prove
there is NO cast function Γ s.t.
for any $(inst, \gamma) \in mt$, if
$honest((\theta, \rho, \sigma))$ then
for any ϵ_1, $\epsilon_2 \in RoleEvent$,
$\epsilon_1 \dashrightarrow \epsilon_2$ and $\epsilon_2 \prec_P \gamma$
imply
there exists $inst_1, inst_2$ s.t.
$(inst_1, \epsilon_1) <_{mt} (inst_2, \epsilon_2) <_{mt} (inst, \gamma)$
and
$runidof(inst_1) = \Gamma((inst, \gamma), role(\epsilon_1))$
and
$runidof(inst_2) = \Gamma((inst, \gamma), role(\epsilon_2))$
and
$cont((inst_1, \epsilon_1)) = cont((inst_2, \epsilon_2))$

The "man in the middle" attack on NSPK

$\rho_1 = \{i \mapsto A, r \mapsto E\}$
$\rho_2 = \{i \mapsto A, r \mapsto B\}$

Let mt be the trace

$[((1, \rho_1, \emptyset), create(i))$,
$((1, \rho_1, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \emptyset), create(r))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), claim_5(r, synch))]$

We prove
there is NO cast function Γ s.t.
for any $(inst, \gamma) \in mt$, if $honest(inst)$
then
for any ϵ_1, $\epsilon_2 \in RoleEvent$,
$\epsilon_1 \dashrightarrow \epsilon_2$ and $\epsilon_2 \prec_P \gamma$
imply
there exists $inst_1, inst_2$ s.t.
$(inst_1, \epsilon_1) <_{mt} (inst_2, \epsilon_2) <_{mt} (inst, \gamma)$
and
$runidof(inst_1) = \Gamma((inst, \gamma), role(\epsilon_1))$
and
$runidof(inst_2) = \Gamma((inst, \gamma), role(\epsilon_2))$
and
$cont((inst_1, \epsilon_1)) = cont((inst_2, \epsilon_2))$

The "man in the middle" attack on NSPK

$\rho_1 = \{i \mapsto A, r \mapsto E\}$
$\rho_2 = \{i \mapsto A, r \mapsto B\}$

Let mt be the trace

$[((1, \rho_1, \emptyset), create(i))$,
$((1, \rho_1, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \emptyset), create(r))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$((1, \rho_1, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$((2, \rho_2, \{W \mapsto ni^{\#1}\}), claim_5(r, synch))]$

Assume Γ is a cast function for mt.
If $inst = (2, \rho_2, \{W \mapsto ni^{\#1}\})$,
$\gamma = claim_5(r, synch)$ and
$cv = (inst, \gamma)$ then
$honest(inst)$,
$\Gamma(cv, r) = 2$ and $\Gamma(cv, i) = 1$.

Let $\epsilon_1 = send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)})$
$\epsilon_2 = recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)})$
and note that
$inst_1 = (1, \rho_1, \emptyset)$ and
$inst_2 = (2, \rho_2, \{W \mapsto ni^{\#1}\})$ are the only choices such that
$(inst_1, \epsilon_1) <_{mt} (inst_2, \epsilon_2) <_{mt} (inst, \gamma)$.

Since
$cont(inst_1, \epsilon_1) = (A, E, \{\!|\, ni^{\#1}, A \,|\!\}_{pk(E)})$
and
$cont(inst_2, \epsilon_2) = (A, B, \{\!|\, ni^{\#1}, A \,|\!\}_{pk(B)})$
are different, our assumption is false.

Secrecy in NSPK

In the sequel we prove that the same trace provides a counter example for the secrecy claim of the responder:

$$\begin{aligned} NS(r) = \quad & (\{i, r, nr, sk(r), pk(r), pk(i)\}, \\ & [recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}), \\ & send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}), \\ & recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}), \\ & claim_5(r, secret, nr)]) \end{aligned}$$

Recall that the adversary knowledge is enriched by the send rule:

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

so we follow the evolution of AKN along the trace.

Secrecy in NSPK

Let $\rho_1 = \{i \mapsto A, r \mapsto E\}$ and $\rho_2 = \{i \mapsto A, r \mapsto B\}$, where $\{A, B\}$ are honest agents, E is a corrupted agent.

$[((1, \rho_1, \emptyset), create(i))$,
$AKN_0 = Agent \cup \{pk(A), pk(B), pk(E)\} \cup \{sk(E)\}$

$((1, \rho_1, \emptyset), send_1(i, r, \{\!|\, ni, i \,|\!\}_{pk(r)}))$,
$AKN_1 = AKN_0 \cup \{\{\!|\, ni^{\#1}, A \,|\!\}_{pk(E)}\}$

$((2, \rho_2, \emptyset), create(r))$,
$AKN_2 = AKN_1$

$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_1(i, r, \{\!|\, W, i \,|\!\}_{pk(r)}))$,
$AKN_3 = AKN_2$

$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$AKN_4 = AKN_3 \cup \{\{\!|\, ni^{\#1}, ni^{\#2} \,|\!\}_{pk(A)}\}$

Secrecy in NSPK

$((2, \rho_2, \{W \mapsto ni^{\#1}\}), send_2(r, i, \{\!|\, W, nr \,|\!\}_{pk(i)}))$,
$AKN_4 = AKN_3 \cup \{\{\!|\, ni^{\#1}, ni^{\#2} \,|\!\}_{pk(A)}\}$

$((1, \rho_1, \{V \mapsto nr^{\#2}\}), recv_2(r, i, \{\!|\, ni, V \,|\!\}_{pk(i)}))$,
$AKN_5 = AKN_4$

$((1, \rho_1, \{V \mapsto nr^{\#2}\}), send_3(i, r, \{\!|\, V \,|\!\}_{pk(r)}))$,
$AKN_6 = AKN_5 \cup \{\{\!|\, nr^{\#2} \,|\!\}_{pk(E)}\}$

$((2, \rho_2, \{W \mapsto ni^{\#1}\}), recv_3(i, r, \{\!|\, nr \,|\!\}_{pk(r)}))$,
$AKN_7 = AKN_6$

$((2, \rho_2, \{W \mapsto ni^{\#1}\}), claim_5(r, secret, nr))]$
$AKN_7 \vdash nr^{\#2}$ since $\{sk(E), \{\!|\, nr^{\#2} \,|\!\}_{pk(E)}\} \subseteq AKN_7$

Thank you!

References

1 Cremers, C. J. F. (2006). Scyther : semantics and verification of security protocols Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR614943

2 Cremers C. and Mauw S. Operational Semantics and Verification of Security Protocols. Springer, 2012.