MPRI 2.30: Proofs of Security Protocols

2. Security Proofs

Adrien Koutsos

2023/2024

Security Proof

Private Authentication: Anonymity

Lets now try to prove that PA v2 provides **anonymity**:

- $\mathsf{I_X}$ is the initiator with identity X;
- $\mathsf{S_X}$ is the server, accepting messages from X;

The adversary must not be able to distinguish $\mathsf{I_A} \mid \mathsf{S_A}$ from $\mathsf{I_C} \mid \mathsf{S_A}$.

$$\begin{array}{lll} \mathsf{I_X} & : \nu\, \mathsf{r}.\ \ \nu\, \mathsf{n_I}. & \mathbf{out}(\mathtt{c_I}, \{\langle \mathsf{pk_X}\,,\, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}}) \\ \mathsf{S_X} & : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{c_I}, x). & \text{if } \pi_1(d) \doteq \mathsf{pk_X} \\ & & \quad \text{then } \mathbf{out}(\mathtt{c_S}, \{\langle \pi_2(d)\,,\, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \\ & & \quad \text{else } \ \mathbf{out}(\mathtt{c_S}, \{0\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \end{array}$$

We assume the encryption is IND-CCA_1 and KP-CCA_1.

Private Authentication: Anonymity

As we saw, an encryption **does not hide the length** of the plain-text. Hence, since $\mathsf{len}(\langle \mathsf{n_I}\, , \mathsf{n_S} \rangle) \neq \mathsf{len}(0)$, there is an attack:

$$\not\models \{\langle \mathsf{n_I}\, , \mathsf{n_S} \rangle\}_{\mathsf{pk_A}}^{\mathsf{r_0}} \sim \{0\}_{\mathsf{pk_C}}^{\mathsf{r_0}}$$

even if the encryption is IND-CCA$_1$ and KP-CCA$_1$.

Private Authentication: Anonymity

We **fix** the protocol by:

- adding a **length check**;
- using a **decoy** message of the correct length.

The PA Protocol, v3

$$
\begin{array}{ll}
\mathsf{I_X} : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}. & \mathbf{out}(\mathtt{c_I}, \{\langle \mathsf{pk_X}\,,\, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}}) \\
\mathsf{S_X} : \nu\, \mathsf{r_0}.\, \nu\, \mathsf{n_S}.\, \mathbf{in}(\mathtt{c_I}, x). & \text{if } \pi_1(d) \doteq \mathsf{pk_X} \mathrel{\dot\wedge} \mathsf{len}(\pi_2(d)) \doteq \mathsf{len}(\mathsf{n_S}) \\
& \quad \text{then } \mathbf{out}(\mathtt{c_S}, \{\langle \pi_2(d)\,,\, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \\
& \quad \text{else } \ \mathbf{out}(\mathtt{c_S}, \{\langle \mathsf{n_S}\,,\, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}})
\end{array}
$$

Private Authentication: Anonymity

$$\begin{array}{ll}\mathsf{I_X} : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}. & \mathbf{out}(\mathtt{c_I}, \{\langle \mathsf{pk_X}, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}}) \\ \mathsf{S_X} : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{c_I}, x). & \text{if } \pi_1(d) \doteq \mathsf{pk_X} \mathbin{\dot\wedge} \mathsf{len}(\pi_2(d)) \doteq \mathsf{len}(\mathsf{n_S}) \\ & \quad \text{then } \mathbf{out}(\mathtt{c_S}, \{\langle \pi_2(d), \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \\ & \quad \text{else } \ \mathbf{out}(\mathtt{c_S}, \{\langle \mathsf{n_S}, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \end{array}$$

To prove $\mathsf{I_A} \mid \mathsf{S_A} \approx \mathsf{I_C} \mid \mathsf{S_A}$, we have several **traces**:

$$\begin{array}{ll} \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}) & \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}), \mathbf{out}(\mathtt{c_I}) \\ \mathbf{out}(\mathtt{c_I}), \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}) & \mathbf{out}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I}) \\ \mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_I}) & \mathbf{out}(\mathtt{c_S}), \mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I}) \end{array}$$

Private Authentication: Anonymity

$$
\begin{array}{lll}
\mathsf{I_X} & : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}. & \mathbf{out}(\mathtt{c_I}, \{\langle \mathsf{pk_X}\ ,\ \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}}) \\
\mathsf{S_X} & : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{c_I}, x). & \text{if } \pi_1(d) \doteq \mathsf{pk_X} \mathrel{\dot\wedge} \mathsf{len}(\pi_2(d)) \doteq \mathsf{len}(\mathsf{n_S}) \\
 & & \quad \text{then } \mathbf{out}(\mathtt{c_S}, \{\langle \pi_2(d)\ ,\ \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \\
 & & \quad \text{else } \ \mathbf{out}(\mathtt{c_S}, \{\langle \mathsf{n_S}\ ,\ \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}})
\end{array}
$$

To prove $\mathsf{I_A} \mid \mathsf{S_A} \approx \mathsf{I_C} \mid \mathsf{S_A}$, we have several **traces**:

$\mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S})$ $\quad$ $\mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}), \mathbf{out}(\mathtt{c_I})$

$\mathbf{out}(\mathtt{c_I}), \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S})$ $\quad$ $\mathbf{out}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I})$

$\mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I}), \mathbf{out}(\mathtt{c_I})$ $\quad$ $\mathbf{out}(\mathtt{c_S}), \mathbf{out}(\mathtt{c_S}), \mathbf{in}(\mathtt{c_I})$

But there is a **more general trace**: its security implies the security of the other traces.

See **partial order reduction** (POR) techniques [1].

Private Authentication: Anonymity

We must prove that:

$$\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{C}]$$

where:

$$\begin{aligned}
\mathsf{out}_1^{X} &\equiv \{\langle \mathsf{pk}_X \,,\, \mathsf{n}_I \rangle\}_{\mathsf{pk}_S}^{\mathsf{r}} \\
\mathsf{out}_2^{X,Y}[M] &\equiv \text{if } \pi_1(d[M]) \doteq \mathsf{pk}_X \mathbin{\dot\wedge} \mathsf{len}(\pi_2(d[M])) \doteq \mathsf{len}(\mathsf{n}_S) \\
&\qquad\quad \text{then } \{\langle \pi_2(d[M]) \,,\, \mathsf{n}_S \rangle\}_{\mathsf{pk}_Y}^{\mathsf{r}_0} \\
&\qquad\quad \text{else } \{\langle \mathsf{n}_S \,,\, \mathsf{n}_S \rangle\}_{\mathsf{pk}_Y}^{\mathsf{r}_0} \\
d[M] &\equiv \mathsf{dec}(\mathbf{att}_0([M]), \mathsf{sk}_S)
\end{aligned}$$

Private Authentication: Anonymity

First, we push the branching under the encryption:

$$\frac{\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}] \quad \overline{\mathsf{out}_2^{A,A}[\mathsf{out}_1^{C}] = \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}]}}{\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{C}]}\ \mathrm{R}$$

where:

$$\underline{\mathsf{out}_2}^{X,Y}[M] \equiv \left\{ \begin{array}{l} \text{if } \pi_1(d[M]) \doteq \mathsf{pk}_X \mathbin{\dot\wedge} \mathsf{len}(\pi_2(d[M])) \doteq \mathsf{len}(n_S) \\ \ \text{then } \langle \pi_2(d[M]) \,,\, n_S \rangle \\ \ \text{else } \ \langle n_S \,,\, n_S \rangle \end{array} \right\}_{\mathsf{pk}_Y}^{r_0}$$

We let $m_X[M]$ be the content of the encryption above.

Private Authentication: Anonymity

Then, we use KP-CCA$_1$ to change the encryption key:

$$\dfrac{\begin{array}{rl} & \mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A,A}}[\mathsf{out}_1^{\mathsf{A}}] \\ \sim & \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A,C}}[\mathsf{out}_1^{\mathsf{C}}] \end{array} \qquad \dfrac{}{\begin{array}{rl} & \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A,C}}[\mathsf{out}_1^{\mathsf{C}}] \\ \sim & \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A,A}}[\mathsf{out}_1^{\mathsf{C}}] \end{array}}\ \textsc{KP-CCA}_1}{\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A,A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A,A}}[\mathsf{out}_1^{\mathsf{C}}]}\ \textsc{Trans}$$

since:

- the encryption randomness r_0 is correctly used;
- the key randomness n_A and n_B appear only in $\mathsf{pk}(\cdot)$ and $\mathsf{dec}(\_, \mathsf{sk}(\cdot))$ positions.

Private Authentication: Anonymity

Then, we use IND-CCA$_1$ to change the encryption content:

$$\dfrac{\begin{array}{l} \mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \\ \sim\ \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}] \end{array} \qquad \dfrac{\left[\mathsf{len}(m_{\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]) \doteq \mathsf{len}(m_{\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}])\right]}{\begin{array}{l} \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{A}}] \\ \sim\ \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}] \end{array}}\ \text{IND-CCA}_1}{\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]}\ \textsc{Trans}$$

since:

- the encryption randomness r_0 is correctly used;
- the key randomness n_{C} appear only in $\mathsf{pk}(\cdot)$ and $\mathsf{dec}(\_, \mathsf{sk}(\cdot))$ positions.

Private Authentication: Anonymity

Recall that:

$$m_{\mathsf{X}}[\mathsf{M}] \equiv \text{if } \pi_1(d[\mathsf{M}]) \doteq \mathsf{pk}_{\mathsf{X}} \mathbin{\dot\wedge} \mathsf{len}(\pi_2(d[\mathsf{M}])) \doteq \mathsf{len}(\mathsf{n}_{\mathsf{S}}) \\ \text{then } \langle \pi_2(d[\mathsf{M}]) \,,\, \mathsf{n}_{\mathsf{S}} \rangle \\ \text{else } \langle \mathsf{n}_{\mathsf{S}} \,,\, \mathsf{n}_{\mathsf{S}} \rangle$$

Then:

$$\frac{\mathcal{A}_{\mathsf{th}} \vdash_{\textsc{Gen}} \mathsf{len}(m_{\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]) \doteq \mathsf{len}(m_{\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}])}{\left[\mathsf{len}(m_{\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]) \doteq \mathsf{len}(m_{\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}])\right]} \ \textsc{gen}$$

if $\mathcal{A}_{\mathsf{th}}$ contains the axiom[1]:

$$\forall x, y. \mathsf{len}(\langle x \,,\, y \rangle) = c_{\langle \_ \,,\, \_ \rangle}(\mathsf{len}(x), \mathsf{len}(y))$$

where $c_{\langle \_ \,,\, \_ \rangle}(\cdot, \cdot)$ is left unspecified.

[1] This axiom must be satisfied by the protocol implementation for the security proof to apply.

Private Authentication: Anonymity

Then, we α-rename the key randomness n_{C}, rewrite back the encryption, and conclude.

$$\frac{}{\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]} \ \alpha\text{-}\textsc{equ} + \mathrm{R} + \textsc{Refl}$$

Privacy

Privacy

We proved **anonymity** of the Private Authentication protocol, which we defined as:

$$\mathsf{I_A} \mid \mathsf{S_A} \approx \mathsf{I_C} \mid \mathsf{S_A}$$

But does this really guarantees that this protocol protects the privacy of its users?

$\Rightarrow$ **No, because of linkability attacks**

Linkability Attacks

Consider the following authentication protocol, called KCL, between a reader R and a tag T_X with identity X:

$$\begin{aligned} R \;&: \nu\, n_R. & \mathbf{out}(c_R, n_R) \\ T_X \;&: \nu\, n_T.\, \mathbf{in}(c_R, x).\, & \mathbf{out}(c_I, \langle X \oplus n_T\,,\, n_T \oplus H(x, k_X)\rangle) \end{aligned}$$

Assuming H is a PRF (Pseudo-Random Function), and $\oplus$ is the exclusive-or, we can prove that KCL provides **anonymity**.

$$T_A \mid R \approx T_B \mid R$$

Linkability Attacks

But there are **privacy attacks** against KCL, using two sessions:

$1: \mathsf{E} \to \mathsf{T_A} : \mathsf{n}_R$	$\mathsf{E} \to \mathsf{T_A} : \mathsf{n}_R$
$2: \mathsf{T_A} \to \mathsf{E} : \langle \mathsf{A} \oplus \mathsf{n}_T , \mathsf{n}_T \oplus \mathsf{H}(\mathsf{n}_R, \mathsf{k_A})\rangle$	$\mathsf{T_A} \to \mathsf{E} : \langle \mathsf{A} \oplus \mathsf{n}_T , \mathsf{n}_T \oplus \mathsf{H}(\mathsf{n}_R, \mathsf{k_A})\rangle$
$3: \mathsf{E} \to \mathsf{T_A} : \mathsf{n_R}$	$\mathsf{E} \to \mathsf{T_B} : \mathsf{n_R}$
$4: \mathsf{T_A} \to \mathsf{E} : \langle \mathsf{A} \oplus \mathsf{n}'_\mathsf{T} , \mathsf{n}'_T \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A})\rangle$	$\mathsf{T_B} \to \mathsf{E} : \langle \mathsf{B} \oplus \mathsf{n}'_\mathsf{T} , \mathsf{n}'_T \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_B})\rangle$

Let t_2 and t_4 be the outputs of T. Then, on the left scenario:

$$\begin{aligned}\pi_2(t_2) \oplus \pi_2(t_4) &= \big(\mathsf{n}_T \oplus \mathsf{H}(\mathsf{n}_R, \mathsf{k_A})\big) \oplus \big(\mathsf{n}'_T \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A})\big) \\ &= \mathsf{n}_T \oplus \mathsf{n}'_T \\ &= \pi_1(t_2) \oplus \pi_1(t_4)\end{aligned}$$

The same equality check will almost never hold on the right, under reasonable assumption on H.

Linkability Attacks

We just saw an **attack** against:

$$(\mathsf{T_A} \mid \mathsf{R}) \mid (\mathsf{T_A} \mid \mathsf{R}) \approx (\mathsf{T_A} \mid \mathsf{R}) \mid (\mathsf{T_B} \mid \mathsf{R})$$

Unlinkability

To prevent such attacks, we need to prove a stronger property, called **unlinkability**. It requires to prove the **equivalence** between:

- a **real-world**, where each agent can run **many sessions**:

$$\nu\, \vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_N.\ !_{\mathrm{id} \leq N}\ !_{\mathtt{sid} \leq M}\ P(\vec{\mathsf{k}}_{\mathrm{id}})$$

- and an **ideal-world**, where each agent run at most a **single session**:

$$\nu\, \vec{\mathsf{k}}_{0,0}, \ldots, \vec{\mathsf{k}}_{N,M}.\ !_{\mathrm{id} \leq N}\ !_{\mathtt{sid} \leq M}\ P(\vec{\mathsf{k}}_{\mathrm{id},\mathtt{sid}})$$

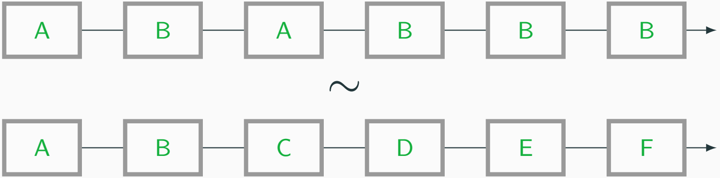
Remark

The processes above are parameterized by $N, M \in \mathbb{N}$. Unlinkability holds if the equivalence holds for any N, M.

For the sack of simplicity, we omit channel names.

Unlinkability

Example An unlinkability scenario.



Unlinkability: Intuition

In the **ideal-world**, relations between sessions **cannot leak** any **information** on identities.
$\Rightarrow$ hence **no link** can be **efficiently found** in the **real word**.

Unlinkability: Adding Servers

Our definition of **unlinkability** did not account for the **server**.

User-specific server, accepting a single identity.
The processes $P(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{U}})$ and $S(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{U}})$ are parameterized by:

- some **global** key material $\vec{\mathsf{k}}_{\mathsf{S}}$;
- and some **user-specific** key material $\vec{\mathsf{k}}_{\mathsf{U}}$.

Then, we require that:

$$
\begin{aligned}
& \nu\, \vec{\mathsf{k}}_{\mathsf{S}}.\ \nu\, \vec{\mathsf{k}}_{0}, \ldots, \vec{\mathsf{k}}_{N}. \quad\ \ !_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ \big(P(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{id}}) \quad \mid S(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{id}})\big) \\
\approx\ & \nu\, \vec{\mathsf{k}}_{\mathsf{S}}.\ \nu\, \vec{\mathsf{k}}_{0,0}, \ldots, \vec{\mathsf{k}}_{N,M}.\ !_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ \big(P(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{id}_{\mathtt{sid}}}) \mid S(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{id}_{\mathtt{sid}}})\big)
\end{aligned}
$$

Unlinkability: Adding Servers

Generic server, accepting all identities.

No changes for the user process $P(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_\mathsf{U})$.

The server $S(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_{\mathsf{U}_1}, \dots, \vec{\mathsf{k}}_{\mathsf{U}_M})$ is parameterized by:

- some **global** key material $\vec{\mathsf{k}}_\mathsf{S}$;
- **all users** key material $\vec{\mathsf{k}}_{\mathsf{U}_1}, \dots, \vec{\mathsf{k}}_{\mathsf{U}_M}$.

The we require that:

$$
\begin{aligned}
\nu\, \vec{\mathsf{k}}_\mathsf{S}.\ \nu\, \vec{\mathsf{k}}_0, \dots, \vec{\mathsf{k}}_N. \quad & \left(!_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ P(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_{\mathsf{id}})\right) \ | \\
& \left(!_{\le L}\ S(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_0, \dots, \vec{\mathsf{k}}_N)\right) \\
\approx \nu\, \vec{\mathsf{k}}_\mathsf{S}.\ \nu\, \vec{\mathsf{k}}_{0,0}, \dots, \vec{\mathsf{k}}_{N,M}. \ & \left(!_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ P(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_{\mathsf{id},\mathtt{sid}})\right) \ | \\
& \left(!_{\le L}\ S(\vec{\mathsf{k}}_\mathsf{S}, \vec{\mathsf{k}}_{0,0}, \dots, \vec{\mathsf{k}}_{N,M})\right)
\end{aligned}
$$

Unlinkability: Remark

Note that **user-specific unlinkability** is a very strong property that does not often hold.

Example

Assume S **leaks whether it succeeded** or not. This models the fact that the adversary can **distinguish success from failure**:

- e.g. because a door opens, which can be observed;
- or because success is followed by further communication, while failure is followed by a new authentication attempt.

Then the following unlinkability scenario **does not hold**:

$$(\underline{P(\vec{\mathsf{k}})} \mid S(\vec{\mathsf{k}})) \mid (P(\vec{\mathsf{k}}) \mid \underline{S(\vec{\mathsf{k}})}) \approx (\underline{P(\vec{\mathsf{k}}_0)} \mid S(\vec{\mathsf{k}}_0)) \mid (P(\vec{\mathsf{k}}_1) \mid \underline{S(\vec{\mathsf{k}}_1)})$$

✓ ✗

Private Authentication: Unlinkability

Private Authentication

We parameterize the initiator and server in **PA** by the key material:

$$\mathsf{I}(\mathsf{k_S}, \mathsf{k_X}) \ : \nu\, \mathsf{r}.\ \ \nu\, \mathsf{n_I}. \qquad \mathbf{out}(\mathsf{c_I}, \{\langle \mathsf{pk_X}\, ,\, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}})$$

$$\begin{aligned}\mathsf{S}(\mathsf{k_S}, \mathsf{k_X}) : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathsf{c_I}, x).\ & \text{if } \pi_1(d) \doteq \mathsf{pk_X} \mathbin{\dot\wedge} \mathsf{len}(\pi_2(d)) \doteq \mathsf{len}(\mathsf{n_S}) \\ & \quad \text{then } \mathbf{out}(\mathsf{c_S}, \{\langle \pi_2(d)\, ,\, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}) \\ & \quad \text{else } \ \mathbf{out}(\mathsf{c_S}, \{\langle \mathsf{n_S}\, ,\, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}})\end{aligned}$$

where $\mathsf{sk_X} \equiv \mathsf{sk}(\mathsf{k_X})$, $\mathsf{pk_X} \equiv \mathsf{pk}(\mathsf{k_X})$ and $d \equiv \mathsf{dec}(x, \mathsf{sk_S})$.

Private Authentication: Unlinkability

Theorem

Private Authentication, v3 satisfies the **unlinkability** property (with user-specific server). I.e., for all $N, M \in \mathbb{N}$:

$$\begin{array}{rl} & \nu\, \mathsf{k_S}.\, \nu\, \mathsf{k}_0, \ldots, \mathsf{k}_N. \quad\quad !_{\mathsf{id} \leq N}\, !_{\mathtt{sid} \leq M} \left(I(\mathsf{k_S}, \mathsf{k_{id}}) \quad \mid S(\mathsf{k_S}, \mathsf{k_{id}}) \right) \\ \approx & \nu\, \mathsf{k_S}.\, \nu\, \mathsf{k}_{0,0}, \ldots, \mathsf{k}_{N,M}.\, !_{\mathsf{id} \leq N}\, !_{\mathtt{sid} \leq M} \left(I(\mathsf{k_S}, \mathsf{k}_{\mathsf{id}_{\mathtt{sid}}}) \mid S(\mathsf{k_S}, \mathsf{k}_{\mathsf{id}_{\mathtt{sid}}}) \right) \end{array}$$

Proof sketch

For all N, M, for all trace of observables $\mathtt{tr}$, we show that:

$$\models \mathsf{fold}(\mathcal{P}_\mathcal{L}, \mathtt{tr}) \sim \mathsf{fold}(\mathcal{P}_\mathcal{R}, \mathtt{tr})$$

by induction over $\mathtt{tr}$, where $\mathcal{P}_\mathcal{L}$ and $\mathcal{P}_\mathcal{R}$ are, resp., the left and right protocols in the theorem above.

For details, see the Squirrel file `private-authentication-many.sp`.

Authentication Protocols

Authentication Protocol

We now focus on another class of security properties: **reachability** and **correspondance properties** (e.g. **authentication**)

These are properties on a **single** protocol, often expressed as a **temporal** property on **events** of the protocol. E.g.

> *If **Alice** accepts **Bob** at time τ then **Bob** must have initiated a session with **Alice** at time $\tau' < \tau$.*

To formalize the **cryptographic arguments** proving such properties, we will design a specialized **framework** and **proof system**.

The Hash-Lock Protocol

Let $\mathcal{I}$ be a finite set of identities.

Hash-Lock

$$\begin{array}{lll}
\mathsf{T}(\mathsf{A},\mathtt{i}) & : \nu\, \mathsf{n}_{\mathsf{T},\mathtt{i}}. & \mathbf{in}(\mathsf{c}^{\mathtt{T}}_{\mathsf{A},\mathtt{i}}, \mathsf{x}).\ \mathbf{out}(\mathsf{c}^{\mathtt{T}}_{\mathsf{A},\mathtt{i}}, \langle \mathsf{n}_{\mathsf{T},\mathtt{i}}\,,\, \mathsf{H}(\langle \mathsf{x}\,,\, \mathsf{n}_{\mathsf{T},\mathtt{i}}\rangle, \mathsf{k}_{\mathsf{A}})\rangle) \\
\mathsf{R}(\mathtt{j}) & : \nu\, \mathsf{n}_{\mathsf{R},\mathtt{j}}. & \mathbf{in}(\mathsf{c}^{\mathtt{R_1}}_{\mathtt{j}}, _).\ \mathbf{out}(\mathsf{c}^{\mathtt{R_1}}_{\mathtt{j}}, \mathsf{n}_{\mathsf{R},\mathtt{j}}). \\
 & & \mathbf{in}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{y}). \\
 & & \text{if } \dot{\bigvee}_{\mathsf{A}\in\mathcal{I}} \pi_2(\mathsf{y}) \doteq \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\, \pi_1(\mathsf{y})\rangle, \mathsf{k}_{\mathsf{A}}) \\
 & & \quad \text{then } \mathbf{out}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{ok}) \\
 & & \quad \text{else } \mathbf{out}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{ko})
\end{array}$$

We consider N sessions of each tag, and M sessions of the reader:

$$\nu\,(\mathsf{k}_{\mathsf{A}})_{\mathsf{A}\in\mathcal{I}}.\ \big(!_{\mathsf{A}\in\mathcal{I}}\ !_{\mathtt{i}<N}\ \mathsf{T}(\mathsf{A},\mathtt{i})\big)\ \big|\ \big(!_{\mathtt{j}<M}\ \mathsf{R}(\mathtt{j})\big)$$

Remark: we let the adversary do the scheduling between parties.

Notations

- we let $\leq$ be the **prefix relation** over observable traces:

$$\mathtt{tr}_0 \leq \mathtt{tr}_1 \text{ iff. } \exists \mathtt{tr}'.\ \mathtt{tr}_1 = \mathtt{tr}_0; \mathtt{tr}'$$

- $\mathtt{tr} : \mathtt{c}$ states that $\mathtt{tr}$ **ends with an output** on $\mathtt{c}$:

$$\mathtt{tr} : \mathtt{c} \text{ iff. } \exists \mathtt{tr}'.\ \mathtt{tr} = \mathtt{tr}'; \mathbf{out}(\mathtt{c})$$

Remark: $\mathtt{tr} : \mathtt{c} \leq \mathtt{tr}'$ means $\mathtt{tr} : \mathtt{c} \wedge \mathtt{tr} \leq \mathtt{tr}'$.

POR Result (Assumed)

We let $\mathcal{T}_{\mathsf{io}}$ be the set of observable traces where all outputs are always **directly preceded** by an input on the same channel, i.e.:

$$\mathtt{tr} \in \mathcal{T}_{\mathsf{io}} \quad \text{iff.} \quad \forall \mathtt{tr}' : \mathtt{c} \leq \mathtt{tr}.\, \exists \mathtt{tr}''.\, \mathtt{tr}' = \mathtt{tr}''; \mathbf{in}(\mathtt{c}); \mathbf{out}(\mathtt{c})$$

Assumption: POR

We admit that to analyze the Hash-Lock protocol, it is sufficient to consider only observables traces in $\mathcal{T}_{\mathsf{io}}$.

Authentication

Informal Definition

If the j-th session of R accepts believing it talked to tag A, then:

- there exists a session i of tag A **properly interleaved** with the j-th session of R;
- **messages** have been **properly forwarded** between the i-th session of tag A and the j-th session of R.

The second condition is often relaxed to require only a partial correspondence between messages.

Authentication of the Hash-Lock Protocol

For any $\mathtt{tr} : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_2} \in \mathcal{T}_{\mathsf{io}}$, we let $\textsf{accept}^{\mathsf{A}}@\mathtt{tr}$ be a term (defined later) stating that the reader accepts the tag A at the end of the trace $\mathtt{tr}$.

Authentication of the Hash-Lock Protocol

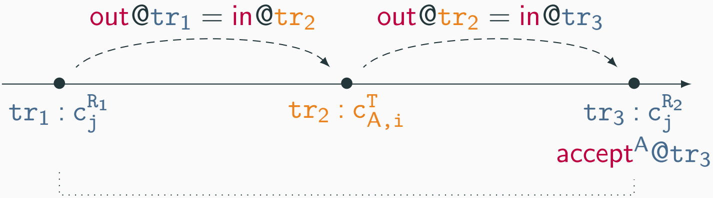
Informally, Hash-Lock provides authentication if for all $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$, $\mathtt{tr}_1 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_1}$ and $\mathtt{tr}_3 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_2}$ such that:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr} \qquad \text{and} \qquad \mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3$$

there must exists $\mathtt{tr}_2 : \mathtt{c}_{\mathtt{A,i}}^{\mathtt{T}}$ such that $\mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3$ and:

$$\mathsf{out}@\mathtt{tr}_1 = \mathsf{in}@\mathtt{tr}_2 \wedge \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3$$

Graphically:



Authentication of the Hash-Lock Protocol

What do we lack to formalize and prove the **authentication** of the **Hash-Lock** protocol?

- define the (generic) **terms representing** the output, input and acceptance, which we need to state the property;
- have a set of sound **one-sided** rules, to do the proof.

Authentication Protocols

Macro Terms

Notations: Predecessor

For any **observable trace** $\texttt{tr}$ and **observable** α, we let:

$$\mathsf{pred}(\texttt{tr}; \alpha) \stackrel{\mathsf{def}}{=} \texttt{tr}$$

Macro Terms

We now define some **generic** terms by **induction** of the observable trace $\mathtt{tr}$.

Let $\mathcal{P}$ be a action-deterministic protocol and $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$ with j inputs. If $\mathsf{fold}(\mathcal{P}, \mathtt{tr}) = t_1, \ldots, t_n$ then we let:

$$\mathsf{out}_{\mathcal{P}}@\mathtt{tr} \stackrel{\mathsf{def}}{=} \begin{cases} t_n & \text{if } \exists \mathtt{c}.\, \mathtt{tr} : \mathtt{c} \\ \mathsf{empty} & \text{otherwise} \end{cases}$$

$$\mathsf{frame}_{\mathcal{P}}@\mathtt{tr} \stackrel{\mathsf{def}}{=} \begin{cases} \langle \mathsf{frame}_{\mathcal{P}}@\mathsf{pred}(\mathtt{tr}) \,,\, \mathsf{out}_{\mathcal{P}}@\mathtt{tr} \rangle & \text{if } \mathtt{tr} \neq \epsilon \\ \mathsf{empty} & \text{if } \mathtt{tr} = \epsilon \end{cases}$$

$$\mathsf{in}_{\mathcal{P}}@(\mathtt{tr}; \mathbf{in}(\mathtt{c}); \mathbf{out}(\mathtt{c})) \stackrel{\mathsf{def}}{=} \begin{cases} \mathbf{att}_j(\mathsf{frame}_{\mathcal{P}}@\mathtt{tr}) & \text{if } \mathtt{tr} \neq \epsilon \\ \mathbf{att}_0() & \text{if } \mathtt{tr} = \epsilon \end{cases}$$

Remark: we omit $\mathcal{P}$ when it is clear from context.

The restriction to traces in $\mathcal{T}_{io}$ simplifies the definition of $\mathsf{in}_{\mathcal{P}}@\mathtt{tr}$.

Macro Terms

$\mathsf{frame}_{\mathcal{P}}@\mathtt{tr}$ contains **all** the **information known** to an **adversary** against $\mathcal{P}$ after the execution of $\mathtt{tr}$.

More precisely, we can show that for all action-deterministic processes $\mathcal{P}$ and $\mathcal{Q}$, for all $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$:

$$\mathcal{M} \models \mathsf{fold}(\mathcal{P}, \mathtt{tr}) \sim \mathsf{fold}(\mathcal{Q}, \mathtt{tr}) \ \text{ iff. } \ \mathcal{M} \models \mathsf{frame}_{\mathcal{P}}@\mathtt{tr} \sim \mathsf{frame}_{\mathcal{Q}}@\mathtt{tr}$$

for any $\mathcal{M}$ satisfying:

$$[\pi_1\langle x\,,\, y\rangle \doteq x] \qquad\qquad [\pi_2\langle x\,,\, y\rangle \doteq y]$$

Proof

$\Rightarrow$ apply FA to build $\mathsf{frame}_{\mathcal{R}}@\mathtt{tr}$ from $\mathsf{fold}(\mathcal{R}, \mathtt{tr})$ for $\mathcal{R} \in \{\mathcal{P}, \mathcal{Q}\}$

$\Leftarrow$ apply FA + $\textsc{Dup}$ + the pair injectivity rules to compute all terms in $\mathsf{fold}(\mathcal{R}, \mathtt{tr})$ from $\mathsf{frame}_{\mathcal{R}}@\mathtt{tr}$ for $\mathcal{R} \in \{\mathcal{P}, \mathcal{Q}\}$

Hash-Lock: Accept

Hash-Lock

$$
\begin{aligned}
&\mathsf{T}(A, \mathtt{i}) : \nu\, n_{T,\mathtt{i}}.\ \mathbf{in}(c^{\mathtt{T}}_{A,\mathtt{i}}, x).\ \mathbf{out}(c^{\mathtt{T}}_{A,\mathtt{i}}, \langle n_{T,\mathtt{i}}\,,\ H(\langle x\,,\ n_{T,\mathtt{i}}\rangle, k_A)\rangle)\\
&\mathsf{R}(\mathtt{j}) \quad\ : \nu\, n_{R,\mathtt{j}}.\ \mathbf{in}(c^{\mathtt{R_1}}_{\mathtt{j}}, \_).\ \mathbf{out}(c^{\mathtt{R_1}}_{\mathtt{j}}, n_{R,\mathtt{j}}).\\
&\qquad\qquad\qquad \mathbf{in}(c^{\mathtt{R_2}}_{\mathtt{j}}, y).\\
&\qquad\qquad\qquad \text{if } \dot{\bigvee}_{A\in\mathcal{I}}\, \pi_2(y) \doteq H(\langle n_{R,\mathtt{j}}\,,\ \pi_1(y)\rangle, k_A)\\
&\qquad\qquad\qquad\ \text{then } \mathbf{out}(c^{\mathtt{R_2}}_{\mathtt{j}}, \mathrm{ok})\\
&\qquad\qquad\qquad\ \text{else } \mathbf{out}(c^{\mathtt{R_2}}_{\mathtt{j}}, \mathrm{ko})
\end{aligned}
$$

To be able to state some **authentication** property of Hash-Lock, we need an additional macro. For all $\mathtt{tr} : c^{\mathtt{R_2}}_{\mathtt{j}} \in \mathcal{T}_{\mathrm{io}}$, we let:

$$
\mathrm{accept}^{A}@\mathtt{tr} \stackrel{\mathrm{def}}{=} \pi_2(\mathrm{in}@\mathtt{tr}) \doteq H(\langle n_{R,\mathtt{j}}\,,\ \pi_1(\mathrm{in}@\mathtt{tr})\rangle, k_A)
$$

We made sure that all names in the protocol are unique, so that they don't have to be renamed during the folding.

Authentication: Hash-Lock

The following formulas encode the fact that the **Hash-Lock** protocol provides **authentication**:

$$\forall \mathsf{A} \in \mathcal{I}.\ \forall \mathtt{tr} \in \mathcal{T}_{\mathsf{io}}.\ \forall \mathtt{tr}_1 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_1}, \mathtt{tr}_3 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_2} \text{ s.t. } \mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr},$$

$$\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3 \mathrel{\dot{\rightarrow}} \dot{\bigvee_{\substack{\mathtt{tr}_2 : \mathtt{c}_{\mathtt{A},\mathtt{i}}^{\mathtt{T}} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}}} \begin{array}{l} \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2 \mathrel{\dot{\wedge}} \\ \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3 \end{array} \sim \mathsf{true}$$

This kind of one-sided formulas are called **reachability formulas**. Proving the validity of such formulas requires **additional rules**, to allow for **propositional reasoning**.

Authentication Protocols

Reachability Proof System

Reachability Judgements

We define a **judgments** dedicated to **reachability correspondance properties**.

Definition

A **reachability judgement** $\Gamma \vdash t$ comprises a sequence of terms $\Gamma = t_1 \dot{\rightarrow} \cdots \dot{\rightarrow} t_n$ and a (boolean) term t.

$\Gamma \vdash t$ is **valid** if and only if the following formula is valid:

$$[t_1 \dot{\rightarrow} \cdots \dot{\rightarrow} t_n \dot{\rightarrow} t]$$

Boolean Connectives in Reachability Judgements

Careful not to confuse the boolean connectives at the **reachability** and **equivalence** levels!

Exercise

Determine which directions are correct.

$$t_\phi \mathbin{\dot\wedge} t_\psi \sim \mathsf{true} \quad \overset{?}{\dot\Leftrightarrow} \quad t_\phi \sim \mathsf{true} \wedge t_\psi \sim \mathsf{true}$$

$$t_\phi \mathbin{\dot\vee} t_\psi \sim \mathsf{true} \quad \overset{?}{\dot\Leftrightarrow} \quad t_\phi \sim \mathsf{true} \vee t_\psi \sim \mathsf{true}$$

$$t_\phi \mathbin{\dot\rightarrow} t_\psi \sim \mathsf{true} \quad \overset{?}{\dot\Leftrightarrow} \quad t_\phi \sim \mathsf{true} \rightarrow t_\psi \sim \mathsf{true}$$

Boolean Connectives in Reachability Judgements

Careful not to confuse the boolean connectives at the **reachability** and **equivalence** levels!

Exercise

Determine which directions are correct.

$$t_\phi \dot{\land} t_\psi \sim \mathsf{true} \quad \Leftrightarrow \quad t_\phi \sim \mathsf{true} \land t_\psi \sim \mathsf{true}$$

$$t_\phi \dot{\lor} t_\psi \sim \mathsf{true} \quad \Leftarrow \quad t_\phi \sim \mathsf{true} \lor t_\psi \sim \mathsf{true}$$

$$t_\phi \dot{\to} t_\psi \sim \mathsf{true} \quad \Rightarrow \quad t_\phi \sim \mathsf{true} \to t_\psi \sim \mathsf{true}$$

The second relation works both ways when t_ϕ or t_ψ is a **constant** formula.

Reachability Proof System

Our **reachability judgements** can be trivially equipped with a **sequent calculus**.

$$\frac{}{\Gamma, t_\phi \vdash t_\phi} \qquad \frac{\Gamma \vdash t_\psi \qquad \Gamma, t_\psi \vdash t_\phi}{\Gamma \vdash t_\phi}$$

$$\frac{\Gamma \vdash t_\psi \qquad \Gamma \vdash t_\phi}{\Gamma \vdash t_\psi \mathbin{\dot\wedge} t_\phi} \qquad \frac{\Gamma, t_\psi, t_\phi \vdash t_\theta}{\Gamma, t_\psi \mathbin{\dot\wedge} t_\phi \vdash t_\theta}$$

$$\frac{\Gamma \vdash t_\phi}{\Gamma \vdash t_\psi \mathbin{\dot\vee} t_\phi} \qquad \frac{\Gamma \vdash t_\psi}{\Gamma \vdash t_\psi \mathbin{\dot\vee} t_\phi} \qquad \frac{\Gamma, t_\psi \vdash t_\theta \qquad \Gamma, t_\phi \vdash t_\theta}{\Gamma, t_\psi \mathbin{\dot\vee} t_\phi \vdash t_\theta}$$

$$\frac{\Gamma \vdash t_\psi \qquad \Gamma, t_\phi \vdash t_\theta}{\Gamma, t_\psi \mathbin{\dot\to} t_\phi \vdash t_\theta} \qquad \frac{\Gamma, t_\psi \vdash t_\phi}{\Gamma \vdash t_\psi \mathbin{\dot\to} t_\phi}$$

Reachability Proof System (cont.)

$$\frac{\Gamma, t_\phi \vdash \bot}{\Gamma \vdash \neg t_\phi} \qquad \frac{}{\Gamma, \bot \vdash t_\phi}$$

$$\frac{\Gamma_1, t_\phi, t_\psi, \Gamma_2 \vdash t_\theta}{\Gamma_1, t_\psi, t_\phi, \Gamma_2 \vdash t_\theta} \qquad \frac{\Gamma, t_\psi, t_\psi \vdash t_\phi}{\Gamma, t_\psi \vdash t_\phi}$$

Reachability Proof System: Soundness

The reachability proof system is **sound**.

Proof

First, recall that for any Γ and t_θ:

$$\Gamma \vdash t_\theta \text{ is valid iff. } \Pr_\rho \left([\![(\dot{\wedge}\Gamma) \mathrel{\dot{\wedge}} \dot{\neg} t_\phi]\!]_{\mathbb{M}}^{\eta,\rho} \right) \text{ is negligible.} \qquad (\dagger)$$

Reachability Proof System: Soundness

We will only detail one rule, say:

$$\frac{\Gamma, t_\psi \vdash t_\theta \qquad \Gamma, t_\phi \vdash t_\theta}{\Gamma, t_\psi \mathbin{\dot\vee} t_\phi \vdash t_\theta.}$$

By the previous remark (†), since $(\Gamma, t_\psi \vdash t_\theta)$ and $(\Gamma, t_\phi \vdash t_\theta)$ are valid

- $\Pr_\rho\left(\llbracket(\dot\wedge\Gamma) \mathbin{\dot\wedge} t_\psi \mathbin{\dot\wedge} \dot\neg t_\theta\rrbracket_{\mathbb{M}}^{\eta,\rho}\right)$ is negligible.
- $\Pr_\rho\left(\llbracket(\dot\wedge\Gamma) \mathbin{\dot\wedge} t_\phi \mathbin{\dot\wedge} \dot\neg t_\theta\rrbracket_{\mathbb{M}}^{\eta,\rho}\right)$ is negligible.

Since the union of two negligible (η-indexed families of) events is a negligible (η-indexed families of) events,

$$\begin{aligned} & \Pr_\rho\left(\llbracket((\dot\wedge\Gamma) \mathbin{\dot\wedge} t_\psi \mathbin{\dot\wedge} \dot\neg t_\theta) \mathbin{\dot\vee} ((\dot\wedge\Gamma) \mathbin{\dot\wedge} t_\phi \mathbin{\dot\wedge} \dot\neg t_\theta)\rrbracket_{\mathbb{M}}^{\eta,\rho}\right) \text{ is negligible} \\ \Leftrightarrow \quad & \Pr_\rho\left(\llbracket(\dot\wedge\Gamma) \mathbin{\dot\wedge} (t_\psi \mathbin{\dot\vee} t_\phi) \mathbin{\dot\wedge} \dot\neg t_\theta\rrbracket_{\mathbb{M}}^{\eta,\rho}\right) \text{ is negligible} \end{aligned}$$

Hence using (†) again, $\Gamma, t_\psi \mathbin{\dot\vee} t_\phi \vdash t_\theta$ is valid.

Authentication Protocols

Cryptographic Rule: Collision Resistance

Cryptographic Hash

A **keyed cryptographic hash** $\mathsf{H}(\_,\_)$ is **computationally collision resistant** if no PPTM adversary can built collisions, even when it has access to a hashing **oracle**.

More precisely, a hash is *collision resistant under hidden key attacks* (CR-HK) iff for every PPTM $\mathcal{A}$, the following quantity:

$$\mathsf{Pr}_{\mathsf{k}}\left(\mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle m_1\,,\, m_2\rangle, m_1 \neq m_2 \text{ and } \mathsf{H}(m_1,\mathsf{k}) = \mathsf{H}(m_2,\mathsf{k})\right)$$

is negligible, where k is drawn uniformly in $\{0,1\}^\eta$.

CR Rule

Collision Resistance

If H is a CR-HK function, then the *ground* rule:

$$\frac{}{\mathsf{H}(m_1, \mathsf{k}) \doteq \mathsf{H}(m_2, \mathsf{k}) \mathrel{\dot{\rightarrow}} m_1 \doteq m_2 \sim \mathsf{true}}\ \mathrm{CR}$$

is sound, when k appears only in H key positions in m_1, m_2.

CR Rule: Exercise

Exercise

Let H be CR-HK. Show that the following rule is **not** sound:

$$\frac{}{\dot{\neg}(\mathsf{H}(m_1, \mathsf{k}) \doteq \mathsf{H}(m_2, \mathsf{k})) \sim \mathsf{true}}\ \mathrm{CR}$$

when k appears only in H key positions in m_1, m_2 and $m_1 \not\equiv m_2$.

Authentication Protocols

Cryptographic Rule: Message Authentication Code

Message Authentication Code

A **message authentication code** is a symmetric cryptographic schema which:

- create **message authentication codes** using $\text{mac}_\_(\_)$
- **verifies** mac using $\text{verify}_\_(\_,\_)$

It must satisfies the functional equality:

$$\text{verify}_k(\text{mac}_k(m), m) = \text{true}$$

MAC Security

A MAC must be **computationally unforgeable**, even when the adversary has access to a mac and verify **oracles**.

A MAC is *unforgeable against chosen-message attacks* (EUF-CMA) iff for every PPTM $\mathcal{A}$, the following quantity:

$$\Pr_{\mathsf{k}}\left(\begin{array}{r}\mathcal{A}^{\mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)},\mathcal{O}_{\mathsf{verify}_{\mathsf{k}}(\cdot,\cdot)}}(1^\eta) = \langle m\,,\,\sigma\rangle,\ m \text{ not queried to } \mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)} \\ \text{and } \mathsf{verify}_{\mathsf{k}}(\sigma, m) = 1 \quad\end{array}\right)$$

is negligible, where k is drawn uniformly in $\{0,1\}^\eta$.

EUF-MAC Rule

Take two messages s, m and a key $\mathsf{k} \in \mathcal{N}$ such that

- s and m are ground.
- $\mathsf{k} \in \mathcal{N}$ appears only in mac or verify key positions in s, m.

Key Idea

To build a rule for EUF-CMA, we proceed as follow:

- Compute $[\![s, m]\!]$ bottum-up, calling $\mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)}$ and $\mathcal{O}_{\mathsf{verify}_{\mathsf{k}}(\cdot,\cdot)}$ if necessary.
- Log all sub-terms $\mathbb{S}_{\mathsf{mac}}(s, m)$ sent to $\mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)}$.

$\Rightarrow$ If $\mathsf{verify}_{\mathsf{k}}(s, m)$ then $m = u$ for some $u \in \mathbb{S}_{\mathsf{mac}}(s, m)$.

*$\mathbb{S}_{mac}(s, m)$ are the **calls** to $\mathcal{O}_{mac_{k}(\cdot)}$ needed to compute s, m.*

EUF-MAC Rule

$\mathbb{S}_{\mathsf{mac}}(\cdot)$ defined by induction on ground terms:

$$
\begin{aligned}
\mathbb{S}_{\mathsf{mac}}(\mathsf{n}) &\stackrel{\mathsf{def}}{=} \emptyset \\
\mathbb{S}_{\mathsf{mac}}(\mathsf{verify}_{\mathsf{k}}(u_1, u_2)) &\stackrel{\mathsf{def}}{=} \mathbb{S}_{\mathsf{mac}}(u_1) \cup \mathbb{S}_{\mathsf{mac}}(u_2) \\
\mathbb{S}_{\mathsf{mac}}(\mathsf{mac}_{\mathsf{k}}(u)) &\stackrel{\mathsf{def}}{=} \{u\} \cup \mathbb{S}_{\mathsf{mac}}(u) \\
\mathbb{S}_{\mathsf{mac}}(f(u_1, \dots, u_n)) &\stackrel{\mathsf{def}}{=} \bigcup_{1 \le i \le n} \mathbb{S}_{\mathsf{mac}}(u_i) \qquad \text{(for other cases)}
\end{aligned}
$$

EUF-MAC Rule

Message Authentication Code Unforgeability

If mac is an EUF-CMA function, then the *ground* rule:

$$\frac{}{\mathsf{verify}_{\mathsf{k}}(s, m) \dot{\rightarrow} \dot{\bigvee}_{u \in \mathcal{S}} m \doteq u \sim \mathsf{true}} \text{ EUF-MAC}$$

is sound, when:

- $\mathcal{S} = \mathbb{S}_{\mathsf{mac}}(s, m)$;
- $\mathsf{k} \in \mathcal{N}$ appears only in mac or verify key positions in s, m.

Example

If t_1 t_2 and t_3 are terms which do not contain k, then:

$$\Phi \equiv \mathsf{mac}_{\mathsf{k}}(t_1), \mathsf{mac}_{\mathsf{k}}(t_2), \mathsf{mac}_{\mathsf{k}_0}(t_3)$$

$$\models \mathsf{verify}_{\mathsf{k}}(g(\Phi), \mathsf{n}) \ \dot{\rightarrow}\ \left(\mathsf{n} \doteq t_1 \mathbin{\dot{\vee}} \mathsf{n} \doteq t_2\right) \sim \mathsf{true}$$

EUF-MAC Rule: Exercise

Exercise

Assume mac is EUF-CMA. Show that the following rule is sound:

$$\frac{}{\mathsf{verify}_{\mathsf{k}}(\text{if } b \text{ then } s_0 \text{ else } s_1, m) \mathrel{\dot{\rightarrow}} \dot{\bigvee}_{u \in \mathcal{S}_1 \cup \mathcal{S}_2} m \doteq u \sim \mathsf{true}}$$

when b, s_0, s_1, m are *ground* terms, and:

- $\mathcal{S}_i = \{u \mid \mathsf{mac}_{\mathsf{k}}(u) \in \mathbb{S}_{\mathsf{mac}}(s_i, m)\}$, for $i \in \{0, 1\}$;
- k appears only in mac or verify key positions in s_0, s_1, m.

Remark: we do not make *any* assumption on b, except that it is ground. E.g., we can have $b \equiv (\mathbf{att}(\mathsf{k}) \doteq \mathsf{mac}_{\mathsf{k}}(0))$.

Authentication Protocols

Authentication of the Hash-Lock Protocol

Authentication: Hash-Lock

Theorem

Assuming that the hash function is EUF-CMA[2], the Hash-Lock protocol provides **authentication**, i.e. for any identity $a \in \mathcal{I}$, for any $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$, $\mathtt{tr}_1 : \mathsf{c}_{\mathtt{j}}^{\mathtt{R}_1}$ and $\mathtt{tr}_3 : \mathsf{c}_{\mathtt{j}}^{\mathtt{R}_2}$ s.t.:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$$

the following formula is valid:

$$\mathsf{accept}^{\mathsf{A}} @ \mathtt{tr}_3 \dot{\rightarrow} \dot{\bigvee_{\substack{\mathtt{tr}_2 : \mathsf{c}^{\mathtt{T}}_{\mathtt{A},\mathtt{i}} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}}} \begin{array}{l} \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2 \ \dot{\wedge} \\ \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3 \end{array} \sim \mathsf{true}$$

[2]Taking $\mathsf{verify}_{\mathsf{k}}(s, m) \stackrel{\mathsf{def}}{=} s \doteq \mathsf{H}(m, \mathsf{k})$.

Authentication: Hash-Lock

Proof. Let $a \in \mathcal{I}$, and let $\mathtt{tr} \in \mathcal{T}_{io}$, $\mathtt{tr}_1 : c_j^{R_1}$ and $\mathtt{tr}_3 : c_j^{R_2}$ be s.t.:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$$

We let:

$$t_{conc} \stackrel{\text{def}}{=} \dot{\bigvee_{\substack{\mathtt{tr}_2 : c^T_{\mathbf{A},i} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}}} \mathrm{out}@\mathtt{tr}_1 \doteq \mathrm{in}@\mathtt{tr}_2 \mathbin{\dot\wedge} \mathrm{out}@\mathtt{tr}_2 \doteq \mathrm{in}@\mathtt{tr}_3$$

We must prove that the following reachability judgement is valid:

$$\mathrm{accept}^A@\mathtt{tr}_3 \vdash t_{conc}$$

i.e. that:

$$\pi_2(\mathrm{in}@\mathtt{tr}_3) \doteq \mathrm{H}(\langle n_{R,j}\ ,\ \pi_1(\mathrm{in}@\mathtt{tr}_3)\rangle, k_A) \vdash t_{conc}$$

Authentication: Hash-Lock

We use the EUF-MAC rule on the equality:

$$\pi_2(\mathsf{in}@\mathtt{tr}_3) \doteq \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathsf{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle, \mathsf{k}_\mathsf{A}) \qquad (\dagger)$$

The terms above are ground, and the key k_A is correctly used in them. Moreover, the set of *honest* hashes using key k_A appearing in (†), excluding the top-level hash, is:

$$\begin{aligned} & \mathbb{S}_{\mathsf{mac}}(\pi_2(\mathsf{in}@\mathtt{tr}_3), \langle \mathsf{n}_{\mathsf{R},\mathsf{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle) \\ = {} & \mathbb{S}_{\mathsf{mac}}(\mathsf{in}@\mathtt{tr}_3) \\ = {} & \left\{ \mathsf{H}(\langle \mathsf{in}@\mathtt{tr}_2\,,\, \mathsf{n}_{\mathsf{T},\mathtt{i}}\rangle, \mathsf{k}_\mathsf{A}) \mid \mathtt{tr}_2 : \mathtt{c}^{\mathtt{T}}_{\mathtt{A},\mathtt{i}} < \mathtt{tr}_3 \right\} \end{aligned}$$

The hashes in the reader's outputs can be seen as verify checks, and can therefore be ignored.

Authentication: Hash-Lock

Hence using EUF-MAC plus some basic reasoning, we have:

$$\dfrac{\dfrac{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3, \begin{array}{c}\langle \mathsf{in}@\mathtt{tr}_2\,,\, \mathsf{n}_{\mathsf{T},\mathtt{i}}\rangle \doteq \\ \langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle\end{array} \vdash t_{\mathsf{conc}} \qquad \text{for every } \mathtt{tr}_2 : \mathtt{c}^{\mathtt{T}}_{\mathtt{A},\mathtt{i}} < \mathtt{tr}_3}{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3, \dot{\bigvee}_{\mathtt{tr}_2 : \mathtt{c}^{\mathtt{T}}_{\mathtt{A},\mathtt{i}} < \mathtt{tr}_3} \begin{array}{c}\langle \mathsf{in}@\mathtt{tr}_2\,,\, \mathsf{n}_{\mathsf{T},\mathtt{i}}\rangle \doteq \\ \langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle\end{array} \vdash t_{\mathsf{conc}}}}{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3 \vdash t_{\mathsf{conc}}}$$

Authentication: Hash-Lock

Assuming that the pair and projections satisfy:

$$\frac{}{(\pi_1\langle x\,,\,y\rangle \doteq x) \sim \mathsf{true}} \qquad\qquad \frac{}{(\pi_2\langle x\,,\,y\rangle \doteq y) \sim \mathsf{true}}$$

We only have to show that for every $\mathtt{tr}_2 : \mathsf{c}^{\mathtt{T}}_{\mathtt{A,i}} < \mathtt{tr}_3$:

$$\Gamma \vdash t_{\mathsf{conc}}$$

is valid, where:

$$\Gamma \stackrel{\mathsf{def}}{=} \mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3,\ \mathsf{in}@\mathtt{tr}_2 \doteq \mathsf{n}_{\mathsf{R},\mathtt{j}},\ \mathsf{n}_{\mathsf{T},\mathtt{i}} \doteq \pi_1(\mathsf{in}@\mathtt{tr}_3)$$

Authentication: Hash-Lock

Since $\mathtt{tr}_1 : \mathsf{c}_{\mathtt{j}}^{\mathtt{R}_1} < \mathtt{tr}_3$ we know that:

$$\mathtt{out@tr}_1 \stackrel{\mathsf{def}}{=} \mathsf{n}_{\mathsf{R},\mathtt{j}}$$

Moreover:

$$\mathtt{out@tr}_2 \stackrel{\mathsf{def}}{=} \langle \mathsf{n}_{\mathsf{T},\mathtt{i}} \,, \mathsf{H}(\langle \mathsf{in@}\mathtt{tr}_2 \,, \mathsf{n}_{\mathsf{T},\mathtt{i}} \rangle, \mathsf{k}_{\mathsf{A}}) \rangle$$

Hence:

$$\Gamma \vdash \pi_1(\mathtt{out@tr}_2) \doteq \pi_1(\mathsf{in@}\mathtt{tr}_3) \qquad (\diamond)$$

Similarly:

$$\begin{aligned} \Gamma \vdash \pi_2(\mathtt{out@tr}_2) &\doteq \mathsf{H}(\langle \mathsf{in@}\mathtt{tr}_2 \,, \mathsf{n}_{\mathsf{T},\mathtt{i}} \rangle, \mathsf{k}_{\mathsf{A}}) \\ &\doteq \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}} \,, \pi_1(\mathsf{in@}\mathtt{tr}_3) \rangle, \mathsf{k}_{\mathsf{A}}) \\ &\doteq \pi_2(\mathsf{in@}\mathtt{tr}_3) \end{aligned}$$

Consequently:

$$\Gamma \vdash \pi_2(\mathtt{out@tr}_2) \doteq \pi_2(\mathsf{in@}\mathtt{tr}_3) \qquad (\star)$$

Authentication: Hash-Lock

Assuming that the pair and projections satisfy the property:

$$\frac{}{\pi_1\ x \doteq \pi_1\ y \mathrel{\dot{\rightarrow}} \pi_2\ x \doteq \pi_2\ y \mathrel{\dot{\rightarrow}} x \doteq y}$$

We deduce from ($\star$) and ($\diamond$) that:

$$\Gamma \vdash \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3$$

Putting everything together, we get:

$$\Gamma \vdash \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2 \mathrel{\dot{\wedge}} \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3 \qquad (\ddagger)$$

Authentication: Hash-Lock

Recall that:

$$t_{\mathsf{conc}} \stackrel{\mathsf{def}}{=} \dot{\bigvee}_{\substack{\mathtt{tr_2}:\mathtt{c}^{\mathtt{T}}_{\mathbf{A},\mathtt{i}} \\ \mathtt{tr_1} \leq \mathtt{tr_2} \leq \mathtt{tr_3}}} \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2 \mathrel{\dot\wedge} \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3$$

and we must show that $\Gamma \vdash t_{\mathsf{conc}}$. Hence, using ($\ddagger$), it only remains to prove that whenever $\mathtt{tr}_2 < \mathtt{tr}_1$, we have:

$$\Gamma,\ \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2,\ \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3 \vdash \bot$$

This follows from the independence rule:

$$\frac{}{(t \doteq \mathsf{n}) = \mathsf{false}}\ {=}\text{-IND} \qquad \text{when } t \text{ is ground and } \mathsf{n} \not\in \mathsf{st}(t)$$

using the fact that:

$$\mathsf{out}@\mathtt{tr}_1 \stackrel{\mathsf{def}}{=} \mathsf{n}_{\mathsf{R},\mathsf{j}}$$

and that if $\mathtt{tr}_2 < \mathtt{tr}_1$ then $\mathsf{n}_{\mathsf{R},\mathsf{j}} \not\in \mathsf{st}(\mathsf{in}@\mathtt{tr}_2)$.

Authentication Protocols

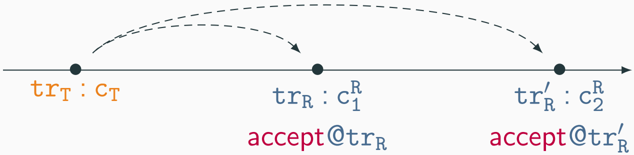
Beyond Authentication

Beyond Authentication

Authentication, which states that we must have:

$\forall \mathtt{tr}_\mathtt{R} : \mathtt{c}_\mathtt{R}.\ \exists \mathtt{tr}_\mathtt{T} : \mathtt{c}_\mathtt{T}.$

does not exclude the scenario:



Replay Attack

This is a **replay attack**: the **same message** (or partial transcript), when replayed, is **accepted again** by the server.

This can yield real-word **attacks**. E.g. an adversary can open a door at will once it eavesdropped one honest interaction.

Example

The following protocol, called Basic Hash, suffer from such attacks:

$$\begin{aligned}
\mathsf{T}(\mathsf{A},\mathtt{i}) &: \nu\, \mathsf{n}_{\mathsf{T},\mathtt{i}}.\ \mathbf{out}(\mathsf{c}^{\mathtt{T}}_{\mathsf{A},\mathtt{i}}, \langle \mathsf{n}_{\mathsf{T},\mathtt{i}}\,,\ \mathsf{H}(\mathsf{n}_{\mathsf{T},\mathtt{i}}, \mathsf{k}_{\mathsf{A}})\rangle) \\
\mathsf{R}(\mathtt{j}) &: \mathbf{in}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{y}).\ \text{if } \dot{\bigvee}_{\mathsf{A}\in\mathcal{I}} \pi_2(\mathsf{y}) \doteq \mathsf{H}(\pi_1(\mathsf{y}), \mathsf{k}_{\mathsf{A}}) \\
&\qquad \text{then } \mathbf{out}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{ok}) \\
&\qquad \text{else } \mathbf{out}(\mathsf{c}^{\mathtt{R_2}}_{\mathtt{j}}, \mathsf{ko})
\end{aligned}$$

Injective Authentication

The **authentication** property is too *weak* for many real-world application.

To prevent replay attacks, we require that the protocol provides a **stronger** property, **injective authentication**.

Injective Authentication: Hash-Lock

The following formulas encode the fact that the **Hash-Lock** protocol provides **injective authentication**:

$\forall A \in \mathcal{I}.\ \forall \mathtt{tr} \in \mathcal{T}_{io}.\ \forall \mathtt{tr}_1 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_1}, \mathtt{tr}_3 : \mathtt{c}_{\mathtt{j}}^{\mathtt{R}_2}$ s.t. $\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$

$$\mathsf{accept}^{A} @ \mathtt{tr}_3 \dot{\rightarrow} \dot{\bigvee_{\substack{\mathtt{tr}_2 : \mathtt{c}^{\mathtt{T}}_{\mathbf{A},\mathtt{i}} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}}} \begin{array}{l} \mathsf{out}@\mathtt{tr}_1 \doteq \mathsf{in}@\mathtt{tr}_2\ \dot{\wedge} \\ \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}_3 \end{array}$$

$$\dot{\wedge} \dot{\bigwedge_{\substack{\mathtt{tr}'_1 : \mathtt{c}_{\mathtt{k}}^{\mathtt{R}_1},\ \mathtt{tr}'_3 : \mathtt{c}_{\mathtt{k}}^{\mathtt{R}_2} \\ \mathtt{tr}'_1 < \mathtt{tr}'_3 \leq \mathtt{tr}}}} \left(\begin{array}{l} \mathsf{accept}^{A} @ \mathtt{tr}'_3\ \dot{\wedge} \\ \mathsf{out}@\mathtt{tr}_2 \doteq \mathsf{in}@\mathtt{tr}'_3 \end{array} \dot{\rightarrow} j = k \right)$$

References i

[1] D. Baelde, S. Delaune, and L. Hirschi.
Partial order reduction for security protocols.
In *CONCUR*, volume 42 of *LIPIcs*, pages 497–510. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.