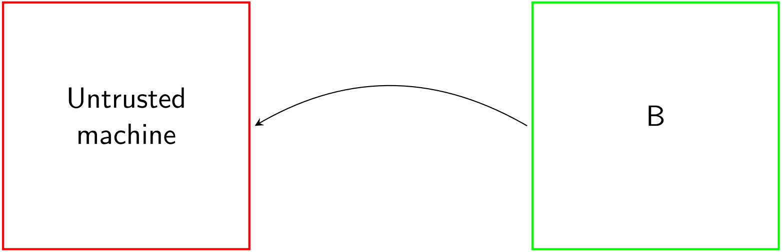
$\pi_{\mathbf{RA}}$: A π-calculus for verifying protocols that use remote attestation

Emiel Lanckriet[1] Matteo Busi [2] Dominique Devriese [1]

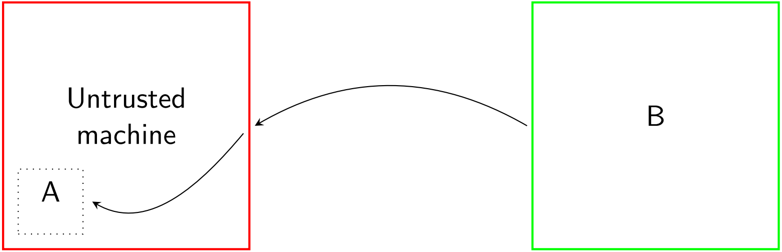
[1]KULeuven [2]Ca' Foscari University of Venice

July 6, 2023

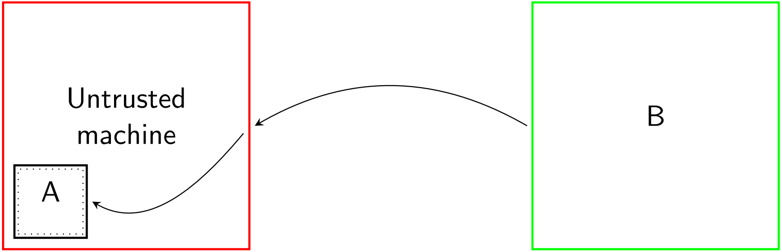
Motivation



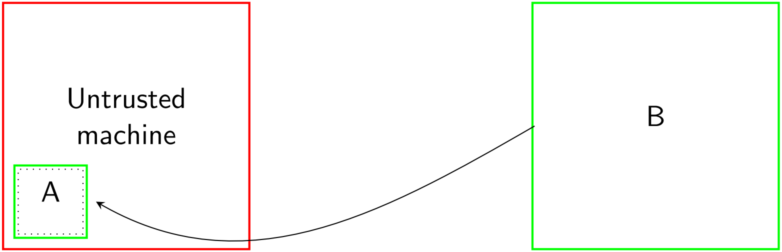
Motivation



Motivation



Motivation



Motivation

- Several implementations of RA: Intel SGX, MIT Sanctum, Sancus, TPM, etc.

Motivation

- Several implementations of RA: Intel SGX, MIT Sanctum, Sancus, TPM, etc.
- Goal: Reason about RA at a high level.
  Ignoring:
  - Implementation RA
  - Isolation primitives
  - Communication primitives

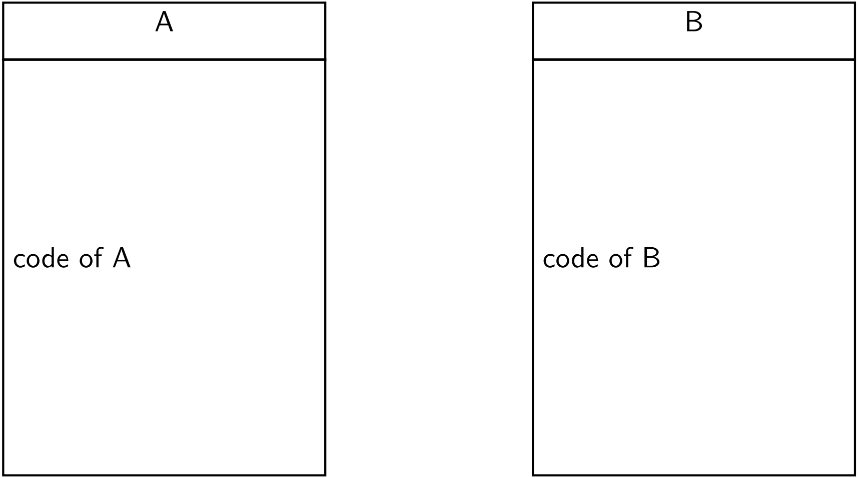
Contributions

- $\pi_{\mathbf{RA}}$: High level abstract model of remote attestation in applied π-calculus
- Application: Proving security of MAGE (solution for mutual authentication) using $\pi_{\mathbf{RA}}$

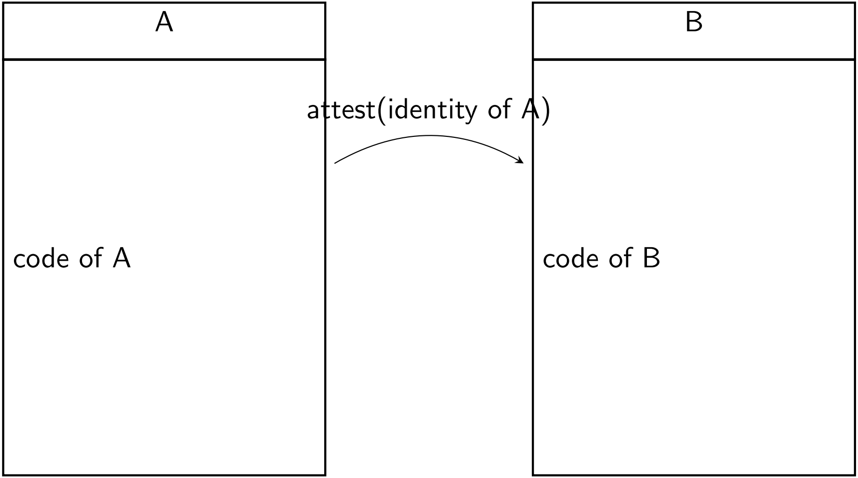
Contributions

- $\pi_{\mathbf{RA}}$: High level abstract model of remote attestation in applied π-calculus
- Application: Proving security of MAGE (solution for mutual authentication) using $\pi_{\mathbf{RA}}$

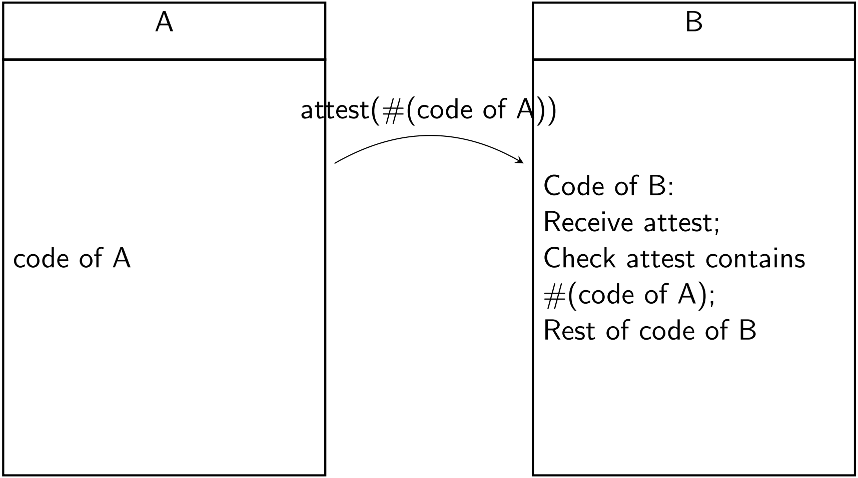
Remote attestation



Remote attestation



Remote attestation



Syntax of π-calculus

Example

$$\overline{N}\langle 42\rangle.\mathbf{0} \mid N(y).print(y).\mathbf{0} \rightarrow \mathbf{0} \mid print(42).\mathbf{0} \xrightarrow{42} \mathbf{0} \mid \mathbf{0}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$getAttest(\mathbf{x}).\mathbf{P}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\begin{array}{ll} & \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\ \rightarrow & \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \end{array}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P}$$
$$\rightarrow \mathbf{Q} \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P}$$
$$\rightarrow Q \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$
\begin{aligned}
&\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\rightarrow Q \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \\
&= \mathbf{N}^{\mathbf{auth}}(y, \#P, \mathit{anon}).\mathit{print}(y).\mathbf{0} \mid
\end{aligned}
$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P}$$

$$\rightarrow \mathbf{Q} \mid P\left\{{}^{\mathit{attest}(\#P)}/_{x}\right\}$$

$$= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#\mathbf{P}, \mathit{anon}).\mathit{print}(\mathbf{y}).\mathbf{0} \mid$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\begin{aligned}
&\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\rightarrow \mathbf{Q} \mid \boxed{P\left\{{}^{\mathit{attest}(\#P)}/_{x}\right\}} \\
&= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#\mathbf{P}, \mathit{anon}).\mathit{print}(\mathbf{y}).\mathbf{0} \mid \\
&\quad \boxed{\left(\overline{\mathbf{N}}^{\mathbf{auth}}\langle 42, x, \mathit{any}\rangle.\mathbf{0}\right)\left\{{}^{\mathit{attest}(\#P)}/_{x}\right\}}
\end{aligned}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\begin{aligned}
&\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\rightarrow \mathbf{Q} \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \\
&= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#\mathbf{P}, \mathit{anon}).\mathit{print}(\mathbf{y}).\mathbf{0} \mid \\
&\qquad \overline{\mathbf{N}}^{\mathbf{auth}}\langle 42, \mathit{attest}(\#P), \mathit{any}\rangle.\mathbf{0}
\end{aligned}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$\begin{aligned}
&\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\to \mathbf{Q} \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \\
&= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#P, \mathit{anon}).\mathit{print}(\mathbf{y}).\mathbf{0} \mid \\
&\qquad \overline{\mathbf{N}}^{\mathbf{auth}}\langle \mathbf{42}, \mathit{attest}(\#P), \mathit{any}\rangle.\mathbf{0}
\end{aligned}$$

Example program in $\pi_{\mathbf{RA}}$

Example

$$
\begin{aligned}
&\mathbf{Q} \mid \mathit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\to \mathbf{Q} \mid \mathbf{P}\left\{{}^{\mathit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \\
&= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#P, \mathit{anon}).\mathit{print}(\mathbf{y}).\mathbf{0} \mid \\
&\quad \overline{\mathbf{N}}^{\mathbf{auth}}\langle \mathbf{42}, \mathit{attest}(\#P), \mathit{any}\rangle.\mathbf{0} \\
&\to \mathit{print}(\mathbf{42}).\mathbf{0} \mid \mathbf{0}
\end{aligned}
$$

Example program in $\pi_{\mathbf{RA}}$

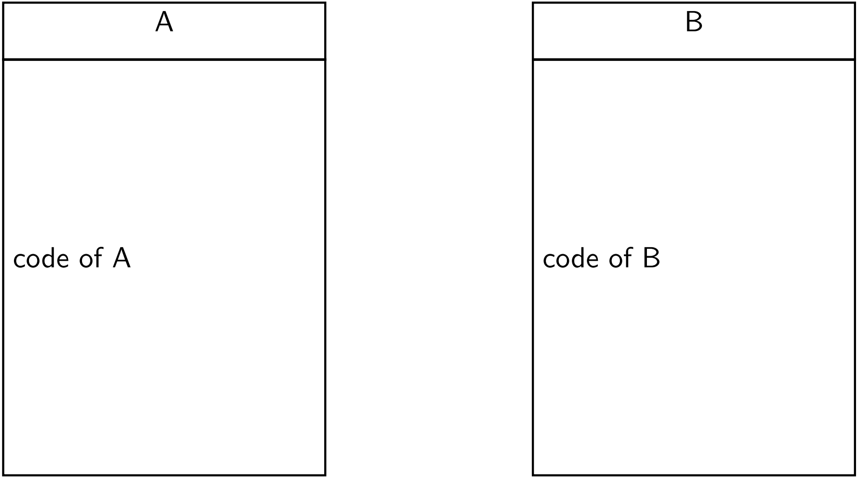
Example

$$
\begin{aligned}
&\mathbf{Q} \mid \textit{getAttest}(\mathbf{x}).\mathbf{P} \\
&\rightarrow \mathbf{Q} \mid \mathbf{P}\left\{{}^{\textit{attest}(\#\mathbf{P})}/_{\mathbf{x}}\right\} \\
&= \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \#\mathbf{P}, \textit{anon}).\textit{print}(\mathbf{y}).\mathbf{0} \mid \\
&\qquad \overline{\mathbf{N}}^{\mathbf{auth}}\langle \mathbf{42}, \textit{attest}(\ \#\mathbf{P}\ ), \textit{any}\rangle.\mathbf{0} \\
&\rightarrow \textit{print}(\mathbf{42}).\mathbf{0} \mid \mathbf{0} \\
&\xrightarrow{\mathbf{42}} \mathbf{0}.
\end{aligned}
$$

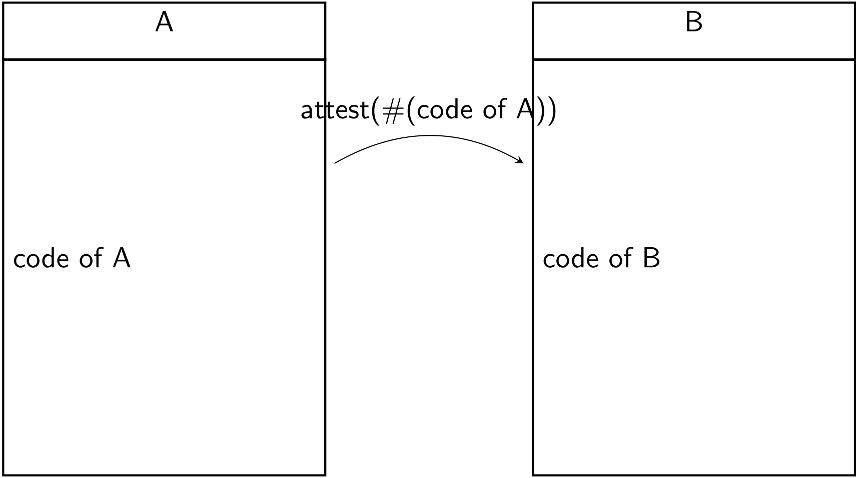
Contributions

- $\pi_{\mathbf{RA}}$: High level abstract model of remote attestation in applied π-calculus
- Application: Proving security of MAGE (solution for mutual authentication) using $\pi_{\mathbf{RA}}$

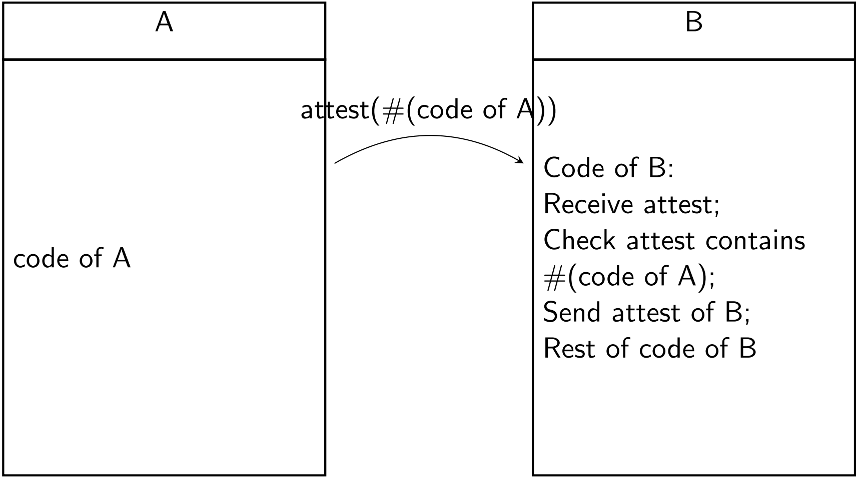
Mutual authentication



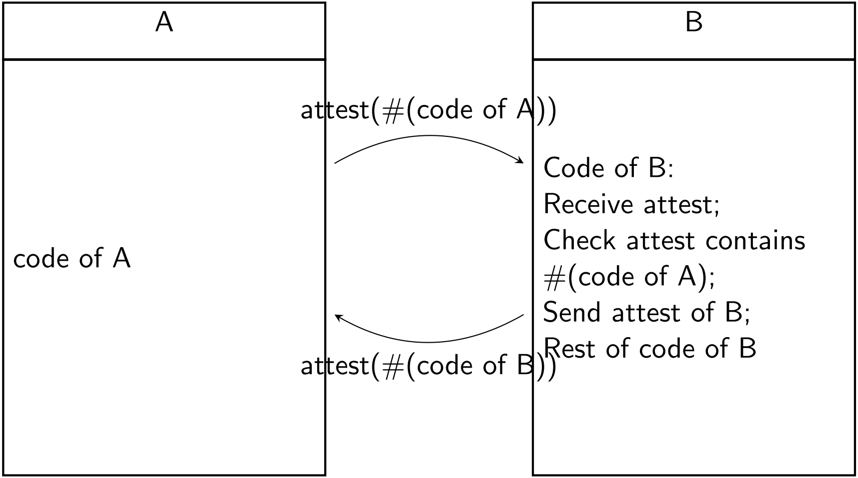
Mutual authentication



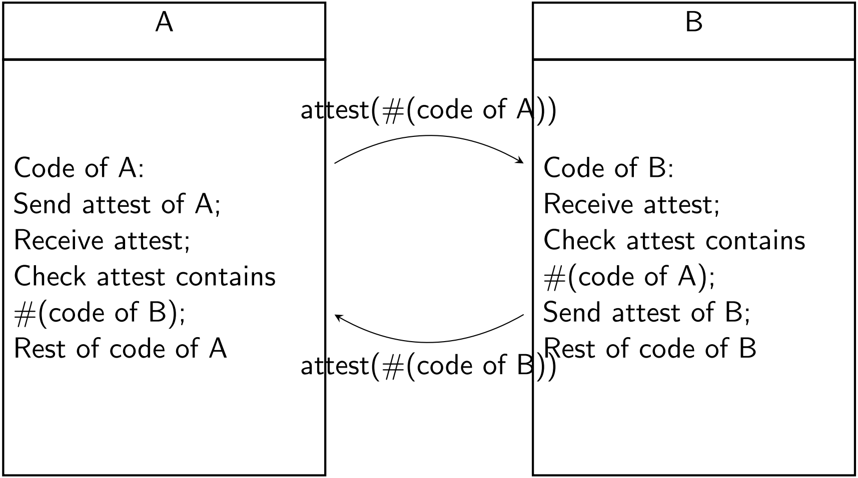
Mutual authentication



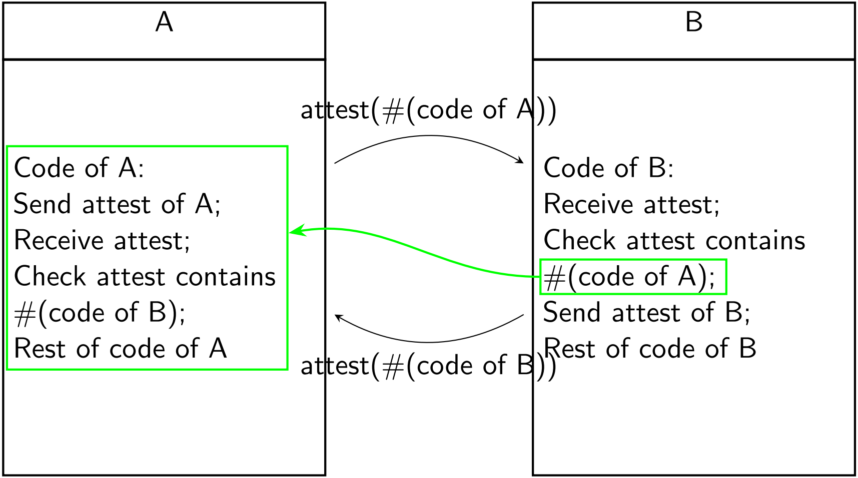
Mutual authentication



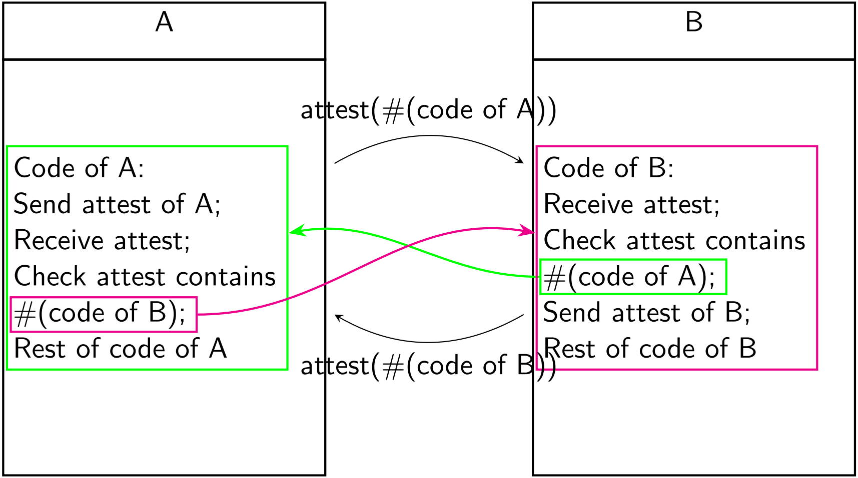
Mutual authentication



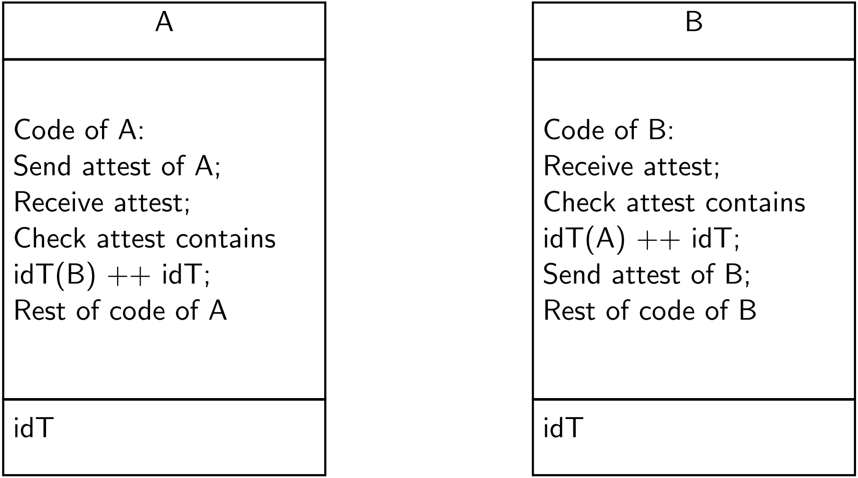
Mutual authentication



Mutual authentication

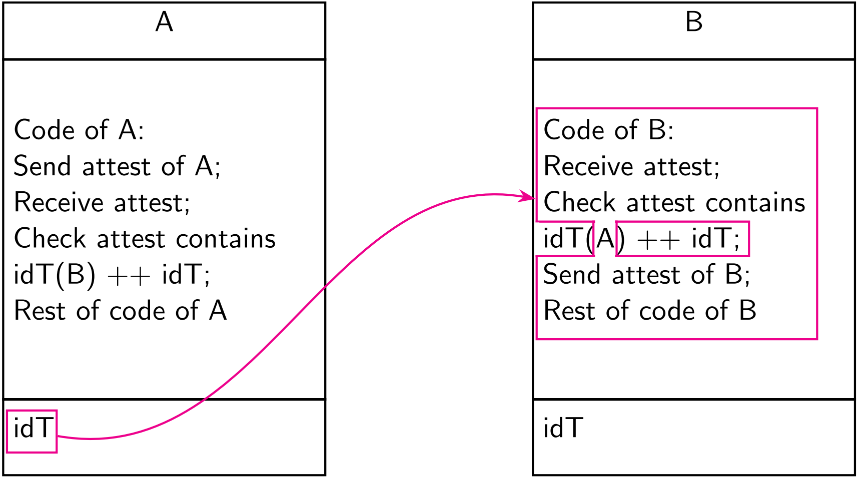


MAGE



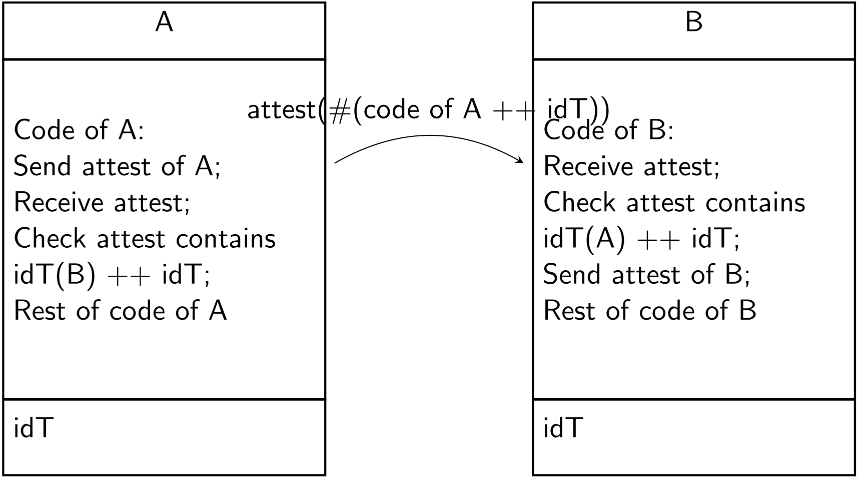
G. Chen and Y. Zhang. MAGE: Mutual attestation for a group of enclaves without trusted third parties. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association

MAGE



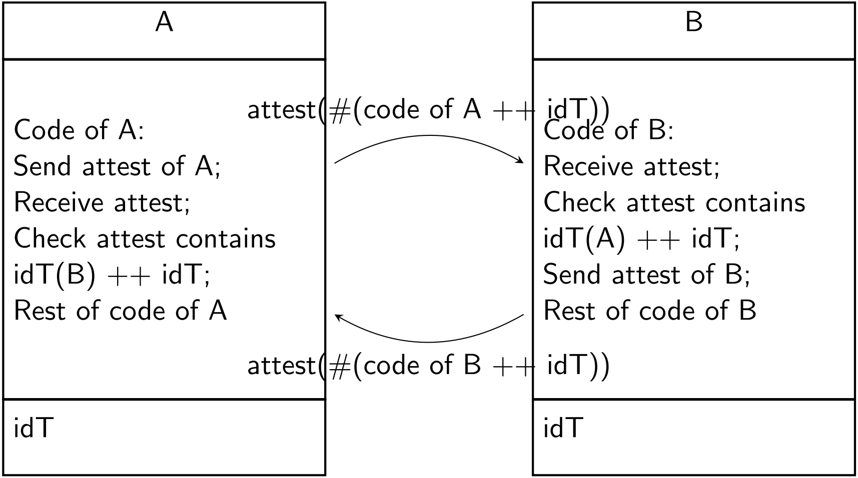
G. Chen and Y. Zhang. MAGE: Mutual attestation for a group of enclaves without trusted third parties. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association

MAGE



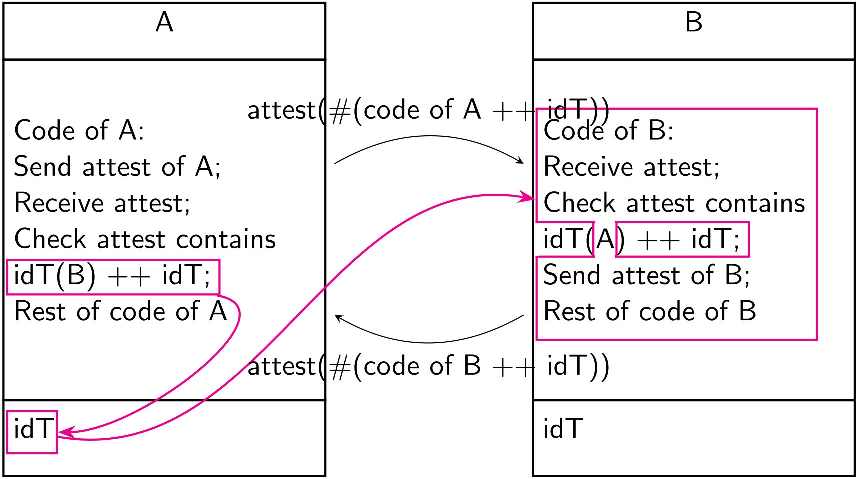
G. Chen and Y. Zhang. MAGE: Mutual attestation for a group of enclaves without trusted third parties. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association

MAGE



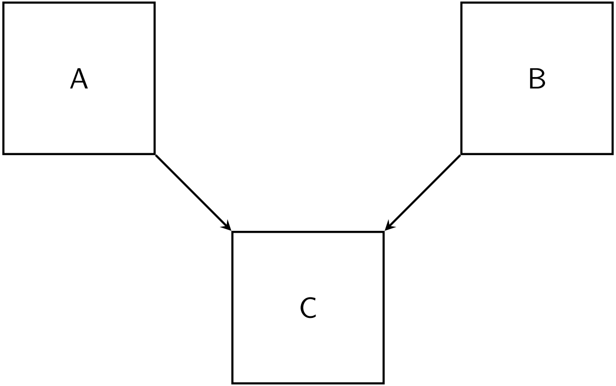
G. Chen and Y. Zhang. MAGE: Mutual attestation for a group of enclaves without trusted third parties. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association

MAGE

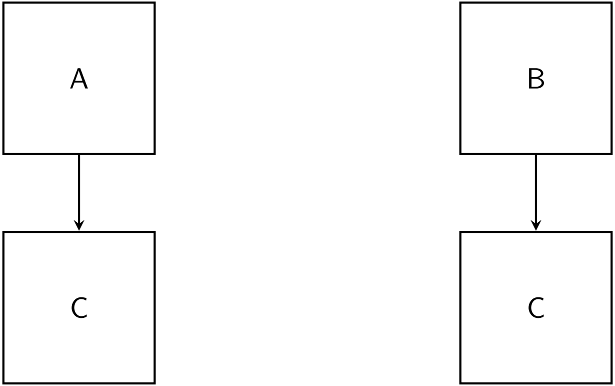


G. Chen and Y. Zhang. MAGE: Mutual attestation for a group of enclaves without trusted third parties. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association

No way to distinguish exact copies



No way to distinguish exact copies



Proof of security of MAGE

MAGE

Proof of security of MAGE

MAGE

$\pi_{\mathbf{RA}}$

Proof of security of MAGE

Secure compilation

Definition (Fully abstract compilation)

$$\mathsf{P} \simeq_{\mathsf{ctx}} \mathsf{Q} \iff [\![\mathsf{P}]\!] \simeq_{\mathbf{ctx}} [\![\mathsf{Q}]\!].$$

Definition (Contextual equivalence)

$$P \simeq_{ctx} Q \iff \forall C : behav(C[P]) = behav(C[Q])$$

Secure compilation

Definition (RrHC)

$$\forall \mathbf{C_T} : \exists \mathsf{C_S} : \forall \mathsf{P} : \mathit{behav}(\mathbf{C_T}[\llbracket \mathsf{P} \rrbracket]) = \mathit{behav}(\mathsf{C_S}[\mathsf{P}])$$

π_{Actor}

$$\boxed{n : \mathsf{N}_{\mathsf{m} \to \mathit{self}}(\mathsf{y}).\mathit{print}(y)} \mid \boxed{m : \overline{\mathsf{N}}_{\mathit{self} \to \mathsf{n}}\langle 42 \rangle.\mathbf{0}}$$

π_{Actor}

$$\boxed{n : \mathsf{N}_{\mathsf{m}\rightarrow \textit{self}}(\mathsf{y}).\textit{print}(y)} \mid \boxed{m : \overline{\mathsf{N}}_{\textit{self}\rightarrow\mathsf{n}}\langle 42\rangle.\mathbf{0}}$$

πActor

$$\boxed{n : \mathsf{N}_{\mathsf{m}\rightarrow self}(\mathsf{y}).print(y)} \mid \boxed{m : \overline{\mathsf{N}}_{self\rightarrow \mathsf{n}}\langle 42\rangle.\mathbf{0}}$$

π_{Actor}

$$\boxed{n : \mathsf{N}_{\mathsf{m}\rightarrow n}(\mathsf{y}).print(y)} \mid \boxed{m : \overline{\mathsf{N}}_{m\rightarrow \mathsf{n}}\langle 42\rangle.\mathbf{0}}$$

π_{Actor}

$$\boxed{n : \mathsf{N}_{\mathsf{m}\to\mathsf{n}}(\mathsf{y}).print(y)} \mid \boxed{m : \overline{\mathsf{N}}_{\mathsf{m}\to\mathsf{n}}\langle 42\rangle.\mathbf{0}}$$

πActor

$$\boxed{n : \mathsf{N}_{\mathsf{m}\to\mathsf{n}}(\mathsf{y}).\mathit{print}(y)} \mid \boxed{m : \overline{\mathsf{N}}_{\mathsf{m}\to\mathsf{n}}\langle 42\rangle.\mathbf{0}}$$

$$\to \boxed{n : \mathit{print}(42).\mathbf{0}} \mid \boxed{m : \mathbf{0}}$$

πActor

$$\boxed{n : \mathsf{N}_{\mathsf{m}\to\mathsf{n}}(\mathsf{y}).\mathit{print}(y)} \mid \boxed{m : \overline{\mathsf{N}}_{\mathsf{m}\to\mathsf{n}}\langle 42\rangle.\mathbf{0}}$$
$$\to \boxed{n : \mathit{print}(42).\mathbf{0}} \mid \boxed{m : \mathbf{0}}$$
$$\xrightarrow{42} \boxed{n : \mathbf{0}} \mid \boxed{m : \mathbf{0}}.$$

Proof of security of MAGE

Precompiler (simplified)

$$[\![\mathit{print}(\mathsf{M}).\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathit{print}(\mathbf{M}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$\left[\!\!\left[\boxed{n : P} \right]\!\!\right]_{\mathbf{idT},\mathbf{x}} = \mathit{getAttest}(\mathbf{y}, \mathbf{d}, \mathbf{unit}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{y}}$$

with **y** and **d** fresh

$$[\![\overline{\mathsf{N}}_{\mathit{self} \to \mathit{id}} \langle \mathsf{M} \rangle.\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \overline{\mathbf{N}}^{\mathbf{auth}} \langle \mathbf{M}, \mathbf{x}, \mathit{ext}(\mathbf{idT}(\mathit{id}), \mathbf{idT}) \rangle.[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$[\![\mathsf{N}_{\mathit{id} \to \mathit{self}}(\mathsf{y}).\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \mathit{ext}(\mathbf{idT}(\mathit{id}), \mathbf{idT}), \mathbf{x}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

Precompiler (simplified)

$$[\![\mathit{print}(\mathrm{M}).\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathit{print}(\mathbf{M}).[\![\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$\left[\!\!\left[\boxed{n : P} \right]\!\!\right]_{\mathbf{idT},\mathbf{x}} = \mathit{getAttest}(\mathbf{y}, \mathbf{d}, \mathbf{unit}).[\![\mathrm{P}]\!]_{\mathbf{idT},\mathbf{y}}$$

with **y** and **d** fresh

$$\left[\!\!\left[\overline{\mathrm{N}}_{\mathit{self} \to \mathit{id}} \langle \mathrm{M} \rangle.\mathrm{P} \right]\!\!\right]_{\mathbf{idT}, x} = \overline{\mathbf{N}}^{\mathbf{auth}} \langle \mathbf{M},\ x,\ \mathit{ext}(\mathbf{idT}(\mathit{id}), \mathbf{idT}) \rangle.[\![\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$[\![\mathrm{N}_{\mathit{id} \to \mathit{self}}(\mathrm{y}).\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathbf{N}^{\mathbf{auth}}(\mathbf{y},\ \mathit{ext}(\mathbf{idT}(\mathit{id}), \mathbf{idT}),\ \mathbf{x}).[\![\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

Precompiler (simplified)

$$[\![\mathit{print}(\mathsf{M}).\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathit{print}(\mathbf{M}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$\left[\!\!\left[\boxed{n : P} \right]\!\!\right]_{\mathbf{idT},\mathbf{x}} = \mathit{getAttest}(\mathbf{y}, \mathbf{d}, \mathbf{unit}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{y}}$$

with **y** and **d** fresh

$$[\![\overline{\mathsf{N}}_{\mathit{self} \to \mathit{id}} \langle \mathsf{M} \rangle.\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \overline{\mathbf{N}}^{\mathbf{auth}} \langle \mathbf{M}, \mathbf{x}, \boxed{\mathit{ext}(\mathit{idT}(\mathit{id}), \mathit{idT})} \rangle.[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

$$[\![\mathsf{N}_{\mathit{id} \to \mathit{self}}(\mathsf{y}).\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}} = \mathbf{N}^{\mathbf{auth}}(\mathbf{y}, \mathit{ext}(\mathbf{idT}(\mathit{id}), \mathbf{idT}), \mathbf{x}).[\![\mathsf{P}]\!]_{\mathbf{idT},\mathbf{x}}$$

Table of identities

Function to calculate table of preprocessed hashes for each actor based on precompiler:

$$cH(\boxed{n_1 : P_1} \mid \boxed{n_2 : P_2} \mid \mathsf{P}_3) = \{\mathsf{n}_1 : \#_{z,x}[\![\mathsf{P}_1]\!]_{z,x}, \mathsf{n}_2 : \#_{z,x}[\![\mathsf{P}_2]\!]_{z,x}\}$$

Compiler

$$[\![\mathsf{P}]\!] = [\![\mathsf{P}]\!]_{cH(\mathsf{P}),\blacksquare}$$

Proof of security of MAGE

Secure compilation

Definition (Adapted version of fully abstract compilation)

If P and Q are source programs
with the **same actors** ($cH(\mathsf{P}) = cH(\mathsf{Q})$), then

$$\mathsf{P} \simeq_{\mathsf{ctx}} \mathsf{Q} \iff [\![\mathsf{P}]\!] \simeq_{\mathbf{ctx}} [\![\mathsf{Q}]\!].$$

Secure compilation

Definition (Adapted version of fully abstract compilation)

If P and Q are source programs
with the **same actors** $(cH(\mathsf{P}) = cH(\mathsf{Q}))$, then

$$\mathsf{P} \simeq_{\mathsf{ctx}} \mathsf{Q} \iff [\![\mathsf{P}]\!] \simeq_{\mathbf{ctx}} [\![\mathsf{Q}]\!].$$

Adapted RrHC (simplified)

Theorem (Adapted RrHC)

$$\forall \mathbf{idT}, \mathbf{C_T} : \exists \mathrm{C_S} : \forall \mathrm{P} : \mathbf{idT} = cH(\mathrm{P})$$
$$\implies \tau(behav(\mathbf{C_T}[\![\mathrm{P}]\!])) = behav(\mathrm{C_S}[\mathrm{P}]).$$

Adapted RrHC (simplified)

Theorem (Adapted RrHC)

$$\forall \mathbf{idT}, \mathbf{C_T} : \exists \mathsf{C_S} : \forall \mathsf{P} : \mathbf{idT} = cH(\mathsf{P}) \implies \tau(behav(\mathbf{C_T}[\![\mathsf{P}]\!])) = behav(\mathsf{C_S}[\mathsf{P}]).$$

Questions?/Suggestions?

Preprocessed table of identities

$$
cH(\mathsf{P}) = \begin{cases}
\emptyset & \text{if } \mathsf{P} = \mathbf{0} \\
\{(\mathsf{n}, \#_{z,x,d}[\![\mathsf{P}_{start}]\!]_{z,x})\} \cup cH(\mathsf{P}_{start}) & \text{if } \mathsf{P} = \boxed{n : P_{start}} \\
 & \text{ or } \mathsf{P} = \boxed{n : P'}\,\mathsf{P}_{start} \\
cH(\mathsf{Q}) & \text{if } \mathsf{P} = \overline{\mathsf{N}}_{id_1 \to id_2}\langle \mathsf{M} \rangle.\mathsf{Q} \\
 & \text{ or } \mathsf{P} = \mathsf{N}_{id_2 \to id_1}(\mathsf{x}).\mathsf{Q} \text{ or } \ldots \\
cH(\mathsf{Q}) \cup cH(\mathsf{R}) & \text{if } \mathsf{P} = \mathsf{Q} \mid \mathsf{R} \text{ or } \mathsf{P} = \mathsf{Q} + \mathsf{R} \\
 & \quad \text{or } \mathsf{P} = \mathit{if}\ \mathsf{M} = \mathsf{N}\ \mathit{then}\ \mathsf{Q}\ \mathit{else}\ \mathsf{R}
\end{cases} .
$$

Figure: The function $cH(\mathsf{P})$ compiles a table of the hashes of compiled actors in P.

Proof of equivalence reflection

reflection direction $\Uparrow$

$$P_1 \overset{?}{\simeq}_{ctx} P_2$$

$$behav(C[P_1]) \overset{?}{=\!=} behav(C[P_2])$$

$$[\![C[P_1]]\!] = [\![C]\!]_{cH(P_1),\blacksquare}[[\![P_1]\!]] \;\Big\|\; (a) \qquad (c) \;\Big\|\; [\![C[P_2]]\!] = [\![C]\!]_{cH(P_2),\blacksquare}[[\![P_2]\!]]$$

$$behav([\![C]\!]_{cH(P_1),\blacksquare}[[\![P_1]\!]]) \overset{(b)}{=\!=} behav([\![C]\!]_{cH(P_2),\blacksquare}[[\![P_2]\!]])$$

$$[\![P_1]\!] \simeq_{\mathbf{ctx}} [\![P_2]\!]$$

$$cH(P_1) = cH(P_2)$$

Figure: Diagram of the proof of equivalence reflection of FAC. This diagram is adapted from the diagram in [2].

Proof of equivalence preservation

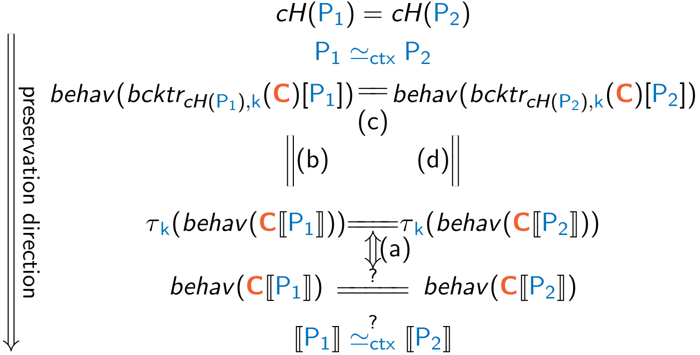


Figure: Diagram of the proof of equivalence preservation of FAC. This diagram is adapted from the diagram in [2].

Back-translation

$$bcktr_{idT,k}(getAttest(\mathbf{x}, \mathbf{d}, \mathbf{D}).\mathbf{P}) =$$

$$\begin{cases} \boxed{n : P} & \text{if } \mathbf{P} = [\![\mathrm{P}]\!]_{\mathbf{idT},\mathbf{x}} \text{with } d \text{ not free in } \mathbf{P} \text{ and } \\ & \#new\ (\mathbf{x}, \mathbf{d})\ \mathbf{P}\ = ext(\mathbf{idT}(\mathrm{n}), \mathbf{idT}) \\ bcktr_{idT,k}(\mathbf{P})\left\{{}^{bT_k(attest(\#new\ (\mathbf{x},\mathbf{d})\ \mathbf{P}\ ))}/_{x}\right\}\left\{{}^{bT_k(\mathbf{D})}/_{d}\right\} & \text{otherwise} \end{cases}$$

$$bcktr_{idT,k}(\overline{\mathbf{N}}^{\mathbf{auth}}\langle \mathbf{M},\ \mathbf{a},\ \mathbf{h}\rangle.\mathbf{P}) =$$

if $\exists \mathrm{n} : \mathrm{bT_k}(ext(\mathbf{idT}(\mathrm{n}), \mathbf{idT})) = \mathrm{bT_k}(\mathbf{h})$

then $\overline{\mathrm{bT_k}(\mathrm{N})}_{anon \to \mathrm{n}}\langle \mathrm{bT_k}(\mathrm{M})\rangle.bcktr_{idT,k}(\mathbf{P})$

else $\overline{MCN_k(\mathrm{bT_k}(\mathbf{N}), \mathrm{bT_k}(\mathbf{a}), pack_k(\mathrm{bT_k}(\mathbf{h})), \mathrm{k})}_{anon \to any}\langle \mathrm{bT_k}(\mathbf{M})\rangle.bcktr_{idT,k}(\mathbf{P}) +$

$\overline{MCN_k(\mathrm{bT_k}(\mathbf{N}), pack_k(any), pack_k(\mathrm{bT_k}(\mathbf{h})), \mathrm{k})}_{anon \to any}\langle \mathrm{bT_k}(\mathbf{M})\rangle.bcktr_{idT,k}(\mathbf{P})$

$$bcktr_{idT,k}(\mathbf{N}^{\mathbf{auth}}(\mathbf{y},\ \mathbf{h},\ \mathbf{a}).\mathbf{P}) =$$

if $\exists \mathrm{n} : \mathrm{bT_k}(ext(\mathbf{idT}(\mathrm{n}), \mathbf{idT})) = \mathrm{bT_k}(\mathbf{h})$

then $\mathrm{bT_k}(\mathrm{N})_{\mathrm{n} \to anon}(\mathrm{y}).bcktr_{idT,k}(\mathbf{P})$

else $MCN_k(\mathrm{bT_k}(\mathbf{N}), pack_k(\mathrm{bT_k}(\mathbf{h})), \mathrm{bT_k}(\mathbf{a}), \mathrm{k})_{anon \to any}(\mathrm{bT_k}(\mathbf{M})).bcktr_{idT,k}(\mathbf{P}) +$

$MCN_k(\mathrm{bT_k}(\mathbf{N}), pack_k(\mathrm{bT_k}(\mathbf{h})), pack_k(any), \mathrm{k})_{anon \to any}(\mathrm{bT_k}(\mathbf{M})).bcktr_{idT,k}(\mathbf{P}).$