Proving physical proximity using symbolic models

Alexandre Debant, Stéphanie Delaune, Cyrille Wielding

Univ Rennes - IRISA - CNRS

2018-04-15

Symbolic verification

Advantages:

- automated proofs
- efficient tools exist: ProVerif, Tamarin, Avispa...
- can express many security properties (authentication, secrecy, untraceability...)

But: cannot express physical proximity !
omniscient and ubiquitous attacker

How can we handle it?

Table of contents

Term algebra

Messages: terms built over a set of names $\mathcal{N}$ and a signature Σ given with an equational theory E and a rewriting system

Example

- Names: $\mathcal{N} = \{a, n, k\}$
- Signature: $\Sigma = \{senc, sdec, pair, proj_1, proj_2, \oplus\}$

$$x \oplus 0 = x \qquad (x \oplus y) \oplus z = x \oplus (y \oplus z)$$
$$x \oplus x = 0 \qquad x \oplus y = y \oplus x$$

$$sdec(senc(x, y), y) \to x \qquad proj_1(pair(x, y)) \to x$$
$$proj_2(pair(x, y)) \to y$$

We have that: $sdec(senc(n \oplus 0), k), k)\downarrow =_E n$

Process algebra

The role of an agent is described by a process following the grammar:

$$
\begin{array}{rcll}
P & := & 0 & \text{null} \\
 & | & \texttt{new}\ n.P & \text{name restriction} \\
 & | & \texttt{let}\ x = u\ \texttt{in}\ P & \text{conditional declaration} \\
 & | & \texttt{out}(u).P & \text{output} \\
 & | & \texttt{in}(x).P & \text{input}
\end{array}
$$

Process algebra

The role of an agent is described by a process following the grammar:

P	$:=$	0	null
	$\mid$	$\texttt{new}\ n.P$	name restriction
	$\mid$	$\texttt{let}\ x = u\ \texttt{in}\ P$	conditional declaration
	$\mid$	$\texttt{out}(u).P$	output
	$\mid$	$\texttt{in}(x).P$	input
	$\mid$	$\texttt{in}^{<t}(x).P$	guarded input
	$\mid$	$\texttt{reset}.P$	personal clock reset

Process algebra

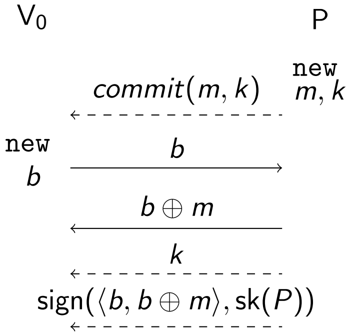
The role of an agent is described by a process following the grammar:

P	$:=$	0	null
	$\mid$	$\mathtt{new}\ n.P$	name restriction
	$\mid$	$\mathtt{let}\ x = u\ \mathtt{in}\ P$	conditional declaration
	$\mid$	$\mathtt{out}(u).P$	output
	$\mid$	$\mathtt{in}(x).P$	input
	$\mid$	$\mathtt{in}^{<t}(x).P$	guarded input
	$\mid$	$\mathtt{reset}.P$	personal clock reset

Protocol

A protocol is a set of roles $(\Pi_1, \cdots, \Pi_k)$ describing the behaviour of each honest agents.

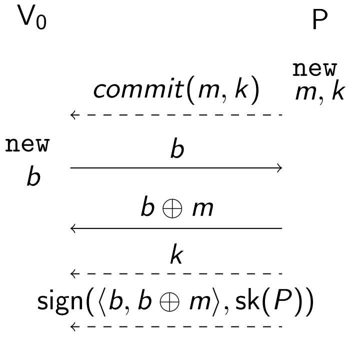
Example: Brands and Chaum - 1993



Brands and Chaum

Example: Brands and Chaum - 1993

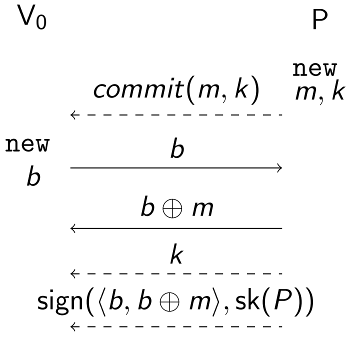
$$
\begin{array}{l}
V(z_V, z_P) := \\
\quad \mathtt{in}(y_c).\mathtt{new}\ b. \\
\quad \mathtt{reset}.\mathtt{out}(b).\mathtt{in}^{<2\times t_0}(y_0). \\
\quad \mathtt{in}(y_k).\mathtt{in}(y_{\mathsf{sign}}). \\
\quad \mathtt{let}\ y_m = \mathsf{open}(y_c, y_k)\ \mathtt{in} \\
\quad \mathtt{let}\ y_{msg} = \mathsf{getmsg}(y_{\mathsf{sign}})\ \mathtt{in} \\
\quad \mathtt{let}\ y_{\mathsf{check}} = \mathsf{check}(y_{\mathsf{sign}}, \mathsf{vk}(z_P))\ \mathtt{in} \\
\quad \mathtt{let}\ y_{\mathsf{eq}} = \mathsf{eq}(\langle b, b \oplus y_m\rangle, y_{msg})\ \mathtt{in} \\
\quad 0
\end{array}
$$



Brands and Chaum

Example: Brands and Chaum - 1993

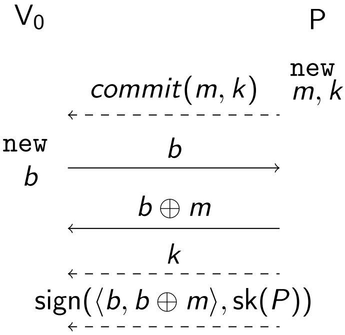
$$
\begin{aligned}
&V(z_V, z_P) := \\
&\quad \texttt{in}(y_c).\texttt{new}\ b. \\
&\quad \color{red}{\texttt{reset}.\texttt{out}(b).\texttt{in}^{<2\times t_0}(y_0)}. \\
&\quad \texttt{in}(y_k).\texttt{in}(y_{\mathsf{sign}}). \\
&\quad \texttt{let}\ y_m = \mathsf{open}(y_c, y_k)\ \texttt{in} \\
&\quad \texttt{let}\ y_{msg} = \mathsf{getmsg}(y_{\mathsf{sign}})\ \texttt{in} \\
&\quad \texttt{let}\ y_{\mathsf{check}} = \mathsf{check}(y_{\mathsf{sign}}, \mathsf{vk}(z_P))\ \texttt{in} \\
&\quad \texttt{let}\ y_{\mathsf{eq}} = \mathsf{eq}(\langle b, b \oplus y_m \rangle, y_{msg})\ \texttt{in} \\
&\quad 0
\end{aligned}
$$



Brands and Chaum

Example: Brands and Chaum - 1993

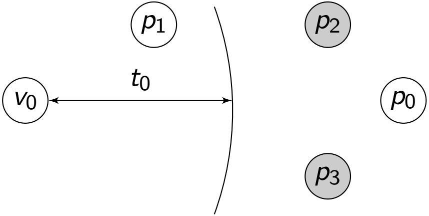
$V(z_V, z_P) :=$
$\quad \mathtt{in}(y_c).\mathtt{new}\ b.$
$\quad \mathtt{reset}.\mathtt{out}(b).\mathtt{in}^{<2\times t_0}(y_0).$
$\quad \mathtt{in}(y_k).\mathtt{in}(y_{\mathsf{sign}}).$
$\quad \mathtt{let}\ y_m = \mathsf{open}(y_c, y_k)\ \mathtt{in}$
$\quad \mathtt{let}\ y_{msg} = \mathsf{getmsg}(y_{\mathsf{sign}})\ \mathtt{in}$
$\quad \mathtt{let}\ y_{\mathsf{check}} = \mathsf{check}(y_{\mathsf{sign}}, \mathsf{vk}(z_P))\ \mathtt{in}$
$\quad \mathtt{let}\ y_{\mathsf{eq}} = \mathsf{eq}(\langle b, b \oplus y_m\rangle, y_{msg})\ \mathtt{in}$
$\quad 0$



Brands and Chaum

Topology

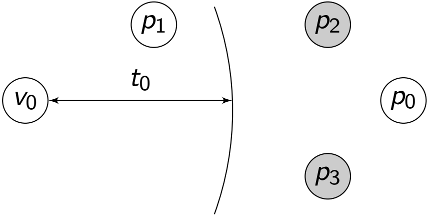
A **topology** is a tuple $\mathcal{T} = (\mathcal{A}, \mathrm{Loc}, \mathcal{M}, v_0, p_0)$.



Topology

A **topology** is a tuple $\mathcal{T} = (\mathcal{A}, \mathrm{Loc}, \mathcal{M}, v_0, p_0)$.

agents

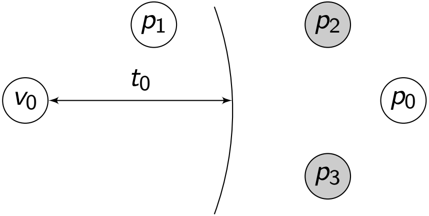


Topology

A **topology** is a tuple $\mathcal{T} = (\mathcal{A}, \mathrm{Loc}, \mathcal{M}, v_0, p_0)$.

agents

locations

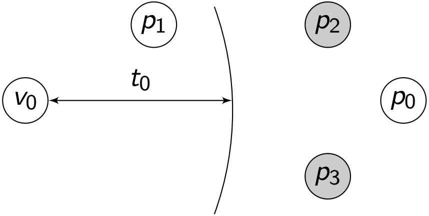


We define $\mathrm{Dist}_{\mathcal{T}}(a, b) = \frac{\|\mathrm{Loc}(a) - \mathrm{Loc}(b)\|}{c}$

Topology

A **topology** is a tuple $\mathcal{T} = (\mathcal{A}, \mathrm{Loc}, \mathcal{M}, v_0, p_0)$.

agents
locations
dishonest agents



We define $\mathrm{Dist}_{\mathcal{T}}(a, b) = \frac{\|\mathrm{Loc}(a) - \mathrm{Loc}(b)\|}{c}$

Topology

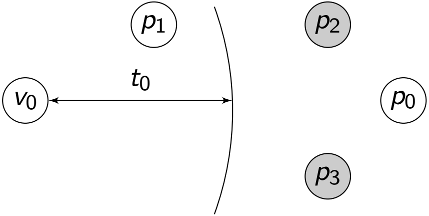
A **topology** is a tuple $\mathcal{T} = (\mathcal{A}, \mathrm{Loc}, \mathcal{M}, v_0, p_0)$.

agents

locations

dishonest agents

specific agents



We define $\mathrm{Dist}_{\mathcal{T}}(a, b) = \frac{\|\mathrm{Loc}(a) - \mathrm{Loc}(b)\|}{c}$

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

OUT $\quad (\lfloor \mathtt{out}(u).P \rfloor_a^{t_a}) \uplus \mathcal{P}; \Phi; t) \xrightarrow{a, \mathtt{out}(u)}_{\mathcal{T}_0} (\lfloor P \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi'; t)$
with $\Phi' = \Phi \cup \{\mathrm{w} \xrightarrow{a,t} u\}$

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

IN $$(\lfloor \mathtt{in}^\star(x).P \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a, \mathtt{in}^\star(u)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto u\} \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t)$$

if u is deducible from Φ

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

IN $\quad (\lfloor \mathtt{in}^\star(x).P \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a, \mathtt{in}^\star(u)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto u\} \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t)$

if $\exists b \in \mathcal{A}, t_b \in \mathcal{R}_+$ such that $t_b \leq t - \mathrm{Dist}_{\mathcal{T}}(b, a)$ and:

- if $b \notin \mathcal{M}$ then $u \in img(\lfloor \Phi \rfloor_b^{t_b})$
- if $b \in \mathcal{M}$ then u is deducible from $\bigcup_{c \in \mathcal{A}} \lfloor \Phi \rfloor_c^{t_b - \mathrm{Dist}_{\mathcal{T}}(c,b)}$

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

IN $\quad (\lfloor \mathtt{in}^\star(x).P \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a, \mathtt{in}^\star(u)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto u\} \rfloor_a^{t_a} \uplus \mathcal{P}; \Phi; t)$

if $\exists b \in \mathcal{A}, t_b \in \mathcal{R}_+$ such that $t_b \leq t - \mathrm{Dist}_{\mathcal{T}}(b, a)$ and:

- if $b \notin \mathcal{M}$ then $u \in img(\lfloor \Phi \rfloor_b^{t_b})$
- if $b \in \mathcal{M}$ then u is deducible from $\bigcup_{c \in \mathcal{A}} \lfloor \Phi \rfloor_c^{t_b - \mathrm{Dist}_{\mathcal{T}}(c,b)}$

Moreover if $\star =< t_g$ then $t_a < t_g$.

Configuration and semantics

A **configuration** is a tuple $(\mathcal{P}; \Phi; t)$ where:

- $\mathcal{P}$ is a multiset of $\lfloor P \rfloor_a^{t_a}$ with $a \in \mathcal{A}$ and $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} m_1, \cdots, \mathrm{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

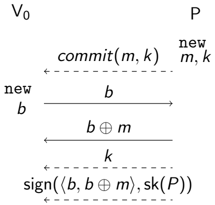
TIME $\quad (\mathcal{P}; \Phi; t) \longrightarrow_{\mathcal{T}} (\mathcal{P}'; \Phi; t')$ with:

- $t' > t$
- $\mathcal{P}' = \{\, \lfloor P \rfloor_a^{t_a + (t' - t)} \mid \lfloor P \rfloor_a^{t_a} \in \mathcal{P} \}$

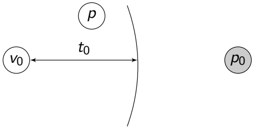
Example: Brands and Chaum - 1993

$$K_0 = (\lfloor V(v_0, p_0) \rfloor_{v_0}^{0} \uplus \lfloor P(p) \rfloor_{p}^{0}; \Phi_0; 0) \text{ with}$$

$$\Phi_0 = \{\mathsf{w}_1 \xrightarrow{p_0, 0} sk(p_0)\}$$



Brands and Chaum



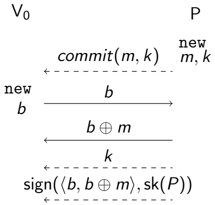
Example: Brands and Chaum - 1993

$$K_0 = (\lfloor V(v_0, p_0) \rfloor_{v_0}^{0} \uplus \lfloor P(p) \rfloor_{p}^{0}; \Phi_0; 0) \text{ with}$$

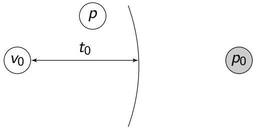
$$\Phi_0 = \{\mathrm{w}_1 \xrightarrow{p_0, 0} sk(p_0)\}$$

Execution of the protocol:

K_0



Brands and Chaum



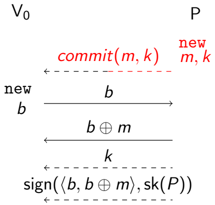
Example: Brands and Chaum - 1993

$$K_0 = (\lfloor V(v_0, p_0) \rfloor_{v_0}^{0} \uplus \lfloor P(p) \rfloor_{p}^{0}; \Phi_0; 0) \text{ with}$$

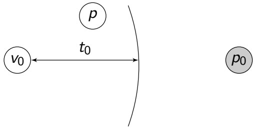
$$\Phi_0 = \{\mathrm{w}_1 \xrightarrow{p_0, 0} sk(p_0)\}$$

Execution of the protocol:

$$K_0 \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\mathtt{out}(\mathtt{commit}(m,k))}_{\mathcal{T}_0}$$



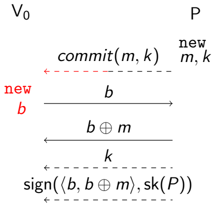
Brands and Chaum



Example: Brands and Chaum - 1993

$$K_0 = (\lfloor V(v_0, p_0)\rfloor^{0}_{v_0} \uplus \lfloor P(p)\rfloor^{0}_{p} ; \Phi_0; 0) \text{ with}$$

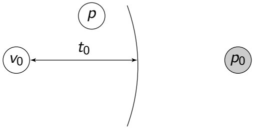
$$\Phi_0 = \{\mathsf{w}_1 \xrightarrow{p_0,0} sk(p_0)\}$$



Brands and Chaum

Execution of the protocol:

$$K_0 \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\mathtt{out}(\mathsf{commit}(m,k))}_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}(\mathsf{commit}(m,k))}_{\mathcal{T}_0} \xrightarrow{v_0,\tau}_{\mathcal{T}_0}$$



Example: Brands and Chaum - 1993

$$K_0 = (\lfloor V(v_0, p_0) \rfloor_{v_0}^{0} \uplus \lfloor P(p) \rfloor_{p}^{0}; \Phi_0; 0) \text{ with}$$

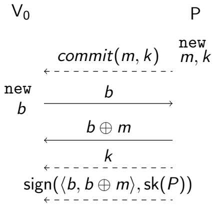
$$\Phi_0 = \{\mathrm{w}_1 \xrightarrow{p_0,0} sk(p_0)\}$$

Execution of the protocol:

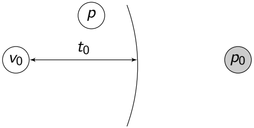
$$K_0 \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\tau}_{\mathcal{T}_0} \xrightarrow{p,\mathtt{out}(\mathrm{commit}(m,k))}_{\mathcal{T}_0} \xrightarrow{}_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}(\mathrm{commit}(m,k))}_{\mathcal{T}_0} \xrightarrow{v_0,\tau}_{\mathcal{T}_0} \xrightarrow{v_0,\tau}_{\mathcal{T}_0} (\mathcal{P}_1; \Phi_1; \delta_0)$$

with:

- $\mathcal{P}_1 = \lfloor V_1 \rfloor_{v_0}^{0} \uplus \lfloor P_1 \rfloor_{p}^{\delta_0}$
- $\Phi_1 = \{\mathrm{w}_1 \xrightarrow{p_0,0} sk(p_0), \mathrm{w}_2 \xrightarrow{p,0} \mathrm{commit}(m,k)\}$



Brands and Chaum



Security property: physical proximity

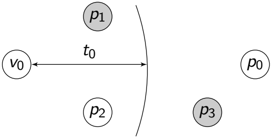
t_0-proximity

A protocol $\mathcal{P}_{prox}$ ensures t_0-proximity w.r.t. a topology $\mathcal{T} = (\mathcal{A}, \mathsf{Loc}, \mathcal{M}, v_0, p_0)$ and a configuration K if:

$$K \xrightarrow{tr}_{\mathcal{T}} (\lfloor \mathsf{end}(v_0, p_0) \rfloor_{v_0}^{t_{v_0}} ; \Phi; t) \Rightarrow \mathsf{Dist}_{\mathcal{T}}(v_0, p_0) < t_0.$$

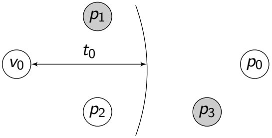
Classes of attacks

Mafia frauds - $\mathcal{C}_{\mathbf{MF}}$
(v_0 and p_0 are honest)

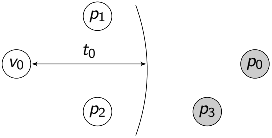


Classes of attacks

Mafia frauds - $\mathcal{C}_{\mathsf{MF}}$
(v_0 and p_0 are honest)

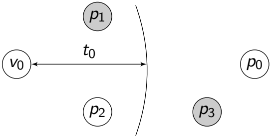


Distance hijacking - $\mathcal{C}_{\mathsf{DH}}$
(no dishonest agents close to v_0)

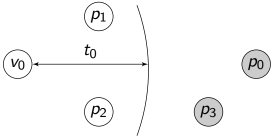


Classes of attacks

Mafia frauds - $\mathcal{C}_{\mathbf{MF}}$
(v_0 and p_0 are honest)



Distance hijacking - $\mathcal{C}_{\mathbf{DH}}$
(no dishonest agents close to v_0)



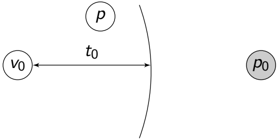
Mafia frauds (resp. Distance hijacking attacks)

A protocol $\mathcal{P}_{prox}$ is resistant against Mafia frauds (resp. Distance hijacking attacks) if for all topologies $\mathcal{T} \in \mathcal{C}_{\mathrm{MF}}$ (resp. $\mathcal{C}_{\mathrm{DH}}$) and initial configurations K:

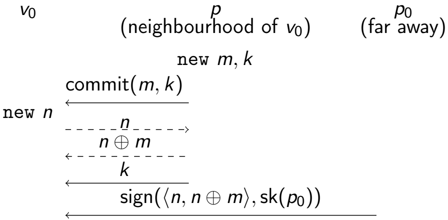
$$K \xrightarrow{tr}_{\mathcal{T}} \left(\lfloor \mathrm{end}(v_0, p_0) \rfloor_{v_0}^{t_{v_0}} ; \Phi ; t \right) \Rightarrow \mathrm{Dist}_{\mathcal{T}}(v_0, p_0) < t_0.$$

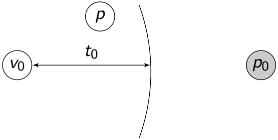
Example: Brands and Chaum - 1993

v_0	p (neighbourhood of v_0)	p_0 (far away)
	$\mathtt{new}\ m, k$	
	$\xleftarrow{\text{commit}(m,k)}$	
$\mathtt{new}\ n$	$\xrightarrow{n}$	
	$\xleftarrow{n \oplus m}$	
	$\xleftarrow{k}$	
		$\xleftarrow{\text{sign}(\langle n, n \oplus m\rangle, \text{sk}(p_0))}$



Example: Brands and Chaum - 1993





Trace of execution:

$$K_0 \xrightarrow{tr}{}^{*}_{\mathcal{T}} \left(\lfloor V_1 \rfloor_{v_0}^{0} \uplus \lfloor P_1 \rfloor_{p}^{\delta_0} ; \Phi_1 ; \delta_0 \right)$$

Example: Brands and Chaum - 1993

v_0 | p (neighbourhood of v_0) | p_0 (far away)

new m, k

$\xleftarrow{\text{commit}(m, k)}$

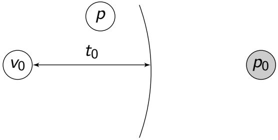
new n

$\xrightarrow{n}$

$\xleftarrow{n \oplus m}$

$\xleftarrow{k}$

$\xleftarrow{\text{sign}(\langle n, n \oplus m\rangle, \text{sk}(p_0))}$



Trace of execution:

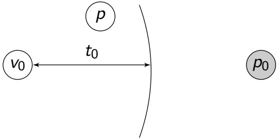
$$K_0 \xrightarrow{tr}{}^{*}_{\mathcal{T}} (\lfloor V_1 \rfloor^{0}_{v_0} \uplus \lfloor P_1 \rfloor^{\delta_0}_{p} ; \Phi_1 ; \delta_0)$$

$$\xrightarrow{v_0,\mathtt{out}(n)}_{\mathcal{T}_0} \rightarrow_{\mathcal{T}_0} \xrightarrow{p,\mathtt{in}(n')}_{\mathcal{T}_0}$$

$$\xrightarrow{p,\mathtt{out}(n \oplus m)}_{\mathcal{T}_0} \xrightarrow{p,\mathtt{out}(k)}_{\mathcal{T}_0} \rightarrow_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}^{<2\times t_0}(n \oplus m)}_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}(k)}_{\mathcal{T}_0}$$

Example: Brands and Chaum - 1993

v_0		p (neighbourhood of v_0)	p_0 (far away)
		new m, k	
	$\xleftarrow{\text{commit}(m,k)}$		
new n	$\xrightarrow{n}$		
	$\xleftarrow{n \oplus m}$		
	$\xleftarrow{k}$		
	$\xleftarrow{\text{sign}(\langle n, n \oplus m\rangle, \text{sk}(p_0))}$		



Trace of execution:

$$K_0 \xrightarrow{tr}{}^{*}_{\mathcal{T}} (\lfloor V_1 \rfloor^{0}_{v_0} \uplus \lfloor P_1 \rfloor^{\delta_0}_{p}; \Phi_1; \delta_0)$$

$$\xrightarrow{v_0,\mathtt{out}(n)}_{\mathcal{T}_0} \rightarrow_{\mathcal{T}_0} \xrightarrow{p,\mathtt{in}(n')}_{\mathcal{T}_0}$$

$$\xrightarrow{p,\mathtt{out}(n \oplus m)}_{\mathcal{T}_0} \xrightarrow{p,\mathtt{out}(k)}_{\mathcal{T}_0} \rightarrow_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}^{<2\times t_0}(n \oplus m)}_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}(k)}_{\mathcal{T}_0}$$

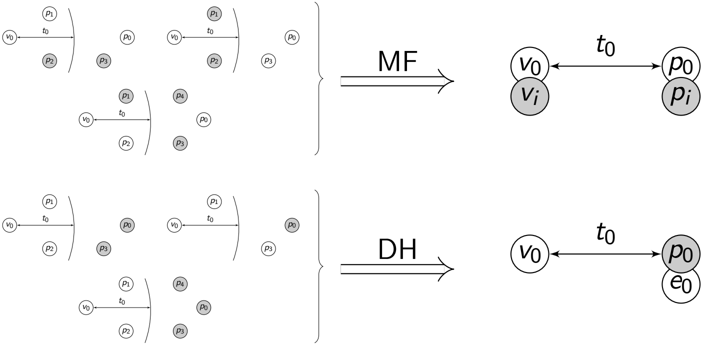
$$\rightarrow_{\mathcal{T}_0} \xrightarrow{v_0,\mathtt{in}(\text{sign}(\langle n, n \oplus m\rangle, \text{sk}(p_0)))}_{\mathcal{T}_0}$$

$$(\lfloor P_2 \rfloor^{3\delta_0 + 2\delta'_0}_{p} \uplus \lfloor \text{end}(v_0, p_0) \rfloor^{2\delta_0 + 2\delta'_0}_{v_0}; \Phi; 3\delta_0 + 2\delta'_0)$$

Table of contents

Reduction results

Only one topology is sufficient !

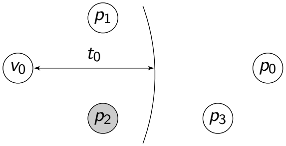


Mafia fraud attacks

Theorem

Let $\mathcal{P}_{prox}$ be an **executable** protocol.
$\mathcal{P}_{\mathrm{prox}}$ admits a Mafia fraud attack w.r.t. t_0-proximity, if and only if, there is an attack against t_0-proximity in the topology $\mathcal{T}_{\mathrm{MF}}$.

Sketch of proof:



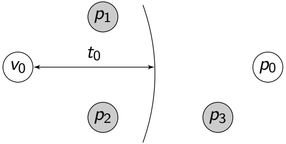
Mafia fraud attacks

Theorem

Let $\mathcal{P}_{prox}$ be an **executable** protocol.
$\mathcal{P}_{\mathrm{prox}}$ admits a Mafia fraud attack w.r.t. t_0-proximity, if and only if, there is an attack against t_0-proximity in the topology $\mathcal{T}_{\mathrm{MF}}$.

Sketch of proof:

1. the honest agents become malicious –> no executed processes



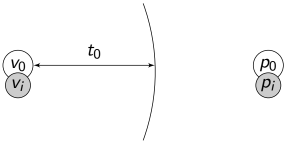
Mafia fraud attacks

Theorem

Let $\mathcal{P}_{prox}$ be an **executable** protocol.
$\mathcal{P}_{\text{prox}}$ admits a Mafia fraud attack w.r.t. t_0-proximity, if and only if, there is an attack against t_0-proximity in the topology $\mathcal{T}_{\text{MF}}$.

Sketch of proof:

1. the honest agents become malicious –> no executed processes
2. we place them ideally [Nigam *et. al.*, 16]



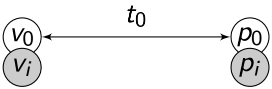
Mafia fraud attacks

Theorem

Let $\mathcal{P}_{prox}$ be an **executable** protocol.
$\mathcal{P}_{\mathrm{prox}}$ admits a Mafia fraud attack w.r.t. t_0-proximity, if and only if, there is an attack against t_0-proximity in the topology $\mathcal{T}_{\mathrm{MF}}$.

Sketch of proof:

1. the honest agents become malicious –> no executed processes
2. we place them ideally [Nigam *et. al.*, 16]
3. we shorten the distance



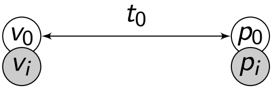
Mafia fraud attacks

Theorem

Let $\mathcal{P}_{prox}$ be an **executable** protocol.
$\mathcal{P}_{\mathsf{prox}}$ admits a Mafia fraud attack w.r.t. t_0-proximity, if and only if, there is an attack against t_0-proximity in the topology $\mathcal{T}_{\mathsf{MF}}$.

Sketch of proof:

1. the honest agents become malicious –> no executed processes
2. we place them ideally [Nigam *et. al.*, 16]
3. we shorten the distance



Remark. This proof cannot be adapted for distance hijacking attacks !

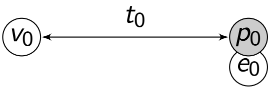
Distance hijacking attacks

Theorem

Let $\mathcal{P}_{prox}$ be a protocol such that the Verifier role respects the following grammar:

$$\begin{array}{rlclcl} P, Q := & \mathrm{end}(z_0, z_1) & \mid & \mathtt{in}(x).P & \mid & \mathtt{let}\ x = v\ \mathtt{in}\ P \\ \mid & \mathtt{new}\ n.P & \mid & \mathtt{out}(u).P & \mid & \color{red}{\mathtt{reset}.\mathtt{out}(u').\mathtt{in}^{<t}(x)}.P \end{array}$$

If $\mathcal{P}_{\mathrm{prox}}$ admits a Distance hijacking attack w.r.t. t_0-proximity, then $\overline{\mathcal{P}_{\mathrm{prox}}}$ admits an attack against t_0-proximity in the topology $\mathcal{T}_{\mathrm{DH}}$.



$\mathcal{T}_{\mathrm{DH}}$

In $\overline{\mathcal{P}_{\mathrm{prox}}}$ we only keep guards computed by v_0.

Table of contents

ProVerif [Blanchet, 01]

ProVerif is a verifier tool for cryptographic protocols.

http://proverif.inria.fr/

Advantages:

- fully automated
- can model many cryptographic primitives
- handles an unbounded number of sessions
- can model protocols defined by phases (*e.g.* e-voting)

Issues:

- termination not guaranteed
- can answer *"cannot be proved"*
- no time and locations

Phases in ProVerif

Common use: model protocols defined by steps (*e.g.* e-voting) and security properties (*e.g.* forward secrecy).

Process algebra: $P \;:=\; 0 \mid \mathtt{new}\ n.P \mid \mathtt{let}\ x = v\ \mathtt{in}\ P$
$\qquad\qquad\qquad\qquad \mid\; \mathtt{out}(u).P \mid \mathtt{in}(x).P \mid !\,P \mid i : P$

The semantics is extended with:

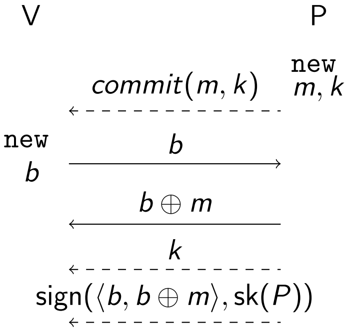
$$(\mathcal{P}; \phi; i) \xrightarrow{\text{phase } i'} (\mathcal{P}; \phi; i') \quad \text{with } i' > i.$$

$$(i : \mathtt{out}(u).P \uplus \mathcal{P}; \phi; i) \xrightarrow{\mathtt{out}(u)} (i : P \uplus \mathcal{P}; \phi \uplus \{\mathrm{w} \to u\}; i)$$

Example: $\mathcal{P} := \{0 : out(n_0).1 : out(n_1);\ 0 : in(x_0).1 : in(x_1)\}$

Translation into ProVerif
$Transf(\mathcal{T}, \mathcal{P}, t_0)$

$$
\begin{array}{l}
V_0(v_0, p_0) := \\
\quad \mathtt{in}(y_c).\mathtt{new}\ b. \\
\quad \mathtt{reset}.\mathtt{out}(b).\mathtt{in}^{<2\times t_0}(y_0). \\
\quad \mathtt{in}(y_k).\mathtt{in}(y_{\mathsf{sign}}). \\
\quad \mathtt{let}\ y_m = \mathsf{open}(y_c, y_k)\ \mathtt{in} \\
\quad \mathtt{let}\ y_{msg} = \mathsf{getmsg}(y_{\mathsf{sign}})\ \mathtt{in} \\
\quad \mathtt{let}\ y_{\mathsf{check}} = \mathsf{check}(y_{\mathsf{sign}}, \mathsf{vk}(z_P))\ \mathtt{in} \\
\quad \mathtt{let}\ y_{\mathsf{eq}} = \mathsf{eq}(\langle b, b \oplus y_m \rangle, y_{msg})\ \mathtt{in} \\
\quad \mathsf{end}(z_V, z_P). \\
\quad 0
\end{array}
$$

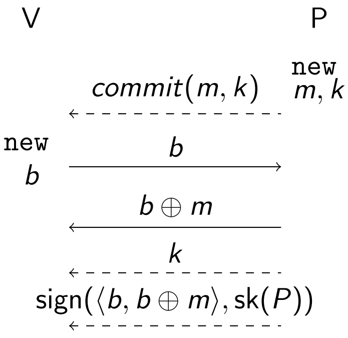


Brands and Chaum

Translation into ProVerif

$\mathit{Transf}(\mathcal{T}, \mathcal{P}, t_0)$

$\overline{V_0}(v_0, p_0) :=$
$\quad \mathtt{in}(y_c).\mathtt{new}\ b.$
$\quad 1: \mathtt{out}(b).\mathtt{in}(y_0).$
$\quad 2: \mathtt{in}(y_k).\mathtt{in}(y_{\mathsf{sign}}).$
$\quad \mathtt{let}\ y_m = \mathsf{open}(y_c, y_k)\ \mathtt{in}$
$\quad \mathtt{let}\ y_{msg} = \mathsf{getmsg}(y_{\mathsf{sign}})\ \mathtt{in}$
$\quad \mathtt{let}\ y_{\mathsf{check}} = \mathsf{check}(y_{\mathsf{sign}}, \mathsf{vk}(z_P))\ \mathtt{in}$
$\quad \mathtt{let}\ y_{\mathsf{eq}} = \mathsf{eq}(\langle b, b \oplus y_m \rangle, y_{msg})\ \mathtt{in}$
$\quad \mathsf{end}(z_V, z_P).$
$\quad 0$



Brands and Chaum

Translation into ProVerif $Transf(\mathcal{T}, \mathcal{P}_{\text{prox}}, t_0)$

Given a process P we define:

- $P^{<}$: all the possible ways of spitting P in the phases 0, 1 and 2
- $P^{\geq}$: all the possible ways of spitting P in the phases 0 and 2

$Transf(\mathcal{T}, \mathcal{P}, t_{\text{prox}})$ is the multiset of processes derived from $\mathcal{P}$ when applying:

- $\cdot^{<}$ for all instantiated roles of $\mathcal{P}$ executed by agents close to v_0
- $\cdot^{\geq}$ for all instantiated roles of $\mathcal{P}$ executed by agents far from v_0

Proposition

If $(\mathcal{P}_{\text{prox}} \cup V_0)$ admits an attack w.r.t. t_0-proximity in $\mathcal{T}$ then $(Transf(\mathcal{T}, \mathcal{P}, t_0) \uplus \overline{V_0}(v_0, p_0); \Phi_{init}; 0)$ admits an attack in ProVerif.

Case analysis - DB protocols

Protocols	MF	DH
Brands and Chaum	✓	×
Meadows *et al.* $(n_V \oplus n_P, P)$	✓	✓
Meadows *et al.* $(n_V, n_P \oplus P)$	✓	×
TREAD-Asymmetric	×	×
TREAD-Symmetric	✓	×
MAD (One-Way)	✓	×
Swiss-Knife	✓	✓
Munilla *et al.*	✓	✓
CRCS	✓	×
Hancke and Kuhn	✓	✓

(×: attack found, ✓: proved secure)
Coherent with the formal analysis recently done by Mauw *et. al.* using Tamarin

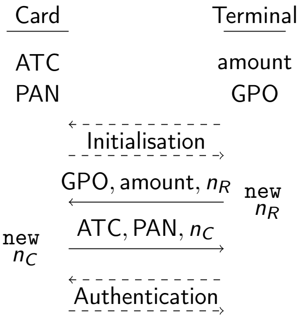
PaySafe

[Chothia *et. al.*, *2015*]

Goal: prevent relay attacks

Existing formal analyses:

- Chothia *et. al.*, 2015: similar approach using ProVerif's phases but without formal development
- Mauw *et. al.*, 2018: cannot analyse specifically mafia fraud attacks
  → find a DF attack which is not relevant



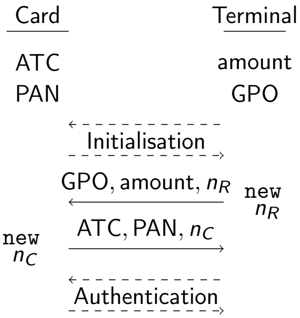
PaySafe

[Chothia *et. al.*, *2015*]

Goal: prevent relay attacks

Existing formal analyses:

- Chothia *et. al.*, 2015: similar approach using ProVerif's phases but without formal development
- Mauw *et. al.*, 2018: cannot analyse specifically mafia fraud attacks
  $\rightarrow$ find a DF attack which is not relevant

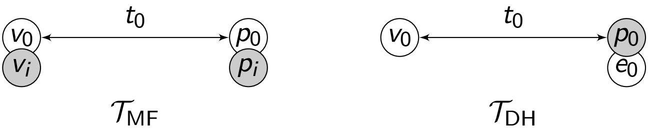


Our analysis: PaySafe is proved MF-resistant

Conclusion

We have adapted the standard applied Pi-Calculus to take into account time and locations.

We obtained **two reduction results** that reduce the number of relevant topologies that need to be studied to only 2.



We provide a solution to prove t_0-proximity using a **usual verification tool**, ProVerif, and we applied it to analyse well-known protocols.

Future work

⇒ Define a more precise notion of time.

⇒ Take into account **Terrorist frauds**:

Terrorist frauds

A remote dishonest prover cooperates with another dishonest agent, close to the verifier, to authenticates himself to the prover without giving any advantages for future attacks.