Symbolic verification of distance bounding protocols

Stéphanie Delaune

Univ Rennes, CNRS, IRISA, France

$\longrightarrow$ joint work with Alexandre Debant and Cyrille Wiedling

Security protocols everywhere !

Cryptographic protocols

- small programs designed to secure communication
  e.g. secrecy, authentication, anonymity, . . .
- use cryptographic primitives
  e.g. encryption, signature, ...

Security protocols everywhere !

Cryptographic protocols

- small programs designed to secure communication
  e.g. secrecy, authentication, anonymity, . . .
- use cryptographic primitives
  e.g. encryption, signature, ...

The network is unsecure!
Communications take place over a public network like the Internet.

Verifying security protocols: a difficult task

- testing their resilience against well-known attacks is not sufficient;
- manual security analysis is error-prone.

⟶ **Caution:** Do not underestimate your opponents!

Security

Defects in e-passports allow real-time tracking

This threat brought to you by RFID

The register - Jan. 2010

Lifestyle › Tech › News

Contactless card theft: Users warned to watch out for 'digital pickpockets'

Independent - Feb. 2016

A sucessful approach: formal symbolic verification

$\longrightarrow$ provides a rigorous framework and automatic tools to analyse security protocols and find their logical flaws.

TAMARIN
Tamarin prover interactive mode

ProVerif

AVANTSSAR

A sucessful approach: formal symbolic verification

$\longrightarrow$ provides a rigorous framework and automatic tools to analyse security protocols and find their logical flaws.

TAMARIN Tamarin prover interactive mode ProVerif AVANTSSAR

Some success stories

- 2011: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail
  $\longrightarrow$ Armando *et al.* using Avantssar
- 2018: TLS 1.3 formally verified before its deployment
  $\longrightarrow$ project miTLS : `https://www.mitls.org`

Contactless systems everywhere !

$\longrightarrow$ security property: authentication with **physical proximity**

Contactless systems everywhere !


$\longrightarrow$ security property: authentication with **physical proximity**

Brands and Chaum distance bounding protocol (1993)

$$P \to V : \quad commit(m, k)$$

$$\boxed{\begin{array}{ll} V \to P : & chall \\ P \to V : & T,\ chall \oplus m \end{array}}$$

$$2 \times dist(V, P) \leq \Delta t \times c$$

$$P \to V : \ k, \ \ Sign_P(m, chall \oplus m)$$

Contactless systems everywhere !

$\longrightarrow$ security property: authentication with **physical proximity**

Brands and Chaum distance bounding protocol (1993)

$P \rightarrow V: \quad commit(m, k)$

$V \rightarrow P: \quad chall$

$P \rightarrow V: \quad T, \; chall \oplus m$

$2 \times dist(V, P) \leq \Delta t \times c$

$P \rightarrow V: \; k, \; Sign_P(m, chall \oplus m)$

$\longrightarrow$ We need a framework that allows one to model transmission delay, location of participants, and timing constraints.

Some related works

1993: 1^{st} DB protocol proposed by Brands and Chaum
$\longrightarrow$ since then, many protocols + "formal" security analysis usually done in the computational model

2007-2016: analysis of DB protocols in the symbolic model

- Basin *et al.* - Isabelle/HOL (CSF'09)
- Cremers *et al.* distance-hijacking attack (S&P'12)

$\longrightarrow$ lack of automation to support the security analysis.

2017-today: A lot of progress has been done !

- Tamarin-based framework: Jorge's thesis (more this afternoon)
- ProVerif-based framework: Chothia et al. (USENIX'18) & PhD thesis of Alexandre Debant (more in one year !)

Contributions

A flavour of the PhD thesis of Alexandre Debant !

Our results:

1. A symbolic model suitable to analyse DB protocols together with some reduction results to automate the security analysis

   $\longrightarrow$ for distance fraud (including distance hijacking), mafia fraud, and also terrorist fraud
2. Integration in the ProVerif verification tool and many case studies

$\longrightarrow$ Results published at FST&TCS 2018 and currently under submission at ESORICS 2019 (terrorist fraud).

Outline

A symbolic model with time and location

Reduction results

Case studies relying on Proverif

Outline

A symbolic model with time and location

Reduction results

Case studies relying on Proverif

Messages as terms

Terms are built from names $\mathcal{N}$, and function symbols in Σ.

Example

$$\Sigma_{\mathrm{ex}} = \{\mathrm{senc}/2,\ \mathrm{sdec}/2,\ \mathrm{kdf}/3,\ \mathrm{shk}/2,\ \mathrm{ok}/0,\ \mathrm{eq}/2,\ \mathrm{ans}/3,\ \oplus/2,\ 0/0\}.$$

Properties of the cryptographic primitives are reflected using an equational theory and some rewriting rules:

Example

$$\begin{array}{rclcrcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) & \qquad & x \oplus 0 & = & x \\ x \oplus y) & = & y \oplus x & & x \oplus x & = & 0 \end{array}$$

$$\mathrm{sdec}(\mathrm{senc}(x, y), y) \ \to \ x \qquad\qquad \mathrm{eq}(x, x) \ \to \ \mathrm{ok}$$

Messages as terms

Terms are built from names $\mathcal{N}$, and function symbols in Σ.

Example

$\Sigma_{\text{ex}} = \{\text{senc}/2, \text{sdec}/2, \text{kdf}/3, \text{shk}/2, \text{ok}/0, \text{eq}/2, \text{ans}/3, \oplus/2, 0/0\}$.

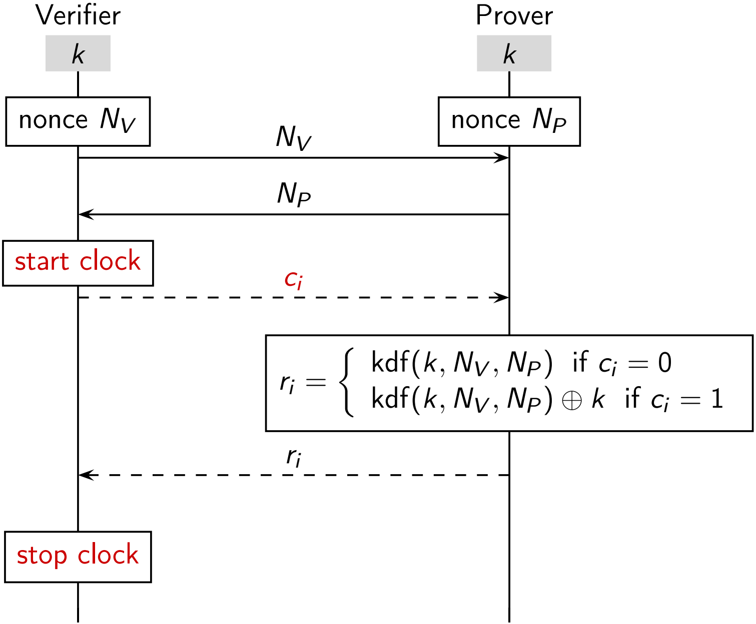
Properties of the cryptographic primitives are reflected using an equational theory and some rewriting rules:

Example

$$\begin{array}{rcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) \\ x \oplus y) & = & y \oplus x \end{array} \qquad \begin{array}{rcl} x \oplus 0 & = & x \\ x \oplus x & = & 0 \end{array}$$

$$\text{sdec}(\text{senc}(x, y), y) \rightarrow x \qquad \text{eq}(x, x) \rightarrow \text{ok}$$

Example: Modified Hancke and Kuhn (2005)



Protocols as processes

$$
\begin{array}{rll}
P, Q := & 0 & \text{null process} \\
 & \mid \mathsf{in}(x).P & \text{input} \\
 & \mid \mathsf{out}(u).P & \text{output} \\
 & \mid \mathsf{let}\ x = v\ \mathsf{in}\ P & \text{computation and test} \\
 & \mid \mathtt{new}\ n.P & \text{fresh name generation} \\
 & \mid \mathsf{reset}.P & \text{reset of the local clock} \\
 & \mid \mathsf{in}^{<t}(x).P & \text{guarded input}
\end{array}
$$

Example: Verifier role parametrized by z_0 and z_1.

$$
\begin{array}{rl}
\mathtt{V}(z_0, z_1) := & \mathtt{new}\ n_V.\mathsf{out}(n_V).\mathsf{in}(x_N). \\
 & \mathsf{reset}.\mathtt{new}\ c.\mathsf{out}(c).\mathsf{in}^{<2\times t_0}(x_{\mathsf{rep}}). \\
 & \mathsf{let}\ x_0 = \mathsf{kdf}(\mathsf{shk}(z_1, z_0), n_V, x_N)\ \mathsf{in} \\
 & \mathsf{let}\ x_1 = \mathsf{shk}(z_1, z_0) \oplus x_0\ \mathsf{in} \\
 & \mathsf{let}\ x_{ok} = \mathsf{eq}(x_{\mathsf{rep}}, \mathsf{ans}(c, x_0, x_1))\ \mathsf{in} \\
 & \mathsf{end}(z_0, z_1)
\end{array}
$$

$\longrightarrow$ the rapid phase is abstracted by a single challenge/response exchange, and operations performed at the bit level are abstracted too.

Protocols as processes

$$
\begin{array}{rll}
P, Q := & 0 & \text{null process} \\
& \mid \mathsf{in}(x).P & \text{input} \\
& \mid \mathsf{out}(u).P & \text{output} \\
& \mid \mathsf{let}\ x = v\ \mathsf{in}\ P & \text{computation and test} \\
& \mid \mathtt{new}\ n.P & \text{fresh name generation} \\
& \mid \mathsf{reset}.P & \text{reset of the local clock} \\
& \mid \mathsf{in}^{<t}(x).P & \text{guarded input}
\end{array}
$$

Example: Verifier role parametrized by z_0 and z_1.

$$
\begin{array}{rl}
\mathtt{V}(z_0, z_1) := & \mathtt{new}\ n_V.\mathsf{out}(n_V).\mathsf{in}(x_N). \\
& \mathsf{reset}.\mathtt{new}\ c.\mathsf{out}(c).\mathsf{in}^{<2\times t_0}(x_{\mathsf{rep}}). \\
& \mathsf{let}\ x_0 = \mathsf{kdf}(\mathsf{shk}(z_1, z_0), n_V, x_N)\ \mathsf{in} \\
& \mathsf{let}\ x_1 = \mathsf{shk}(z_1, z_0) \oplus x_0\ \mathsf{in} \\
& \mathsf{let}\ x_{ok} = \mathsf{eq}(x_{\mathsf{rep}}, \mathsf{ans}(c, x_0, x_1))\ \mathsf{in} \\
& \mathsf{end}(z_0, z_1)
\end{array}
$$

$\longrightarrow$ the rapid phase is abstracted by a single challenge/response exchange, and operations performed at the bit level are abstracted too.

Topology and Configuration

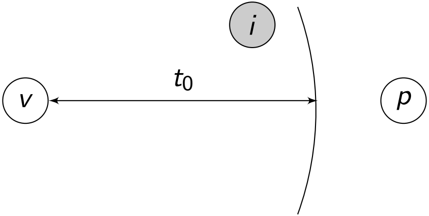
A topology is a tuple $\mathcal{T}_0 = (\mathcal{A}_0, \mathcal{M}_0, \mathsf{Loc}_0)$ where:

- $\mathcal{A}_0$ the agents;
- $\mathcal{M}_0$ the subset of malicious agents;
- $\mathsf{Loc}_0 : \mathcal{A}_0 \to \mathbb{R}^3$ defines the location of each agent.

We define: $\mathsf{Dist}_{\mathcal{T}_0}(a, b) = \frac{\|\mathsf{Loc}_0(a) - \mathsf{Loc}_0(b)\|}{c_0}$ for any $a, b \in \mathcal{A}_0$

$\longrightarrow$ **only the distance between nodes matters !**

Example:



Topology and Configuration

A topology is a tuple $\mathcal{T}_0 = (\mathcal{A}_0, \mathcal{M}_0, \mathrm{Loc}_0)$ where:

- $\mathcal{A}_0$ the agents;
- $\mathcal{M}_0$ the subset of malicious agents;
- $\mathrm{Loc}_0 : \mathcal{A}_0 \to \mathbb{R}^3$ defines the location of each agent.

We define: $\mathrm{Dist}_{\mathcal{T}_0}(a, b) = \frac{\|\mathrm{Loc}_0(a) - \mathrm{Loc}_0(b)\|}{c_0}$ for any $a, b \in \mathcal{A}_0$

$\longrightarrow$ **only the distance between nodes matters !**

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

- $\mathcal{P}$ is a multiset of extended process $\lfloor \mathcal{P} \rfloor_a^{t_a}$ with $a \in \mathcal{A}$, $t_a \in \mathbb{R}^+$;
- $\Phi = \{\mathrm{w}_1 \xrightarrow{a_1, t_1} u_1, \ldots, \mathrm{w}_n \xrightarrow{a_n, t_n} u_n\}$ is a *a frame*;
- $t \in \mathbb{R}^+$ is the global time.

Semantics

$\longrightarrow$ transition system over configurations, parametrised by a topology $\mathcal{T}_0$

- $(\mathcal{P}; \Phi; t) \longrightarrow_{\mathcal{T}_0} (\mathsf{Shift}(\mathcal{P}, \delta); \Phi; t + \delta)$ with $\delta \geq 0$;
- $(\lfloor \mathsf{out}(u).P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathsf{out}(u)}_{\mathcal{T}_0} (\lfloor P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi \uplus \mathsf{w} \xrightarrow{a,t} u; t)$
  with $\mathsf{w} \in \mathcal{W}$ fresh
- $\ldots$
- $(\lfloor \mathsf{in}^{<t_g}(x).P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathsf{in}(v)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto v\} \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t)$

"An agent is responsible of the corresponding output v", i.e.

There exist an agent b, a time t_b and a recipe R such that:

(i) $t_b \leq t - \mathsf{Dist}_{\mathcal{T}_0}(b, a)$,

(ii) $R\Phi\downarrow = v$, and

(iii) all $\mathsf{w} \in vars(R)$ are available to b at time t_b.

Moreover, $|R| > 1$ only if b is malicious, i.e. $b \in \mathcal{M}_0$, and $t' < t_g$.

Semantics

$\longrightarrow$ transition system over configurations, parametrised by a topology $\mathcal{T}_0$

- $(\mathcal{P}; \Phi; t) \longrightarrow_{\mathcal{T}_0} (\mathrm{Shift}(\mathcal{P}, \delta); \Phi; t + \delta)$ with $\delta \geq 0$;
- $(\lfloor \mathrm{out}(u).P \rfloor_a^{t'}) \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathrm{out}(u)}_{\mathcal{T}_0} (\lfloor P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi \uplus \mathrm{w} \xrightarrow{a,t} u; t)$
  with $\mathrm{w} \in \mathcal{W}$ fresh
- . . .
- $(\lfloor \mathrm{in}^{<t_g}(x).P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathrm{in}(v)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto v\} \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t)$

"An agent is responsible of the corresponding output v", i.e.

There exist an agent b, a time t_b and a recipe R such that:

(i) $t_b \leq t - \mathrm{Dist}_{\mathcal{T}_0}(b, a)$,

(ii) $R\Phi\downarrow = v$, and

(iii) all $\mathrm{w} \in vars(R)$ are available to b at time t_b.

Moreover, $|R| > 1$ only if b is malicious, i.e. $b \in \mathcal{M}_0$, and $t' < t_g$.

Semantics

$\longrightarrow$ transition system over configurations, parametrised by a topology $\mathcal{T}_0$

- $(\mathcal{P}; \Phi; t) \longrightarrow_{\mathcal{T}_0} (\mathsf{Shift}(\mathcal{P}, \delta); \Phi; t + \delta)$ with $\delta \geq 0$;
- $(\lfloor \mathsf{out}(u).P \rfloor_a^{t'}) \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathsf{out}(u)}_{\mathcal{T}_0} (\lfloor P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi \uplus \mathsf{w} \xrightarrow{a,t} u; t)$
  with $\mathsf{w} \in \mathcal{W}$ fresh
- ...
- $(\lfloor \mathsf{in}^{<t_g}(x).P \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t) \xrightarrow{a,\mathsf{in}(v)}_{\mathcal{T}_0} (\lfloor P\{x \mapsto v\} \rfloor_a^{t'} \uplus \mathcal{P}; \Phi; t)$

"An agent is responsible of the corresponding output v", i.e.

There exist an agent b, a time t_b and a recipe R such that:

(i) $t_b \leq t - \mathsf{Dist}_{\mathcal{T}_0}(b, a)$,

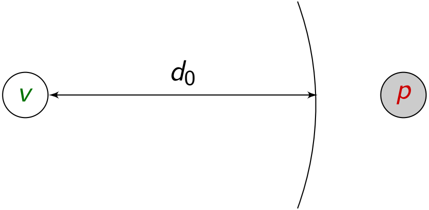
(ii) $R\Phi\downarrow = v$, and

(iii) all $\mathsf{w} \in vars(R)$ are available to b at time t_b.

Moreover, $|R| > 1$ only if b is malicious, i.e. $b \in \mathcal{M}_0$, and $t' < t_g$.

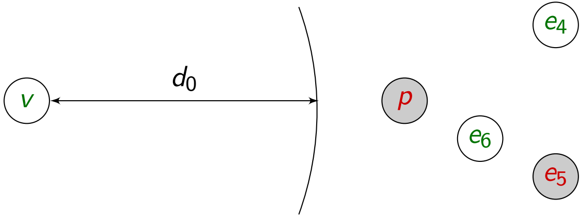
Different types of frauds

Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)



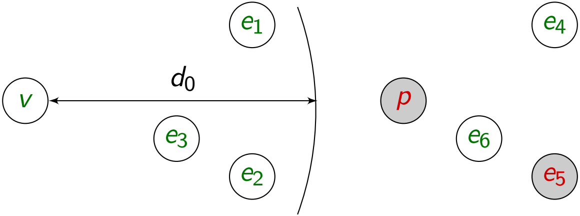
Different types of frauds

Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)



Different types of frauds

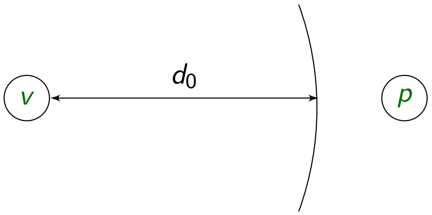
Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)



Different types of frauds

Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)

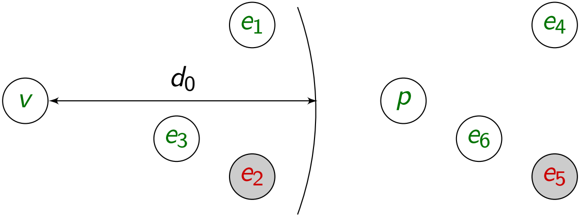
Mafia fraud: An attacker should not be able to abuse a far away honest prover to pass the protocol.



Different types of frauds

Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)

Mafia fraud: An attacker should not be able to abuse a far away honest prover to pass the protocol.



Different types of frauds

Distance fraud (including distance hijacking): A malicious prover should not be able to successfully complete a session with an honest verifier who is far away (even with the help of some honest agents in the neighbourhood)

Mafia fraud: An attacker should not be able to abuse a far away honest prover to pass the protocol.

Terrorist fraud: A far away malicious prover colludes with the attacker who is close to the verifier to pass the protocol, and this help should not allow the attacker to authenticate later on.

Security properties

A valid initial configuration $(\mathcal{P}; \Phi_0; 0)$ w.r.t. a topology $\mathcal{T}$ is a configuration such that:

- $\mathcal{P}$ contains instances of $\lfloor \mathtt{P}(a, b) \rfloor_{a}^{0}$ and $\lfloor \mathtt{V}(a, b) \rfloor_{a}^{0}$;
- Φ_0 is the initial knowledge (uniform w.r.t. honest/malicious agent names)

Mafia fraud

$\mathcal{P}_{\text{prox}}$ admits a mafia fraud w.r.t. t_0-proximity if there exists $\mathcal{T} \in \mathcal{C}_{\text{MF}}$, a valid initial configuration K_0 w.r.t. $\mathcal{T}$ such that:

$$K_0 \rightarrow_{\mathcal{T}} \left(\lfloor \mathsf{end}(v_0, p_0) \rfloor_{v_0}^{t'} \uplus \mathcal{P}; \Phi; t \right)$$

$\longrightarrow$ Distance fraud (including hijacking) can be defined in a rather similar way.

Security properties

A valid initial configuration $(\mathcal{P};\Phi_0;0)$ w.r.t. a topology $\mathcal{T}$ is a configuration such that:

- $\mathcal{P}$ contains instances of $\lfloor \mathtt{P}(a,b)\rfloor_a^0$ and $\lfloor \mathtt{V}(a,b)\rfloor_a^0$;
- Φ_0 is the initial knowledge (uniform w.r.t. honest/malicious agent names)

Mafia fraud

$\mathcal{P}_{\mathsf{prox}}$ admits a mafia fraud w.r.t. t_0-proximity if there exists $\mathcal{T} \in \mathcal{C}_{\mathsf{MF}}$, a valid initial configuration K_0 w.r.t. $\mathcal{T}$ such that:

$$K_0 \rightarrow_{\mathcal{T}} (\lfloor \mathsf{end}(v_0, p_0)\rfloor_{v_0}^{t'} \uplus \mathcal{P};\Phi;t)$$

$\longrightarrow$ Distance fraud (including hijacking) can be defined in a rather similar way.

Security properties

A valid initial configuration $(\mathcal{P}; \Phi_0; 0)$ w.r.t. a topology $\mathcal{T}$ is a configuration such that:

- $\mathcal{P}$ contains instances of $\lfloor \mathtt{P}(a, b) \rfloor_a^0$ and $\lfloor \mathtt{V}(a, b) \rfloor_a^0$;
- Φ_0 is the initial knowledge (uniform w.r.t. honest/malicious agent names)

Mafia fraud

$\mathcal{P}_{\text{prox}}$ admits a mafia fraud w.r.t. t_0-proximity if there exists $\mathcal{T} \in \mathcal{C}_{\text{MF}}$, a valid initial configuration K_0 w.r.t. $\mathcal{T}$ such that:

$$K_0 \rightarrow_{\mathcal{T}} (\lfloor \text{end}(v_0, p_0) \rfloor_{v_0}^{t'} \uplus \mathcal{P}; \Phi; t)$$

$\longrightarrow$ Distance fraud (including hijacking) can be defined in a rather similar way.

Terrorist fraud

$\longrightarrow$ **More tricky !** A semi-dishonest prover who colludes with the attacker to authenticate once.

A semi-dishonest prover for $\mathcal{P}_{\mathsf{prox}}$ is a process P_{sd} together with an initial frame Φ_{sd} such that:

$$(\{\lfloor \mathtt{V}(v_0, p_0) \rfloor^{0}_{v_0} ; \lfloor P_{\mathsf{sd}} \rfloor^{0}_{p_0}\}; \emptyset; 0) \rightarrow_{\mathcal{T}_0} (\{\lfloor \mathsf{end}(v_0, p_0) \rfloor^{t_v}_{v_0} ; \lfloor 0 \rfloor^{t_p}_{p_0}\}; \Phi_{\mathsf{sd}}; t)$$

Terrorist fraud resistant

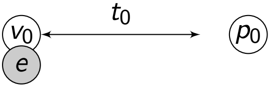
$\mathcal{P}_{\mathsf{prox}}$ is terrorist fraud resistant w.r.t. t_0-proximity if for all semi-dishonest prover P_{sd} with frame Φ_{sd}, there exist $\mathcal{T} \in \mathcal{C}_{\mathsf{MF}}$, a valid initial configuration K_0 with $\Phi_0 \cup \Phi_{\mathsf{sd}}$ as initial frame such that:

$$K_0 \rightarrow_{\mathcal{T}} (\lfloor \mathsf{end}(v_0, p_0) \rfloor^{t'}_{v_0} \uplus \mathcal{P}; \Phi; t).$$

Terrorist fraud

$\longrightarrow$ **More tricky !** A semi-dishonest prover who colludes with the attacker to authenticate once.

A semi-dishonest prover for $\mathcal{P}_{\mathsf{prox}}$ is a process P_{sd} together with an initial frame Φ_{sd} such that:



$$(\{\lfloor \mathtt{V}(v_0, p_0) \rfloor_{v_0}^{0} \,;\, \lfloor P_{\mathsf{sd}} \rfloor_{p_0}^{0}\}; \emptyset; 0) \rightarrow_{\mathcal{T}_0} (\{\lfloor \mathsf{end}(v_0, p_0) \rfloor_{v_0}^{t_v} \,;\, \lfloor 0 \rfloor_{p_0}^{t_p}\}; \Phi_{\mathsf{sd}}; t)$$

Terrorist fraud resistant

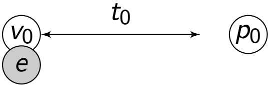
$\mathcal{P}_{\mathsf{prox}}$ is terrorist fraud resistant w.r.t. t_0-proximity if for all semi-dishonest prover P_{sd} with frame Φ_{sd}, there exist $\mathcal{T} \in \mathcal{C}_{\mathsf{MF}}$, a valid initial configuration K_0 with $\Phi_0 \cup \Phi_{\mathsf{sd}}$ as initial frame such that:

$$K_0 \rightarrow_{\mathcal{T}} (\lfloor \mathsf{end}(v_0, p_0) \rfloor_{v_0}^{t'} \uplus \mathcal{P}; \Phi; t).$$

Terrorist fraud

$\longrightarrow$ **More tricky !** A semi-dishonest prover who colludes with the attacker to authenticate once.

A semi-dishonest prover for $\mathcal{P}_{\mathsf{prox}}$ is a process P_{sd} together with an initial frame Φ_{sd} such that:



$$(\{\lfloor \mathtt{V}(v_0,p_0)\rfloor^{0}_{v_0}; \lfloor P_{\mathsf{sd}}\rfloor^{0}_{p_0}\}; \emptyset; 0) \rightarrow_{\mathcal{T}_0} (\{\lfloor \mathsf{end}(v_0,p_0)\rfloor^{t_v}_{v_0}; \lfloor 0\rfloor^{t_p}_{p_0}\}; \Phi_{\mathsf{sd}}; t)$$

Terrorist fraud resistant

$\mathcal{P}_{\mathsf{prox}}$ is terrorist fraud resistant w.r.t. t_0-proximity if for all semi-dishonest prover P_{sd} with frame Φ_{sd}, there exist $\mathcal{T} \in \mathcal{C}_{\mathsf{MF}}$, a valid initial configuration K_0 with $\Phi_0 \cup \Phi_{\mathsf{sd}}$ as initial frame such that:

$$K_0 \rightarrow_{\mathcal{T}} (\lfloor \mathsf{end}(v_0,p_0)\rfloor^{t'}_{v_0} \uplus \mathcal{P}; \Phi; t).$$

Terrorist fraud

Proposition

$\mathcal{P}$ admits a mafia fraud $\Rightarrow$ $\mathcal{P}$ is terrorist fraud resistant.

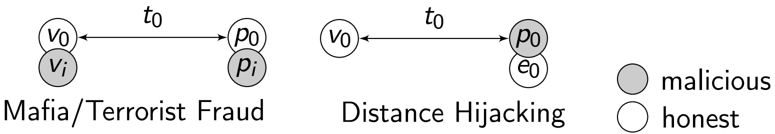
Brief comparison (with other definition in the symbolic setting):

- Chothia et al.'18: the terrorist prover is allowed to perform operations on behalf of the attacker ... and secrets may be revealed indirectly !
- Jorge's PhD thesis: share some similarities with ours. Their notion of valid extension seems to allow more behaviours than our notion of semi-dishonest prover.

Outline

One topology is enough !

It is actually sufficient to consider the following topology:

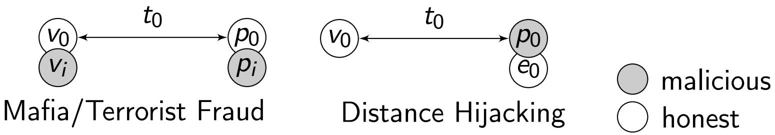


Main limitations regarding automation:

- Distance fraud (including distance hijacking): a topology with no attacker in the neighbourhood fo v_0;
- Terrorist fraud: We still have the "for all semi-dishonest prover" to handle.

One topology is enough !

It is actually sufficient to consider the following topology:



Main limitations regarding automation:

- Distance fraud (including distance hijacking): a topology with no attacker in the neighbourhood fo v_0;
- Terrorist fraud: We still have the "for all semi-dishonest prover" to handle.

One semi-dishonest prover is enough !

Our hypotheses: We consider a DB protocol such that:

- $\mathtt{V}(z_0, z_1) = \mathsf{block}_V.\mathtt{reset}.\mathtt{new}\ c.\mathsf{out}(c).\mathsf{in}^{<2\times t_0}(x).\mathsf{block}'_V$; and
- $\mathtt{P}(z_0, z_1) = \mathsf{block}_P.\mathsf{in}(y_c).\mathsf{out}(u).\mathsf{block}'_P$

where $\mathsf{block}^{(\prime)}_X$ do not contain reset and guarded input instructions.

Moreover, we assume that $u = C[y_c, u_1, \ldots, u_p]$ for some C made of quasi-free public symbols, with no occurrence of y_c in $u_1, \ldots, u_p$.

+ some mild hypotheses

Reduction result

We may restrict our attention to the most general semi-dishonest prover $\mathtt{P}^*$ defined as follows (with its associated frame Φ^*):

$$\mathsf{block}_P.\mathsf{out}(u_1)\ldots\mathsf{out}(u_k).\mathsf{in}(y_c).\mathsf{out}(u).\mathsf{block}'_P$$

One semi-dishonest prover is enough !

Our hypotheses: We consider a DB protocol such that:

- $\mathtt{V}(z_0, z_1) = \text{block}_V.\mathtt{reset}.\mathtt{new}\ c.\text{out}(c).\text{in}^{<2\times t_0}(x).\text{block}'_V$; and
- $\mathtt{P}(z_0, z_1) = \text{block}_P.\text{in}(y_c).\text{out}(u).\text{block}'_P$

where $\text{block}_X^{(')}$ do not contain reset and guarded input instructions.

Moreover, we assume that $u = C[y_c, u_1, \ldots, u_p]$ for some C made of quasi-free public symbols, with no occurrence of y_c in $u_1, \ldots, u_p$.

+ some mild hypotheses

Reduction result

We may restrict our attention to the most general semi-dishonest prover $\mathtt{P}^*$ defined as follows (with its associated frame Φ^*):

$$\text{block}_P.\text{out}(u_1)\ldots\text{out}(u_k).\text{in}(y_c).\text{out}(u).\text{block}'_P$$

One semi-dishonest prover is enough !

Our hypotheses: We consider a DB protocol such that:

- $\mathtt{V}(z_0, z_1) = \mathsf{block}_V.\mathsf{reset}.\mathtt{new}\ c.\mathsf{out}(c).\mathsf{in}^{<2\times t_0}(x).\mathsf{block}'_V$; and
- $\mathtt{P}(z_0, z_1) = \mathsf{block}_P.\mathsf{in}(y_c).\mathsf{out}(u).\mathsf{block}'_P$

where $\mathsf{block}_X^{(\prime)}$ do not contain reset and guarded input instructions.

Moreover, we assume that $u = C[y_c, u_1, \ldots, u_p]$ for some C made of quasi-free public symbols, with no occurrence of y_c in $u_1, \ldots, u_p$.

+ some mild hypotheses

Reduction result

We may restrict our attention to the most general semi-dishonest prover $\mathtt{P}^*$ defined as follows (with its associated frame Φ^*):

$$\mathsf{block}_P.\mathsf{out}(u_1)\ldots\mathsf{out}(u_k).\mathsf{in}(y_c).\mathsf{out}(u).\mathsf{block}'_P$$

Example: Modified Hancke and Kuhn

The original prover's role:

$$\begin{aligned}\mathtt{P}(p_0, v_0) := \quad & \mathtt{new}\ n_P.\mathsf{in}(y_N).\mathsf{out}(n_P).\\ & \mathsf{let}\ y_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), y_N, n_P)\ \mathsf{in}\\ & \mathsf{let}\ y_1 = \mathsf{shk}(p_0, v_0) \oplus y_0\ \mathsf{in}\\ & \\ & \mathsf{in}(y_c).\mathsf{out}(\mathsf{ans}(y_c, y_0, y_1)).0\end{aligned}$$

with its associated frame Φ^*

$$\begin{aligned}\Phi^* = \{\mathrm{w}_1 \xrightarrow{v_0,0} n_V, \mathrm{w}_2 \xrightarrow{p_0,0} n_P, \mathrm{w}_3 \xrightarrow{p_0,0} m_0,\\ \mathrm{w}_4 \xrightarrow{p_0,0} \mathsf{shk}(p_0, v_0) \oplus m_0, \mathrm{w}_5 \xrightarrow{v_0,0} c\}\end{aligned}$$

where $m_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), n_V, n_P)$.

Example: Modified Hancke and Kuhn

The most general semi-dishonest prover:

$$\begin{aligned} \mathtt{P}^* := \quad & \mathtt{new}\ n_P.\mathsf{in}(y_N).\mathsf{out}(n_P).\\ & \mathsf{let}\ y_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), y_N, n_P)\ \mathsf{in}\\ & \mathsf{let}\ y_1 = \mathsf{shk}(p_0, v_0) \oplus y_0\ \mathsf{in}\\ & \mathsf{out}(y_0).\mathsf{out}(y_1).\\ & \mathsf{in}(y_c).\mathsf{out}(\mathsf{ans}(y_c, y_0, y_1)).0 \end{aligned}$$

with its associated frame Φ^*

$$\begin{aligned} \Phi^* = \{ \mathrm{w}_1 \xrightarrow{v_0,0} n_V, \mathrm{w}_2 \xrightarrow{p_0,0} n_P, \mathrm{w}_3 \xrightarrow{p_0,0} m_0, \\ \mathrm{w}_4 \xrightarrow{p_0,0} \mathsf{shk}(p_0, v_0) \oplus m_0, \mathrm{w}_5 \xrightarrow{v_0,0} c \} \end{aligned}$$

where $m_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), n_V, n_P)$.

Example: Modified Hancke and Kuhn

The most general semi-dishonest prover:

$$\begin{aligned}\mathtt{P}^* := \quad & \mathtt{new}\ n_P.\mathsf{in}(y_N).\mathsf{out}(n_P).\\ & \mathsf{let}\ y_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), y_N, n_P)\ \mathsf{in}\\ & \mathsf{let}\ y_1 = \mathsf{shk}(p_0, v_0) \oplus y_0\ \mathsf{in}\\ & \mathsf{out}(y_0).\mathsf{out}(y_1).\\ & \mathsf{in}(y_c).\mathsf{out}(\mathsf{ans}(y_c, y_0, y_1)).0\end{aligned}$$

with its associated frame Φ^*

$$\Phi^* = \{\mathrm{w}_1 \xrightarrow{v_0,0} n_V, \mathrm{w}_2 \xrightarrow{p_0,0} n_P, \mathrm{w}_3 \xrightarrow{p_0,0} m_0,$$
$$\mathrm{w}_4 \xrightarrow{p_0,0} \mathsf{shk}(p_0, v_0) \oplus m_0, \mathrm{w}_5 \xrightarrow{v_0,0} c\}$$

where $m_0 = \mathsf{kdf}(\mathsf{shk}(p_0, v_0), n_V, n_P)$.

Our reduction result applies

re-authentication is possible with $\mathtt{P}^*$ $\Longrightarrow$ Modified Hancke and Kuhn is terrorist fraud resistant.

Outline

A symbolic model with time and location

Reduction results

Case studies relying on Proverif

ProVerif

$\longrightarrow$ mainly developed by B. Blanchet

http://proverif.inria.fr

- automatic and efficient tool for unbounded number of sessions;
- handle various primitives but not the exclusive-or operator

Some features:

- phase mechanism useful to model the fact that entities that are far away can not interact during the rapid phase.
- attacker behaviour is built-in and thus we slightly modify the tool to analyse distance hijacking

No miracle ! It may not terminate or sometimes simply say can not be proved, but works well in practice.

ProVerif

⟶ mainly developed by B. Blanchet

http://proverif.inria.fr

- automatic and efficient tool for unbounded number of sessions;
- handle various primitives but not the exclusive-or operator

Some features:

- phase mechanism useful to model the fact that entities that are far away can not interact during the rapid phase.
- attacker behaviour is built-in and thus we slightly modify the tool to analyse distance hijacking

No miracle ! It may not terminate or sometimes simply say can not be proved, but works well in practice.

ProVerif

$\longrightarrow$ mainly developed by B. Blanchet

http://proverif.inria.fr

- automatic and efficient tool for unbounded number of sessions;
- handle various primitives but not the exclusive-or operator

Some features:

- phase mechanism useful to model the fact that entities that are far away can not interact during the rapid phase.
- attacker behaviour is built-in and thus we slightly modify the tool to analyse distance hijacking

No miracle ! It may not terminate or sometimes simply say can not be proved, but works well in practice.

Case studies - Distance bounding protocols

We consider three kinds of fraud:

- Mafia fraud: the attacker aims at convincing an honest verifier that a far honest prover is actually close to it.
- Distance fraud (including hijacking): a far away dishonest prover aims at convincing an honest verifier that he is actually close to it.
- Terrorist fraud: a far away prover helps the attacker to authenticate on his behalf but this help can not be reused later on.

For our analysis, we consider the reduced topology, and the most general semi-dishonest prover when our result applies.

Results on distance bounding protocols

Protocols	MFR	DHR	TFR
Hancke and Kuhn	✓	✓	×
Modified Hancke and Kuhn	✓	✓	✓
Brands and Chaum	✓	×	(×)
MAD (One-Way)	✓	×	(×)
Munilla *et al.*	✓	✓	×
Swiss-Knife	✓	✓	✓
SKI	✓	✓	✓
SPADE	×	×	✓
SPADE Fixed	✓	×	✓
TREAD-SKey	✓	×	✓
TREAD-PKey	×	×	✓
TREAD-PKey Fixed	✓	×	✓

(×) not TFR considering a specific P_{sd} – our result does not apply.

Case studies - Payment protocols

Which frauds do we need to consider?

$\longrightarrow$ Perhaps more in Ioana's talk

Some additional difficulties:

- more complex messages, and a larger number of exchanges;
  $\longrightarrow$ not a real issue for ProVerif
- NXP: the threshold (used in the timing constraint) is not fixed in advance.
  $\longrightarrow$ we simply fix it !

Protocols	MFR	DHR	TFR
NXP	✓	×	×
PaySafe	✓	×	×

Not surprisingly, these protocols admit a distance hijacking attack and a terrorist fraud.

Case studies - Payment protocols

Which frauds do we need to consider?

$\longrightarrow$ Perhaps more in Ioana's talk

Some additional difficulties:

- more complex messages, and a larger number of exchanges;
  $\longrightarrow$ not a real issue for ProVerif
- NXP: the threshold (used in the timing constraint) is not fixed in advance.
  $\longrightarrow$ we simply fix it !

Protocols	MFR	DHR	TFR
NXP	✓	×	×
PaySafe	✓	×	×

Not surprisingly, these protocols admit a distance hijacking attack and a terrorist fraud.

Case studies - Payment protocols

Which frauds do we need to consider?

$\longrightarrow$ Perhaps more in Ioana's talk

Some additional difficulties:

- more complex messages, and a larger number of exchanges;
  $\longrightarrow$ not a real issue for ProVerif
- NXP: the threshold (used in the timing constraint) is not fixed in advance.
  $\longrightarrow$ we simply fix it !

Protocols	MFR	DHR	TFR
NXP	✓	×	×
PaySafe	✓	×	×

Not surprisingly, these protocols admit a distance hijacking attack and a terrorist fraud.

Conclusion

Our contributions:

- reduction results to automate the security analysis of distance bounding protocols in the symbolic setting;
- integration in ProVerif with many case studies;
- attack on the SPADE protocol (regarding mafia) and a fix has been proposed by the authors of SPADE.

Future work:

- Relax some conditions regarding our reduction result for the terrorist fraud;
- Improve the way the exclusive-or operator is considered in the existing tools.

Thanks for your attention!