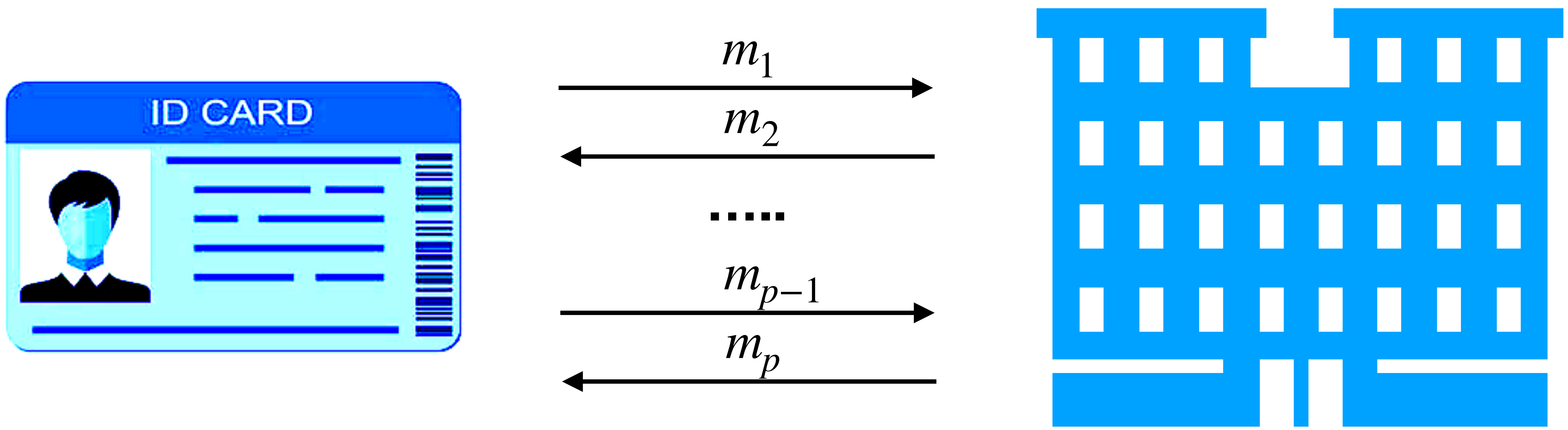
Symbolic verification of terrorist fraud

Alexandre Debant, Stéphanie Delaune, Cyrille Wiedling

Univ Rennes - IRISA - CNRS

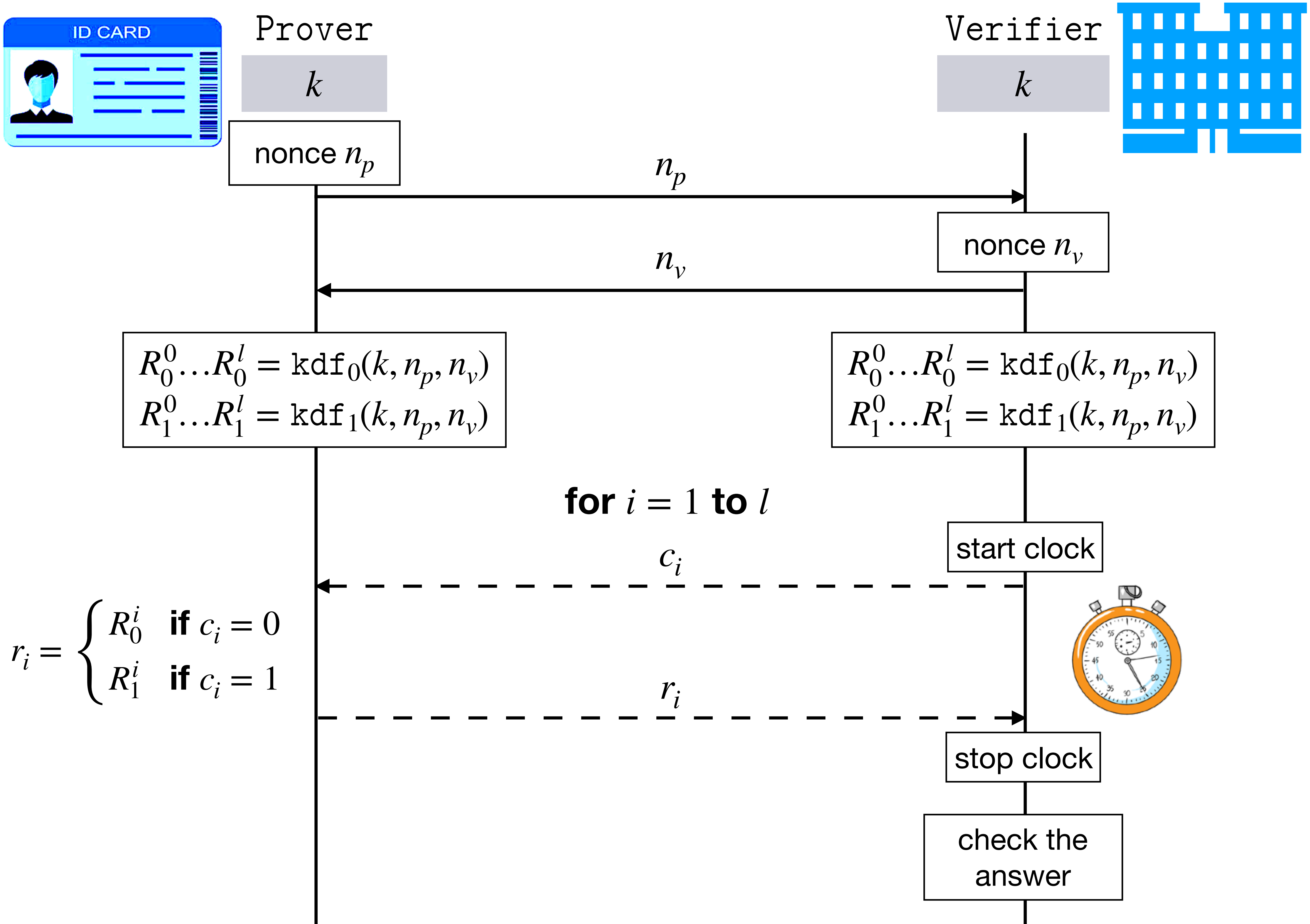
ESORICS 2019
September 23th 2019

Distance bounding protocols



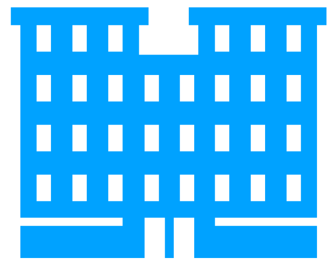
The access reader must **authenticate**
AND
verify the proximity of the card.

Hancke and Kuhn protocol (2005)



Attack scenarios

Mafia fraud (i.e., Man In the Middle)



An attacker, located in-between a verifier and a remote prover, tries to make the verifier think that they are close.

Attack scenarios

Mafia fraud (i.e., Man In the Middle)

An attacker, located in-between a verifier and a remote prover, tries to make the verifier think that they are close.

Distance fraud (or distance hijacking)

An attacker tries to abuse honest provers to be authenticated by a remote verifier.

Attack scenarios

Mafia fraud (i.e., Man In the Middle)

An attacker, located in-between a verifier and a remote prover, tries to make the verifier think that they are close.

Distance fraud (or distance hijacking)

An attacker tries to abuse honest provers to be authenticated by a remote verifier.

Terrorist fraud

An attacker accepts to collude with an accomplice to be authenticated once by a remote verifier but without giving him any avantage for future attacks.

Symbolic verification in a nutshell

Symbolic models:

– few abstractions (messages, attacker…)

\+ automatic procedures and existing tools

Existing tools exist:

TAMARIN
Tamarin prover interactive mode

ProVerif

AVANTSSAR

Some success stories:

Symbolic models for distance-bounding

Existing tools are not suitable to analyse DB protocols…

⟶ no model of time

+ the attacker relay messages without introducing any delay!

First models for DB protocols

Basin *et al*. **[CSF'09]** and Cremers *et al.* **[S&P'12]**

⟶ lack of automation…

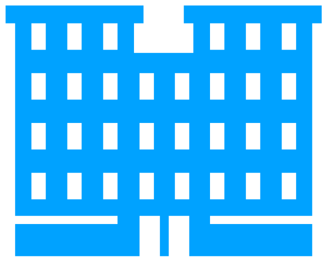
A lot of progress since last year:

➡ ProVerif encoding

- Chothia *et al.* - **[USENIX'18]**
- Our works - **[FSTTCS'18] [ESORICS'19]**

➡ Tamarin encoding: Mauw *et al.* - **[S&P'18] [CCS'19]**

About terrorist fraud



An attacker accepts to collude with an accomplice to be authenticated once by a remote verifier but without giving him any avantage for future attacks.

Chothia *et al*. - [USENIX'18]

- Non realistic definition of terrorist fraud
+ Fully automated verification using ProVerif

Mauw *et al*. - [CCS'19]

+ Satisfying definition of terrorist fraud (which corresponds to ours)
- Not fully automated verification (unbounded number of behaviors for collusion)

Contributions

1. A formal definition of terrorist fraud

2. Towards automation
   ➡ The attacker has a best strategy to collude
   ➡ There exists a most general topology

3. Case studies

Term algebra

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

Example

- Names: $\mathcal{N} = \{a, n, k\}$
- Signature: $\Sigma = \{\mathtt{senc}, \mathtt{sdec}, \mathtt{pair}, \mathtt{proj}_1, \mathtt{proj}_2, \mathtt{kdf}\}$

$$\mathtt{sdec}(\mathtt{senc}(x, y), y) \to x \qquad \mathtt{proj}_1(\mathtt{pair}(x, y)) \to x$$

$$\mathtt{proj}_2(\mathtt{pair}(x, y)) \to y$$

For example: $\mathtt{sdec}(\mathtt{senc}(\mathtt{proj}_1(\mathtt{pair}(n, m)), k), k) \downarrow =_E n$

Process algebra

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

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

Process algebra

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

$$
\begin{array}{rcll}
P & := & 0 & \text{null process} \\
 & | & \mathtt{new}\ n\,.\,P & \text{name restriction} \\
 & | & \mathtt{let}\ x = u\ \mathtt{in}\ P & \text{conditional declaration} \\
 & | & \mathtt{out}(u)\,.\,P & \text{output} \\
 & | & \mathtt{in}(x)\,.\,P & \text{input} \\
 & | & \mathtt{reset}\,.\,P & \text{personal clock reset} \\
 & | & \mathtt{in}^{<t}(x)\,.\,P & \text{guarded input}
\end{array}
$$

Process algebra

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

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

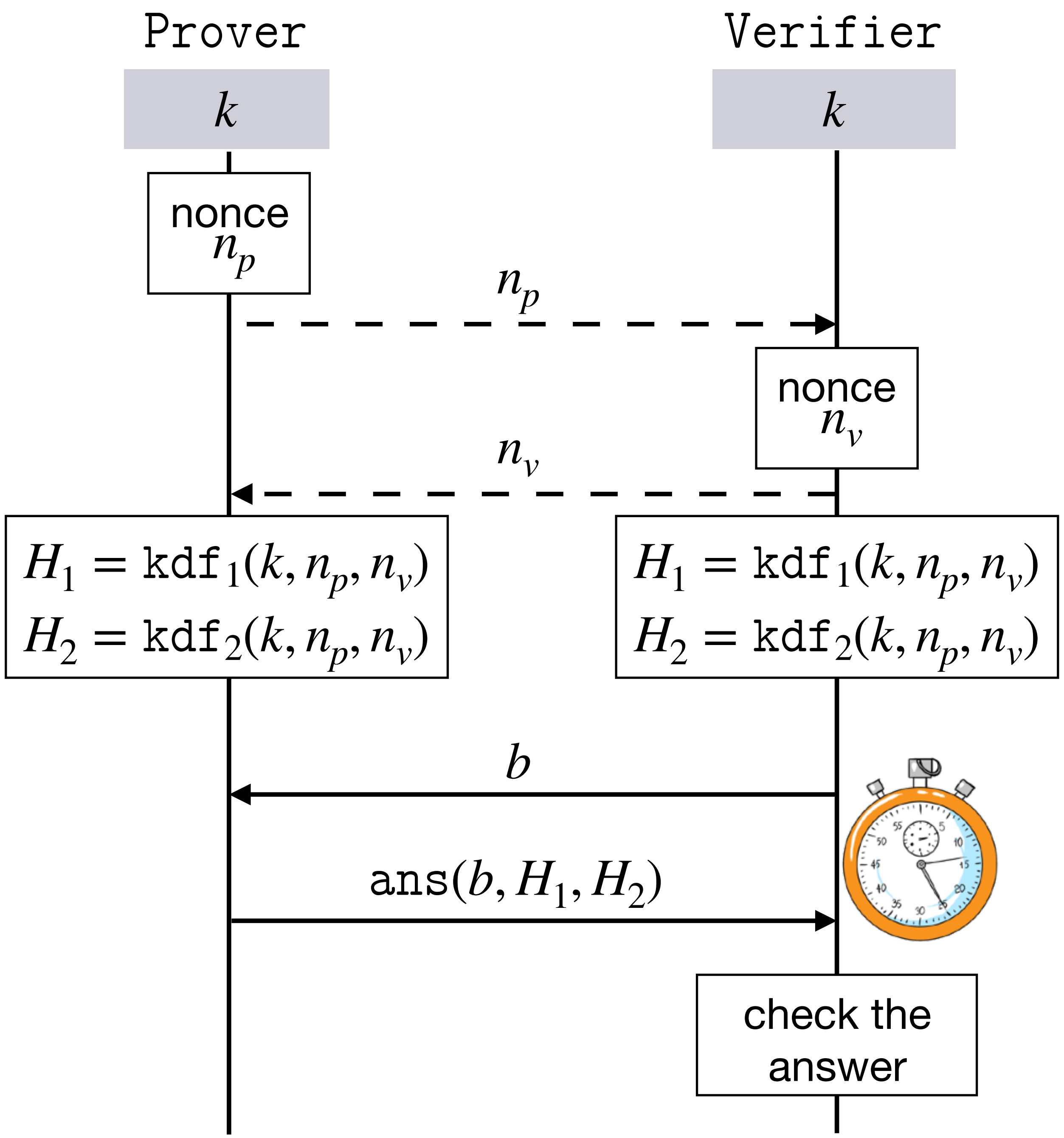
Distance-bounding protocol

A distance-bounding protocol is a pair (V, P) representing the verifier and the prover role.

Moreover, we assume that:

- ➡ $V = \mathtt{block}_V \,.\, \mathtt{reset} \,.\, \mathtt{new}\ b \,.\, \mathtt{out}(b) \,.\, \mathtt{in}^{<2\times t_0}(x) \,.\, \mathtt{block}'_V$
- ➡ $P = \mathtt{block}_P \,.\, \mathtt{in}(y_c) \,.\, \mathtt{out}(u) \,.\, \mathtt{block}'_P$

Process algebra: Hancke and Kuhn protocol



Process algebra: Hancke and Kuhn protocol

$V(z_k) :=$

$\quad \mathtt{in}(x)\,.$

$\quad \mathtt{new}\ n_v\,.\,\mathtt{out}(n_v)\,.$

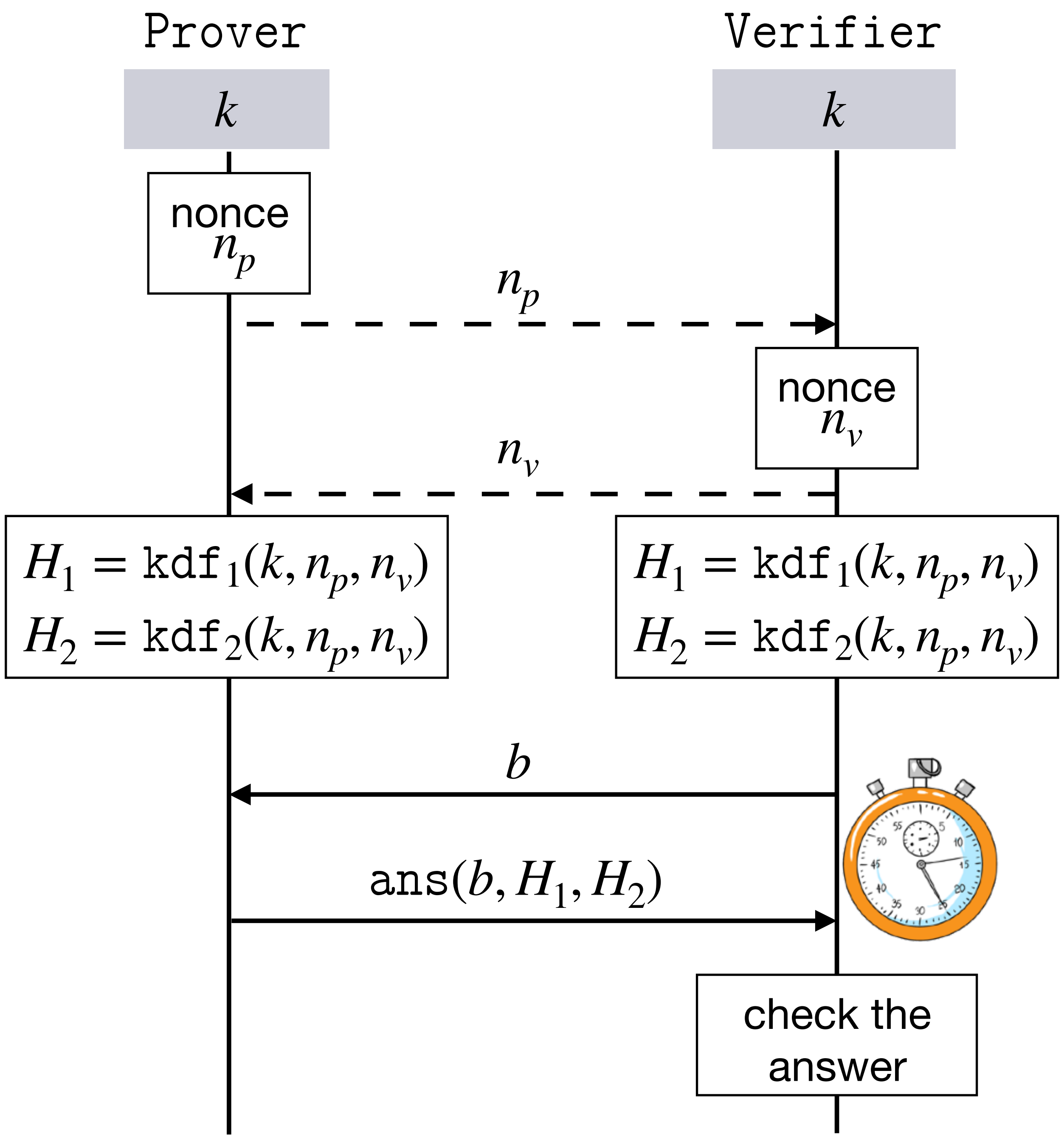
$\quad \mathtt{let}\ H_1 = \mathtt{kdf}_1(z_k, x, n_v)\ \mathtt{in}$

$\quad \mathtt{let}\ H_2 = \mathtt{kdf}_2(z_k, x, n_v)\ \mathtt{in}$

$\quad \mathtt{reset}\,.\,\mathtt{new}\ b\,.\,\mathtt{out}(b)\,.\,\mathtt{in}^{<2\times t_0}(y)\,.$

$\quad \mathtt{let}\ y_{test} = \mathtt{eq}(y, \mathtt{ans}(b, H_1, H_2))\ \mathtt{in}$

$\quad 0$



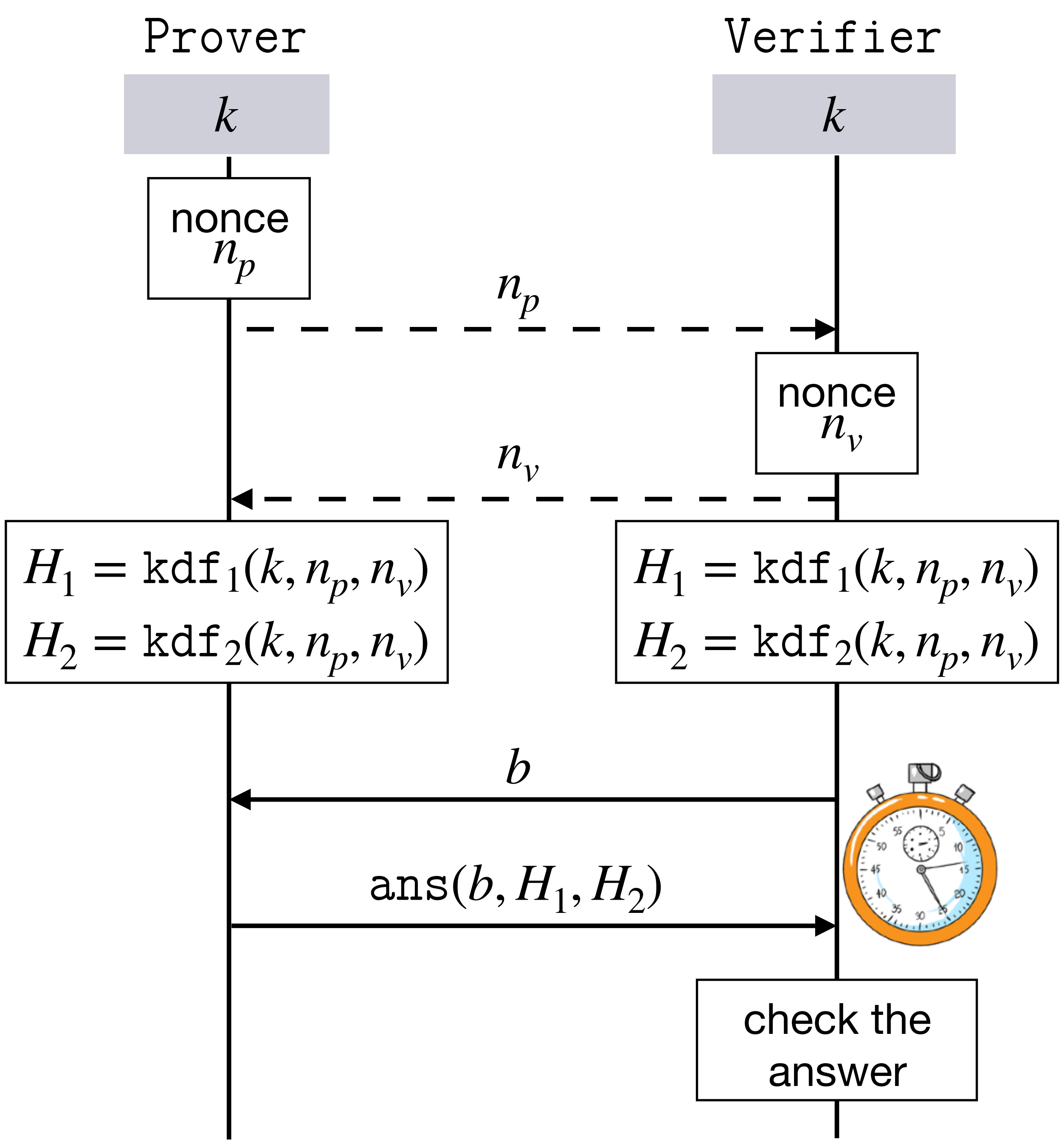
Process algebra: Hancke and Kuhn protocol

$V(z_k) :=$

$\mathtt{in}(x)\,.$
$\mathtt{new}\ n_v\,.\,\mathtt{out}(n_v)\,.$
$\mathtt{let}\ H_1 = \mathtt{kdf}_1(z_k, x, n_v)\ \mathtt{in}$
$\mathtt{let}\ H_2 = \mathtt{kdf}_2(z_k, x, n_v)\ \mathtt{in}$
$\mathtt{reset}\,.\,\mathtt{new}\ b\,.\,\mathtt{out}(b)\,.\,\mathtt{in}^{<2\times t_0}(y)\,.$
$\mathtt{let}\ y_{test} = \mathtt{eq}(y, \mathtt{ans}(b, H_1, H_2))\ \mathtt{in}$
0

$P(z_k) :=$

$\mathtt{new}\ n_p\,.\,\mathtt{out}(n_p)\,.$
$\mathtt{in}(x)\,.$
$\mathtt{let}\ H_1 = \mathtt{kdf}_1(z_k, n_p, x)\ \mathtt{in}$
$\mathtt{let}\ H_2 = \mathtt{kdf}_2(z_k, n_p, x)\ \mathtt{in}$
$\mathtt{in}(y)\,.$
$\mathtt{out}(\mathtt{ans}(y, H_1, H_2))$
0



Topology

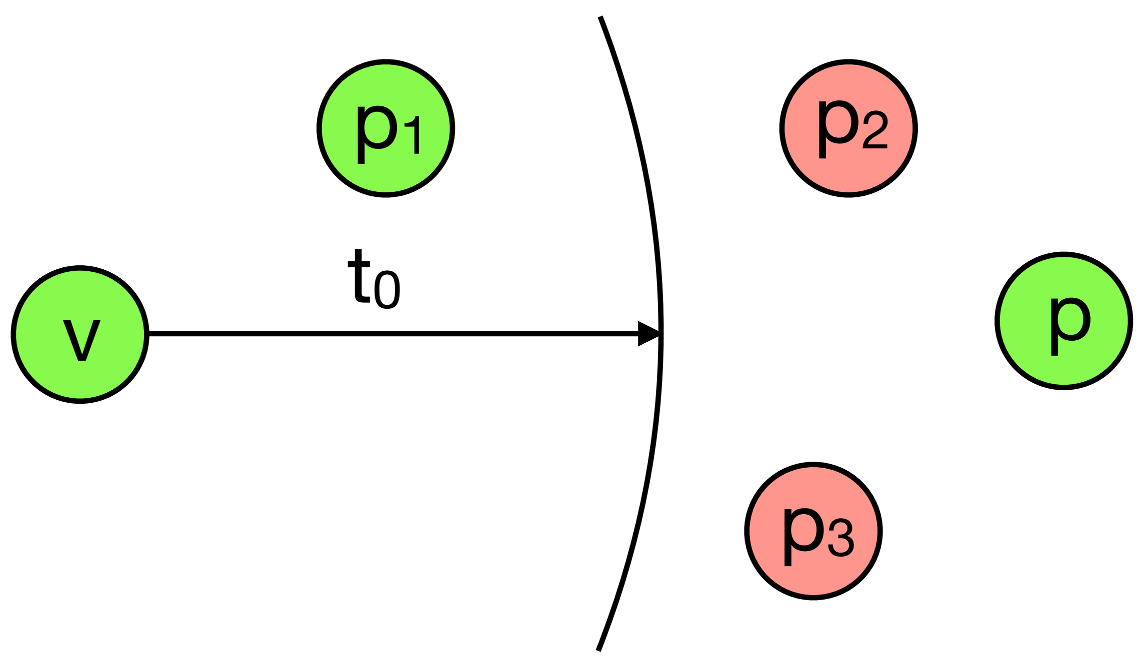
A **topology** is a tuple $\mathscr{T} = (\mathscr{A}, \texttt{Loc}, \mathscr{M}, v, p)$.

agents

locations

dishonest agents

specific agents



We define $\texttt{Dist}_{\mathscr{T}}(a,b) = \dfrac{\|\texttt{Loc}(a) - \texttt{Loc}(b)\|}{c}$

Configuration and semantics

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

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

TIME $\quad (\mathcal{P}; \Phi; t) \longrightarrow_{\mathcal{T}_0} (\mathcal{P}'; \Phi; t')$

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

Configuration and semantics

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

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

OUT $$(\lfloor \mathtt{out}(u).P \rfloor_a^{t_a} \uplus \mathscr{P}; \Phi; t) \xrightarrow{a, \mathtt{out}(u)}_{\mathscr{T}_0} (\lfloor P \rfloor_a^{t_a} \uplus \mathscr{P}; \Phi'; t)$$

with $\Phi' = \Phi \cup \{w \xrightarrow{a, t} u\}$

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 $t_a \in \mathcal{R}_+$
- $\Phi = \{\mathtt{w}_1 \xrightarrow{a_1, t_1} m_1, \ldots, \mathtt{w}_n \xrightarrow{a_n, t_n} m_n\}$ is a frame
- $t \in \mathcal{R}_+$ is the global time

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

if $\exists b \in \mathcal{A}, t_b \in \mathcal{R}_+$ such that $t_b \leq t - \mathtt{Dist}_{\mathcal{T}_0}(b, a)$ 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 $\bigcup_{c \in \mathcal{A}} \lfloor \Phi \rfloor_c^{t_b - \mathtt{Dist}_{\mathcal{T}_0}(c, b)}$

Configuration and semantics

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

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

NEW, LET, RESET…

Terrorist fraud resistance

Terrorist fraud resistance: A protocol is terrorist fraud resistant if for any possible attacker's behavior enabling his accomplice to be authenticated once, the accomplice gets an advantage to be authenticated later on.

Terrorist fraud resistance

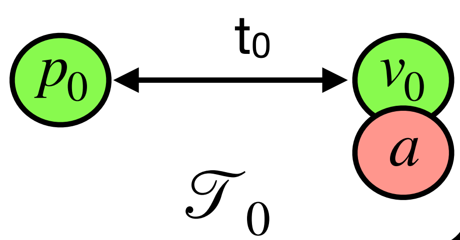
A protocol $\mathscr{P}$ is terrorist fraud resistant if

Terrorist fraud resistance

Terrorist fraud resistance: A protocol is terrorist fraud resistant if for any possible attacker's behavior enabling his accomplice to be authenticated once, the accomplice gets an advantage to be authenticated later on.

Given a protocol, a semi-dishonest prover is a process P_{sd} such that:

$$(\lfloor V(v_0, p_0)\rfloor_{v_0}^{0} \uplus \lfloor P_{\mathsf{sd}}\rfloor_{p_0}^{0}; \{\,\}; 0) \xrightarrow{tr}_{\mathcal{T}_0} (\lfloor 0\rfloor_{v_0}^{0} \uplus \lfloor 0\rfloor_{p_0}^{0}; \Phi_{\mathsf{sd}}; t)$$



Terrorist fraud resistance

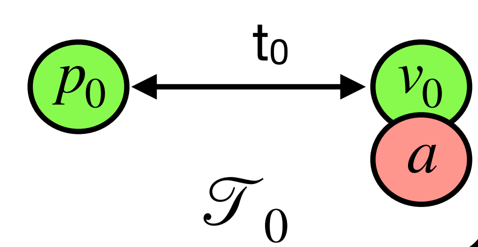
A protocol $\mathscr{P}$ is terrorist fraud resistant if for every semi-dishonest prover P_{sd},

Terrorist fraud resistance

Terrorist fraud resistance: A protocol is terrorist fraud resistant if for any possible attacker's behavior enabling his accomplice to be authenticated once, the accomplice gets an advantage to be authenticated later on.

Given a protocol, a semi-dishonest prover is a process P_{sd} such that:

$$(\lfloor V(v_0, p_0)\rfloor_{v_0}^{0} \uplus \lfloor P_{\mathsf{sd}}\rfloor_{p_0}^{0}; \{\}; 0) \xrightarrow{tr}_{\mathcal{T}_0} (\lfloor 0\rfloor_{v_0}^{0} \uplus \lfloor 0\rfloor_{p_0}^{0}; \Phi_{\mathtt{sd}}; t)$$



Terrorist fraud resistance

A protocol $\mathcal{P}$ is terrorist fraud resistant if for every semi-dishonest prover P_{sd}, there exists a topology $\mathcal{T} = (\mathcal{A}_0, \mathcal{M}_0, \mathtt{Loc}_0, v_0, p_0)$ such that $v_0, p_0 \notin \mathcal{M}_0$ and $\mathtt{Dist}_{\mathcal{T}}(v_0, p_0) \geq t_0$ and an initial configuration $(\mathcal{P}_0; \Phi_0; t_0)$ such that:

$$(\mathcal{P}_0; \Phi_0 \cup \Phi_{\mathsf{sd}}; t) \xrightarrow{tr}_{\mathcal{T}} (\lfloor \mathtt{end}(v_0, p_0)\rfloor_{v_0}^{t_v} \uplus \mathcal{P}; \Phi; t')$$

Contributions

1. A formal definition of terrorist fraud

2. **Towards automation**
   ➡ **The attacker has a best strategy to collude**
   ➡ **There exists a most general topology**

3. Case studies

Best strategy

Given a distance-bounding protocol, with a prover role

$$P = \texttt{block}_P \,.\, \texttt{in}(y_c) \,.\, \texttt{out}(u) \,.\, \texttt{block}'_P$$

the most general semi-dishonest prover P^* is defined as follows:

$$P^* = \texttt{block}_P \,.\, \texttt{out}(u_1, \ldots, u_n) \,.\, \texttt{in}(y_c) \,.\, \texttt{out}(u) \,.\, \texttt{block}'_P$$

where $u_1, \ldots, u_n$ are terms such that $u = \mathscr{C}[y_c, u_1, \ldots, u_n]$

Continuing our example:

$$\begin{aligned} P :=\ & \texttt{new}\ n_p.\texttt{out}(n_p) \,.\, \texttt{in}(x) \,.\, \texttt{let}\ H_1 = \texttt{kdf}_1(k, n_p, x)\ \texttt{in}\ \texttt{let}\ H_2 = \texttt{kdf}_2(k, n_p, x)\ \texttt{in} \\ & \quad \texttt{out}(\texttt{pair}(H_1, H_2)) \,. \\ & \quad \texttt{in}(y) \,.\, \texttt{out}(\texttt{ans}(y, H_1, H_2)) \,.\, 0 \end{aligned}$$

Best strategy

Given a distance-bounding protocol, with a prover role

$$P = \mathtt{block}_P \,.\, \mathtt{in}(y_c) \,.\, \mathtt{out}(u) \,.\, \mathtt{block}'_P$$

the most general semi-dishonest prover P^* is defined as follows:

$$P^* = \mathtt{block}_P \,.\, \mathtt{out}(u_1, \ldots, u_n) \,.\, \mathtt{in}(y_c) \,.\, \mathtt{out}(u) \,.\, \mathtt{block}'_P$$

where $u_1, \ldots, u_n$ are terms such that $u = \mathscr{C}[y_c, u_1, \ldots, u_n]$

Continuing our example:

$$\begin{aligned} P := {} & \mathtt{new}\ n_p.\mathtt{out}(n_p) \,.\, \mathtt{in}(x) \,.\, \mathtt{let}\ H_1 = \mathtt{kdf}_1(k, n_p, x)\ \mathtt{in}\ \mathtt{let}\ H_2 = \mathtt{kdf}_2(k, n_p, x)\ \mathtt{in} \\ & \mathtt{out}(\mathtt{pair}(H_1, H_2)) \,. \\ & \mathtt{in}(y) \,.\, \mathtt{out}(\mathtt{ans}(y, H_1, H_2)) \,.\, 0 \end{aligned}$$

Theorem: one semi-dishonest prover is enough

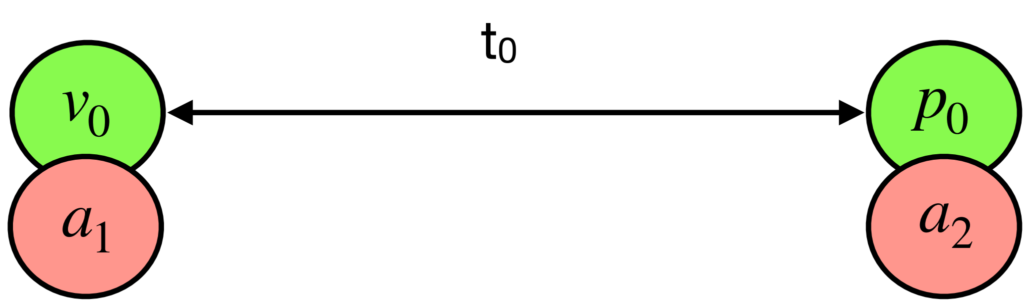
A distance-bounding protocol $\mathscr{P}_{\mathtt{prox}}$ is terrorist fraud resistant if and only if $\mathscr{P}_{\mathtt{prox}}$ is terrorist fraud resistant w.r.t. P^*.

One topology is enough

Theorem: one topology is enough

An executable distance-bounding protocol $\mathscr{P}_{\mathtt{prox}}$ is terrorist fraud resistant w.r.t. P^* if and only if there exists a valid initial configuration $(\mathscr{P}_0; \Phi_0; t_0)$ such that:

$$(\mathscr{P}_0; \Phi_0 \cup \Phi^*; t_0) \xrightarrow{tr}_{\mathscr{T}_{\text{MF}}} (\lfloor \mathtt{end}(v_0, p_0) \rfloor_{v_0}^{t_v} \uplus \mathscr{P}; \Phi; t)$$



$\mathscr{T}_{\mathtt{MF}}$

(similar to the reduction result proposed for mafia fraud at FSTTCS'18)

Contributions

1. A formal definition of terrorist fraud

2. Towards automation
   - ➡ The attacker has a best strategy to collude
   - ➡ There exists a most general topology

3. Case studies

Terrorist fraud resistance

Protocols	Assumptions for reducing: topologies	Assumptions for reducing: semi-dis. prover	Terrorist fraud resistance
Hancke and Kuhn	✓	✓	✗
Hancke and Kuhn *modified*	✓	✓	✓
Brands and Chaum	✓	✗	✗
Swiss-Knife	✓	✓	✓
SKI	✓	✓	✓
TREAD-Asymmetric	✓	✓	✓
TREAD-Asymmetric *fixed*	✓	✓	✓
TREAD-Symmetric	✓	✓	✓
Spade	✓	✓	✓
Spade *fixed*	✓	✓	✓
Munilla *et al.*	✓	✓	✗
MAD	✓	✗	✗
PaySafe	✓	✓	✗
NXP	✓	✓	✗

(✗: doesn't hold/attack found, ✓: holds/proved secure)
(we never obtained false attacks or non-termination)

Conclusion

Contributions

1. We propose a symbolic definition of terrorist fraud
2. We prove two reduction results enabling automation
   ➡ The attacker has a best strategy to collude
   ➡ There exists a most general topology
3. We verify numbers of protocols with the ProVerif tool

[FSTTCS'18] + [ESORICS'19] provide a framework to automatically analyse DB protocols w.r.t. the three main classes of attacks (i.e., MF, DH, TF).
(under few abstractions like bit-level operations,)

Conclusion

Contributions

1. We propose a symbolic definition of terrorist fraud
2. We prove two reduction results enabling automation
   ➡ The attacker has a best strategy to collude
   ➡ There exists a most general topology
3. We verify numbers of protocols with the ProVerif tool

[FSTTCS'18] + [ESORICS'19] provide a framework to automatically analyse DB protocols w.r.t. the three main classes of attacks (i.e., MF, DH, TF).
(under few abstractions like bit-level operations,)

Future work:

Extend the model with mobility i.e., **enable agents to move** during a session:
→ redefine each class of attacks
→ adapt existing results to enable automation
→ ….