Automated Verification of Privacy in Security Protocols:
Back and Forth Between Theory & Practice

Lucca Hirschi

LSV, ENS Paris-Saclay, Université Paris-Saclay, CNRS

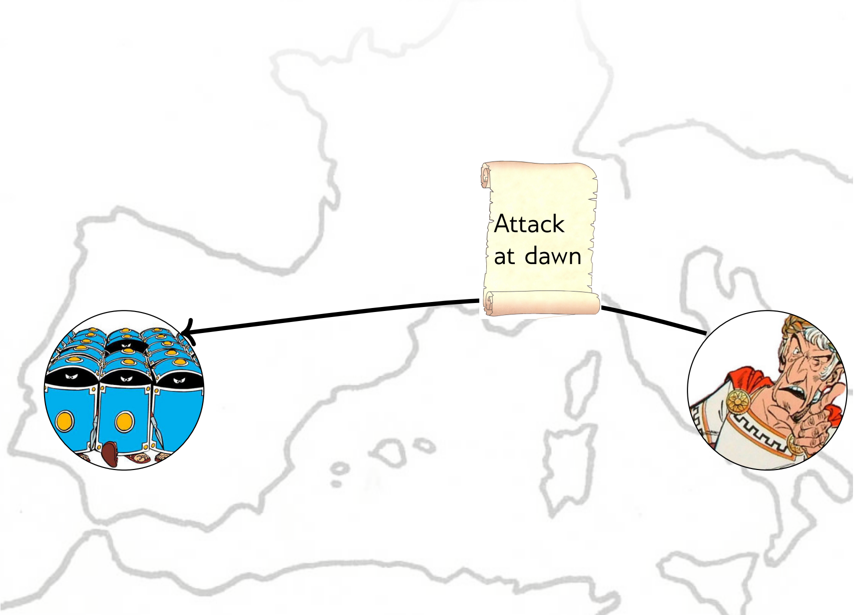
April 21st 2017

PhD advisors: David Baelde & Stéphanie Delaune

Automated Verification of Privacy in Security Protocols

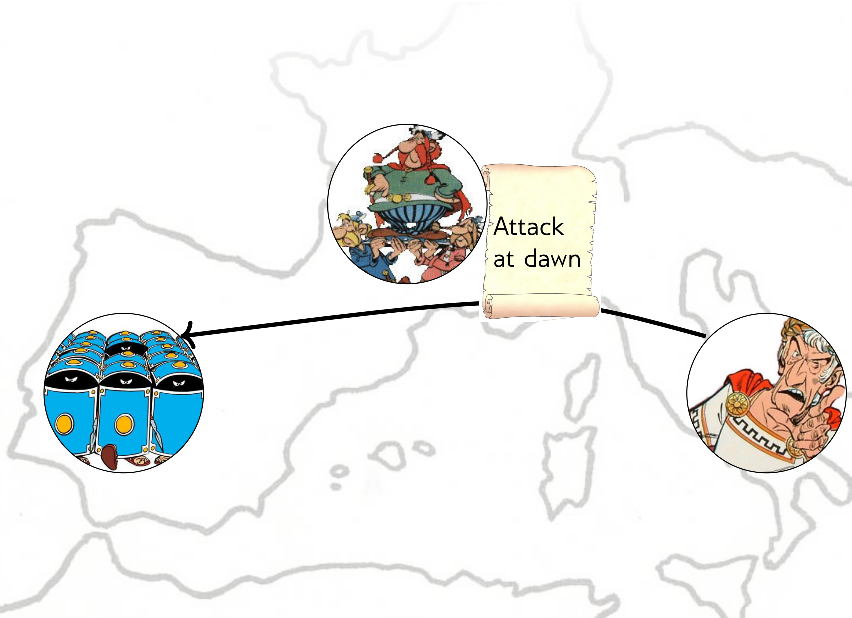
Security Protocols

Once upon a time...



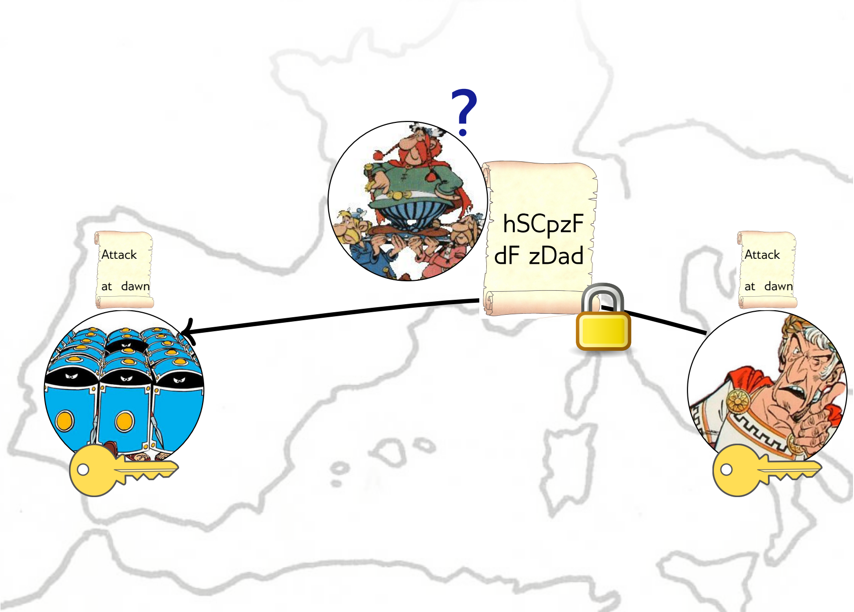
Security Protocols

Once upon a time...



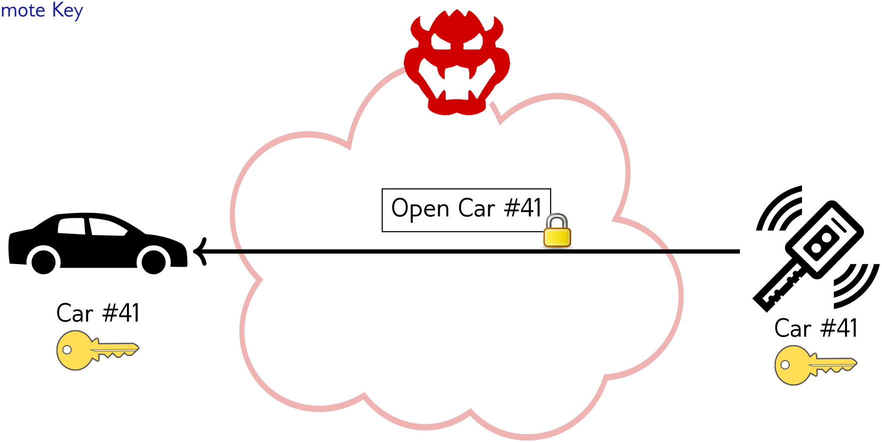
Security Protocols

Once upon a time...



Security Protocols

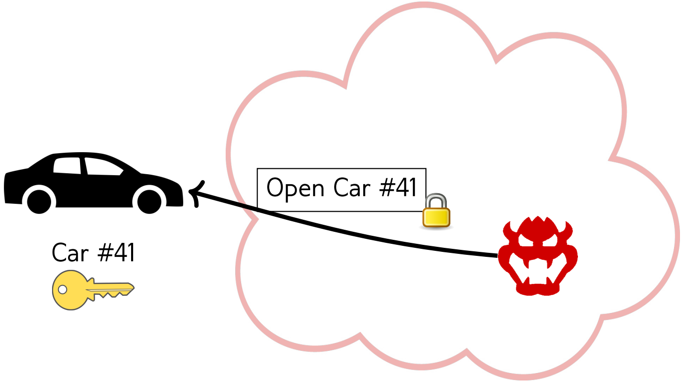
Remote Key



Security Protocols

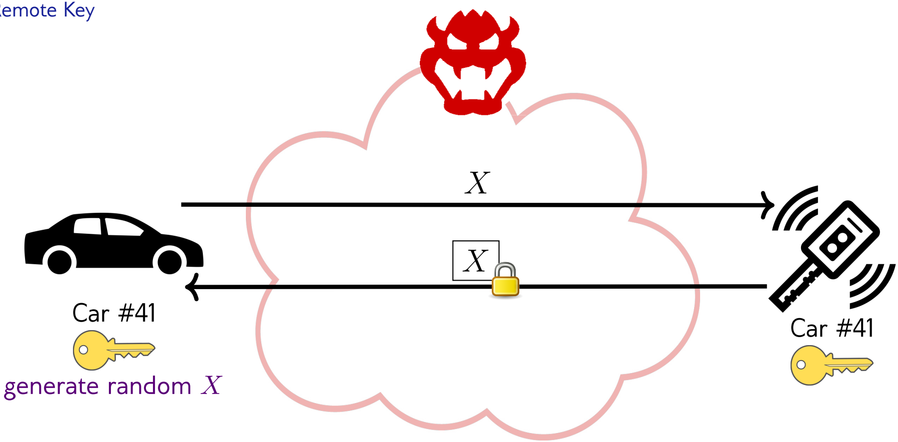
Remote Key

Attack: steals



Security Protocols

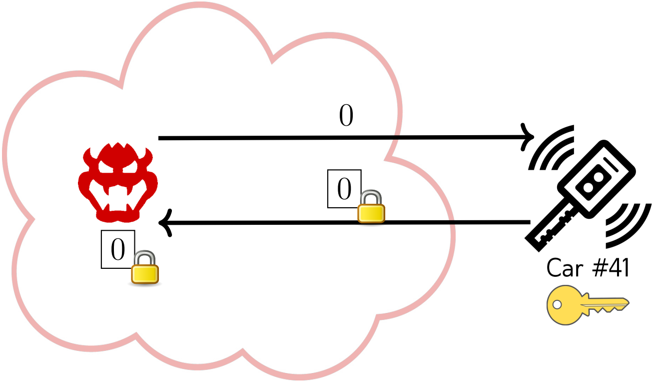
Remote Key



Security Protocols

Remote Key

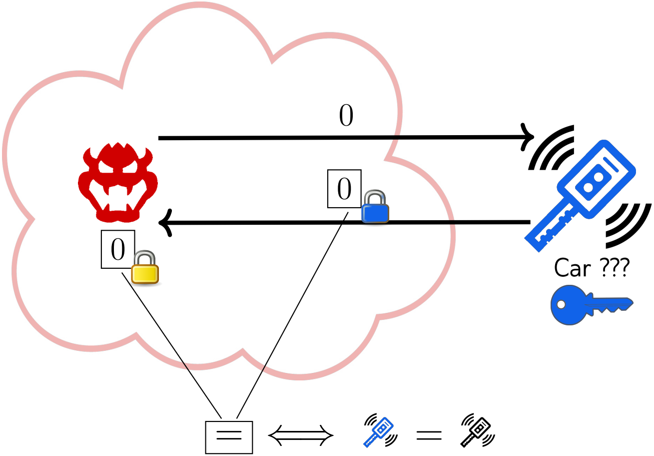
Attack:  tracks  (attack on unlinkability)



Security Protocols

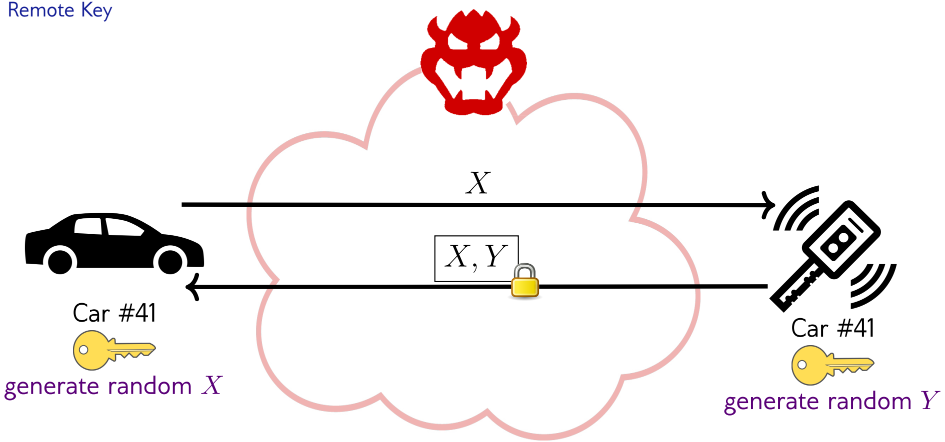
Remote Key

Attack: tracks (attack on unlinkability)



Security Protocols

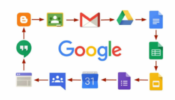
Remote Key



End of attack/fix cycle ?

Secure?

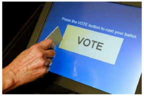
wins (BlackHat'15)



wins (FMSE'08)

wins (CSF'10)

wins (CCS'10)



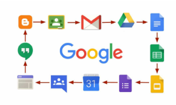
wins (CSF'11)

Extremely complex setting

- unsecure network
- active attacker
- parties running concurrently

Secure?

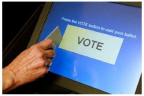
wins (BlackHat'15)



wins (FMSE'08)

wins (CSF'10)

wins (CCS'10)



wins (CSF'11)

Extremely complex setting

- unsecure network
- active attacker
- parties running concurrently

Formal methods

- mathematical & exhaustive analysis
- formal guarantees
- automated & mechanised

Automated Verification of Privacy in Security Protocols

Symbolic Model

Cryptographic primitives

- assumed perfect
- primitives modelled as function symbols & equational theory
- *e.g.* ,  $\longmapsto$ $\mathsf{enc}(\cdot,\cdot), \mathsf{dec}(\cdot,\cdot)$ & $\mathsf{dec}(\mathsf{enc}(m,k),k) = m$

Symbolic Model

Cryptographic primitives

- assumed perfect
- primitives modelled as function symbols & equational theory
- *e.g.* ,  $\longmapsto$ $\mathsf{enc}(\cdot,\cdot), \mathsf{dec}(\cdot,\cdot)$ & $\mathsf{dec}(\mathsf{enc}(m,k),k) = m$

Security protocols

- in a process algebra
- each party $\longmapsto$ process

 $\longmapsto$ P $=$ $\mathsf{in}(x).$
$\mathsf{new}\ Y.$
$\mathsf{out}(\mathsf{enc}((x,Y),k))$

Symbolic Model

Cryptographic primitives

- assumed perfect
- primitives modelled as function symbols & equational theory
- *e.g.* ,  $\longmapsto$ $\mathsf{enc}(\cdot,\cdot), \mathsf{dec}(\cdot,\cdot)$ & $\mathsf{dec}(\mathsf{enc}(m,k),k) = m$

Security protocols

- in a process algebra
- each party $\longmapsto$ process

 $\longmapsto$ $P = \mathsf{in}(x).\ \mathsf{new}\ Y.\ \mathsf{out}(\mathsf{enc}((x,Y),k))$

Attacker

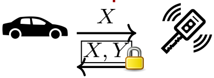
-  = network (worst case scenario)
- eavesdrop: he learns all protocol outputs
- injections: he chooses all protocol inputs

Benefit: high level of automation !

Symbolic Model

Big Picture

Protocol's specification



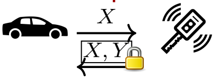
Security goal

e.g. cannot steal

Symbolic Model

Big Picture

Protocol's specification $\longmapsto$ Protocol's model



$$P = \mathsf{in}(x).$$
$$\mathsf{new}\ Y.$$
$$\mathsf{out}(\mathsf{enc}((x, Y), k))$$
$$P = \ldots$$

Security goal $\longmapsto$ Unreachability of bad states

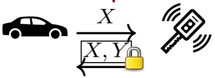
e.g. cannot steal

e.g. States(knows k)

Symbolic Model

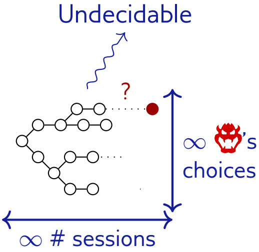
Big Picture

Protocol's specification $\longmapsto$ **Protocol's model**



$$\begin{aligned} P &= \mathsf{in}(x). \\ &\ \mathsf{new}\ Y. \\ &\ \mathsf{out}(\mathsf{enc}((x,Y),k)) \\ P &= \ldots \end{aligned}$$

Reachability in a transition system



Security goal $\longmapsto$ **Unreachability of bad states**

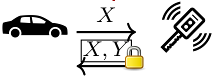
e.g. cannot steal

e.g. States(knows k)

Symbolic Model

Big Picture

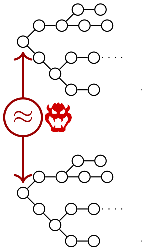
Protocol's specification $\longmapsto$ Protocol's model



$P = \mathsf{in}(x).$
$\quad \mathsf{new}\ Y.$
$\quad \mathsf{out}(\mathsf{enc}((x, Y), k))$

$P = \ldots$

Undecidable



≈ between transition systems

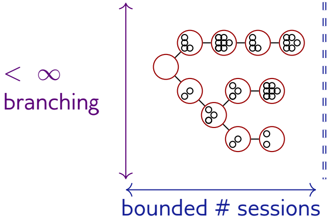
Privacy goal $\longmapsto$ ≈ between scenarios

e.g. cannot track

e.g. , ≈ ,

Two Approaches for Verifying ≈ Automatically

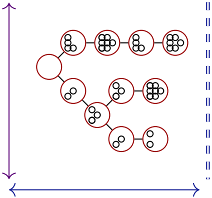
Decision for $< \infty$ sessions



- bound the number of sessions
- symbolic semantics
  $\rightsquigarrow$ finite description of 
- exhaustive exploration of symbolic executions
- Tools: Apte, Akiss, Spec

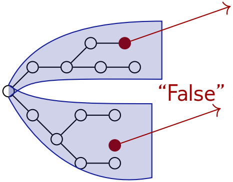
Two Approaches for Verifying ≈ Automatically

Decision for $< \infty$ sessions



- bound the number of sessions
- symbolic semantics
  $\rightsquigarrow$ finite description of
- exhaustive exploration of symbolic executions
- Tools: Apte, Akiss, Spec

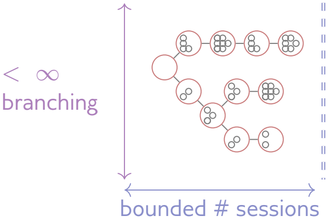
Semi-decision for ∞ sessions



- over-approximations of & semantics
- strong form of ≈
  (*i.e.* diff-equivalence)
- Tools: ProVerif, Tamarin, Maude-NPA

Limitation of Decision Procedures

Decision for $< \infty$ sessions



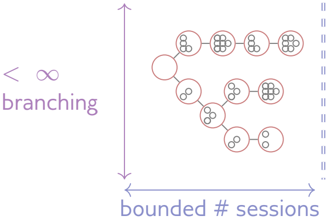
- bound the number of sessions
- symbolic semantics $\rightsquigarrow$ finite description of
- exhaustive exploration of symbolic executions
- Tools: Apte, Akiss, Spec

- States Space Explosion (concurrency)
- $\rightsquigarrow$ scales very badly

PA: 1 sess. $\mapsto$ 1sec. vs. 3 sess. $\mapsto$ >2days

Limitation of Decision Procedures

Decision for $< \infty$ sessions



- bound the number of sessions
- symbolic semantics $\rightsquigarrow$ finite description of
- exhaustive exploration of symbolic executions
- Tools: Apte, Akiss, Spec

Contributions

- Partial Order Reduction techniques
- adequate for security & $\approx$
- integration (proof & implem)

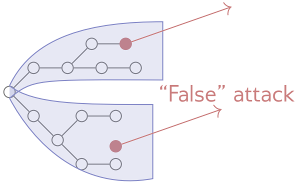
$\mapsto$ POST'14 & CONCUR'15

- States Space Explosion (concurrency)
- $\rightsquigarrow$ scales very badly

PA: 1 sess. $\mapsto$ 1sec. vs. 3 sess. $\mapsto$ >2days

Limitation of Semi-decision Procedures

Semi-decision for ∞ sessions



- over-approximations of  & semantics
- strong form of ≈ (*i.e.* diff-equivalence)
- Tools: ProVerif, Tamarin, Maude-NPA

- Serious Precision Issue (privacy)
- ⇝ systematic false attacks (unlinkability) *e.g.* e-Passport, RFID protocols, ...

Limitation of Semi-decision Procedures

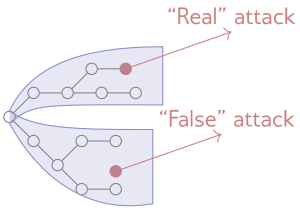
Contributions

- **Privacy via Sufficient Conditions**
- 2 conditions $\Rightarrow$ unlinkability & ano.
- automatic verification (our tool UKano)
- new proofs/attacks on real-life protocols

$\mapsto$ S&P'16

- **Serious Precision Issue** (privacy)
- $\rightsquigarrow$ systematic false attacks (unlinkability)
  e.g. e-Passport, RFID protocols, ...

Semi-decision for ∞ sessions



- over-approximations of 👹 & semantics
- **strong form of $\approx$** (*i.e.* diff-equivalence)
- Tools: ProVerif, Tamarin, Maude-NPA

Applied π-Calculus

Model of messages: Term algebra

- Function symbols $\quad \mathsf{enc}(\cdot,\cdot),\ \mathsf{dec}(\cdot,\cdot)$
- Equational theory $=_{\mathsf{E}}$ $\quad \mathsf{dec}(\mathsf{enc}(x,y),y) =_{\mathsf{E}} x$

Model of protocols: Process calculus

- Process:

$$
\begin{array}{rcll}
P, Q & := & 0 & \text{null} \\
& \mid & \mathsf{in}(c,x).P & \text{input} \\
& \mid & \mathsf{out}(c,m).P & \text{output} \\
& \mid & \texttt{if } Test \texttt{ then } P \texttt{ else } Q & \text{conditional} \\
& \mid & P \mid Q & \text{parallel} \\
& \mid & !\, P & \text{replication} \\
& \mid & \mathsf{new}\ X.P & \text{creation of name}
\end{array}
$$

Applied π-Calculus

Model of messages: Term algebra

- Function symbols — $\mathsf{enc}(\cdot,\cdot),\ \mathsf{dec}(\cdot,\cdot)$
- Equational theory $=_\mathsf{E}$ — $\mathsf{dec}(\mathsf{enc}(x,y),y) =_\mathsf{E} x$

Model of protocols: Process calculus

- Process:

$$
\begin{array}{lcll}
P, Q & := & 0 & \text{null} \\
 & \mid & \mathsf{in}(c,x).P & \text{input} \\
 & \mid & \mathsf{out}(c,m).P & \text{output} \\
 & \mid & \texttt{if } Test \texttt{ then } P \texttt{ else } Q & \text{conditional} \\
 & \mid & P \mid Q & \text{parallel} \\
 & \mid & !\,P & \text{replication} \\
 & \mid & \mathsf{new}\ X.P & \text{creation of name}
\end{array}
$$

- Frame (ϕ): the set of messages revealed to (’s knowledge)

$$\phi = \{\underbrace{w_1}_{\text{handle}} \mapsto \underbrace{\mathsf{enc}(m,k)}_{\text{out. message}}, w_2 \mapsto k\}$$

- Configuration: $A = (\mathcal{P}; \phi)$

Applied-π - Semantics

- Recipes: terms built using handles

e.g. $\begin{array}{l} R = \mathsf{dec}(w_1, w_2) \\ R\phi =_{\mathsf{E}} m \end{array}$ for $\phi = \{w_1 \mapsto \mathsf{enc}(m, k), w_2 \mapsto k\}$

"How  builds messages from its knowledge"

Applied-π - Semantics

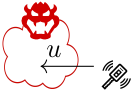
- Recipes: terms built using handles

  e.g. $R = \mathsf{dec}(w_1, w_2)$, $R\phi =_{\mathsf{E}} m$ for $\phi = \{w_1 \mapsto \mathsf{enc}(m,k), w_2 \mapsto k\}$

  "How builds messages from its knowledge"

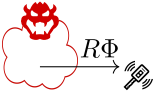
- Protocol's output:

$$(\{\mathsf{out}(c,u).P\} \cup \mathcal{P}; \phi) \xrightarrow{\mathsf{out}(c,w)} (\{P\} \cup \mathcal{P}; \phi \cup \{w \mapsto u\}) \quad \text{if } w \text{ fresh}$$



- Protocol's input:

$$(\{\mathsf{in}(c,x).P\} \cup \mathcal{P}; \phi) \xrightarrow{\mathsf{in}(c,R)} (\{P\{x \mapsto R\phi\}\} \cup \mathcal{P}; \phi)$$



- $+$ expected rules for conditional (modulo $=_{\mathsf{E}}$) & others

$\rightsquigarrow$ controls all the network

Applied-π - Trace Equivalence

Static Equivalence (intuitively)

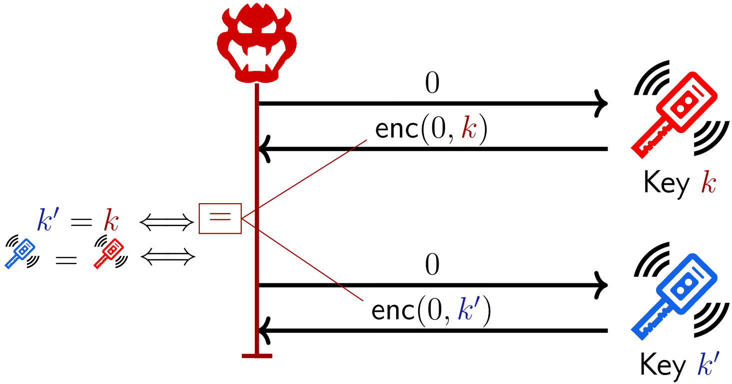
$\Phi \sim \Psi$ when

- $\mathrm{dom}(\Phi) = \mathrm{dom}(\Psi)$ and
- for all tests, it holds on $\Phi \iff$ it holds on Ψ (modulo $=_\mathsf{E}$)

Trace Equivalence

$A \approx B$: for any $A \xrightarrow{\mathsf{t}} A'$ there exists $B \xrightarrow{\mathsf{t}} B'$ such that $\Phi(A') \sim \Phi(B')$ (and the converse).

Trace Equivalence: Example



$$\boxed{P \mid P \not\approx P \mid P}$$

$$P = \mathsf{in}(c,x).\mathsf{out}(c,\mathsf{enc}(x,k))$$

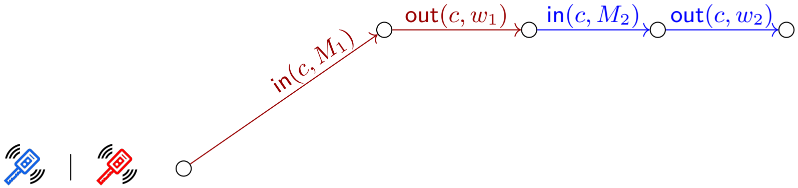
$$P = \mathsf{in}(c,x).\mathsf{out}(c,\mathsf{enc}(x,k'))$$

$$(P \mid P\, ; \emptyset) \xrightarrow{\mathsf{in}(c,0).\mathsf{out}(c,w_1).\mathsf{in}(c,0).\mathsf{out}(c,w_2)} (\emptyset; \{w_1 \mapsto \mathsf{enc}(0,k), w_2 \mapsto \mathsf{enc}(0,k)\})$$

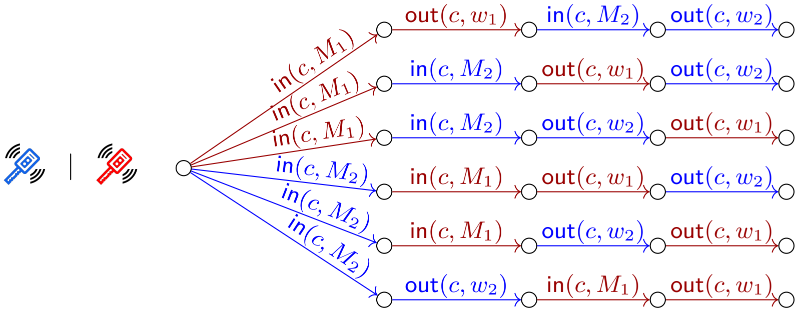
$$(P \mid P\, ; \emptyset) \xrightarrow{\mathsf{in}(c,0).\mathsf{out}(c,w_1).\mathsf{in}(c,0).\mathsf{out}(c,w_2)} (\emptyset; \{w_1 \mapsto \mathsf{enc}(0,k), w_2 \mapsto \mathsf{enc}(0,k')\})$$

$$\not\sim$$

States Space Explosion Problem



States Space Explosion Problem

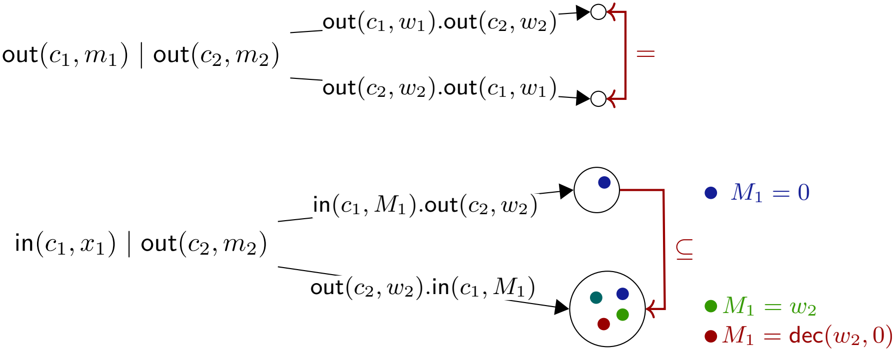


Example: Private Authentication protocol — 2 parties, 4 actions

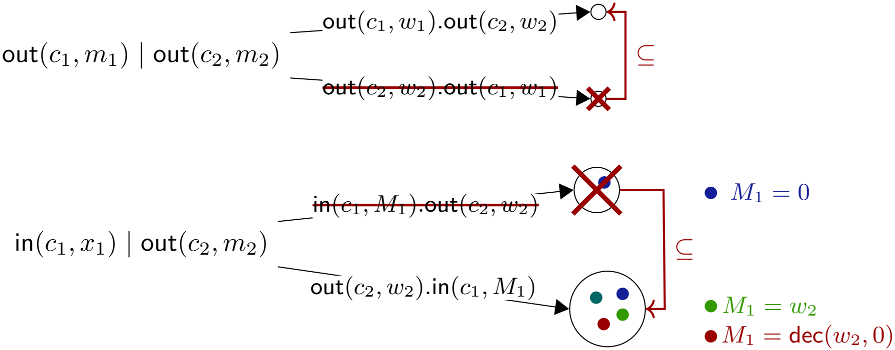
Verification of anonymity: (with APTE)

- 1 session $\mapsto$ 1 second
- 2 sessions $\mapsto$ 1 hour
- 3 sessions $\mapsto$ >2 days

1^{st} Type of Redundancies & Compression



1st Type of Redundancies & Compression



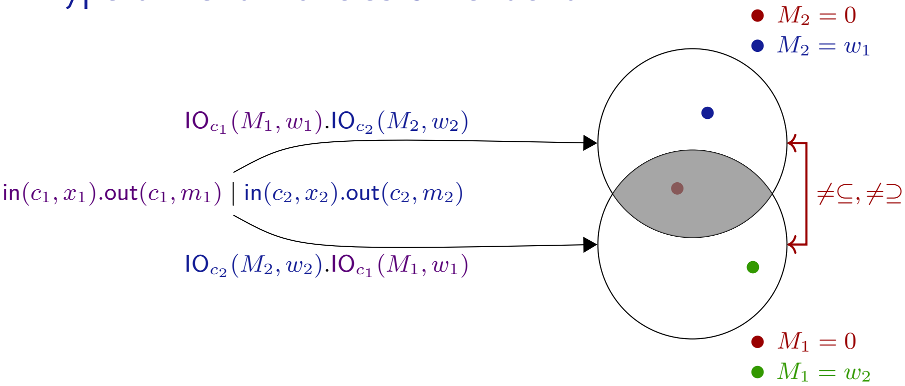
Goal: do not explore states ✗

+ generic class

Compressed semantics $\rightarrow_c$

- exploration strategy based on nature of available actions indep. from data
- actions are executed in a row $\rightsquigarrow$ blocks (big steps)

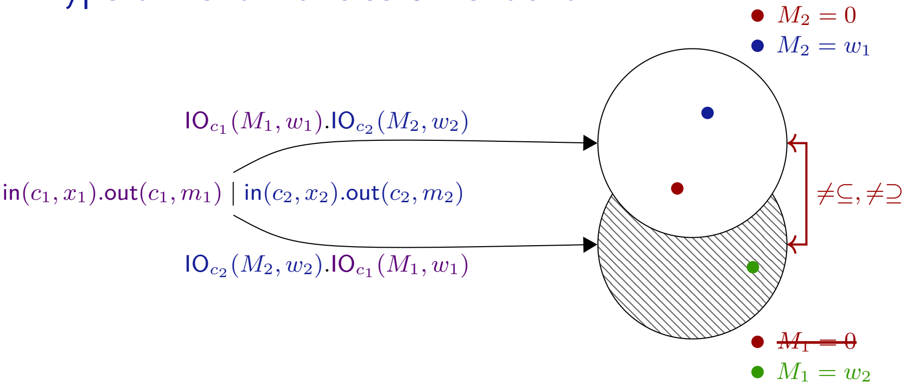
2^{nd} Type of Redundancies & Reduction



Goal: do not explore twice states in the  area

+ generic class

2^{nd} Type of Redundancies & Reduction



Goal: do not explore twice states in the area + generic class

Reduced semantics $\rightarrow_r$

- refines further $\rightarrow_c$ by analyzing data
- exploration strategy relying on data dependencies "M_1 needs w_2": $M_1 \ltimes w_2$

Soundness & Completeness

Reachability: soundness & completeness of $\to_r$ / $\to_c$ w.r.t. $\to$

same states are reachable

Equivalence is more involved and requires additional assumption

Action-determinacy

A is *action-deterministic* if: two reachable actions in parallel must be $\neq$

Attacker knows to/from whom he is sending/receiving messages.

Theorem: $\approx_r = \approx_c = \approx$

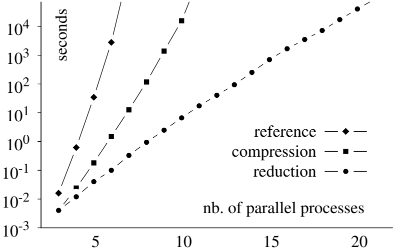
Let A and B be two action-deterministic configurations.

$$A \approx_r B \iff A \approx_c B \iff A \approx B$$

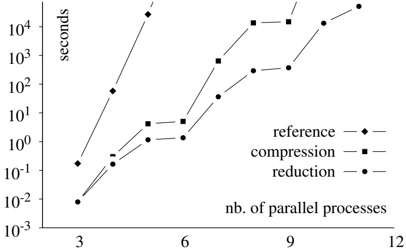
Integration, Implementation & Practical Impact

- Integration in symbolic & constraints solving setting
- Proof of soundness of integration in APTE
- Fully implemented in the distributed version of APTE `github.com/APTE`

Selection of benchmarks:



Toy example



Wide Mouthed Frog

- New scenarios & protocols can be analysed

Unlinkability

[ISO/IEC 15408] *Ensuring that a user may make multiple uses of a service or resource without others being able to link these uses together.*

1 user
2 sessions

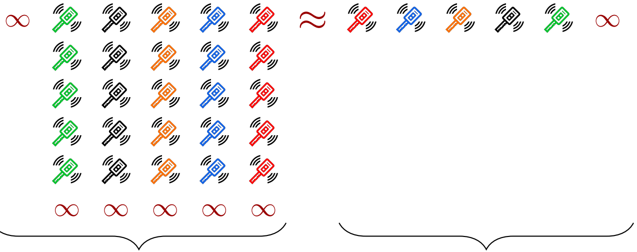
Id 1 Id 1 Id 2

Unlinkability

[ISO/IEC 15408] *Ensuring that a user may make multiple uses of a service or resource without others being able to link these uses together.*

∞ users
∞ sessions

Id 5 Id 4 Id 3 Id 2 Id 1   Id 1 Id 2 Id 3 Id 4 Id 5

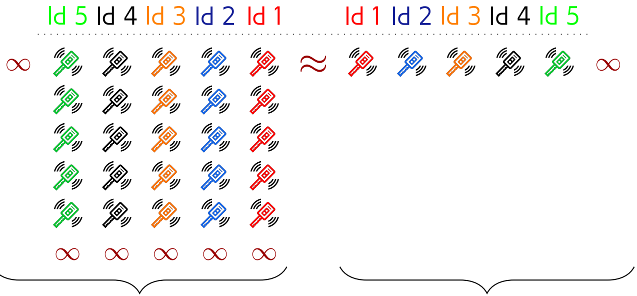


!new Id. !new Sess. P $\approx^?$!new Id. new Sess. P

Unlinkability

[ISO/IEC 15408] *Ensuring that a user may make multiple uses of a service or resource without others being able to link these uses together.*

∞ users
∞ sessions

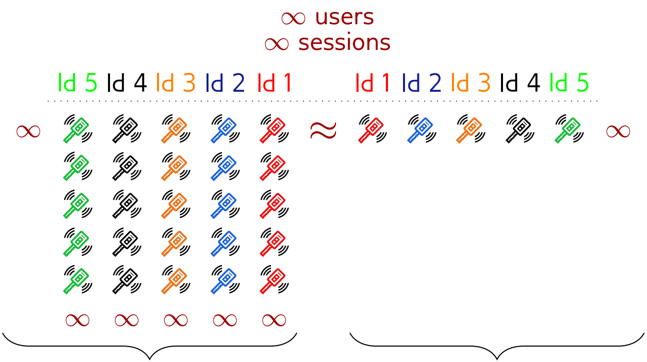


$$\mathcal{M} = !\text{new Id}.\ !\text{new Sess}.\ (P \mid P) \approx^{?}\ !\text{new Id}.\ \text{new Sess}.\ (P \mid P) = \mathcal{S}$$

Strong Unlinkability [Arapinis, Chothia, Ritter, Ryan CSF'10]

Unlinkability

[ISO/IEC 15408] *Ensuring that a user may make multiple uses of a service or resource without others being able to link these uses together.*



$$\mathcal{M} = \text{!new Id. !new Sess. } (P \mid P) \approx^{?} \text{ !new Id. new Sess. } (P \mid P) = \mathcal{S}$$

never diff-equivalent
(false attacks)

Contributions

≈ between simpler systems

with precise verification

Reachability property

$\Longrightarrow$ Unlinkability & Anonymity

Theory

- 2 conditions implying unlinkability and anonymity
- for a large class of 2-party protocols for any crypto. primitives
- each condition is fundamentally simpler & captures key ingredient

Practice

- our conditions can be checked automatically using encodings
- we provide tool support for that: UKano

Applications

- new proofs & attacks on real-life protocols *e.g.* e-passport

Class of Protocols

2-party Protocols

- Intuitively, a party P is a process of the form:

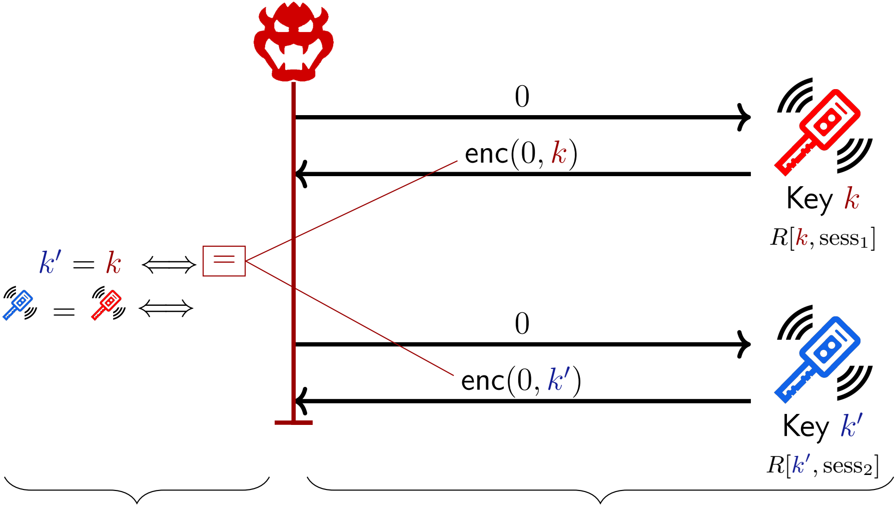
$$
\begin{aligned}
P &::= 0 \mid \mathsf{in}(c,x).\ \texttt{if}\ \mathit{Test}\ \texttt{then}\ \mathsf{out}(c,u).P\ \texttt{else}\ P_{\text{else}} \\
P_{\text{else}} &::= 0 \mid \mathsf{out}(c',u')
\end{aligned}
$$

- Two parties: I (initiator) & R (responder)

Example:

- $R = P = \mathsf{in}(c,x).\mathsf{out}(c,\mathsf{enc}(x,k))$
- $I = P = \mathsf{out}(c,X).\mathsf{in}(c,z).\ \texttt{if}\ \mathsf{dec}(z,k) = X\ \texttt{then}\ \mathsf{out}(c,\mathsf{open})$
- $\mathcal{M} = !\mathsf{new}\ k.\ !\mathsf{new}\ X.(I \mid R)$
- $\mathcal{S} = !\mathsf{new}\ k.\ \mathsf{new}\ X.(I \mid R)$

1st Class: Leaks through Relations over Messages



For some behaviour...

... ∃ relation over messages that leaks info about identities.

1^{st} Class: Leaks through Relations over Messages

Problem

For some 's behaviours, relations over messages leak info about involved agents.

Ideas of our condition preventing such attacks

- Avoid rel. $\mathcal{R}$: $\mathcal{R}$ holds $\iff$ specific mapping [sessions $\mapsto$ identities]

  e.g. $w_1 =_{\mathsf{E}} w_2 \iff [\mathrm{sess}_1 \mapsto \mathrm{id}, \mathrm{sess}_2 \mapsto \mathrm{id}]$
- Introduce Ideal(Φ): frame one obtains for [session $\mapsto$ fresh identity]

1^{st} Condition: Frame Opacity

For all $\mathcal{M} \xrightarrow{t} (\mathcal{P}; \Phi)$, we have that $\Phi \sim$ Ideal(Φ).

Example:

$$\begin{array}{rcl} \Phi & = & \{w_1 \mapsto \mathsf{enc}(0,k), \quad w_2 \mapsto \mathsf{enc}(0,k)\} \\ \mathsf{Ideal}(\Phi) & = & \{w \mapsto \mathsf{enc}(0,k_1), \quad w_2 \mapsto \mathsf{enc}(0,k_2)\} \end{array}$$

2nd Class: Leaks through Conditionals' Outcomes

So, let's introduce two modifications ...

Key k
new X

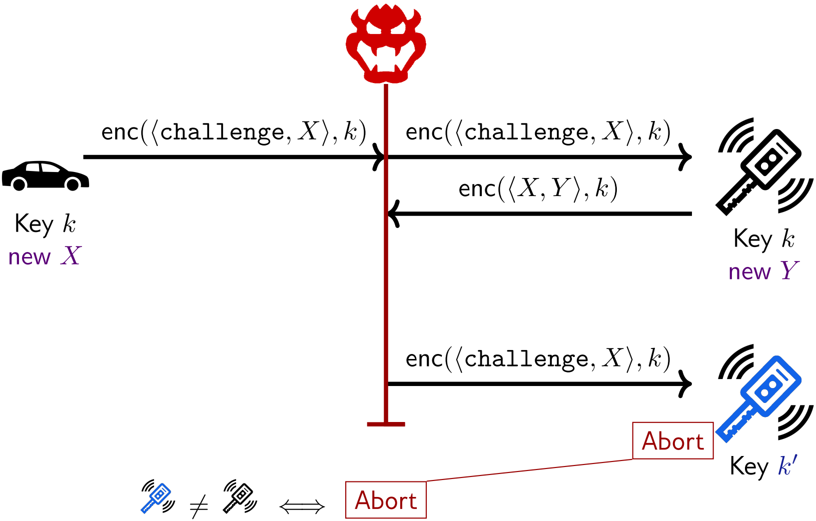
$\mathsf{enc}(\langle \mathtt{challenge}, X\rangle, k)$ $\longrightarrow$

$\longleftarrow$ $\mathsf{enc}(\langle X, Y\rangle, k)$

Key k
new Y

2nd Class: Leaks through Conditionals' Outcomes

Attack: tracks



2nd Class: Leaks through Conditionals' Outcomes

Problem

For some 's behaviours, conditionals' outcomes leak info about involved agents.

Ideas of our condition preventing such attacks

- Expected: does not interfere $\Rightarrow$ conditionals ✓
- Problems when: did interfere $\Rightarrow$ conditionals ✓/✗ binary info about agents
- Require: conditional ✓ $\Longleftrightarrow$ did not interfere

2nd Condition: Well-Authentication

For any execution of $\mathcal{M}$, if an agent $I(\mathrm{id}, \mathrm{sess})$ successfully passes a test, he must be interacting honestly with some unique $R(\mathrm{id}, \mathrm{sess}')$.

Main Result

Theorem

For any protocol in our class, for any term algebra:

$$\left.\begin{array}{c}\textit{Frame Opacity}\\ \&\\ \textit{Well-Authentication}\end{array}\right\} \Rightarrow \left\{\begin{array}{c}\textit{Unlinkability}\\ \&\\ \textit{Anonymity}\end{array}\right.$$

Idea of the proof $\mathcal{M} \xrightarrow{\mathsf{t}} (\mathcal{P};\Phi) \rightsquigarrow \mathcal{S} \xrightarrow{\mathsf{t}} (\mathcal{Q};\Psi)$ with $\Phi \sim \Psi$

- Seen as exchanges between threads: [id,session]
- Rename ids to pairwise distinct ids (keeping "connected" threads together)

Goal: (i) still executable & (ii) frames $\sim$

(i) "Have honest interactions" stable by our renaming $+$ Well-Authentication

(ii) Stability of Ideal$(\cdot)$ by renamings $+$ Frame Opacity

Practical Impact

Mechanisation & UKano

Benefit: each condition is fundamentally simpler

- Unlinkability: $\forall.\exists.\sim$
- Frame Opacity: $\forall.\sim$
- Well-Authentication: $\forall.\mathrm{Reach}$

Both conditions can be automatically verified using ProVerif & encodings

- **Well-Authentication:**
  - just reachability properties
- **Frame Opacity:**
  - checkable with good precision via diff-equivalence

Tool: UKano (built on top of ProVerif)

Automatically checks our conditions

Practical Impact

Case Studies: verification of unlinkability (UK)

RFID protocols	FO	WA	UK
Feldhofer	✓	✓	**safe**
Hash-Lock	✓	✓	**safe**
LAK (stateless)	–	×	
Fixed LAK	✓	✓	**safe**

[*]	FO	WA	UK
DAA sign	✓	✓	**safe**
DAA join	✓	✓	**safe**
abcdh (irma)	✓	✓	**safe**

e-passport	FO	WA	UK
BAC	✓	✓	**safe**
BAC/PA/AA	✓	✓	**safe**
PACE (fallible dec)	–	×	
PACE (missing test)	–	×	
PACE	–	×	
PACE with tags	✓	✓	**safe**

- Our conditions are tight
- Established new proofs and found new attacks using UKano
- Was impossible before: systematic false attacks

except for [*]

Practical Impact

Case Studies: verification of unlinkability (UK)

RFID protocols	FO	WA	UK
Feldhofer	✓	✓	**safe**
Hash-Lock	✓	✓	**safe**
LAK (stateless)	—	✗	
Fixed LAK	✓	✓	**safe**

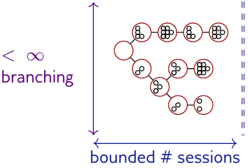
[*]	FO	WA	UK
DAA sign	✓	✓	**safe**
DAA join	✓	✓	**safe**
abcdh (irma)	✓	✓	**safe**

e-passport	FO	WA	UK
BAC	✓	✓	**safe**
BAC/PA/AA	✓	✓	**safe**
PACE (fallible dec)	—	✗	
PACE (missing test)	—	✗	
PACE	—	✗	
PACE with tags	✓	✓	**safe**

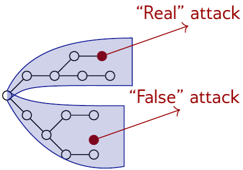
- Our conditions are tight
- Established new proofs and found new attacks using UKano
- Was impossible before: systematic false attacks   except for [*]

Summary

Decision for $< \infty$ sessions



Semi-decision for ∞ sessions



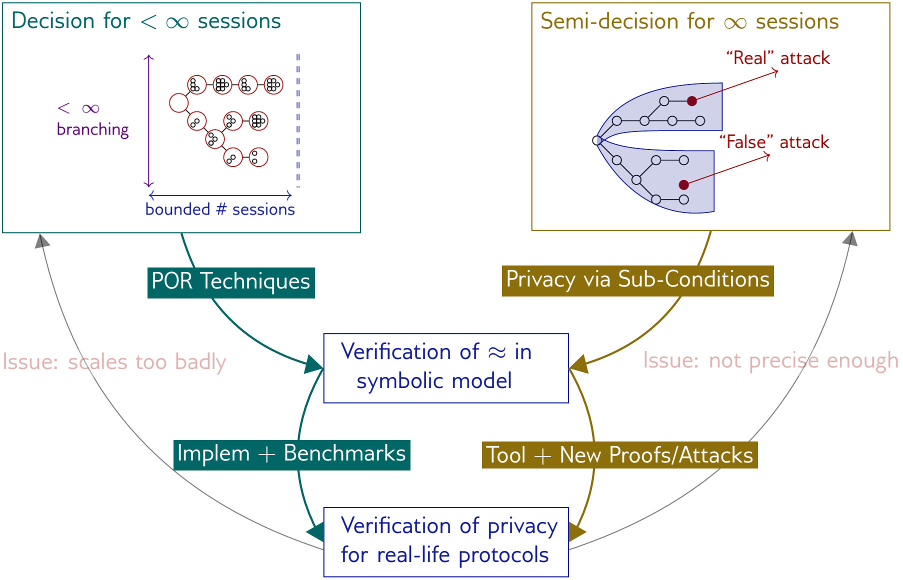
Issue: scales too badly

Verification of $\approx$ in symbolic model

Issue: not precise enough

Verification of privacy for real-life protocols

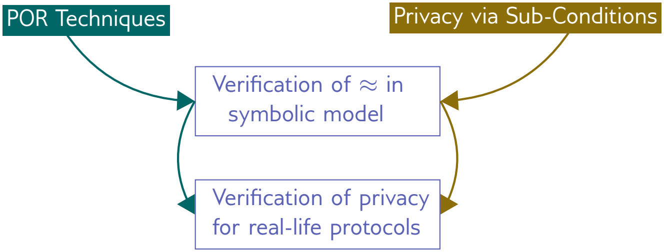
Summary



Decision for $< \infty$ sessions
$< \infty$ branching
bounded # sessions
Semi-decision for ∞ sessions
"Real" attack
"False" attack
POR Techniques
Privacy via Sub-Conditions
Issue: scales too badly
Verification of $\approx$ in symbolic model
Issue: not precise enough
Implem + Benchmarks
Tool + New Proofs/Attacks
Verification of privacy for real-life protocols

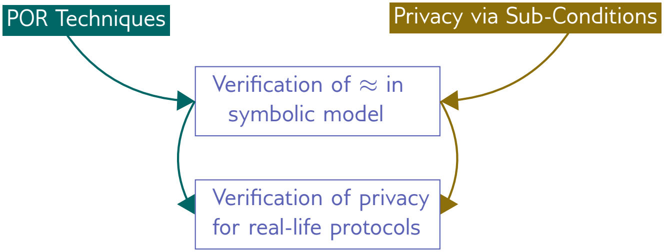
Future Work

- Drop action-determinacy assumption
- POR for backward search (*e.g.* Tamarin)



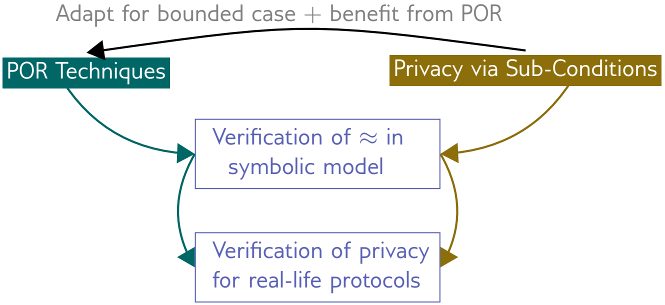
Future Work

- Drop action-determinacy assumption
- POR for backward search (*e.g.* Tamarin)
- Extend the class: stateful & >2 parties
- Verification of FO via reachability:
  UK & ANO. $\mapsto$ pure reachability



Future Work

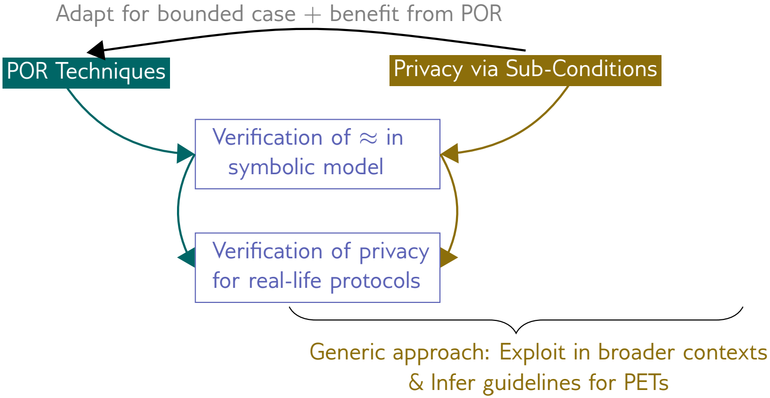
- Drop action-determinacy assumption
- POR for backward search (*e.g.* Tamarin)
- Extend the class: stateful & $>$ 2 parties
- Verification of FO via reachability:
  UK & ANO. $\mapsto$ pure reachability



Future Work

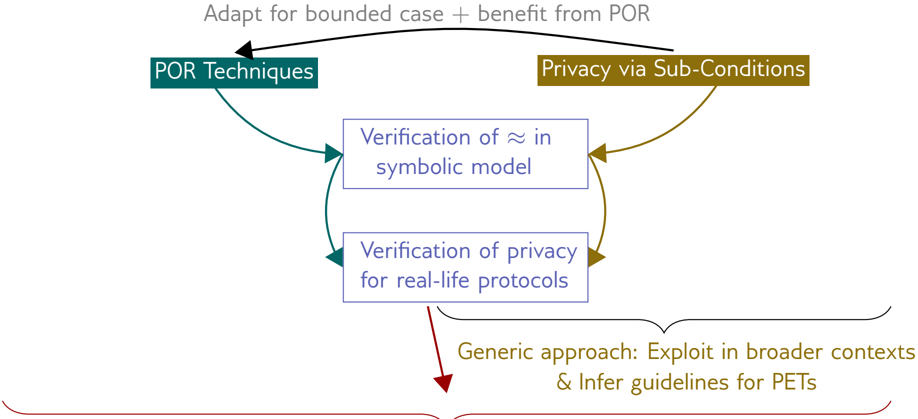
- Drop action-determinacy assumption
- POR for backward search (*e.g.* Tamarin)

- Extend the class: stateful & >2 parties
- Verification of FO via reachability:
  UK & ANO. $\mapsto$ pure reachability



Future Work

- Drop action-determinacy assumption
- POR for backward search (*e.g.* Tamarin)
- Extend the class: stateful & > 2 parties
- Verification of FO via reachability:
  UK & ANO. $\mapsto$ pure reachability



Ready to guide analysis/design/standardisation ?

Backup Slides

Compressed Strategy

Compressed semantics $\rightarrow_c$

- Polarities: Negative: out$().P, (P_1 \mid P_2), 0$ & Positive: in$().P$
- Negative: explored greedily, in a given order *e.g.* $c_1 < c_2$
- Positive: explored only when $\nexists$ Negative,
  - chooses one and put it under focus
  - focus is released when becomes begative

Replication: $!^{a}_{c,\overline{n}} P$ is positive but releases the focus.

Reduced Strategy

We assume an arbitrary order $\prec$ over blocks priority order.

Reduced Semantics $\rightarrow_r$

$\rightarrow_r$ explores a block b after a trace t only when:

- $\rightarrow_c$ explores $\mathrm{t}.b$ and
- $\mathrm{t} \ltimes b$.

Informally, $\mathrm{t} \ltimes b$ means:

> *there is no way to swap b towards the beginning of* t *before a block $b_0 \succ b$ (even by modifying recipes)*

Theorem: $\approx_r = \approx$

Let A and B be two action-deterministic configurations.

$$A \approx B \text{ if, and, only if, } A \approx_r B.$$

POR & Trace Equivalence

What about trace equivalence ($\approx_c$) ?

$$\text{e.g., } (\mathsf{in}(c_1, x) \quad | \quad \mathsf{out}(c_2, m)) \not\approx (\mathsf{out}(c_2, m).\mathsf{in}(c_1, x))$$

- $\rightsquigarrow$ same swaps are possible ($\equiv$ same sequential dependencies)
- Lemma: A, B action-det, $A \approx B \Rightarrow$ same sequential dependencies