A method for proving unlinkability of stateful protocols

David BAELDE, Stéphanie DELAUNE and **Solène MOREAU**

33rd IEEE Computer Security Foundations Symposium — June 23, 2020

Introduction

A protocol is a set of rules detailing how entities communicate.

Security protocols must ensure some **security properties** (authentication, anonymity, unlinkability, etc).

Even assuming **perfect cryptographic primitives**, there may be **logical attacks**, *e.g.* replay attacks, man-in-the-middle attacks.

How to have **guarantees** that the protocols we use are secure?

Let's use **formal methods**!

Introduction

A protocol is a set of rules detailing how entities communicate.

Security protocols must ensure some **security properties** (authentication, anonymity, unlinkability, etc).

Even assuming **perfect cryptographic primitives**, there may be **logical attacks**, *e.g.* replay attacks, man-in-the-middle attacks.

How to have **guarantees** that the protocols we use are secure?

Let's use **formal methods**!

Introduction

A protocol is a set of rules detailing how entities communicate.

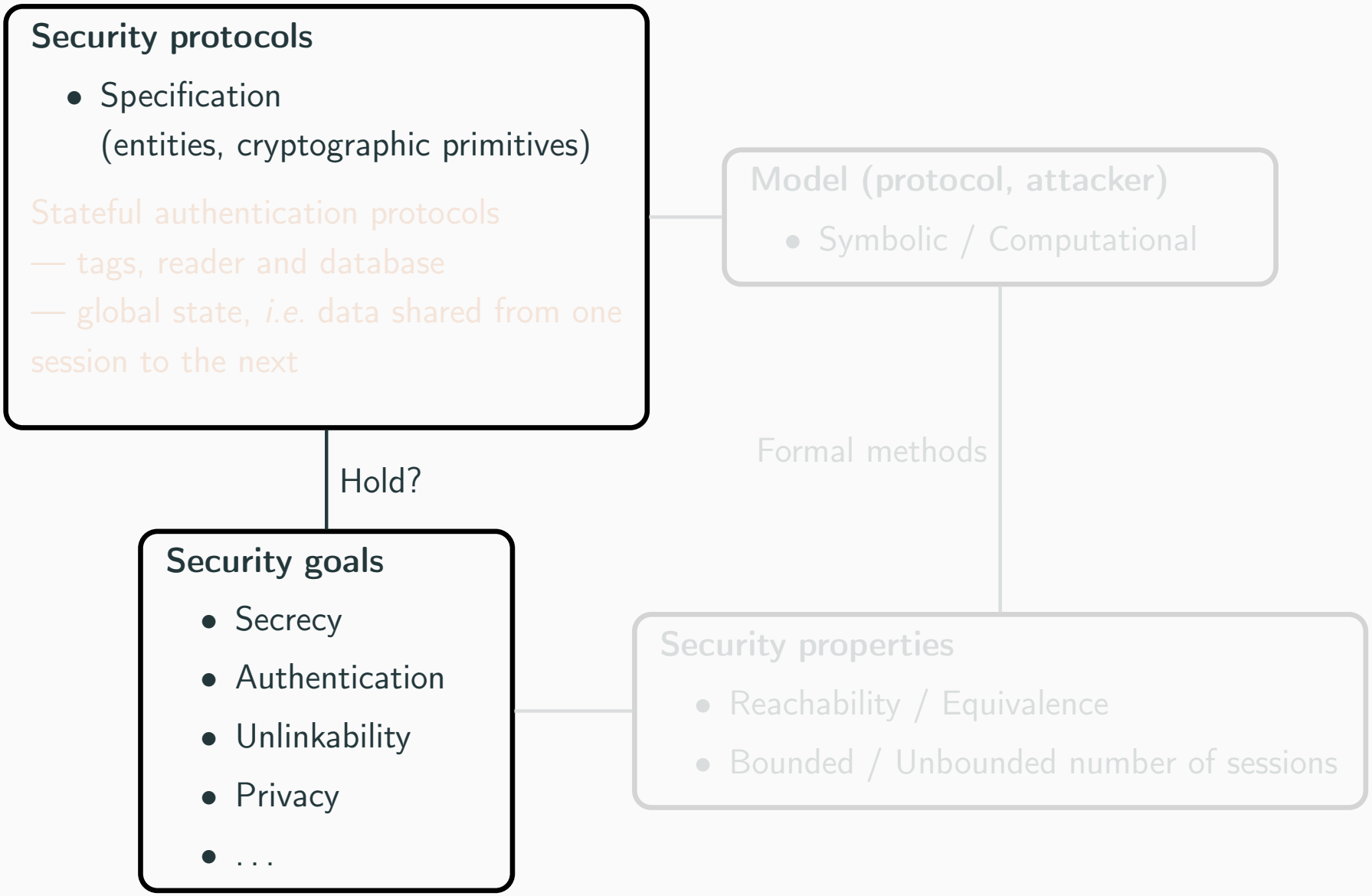
Security protocols must ensure some **security properties** (authentication, anonymity, unlinkability, etc).

Even assuming **perfect cryptographic primitives**, there may be **logical attacks**, *e.g.* replay attacks, man-in-the-middle attacks.

How to have **guarantees** that the protocols we use are secure?

Let's use formal methods!

Formal verification of security protocols



Formal verification of security protocols

Security protocols

- Specification
  (entities, cryptographic primitives)

Stateful authentication protocols
— tags, reader and database
— global state, *i.e.* data shared from one session to the next

Model (protocol, attacker)

- Symbolic / Computational

Formal methods 

Hold?

Security goals

- Secrecy
- Authentication
- Unlinkability
- Privacy
- ...

Security properties

- Reachability / Equivalence
- Bounded / Unbounded number of sessions

Formal verification of security protocols

Security protocols

- Specification
  (entities, cryptographic primitives)

Stateful authentication protocols
— tags, reader and database
— global state, *i.e.* data shared from one session to the next

Model (protocol, attacker)

- Symbolic / Computational

Formal methods 

Hold?

Security goals

- Secrecy
- Authentication
- Unlinkability
- Privacy
- ...

Security properties

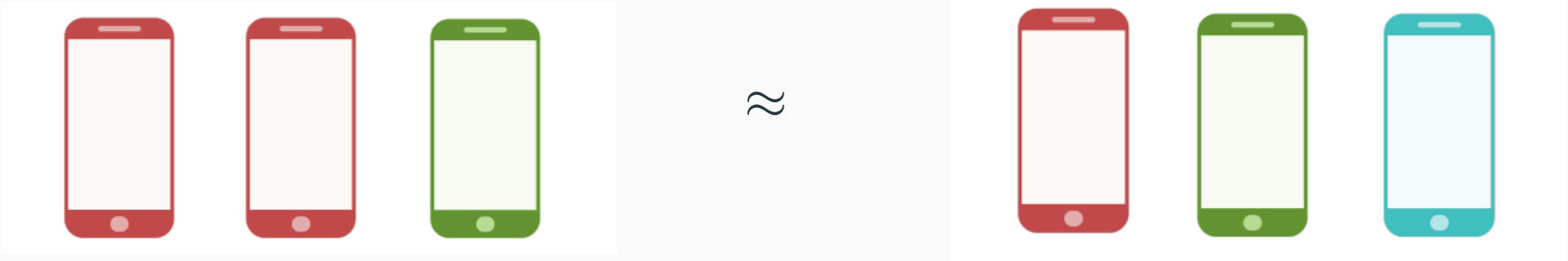
- Reachability / Equivalence
- Bounded / Unbounded number of sessions

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."
- "The **real** system is indistinguishable from an **ideal** system."

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."
- "The **real** system is indistinguishable from an **ideal** system."



A quick overview of equivalence notions

Static equivalence

$\phi \sim \psi$ an attacker cannot distinguish two frames by performing some computation that succeeds on one frame and not on the other

Trace equivalence

$P \approx_t Q$ for each trace of one process, there exists a statically equivalent trace of the other process

Different flavours of "diff-equivalence"

$P \approx Q$ all based on a strong notion of equivalence processes are trace equivalent and can be "internally" matched step by step

ProVerif, Tamarin

A quick overview of equivalence notions

Static equivalence

$\phi \sim \psi$ an attacker cannot distinguish two frames by performing some computation that succeeds on one frame and not on the other

Trace equivalence

$P \approx_t Q$ for each trace of one process, there exists a statically equivalent trace of the other process

Different flavours of "diff-equivalence"

$P \approx Q$ all based on a strong notion of equivalence processes are trace equivalent and can be "internally" matched step by step

ProVerif, Tamarin

A quick overview of equivalence notions

Static equivalence

$\phi \sim \psi$ an attacker cannot distinguish two frames by performing some computation that succeeds on one frame and not on the other

Trace equivalence

$P \approx_t Q$ for each trace of one process, there exists a statically equivalent trace of the other process

Different flavours of "diff-equivalence"

$P \approx Q$ all based on a strong notion of equivalence processes are trace equivalent and can be "internally" matched step by step

ProVerif, Tamarin

A challenging problem

- Verifying **equivalence-based** properties on protocols is **difficult**.
  - Unbounded case: undecidable in general.
  - Bounded case: tools scale badly.
- Existing tools [1] implement a procedure to check equivalence in the unbounded case:
  - based on a **strong notion of equivalence**,
  - no termination guarantee,
  - other limitations: states, XOR, etc.

[1] ProVerif, Tamarin

A challenging problem

- Verifying **equivalence-based** properties on protocols is **difficult**.
  - Unbounded case: undecidable in general.
  - Bounded case: tools scale badly.
- Existing tools [1] implement a procedure to check equivalence in the unbounded case:
  - based on a **strong notion of equivalence**,
  - no termination guarantee,
  - other limitations: states, XOR, etc.

[1] ProVerif, Tamarin

Related work

Symbolic verification of unlinkability in the unbounded case.

	Model for unlinkability	**Case studies**	**Tool**
[BCH10]	sufficient and necessary conditions for unlinkability for simple RFID protocols some attacks not caught by this model	OSK, Basic Hash	manual + ProVerif
[Ara+10]	formal framework for analysing unlinkability	French e-passport	ProVerif "false" proof
[Ara+12]		3G AKA simplified version	ProVerif
[HBD16]	sufficient conditions for verifying unlinkability of 2-party stateless protocols	Feldhofer, Hash Lock LAK, PACE BAC (e-passport)	ProVerif

Related work

Symbolic verification of unlinkability in the unbounded case.

	Model for unlinkability	**Case studies**	**Tool**
[BCH10]	sufficient and necessary conditions for unlinkability for simple RFID protocols some attacks not caught by this model	OSK, Basic Hash	manual + ProVerif
[Ara+10]	formal framework for analysing unlinkability	French e-passport	ProVerif "false" proof
[Ara+12]		3G AKA simplified version	ProVerif
[HBD16]	sufficient conditions for verifying unlinkability of 2-party stateless protocols	Feldhofer, Hash Lock LAK, PACE BAC (e-passport)	ProVerif

Outline of the talk

Model

Sufficient conditions for unlinkability

Case studies

Conclusion

Model

Syntax and semantics

Let's take an example, the **Basic Hash protocol** [JW09].

$$\begin{array}{rcll} \text{Tag} & \rightarrow & \text{Reader + DB} & : \langle n, \mathsf{h}(k,n)\rangle \\ \text{Reader + DB} & \rightarrow & \text{Tag} & : \mathsf{ok} \end{array}$$

Reader role in our **syntax**.

TEST   DB UPDATE

$$\begin{array}{rcll} R & = & \mathsf{in}(c,x). & \\ & & \text{lookup } y \text{ such that} & \mathsf{snd}(x) = \mathsf{h}(y, \mathsf{fst}(x)), \quad y' = y \\ & & & \text{in } \mathsf{out}(c, \mathsf{ok}) \\ & & & \text{else } \mathsf{out}(c, \mathsf{error}). \end{array}$$

An extract of **semantics**.

$$\begin{array}{rcl} (\mathsf{in}(c,x).\mathcal{P}_1; \phi; \mathcal{S}) & \xrightarrow{\text{input}} & (\mathcal{P}_2; \phi; \mathcal{S}) \\ (\mathsf{out}(c,m).\mathcal{P}; \phi_1; \mathcal{S}) & \xrightarrow{\text{output}} & (\mathcal{P}; \phi_2; \mathcal{S}) \\ (\text{lookup } [\ldots].\mathcal{P}; \phi; \mathcal{S}_1) & \xrightarrow{\text{lookup}} & (\mathcal{P}; \phi; \mathcal{S}_2) \end{array}$$

Syntax and semantics

Let's take an example, the **Basic Hash protocol** [JW09].

$$\begin{array}{rcll} \text{Tag} & \rightarrow & \text{Reader} + \text{DB} & : \quad \langle n, \mathsf{h}(k, n) \rangle \\ \text{Reader} + \text{DB} & \rightarrow & \text{Tag} & : \quad \text{ok} \end{array}$$

Reader role in our **syntax**.

$$\begin{array}{rcll} R & = & \mathsf{in}(c, x). & \text{TEST} \qquad\qquad\qquad\qquad\qquad\qquad \text{DB UPDATE} \\ & & \text{lookup } y \text{ such that} & \mathsf{snd}(x) = \mathsf{h}(y, \mathsf{fst}(x)), \quad y' = y \\ & & & \text{in } \mathsf{out}(c, \text{ok}) \\ & & & \text{else } \mathsf{out}(c, \text{error}). \end{array}$$

An extract of **semantics**.

$$\begin{array}{rcl} (\mathsf{in}(c, x).\mathcal{P}_1; \phi; \mathcal{S}) & \xrightarrow{\text{input}} & (\mathcal{P}_2; \phi; \mathcal{S}) \\ (\mathsf{out}(c, m).\mathcal{P}; \phi_1; \mathcal{S}) & \xrightarrow{\text{output}} & (\mathcal{P}; \phi_2; \mathcal{S}) \\ (\text{lookup } [\ldots].\mathcal{P}; \phi; \mathcal{S}_1) & \xrightarrow{\text{lookup}} & (\mathcal{P}; \phi; \mathcal{S}_2) \end{array}$$

Syntax and semantics

Let's take an example, the **Basic Hash protocol** [JW09].

$$\begin{array}{rcll} \text{Tag} & \rightarrow & \text{Reader + DB} & : \langle n, \mathsf{h}(k, n)\rangle \\ \text{Reader + DB} & \rightarrow & \text{Tag} & : \text{ok} \end{array}$$

Reader role in our **syntax**.

$$\begin{array}{rcll} R & = & \mathsf{in}(c, x). & \text{TEST} \quad\quad \text{DB UPDATE} \\ & & \text{lookup } y \text{ such that} & \mathsf{snd}(x) = \mathsf{h}(y, \mathsf{fst}(x)), \quad y' = y \\ & & & \text{in } \mathsf{out}(c, \mathsf{ok}) \\ & & & \text{else } \mathsf{out}(c, \mathsf{error}). \end{array}$$

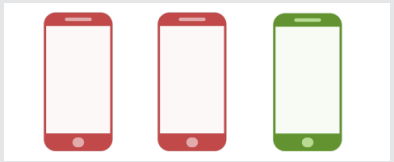
An extract of **semantics**.

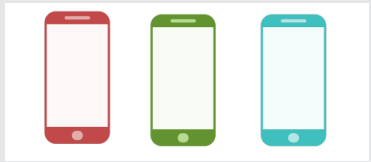
$$\begin{array}{rcl} (\mathsf{in}(c, x).\mathcal{P}_1; \phi; \mathcal{S}) & \xrightarrow{\text{input}} & (\mathcal{P}_2; \phi; \mathcal{S}) \\ (\mathsf{out}(c, m).\mathcal{P}; \phi_1; \mathcal{S}) & \xrightarrow{\text{output}} & (\mathcal{P}; \phi_2; \mathcal{S}) \\ (\text{lookup } [\ldots].\mathcal{P}; \phi; \mathcal{S}_1) & \xrightarrow{\text{lookup}} & (\mathcal{P}; \phi; \mathcal{S}_2) \end{array}$$

Finding the "good" notion of unlinkability

We use **trace equivalence** $\approx_t$.

A protocol Π ensures unlinkability if $\mathcal{M}_\Pi \approx_t \mathcal{S}_\Pi$.

$\mathcal{M}_\Pi$ $:=$  | generic readers and a database

$\mathcal{S}_\Pi$ $:=$  | generic readers and a database

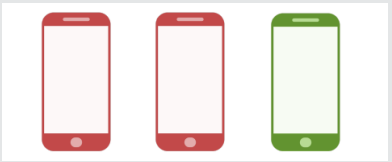
- Modeling only the tags can lead to missing attacks.
- The way readers are modeled is also important.

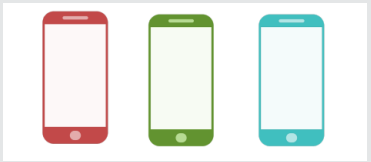
(More details in the paper. . .)

Finding the "good" notion of unlinkability

We use **trace equivalence** $\approx_t$.

A protocol Π ensures unlinkability if $\mathcal{M}_\Pi \approx_t \mathcal{S}_\Pi$.

$\mathcal{M}_\Pi$:=  | generic readers and a database

$\mathcal{S}_\Pi$:=  | generic readers and a database

- Modeling only the tags can lead to missing attacks.
- The way readers are modeled is also important.

(More details in the paper. . .)

Sufficient conditions for unlinkability

A result for stateless 2-party protocols [HBD16]

Definition

$$\begin{aligned} \mathcal{M}_\Pi &:= \; !\ \mathsf{new}\,\overline{k}.\quad (!\ \mathsf{new}\,\overline{n}_T.\mathcal{T} \mid !\ \mathsf{new}\,\overline{n}_R.\mathcal{R}) \\ \mathcal{S}_\Pi &:= \; !\ \mathsf{new}\,\overline{k}.\quad (\ \ \mathsf{new}\,\overline{n}_T.\mathcal{T} \mid \ \ \mathsf{new}\,\overline{n}_R.\mathcal{R}) \end{aligned}$$

A protocol Π ensures **unlinkability** if $\mathcal{M}_\Pi \approx_t \mathcal{S}_\Pi$.

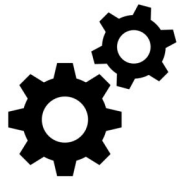
Theorem

If a protocol Π ensures both well-authentication and frame opacity, then Π ensures unlinkability.

Intuition behind the sufficient conditions [HBD16]

Well-Authentication

- To avoid leaks through outcomes of conditionals.
- "Whenever a conditional is positively evaluated, the agents involved are having so far an honest interaction."



It's a reachability property!

Frame Opacity

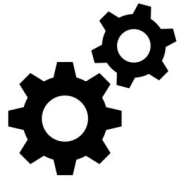
- To avoid leaks through relations over messages.
- "Any reachable frame must be statically equivalent to an idealised frame that only depends on data already observed during the execution."

Can be verified with diff-equivalence!

Intuition behind the sufficient conditions [HBD16]

Well-Authentication

- To avoid leaks through outcomes of conditionals.
- "Whenever a conditional is positively evaluated, the agents involved are having so far an honest interaction."



It's a reachability property!

Frame Opacity

- To avoid leaks through relations over messages.
- "Any reachable frame must be statically equivalent to an idealised frame that only depends on data already observed during the execution."

Can be verified with diff-equivalence!

Extending this result to stateful protocols

Definition

$$\mathcal{M}_\Pi := \left(!\ \mathsf{new}\,\overline{k}.\mathsf{init}(\mathsf{DB},\mathsf{cells}).\ \mathsf{i}\ \mathsf{new}\,\overline{n}_T.\mathcal{T}\right) \mid \left(!\ \mathsf{new}\,\overline{n}_R.\mathcal{R}\right)$$
$$\mathcal{S}_\Pi := \left(!\ \mathsf{new}\,\overline{k}.\mathsf{init}(\mathsf{DB},\mathsf{cells}).\ \ \mathsf{new}\,\overline{n}_T.\mathcal{T}\right) \mid \left(!\ \mathsf{new}\,\overline{n}_R.\mathcal{R}\right)$$

A protocol Π ensures **unlinkability** if $\mathcal{M}_\Pi \approx_t \mathcal{S}_\Pi$.

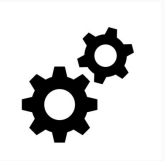
Theorem

If a protocol Π ensures well-authentication, frame opacity and **no desynchronisation** then Π ensures unlinkability.

Intuition behind no desynchronisation

No desynchronisation

- To avoid leaks through desynchronisations between agents.
- "An honest interaction between a tag and a reader cannot fail."



Also a reachability property! (But a little more tricky...)

Case studies

Case studies: checking conditions with Tamarin

	unlink.	WA	FO	ND
Basic Hash	ok	✓	✓	✓
Hash-Lock	ok	✓	✓	✓
Feldhofer	ok	✓	✓	✓
OSK (v1)	attack	✓		×
OSK (v2)	ok	✓	✓	✓
LAK (pairs)	attack	✓		×
LAK (pairs, fixed)	ok	✓	✓	✓
LAK (pairs, no update)	ok	✓	✓	✓
5G-AKA (simplified)	ok	✓	✓	✓

✓ = property holds, automatically proven with Tamarin
× = property does not hold

Conclusion

Take-aways

Improvement of existing work:

- for the verification of **unlinkability** of **stateful protocols**,
- in the **symbolic** model, for an **unbounded** number of sessions.

Our contribution

- A **formal definition of unlinkability** reflecting some keypoints:
  - importance of **how is modeled the reader**;
  - **states can introduce observables**.
- **Sufficient conditions** for verifying unlinkability, including a new **no desynchronisation** condition.
- **Case studies**: new attacks and new proofs.

Take-aways

Improvement of existing work:

- for the verification of **unlinkability** of **stateful protocols**,
- in the **symbolic** model, for an **unbounded** number of sessions.

Our contribution

- A formal definition of unlinkability reflecting some keypoints:
  - importance of **how is modeled the reader**;
  - **states can introduce observables**.
- **Sufficient conditions** for verifying unlinkability, including a new no desynchronisation condition.
- Case studies: new attacks and new proofs.

Going a step further

- Simple conditions in the theory but **not so easily checkable** in practice.
  $\Rightarrow$ Existing tools are not designed to verify our conditions.

- **Bridge the gap** between the formal model of Tamarin (based on multiset rewriting) and our theoretical model (based on applied-pi calculus).

Thank you for your attention!

Going a step further

- Simple conditions in the theory but **not so easily checkable** in practice.
  $\Rightarrow$ Existing tools are not designed to verify our conditions.

- **Bridge the gap** between the formal model of Tamarin (based on multiset rewriting) and our theoretical model (based on applied-pi calculus).

Thank you for your attention!

References

[BCH10] Brusò, Chatzikokolakis, and den Hartog.
"Formal Verification of Privacy for RFID Systems".
In: Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF'10.

[Ara+10] Arapinis et al.
"Analysing Unlinkability and Anonymity Using the Applied Pi Calculus".
In: Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF'10.

[Ara+12] Arapinis et al.
"New privacy issues in mobile telephony: fix and verification".
In: the ACM Conference on Computer and Communications Security, CCS'12.

[HBD16] Hirschi, Baelde, and Delaune.
"A Method for Verifying Privacy-Type Properties: The Unbounded Case".
In: IEEE Symposium on Security and Privacy, SP'16.

[JW09] Juels and Weis.
"Defining strong privacy for RFID".
In: ACM Trans. Inf. Syst. Secur. 13.1 (2009).