Analysing privacy-type properties using formal methods

Stéphanie Delaune

LSV, CNRS & ENS Cachan & INRIA Saclay Île-de-France, France

Wednesday, March 14th, 2012

Context: cryptographic protocols

PayPal™

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

The network is unsecure!

Communications take place over a public network like the Internet.

Context: cryptographic protocols

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

Context: cryptographic protocols

PayPal™

Cryptographic protocols

- small programs designed to secure communication (*e.g.* confidentiality, authentication, ...)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

It becomes more and more important to protect our privacy.

Example: electronic passport

$\longrightarrow$ studied in [Arapinis *et al.*, 10]

An electronic passport is a passport with an RFID tag embedded in it.

The RFID tag stores:

- the information printed on your passport,
- a JPEG copy of your picture.

Example: electronic passport

$\longrightarrow$ studied in [Arapinis *et al.*, 10]

An electronic passport is a passport with an RFID tag embedded in it.

The RFID tag stores:

- the information printed on your passport,
- a JPEG copy of your picture.

The Basic Access Control (BAC) protocol is a key establishment protocol that has been designed to also ensure unlinkability.

ISO/IEC standard 15408

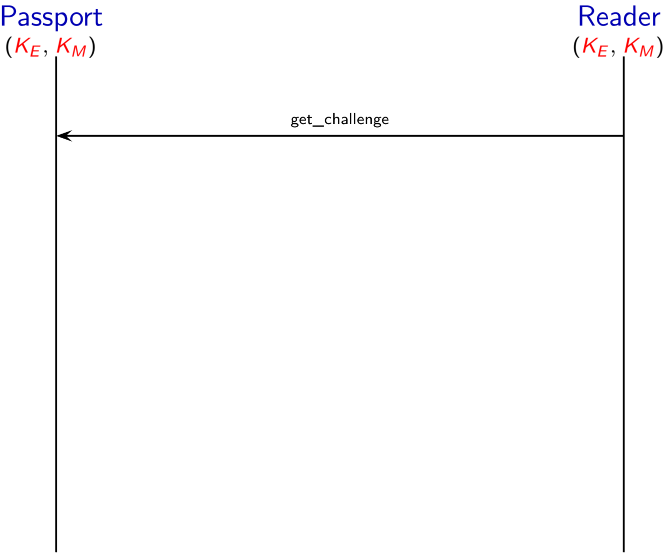
Unlinkability aims to ensure *that a user may make multiple uses of a service or resource without others being able to link these uses together.*

The electronic passport protocol

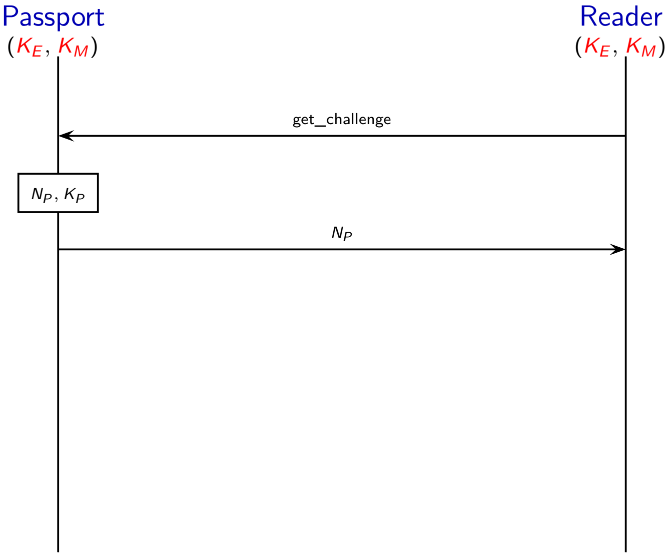
Passport
(K_E, K_M)

Reader
(K_E, K_M)

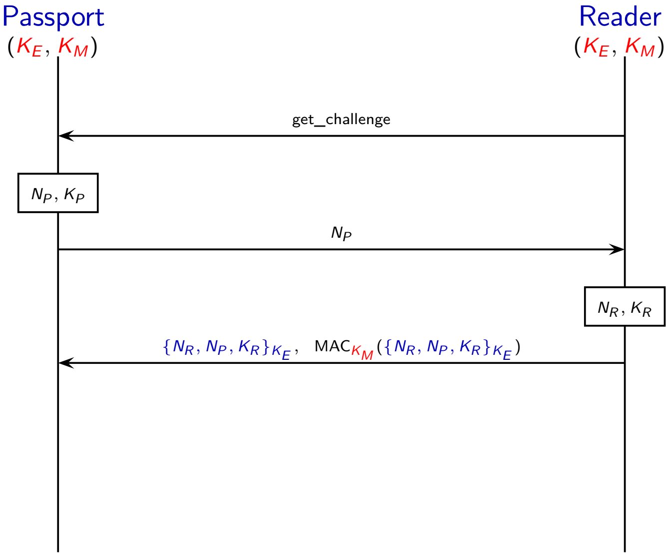
The electronic passport protocol



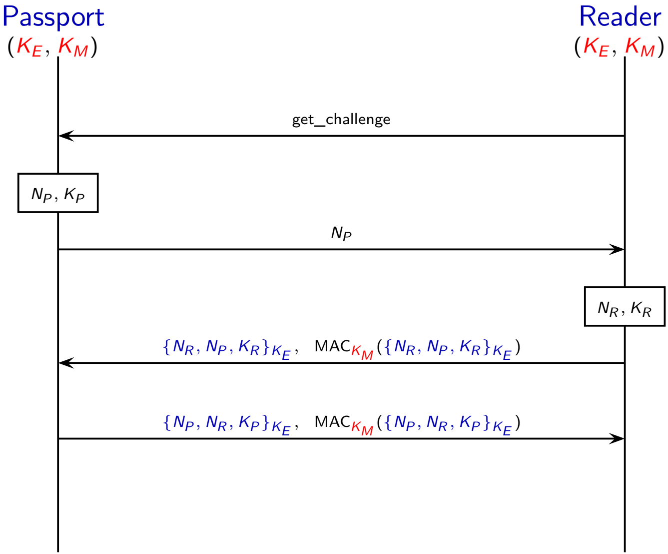
The electronic passport protocol



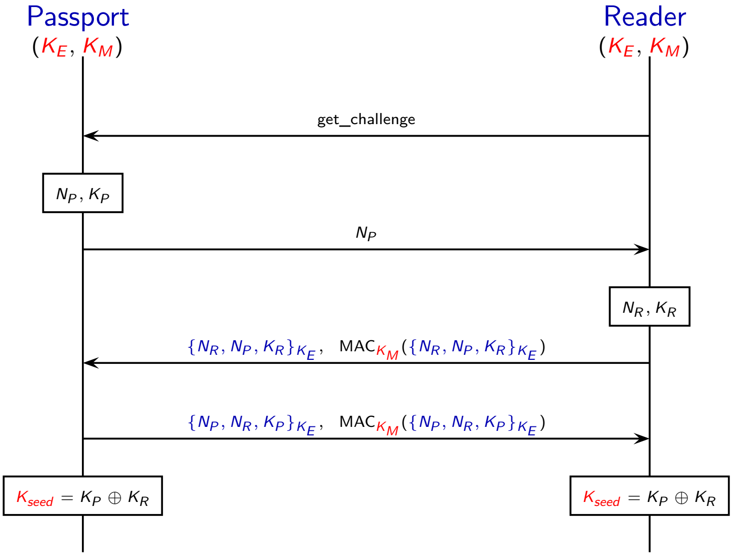
The electronic passport protocol



The electronic passport protocol



The electronic passport protocol



How cryptographic protocols can be attacked?

Some famous examples

The Serge Humpich case (1997)

He factorizes the number (320 bits) used to protect credit cards and he builds a false credit card. (the « YesCard »).

⟶ this makes it possible to withdraw a bank account that does not exist!

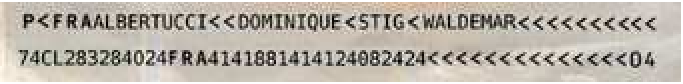
Some famous examples

The Serge Humpich case (1997)

He factorizes the number (320 bits) used to protect credit cards and he builds a false credit card. (the « YesCard »).

$\longrightarrow$ this makes it possible to withdraw a bank account that does not exist!

Attack on the Belgian e-passport (2006)



$\longrightarrow$ this makes it possible to obtain the personnal data of the user (*e.g.* the signature)

How cryptographic protocols can be attacked?

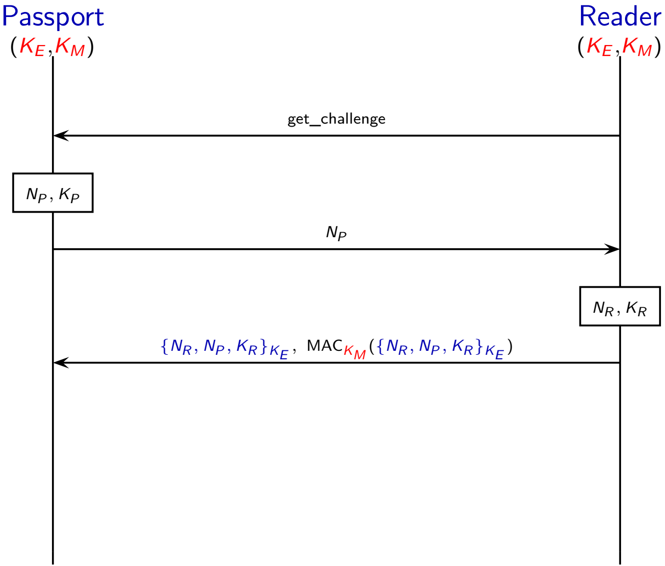
How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- are numerous,
  $\hookrightarrow$ a flaw discovered in 2010 in Single Sign On Protocols used in Google App (Avantssar european project)
- subtle and hard to detect by "eyeballing" the protocol

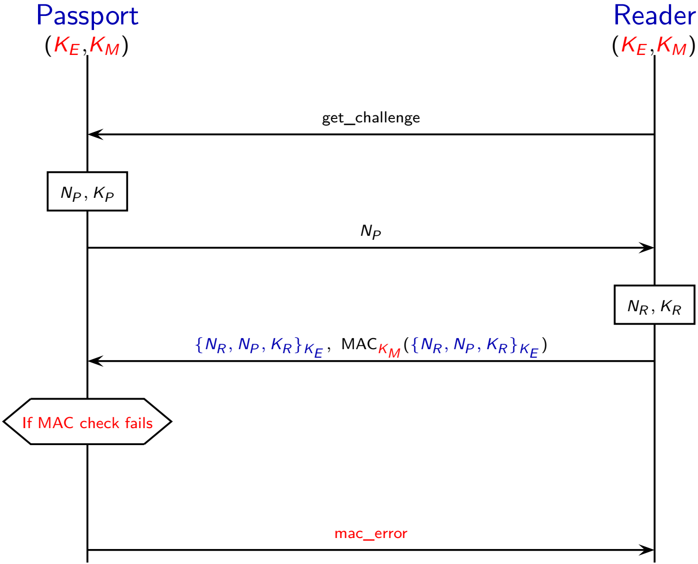
French electronic passport

$\longrightarrow$ the passport must reply to all received messages.



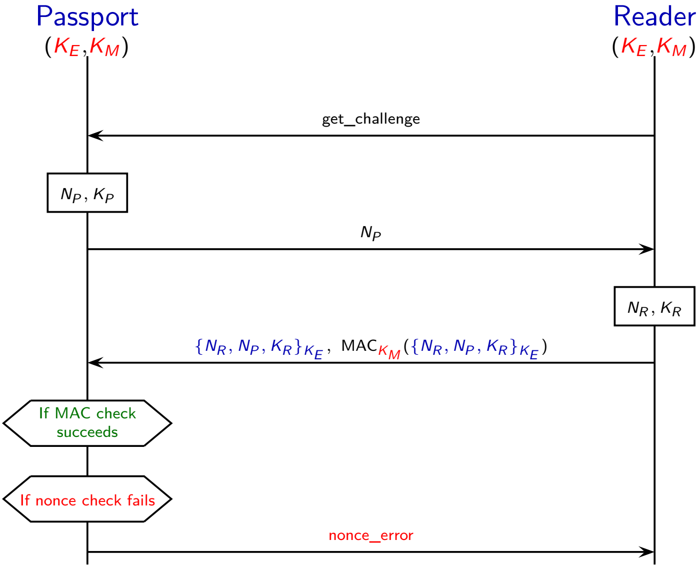
French electronic passport

$\longrightarrow$ the passport must reply to all received messages.



French electronic passport

$\longrightarrow$ the passport must reply to all received messages.



An attack on the French passport [Chothia & Smirnov, 10]

Attack against unlinkability

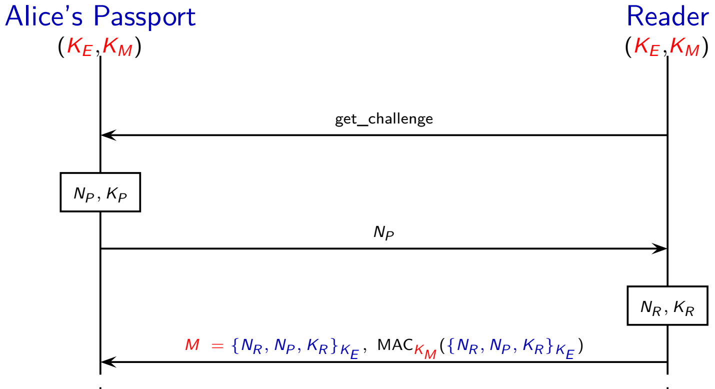
An attacker can track a French passport, provided he has once witnessed a successful authentication.

An attack on the French passport [Chothia & Smirnov, 10]

Attack against unlinkability

An attacker can track a French passport, provided he has once witnessed a successful authentication.

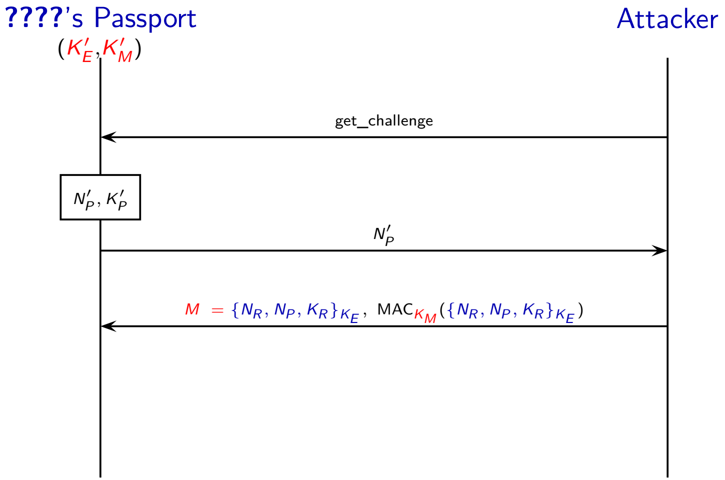
Part 1 of the attack. The attacker eavesdropes on Alice using her passport and records message M.



An attack on the French passport [Chothia & Smirnov, 10]

Part 2 of the attack.

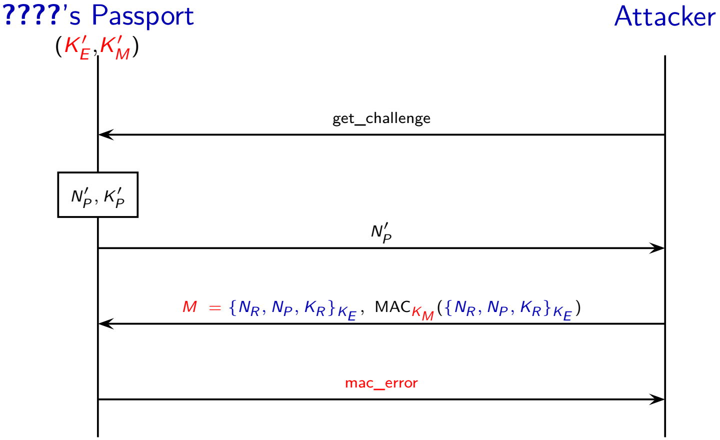
The attacker replays the message M and checks the error code he receives.



An attack on the French passport [Chothia & Smirnov, 10]

Part 2 of the attack.

The attacker replays the message M and checks the error code he receives.

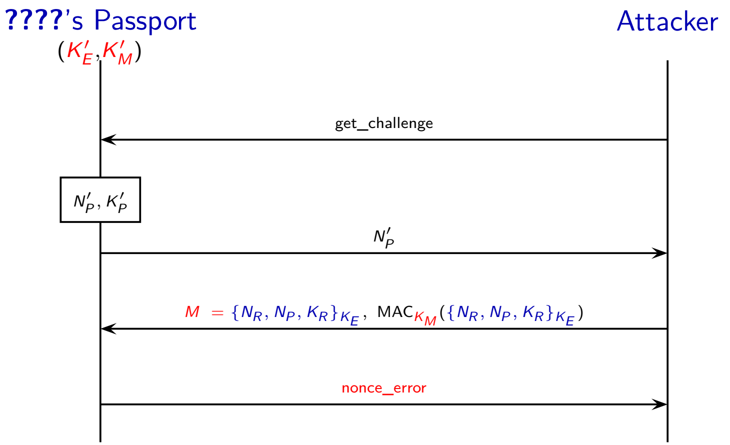


$\Longrightarrow$ MAC check failed $\Longrightarrow$ $K'_M \neq K_M$ $\Longrightarrow$ **????** is not Alice

An attack on the French passport [Chothia & Smirnov, 10]

Part 2 of the attack.

The attacker replays the message M and checks the error code he receives.



$\Longrightarrow$ MAC check succeeded $\Longrightarrow$ $K'_M = K_M$ $\Longrightarrow$ **????** is Alice

DEMO

(thanks to Myrto Arapinis, Tom Chothia, and Vincent Cheval
... and to those who lend me their e-passport.)

Attack found in 2010 by T. Chothia and V. Smirnov

Objectives

Formal and automatic analysis of new applications

Target applications: electronic voting protocols, RFID protocols, routing protocols, vehicular ad hoc networks, electronic auction protocols, . . .

Objectives

Formal and automatic analysis of new applications

Target applications: electronic voting protocols, RFID protocols, routing protocols, vehicular ad hoc networks, electronic auction protocols, . . .

Challenges:

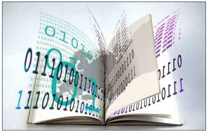
1. Formal definitions of the expected security properties
   $\longrightarrow$ privacy-type security properties
2. Designing appropriate verification algorithms
3. Modularity issues

Some basic features (symbolic models)

$\longrightarrow$ Various models (*e.g.* [Dolev & Yao, 81]) having some common features

Some basic features (symbolic models)

$\longrightarrow$ Various models (*e.g.* [Dolev & Yao, 81]) having some common features

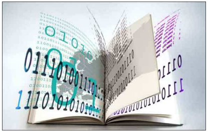


Messages

They are abstracted by terms together with an equational theory.

Some basic features (symbolic models)

$\longrightarrow$ Various models (*e.g.* [Dolev & Yao, 81]) having some common features



Messages

They are abstracted by terms together with an equational theory.

Examples:

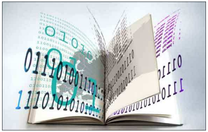
$\rightarrow$ symmetric encryption/decryption: $\mathsf{dec}(\mathsf{enc}(x, y), y) = x$

$\rightarrow$ exclusive or operator:

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

Some basic features (symbolic models)

$\longrightarrow$ Various models (*e.g.* [Dolev & Yao, 81]) having some common features



Messages

They are abstracted by terms together with an equational theory.

The attacker

- may read every message sent on the network,
- may intercept and send new messages according to its deduction capabilities.
  $\longrightarrow$ only symbolic manipulations on terms.

Formal definition of privacy-type properties

Equivalence based properties

"An observer cannot observe any difference between P and Q"

$\longrightarrow$ unlinkability, anonymity, privacy related properties in e-voting, ...

Formal definition of privacy-type properties

Equivalence based properties

"An observer cannot observe any difference between P and Q"

$\longrightarrow$ unlinkability, anonymity, privacy related properties in e-voting, ...

Recently, some formal definitions have been proposed:

- vote-privacy [Delaune *et al.*, 2008],
- unlinkability in RFID systems [Arapinis *et al.*, 2010], [Bruso *et al.*, 2010],

... but some definitions are still missing for many applications (*e.g.* anonymous routing protocols)

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions

e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions

e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Unbounded number of sessions [Blanchet, Abadi & Fournet, 05]

ProVerif tool [Blanchet, 01] `http://www.proverif.ens.fr/`

- $+$ unbounded number of sessions; various cryptographic primitives;
- $-$ termination is not guaranteed; diff-equivalence (too strong)

$\longrightarrow$ ProSwapper extension [Smyth, 10]

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions

e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Unbounded number of sessions [Blanchet, Abadi & Fournet, 05]

ProVerif tool [Blanchet, 01] `http://www.proverif.ens.fr/`

- + unbounded number of sessions; various cryptographic primitives;
- – termination is not guaranteed; diff-equivalence (too strong)

$\longrightarrow$ ProSwapper extension [Smyth, 10]

$\longrightarrow$ None of these results is able to analyse the e-passport protocol.

A recent contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune CCS 2011

Main result

A procedure for deciding testing equivalence for a large class of processes.

A recent contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune CCS 2011

Main result

A procedure for deciding testing equivalence for a large class of processes.

Our class of processes:

- + non-trivial else branches, private channels, and non-deterministic choice;
- – but no replication, and a fixed set of cryptographic primitives (signature, encryption, hash function, mac).

A recent contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune CCS 2011

Main result

A procedure for deciding testing equivalence for a large class of processes.

Our class of processes:

- + non-trivial else branches, private channels, and non-deterministic choice;
- – but no replication, and a fixed set of cryptographic primitives (signature, encryption, hash function, mac).

$\longrightarrow$ this allows us in particular to deal with the e-passport example

Modularity issues (1/2)

Some motivations:

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Modularity issues (1/2)

Some motivations:

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Example:

$$P_1 : \quad A \to B : \quad \{s\}_{\mathsf{pub}(B)}$$

Question: What about the secrecy of s?

Modularity issues (1/2)

Some motivations:

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Example:

$P_1: \quad A \to B: \quad \{s\}_{\mathsf{pub}(B)}$

$P_2: \quad A \to B: \quad \{N_a\}_{\mathsf{pub}(B)}$
$\qquad\; B \to A: \quad N_a$

Question: What about the secrecy of s?

Modularity issues (2/2)

Our goals

investigate sufficient conditions to ensure that protocols (that may share some keys) can be safely used in an environment where:

1. other sessions of the same protocol may be executed;
2. other sessions of another protocol may be executed as well.

Modularity issues (2/2)

Our goals

investigate sufficient conditions to ensure that protocols (that may share some keys) can be safely used in an environment where:

1. other sessions of the same protocol may be executed;
2. other sessions of another protocol may be executed as well.

Several results have been proposed for sequential/parallel composition, *e.g.*:

- parallel composition using tagging
  $\longrightarrow$ [Guttman & Thayer, 2000], [Cortier *et al.*, 2007]
- sequential composition for arbitrary primitives
  $\longrightarrow$ [Ciobaca & Cortier, 2010]

... but none of them are well-suited for analysing privacy-type properties

Conclusion

Conclusion

- need of formal methods in verification of security protocols
- state-of-the-art is quite satisfactory to anlayse classical security properties (secrecy, authentication, ...)

Conclusion

Conclusion

- need of formal methods in verification of security protocols
- state-of-the-art is quite satisfactory to anlayse classical security properties (secrecy, authentication, ...)

It remains a lot to do for analysing privacy-type properties:

- formal definitions of some sublte security properties (receipt-freeness, coercion-resistance, . . .)
- algorithms (and tools!) for checking automatically trace equivalence for various cryptographic primitives;
- more composition results.

Main topics of the ANR JCJC - VIP project
(Jan. 2012 - Dec 2015)
`http://www.lsv.ens-cachan.fr/Projects/anr-vip/`

Research Theme 2 (RT2)

More precisely in "privacy analysis using logical approach" (RT 2.1)

Some expectations

1. new collaborations
   $\longrightarrow$ in particular with the COMÈTE team
   - on privacy analysis using logical approach
     Mayla Brusò, Konstantinos Chatzikokolakis, Jerry den Hartog, *Formal Verification of Privacy for RFID Systems*. CSF 2010: 75-88
   - on privacy analysis using probabilistic approach
2. new case studies
   $\longrightarrow$ Examples: protocols used to protect online social networks and/or electronic health record systems