Verification of security protocols: from confidentiality to privacy

Stéphanie Delaune

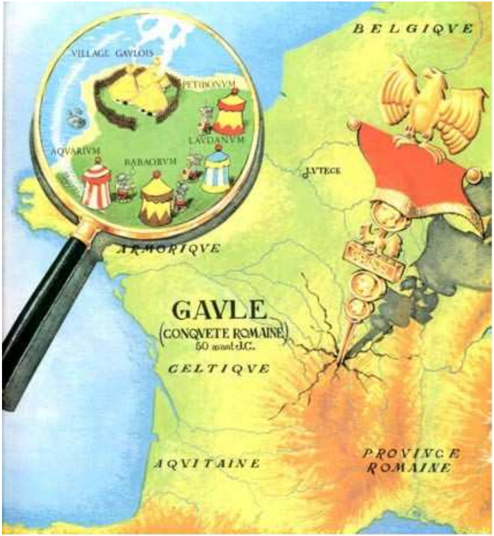
CNRS & IRISA, Rennes, France

Wednesday, May 10th, 2017

Research at IRISA (Rennes)

$\longrightarrow$ 800 members (among which about 400 reasearchers)

Where is it?



Coming soon !

september 2017

Rennes to Paris in 90 min. by train.

EMSEC team

Embedded Security & Cryptography

$\longrightarrow$ 6 permanent researchers, 12 PhD students, and 2 post-docs

P. Derbez, G. Avoine, A. Roux-Langlois, B. Kordy, & P.-A. Fouque.

Cryptographic 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.

Cryptographic protocols everywhere !

Cryptographic protocols

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

It becomes more and more important to protect our privacy.

Electronic passport

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

An e-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.

Electronic passport

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

An e-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

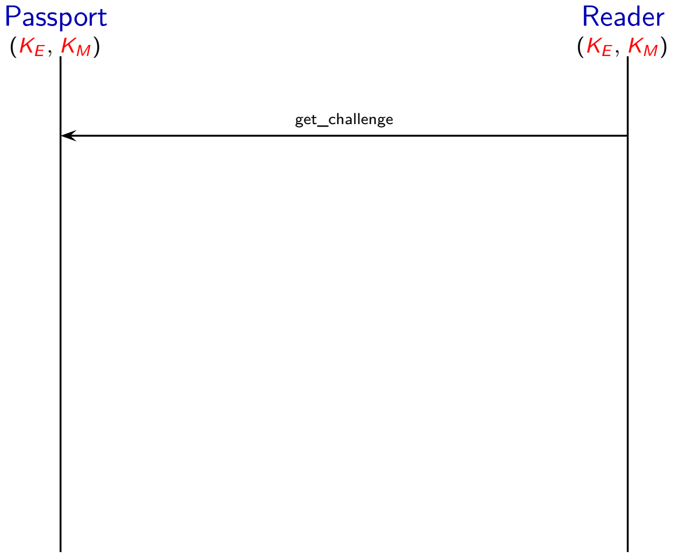
Unlinkability aims to ensure *that a user may make multiple uses of a service or resource without others being able to link these uses together*.

Basic Acccess Control (BAC) protocol

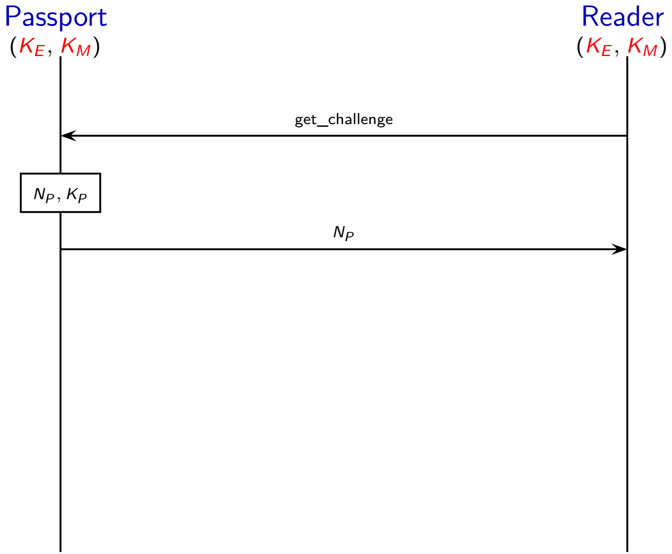
Passport
(K_E, K_M)

Reader
(K_E, K_M)

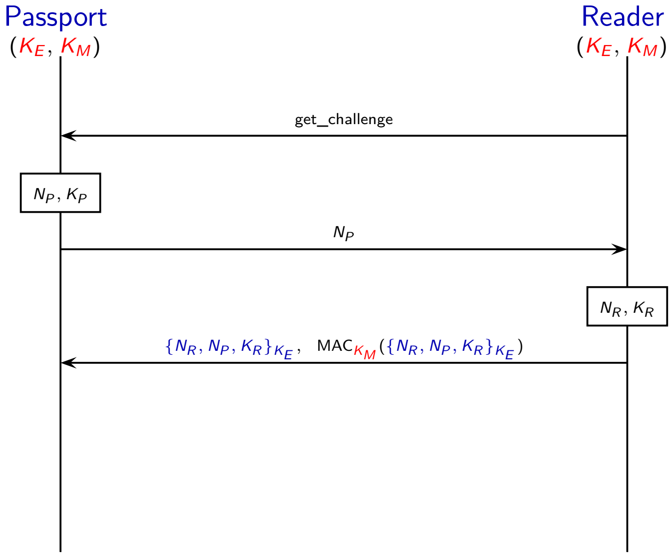
Basic Acccess Control (BAC) protocol



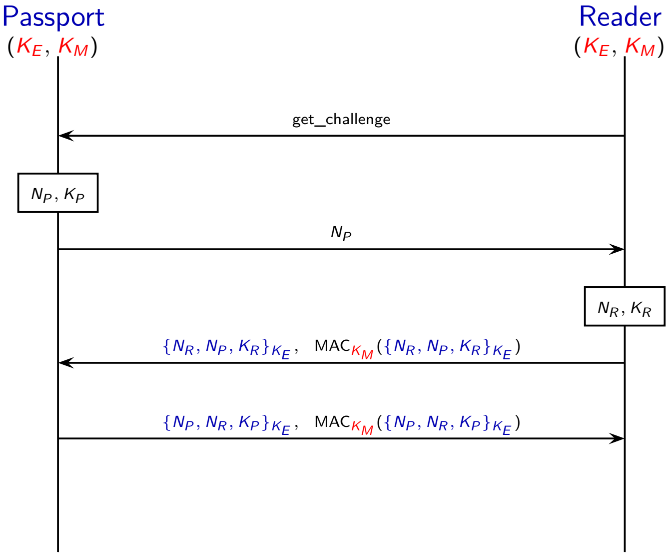
Basic Acccess Control (BAC) protocol



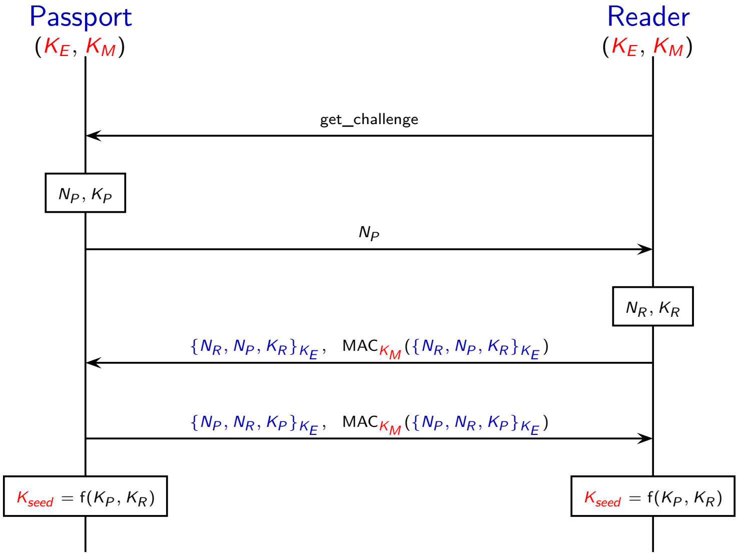
Basic Acccess Control (BAC) protocol



Basic Acccess Control (BAC) protocol



Basic Acccess Control (BAC) protocol



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, ...
- subtle and hard to detect by "eyeballing" the protocol

This is the so-called **Dolev-Yao attacker** !

as explained on Monday in the talk of Alessandro Armando

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

$\longrightarrow$ A traceability attack on the BAC protocol (2010)

Security

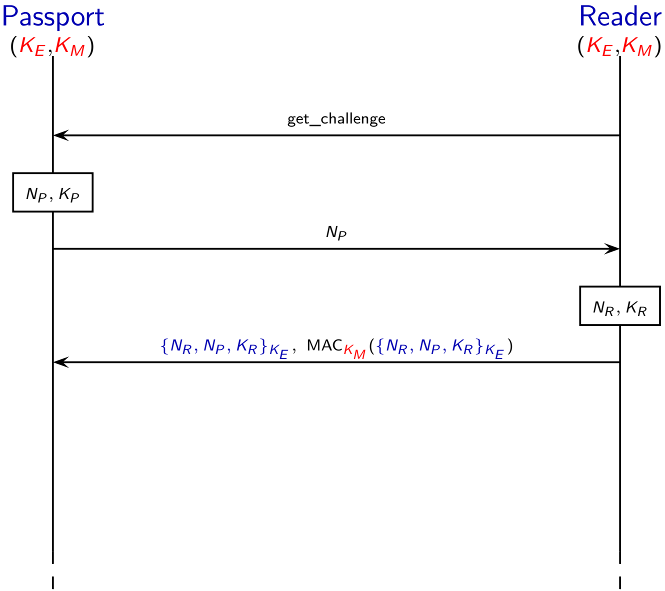
Defects in e-passports allow real-time tracking

This threat brought to you by RFID

The register - Jan. 2010

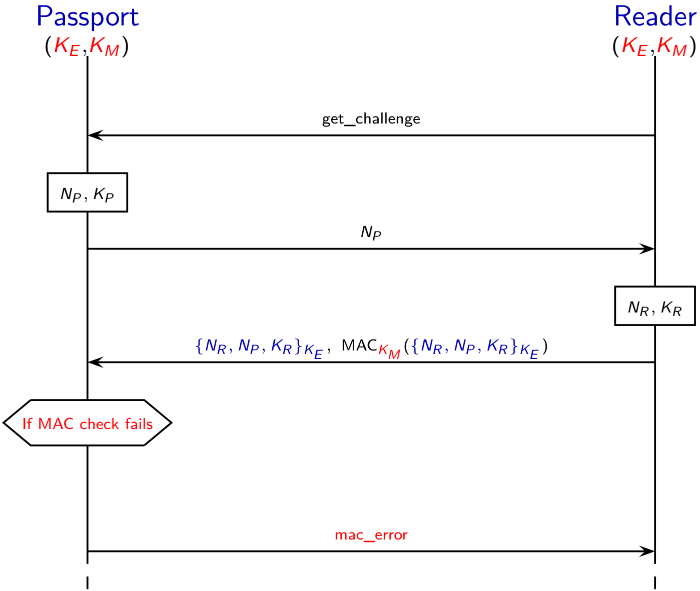
French electronic passport

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



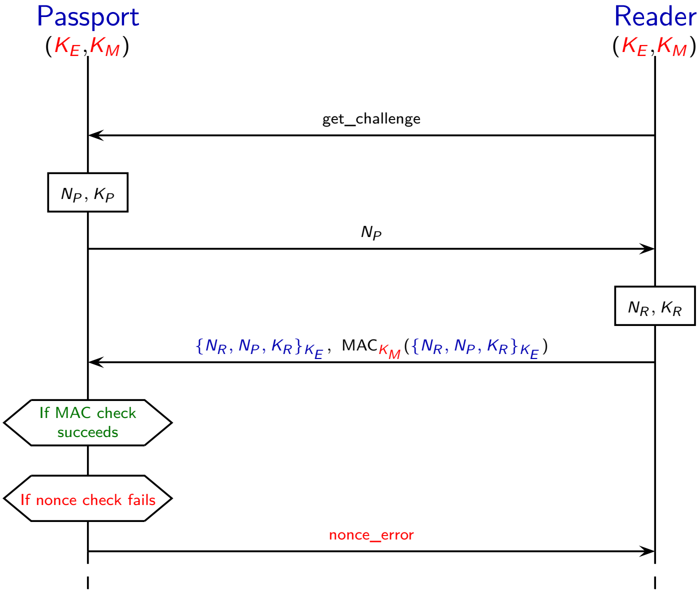
French electronic passport

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



French electronic passport

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



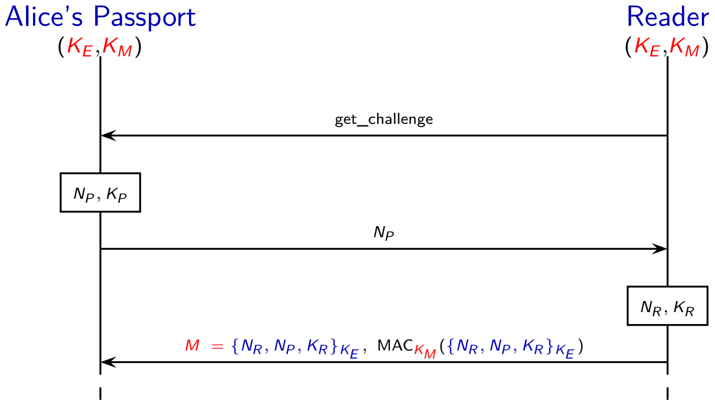
An attack on the French passport [Chothia & Smirnov, 10]

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

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

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

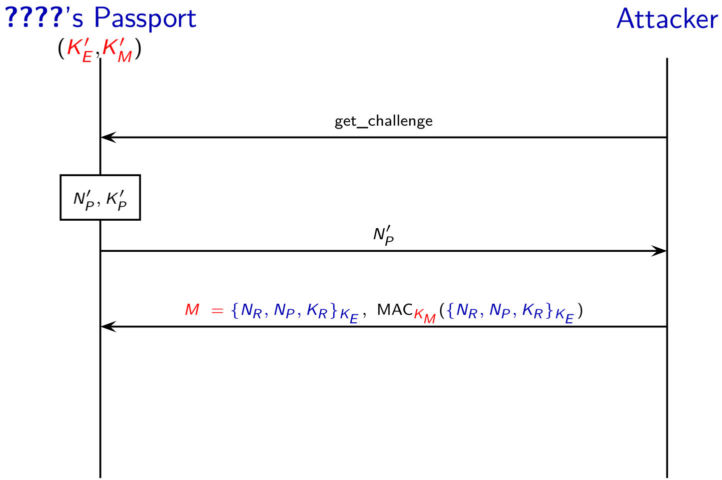
Part 1 of the attack. The attacker eavesdropes on Alice using her passport and records message M.



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

Part 2 of the attack.

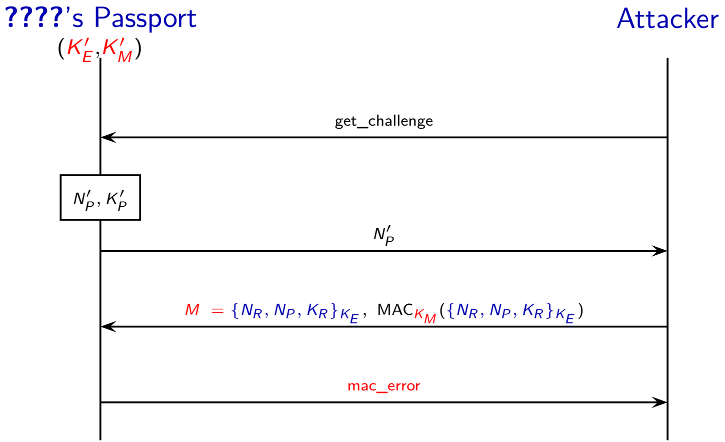
The attacker replays the message M and checks the error code he receives.



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.

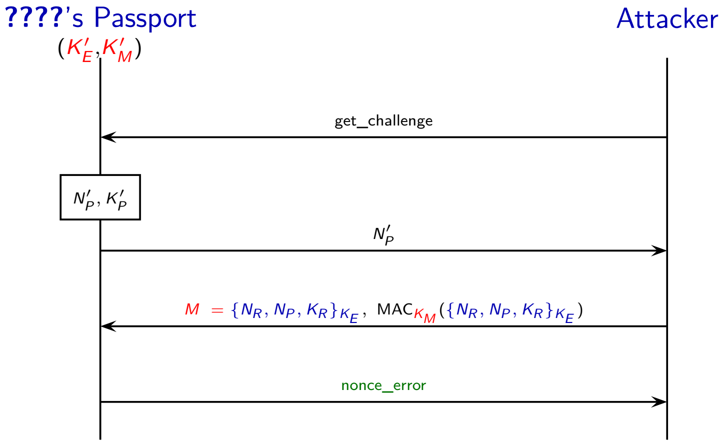


MAC check failed $\implies$ $K'_M \neq K_M$ $\implies$ **????** is not Alice

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.

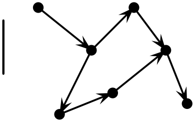


MAC check succeeded $\Longrightarrow$ $K'_M = K_M$ $\Longrightarrow$ **????** is Alice

Outline

Does the protocol *satisfy* a security property?

Modelling



$\models \quad \varphi$

Outline of the remaining of this talk

1. Modelling cryptographic protocols and their security properties
2. Designing verification algorithms

$\longrightarrow$ we focus here on privacy-type security properties

Part I

Modelling cryptographic protocols and their security properties

Two major families of models ...

... with some advantages and some drawbacks.

Computational model

- + messages are bitstring, a general and powerful adversary
- – manual proofs, tedious and error-prone

Symbolic model

- – abstract model, e.g. messages are terms
- + automatic proofs

Two major families of models ...

... with some advantages and some drawbacks.

Computational model

- + messages are bitstring, a general and powerful adversary
- – manual proofs, tedious and error-prone

Symbolic model

- – abstract model, e.g. messages are terms
- + automatic proofs

Some results allowed to make a link between these two very different models.
$\longrightarrow$ Abadi & Rogaway 2000

Messages as terms

Terms are built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

$$
\begin{array}{lcll}
\mathrm{t} & ::= & n & \text{name } n \\
 & | & f(t_1, \ldots, t_k) & \text{application of symbol } f \in \mathcal{F}
\end{array}
$$

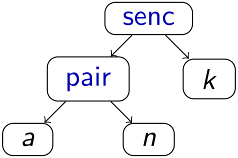
Messages as terms

Terms are built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

$$\begin{array}{lcll} t & ::= & n & \text{name } n \\ & | & f(t_1, \dots, t_k) & \text{application of symbol } f \in \mathcal{F} \end{array}$$

Example: representation of $\{a, n\}_k$

- Names: n, k, a
- constructors: senc, pair,



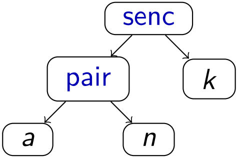
Messages as terms

Terms are built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

$$\begin{array}{rcll} \mathrm{t} & ::= & n & \text{name } n \\ & | & f(t_1, \ldots, t_k) & \text{application of symbol } f \in \mathcal{F} \end{array}$$

Example: representation of $\{a, n\}_k$

- Names: n, k, a
- constructors: senc, pair,
- destructors: sdec, proj_1, proj_2.



The term algebra is equipped with an equational theory E.

$$\mathsf{sdec}(\mathsf{senc}(x, y), y) = x \qquad \begin{array}{rcl} \mathsf{proj}_1(\mathsf{pair}(x, y)) & = & x \\ \mathsf{proj}_2(\mathsf{pair}(x, y)) & = & y \end{array}$$

Example: $\mathsf{sdec}(\mathsf{senc}(s, k), k) =_{\mathsf{E}} s$.

Protocols as processes

The Applied pi calculus is a basic programming language with constructs for concurrency and communication

[Abadi & Fournet, 01]

Protocols as processes

The Applied pi calculus is a basic programming language with constructs for concurrency and communication

[Abadi & Fournet, 01]

$$
\begin{array}{lcll}
P, Q & := & 0 & \text{null process} \\
 & & \mathrm{in}(c, x).P & \text{input} \\
 & & \mathrm{out}(c, u).P & \text{output} \\
 & & \text{if } u = v \text{ then } P \text{ else } Q & \text{conditional} \\
 & & P \mid Q & \text{parallel composition} \\
 & & !P & \text{replication} \\
 & & \text{new } n.P & \text{fresh name generation}
\end{array}
$$

Back to the BAC protocol

Back to the BAC protocol

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc/2, sdec/2
- concatenation/projections: $\langle\ ,\ \rangle/2$, $\mathsf{proj}_1/1$, $\mathsf{proj}_2/1$
- mac construction: mac/2

$\longrightarrow$ $\mathsf{sdec}(\mathsf{senc}(x, y), y) = x$, $\mathsf{proj}_1(\langle x, y\rangle) = x$, $\mathsf{proj}_2(\langle x, y\rangle) = y$.

Nonces n_r, n_p, and keys k_r, k_p, k_e, k_m are modelled using names

Back to the BAC protocol

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc/2, sdec/2
- concatenation/projections: $\langle\ ,\ \rangle/2$, $\mathsf{proj}_1/1$, $\mathsf{proj}_2/1$
- mac construction: mac/2

$\longrightarrow \quad \mathsf{sdec}(\mathsf{senc}(x,y),y) = x, \quad \mathsf{proj}_1(\langle x,y\rangle) = x, \quad \mathsf{proj}_2(\langle x,y\rangle) = y.$

Nonces n_r, n_p, and keys k_r, k_p, k_e, k_m are modelled using names

Modelling Passport's role

$$
\begin{array}{l}
P_{\mathrm{BAC}}(k_E, k_M) \ = \mathsf{new}\ n_P.\mathsf{new}\ k_P.\mathsf{out}(n_P).\mathsf{in}(\langle z_E, z_M\rangle). \\
\quad \mathtt{if}\ z_M = \mathsf{mac}(z_E, k_M)\ \mathtt{then}\ \mathtt{if}\ n_P = \mathsf{proj}_1(\mathsf{proj}_2(\mathsf{sdec}(z_E, k_E))) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathtt{then}\ \mathsf{out}(\langle m, \mathsf{mac}(m, k_M)\rangle) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathtt{else}\ \mathsf{out}(\mathit{nonce\_error}) \\
\qquad\qquad\qquad\qquad\qquad \mathtt{else}\ \mathsf{out}(\mathit{mac\_error})
\end{array}
$$

where $m = \mathsf{senc}(\langle n_P, \langle \mathsf{proj}_1(z_E), k_P\rangle\rangle, k_E)$.

Semantics

Semantics $\rightarrow$:

COMM	$\mathsf{out}(c, u).P \mid \mathsf{in}(c, x).Q \rightarrow P \mid Q\{u/x\}$
THEN	$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \rightarrow P$ when $u =_{\mathsf{E}} v$
ELSE	$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \rightarrow Q$ when $u \neq_{\mathsf{E}} v$

Semantics

Semantics $\rightarrow$:

$$
\begin{array}{ll}
\textsc{Comm} & \mathrm{out}(c,u).P \mid \mathrm{in}(c,x).Q \rightarrow P \mid Q\{u/x\} \\
\textsc{Then} & \text{if } u = v \text{ then } P \text{ else } Q \rightarrow P \text{ when } u =_{\mathsf{E}} v \\
\textsc{Else} & \text{if } u = v \text{ then } P \text{ else } Q \rightarrow Q \quad \text{ when } u \neq_{\mathsf{E}} v
\end{array}
$$

closed by

- structural equivalence ($\equiv$):

$$P \mid Q \equiv Q \mid P, \qquad P \mid 0 \equiv P, \qquad \ldots$$

- application of evaluation contexts:

$$\frac{P \rightarrow P'}{\mathrm{new}\, n.\, P \rightarrow \mathrm{new}\, n.\, P'} \qquad \frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q}$$

What does unlinkability mean?

Informally, an observer/attacker can not observe the difference between the two following situations:

1. a situation where the same passport may be used twice (or even more);
2. a situation where each passport is used at most once.

What does unlinkability mean?

Informally, an observer/attacker can not observe the difference between the two following situations:

1. a situation where the same passport may be used twice (or even more);
2. a situation where each passport is used at most once.

More formally,

$$!\mathrm{new}\ ke.\mathrm{new}\ km.(\underset{\substack{\uparrow\\ \boxed{\text{many sessions for each passport}}}}{!P_{\mathrm{BAC}}} \mid !R_{\mathrm{BAC}}) \overset{?}{\approx} !\mathrm{new}\ ke.\mathrm{new}\ km.(\underset{\substack{\uparrow\\ \boxed{\text{only one session for each passport}}}}{P_{\mathrm{BAC}}} \mid R_{\mathrm{BAC}})$$

(we still have to formalize the notion of equivalence)

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Example 1: $\mathrm{out}(a, yes) \stackrel{?}{\approx} \mathrm{out}(a, no)$

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Example 1: $\mathrm{out}(a, \mathit{yes}) \not\approx \mathrm{out}(a, \mathit{no})$

$$\longrightarrow \quad A = \mathrm{in}(a, x).\mathrm{if}\ x = \mathit{yes}\ \mathrm{then}\ \mathrm{out}(c, \mathit{ok})$$

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Example 2:

$$\mathsf{new}\, s.\mathsf{out}(a, \mathsf{senc}(s, k)).\mathsf{out}(a, \mathsf{senc}(s, k'))$$
$$\overset{?}{\approx}$$
$$\mathsf{new}\, s, s'.\mathsf{out}(a, \mathsf{senc}(s, k)).\mathsf{out}(a, \mathsf{senc}(s', k'))$$

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Example 2:

$$\begin{gathered}\text{new } s.\mathsf{out}(a, \mathsf{senc}(s, k)).\mathsf{out}(a, \mathsf{senc}(s, k')) \\ \not\approx \\ \text{new } s, s'.\mathsf{out}(a, \mathsf{senc}(s, k)).\mathsf{out}(a, \mathsf{senc}(s', k'))\end{gathered}$$

$\longrightarrow A = \mathsf{in}(a, x).\mathsf{in}(a, y).\mathsf{if}\ (\mathsf{sdec}(x, k) = \mathsf{sdec}(y, k'))$ then $\mathsf{out}(c, ok)$

Testing equivalence

Definition - Testing equivalence - $P \approx Q$

for all processes A, we have that:

$$(A \mid P) \Downarrow_c \text{ if, and only if, } (A \mid Q) \Downarrow_c$$

where $P \Downarrow_c$ means that P can evolve and emits on channel c.

Exercise: Are the two following processes in testing equivalence?

$$\mathsf{new}\, s.\mathsf{out}(a, s) \overset{?}{\approx} \mathsf{new}\, k.\mathsf{out}(a, \mathsf{senc}(\mathsf{yes}, k))$$

Some other equivalence-based security properties

The notion of testing equivalence can be used to express:

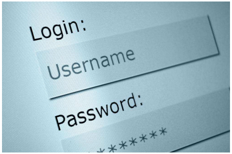
Vote privacy

the fact that a particular voted in a particular way is not revealed to anyone

Strong secrecy

the fact that an adversary cannot see any difference when the value of the secret changes

$\longrightarrow$ stronger than the notion of secrecy as non-deducibility.



Guessing attack

the fact that an adversary can not learn the value of passwords even if he knows that they have been choosen in a particular dictionary.

Part II

Designing verification algorithms for privacy-type properties

How can we check testing equivalence?

The problem is undecidable in general

$\longrightarrow$ even under quite severe restrictions [Chrétien PhD thesis, 2016]

How can we check testing equivalence?

The problem is undecidable in general

$\longrightarrow$ even under quite severe restrictions [Chrétien PhD thesis, 2016]

Several procedures and automatic tools already exist !

How can we check testing equivalence?

The problem is undecidable in general

$\longrightarrow$ even under quite severe restrictions [Chrétien PhD thesis, 2016]

Several procedures and automatic tools already exist !

Two main categories of tools have been developed:

- bounded number of sessions: Spec [Dawson & Tiu, 2010], Apte [Cheval et al, 2011], and Akiss [Chadha et al, 2012].
- unbounded number of sessions: ProVerif [Blanchet et al, 2005], Tamarin [Basin et al, 2015], and Maude-NPA [Yang et al, 2016].

Part II.A

Designing verification algorithms for privacy-type properties

for a bounded number of sessions

Testing equivalence for a bounded number of sessions

$\longrightarrow$ decidable when considering classical primitives

- A decision procedure implemented in the tool Apte:
  non-trivial else branches, private channels, and
  non-deterministic choice, a fixed set of primitives
  [Cheval, Comon & D., 11]
- A procedure implemented in the tool Akiss:
  no else branches, but a larger class of primitives
  [Chadha et al, 12]

$\longrightarrow$ Work in progress: a procedure that takes advantage of both !

Testing equivalence for a bounded number of sessions

$\longrightarrow$ decidable when considering classical primitives

- A decision procedure implemented in the tool Apte: non-trivial else branches, private channels, and non-deterministic choice, a fixed set of primitives

  [Cheval, Comon & D., 11]

- A procedure implemented in the tool Akiss: no else branches, but a larger class of primitives

  [Chadha et al, 12]

$\longrightarrow$ Work in progress: a procedure that takes advantage of both !

Main limitation: a limited practical impact because these tools scale badly, e.g. unlinkability of a fixed version of BAC (2 sessions)

$\longrightarrow$ **more than 2 days !**

Partial order reduction for security protocols

[Hirschi PhD thesis, 2017]

Main objective

to develop POR techniques that are suitable for analysing security protocols (especially testing equivalence)

Partial order reduction for security protocols

[Hirschi PhD thesis, 2017]

Main objective

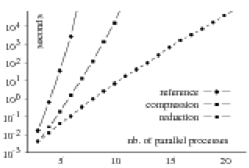
to develop POR techniques that are suitable for analysing security protocols (especially testing equivalence)

Example: $\mathsf{in}(c_1, x_1).\mathsf{out}(c_1, \mathsf{ok}) \mid \mathsf{in}(c_2, x_2).\mathsf{out}(c_2, \mathsf{ok})$

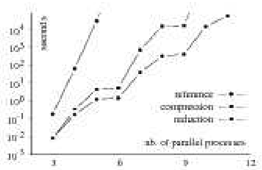
We propose two optimizations:

1. compression: we impose a simple strategy on the exploration of the available actions (roughly outputs are performed first and using a fixed arbitrary order)
2. reduction: we avoid exploring some redundant traces taking into account the data that are exchanged

Practical impact of our optimizations (in APTE)



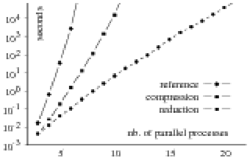
Toy example



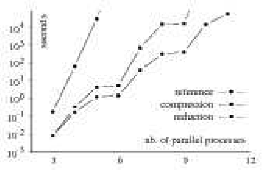
Denning Sacco protocol

$\longrightarrow$ Each optimisation brings an exponential speedup.

Practical impact of our optimizations (in APTE)



Toy example



Denning Sacco protocol

⟶ Each optimisation brings an exponential speedup.

Protocol	reference	with POR
Yahalom (3-party)	4	5
Needham Schroeder (3-party)	4	7
Private Authentication (2-party)	4	7
E-Passport PA (2-party)	4	9
Denning-Sacco (3-party)	5	10
Wide Mouthed Frog (3-party)	6	13

Maximum number of parallel processes verifiable in 20 hours.

⟶ Our optimisations make Apte much more useful in practice for investigating interesting scenarios.

SAT-Equiv: a new tool for checking testing equivalence

[CSF, 2017]

SAT-Equiv in a nutshell:

- inspired from SATMC [Armando et al, 2014];
- bounded verification (messages of bounded size)
  $\longrightarrow$ this is possible without missing any attacks when protocols are type-compliant. [Chretien PhD thesis, 2016]
- a successful combination of techniques developed for planning, and the use of SAT solvers;
- less sensitive to the number of concurrent sessions analysed.

SAT-Equiv: a new tool for checking testing equivalence

[CSF, 2017]

SAT-Equiv in a nutshell:

- inspired from SATMC [Armando et al, 2014];
- bounded verification (messages of bounded size)
  $\longrightarrow$ this is possible without missing any attacks when protocols are type-compliant. [Chretien PhD thesis, 2016]
- a successful combination of techniques developed for planning, and the use of SAT solvers;
- less sensitive to the number of concurrent sessions analysed.

Work in progress:

- more cryptographic primitives: asymmetric encryption, signature, . . .
- a larger class of processes: else branches, beyond simple processes, . . .

Some encouraging results with SAT-Equiv

Denning-Sacco protocol:

$$\begin{array}{ll} 1. & A \to S : \ A, B \\ 2. & S \to A : \ \{B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}} \\ 3. & A \to B : \ \{K_{ab}, A\}_{K_{bs}} \end{array}$$

Comparison of the different tools:

# roles	Spec	Akiss	Apte	Apte-por	**Sat-Eq**
3	12s	0.10s	0.3s	0.03s	0.25s
6	MO	15s	TO	8s	1s
7		101s		13s	2s
10		SO		39m	4s
12				TO	7s
14					10s

$\longrightarrow$ similar results when considering other protocols, e.g. Needham-Schroeder, Wide-Mouth-Frog, Yahalom, Otway Rees, ...

Part II.B

Designing verification algorithms
for privacy-type properties

for an unbounded number of sessions

Testing equivalence for an unbounded number of sessions

Some recent theoretical results [Chrétien PhD thesis, 2016]

- undecidable in general (and even under quite severe restriction)
- a first decidability result through a characterization of equivalence of protocols in terms of equality of languages of deterministic pushdown automata. [Icalp'13, TOCL'15]
- decidable for a subclass of tagged protocols [CSF'15]

Testing equivalence for an unbounded number of sessions

Some recent theoretical results [Chrétien PhD thesis, 2016]

- undecidable in general (and even under quite severe restriction)
- a first decidability result through a characterization of equivalence of protocols in terms of equality of languages of deterministic pushdown automata. [Icalp'13, TOCL'15]
- decidable for a subclass of tagged protocols [CSF'15]

Main limitations:

- a restricted set of primitives: symmetric encryption, and concatenation only;
- not really practical (no verification tool).

A more pragmatic approach

[Blanchet *et al.*, LICS'05]

ProVerif tool: `http://www.proverif.ens.fr`

- various cryptographic primitives modeled using equations;
- various security properties: secrecy, authentication, and equivalence-based properties (namely diff-equivalence);

The tool may not terminate or give false attacks.

Works very well in many situations, e.g. strong secrecy

A more pragmatic approach

[Blanchet *et al.*, LICS'05]

ProVerif tool: `http://www.proverif.ens.fr`

- various cryptographic primitives modeled using equations;
- various security properties: secrecy, authentication, and equivalence-based properties (namely diff-equivalence);

The tool may not terminate or give false attacks.

Works very well in many situations, e.g. strong secrecy

Main issue: diff-equivalence is too strong in many situations.
$\longrightarrow$ ProVerif is not suitable to analyse unlinkability properties.

The Tamarin and Maude-NPA tools are also based on diff-equivalence and they suffer from the same problem.

Our approach is pragmatic too [S&P, 2016]

Provide a method to analyse unlinkability for a large class of 2 party protocols, and tool support for that.

Our approach is pragmatic too [S&P, 2016]

> Provide a method to analyse unlinkability for a large class of 2 party protocols, and tool support for that.

On the theoretical side

2 reasonable conditions implying anonymity and unlinkability for a large class of 2 party protocols

On the practical side

- our conditions can be checked automatically using existing tools, and we provide tool support for that.
- new proofs and attacks on several RFID protocols.

$\longrightarrow$ first results published at Security & Privacy in 2016 extended since to deal with a larger class of processes

Tool support

Our two conditions can be automatically verified using ProVerif:

- well-authentication: this is a pure reachability property
  $\longrightarrow$ ProVerif (and other existing tools) works well
- frame opacity: equivalence between sequences of messages
  $\longrightarrow$ checkable with good precision via diff-equivalence

Tool support

Our two conditions can be automatically verified using ProVerif:

- well-authentication: this is a pure reachability property
  $\longrightarrow$ ProVerif (and other existing tools) works well
- frame opacity: equivalence between sequences of messages
  $\longrightarrow$ checkable with good precision via diff-equivalence

Tool UKANO

A tool built on top of ProVerif that automatically checks our two conditions.

`http://projects.lsv.ens-cachan.fr/ukano/`

Summary of our case studies using UKANO

Protocol	FO	WA	unlinkability
Feldhofer	✓	✓	**safe**
Feldhofer variant (with !)	✓	✗	attack
Hash-Lock	✓	✓	**safe**
LAK (stateless)	–	✗	attack
Fixed LAK	✓	✓	**safe**
BAC	✓	✓	**safe**
BAC/PA/AA	✓	✓	**safe**
PACE (faillible dec)	–	✗	attack
PACE (as in [Bender et al, 09])	–	✗	attack
PACE	–	✗	attack
PACE with tags	✓	✓	**safe**
DAA sign	✓	✓	**safe**
DAA join	✓	✓	**safe**
abcdh (irma)	✓	✓	**safe**

Conclusion

To sum up

Cryptographic protocols are:

- difficult to design and analyse;
- particularly vulnerable to logical attacks.

Strong primitives are necessary . . .

. . . **but this is not sufficient !**

To sum up

Cryptographic protocols are:

- difficult to design and analyse;
- particularly vulnerable to logical attacks.

It is important to ensure that
the protocols we are using every day work properly.

We now have automatic and powerful verification tools to analyse:

- classical security goals, *e.g.* secrecy and authentication;
- relatively small protocols;
- protocols that rely on standard cryptographic primitives.

Limitations of the symbolic approach

1. the algebraic properties of the primitives are abstracted away
   $\longrightarrow$ no guarantee if the protocol relies on an encryption that satisfies some additional properties (*e.g.* RSA, ElGamal)

2. only the specification is analysed and not the implementation
   $\longrightarrow$ most of the passports are actually linkable by a carefull analysis of time or message length.

   http://www.loria.fr/~glondu/epassport/attaque-tailles.html

3. when considering a bounded number of sessions, not all scenario are checked
   $\longrightarrow$ no guarantee if the protocol is used one more time !

Regarding privacy-type security properties

It remains a lot to do

- formal definitions of some sublte security properties
  $\longrightarrow$ receipt-freeness, coercion-resistance in e-voting
- algorithms (and tools!) for checking automatically trace equivalence for various cryptographic primitives;
  $\longrightarrow$ homomorphic encryption used in e-voting, exclusive-or used in RFID protocols [CSF, 2017]
- more composition results
  $\longrightarrow$ Could we derive some security guarantees of the whole e-passport application from the analysis performed on each subprotocol?
- develop more fine-grained models (and tools) to take into account side channel attacks
  $\longrightarrow$ e.g. timing attacks

Advertisement

POPSTAR ERC Project (2017-2022)
Reasoning about Physical properties
Of security Protocols
with an Application To contactless Systems

`https://project.inria.fr/popstar/`

Regular job offers:

- PhD positions and Post-doc positions;
- One research associate position (up to 5 years).

$\longrightarrow$ contact me: `stephanie.delaune@irisa.fr`

Questions ?