Formal verification of privacy

(for RFID protocols)

Stéphanie Delaune

Équipe EMSEC (IRISA), CNRS, France

Tuesday, September 13th, 2016

Security protocols everywhere !

Cryptographic protocols

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

Security 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.

Security 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

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

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

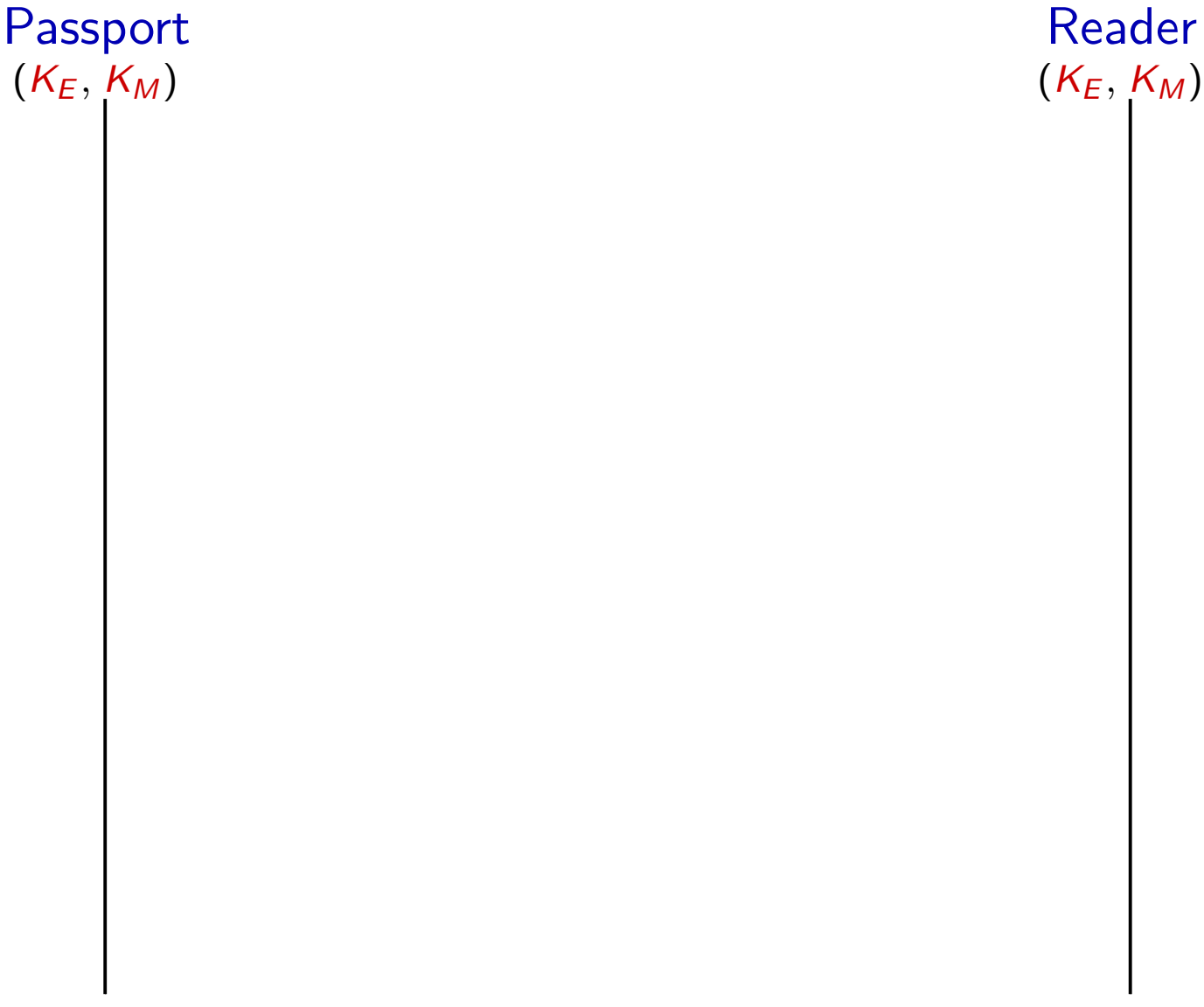
The RFID tag stores:

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

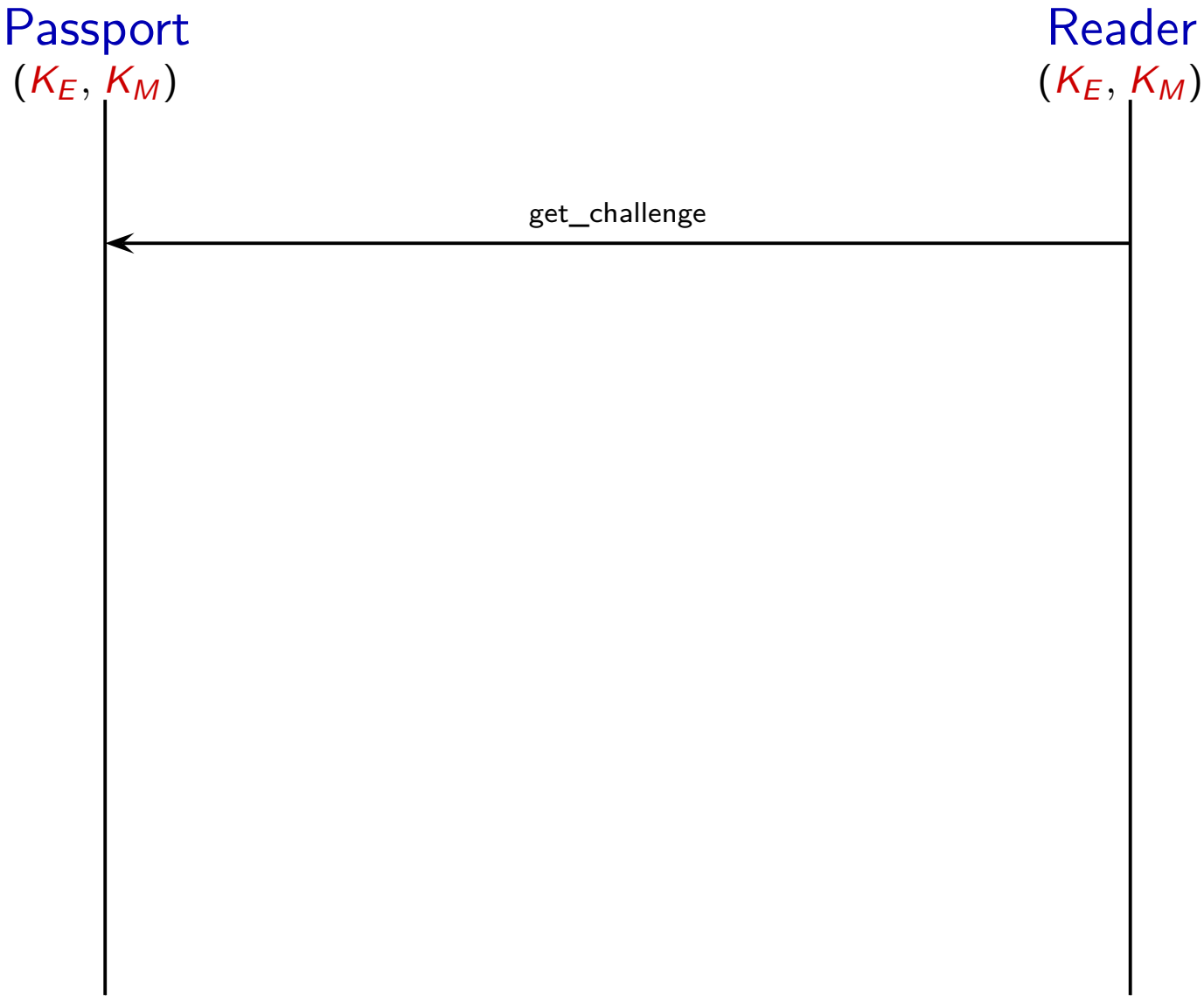
The Basic Access Control (BAC) protocol is a key establishment protocol that has been designed to protect our personnal data, and to ensure unlinkability.

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*. [ISO/IEC standard 15408]

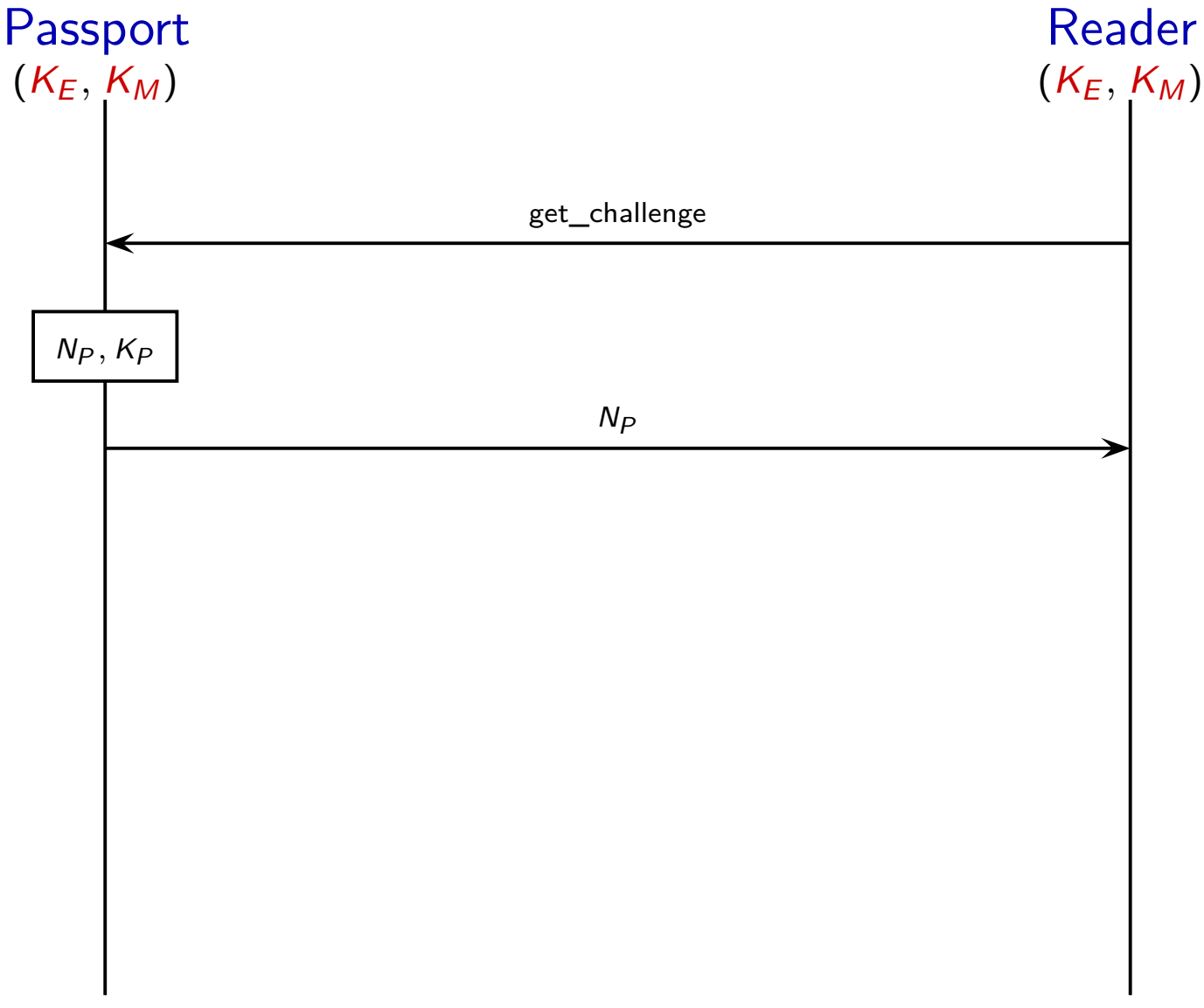
BAC protocol



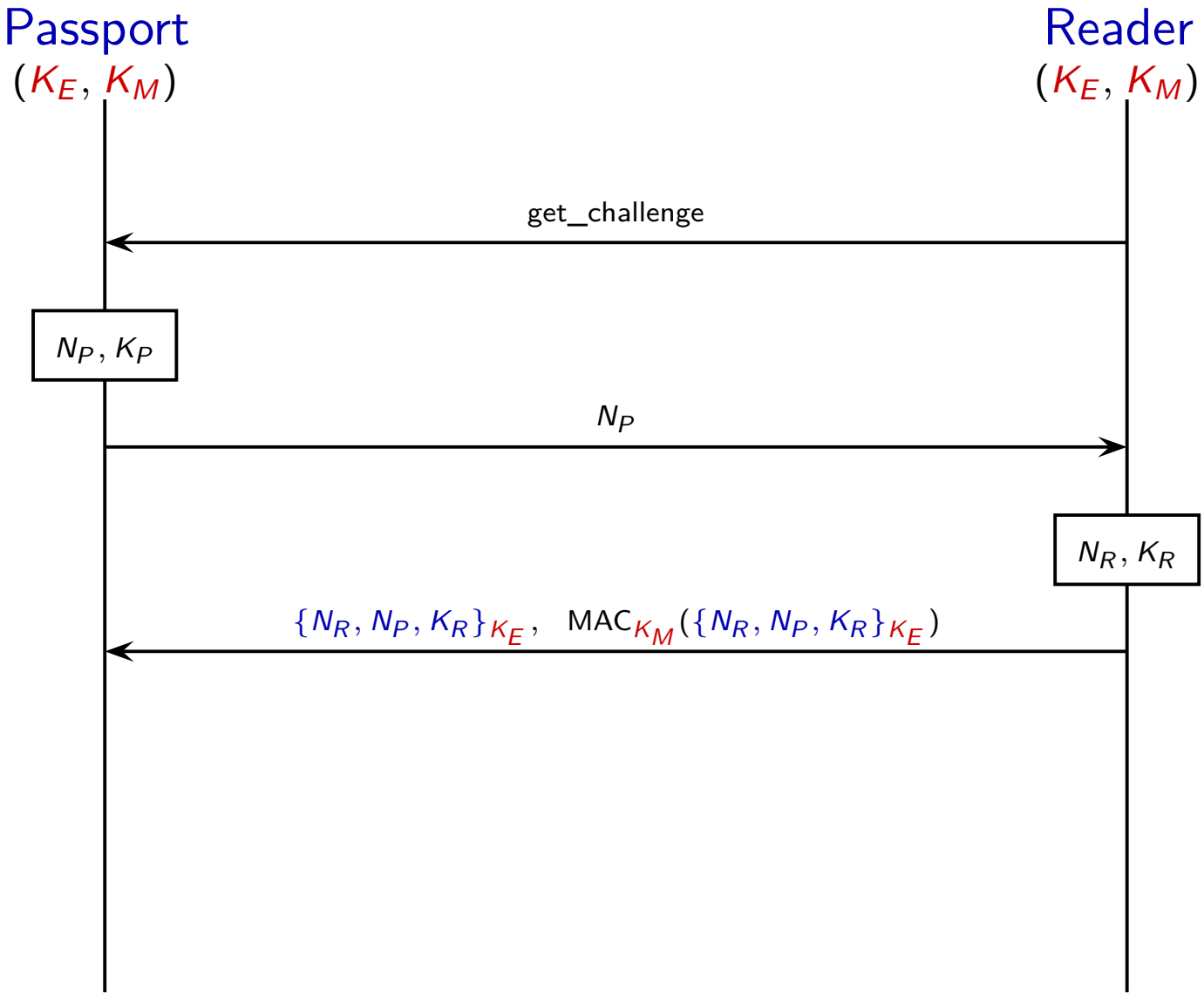
BAC protocol



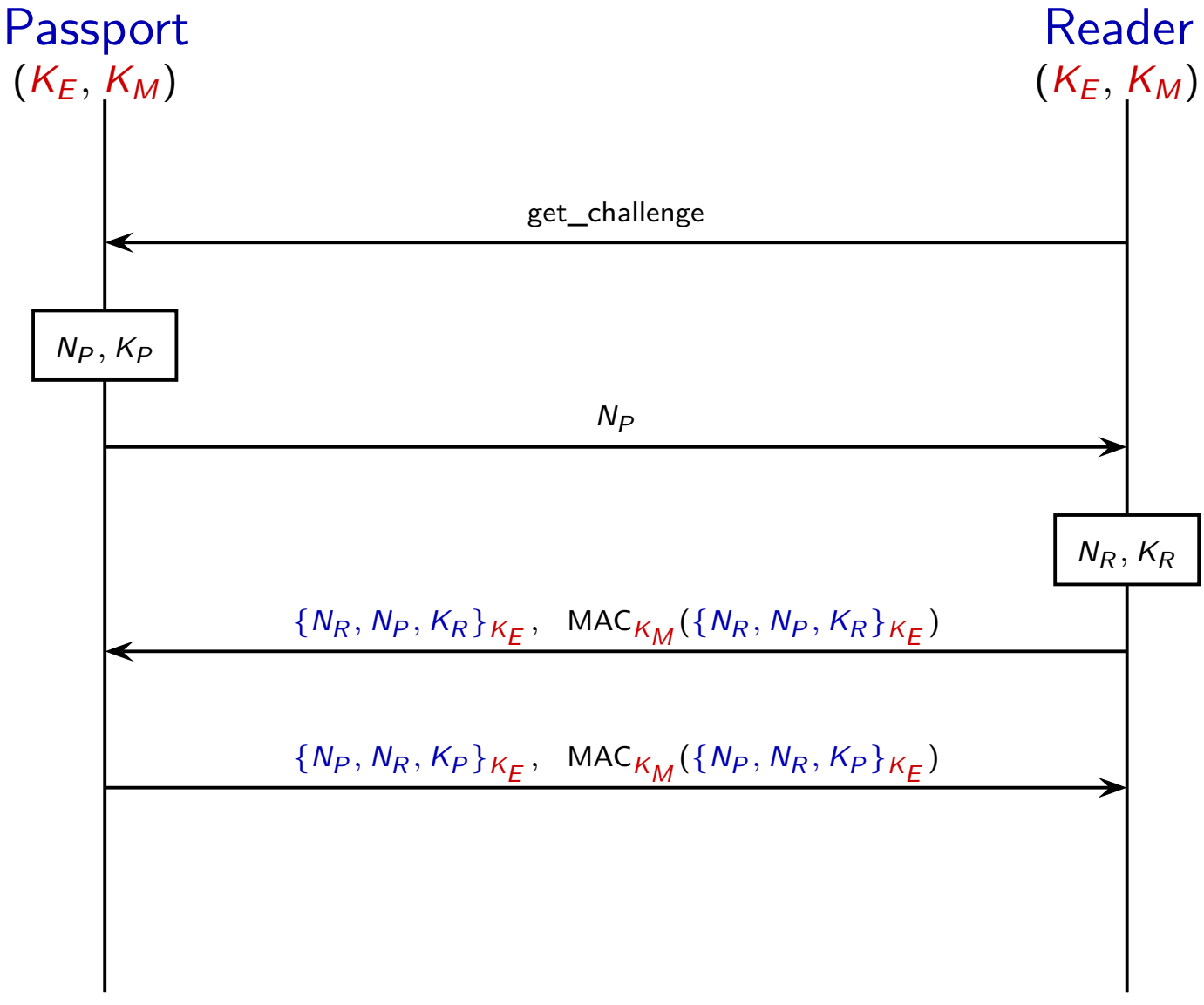
BAC protocol



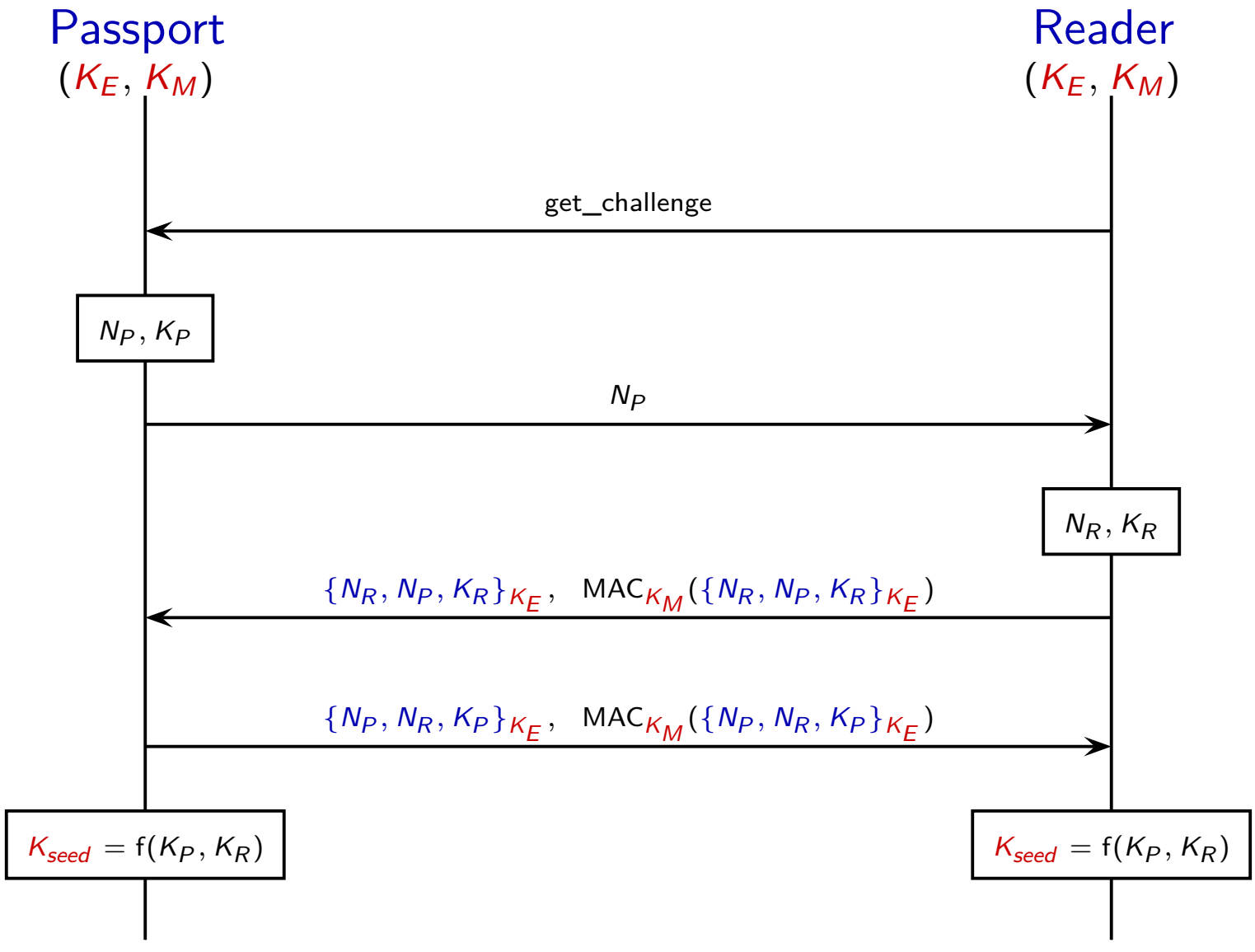
BAC protocol



BAC protocol



BAC protocol



Verifying security protocols: a difficult task

- testing their resilience against well-known attacks is not sufficient;
- manual security analysis is error-prone.

$\longrightarrow$ **Caution:** Do not underestimate your opponents!

Verifying security protocols: a difficult task

- testing their resilience against well-known attacks is not sufficient;
- manual security analysis is error-prone.

$\longrightarrow$ **Caution:** Do not underestimate your opponents!

Security

Defects in e-passports allow real-time tracking

This threat brought to you by RFID

The register - Jan. 2010

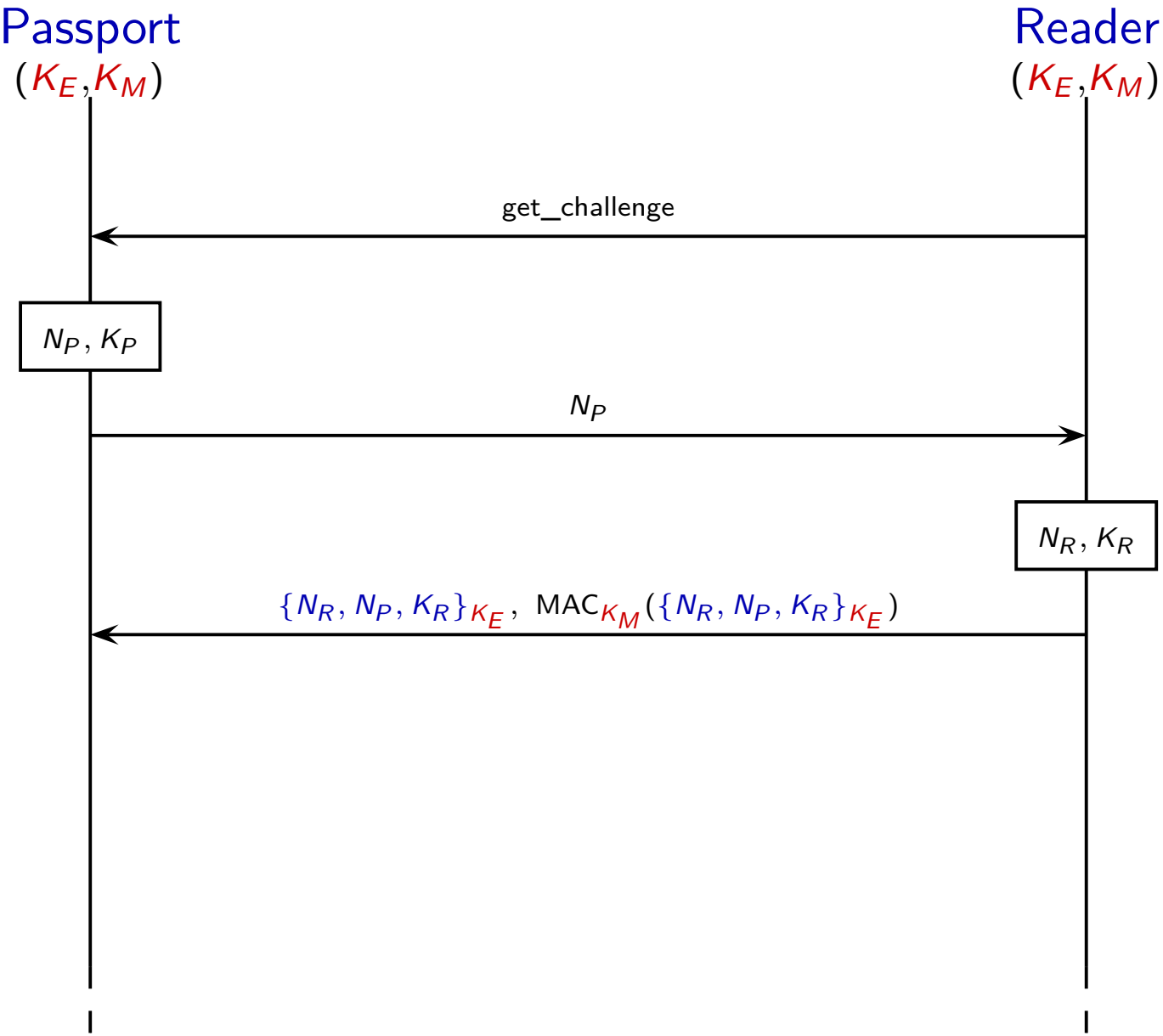
Lifestyle › Tech › News

Contactless card theft: Users warned to watch out for 'digital pickpockets'

Independent - Feb. 2016

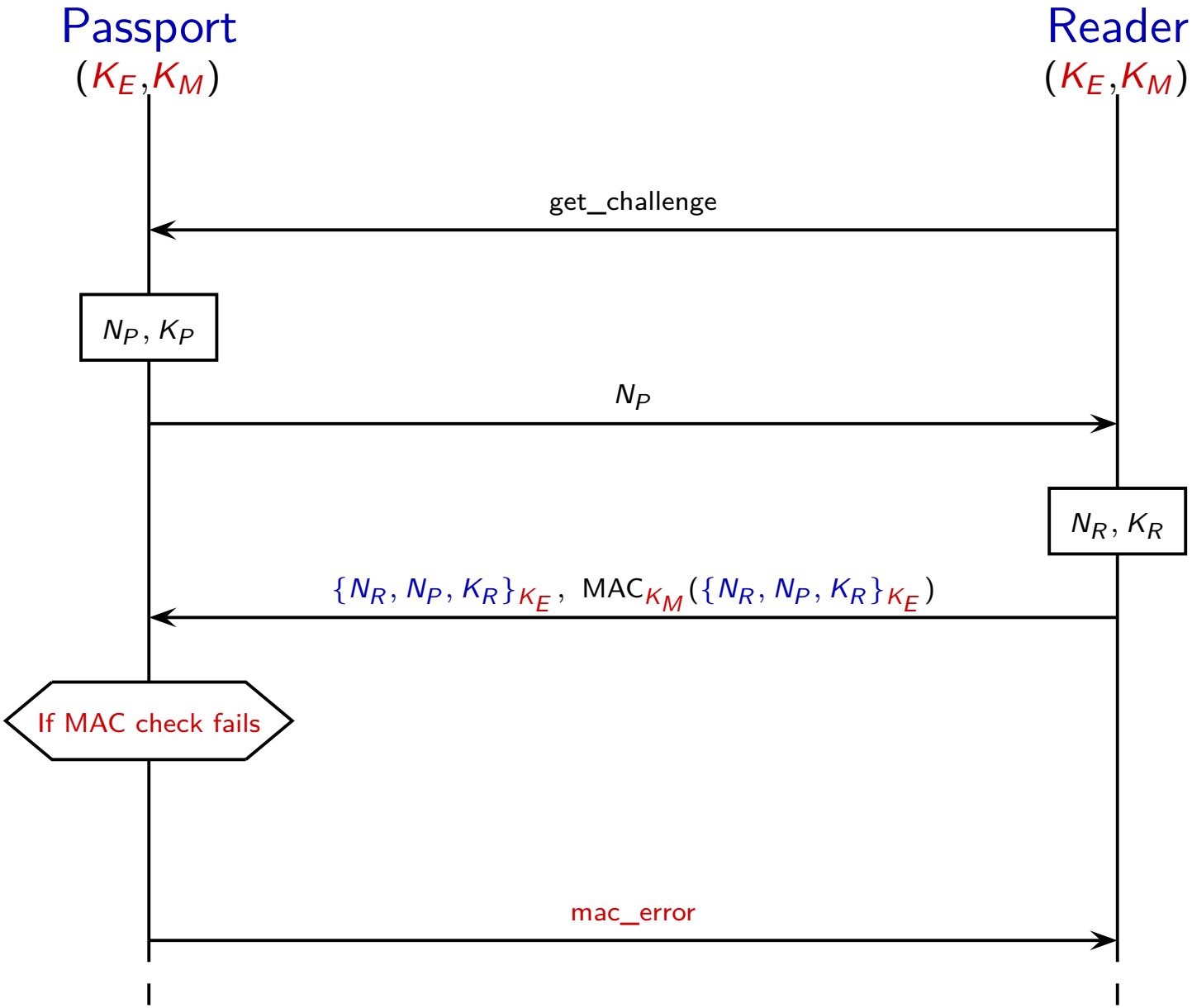
French electronic passport

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



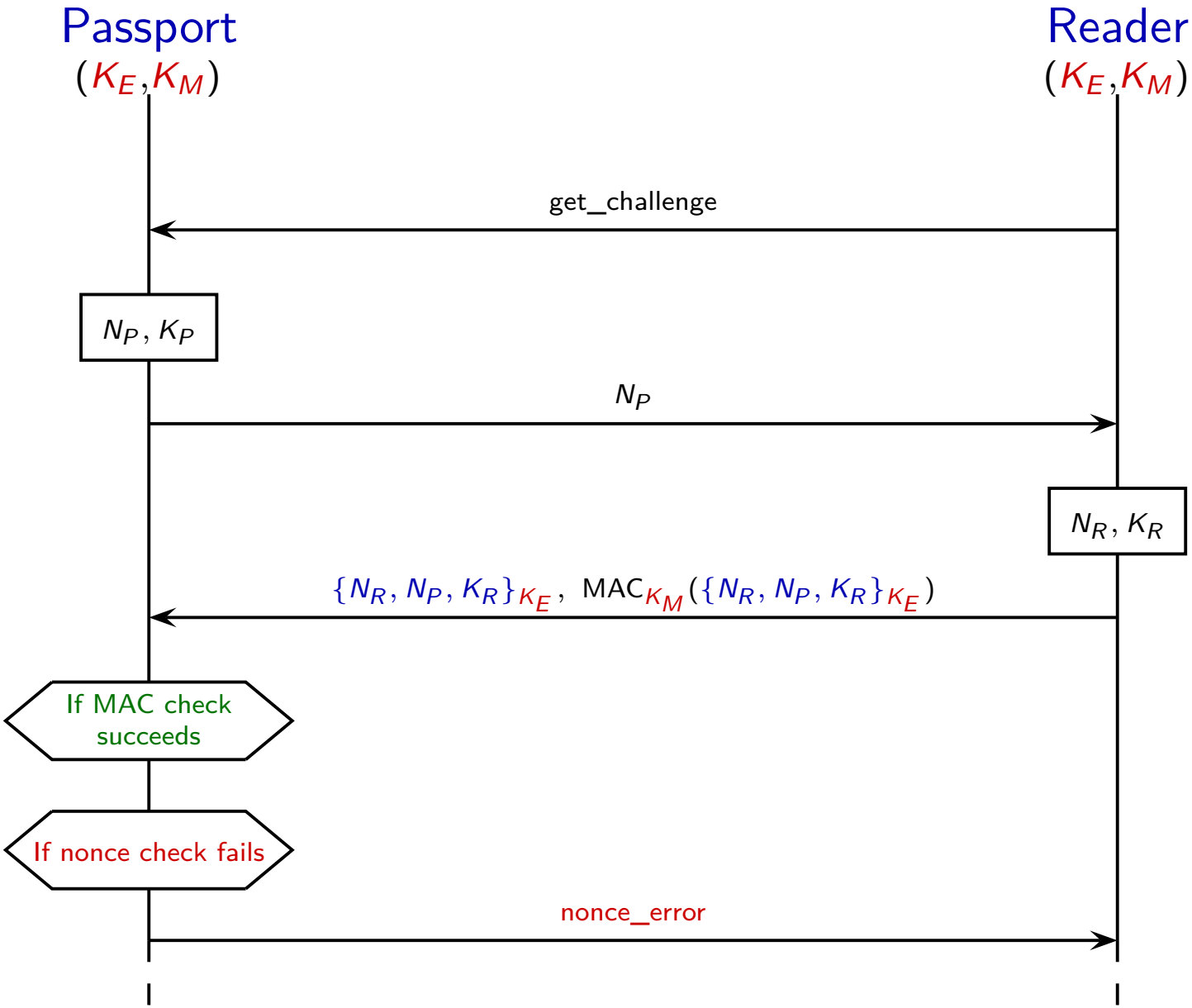
French electronic passport

$\longrightarrow$ 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]

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

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

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.

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

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

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.

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

Part 2 of the attack.

In presence of an unknown passport (K'_E, K'_M), the attacker replays the message M and checks the error code he receives.

1. MAC check failed: $K'_M \neq K_M \implies$ **????** is not Alice
2. MAC check succeeded: $K'_M = K_M \implies$ **????** is Alice

A sucessful approach: formal symbolic verification

$\longrightarrow$ provides a rigorous framework and automatic tools to analyse security protocols and find their flaws.

Example: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail

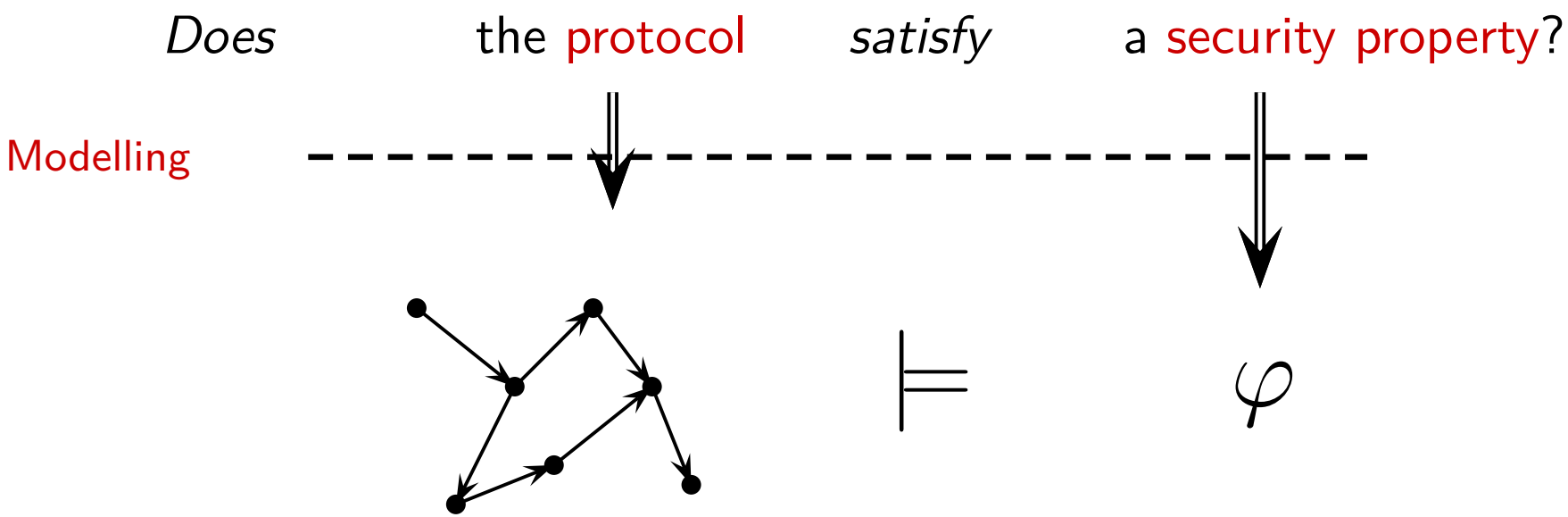
[Armando *et al.* (2011)] using SATMC (Avantssar verification platform)

A sucessful approach: formal symbolic verification

$\longrightarrow$ provides a rigorous framework and automatic tools to analyse security protocols and find their flaws.

Example: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail

[Armando *et al.* (2011)] using SATMC (Avantssar verification platform)



A sucessful approach: formal symbolic verification

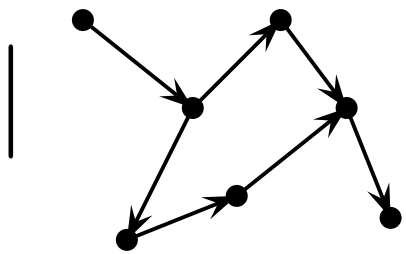
$\longrightarrow$ provides a rigorous framework and automatic tools to analyse security protocols and find their flaws.

Example: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail

[Armando *et al.* (2011)] using SATMC (Avantssar verification platform)

Does the protocol *satisfy* a security property?

Modelling



$\models$ φ

State of the art (in a nutshell)

for analysing confidentiality/authentication properties

Unbounded number of sessions

- undecidable in general [Even & Goldreich, 83; Durgin *et al*, 99]
- decidable for restricted classes [Lowe, 99; Rammanujam & Suresh, 03]

Bounded number of sessions

- a decidability result (NP-complete)
  [Rusinowitch & Turuani, 01; Millen & Shmatikov, 01]

TAMARIN
Tamarin prover interactive mode

ProVerif

AVANTSSAR

Main limitations of existing verifcation tools

- They are not suitable to analyse privacy-type properties.
  $\longrightarrow$ unlinkability, anonymity, vote-privacy ...
- They do not allow one to reason modulo the algebraic properties of some primitives.
  $\longrightarrow$ exclusive or, homomorphic encryption, ...
- They do not allow to take physical properties into account.
  $\longrightarrow$ transmission delay, location of participants, network topology

Main limitations of existing verifcation tools

- They are not suitable to analyse privacy-type properties.
  $\longrightarrow$ unlinkability, anonymity, vote-privacy ...
- They do not allow one to reason modulo the algebraic properties of some primitives.
  $\longrightarrow$ exclusive or, homomorphic encryption, ...
- They do not allow to take physical properties into account.
  $\longrightarrow$ transmission delay, location of participants, network topology

These features are important for analysing **contactless systems**!

POPSTAR (janv. 2017- déc. 2021)

Reasoning about Physical properties
Of security Protocols
with an Application To contactless Systems

Main limitations of existing verifcation tools

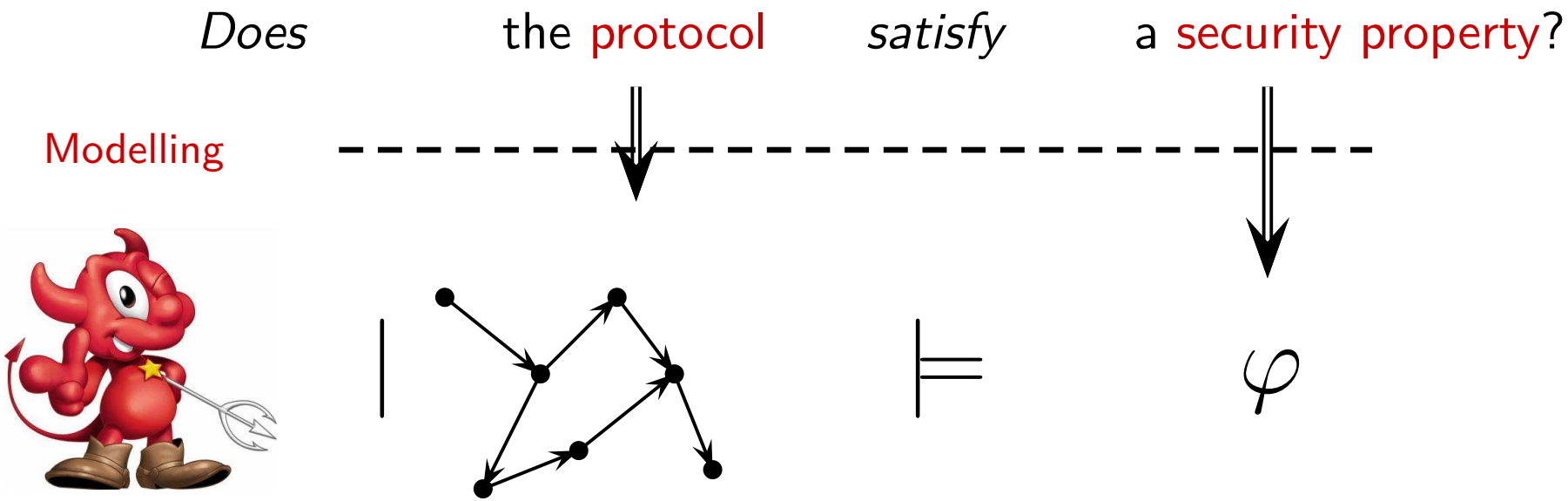
- They are not suitable to analyse privacy-type properties.
  $\longrightarrow$ unlinkability, anonymity, vote-privacy ...
- They do not allow one to reason modulo the algebraic properties of some primitives.
  $\longrightarrow$ exclusive or, homomorphic encryption, ...
- They do not allow to take physical properties into account.
  $\longrightarrow$ transmission delay, location of participants, network topology

These features are important for analysing **contactless systems**!

POPSTAR (janv. 2017- déc. 2021)

Reasoning about Physical properties
Of security Protocols
with an Application To contactless Systems

Outline



Outline of the remaining of this talk

1. Modelling: protocols, security properties, and the attacker !
2. Designing verification algorithms

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

Part I

Modelling: protocols, security properties, and the attacker

Protocols as processes

Applied pi calculus: basic programming language with constructs for concurrency and communication [Abadi & Fournet, 01]

$\longrightarrow$ based on the π-calculus [Milner *et al.*, 92] ...

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

Protocols as processes

Applied pi calculus: basic programming language with constructs for concurrency and communication [Abadi & Fournet, 01]

$\longrightarrow$ based on the π-calculus [Milner *et al.*, 92] ...

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

... but messages that are exchanged are not necessarily atomic !

Messages as terms

Messages are abstracted by terms built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

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

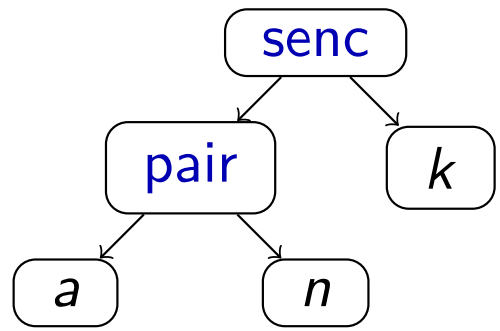
Messages as terms

Messages are abstracted by terms built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

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

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

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



Messages as terms

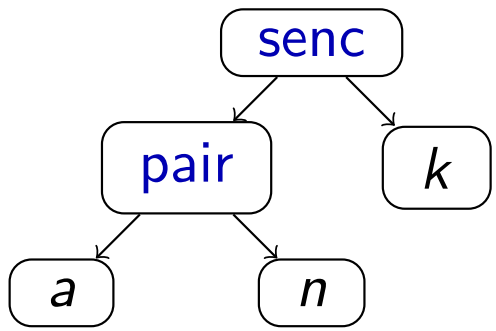
Messages are abstracted by terms built over a set of names $\mathcal{N}$, and a signature $\mathcal{F}$.

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

$\longrightarrow$ The term algebra is equipped with an equational theory E.

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

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



$\longrightarrow \mathrm{proj}_1(\mathrm{pair}(x, y)) = x, \quad \mathrm{proj}_2(\mathrm{pair}(x, y)) = y,$
$\mathrm{sdec}(\mathrm{senc}(x, y), y) = x.$

Semantics of the applied pi

How processes can evolve?

Basically, we have the following rules:

REPL	$!P \rightarrow P \mid !P$
NEW	$\text{new}\, n.P \rightarrow P\{n'/n\}$ where n' is a fresh name
COMM	$\text{out}(c, u).P \mid \text{in}(c, x).Q \rightarrow P \mid Q\{u/x\}$
THEN	if $u = v$ then P else $Q \rightarrow P$ when $u =_{\mathsf{E}} v$
ELSE	if $u = v$ then P else $Q \rightarrow Q$ when $u \neq_{\mathsf{E}} v$

Going back to the e-passport

Cryptographic primitives are modelled using function symbols

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

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

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

Going back to the e-passport

Cryptographic primitives are modelled using function symbols

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

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

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

Modelling Passport's role

$$P_{\mathsf{BAC}}(k_E, k_M) = \mathsf{new}\ n_P.\mathsf{new}\ k_P.\mathsf{out}(n_P).\mathsf{in}(\langle z_E, z_M\rangle).$$
$$\texttt{if}\ z_M = \mathsf{mac}(z_E, k_M)\ \texttt{then}\ \texttt{if}\ n_P = \mathsf{proj}_1(\mathsf{proj}_2(\mathsf{sdec}(z_E, k_E)))$$
$$\texttt{then}\ \mathsf{out}(\langle m, \mathsf{mac}(m, k_M)\rangle)$$
$$\texttt{else}\ \mathsf{out}(\mathit{nonce\_error})$$
$$\texttt{else}\ \mathsf{out}(\mathit{mac\_error})$$

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

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

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

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

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

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

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

Example 2: k and k' are known to the attacker

$$\begin{array}{c} \text{new } s.\text{out}(a, \text{senc}(s, k)).\text{out}(a, \text{senc}(s, k')) \\ \stackrel{?}{\approx} \\ \text{new } s, s'.\text{out}(a, \text{senc}(s, k)).\text{out}(a, \text{senc}(s', k')) \end{array}$$

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

Example 2: k and k' are known to the attacker

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

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

Privacy-type security properties

$\longrightarrow$ they are modelled as equivalence-based properties

Testing equivalence $P \approx Q$

For all processes A, we have that:

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

$R \Downarrow_c$ means that R can evolve and emits on public channel c.

Example 3:

$$\text{new } s.\text{out}(a, s) \approx \text{new } s.\text{new } k.\text{out}(a, \text{senc}(s, k))$$

Some privacy-type properties

Unlinkability

[Arapinis et al, 2010]

$$!\text{new } ke.\text{new } km.(!P_{\text{BAC}} \mid !R_{\text{BAC}}) \approx !\text{new } ke.\text{new } km.(P_{\text{BAC}} \mid !R_{\text{BAC}})$$

many sessions for each passport

only one session for each passport

Some privacy-type properties

Unlinkability

[Arapinis et al, 2010]

$$!\text{new } ke.\text{new } km.(!P_{\text{BAC}} \mid !R_{\text{BAC}}) \approx !\text{new } ke.\text{new } km.(P_{\text{BAC}} \mid !R_{\text{BAC}})$$

many sessions for each passport

only one session for each passport

Vote privacy

[Kremer and Ryan, 2005]

$$S[V_A(yes) \mid V_B(no)] \approx S[V_A(no) \mid V_B(yes)]$$

A votes yes
B votes no

A votes no
B votes yes

Part II

How can we check testing equivalence?

Some recent theoretical results

$\longrightarrow$ [Chrétien PhD thesis, 16]

- 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 an interesting subclass of tagged protocols (with nonces) [CSF'15]

Some recent theoretical results

$\longrightarrow$ [Chrétien PhD thesis, 16]

- 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 an interesting subclass of tagged protocols (with nonces) [CSF'15]

Main limitations:

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

A more pragmatic approach

$\longrightarrow$ [Blanchet *et al.*, LICS'05]

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

- processes with replication;
- 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.

A more pragmatic approach

$\longrightarrow$ [Blanchet *et al.*, LICS'05]

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

- processes with replication;
- 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.

Main issue: diff-equivalence is too strong in many situations.
$\longrightarrow$ ProVerif is not suitable to analyse vote-privacy, or unlinkability of the BAC protocol.

Some extensions have been recently developed to go beyond diff-equivalence, e.g. vote-privacy [Blanchet & Smyth, 16], unlinkability [Hirschi et al, 16].

Another decidability result (bounded number of sessions)

$\longrightarrow$ [Cheval PhD thesis, 12]

Main result

A procedure for deciding testing equivalence for a large class of processes implemented in a tool called APTE

The class of processes:

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

Another decidability result (bounded number of sessions)

$\longrightarrow$ [Cheval PhD thesis, 12]

Main result

A procedure for deciding testing equivalence for a large class of processes implemented in a tool called APTE

The class of processes:

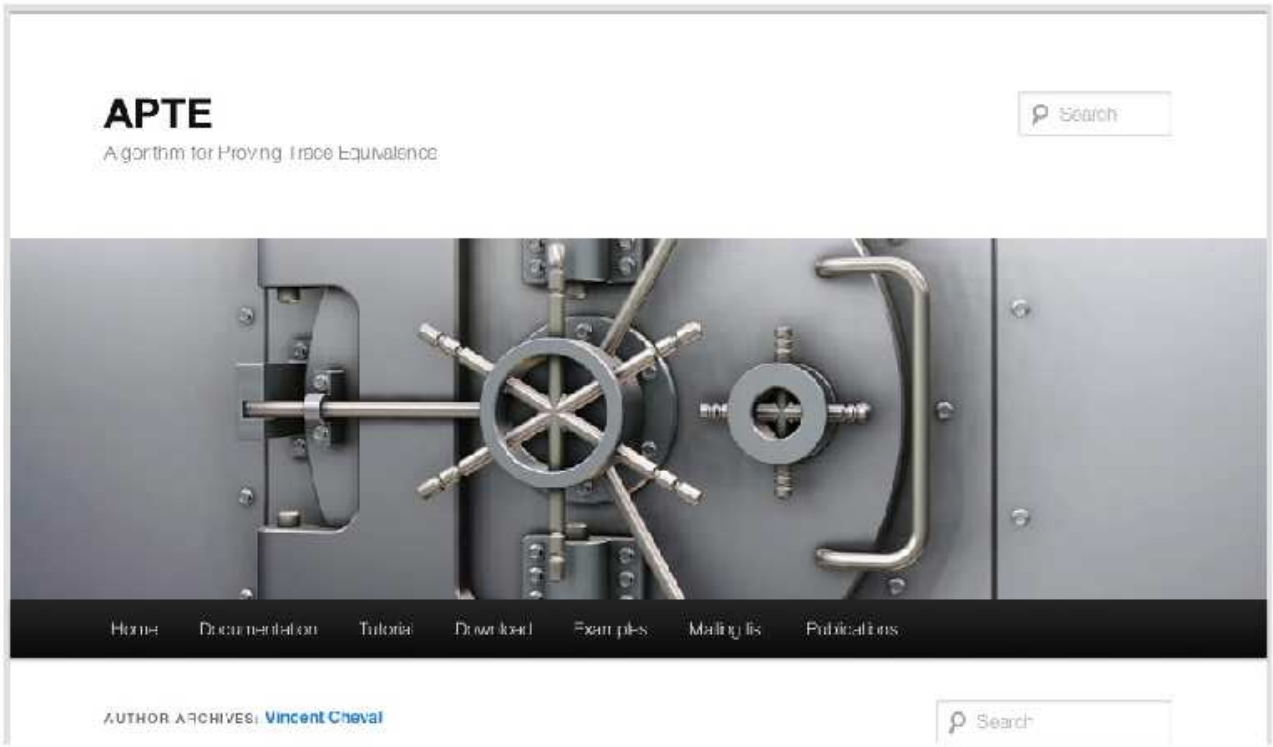
- $+$ non-trivial else branches, private channels, and non-deterministic choice;
- $-$ but no replication, and a fixed set of cryptographic primitives (signature, symmetric and asymmetric encryptions, hash function, mac, pairs).

Similar results for restricted class of processes have been obtained in [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], [Chadha *et al.*, 12], ...

APTE- Algorithm for Proving Trace Equivalence

`http://projects.lsv.ens-cachan.fr/APTE` (Ocaml - 12 KLocs)

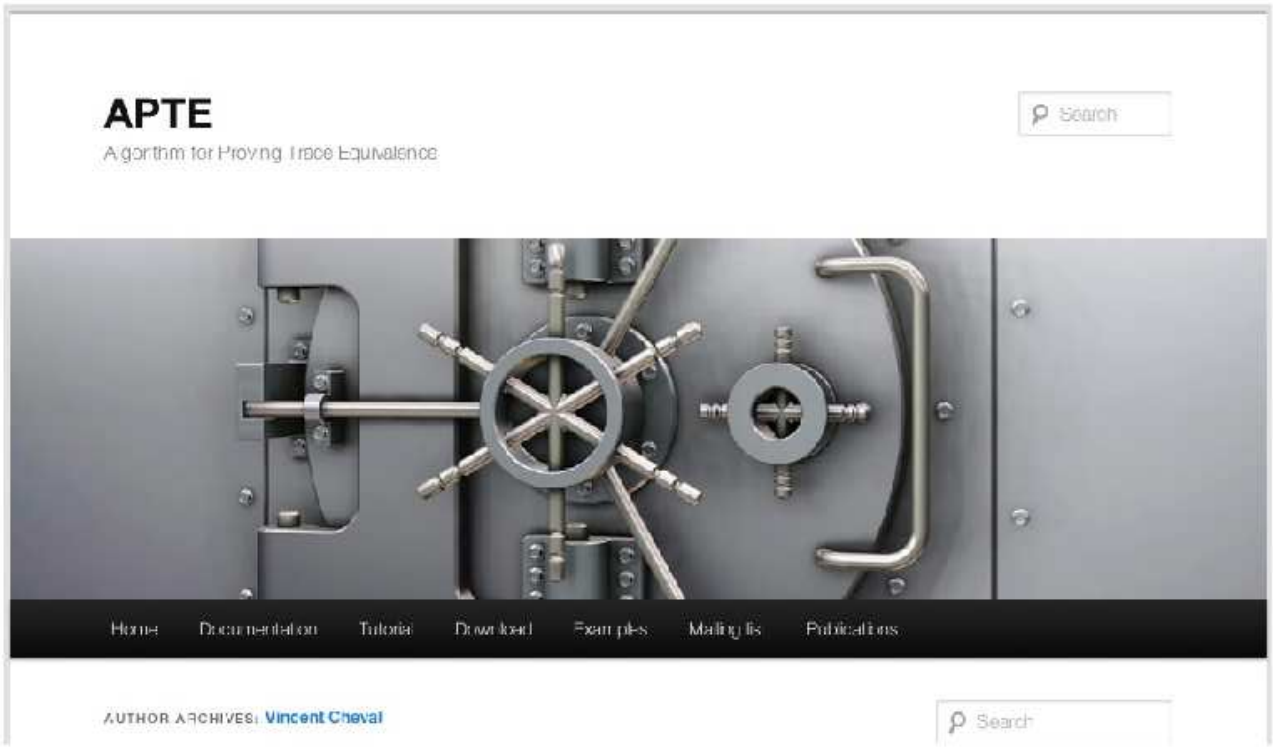
$\longrightarrow$ developed by Vincent Cheval [Cheval, TACAS'14]



APTE- Algorithm for Proving Trace Equivalence

`http://projects.lsv.ens-cachan.fr/APTE` (Ocaml - 12 KLocs)

$\longrightarrow$ developed by Vincent CHEVAL [Cheval, TACAS'14]



$\longrightarrow$ but a limited practical impact because it scales badly

Conclusion

POPSTAR in a nutshell

PI: Stéphanie Delaune, **CNRS**, France

Reasoning about **Physical properties** Of
security Protocols
with an Application To **contactless Systems**

Main issues:

- specificities of contactless systems are not well understood;
- a lack of formal model to reason about these systems.

Main outcomes:

- solid foundations to reason about physical properties;
- new algorithms and tools to analyse the security and privacy of modern protocols;
- make the upcoming generation of nomadic contactless devices more secure.