# Verification of security protocols: from confidentiality to privacy

Stéphanie Delaune

Univ Rennes, CNRS, IRISA, France

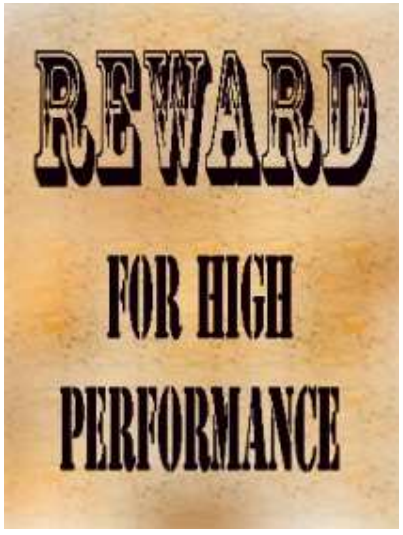
Thursday, June 28th, 2018



# Challenge (1/2)

Would you be able to find the attack on the well-known Needham-Schroeder protocol (1978)?

$$
\begin{aligned}
A \to B: &\quad \{A, N_a\}_{\mathrm{pub}(B)} \\
B \to A: &\quad \{N_a, N_b\}_{\mathrm{pub}(A)} \\
A \to B: &\quad \{N_b\}_{\mathrm{pub}(B)}
\end{aligned}
$$




Questions

- Is $N_b$ secret between $A$ and $B$ ?
- When $B$ receives $\{N_b\}_{\mathrm{pub}(B)}$, does this message really comes from $A$ ?

# Challenge (2/2)

**An attack has been found 17 years after the publication of this protocol !**

Man in the middle attack due to G. Lowe 1995

- involving 2 sessions in parallel,
- an honest agent has to initiate a session with C.

Fixed version of the protocol

$$\begin{aligned} \mathsf{A} \rightarrow \mathsf{B} &: \{A, N_a\}_{\mathsf{pub}(B)} \\ \mathsf{B} \rightarrow \mathsf{A} &: \{N_a, N_b, B\}_{\mathsf{pub}(A)} \\ \mathsf{A} \rightarrow \mathsf{B} &: \{N_b\}_{\mathsf{pub}(B)} \end{aligned}$$

$\longrightarrow$ the responder's identity has been added to the second message

# Security protocols everywhere !



**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.

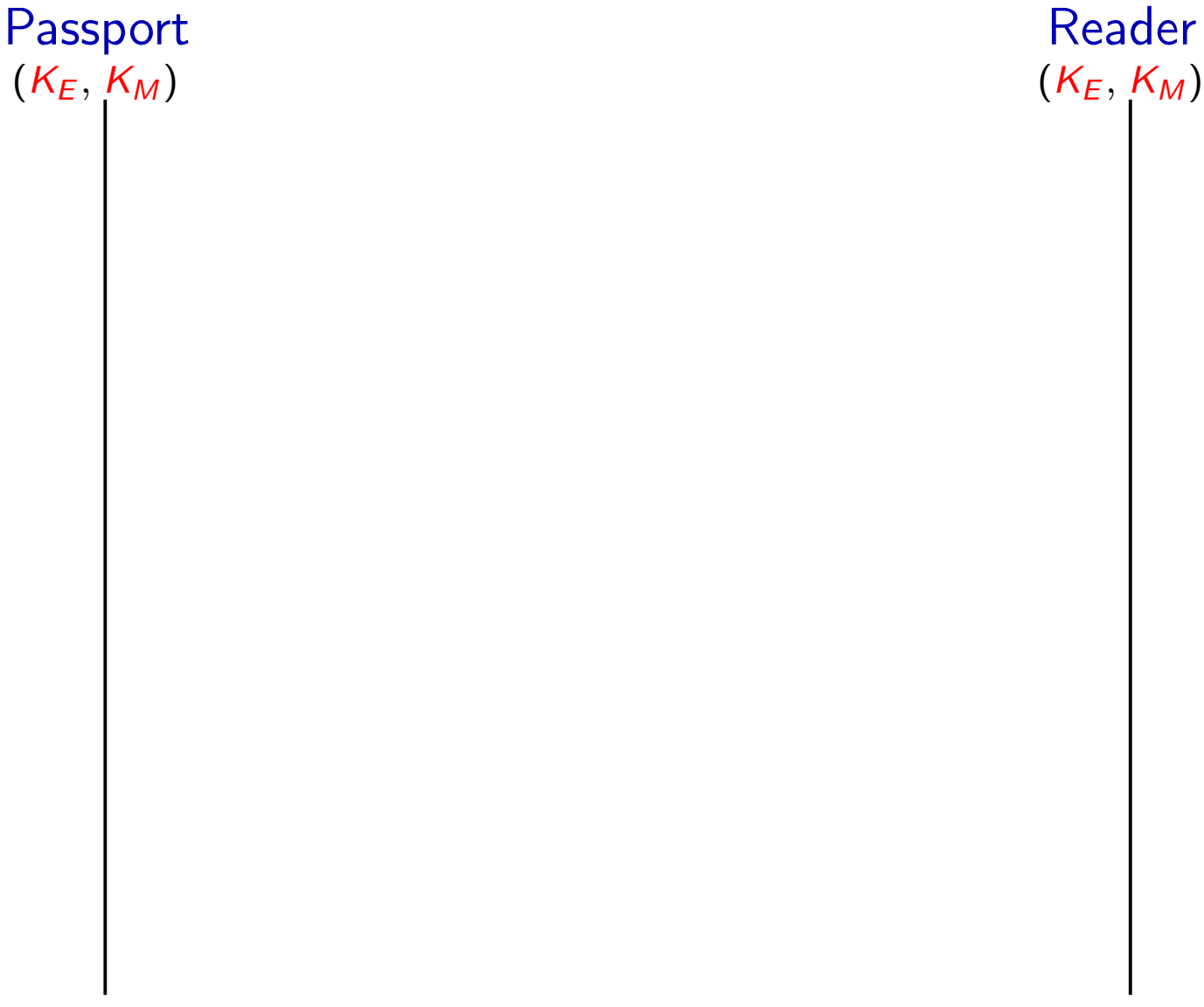
The RFID tag stores:

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

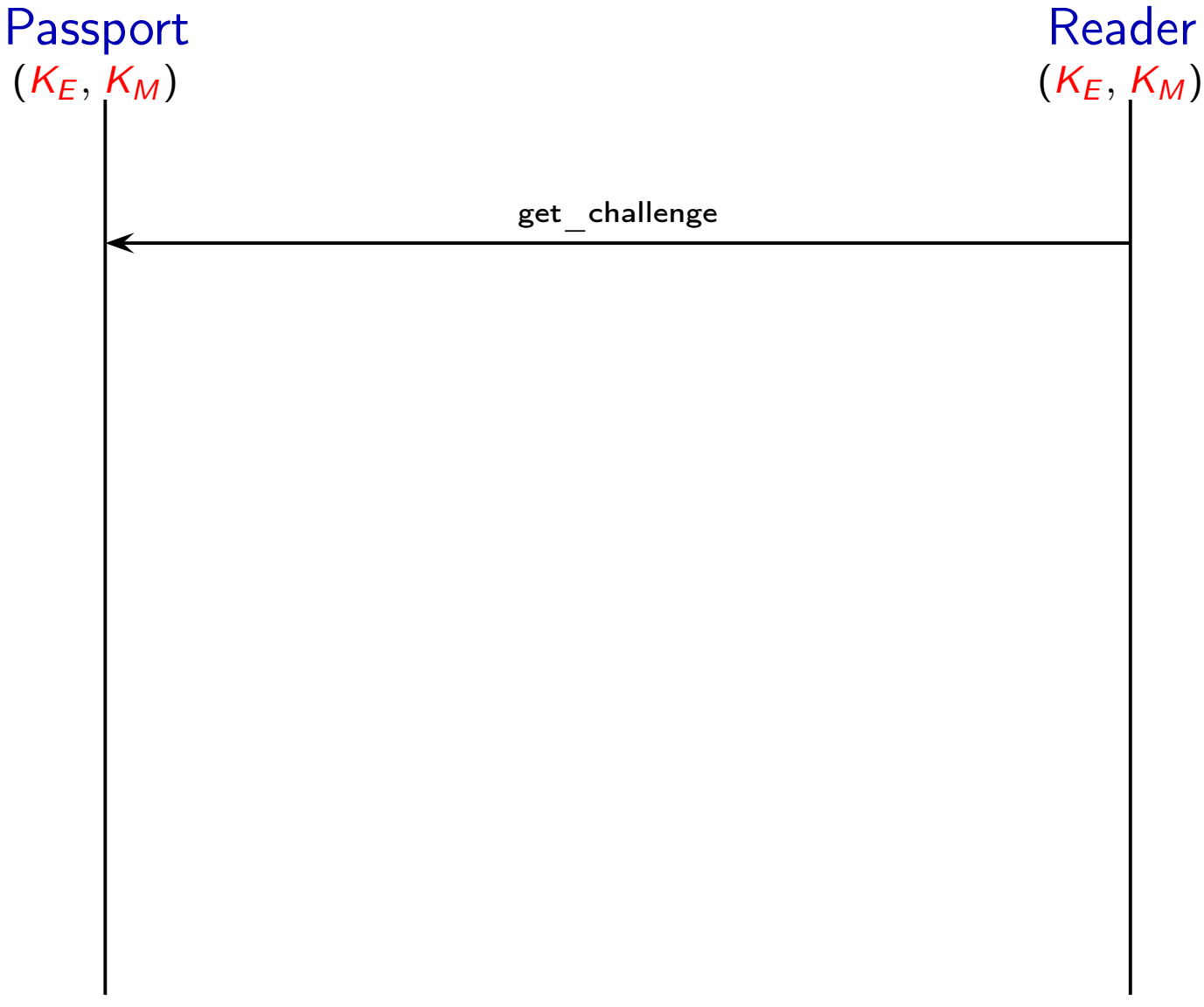
The Basic Access Control (BAC) protocol is a key establishment protocol that has been designed to protect our personnal data, and to ensure unlinkability.

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]

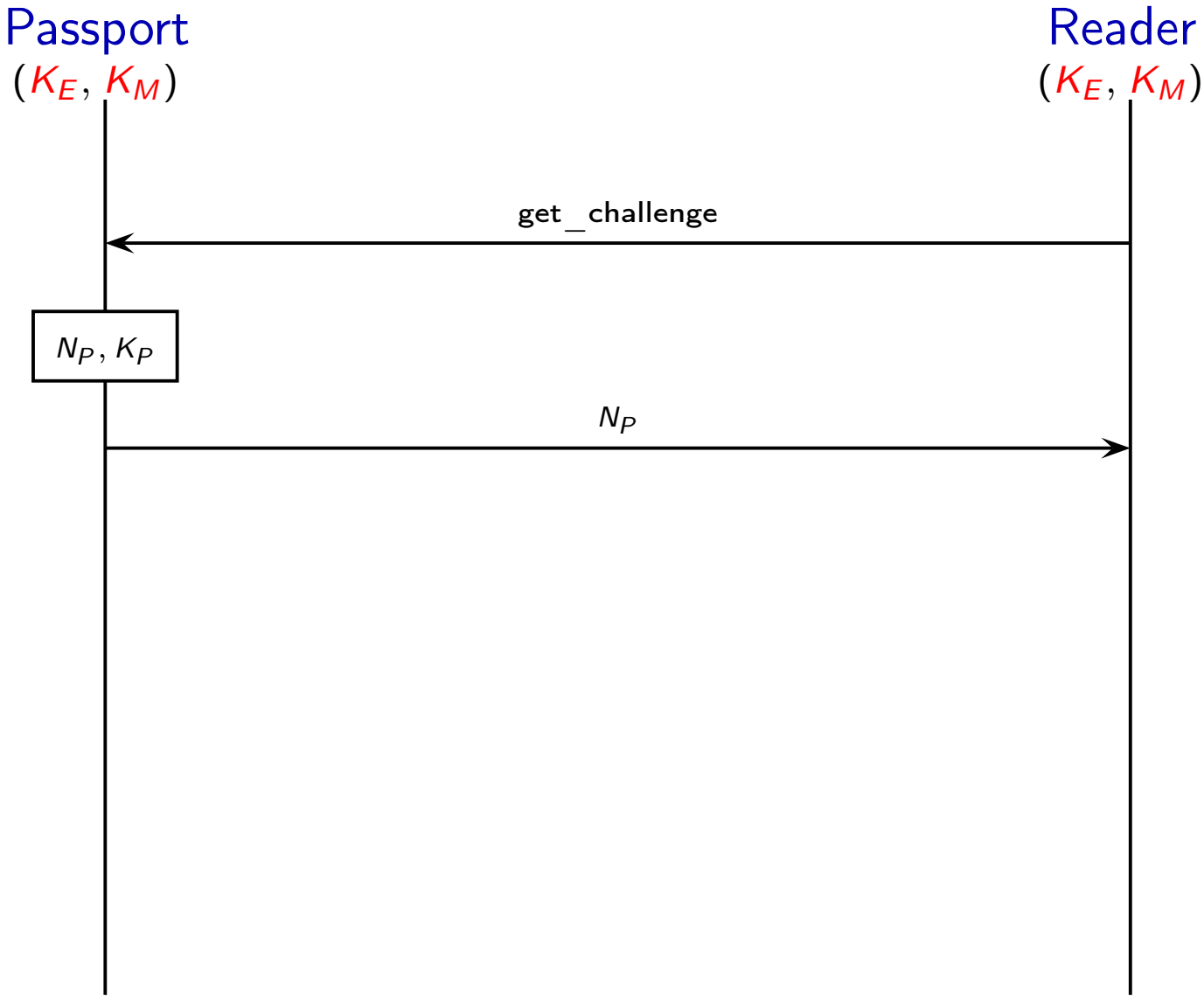
# BAC protocol



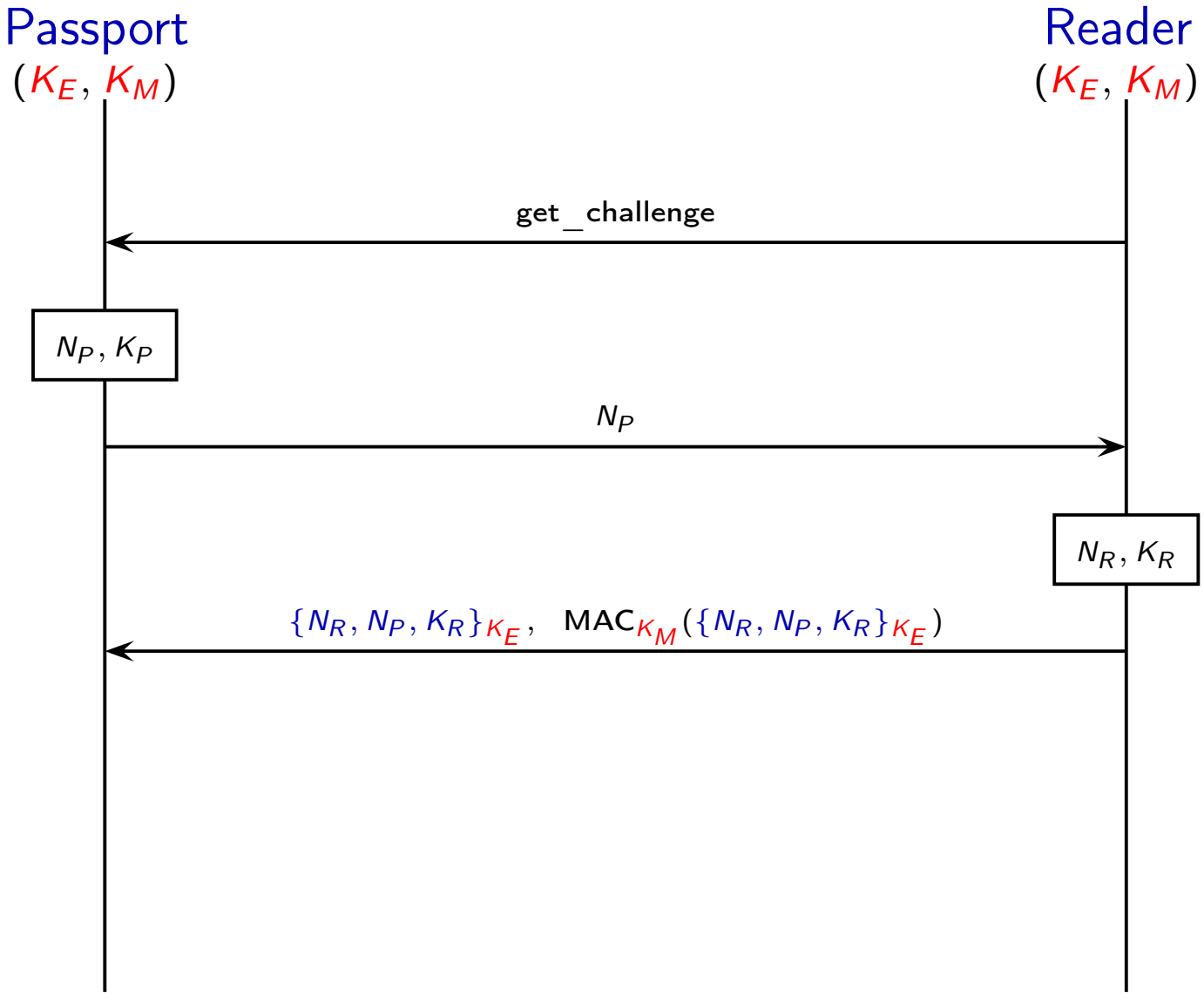
# BAC protocol



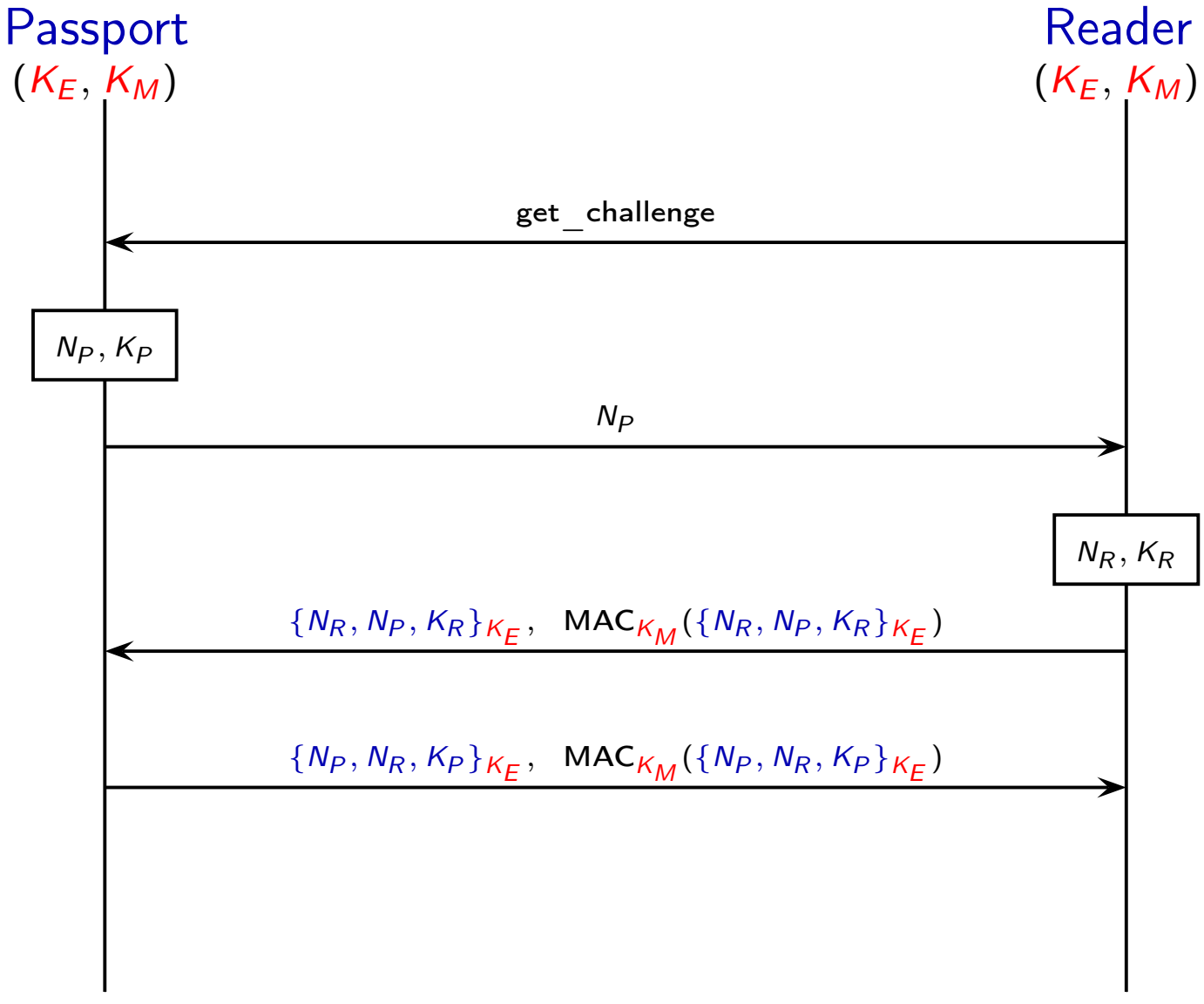
# BAC protocol



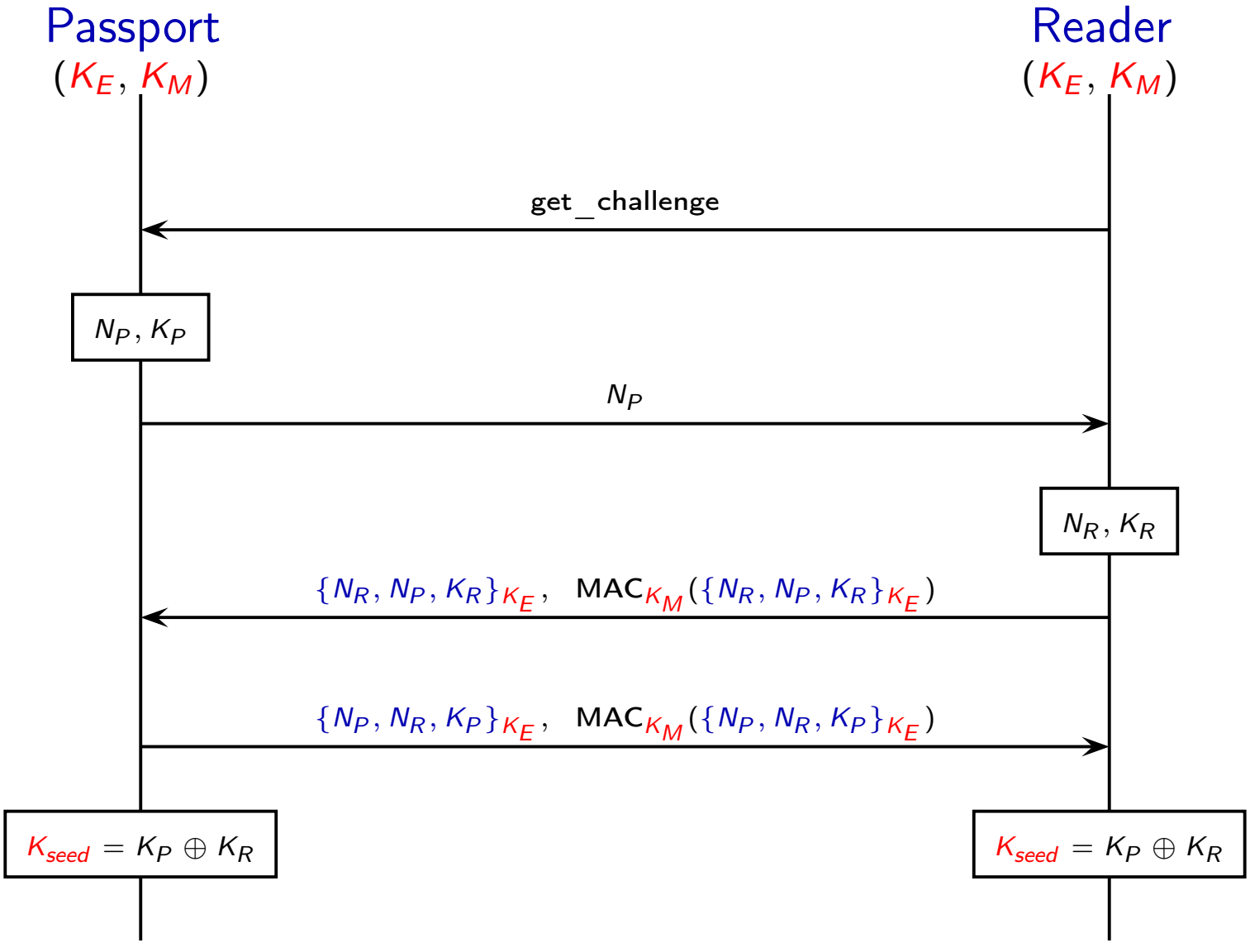
# BAC protocol



# BAC protocol



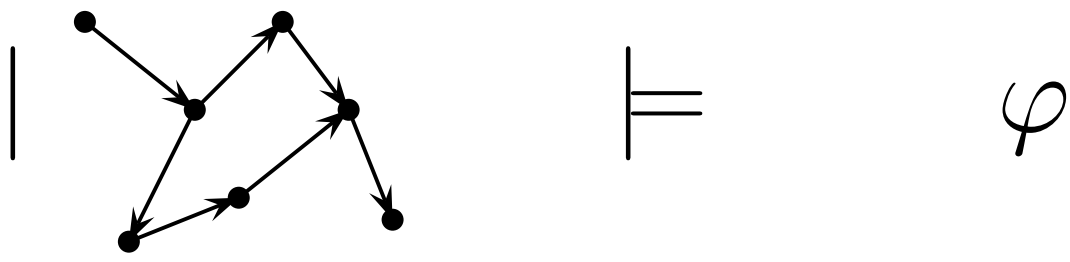
# BAC protocol



# A brief recap

*Does* the protocol *satisfy* a security property?

Modelling




How can we check privacy-type security properties?

# Part I

Modelling protocols, **security properties**

and the attacker

# Messages as terms (on an example)

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


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: $\mathsf{mac}/2$

Properties of the primitives are modelled using an equational theory.

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

# Protocols as processes (on an example)

$$
\begin{array}{lll}
P \to R: & N_p & \\
R \to P: & \{N_R, N_P, K_R\}_{K_E}, & \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E}) \\
P \to R: & \{N_P, N_R, K_P\}_{K_E}, & \mathrm{MAC}_{K_M}(\{N_P, N_R, K_P\}_{K_E})
\end{array}
$$

## Modelling Passport's role

$$
\begin{aligned}
P_{\mathrm{BAC}}(k_E, k_M) = {} & \mathsf{new}\ n_P.\mathsf{new}\ k_P.\mathsf{out}(n_P).\mathsf{in}(\langle z_E, z_M\rangle). \\
& \mathtt{if}\ z_M = \mathsf{mac}(z_E, k_M)\ \mathtt{then}\ \mathtt{if}\ n_P = \mathsf{proj}_1(\mathsf{proj}_2(\mathsf{sdec}(z_E, k_E))) \\
& \qquad \mathtt{then}\ \mathsf{out}(\langle m, \mathsf{mac}(m, k_M)\rangle) \\
& \qquad \mathtt{else}\ 0 \\
& \mathtt{else}\ 0
\end{aligned}
$$

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

# What does unlinkability mean?

Informally, an 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 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,

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

many sessions for each passport

only one session for each passport

(we still have to formalize the notion of equivalence)

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$

for all processes $A$, we have that:

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

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

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$
for all processes $A$, we have that:

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

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

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

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$
for all processes $A$, we have that:

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

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

Exercise 1: $\quad \mathsf{out}(a, \mathit{yes}) \not\approx \mathsf{out}(a, \mathit{no})$

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

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$
for all processes $A$, we have that:

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

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

Exercise 2: $k$ and $k'$ are known to the attacker

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

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$
for all processes $A$, we have that:

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

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

Exercise 2: $k$ and $k'$ are known to the attacker

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

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

# Security properties - privacy

Privacy-type properties are modelled relying on testing equivalence.

Testing equivalence between $P$ and $Q$, denoted $P \approx Q$
for all processes $A$, we have that:

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

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

Exercise 3: Are the two following processes in testing equivalence?

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

# Some other equivalence-based security properties

The notion of testing equivalence can be used to express:

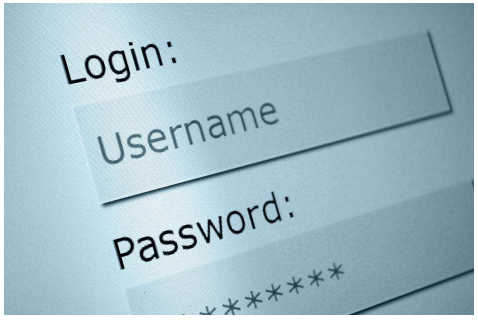
## Vote privacy

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

## privacy-type properties

# State of the art for testing equivalence (no !)

**for analysing testing equivalence**
bounded number of sessions

# State of the art for testing equivalence (no !)

**for analysing testing equivalence**
bounded number of sessions

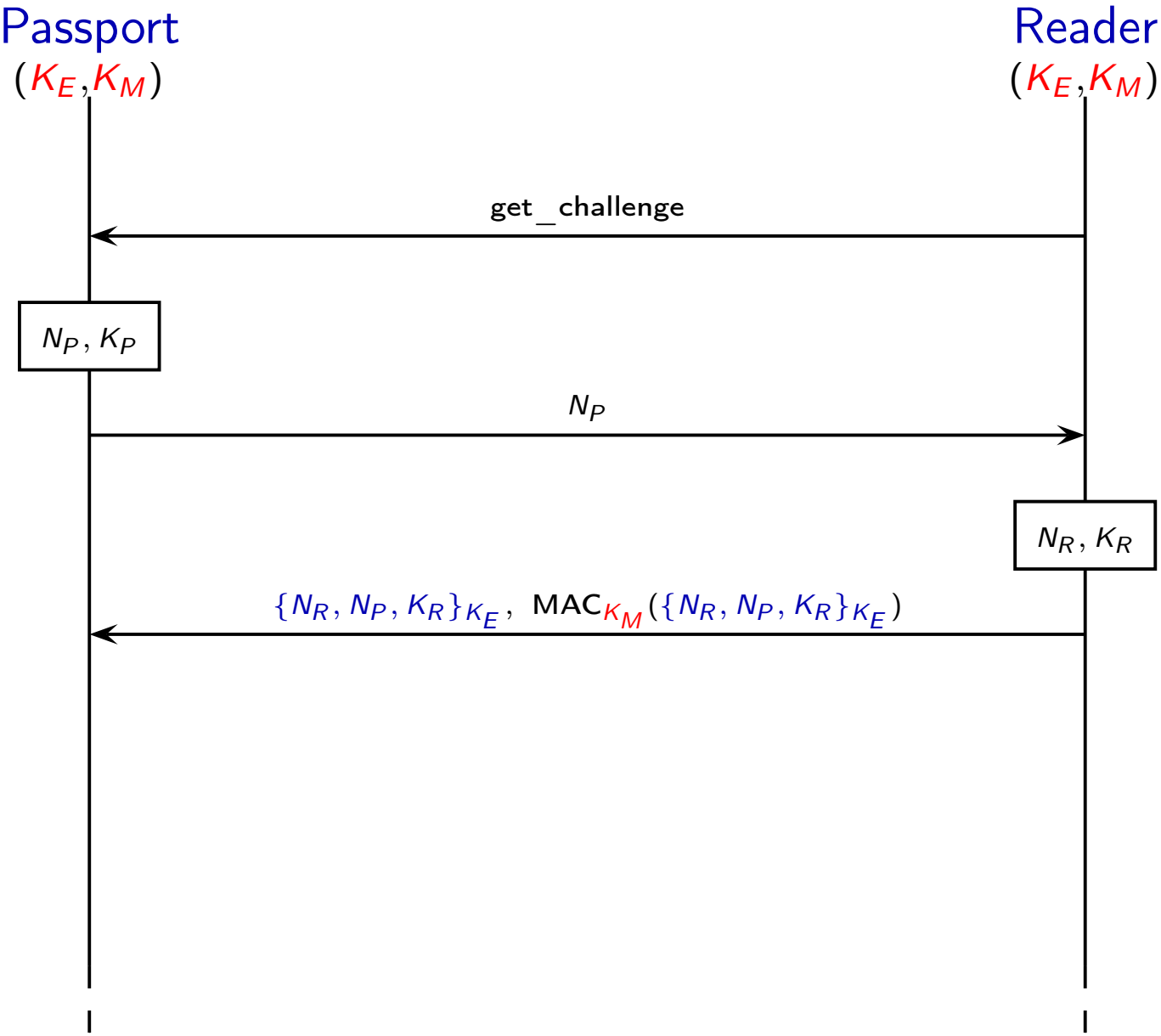
Some important results:

- 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]

$\longrightarrow$ A decision procedure implemented in the tool DEEPSEC
[Cheval, Kremer & Rakotonirina, 2018]

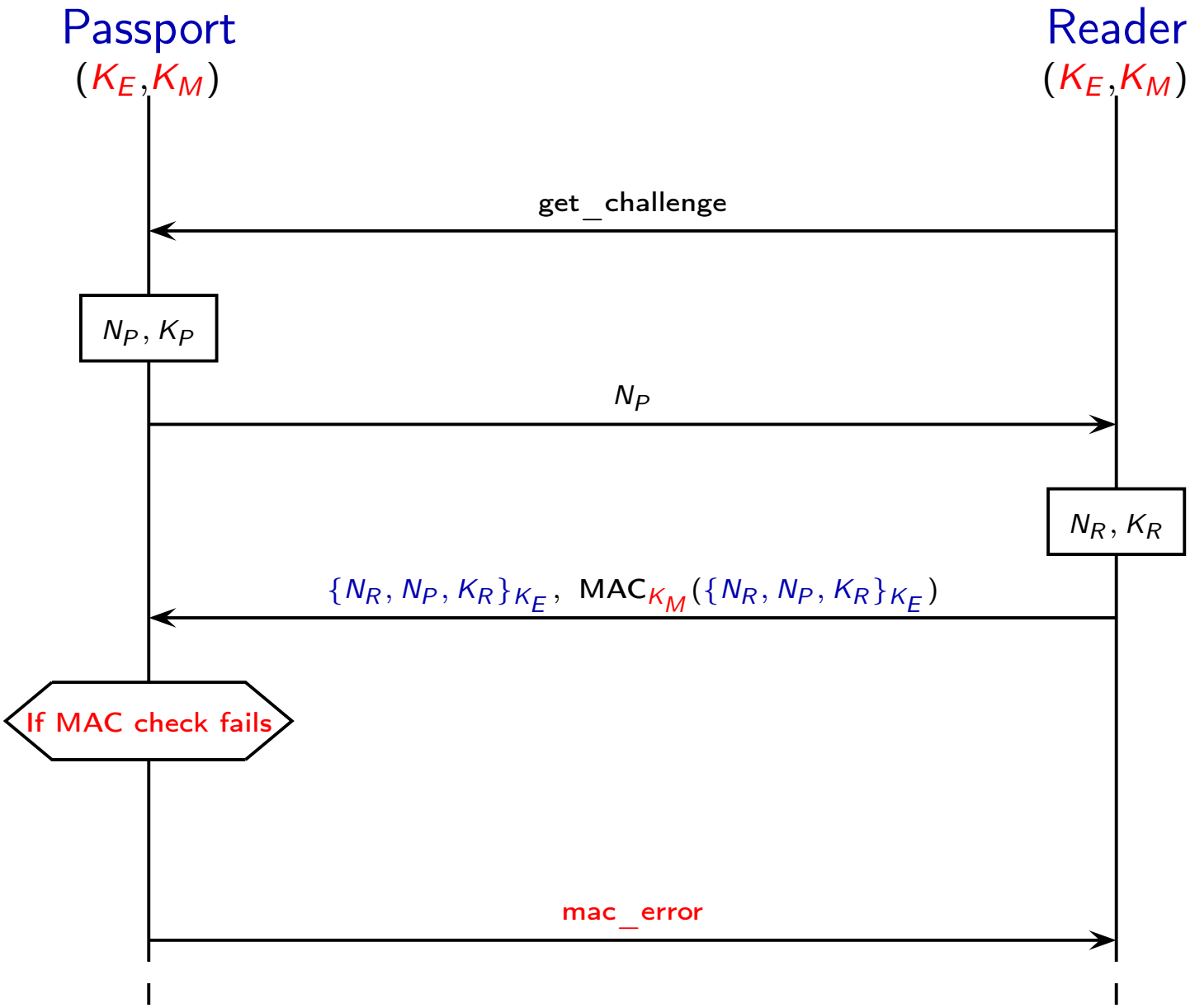
# French electronic passport

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



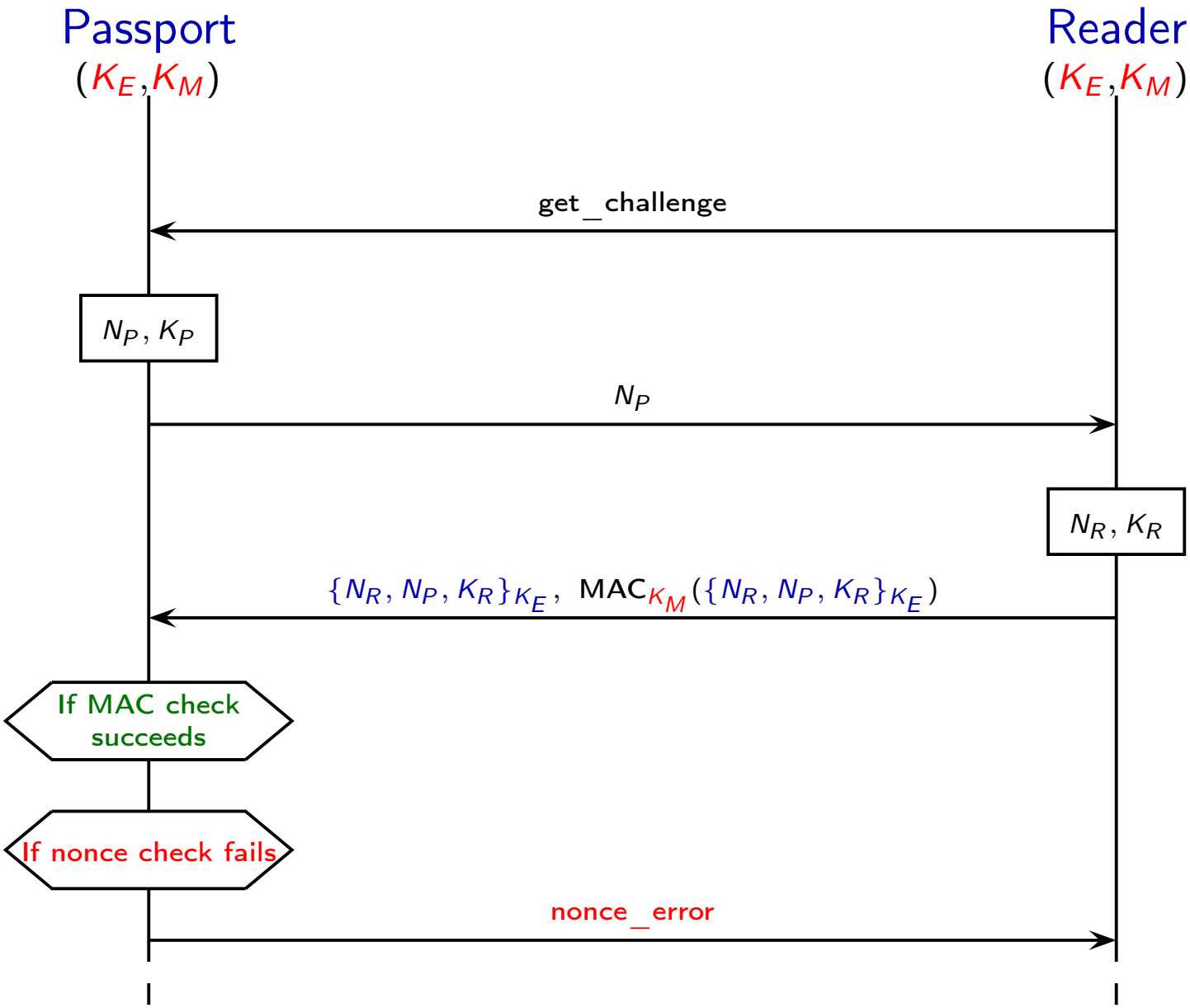
# French electronic passport

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



# French electronic passport

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



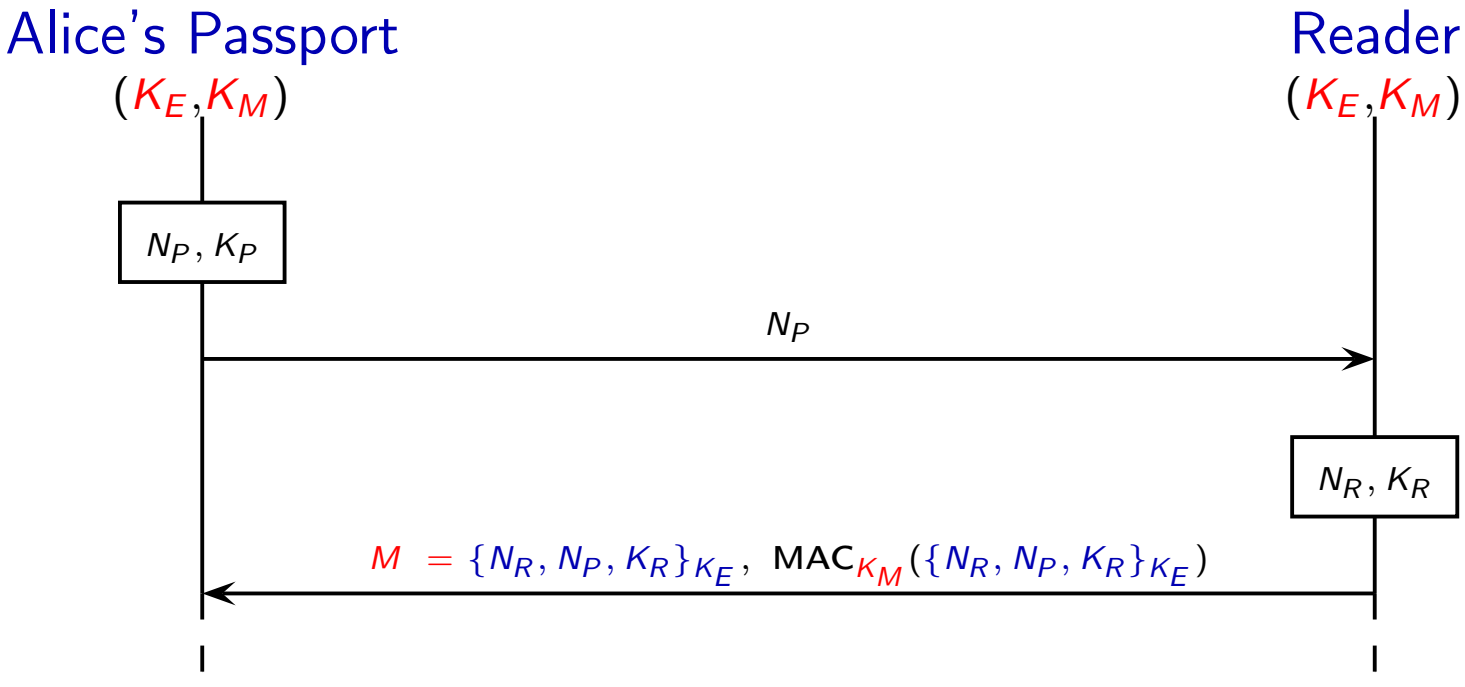
# An attack on the French passport [Chothia & Smirnov, 10]

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 $M$.

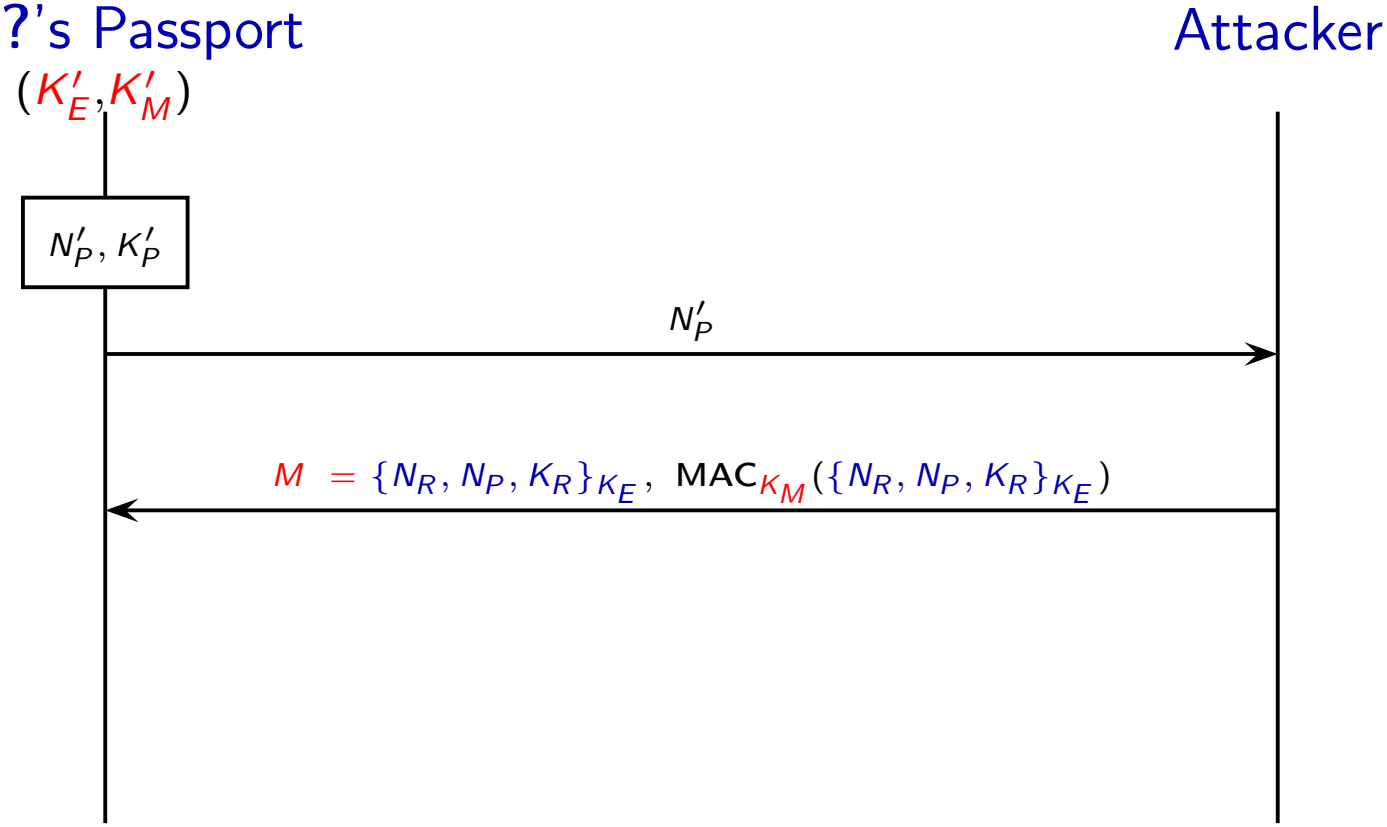


# An attack on the French passport [Chothia & Smirnov, 10]

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

Part 2 of the attack.

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

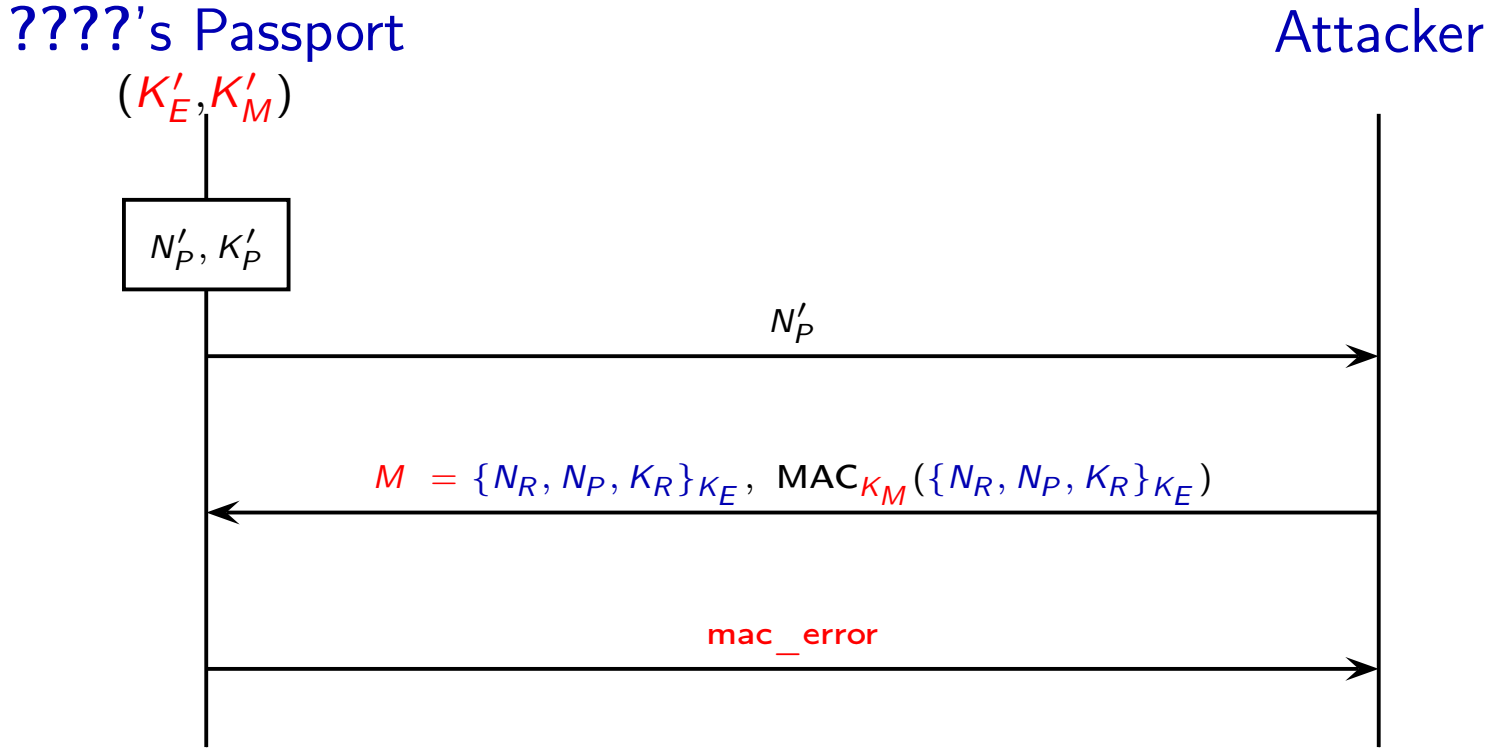


# An attack on the French passport [Chothia & Smirnov, 10]

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

Part 2 of the attack.

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



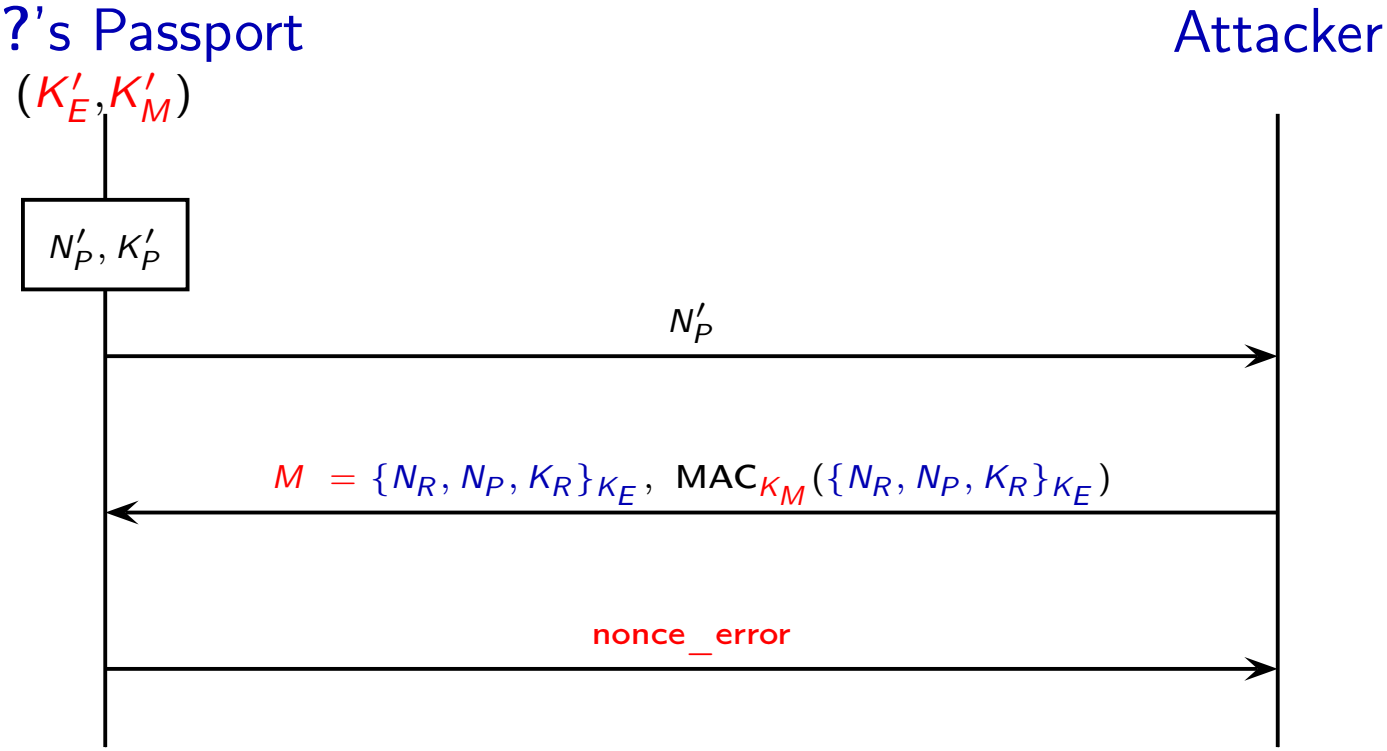

$\Longrightarrow$ MAC check failed $\Longrightarrow$ $K'_M \neq K_M$ $\Longrightarrow$ **????** is not Alice

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

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

Part 2 of the attack.

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




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

# State of the art for testing equivalence (with !)

**for analysing testing equivalence**

unbounded number of sessions

# State of the art for testing equivalence (with !)

**for analysing testing equivalence**
unbounded number of sessions

- undecidable in general even for some fragment for which confidentiality is decidable [Chrétien, Cortier & D., 13]
- some recent decidability results for some restricted fragment *e.g.* tagged protocols, no nonces, a particular set of primitives ... [Chrétien, Cortier & D., Icalp'13, Concur'14, CSF'15]
- some existing verification tools: ProVerif, Tamarin, ... for analysing the notion of diff-equivalence (stronger than testing equivalence) [Blanchet, Abadi & Fournet, 05] [Basin, Dreier & Sasse, 15]

None of these results is suitable to analyse vote-privacy, or unlinkability of the BAC protocol.

# Diff-equivalence is often too strong in practice

Vote privacy

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


$$V_A(\mathit{yes}) \mid V_B(\mathit{no}) \approx V_A(\mathit{no}) \mid V_B(\mathit{yes})$$

$\longrightarrow$ ProSwapper extension [Blanchet & Smyth, 2016]

# Diff-equivalence is often too strong in practice

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


$$V_A(\mathit{yes}) \mid V_B(\mathit{no}) \approx V_A(\mathit{no}) \mid V_B(\mathit{yes})$$

$\longrightarrow$ ProSwapper extension [Blanchet & Smyth, 2016]


Unlinkability a user may make multiple uses of a resource without other being able to link these uses together.

$$!\,\mathsf{new}\ k.!P \ \approx !\,\mathsf{new}\ k.P$$

$\longrightarrow$ UKANO extension [Hirschi, Baelde, & D, 2016]

# UKANO extension (1/2) [Hirschi, Baelde, & D, 2016]

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

# UKANO extension (1/2) [Hirschi, Baelde, & D, 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

# UKANO extension (2/2) – summary of our case studies

| 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/g̃londu/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 !

# 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
- 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

Questions ?