Formal verification of cryptographic protocols

Stéphanie **DELAUNE**

Univ Rennes, CNRS, IRISA

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 !

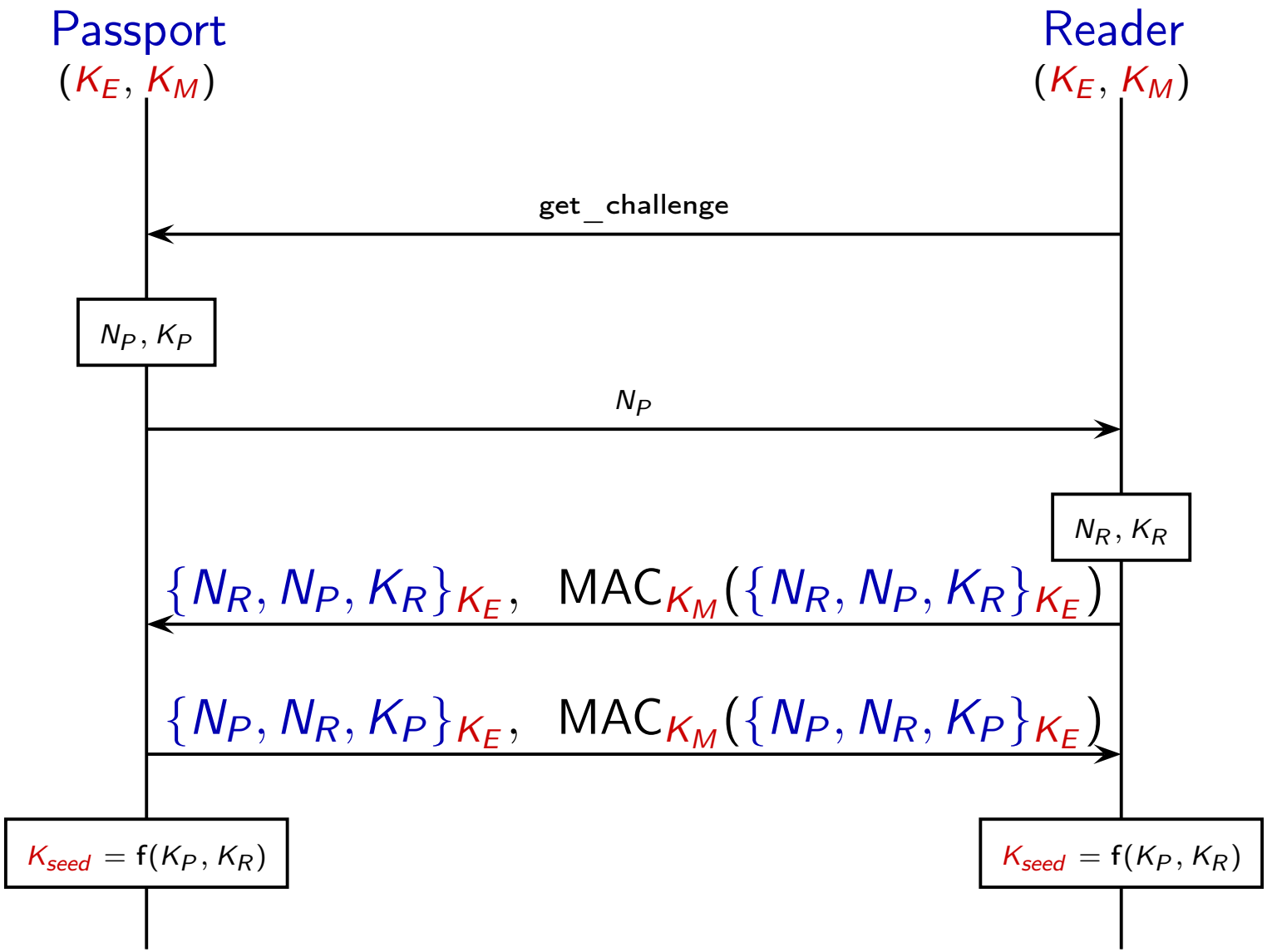
Cryptographic protocols

- 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.

BAC protocol used in e-passport



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

A sucessful approach: formal symbolic verification

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

TAMARIN
Tamarin prover interactive mode

ProVerif

AVANTSSAR

Some success stories

- 2011: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail
  $\longrightarrow$ Armando *et al.* using Avantssar
- 2018: TLS 1.3 formally verified before its deployment
  $\longrightarrow$ project miTLS : `https://www.mitls.org`

A sucessful approach: formal symbolic verification

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

TAMARIN
Tamarin prover interactive mode

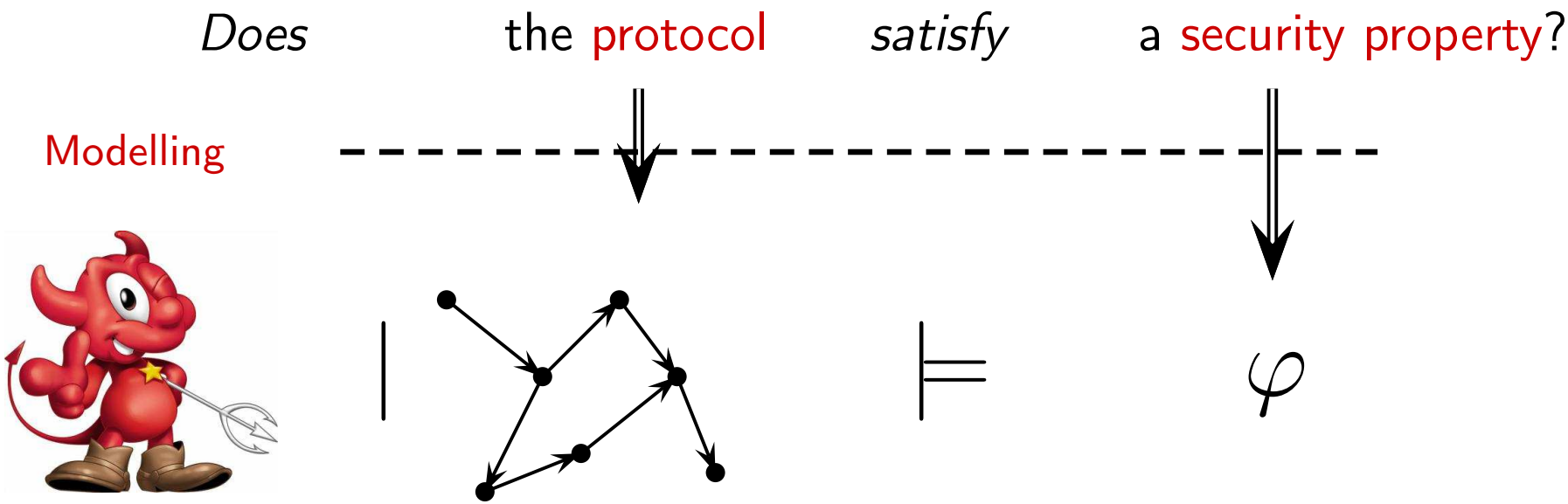
ProVerif

AVANTSSAR

Some success stories

- 2011: Authentication flaw in the Single Sign-On protocol used *e.g.* in GMail
  $\longrightarrow$ Armando *et al.* using Avantssar
- 2018: TLS 1.3 formally verified before its deployment
  $\longrightarrow$ project miTLS : `https://www.mitls.org`

Formal symbolic verification in a nutshell



Two main tasks:

1. Modelling: protocols, security properties, and the attacker !
   $\longrightarrow$ messages are abstracted by terms
2. Designing verification algorithms

Going back to the e-passport

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc, sdec
- concatenation/projections: $\langle\,,\,\rangle$, proj_1, and proj_2
- mac construction: mac

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

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

Modelling Passport's role

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

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

Going back to the e-passport

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc, sdec
- concatenation/projections: $\langle\,,\,\rangle$, proj_1, and proj_2
- mac construction: mac

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

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

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

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

POPSTAR (fev. 2017- jan. 2022)

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

Analysing distance bounding protocols

Keyless system:

- authentication
- physical proximity
  $dist(R, T) \leq \Delta t \times c$

$$R \to T : \quad Request$$

$$\boxed{\begin{array}{ll} R \to T : & N_R \\ T \to R : & T,\ N_T \oplus N_R \end{array}}$$

$$T \to R : \quad Sign_T(T, N_T, N_R)$$

We need a framework that takes into account:

- transmission delay, location of participants, mobility issues, ...
- low-level operators and their algebraic properties.

Example: exclusive-or operator

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

$\longrightarrow$ A. Debant - PhD student (2017-2020), C. Wiedling - DGA

Have a look to his poster !

Analysing distance bounding protocols

Keyless system:

- authentication
- physical proximity
  $dist(R, T) \leq \Delta t \times c$

$$R \to T : \quad Request$$

$$\boxed{\begin{array}{ll} R \to T : & N_R \\ T \to R : & T,\ N_T \oplus N_R \end{array}}$$

$$T \to R : \quad Sign_T(T, N_T, N_R)$$

We need a framework that takes into account:

- transmission delay, location of participants, mobility issues, ...
- low-level operators and their algebraic properties.

Example: exclusive-or operator

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

$\longrightarrow$ A. Debant - PhD student (2017-2020), C. Wiedling - DGA

Have a look to his poster !

Dealing with privacy-type properties

Unlinkability, anonymity, strong secrecy ... are usually expressed as an equivalence:

$$P \approx Q \quad \text{iff} \quad (P \mid \mathbf{A})\Downarrow_c \;\Leftrightarrow\; (Q \mid \mathbf{A})\Downarrow_c \text{ for any attacker } \mathbf{A}$$

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

Example: Unlinkability property [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

Dealing with privacy-type properties

Unlinkability, anonymity, strong secrecy ... are usually expressed as an equivalence:

$$P \approx Q \quad \text{iff} \quad (P \mid \mathbf{A})\Downarrow_c \;\Leftrightarrow\; (Q \mid \mathbf{A})\Downarrow_c \text{ for any attacker } \mathbf{A}$$

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

Example: Unlinkability property [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

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 through a "small" attack result (with a large bound) [CSF'15]

Main limitations:

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

More pragmatic approaches

ProVerif

ProVerif tool: [Blanchet *et al.*, LICS'05]
$\longrightarrow$ fully automatic, quite efficient, but it may not terminate or give false attacks

Tamarin prover: [Basin *et al.*, CCS'15]
$\longrightarrow$ suitable to model protocols with mutable states, various primitives (e.g. modular exponentiation, exclusive-or, ...), it may fail to establish a given property (an interactive mode is available)

TAMARIN
Tamarin prover interactive mode

Main limitation:

- they consider a strong form of equivalence, namely diff-equivalence which is too strong to analyse e.g. vote-privacy, and unlinkability properties.

Equivalence for a bounded number of sessions (a long story)

2005-2011: theoretical algorithms with **no implementation**

2010-2015: "practical" algorithms and tools (e.g. SPEC, APTE, Akiss, . . .) but they scale badly

2018: The DeepSec tool [Cheval, Kremer & Rakotonirina, 2018]

- large class of processes: else branches, standard cryptographic primitives and beyond (e.g. blind signatures)
- quite efficient: exploit multicore architectures, integrate POR optimisations

Protocol		APTE	Deepsec
BAC (4 roles)	attack	38 min	1s
BAC (6 roles)		time out	time out

Intel Xeon 3.10GHz cores, with 50Go of memory – **35 cores**

time out = 12h

My contributions / Work in progress

A novel procedure based on graph planning/SAT solving:
SAT-Equiv (developed by A. Dallon - PhD student (2015-2018)) outperformed existing tools and we would like to improve its scope.
$\longrightarrow$ J. Peignier - Master student (2019) co-supervised with V. Cortier

Another pragmatic approach (unbounded case):
designing sufficient conditions checkable with existing verification tools to conclude on specific equivalences.
$\longrightarrow$ S. Moreau - PhD student (2018-2021) co-supervised with D. Baelde

Theoretical results:
Improving the scope of the decidability result (more protocols) and compute a better bound (now within reach thanks to SAT-Equiv).
$\longrightarrow$ joint work with V. Sundararajan (post-doc) and V. Cortier

Thanks for your attention!