Verification of security protocols: from confidentiality to privacy

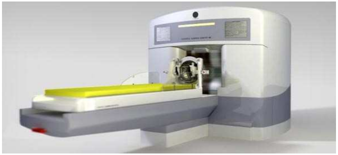
Stéphanie Delaune

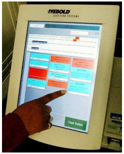
LSV, CNRS & ENS Cachan & INRIA Saclay Île-de-France, France

Friday, March 18th, 2011

Context: verification of critical softwares

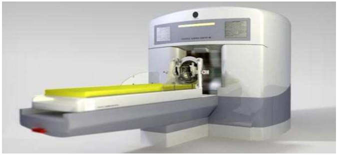
Computers are everywhere!

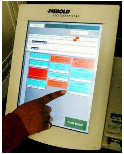




Context: verification of critical softwares

Computers are everywhere!





A need for automated formal verification

- testing the system is not always sufficient
  $\longrightarrow$ we want to consider all the possible behaviours
- manual proofs are tedious and error-prone
  $\longrightarrow$ automated verification techniques

Cryptographic protocols

Cryptographic protocols

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

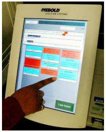
The network is unsecure!

Communications take place over a public network like the Internet.

Cryptographic protocols

Cryptographic protocols

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

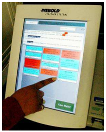


PayPal™

Cryptographic protocols

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

It becomes more and more important to protect our privacy.



Example: electronic passport

⟶ 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

⟶ 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.

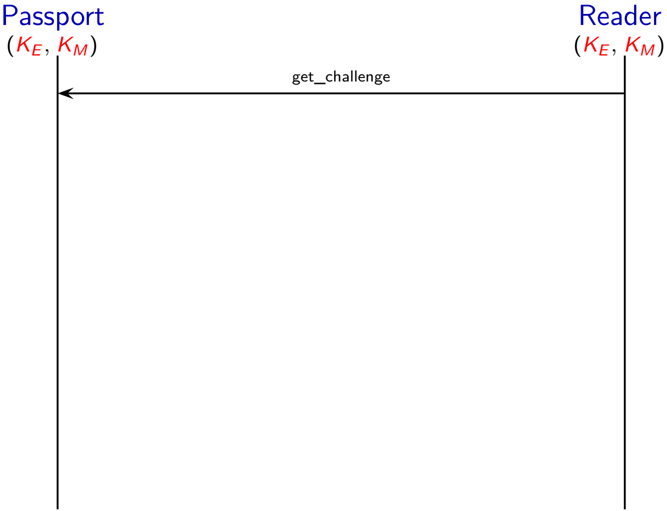
The Basic Access Control (BAC) protocol is a key establishment protocol that has been designed to also ensure unlinkability.

ISO/IEC standard 15408

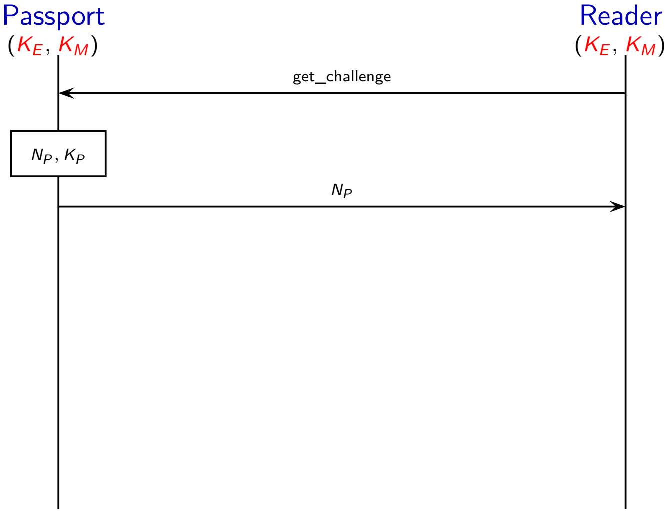
Unlinkability aims to ensure *that a user may make multiple uses of a service or resource without others being able to link these uses together.*

The electronic passport protocol

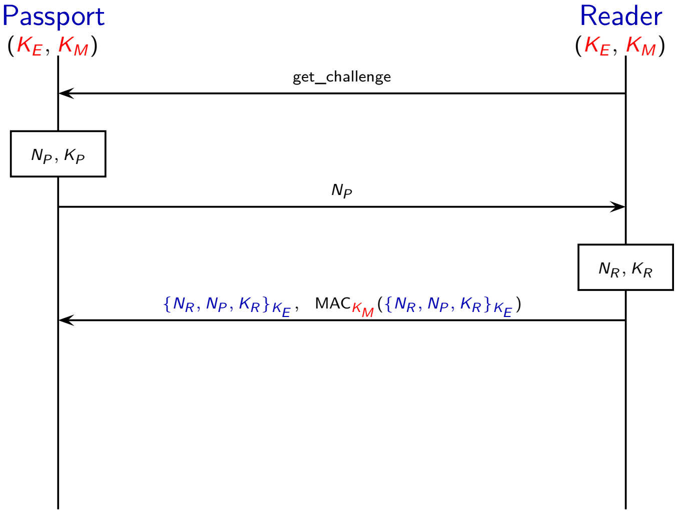
The electronic passport protocol



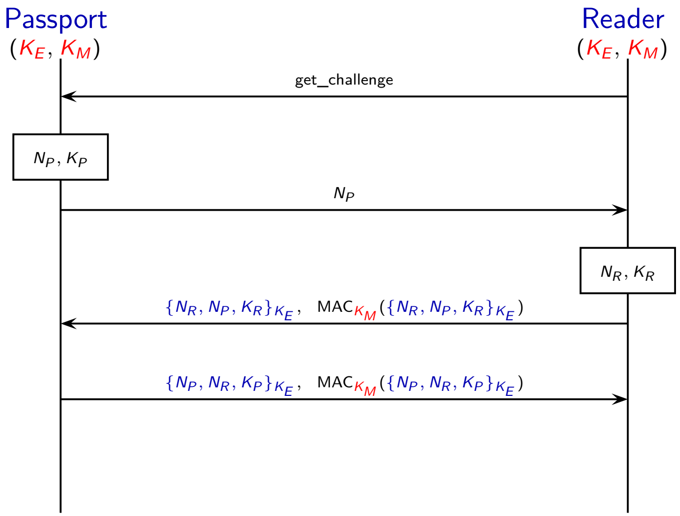
The electronic passport protocol



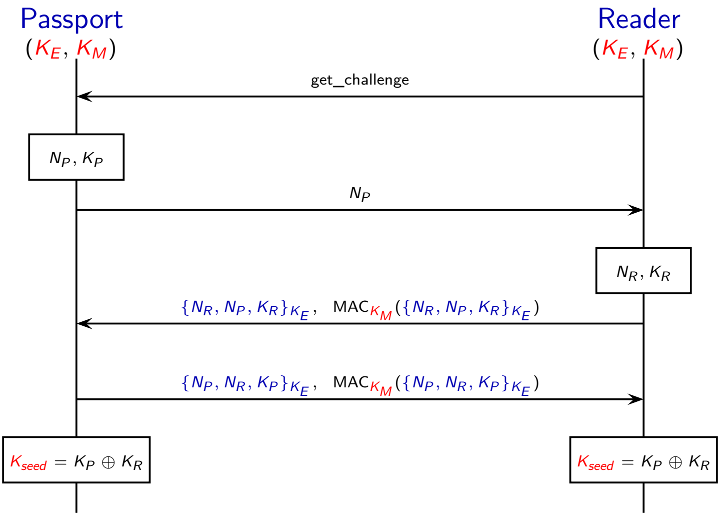
The electronic passport protocol



The electronic passport protocol



The electronic passport protocol

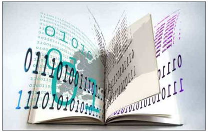


Verification of cryptographic protocols (symbolic models)

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

Verification of cryptographic protocols (symbolic models)

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

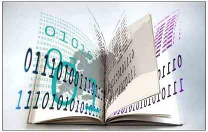


Messages

They are abstracted by terms together with an equational theory.

Verification of cryptographic protocols (symbolic models)

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



Messages

They are abstracted by terms together with an equational theory.

Examples:

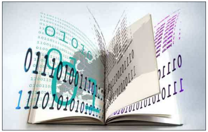
→ symmetric encryption/decryption: $\mathsf{dec}(\mathsf{enc}(x, y), y) = x$

→ exclusive or operator:

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

Verification of cryptographic protocols (symbolic models)

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



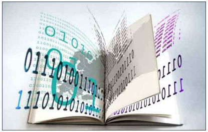
Messages

They are abstracted by terms together with an equational theory.

The attacker

Verification of cryptographic protocols (symbolic models)

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



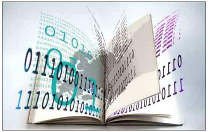
Messages

They are abstracted by terms together with an equational theory.

The attacker

Verification of cryptographic protocols (symbolic models)

⟶ 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.
  ⟶ only symbolic manipulations on terms.

State of the art in a nutshell

What about secrecy?

- several undecidability results for an unbounded number of sessions
  [Even & Goldreich, 83; Durgin *et al*, 99]
- decidability results for a bounded number of sessions (NP-complete)
  [Rusinowitch & Turuani, 01; Millen & Shmatikov, 01]

  $\longrightarrow$ extended by many authors to deal with various primitives.

State of the art in a nutshell

What about secrecy?

- several undecidability results for an unbounded number of sessions
  [Even & Goldreich, 83; Durgin *et al*, 99]
- decidability results for a bounded number of sessions (NP-complete)
  [Rusinowitch & Turuani, 01; Millen & Shmatikov, 01]
  ⟶ extended by many authors to deal with various primitives.

Some automatic verification tools

- AVISPA platform [Armando *et al.*, 05]
  ⟶ state-of-the-art for bounded verification
- ProVerif tool [Blanchet, 01]
  ⟶ quite flexible to analyse security properties

⟶ None of the existing tools is able to analyse the e-passport protocol.

Objectives

Formal analysis of new applications

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

Objectives

Formal analysis of new applications

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

Challenges:

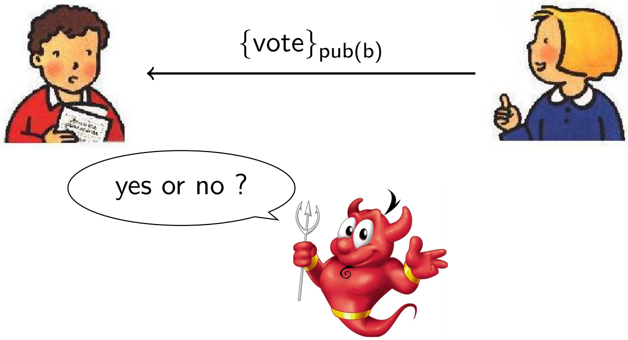
1. Formal definitions of the expected security properties
   $\longrightarrow$ privacy-type security properties
2. Designing appropriate verification algorithms that take into account the specific features of this new type of protocols
3. Composition results

Outline

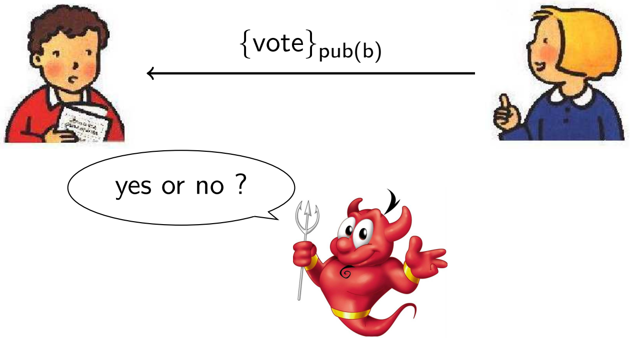
1. Introduction
2. A simple setting: the passive case
3. A more complexe setting: the active case
   - Going beyond with the ProVerif tool
   - Constraint solving approach
4. Perspectives

Outline

A simple protocol



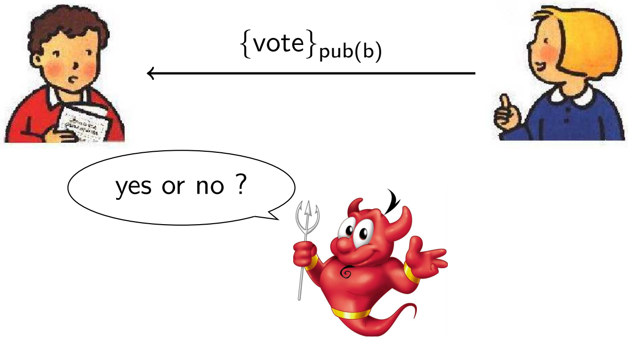
A simple protocol



Question

Does the attacker know Alice's vote?

A simple protocol



The real question

Is the attacker able to tell whether Alice sends yes or no?

Static equivalence (indistinguishability relation)

frame $\phi = \{^{M_1}/_{x_1}, \ldots, ^{M_\ell}/_{x_\ell}\}$

Static equivalence ($\phi \sim \phi'$) [Abadi & Fournet, 01]

Two frames ϕ and ϕ' are statically equivalent if, and only if

$$C_1[M_1, \ldots, M_\ell] = C_2[M_1, \ldots, M_\ell] \Leftrightarrow C_1[M'_1, \ldots, M'_\ell] = C_2[M'_1, \ldots, M'_\ell]$$

for all public contexts C_1, and C_2.

Static equivalence (indistinguishability relation)

frame $$\phi = \{^{M_1}/_{x_1}, \ldots, ^{M_\ell}/_{x_\ell}\}$$

Static equivalence ($\phi \sim \phi'$) [Abadi & Fournet, 01]

Two frames ϕ and ϕ' are statically equivalent if, and only if

$$C_1[M_1, \ldots, M_\ell] = C_2[M_1, \ldots, M_\ell] \Leftrightarrow C_1[M'_1, \ldots, M'_\ell] = C_2[M'_1, \ldots, M'_\ell]$$

for all public contexts C_1, and C_2.

Example: ϕ_1 and ϕ_2 are not in static equivalence.

$$\phi_1 = \{^{\{yes\}_{\mathsf{pub}(b)}}/_x\} \text{ and } \phi_2 = \{^{\{no\}_{\mathsf{pub}(b)}}/_x\}$$

$\longrightarrow C_1 = \{yes\}_{\mathsf{pub}(b)}$ and $C_2 = x$

Static equivalence (indistinguishability relation)

frame $\phi = \{^{M_1}/_{x_1}, \ldots, ^{M_\ell}/_{x_\ell}\}$

Static equivalence ($\phi \sim \phi'$) [Abadi & Fournet, 01]

Two frames ϕ and ϕ' are statically equivalent if, and only if

$$C_1[M_1, \ldots, M_\ell] = C_2[M_1, \ldots, M_\ell] \Leftrightarrow C_1[M'_1, \ldots, M'_\ell] = C_2[M'_1, \ldots, M'_\ell]$$

for all public contexts C_1, and C_2.

State of the art in 2006: [Abadi & Cortier, 06]

- PTIME decision procedure for subterm convergent equational theories
  $\longrightarrow$ *e.g.* symmetric/asymmetric encryption, signature, ...
- some abstract conditions that ensure decidability for many more theories
  $\longrightarrow$ exclusive or, homomorphic encryption, ...

Some results for deduction and static equivalence (1/2)

A generic procedure implemented in the YAPA tool for deciding both notions for subterm convergent equational theories, blind signatures, homomorphic encryption, ...

`http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html`

$\longrightarrow$ in collaboration with M. Baudet & V. Cortier

Some results for deduction and static equivalence (1/2)

A generic procedure implemented in the YAPA tool for deciding both notions for subterm convergent equational theories, blind signatures, homomorphic encryption, ...

`http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html`

$\longrightarrow$ in collaboration with M. Baudet & V. Cortier

Some equational theories motivated by the e-voting application
e.g. re-encryption, trapdoor bit commitment (KiSs tool), ...

`http://www.lsv.ens-cachan.fr/~ciobaca/kiss`

$\longrightarrow$ in collaboration with S. Ciobaca & S. Kremer

Some results for deduction and static equivalence (2/2)

Monoidal equational theories (AC operators)

e.g. exclusive or, abelian groups, ... together with some homomorphism laws

$$h(x + y) = h(x) + h(y)$$

General schema for deciding both problems:

1. Reduce both problems to classical algebraic problems.
2. Use existing results to conclude for many interesting equational theories.

$\longrightarrow$ in collaboration with V. Cortier

Some results for deduction and static equivalence (2/2)

Monoidal equational theories (AC operators)

e.g. exclusive or, abelian groups, ... together with some homomorphism laws

$$h(x+y) = h(x) + h(y)$$

General schema for deciding both problems:

1. Reduce both problems to classical algebraic problems.
2. Use existing results to conclude for many interesting equational theories.

$\longrightarrow$ in collaboration with V. Cortier

Combination results for disjoint theories

If deduction and static equivalence are decidable for E_1 and E_2, then deduction and static equivalence are decidable for $E_1 \cup E_2$.

$\longrightarrow$ in collaboration with V. Cortier

Conclusion and perspectives

Conclusion

- Several new decidability and complexity results ... .

Theory E	Deduction	Static equivalence
subterm convergent	PTIME	
blind signature	PTIME	
homomorphic encryption	decidable	
trapdoor commitment	PTIME	
ACUN	PTIME	PTIME
AG	PTIME	PTIME
ACUNh/AGh	PTIME	decidable
$AGh_1 \ldots h_n$	decidable	

Conclusion and perspectives

Conclusion

- Several new decidability and complexity results ...
- that have been partly implemented (YAPA and KiSs).

Conclusion and perspectives

Conclusion

- Several new decidability and complexity results ...
- that have been partly implemented (YAPA and KiSs).

Some perspectives

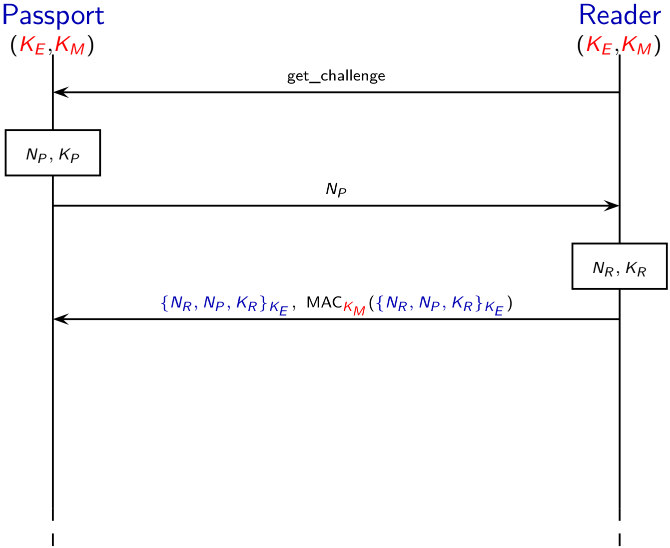
- Extension of YAPA and/or KiSs to theories with AC operators
- Combination for non-disjoint equational theories

More importantly, we have to move to the active case.

Outline

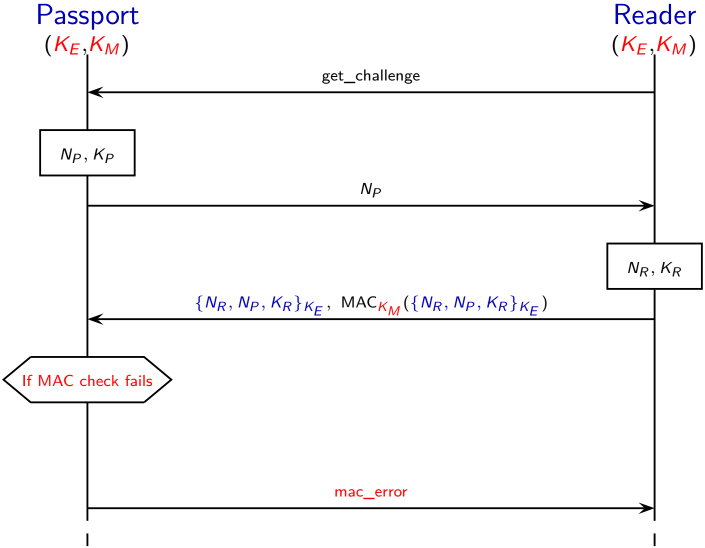
French electronic passport

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



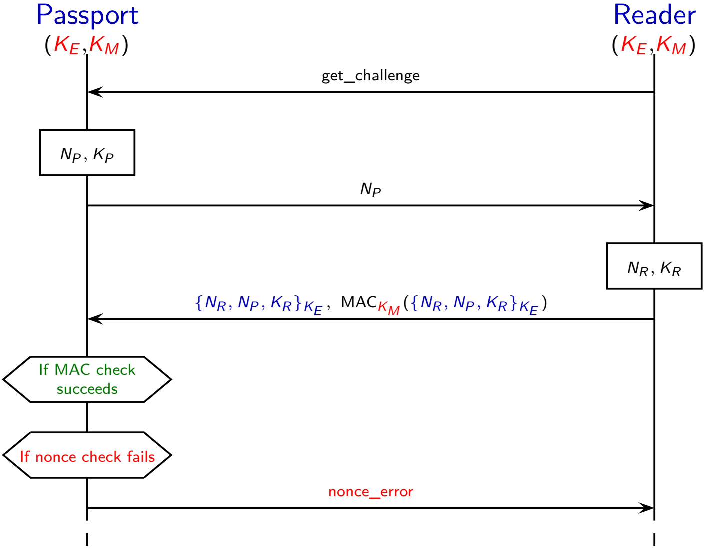
French electronic passport

⟶ 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

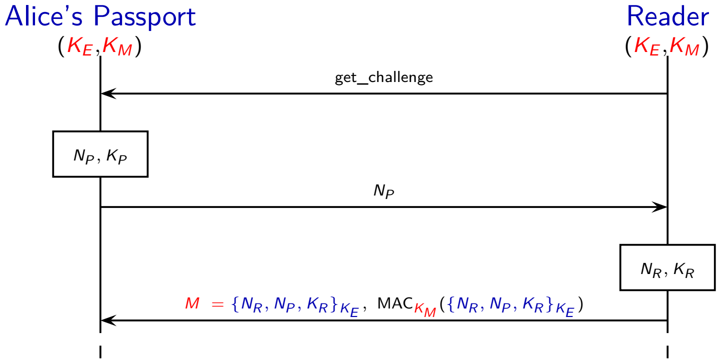
An attacker can track a French passport, provided he has once witnessed a successful authentication.

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.

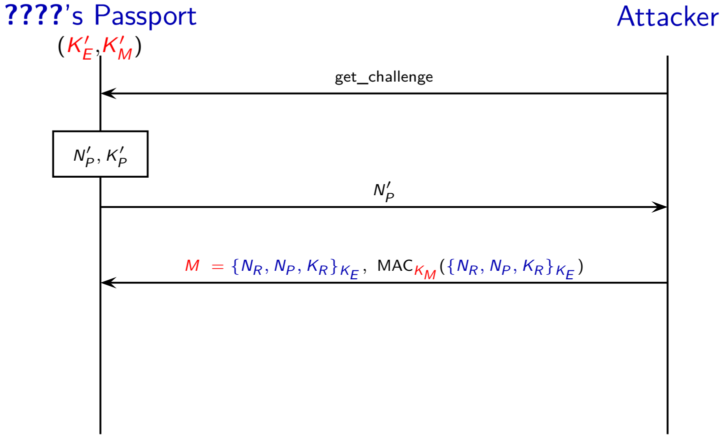
Part 1 of the attack. The attacker eavesdropes on Alice using her passport and records message M.



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

Part 2 of the attack.

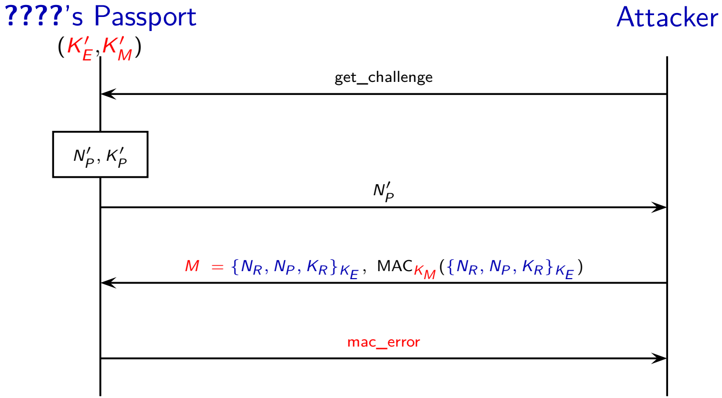
The attacker replays the message M and checks the error code he receives.



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.

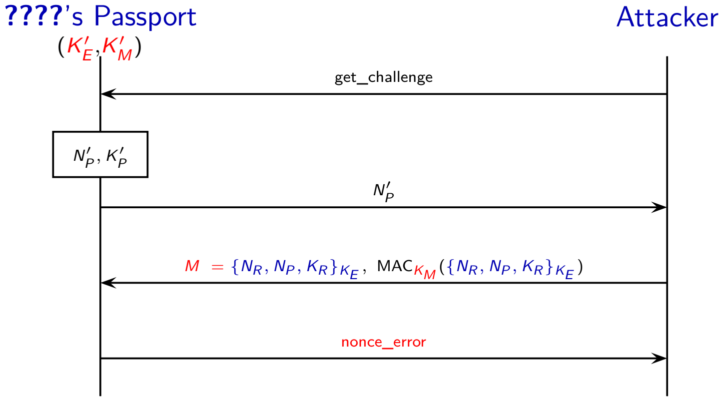


$\Longrightarrow$ MAC check failed $\Longrightarrow$ $K'_M \neq K_M$ $\Longrightarrow$ **????** is not Alice

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

Some difficulties

Formalizing security properties and privacy-type properties is rather subtle.

- Privacy for electronic voting protocols

  $\longrightarrow$ in collaboration with S. Kremer & M. Ryan
- Privacy in vehicular ad hoc network

  $\longrightarrow$ in collaboration with M. Dahl & G. Steel

Some difficulties

Formalizing security properties and privacy-type properties is rather subtle.

- Privacy for electronic voting protocols
  $\longrightarrow$ in collaboration with S. Kremer & M. Ryan
- Privacy in vehicular ad hoc network
  $\longrightarrow$ in collaboration with M. Dahl & G. Steel

Observational equivalence [Abadi & Fournet, 01]

The processes P and Q are indistinguishable, denoted $P \approx Q$, if for all attacker A we have that:

$$A \mid P \text{ can emit on } c \iff A \mid Q \text{ can emit on } c$$

Outline

1. Introduction
2. A simple setting: the passive case
3. A more complexe setting: the active case
   - Going beyond with the ProVerif tool
   - Constraint solving approach
4. Perspectives

ProVerif

Automated protocol verifier mainly developed by B. Blanchet.

Main features

- unbounded number of sessions;
- various cryptographic primitives modeled using rewriting rules and equations;
- various security properties: (strong) secrecy, authentication, equivalence-based security properties.

The tool may not terminate or give false attacks. It works well in practice.

ProVerif

Automated protocol verifier mainly developed by B. Blanchet.

Main features

- unbounded number of sessions;
- various cryptographic primitives modeled using rewriting rules and equations;
- various security properties: (strong) secrecy, authentication, equivalence-based security properties.

The tool may not terminate or give false attacks. It works well in practice.

Some results obtained with ProVerif

Formal analysis of secrecy and authentication properties in the TPM.

⟶ in collaboration with S. Kremer, G. Steel, & M. Ryan

Some limitations of the ProVerif tool

Observational equivalence

- ProVerif considers processes having the same structure (bi-process);
- the notion of equivalence, diff-equivalence, is too strong.

Some limitations of the ProVerif tool

Observational equivalence

- ProVerif considers processes having the same structure (bi-process);
- the notion of equivalence, diff-equivalence, is too strong.

Example

$$P = \text{out}(a) \mid \text{out}(b) \quad \text{and} \quad Q = \text{out}(b) \mid \text{out}(a)$$

We have that P and Q are indistinguishable, *i.e.* $P \approx Q$.

Some limitations of the ProVerif tool

Observational equivalence

- ProVerif considers processes having the same structure (bi-process);
- the notion of equivalence, diff-equivalence, is too strong.

Example

$$P = \mathrm{out}(a) \mid \mathrm{out}(b) \quad \text{and} \quad Q = \mathrm{out}(b) \mid \mathrm{out}(a)$$

We have that P and Q are indistinguishable, *i.e.* $P \approx Q$.

Forming a bi-process, we obtain:

$$\mathrm{out}(choice[a, b]) \mid \mathrm{out}(choice[b, a]).$$

$\longrightarrow$ ProVerif is not able to conclude since they are not in diff-equivalence.

Some limitations of the ProVerif tool

Observational equivalence

- ProVerif considers processes having the same structure (bi-process);
- the notion of equivalence, diff-equivalence, is too strong.

Example

$$P = \mathrm{out}(a) \mid \mathrm{out}(b) \quad \text{and} \quad Q = \mathrm{out}(b) \mid \mathrm{out}(a)$$

We have that P and Q are indistinguishable, *i.e.* $P \approx Q$.

We can also form the bi-process:

$$\mathrm{out}(choice[a, a]) \mid \mathrm{out}(choice[b, b]).$$

$\longrightarrow$ ProVerif is able to conclude. They are in diff-equivalence.

Contributions

We propose a transformation to expand the scope of ProVerif

Input: a bi-process P with some additional comment (** swap *)

Output: a bi-process Q on which ProVerif can directly reason, and such that: P satisfies obs. equiv. $\Leftrightarrow$ Q satisfies obs. equiv.

$\longrightarrow$ in collaboration with B. Smyth & M. Ryan

Contributions

We propose a transformation to expand the scope of ProVerif

Input: a bi-process P with some additional comment (** swap *)

Output: a bi-process Q on which ProVerif can directly reason, and such that: P satisfies obs. equiv. $\Leftrightarrow$ Q satisfies obs. equiv.

$\longrightarrow$ in collaboration with B. Smyth & M. Ryan

Recently, the transformation has been revisited [Smyth & Blanchet,10], and implemented in the ProSwapper tool.

Contributions

We propose a transformation to expand the scope of ProVerif

Input: a bi-process P with some additional comment (** swap *)

Output: a bi-process Q on which ProVerif can directly reason, and such that: P satisfies obs. equiv. $\Leftrightarrow$ Q satisfies obs. equiv.

$\longrightarrow$ in collaboration with B. Smyth & M. Ryan

Recently, the transformation has been revisited [Smyth & Blanchet,10], and implemented in the ProSwapper tool.

Applications

- Electronic voting protocol by Fujioka, Okamoto, and Ohta (FOO)
- Direct Anonymous Attestation protocol based on the TPM (DAA)

  $\longrightarrow$ in collaboration with B. Smyth & M. Ryan
- Vehicular ad hoc network (CMIX protocol, E-toll collection protocol)

  $\longrightarrow$ in collaboration with M. Dahl & G. Steel

Outline

Secrecy problem via constraint solving

$\longrightarrow$ for a fixed number of sessions

Protocol rules

$\mathsf{in}(u_1);\ \mathsf{out}(v_1)$

$\mathsf{in}(u_2);\ \mathsf{out}(v_2)$

$\ldots$

$\mathsf{in}(u_n);\ \mathsf{out}(v_n)$

Constraint system

$$\mathcal{C} = \begin{cases} T_0 \stackrel{?}{\vdash} u_1 \\ T_0, v_1 \stackrel{?}{\vdash} u_2 \\ \quad \ldots \\ T_0, v_1, .., v_n \stackrel{?}{\vdash} s \end{cases}$$

Secrecy problem via constraint solving

$\longrightarrow$ for a fixed number of sessions

Protocol rules	Constraint system
$\mathsf{in}(u_1);\ \mathsf{out}(v_1)$ $\mathsf{in}(u_2);\ \mathsf{out}(v_2)$ $\ldots$ $\mathsf{in}(u_n);\ \mathsf{out}(v_n)$	$\mathcal{C} = \begin{cases} T_0 \overset{?}{\vdash} u_1 \\ T_0, v_1 \overset{?}{\vdash} u_2 \\ \quad \ldots \\ T_0, v_1, .., v_n \overset{?}{\vdash} s \end{cases}$

Solution of a constraint system $\mathcal{C}$

A substitution σ such that

for every $T \overset{?}{\vdash} u \in \mathcal{C}$, *we have that* $u\sigma$ *is deducible from* $T\sigma$.

Main idea of the decision procedure

There exist some algorithms (actually a set of simplification rules) to decide whether such kind of constraint systems have a solution or not.

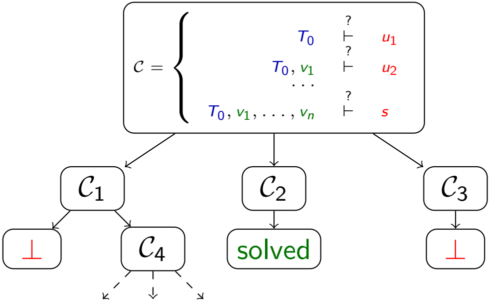
[Millen & Shmatikov, 01; Comon *et al.*, 09]

Main idea of the decision procedure

There exist some algorithms (actually a set of simplification rules) to decide whether such kind of constraint systems have a solution or not.

[Millen & Shmatikov, 01; Comon *et al.*, 09]

Main idea of the procedure:



$\longrightarrow$ this gives us a symbolic representation of all the solutions.

Some results

We extend this procedure to other kind of contraints

Some results

We extend this procedure to other kind of contraints

Other cryptographic primitives

- a generic result for good inference systems that are finite;
- blind signatures (used in e-voting): $v \stackrel{?}{\in} \mathcal{B}d(T, u)$,

$$\frac{\mathsf{sign}(\mathsf{blind}(x, y), z) \quad y}{\mathsf{sign}(x, z)}$$

$\longrightarrow$ Part of PhD work of S. Bursuc

Some results

We extend this procedure to other kind of contraints

Other cryptographic primitives

- a generic result for good inference systems that are finite;
- blind signatures (used in e-voting): $v \stackrel{?}{\in} \mathcal{B}d(T, u)$,

$$\frac{\mathrm{sign}(\mathrm{blind}(x, y), z) \quad y}{\mathrm{sign}(x, z)}$$

$\longrightarrow$ Part of PhD work of S. Bursuc

Routing protocols

- Disequality constraints of the form $\forall X. v \neq u$.
- Neigboorhood constraints: *e.g.* check(a,b)

$\longrightarrow$ Part of PhD work of M. Arnaud

Equivalence-based security properties via constraint solving

Step 1: From observational equivalence to symbolic equivalence
$\longrightarrow$ reduce the problem of deciding an equivalence-based properties on processes to a decision problem on constraint systems.

1. general processes expressed in the applied pi calculus
   $\longrightarrow$ in collaboration with S. Kremer & M. Ryan
2. simple processes
   $\longrightarrow$ in collaboration with V. Cortier

Equivalence-based security properties via constraint solving

Step 1: From observational equivalence to symbolic equivalence

⟶ reduce the problem of deciding an equivalence-based properties on processes to a decision problem on constraint systems.

1. general processes expressed in the applied pi calculus
   ⟶ in collaboration with S. Kremer & M. Ryan
2. simple processes
   ⟶ in collaboration with V. Cortier

Step 2: Decision procedure for symbolic equivalence

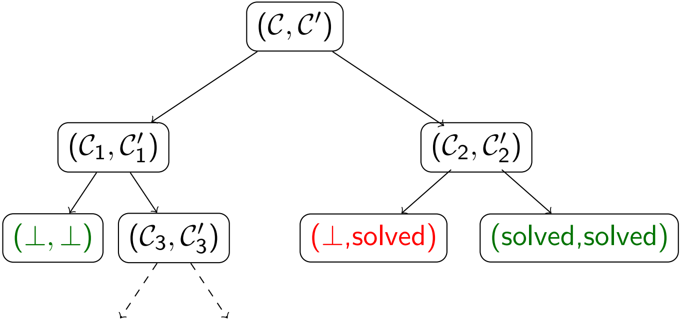
⟶ several procedures already exist,
e.g. [Baudet, 05; Chevalier & Rusinowitch, 09].

1. a new procedure based on a set of simplification rules
2. implementation: the ADECS tool
   `http://www.lsv.ens-cachan.fr/~cheval/program/adecs/`

⟶ part of PhD work of V. Cheval

Our procedure in a nutshell

Main idea: We rewrite pairs of constraint systems (extended to keep track of some information) until a trivial failure or a trivial success is found.



Some perspectives

How can we expand further the scope of ProVerif?

$\longrightarrow$ more cryptographic primitives (*e.g.* exclusive or)

- by relying on the finite variant property as done in [Küsters & Truderung, 10] for trace-based security properties;
- Application: RFID protocols.

Some perspectives

How can we expand further the scope of ProVerif?
$\longrightarrow$ more cryptographic primitives (*e.g.* exclusive or)

- by relying on the finite variant property as done in [Küsters & Truderung, 10] for trace-based security properties;
- Application: RFID protocols.

Constraint solving approach

- Algorithms for symbolic equivalence for more general systems *e.g.* disequality tests, more primitives
- Moving from symbolic equivalence of pairs of constraints to symbolic equivalence of sets of constraints
  $\longrightarrow$ *This will allow us to analyse the e-passport protocol*
- Efficient procedure to reduce equivalence of processes to symbolic equivalence of constraints

Outline

Objectives

Formal 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 that take into account the specific features of this new type of protocols
3. Composition results

Security issues in mobile ad hoc network

Applications: RFID protocols, routing protocols, protocols in vehicular ad hoc network (*e.g.* e-toll collection protocol)

Security issues in mobile ad hoc network

Applications: RFID protocols, routing protocols, protocols in vehicular ad hoc network (*e.g.* e-toll collection protocol)

Modelling issues

- security properties: privacy, route validity
- classical Dolev-Yao attacker model is too strong
  $\longrightarrow$ local attacker, rushing attacks
- taking into account mobility

Security issues in mobile ad hoc network

Applications: RFID protocols, routing protocols, protocols in vehicular ad hoc network (*e.g.* e-toll collection protocol)

Modelling issues

- security properties: privacy, route validity
- classical Dolev-Yao attacker model is too strong
  $\longrightarrow$ local attacker, rushing attacks
- taking into account mobility

Verification issues

- we need to extend the verification techniques to integrate these new features
- reduction results to simplify the topology, the attacker model, ...

Privacy-type properties

A taxonomy for privacy-type properties

For many applications (*e.g.* routing protocols), formal definitions of privacy-type properties are still missing.

Privacy-type properties

A taxonomy for privacy-type properties

For many applications (*e.g.* routing protocols), formal definitions of privacy-type properties are still missing.

Verification algorithms (in the active setting)

First step: an efficient verification tool (for a bounded number of sessions) allowing us to deal with:

- e-passport protocol - see [Arapinis *et al.*, 10]
- private authentication protocols - see [Abadi & Fournet, 04]

$\longrightarrow$ *those protocols are out of reach of the current existing tools*

Privacy-type properties

A taxonomy for privacy-type properties

For many applications (*e.g.* routing protocols), formal definitions of privacy-type properties are still missing.

Verification algorithms (in the active setting)

First step: an efficient verification tool (for a bounded number of sessions) allowing us to deal with:

- e-passport protocol - see [Arapinis *et al.*, 10]
- private authentication protocols - see [Abadi & Fournet, 04]

$\longrightarrow$ *those protocols are out of reach of the current existing tools*

Second step:

- more primitives: subterm convergent, monoidal, and combination results
- integrate some specific features depending on the target applications

Composition (1/2)

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

Protocols do not compose well as soon as they share data.

Composition (1/2)

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

Protocols do not compose well as soon as they share data.

Example:

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

Question: What about the secrecy of s?

Composition (1/2)

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

Protocols do not compose well as soon as they share data.

Example:

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

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

Question: What about the secrecy of s?

Composition (1/2)

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

Protocols do not compose well as soon as they share data.

Main goal: Investigate sufficient conditions under which protocols can be safely composed.

Composition (1/2)

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

Protocols do not compose well as soon as they share data.

Main goal: Investigate sufficient conditions under which protocols can be safely composed.

- From one protocol to many (secrecy, authentication, password-based protocols)

  $\longrightarrow$ in collaboration with V. Cortier, S. Kremer, & M. Ryan
- From one sessions to many

  $\longrightarrow$ in collaboration with M. Arapinis & S. Kremer

Composition (2/2)

Composition

- What about protocols that involve an arbitrary number of agents?
- What about equivalence-based properties?

$\longrightarrow$ establish unlinkability for two tags and obtain guarantee in a setting that involves an arbitrary number of tags.

Composition (2/2)

Composition

- What about protocols that involve an arbitrary number of agents?
- What about equivalence-based properties?

$\longrightarrow$ establish unlinkability for two tags and obtain guarantee in a setting that involves an arbitrary number of tags.

Symbolic Universal Composability (UC)

A paradigm that has been quite successful in the computational approach.

$$\exists \mathcal{S} \text{ such that } \mathcal{F} \approx \mathcal{S}[P]$$

$\longrightarrow$ in collaboration with S. Kremer & O. Pereira

- bring the benefit of this approach in the symbolic setting;
- analysis of more sophisticated protocols specified by an ideal functionality.

Collaborations

The results presented in this habilitation thesis have been obtained in collaboration with many other researchers that are listed below:

- Myrto Arapinis
- Mathilde Arnaud
- Mathieu Baudet
- Sergiu Bursuc
- Rohit Chadha
- Vincent Cheval
- Ștefan Ciobâcă
- Hubert Comon-Lundh
- Véronique Cortier
- Morten Dahl
- Jérémie Delaitre
- Steve Kremer
- Olivier Pereira
- Mark D. Ryan
- Ben Smyth
- Graham Steel

Many thanks to all of them!