APTE: an automatic tool for verifying privacy-type security properties

Stéphanie Delaune

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

$\longrightarrow$ tool developed by Vincent CHEVAL

Tuesday, March 18th, 2014

Context: cryptographic protocols

Cryptographic protocols

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

Context: cryptographic protocols

Cryptographic protocols

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

It becomes more and more important to protect our privacy.

Example: private authentication protocol

$$A \to B: \quad \{N_a, \mathsf{pub}_A\}_{\mathsf{pub}_B}$$

$$B \to A: \quad \{N_a, N_b, \mathsf{pub}_B\}_{\mathsf{pub}_A}$$

Example: private authentication protocol

$$A \rightarrow B: \quad \{N_a, \mathsf{pub}_A\}_{\mathsf{pub}_B}$$
$$B \rightarrow A: \quad \{N_a, N_b, \mathsf{pub}_B\}_{\mathsf{pub}_A}$$

Is an attacker able to distinguish the two scenarios?

1. the protocol is played between the agents a and b;
2. the protocol is played between the agents a' and b.

Example: private authentication protocol

$$A \to B: \quad \{N_a, \mathsf{pub}_A\}_{\mathsf{pub}_B}$$
$$B \to A: \quad \{N_a, N_b, \mathsf{pub}_B\}_{\mathsf{pub}_A}$$

Is an attacker able to distinguish the two scenarios?

1. the protocol is played between the agents a and b;
2. the protocol is played between the agents a' and b.

Description of the attack:

$\longrightarrow$ the attacker sends $\{N, \mathsf{pub}_A\}_{\mathsf{pub}_B}$ and observes the answer sent by B.

1. b will answer with a message of the form $\{N, N_b, \mathsf{pub}_B\}_{\mathsf{pub}_A}$;
2. b will not give any answer.

Example: private authentication protocol

$$A \to B: \quad \{N_a, \mathrm{pub}_A\}_{\mathrm{pub}_B}$$

$$B \to A: \quad \{N_a, N_b, \mathrm{pub}_B\}_{\mathrm{pub}_A} \quad \text{in case } B \text{ is willing to talk to } A$$

$$\{N_b\}_{\mathrm{pub}_B} \quad \text{otherwise}$$

Is an attacker able to distinguish the two scenarios?

1. the protocol is played between the agents a and b;
2. the protocol is played between the agents a' and b.

Description of the attack:

$\longrightarrow$ the attacker sends $\{N, \mathrm{pub}_A\}_{\mathrm{pub}_B}$ and observes the answer sent by B.

1. b will answer with a message of the form $\{N, N_b, \mathrm{pub}_B\}_{\mathrm{pub}_A}$;
2. b will not give any answer.

$\longrightarrow$ a possible fix in red

Example continued - more formally

Modelling the protocol

$A(a, b) =$
$\quad \mathsf{new}\, n_a.$
$\quad \mathsf{out}(c, \{\langle n_a, \mathsf{pk}(sk_a)\rangle\}_{\mathsf{pk}(sk_b)}).$
$\quad \mathsf{in}(c, z). \ldots$

$B(b, a) = \mathsf{new}\, n_b.\, \mathsf{in}(c, y).$
$\quad \mathsf{if}\ \pi_2(\mathsf{adec}(y, sk_b)) = \mathsf{pk}(sk_a)$
$\quad\quad \mathsf{then}\ \mathsf{out}(c, \{..., n_b, \mathsf{pk}(sk_a)\}_{\mathsf{pk}(sk_a)}).$
$\quad\quad \mathsf{else}\ \mathsf{out}(c, \{n_b\}_{\mathsf{pk}(sk_b)})$

Example continued - more formally

Modelling the protocol

$$A(a,b) = \\ \quad \mathsf{new}\, n_a. \\ \quad \mathsf{out}(c, \{\langle n_a, \mathsf{pk}(sk_a)\rangle\}_{\mathsf{pk}(sk_b)}). \\ \quad \mathsf{in}(c,z).\ldots$$

$$B(b,a) = \mathsf{new}\, n_b.\,\mathsf{in}(c,y). \\ \quad \mathsf{if}\ \pi_2(\mathsf{adec}(y, sk_b)) = \mathsf{pk}(sk_a) \\ \quad\quad \mathsf{then}\ \mathsf{out}(c, \{\ldots, n_b, \mathsf{pk}(sk_a)\}_{\mathsf{pk}(sk_a)}). \\ \quad\quad \mathsf{else}\ \mathsf{out}(c, \{n_b\}_{\mathsf{pk}(sk_b)})$$

Modelling the property

$$C[A(a,b) \mid B(b,a)] \ \stackrel{?}{\approx}_t \ C[A(a',b) \mid B(b,a')]$$

where $C = \mathsf{new}\, sk_a, \mathsf{new}\, sk_{a'}, \mathsf{new}\, sk_b.$
$\mathsf{out}(c, \mathsf{pk}(sk_a)).\mathsf{out}(c, \mathsf{pk}(sk_{a'})).\mathsf{out}(c, \mathsf{pk}(sk_b)).\, \_.$

Example continued - more formally

Modelling the protocol

$$A(a,b) = \\ \quad \mathsf{new}\, n_a. \\ \quad \mathsf{out}(c, \{\langle n_a, \mathsf{pk}(sk_a)\rangle\}_{\mathsf{pk}(sk_b)}). \\ \quad \mathsf{in}(c,z).\ldots$$

$$B(b,a) = \mathsf{new}\, n_b.\, \mathsf{in}(c,y). \\ \quad \mathsf{if}\ \pi_2(\mathsf{adec}(y, sk_b)) = \mathsf{pk}(sk_a) \\ \quad\quad \mathsf{then\ out}(c, \{\ldots, n_b, \mathsf{pk}(sk_a)\}_{\mathsf{pk}(sk_a)}). \\ \quad\quad \mathsf{else\ out}(c, \{n_b\}_{\mathsf{pk}(sk_b)})$$

Modelling the property

$$C[A(a,b) \mid B(b,a)] \ \stackrel{?}{\approx}_t \ C[A(a',b) \mid B(b,a')]$$

where $C = \mathsf{new}\, sk_a, \mathsf{new}\, sk_{a'}, \mathsf{new}\, sk_b.$
$\quad\quad \mathsf{out}(c, \mathsf{pk}(sk_a)).\mathsf{out}(c, \mathsf{pk}(sk_{a'})).\mathsf{out}(c, \mathsf{pk}(sk_b)).\,\_.$

Each experiment performed by the attacker on the left leads to a sequence of messages Φ_1 which is indistinguishable from the sequence Φ_2 obtained when performing the same expriment on the right.

Difficulties when checking trace equivalence

$\longrightarrow$ even considering a fixed number of protocol executions.

Main difficulties:

1. the attacker can build arbitrary messages (provided that they are deducible from his knowledge)

   $\longrightarrow$ no hope to test each experiment in turn

2. once the experiment is fixed, we still have to decide whether the resulting sequence of messages are indistinguishable or not.

Difficulties when checking trace equivalence

$\longrightarrow$ even considering a fixed number of protocol executions.

Main difficulties:

1. the attacker can build arbitrary messages (provided that they are deducible from his knowledge)

   $\longrightarrow$ no hope to test each experiment in turn

2. once the experiment is fixed, we still have to decide whether the resulting sequence of messages are indistinguishable or not.

Running example: fix version

$\longrightarrow$ consider the experiment where the attacker sends $\{N, \mathsf{pk}(sk_a)\}_{\mathsf{pk}(sk_b)}$

The resulting sequences of messages are:

1. $\Phi_1 = \mathsf{pk}(sk_a),\ \mathsf{pk}(sk_{a'}),\ \mathsf{pk}(sk_b),\ \{n, n_b, \mathsf{pk}(sk_b)\}_{\mathsf{pk}(sk_a)}$
2. $\Phi_2 = \mathsf{pk}(sk_a),\ \mathsf{pk}(sk_{a'}),\ \mathsf{pk}(sk_b),\ \{n_b\}_{\mathsf{pk}(sk_b)}$.

where sk_a, $sk_{a'}$, sk_b, and n_b are unknown.

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions

e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions
e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Unbounded number of sessions [Blanchet, Abadi & Fournet, 05]

ProVerif tool [Blanchet, 01] `http://www.proverif.ens.fr/`

- + unbounded number of sessions; various cryptographic primitives;
- – termination is not guaranteed; diff-equivalence (too strong)

Algorithms for checking trace equivalence

trace equivalence is undecidable in general

Bounded number of sessions

e.g. [Baudet, 05], [Dawson & Tiu, 10], [Chevalier & Rusinowitch, 10], ...

$\longrightarrow$ this allows us to decide trace equivalence between simple processes with trivial else branches. [Cortier & Delaune, 09]

Unbounded number of sessions [Blanchet, Abadi & Fournet, 05]

ProVerif tool [Blanchet, 01] `http://www.proverif.ens.fr/`

- + unbounded number of sessions; various cryptographic primitives;
- – termination is not guaranteed; diff-equivalence (too strong)

$\longrightarrow$ None of these results is able to analyse the private authentication protocol.

Our contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune   CCS 2011

Main result

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

Our contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune CCS 2011

Main result

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

Our class of processes:

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

Our contribution

$\longrightarrow$ V. Cheval, H. Comon-Lundh, and S. Delaune CCS 2011

Main result

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

Our class of processes:

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

Some applications

- unlinkability in RFID protocols (*e.g.* e-passport protocol)
- anonymity (*e.g.* private authentication protocol)

Our procedure in a nutshell

Two main steps:

1. A symbolic exploration of all the possible traces

   The infinite number of possible traces (*i.e.* experiment) are represented by a finite set of symbolic traces.

   $\longrightarrow$ this set is still huge (exponential) !

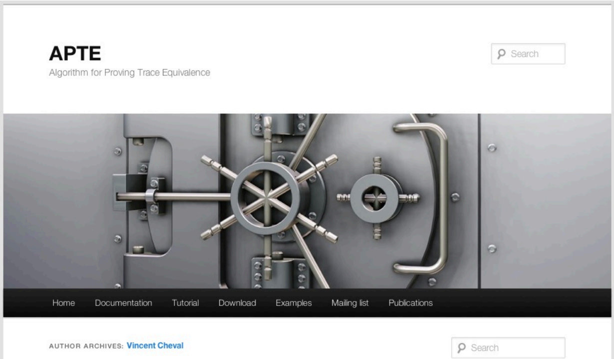
2. A decision procedure for deciding (symbolic) equivalence between sets of symbolic traces.

   $\longrightarrow$ this algorithm works quite well

APTE- Algorithm for Proving Trace Equivalence

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

$\longrightarrow$ developed by Vincent Cheval



$\longrightarrow$ written in Ocaml, around 12 KLocs

Demo

Conclusion

APTE is an automatic tool for analysing privacy type properties expressed using trace equivalence

Case studies:

- private authentication protocol
- several protocols from the e-passport application
- some classical protocols from the literature (e.g. Needham-Schroeder, Wide Mouthed Frog protocol, ...)

$\longrightarrow$ This is the only automatic tool that is able to analyse the BAC protocol (e-passport)

Conclusion

APTE is an automatic tool for analysing privacy type properties expressed using trace equivalence

Case studies:

- private authentication protocol
- several protocols from the e-passport application
- some classical protocols from the literature (e.g. Needham-Schroeder, Wide Mouthed Frog protocol, ...)

$\longrightarrow$ This is the only automatic tool that is able to analyse the BAC protocol (e-passport)

Main limitations:

- APTE can only handle standard cryptographic primitives
  $\longrightarrow$ e-voting protocols are out of reach of APTE
- APTE can only consider a bounded number of sessions (and actually a very small number)