APTE: an Algorithm for Proving Trace Equivalence

Vincent Cheval

UNIVERSITY OF BIRMINGHAM

11 April 2014, TACAS, Grenoble

Context

Most communications take place over a public network

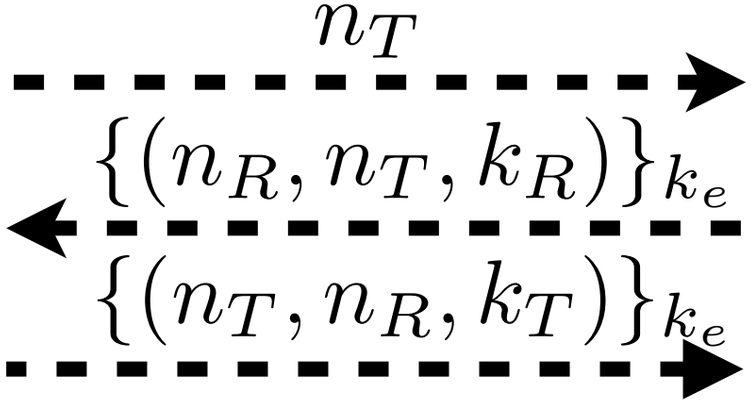
Cryptographic protocols

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

Security properties

Example : E-passport

Passport



Officer

Formal methods and symbolic models

Security properties

Example : E-passport

$$n_T \dashrightarrow$$
$$\dashleftarrow \{(n_R, n_T, k_R)\}_{k_e}$$
$$\{(n_T, n_R, k_T)\}_{k_e} \dashrightarrow$$

Passport Officer

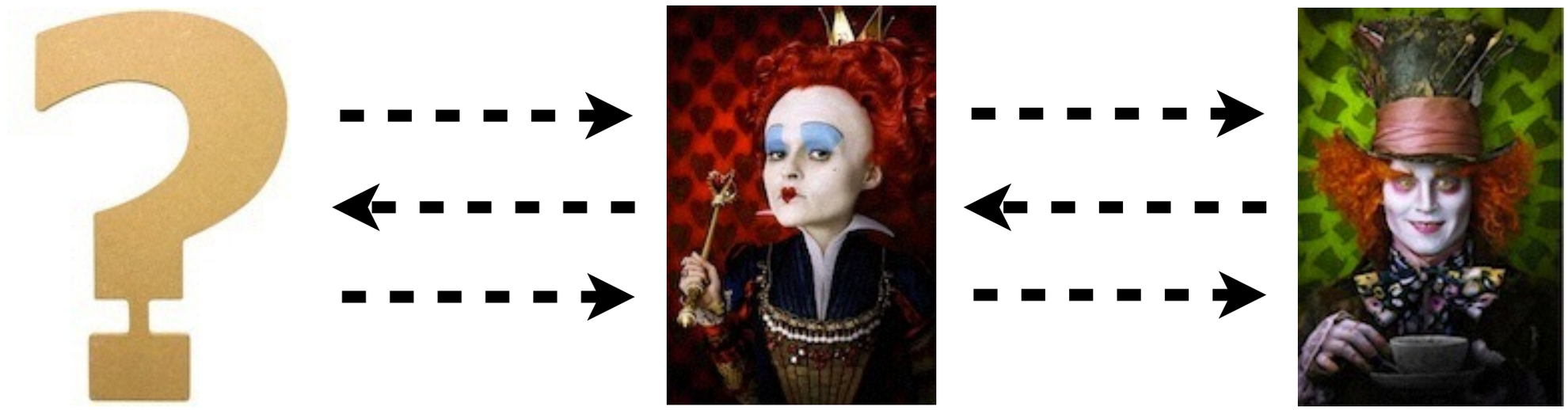
Formal methods and symbolic models

Security properties

- Confidentiality ⟶ Reachability properties
- Anonymity ⟶ Equivalence properties

Examples : Privacy, properties for electronic voting, unlinkability..

Anonymity



Unknown Attacker Bob

Anonymity

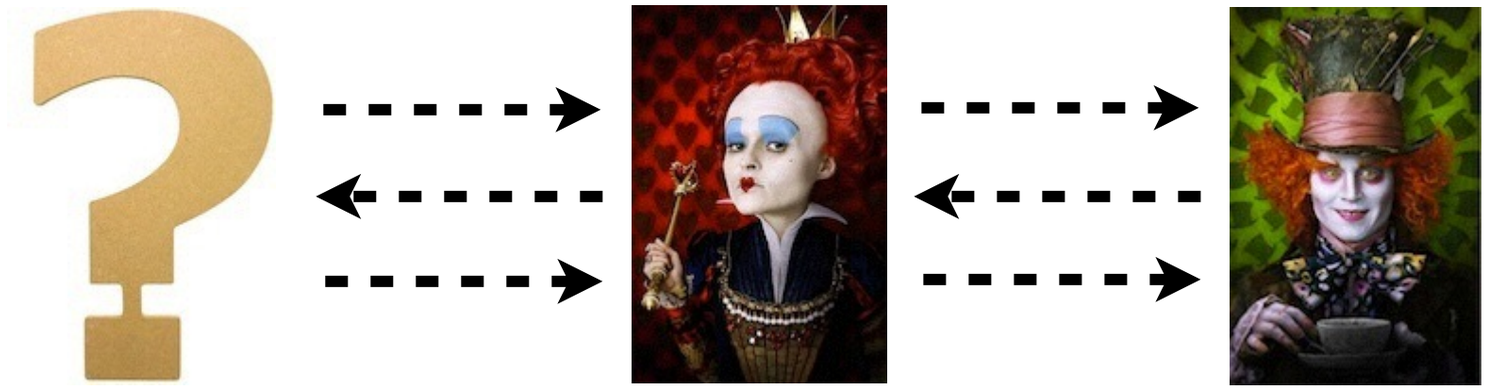
- Intercept all messages
- Can transmit or modify messages
- Test equalities between messages

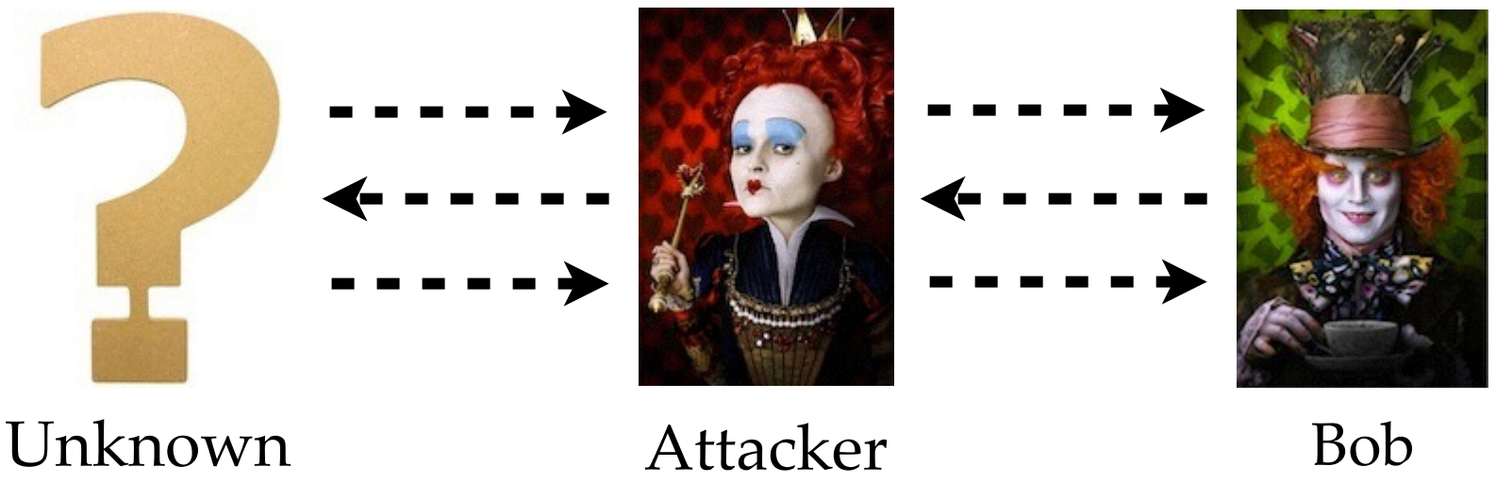
Unknown

Attacker

Bob

Anonymity





Anonymity

Charlene

Unknown

Attacker

Bob

Alice

Unknown

Attacker

Bob

Anonymity

Charlene

Unknown

Attacker

Bob

Alice

Unknown

Attacker

Bob

Can the intruder distinguish the two situations ?

Anonymity

Charlene

Unknown

Attacker

Bob

Alice

Unknown

Attacker

Bob

Trace equivalence

Private authentication protocol

Private authentication protocol

Alice

$$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$$

Bob

Private authentication protocol

Alice

$$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$$

$\mathsf{pk}(k_A)?$

Bob

Private authentication protocol

Alice

$$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$$

$$\{\langle N_a, N_b, \mathsf{pk}(k_B)\rangle\}_{\mathsf{pk}(k_A)}$$

$\mathsf{pk}(k_A)?$

Bob

Private authentication protocol

Alice

$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$

$\{\langle N_a, N_b, \mathsf{pk}(k_B)\rangle\}_{\mathsf{pk}(k_A)}$

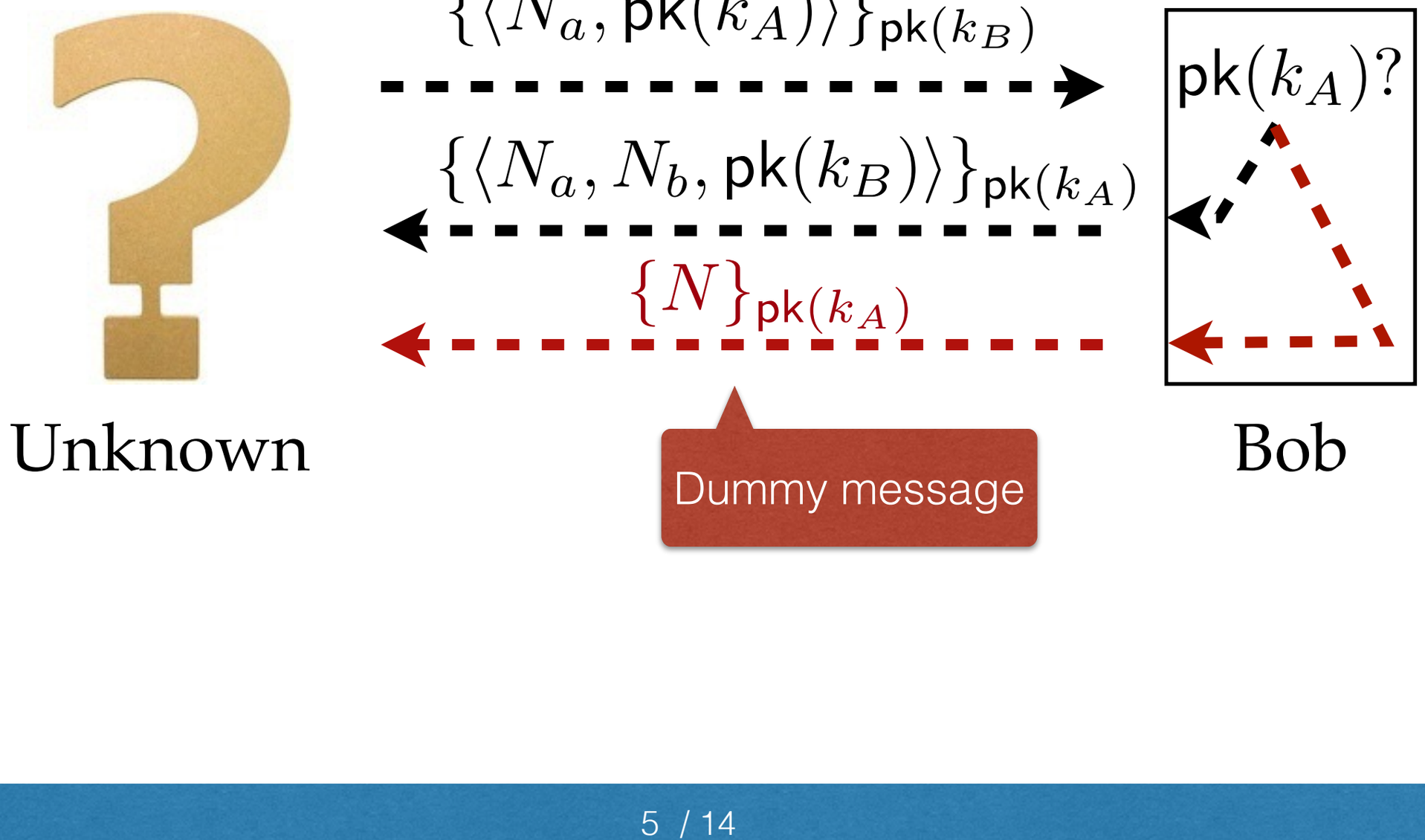
$\{N\}_{\mathsf{pk}(k_A)}$

$\mathsf{pk}(k_A)?$

Dummy message

Bob

Private authentication protocol



Automatic tools

- For reachability properties

  Avispa, CSP/FDR, ProVerif, Scyther, Maude-NPA, ...

Automatic tools

- For reachability properties

  Avispa, CSP/FDR, ProVerif, Scyther, Maude-NPA, ...

- For equivalence properties

  - **ProVerif:** Bruno Blanchet. *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules.*
  - **SPEC:** Alwen Tiu and Jeremy E. Dawson. *Automating open bisimulation checking for the spi calculus.*
  - **AKiSs:** Ştefan Ciobâcă. *Automated Verification of Security Protocols with Applications to Electronic Voting.*

Automatic tools

- For reachability properties

  Avispa, CSP/FDR, ProVerif, Scyther, Maude-NPA, ...

- For equivalence properties

  - **ProVerif:** Bruno Blanchet. *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules.*
  - **SPEC:** Alwen Tiu and Jeremy E. Dawson. *Automating open bisimulation checking for the spi calculus.*
  - **AKiSs:** Ştefan Ciobâcă. *Automated Verification of Security Protocols with Applications to Electronic Voting.*

Do not handle private authentication protocol and e-passport protocol

Automatic tools

- For reachability properties

  Avispa, CSP/FDR, ProVerif, Scyther, Maude-NPA, ...

- For equivalence properties

  - **ProVerif:** Bruno Blanchet. *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules.*
  - **SPEC:** Alwen Tiu and Jeremy E. Dawson. *Automating open bisimulation checking for the spi calculus.*
  - **AKiSs:** Ştefan Ciobâcă. *Automated Verification of Security Protocols with Applications to Electronic Voting.*

Do not handle private authentication protocol and e-passport protocol

APTE can do it !

Demonstration

Private_Authentication_Protocol.txt — Edited

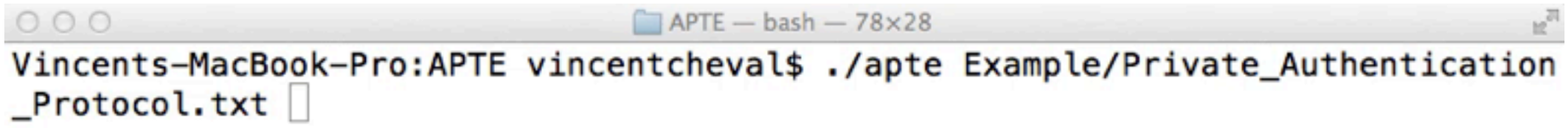
```
# Private Authentication Protocol

free c.

### Description of the role of Alice

let process_Alice k_a k_b =
  new N_a;
  out(c,aenc((N_a,pk(k_a)),pk(k_b)));
  in(c,x).

### Description of the role of Bob

let process_Bob k_a k_b =
  in(c,x);
  let (na,pka) = adec(x,k_b) in
  if pka = pk(k_a)
  then new N_b; out(c,aenc((na,N_b,pk(k_b)),pk(k_a)))
  else new N; out(c,aenc(N,pk(k_a))).

### Main

let instance1 =
  new k_a ; new k_b ; new k_c ; out(c,pk(k_a)) ; out(c,pk(k_b)) ;
out(c,pk(k_c));
  ( process_Alice k_a k_b | process_Bob k_a k_b ).
```

Demonstration

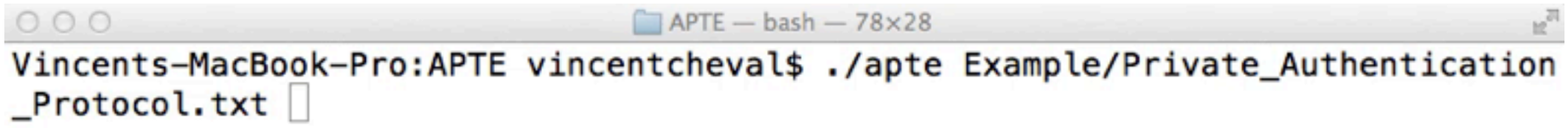
Private_Authentication_Protocol.txt — Edited

```
# Private Authentication Protocol

free c.

### Description of the role of Alice

let process_Alice k_a k_b =
  new N_a;
  out(c,aenc((N_a,pk(k_a)),pk(k_b)));
  in(c,x).

### Description of the role of Bob

let process_Bob k_a k_b =
  in(c,x);
  let (na,pka) = adec(x,k_b) in
  if pka = pk(k_a)
  then new N_b; out(c,aenc((na,N_b,pk(k_b)),pk(k_a)))
  else new N; out(c,aenc(N,pk(k_a))).

### Main

let instance1 =
  new k_a ; new k_b ; new k_c ; out(c,pk(k_a)) ; out(c,pk(k_b)) ;
out(c,pk(k_c));
  ( process_Alice k_a k_b | process_Bob k_a k_b ).
```

Demonstration

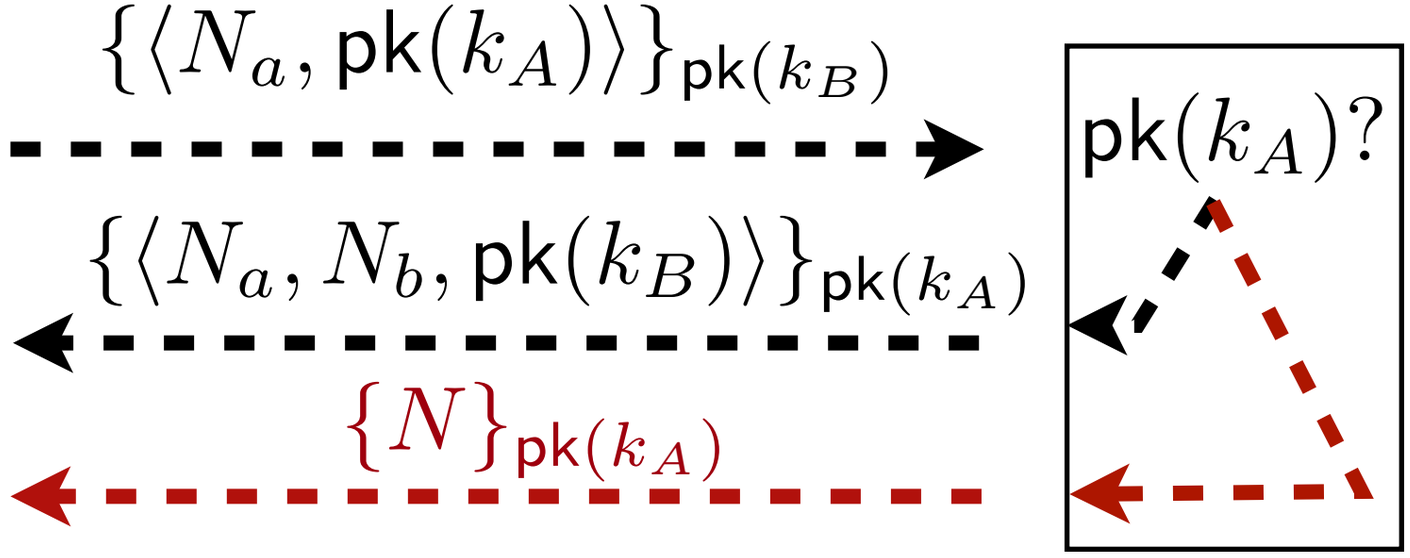


Demonstration



Length of messages

Alice



Bob

Length of messages

Alice

$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$

$\{\langle N_a, N_b, \mathsf{pk}(k_B)\rangle\}_{\mathsf{pk}(k_A)}$

$\{N\}_{\mathsf{pk}(k_A)}$

Size of dummy message differs

$\mathsf{pk}(k_A)?$

Bob

Length of messages

Alice

$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$

$\{\langle N_a, N_b, \mathsf{pk}(k_B)\rangle\}_{\mathsf{pk}(k_A)}$

$\{N\}_{\mathsf{pk}(k_A)}$

Size of dummy message differs

$\mathsf{pk}(k_A)?$

Bob

Length functions

$\mathcal{N} = \mathcal{N}_1 \cup \ldots \cup \mathcal{N}_n \cup \ldots$

nonces of any length

$\mathsf{enc} \longrightarrow \mathsf{len}_{\mathsf{enc}} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

Length of messages

Alice

$\{\langle N_a, \mathsf{pk}(k_A)\rangle\}_{\mathsf{pk}(k_B)}$

$\{\langle N_a, N_b, \mathsf{pk}(k_B)\rangle\}_{\mathsf{pk}(k_A)}$

$\{N\}_{\mathsf{pk}(k_A)}$

Size of dummy message differs

$\mathsf{pk}(k_A)?$

Bob

Length functions

Length of a constructor only depends on the length of its arguments

Demonstration

Private_Authentication_Protocol.txt — Edited

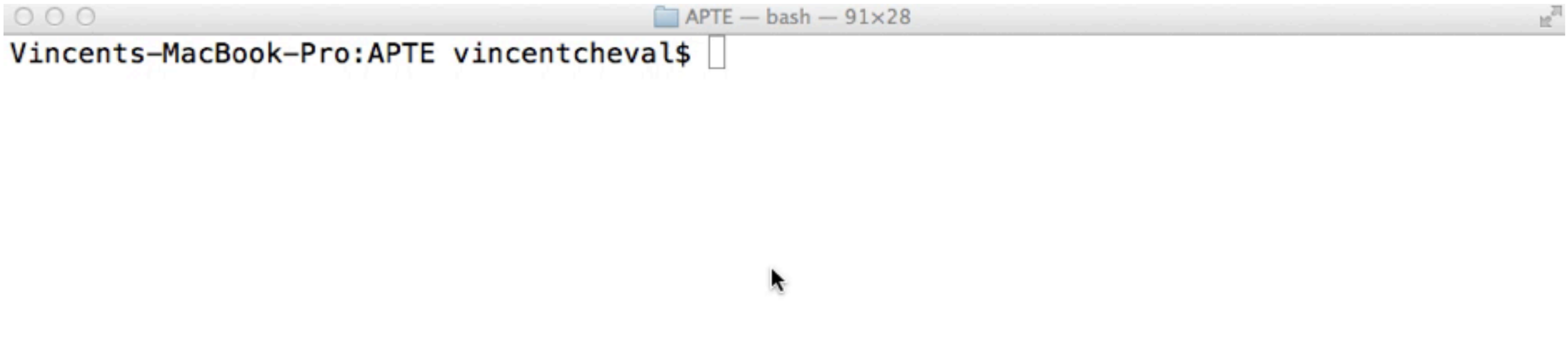
```
free c.

### Definitions of the classic length functions

length aenc [constant=0; arguments = 1,0].
length pk [constant=0; arguments = 1].
length tuple(2) [constant=1; arguments = 1,1].

### Description of the role of Alice

let process_Alice k_a k_b =
  new N_a;
  out(c,aenc((N_a,pk(k_a)),pk(k_b)));
  in(c,x).

### Description of the role of Bob

let process_Bob k_a k_b =
  in(c,x);
  let (na,pka) = adec(x,k_b) in
  if pka = pk(k_a)
  then new N_b; out(c,aenc((na,N_b,pk(k_b)),pk(k_a)))
  else new N; out(c,aenc(N,pk(k_a))).

### Main

let instance1 =
  new k_a ; new k_b ; new k_c ; out(c,pk(k_a)) ; out(c,pk(k_b)) ; out(c,pk(k_c));
  ( process_Alice k_a k_b | process_Bob k_a k_b ).
```

Demonstration

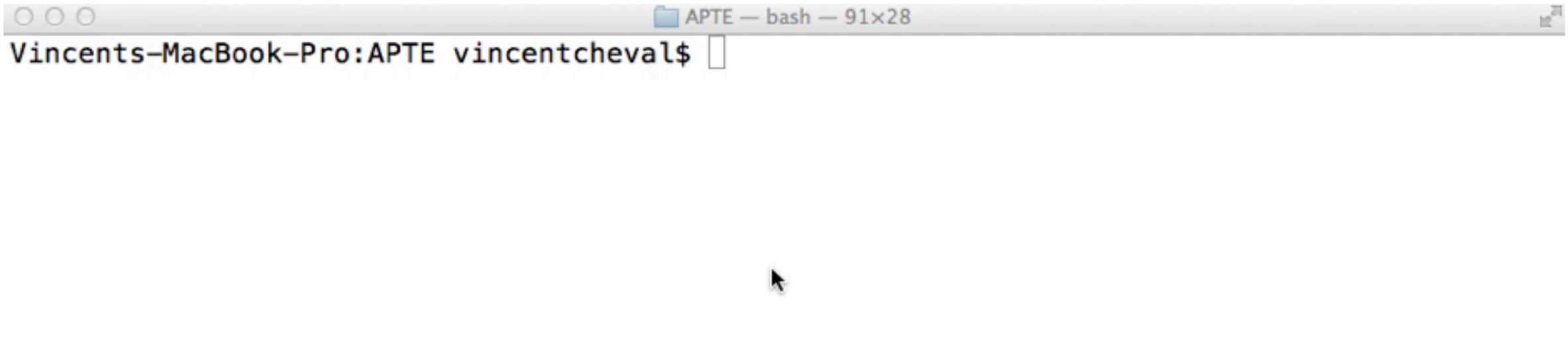
Private_Authentication_Protocol.txt — Edited

```
free c.

### Definitions of the classic length functions

length aenc [constant=0; arguments = 1,0].
length pk [constant=0; arguments = 1].
length tuple(2) [constant=1; arguments = 1,1].

### Description of the role of Alice

let process_Alice k_a k_b =
  new N_a;
  out(c,aenc((N_a,pk(k_a)),pk(k_b)));
  in(c,x).

### Description of the role of Bob

let process_Bob k_a k_b =
  in(c,x);
  let (na,pka) = adec(x,k_b) in
  if pka = pk(k_a)
  then new N_b; out(c,aenc((na,N_b,pk(k_b)),pk(k_a)))
  else new N; out(c,aenc(N,pk(k_a))).

### Main

let instance1 =
  new k_a ; new k_b ; new k_c ; out(c,pk(k_a)) ; out(c,pk(k_b)) ; out(c,pk(k_c));
  ( process_Alice k_a k_b | process_Bob k_a k_b ).
```

Demonstration



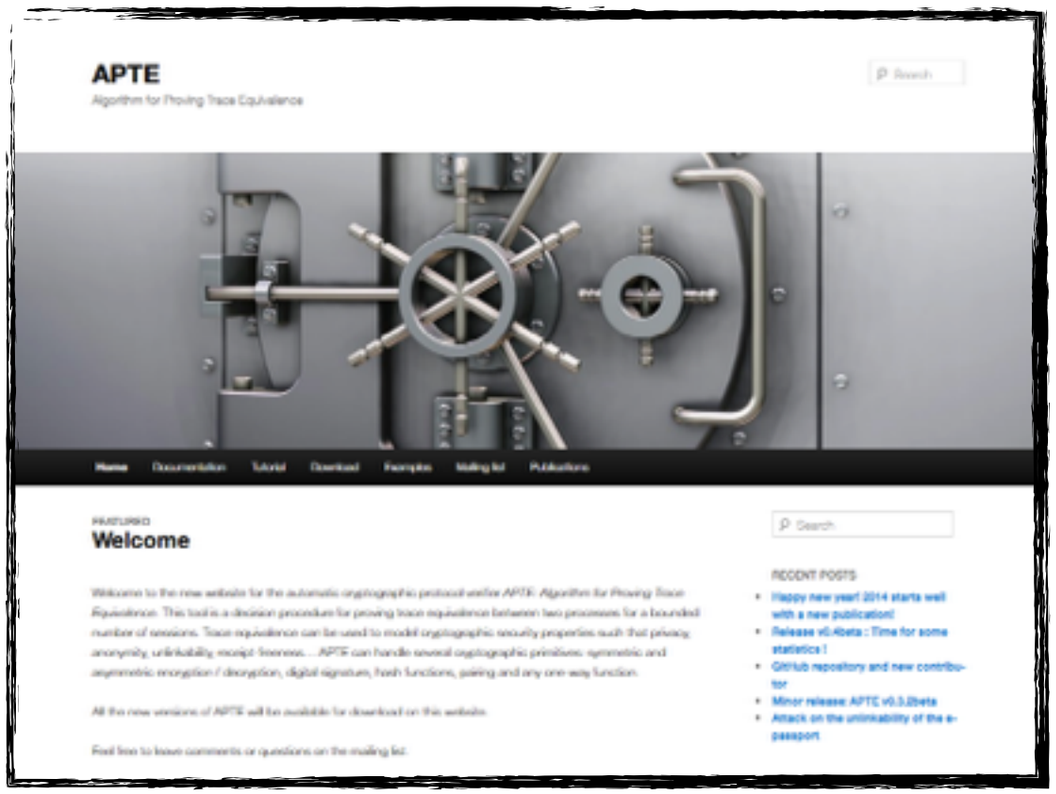
Demonstration



APTE

APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of constraint system



Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages

IJCAR-CCD'10, CCS-CCD'11, Cheval'12, CAV-CCP'13

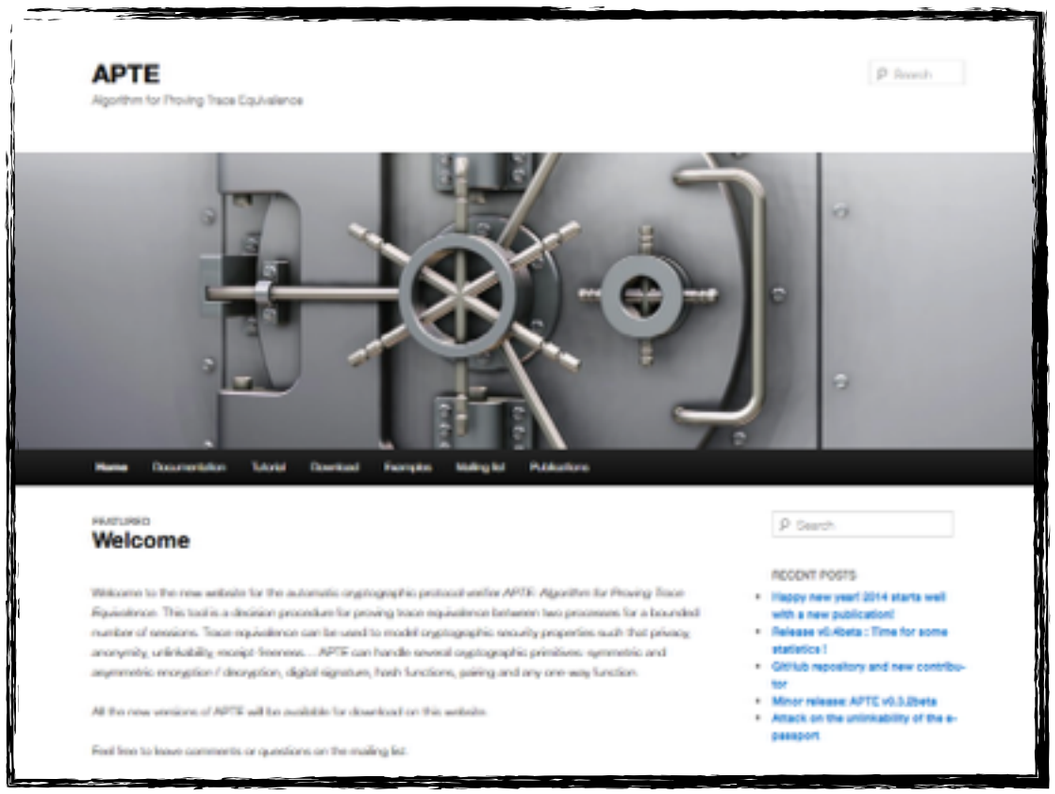
APTE

APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of constraint system

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages



Rediscovered 2 known attacks on the e-passport protocol

Discovered a new attack on the e-passport and private authentication protocol

APTE

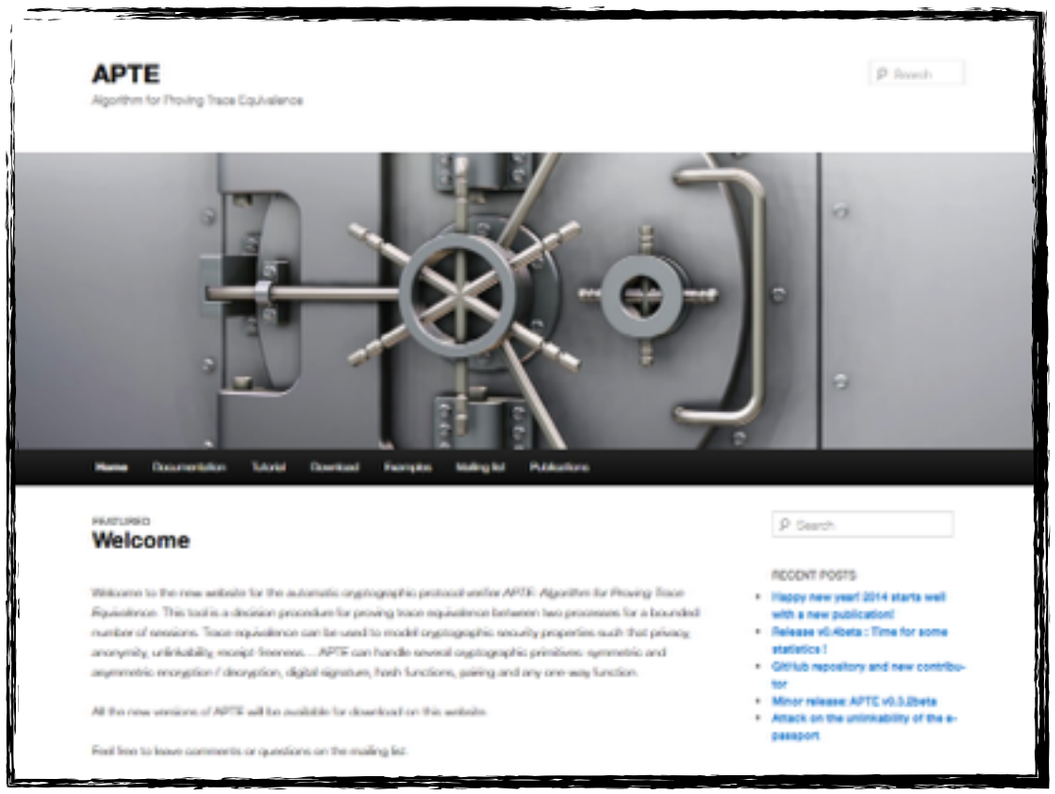
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages



Rediscovered 2 known attacks on the e-passport protocol

Discovered a new attack on the e-passport and private authentication protocol

APTE

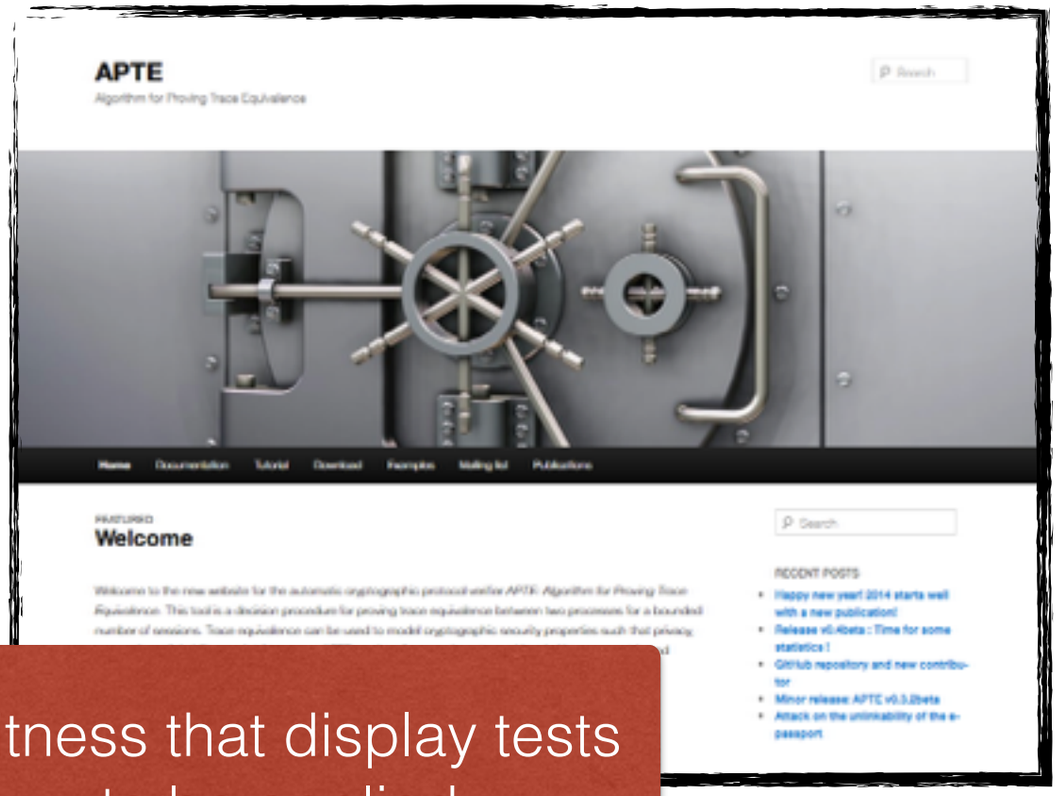
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages



Witness that display tests to be applied

Rediscovered 2 known attacks on the e-passport protocol

Discovered a new attack on the e-passport and private authentication protocol

APTE

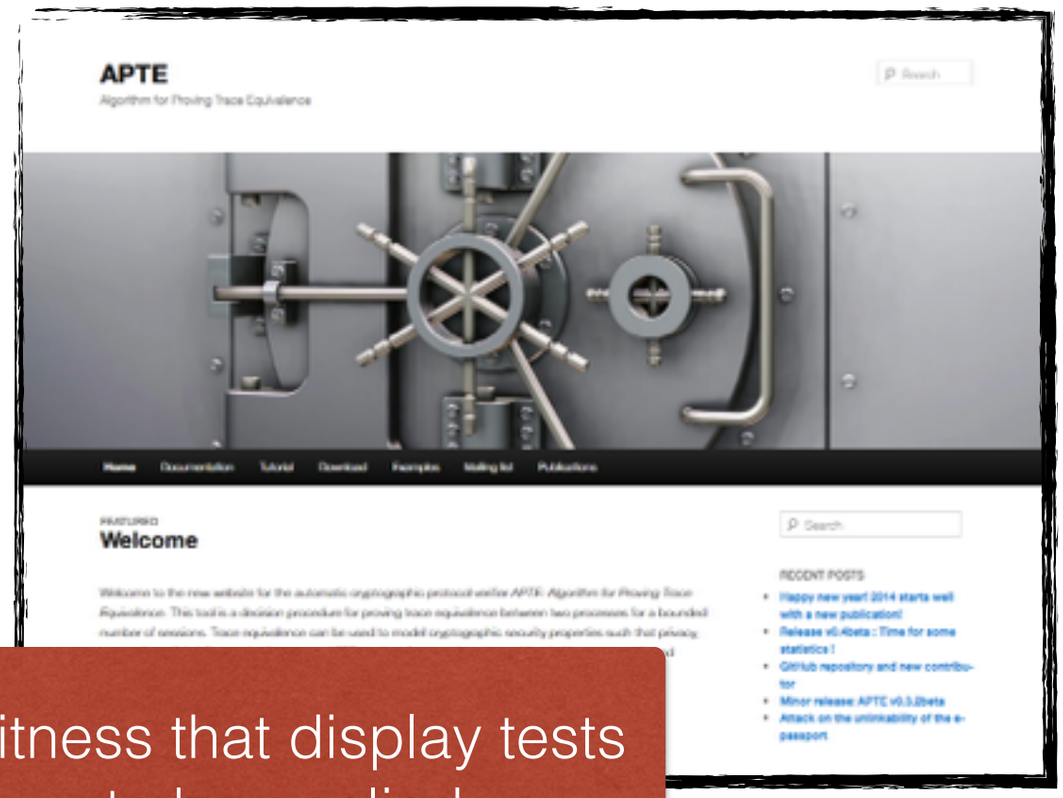
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages
- **Handle equivalence with respect to computation time**



Witness that display tests to be applied

APTE

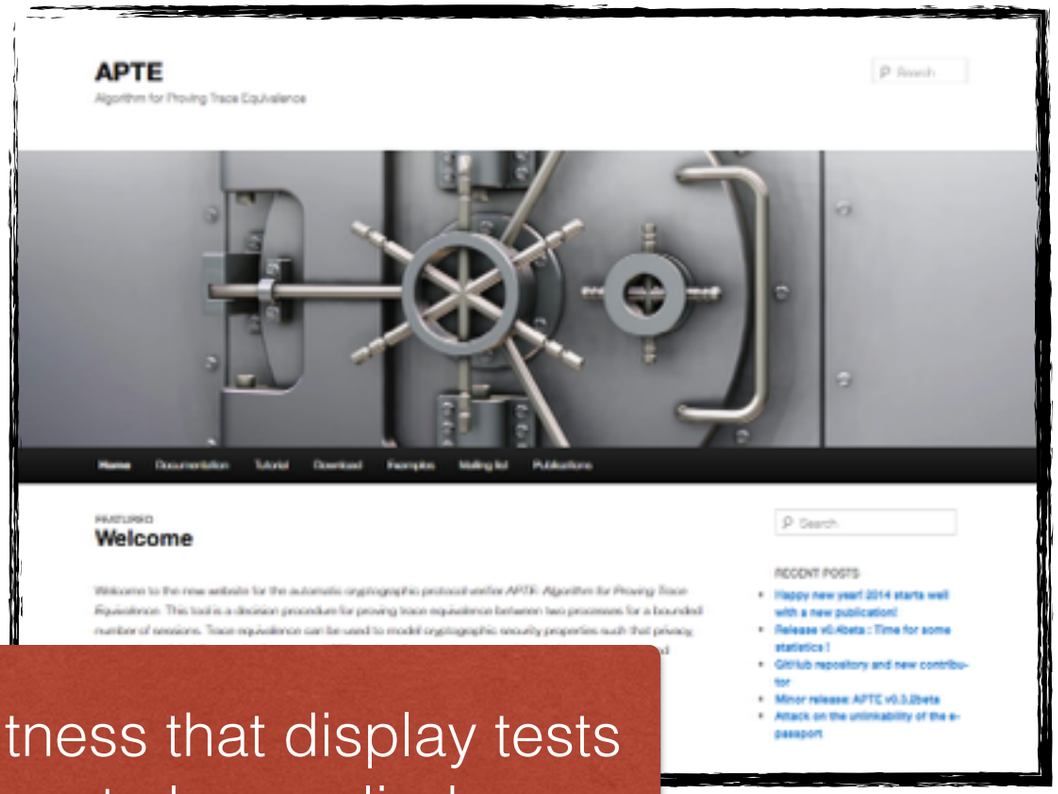
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages
- **Handle equivalence with respect to computation time**
- **Optimisation of interleaving search space [BaeldeDelauneHirschi'14]**



Witness that display tests to be applied

APTE

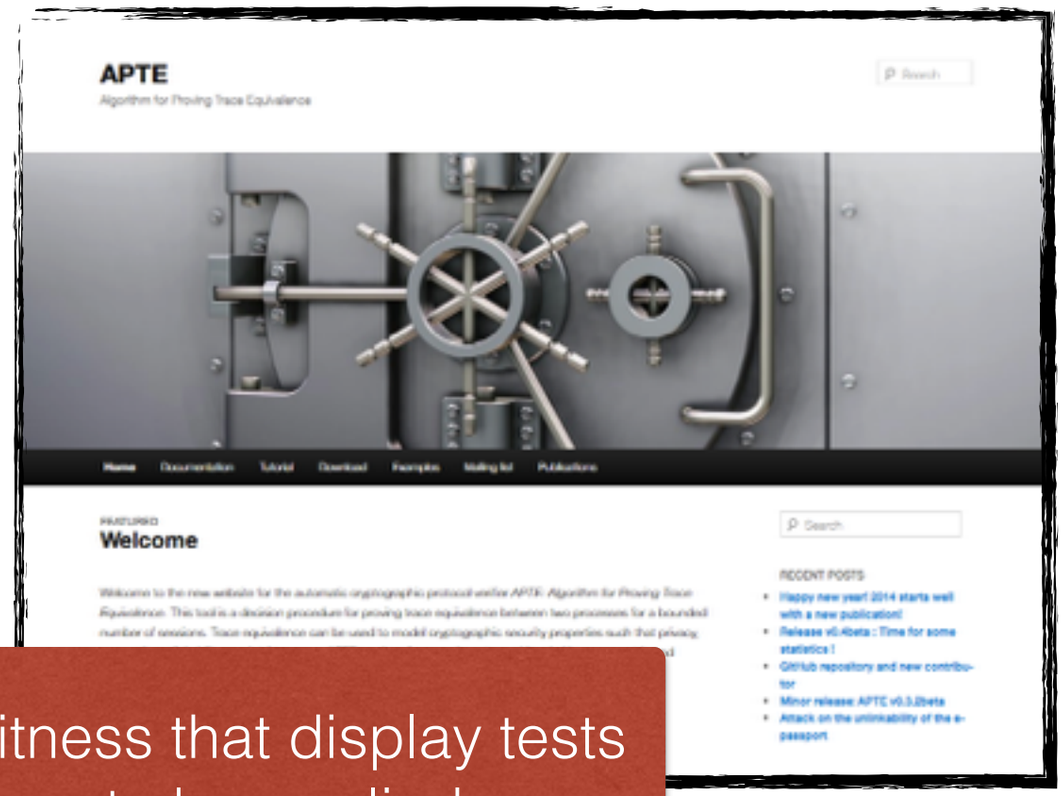
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages
- **Handle equivalence with respect to computation time**
- **Optimisation of interleaving search space [BaeldeDelauneHirschi'14]**



Witness that display tests to be applied

- **Concurrent implementation (for multicore and distributed computing)**

APTE

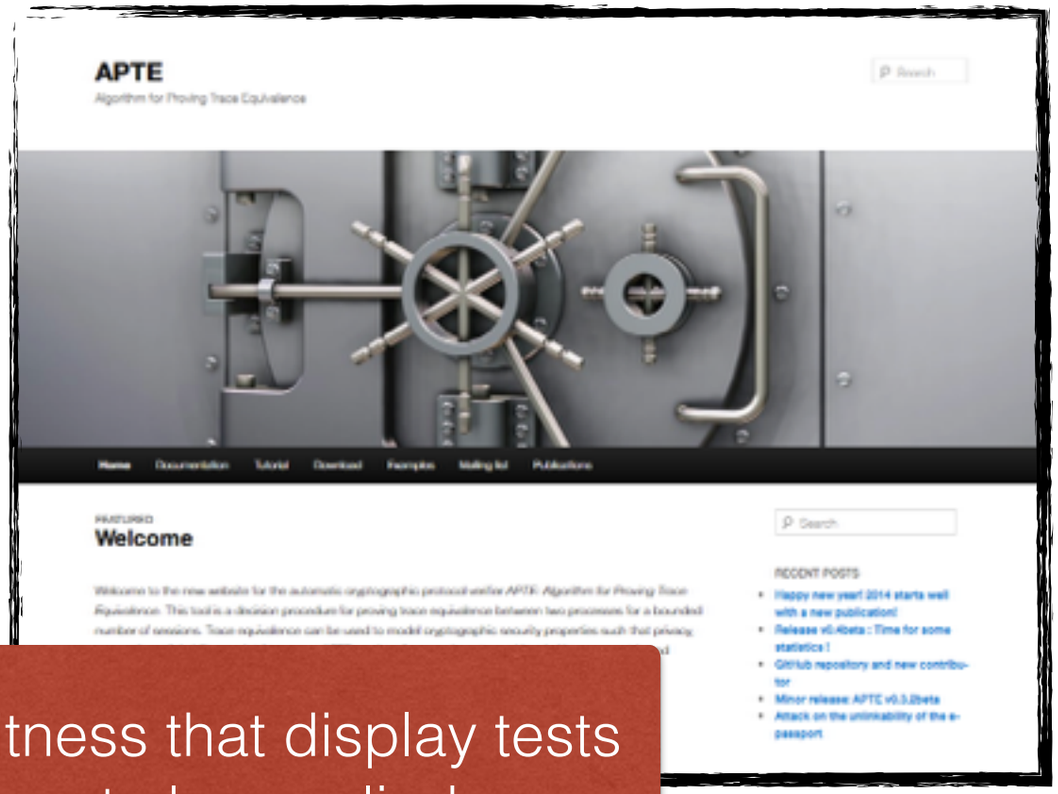
APTE v0.4beta
http://projects.lsv.ens-cachan.fr/APTE/

Based on equivalence of … m

More cryptographic primitives

Content:

- Equivalence between bounded processes with fix set of cryptographic primitives
- Display witness of non-equivalence
- Handle equivalence with respect to length of messages
- **Handle equivalence with respect to computation time**
- **Optimisation of interleaving search space [BaeldeDelauneHirschi'14]**



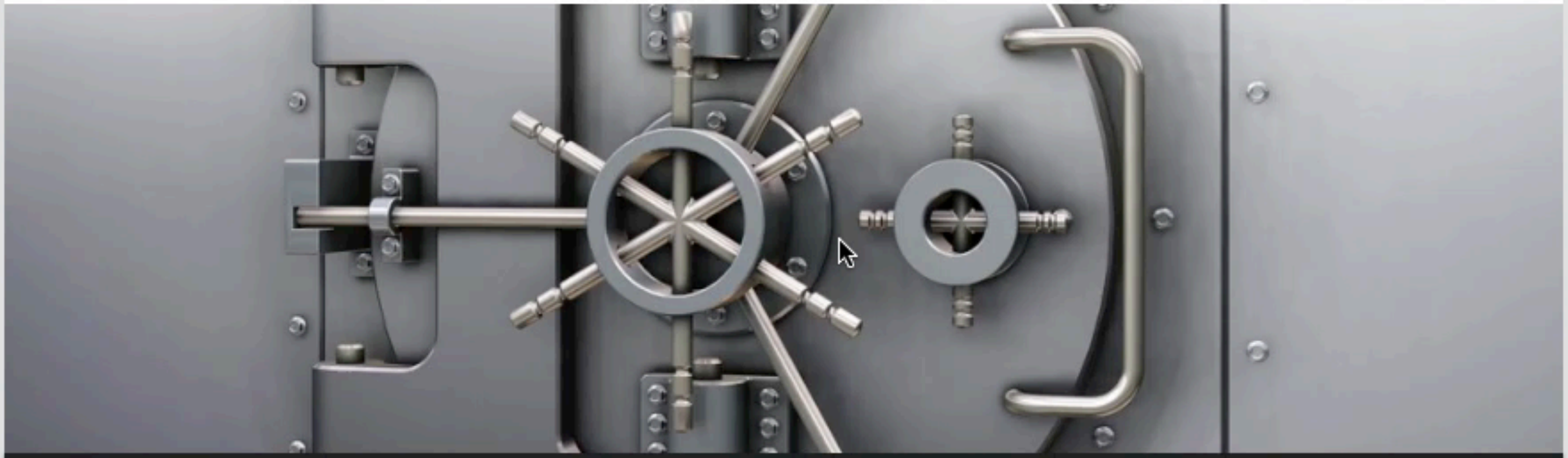
Witness that display tests to be applied

- **Concurrent implementation (for multicore and distributed computing)**
- **Optimisation of constraint equivalence solving**

APTE

Algorithm for Proving Trace Equivalence

Search



Home Documentation Tutorial Download Examples Mailing list Publications

FEATURED

Welcome

Search

RECENT POSTS

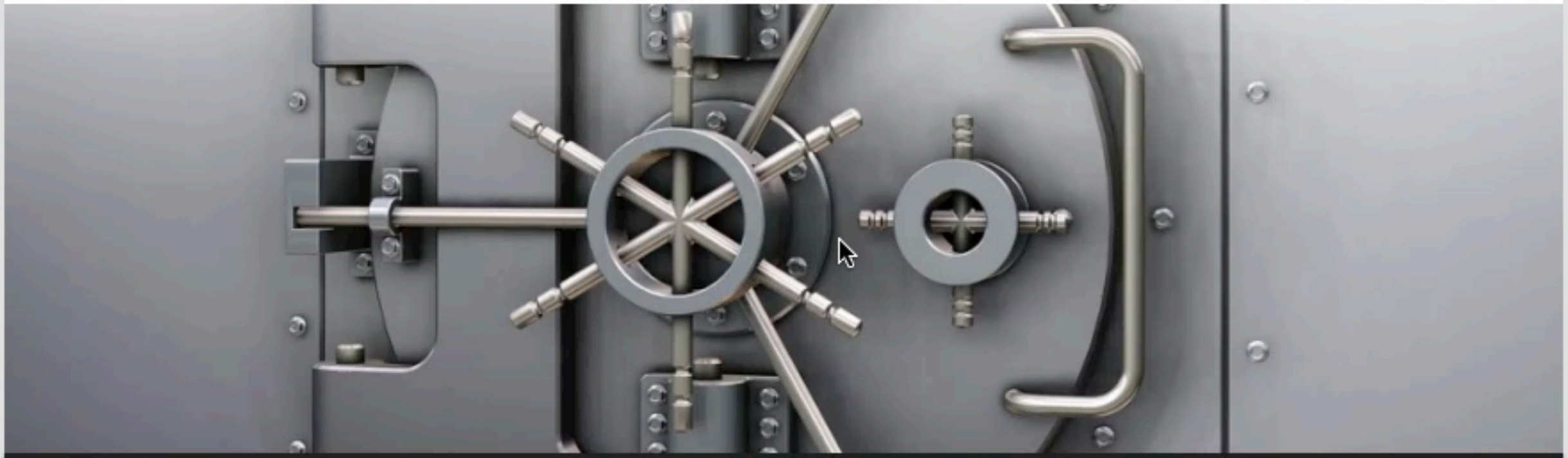
- ETAPS 2014

Welcome to the new website for the automatic cryptographic protocol verifier

APTE

Algorithm for Proving Trace Equivalence

Search



Home Documentation Tutorial Download Examples Mailing list Publications

FEATURED

Welcome

Search

RECENT POSTS

- ETAPS 2014

Welcome to the new website for the automatic cryptographic protocol verifier

Thank you !

APTE v0.4beta

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