La vérification formelle appliquée aux protocoles cryptographiques

Stéphanie Delaune

LSV, CNRS & ENS Cachan, Université Paris-Saclay, France

6 avril 2016

Les protocoles cryptographiques sont partout !

$\longrightarrow$ sécuriser nos communications : authentification sur les services de banque en ligne, confidentialité des données échangées, protection de nos données personnelles, ...

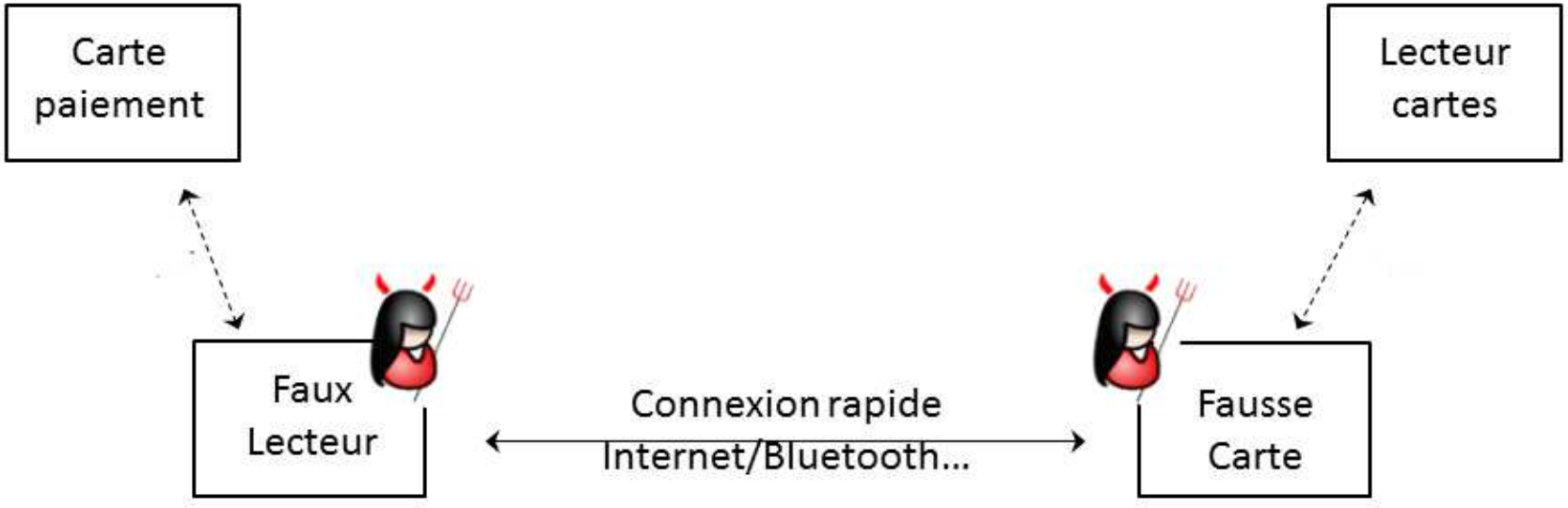
Le paiement sans contact ...

Plus de 30 millions de cartes de paiement sans contact sont en circulation en France.

Le paiement sans contact ...

Plus de 30 millions de cartes de paiement sans contact sont en circulation en France.

... est vulnérable à l'attaque par relais :



Problème : aucun dispositif ne permet d'assurer la proximité physique de la carte qui réalise la transaction.

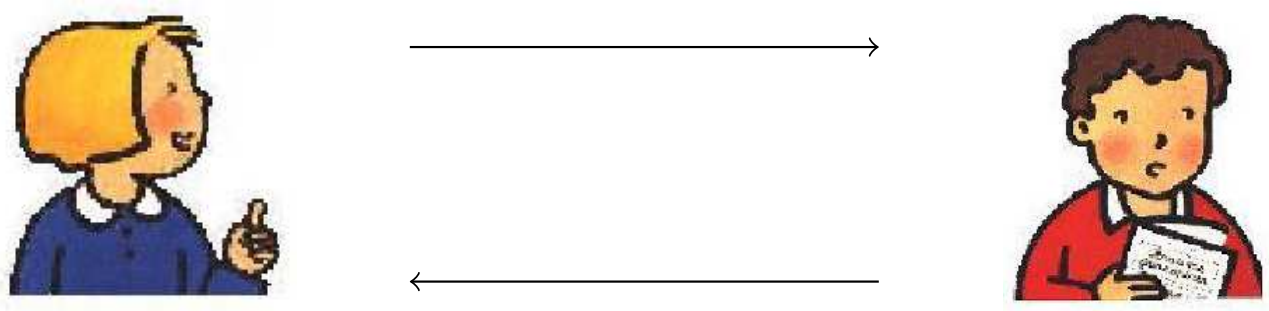
Les voitures connectées ...

Février 2011
Attaque par relais de systèmes d'accès et démarrage mains libres

Juillet 2015
Contrôle d'une Jeep à distance : essuie-glaces, radio, climatisation, ...
mais aussi les **freins** !

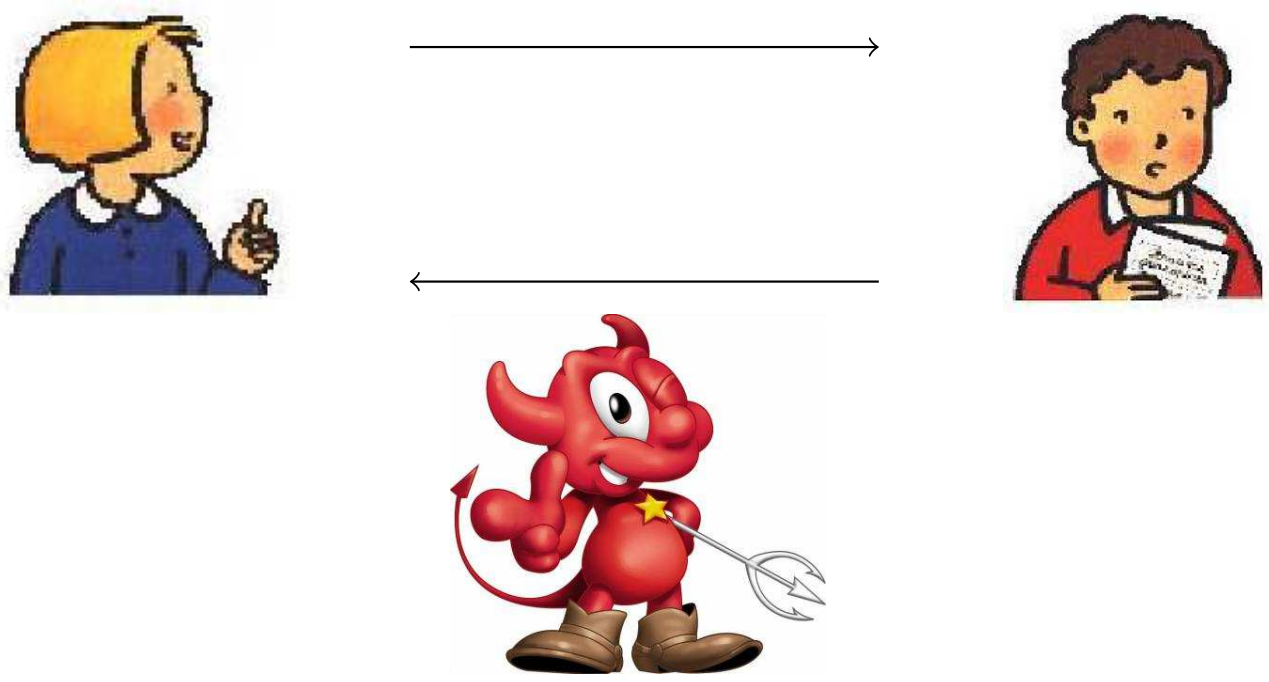
$\longrightarrow$ les experts s'inquiètent de la vulnérabilité des nouvelles générations de voitures dites connectées.

Un protocole cryptographique : qu'est-ce que c'est ?



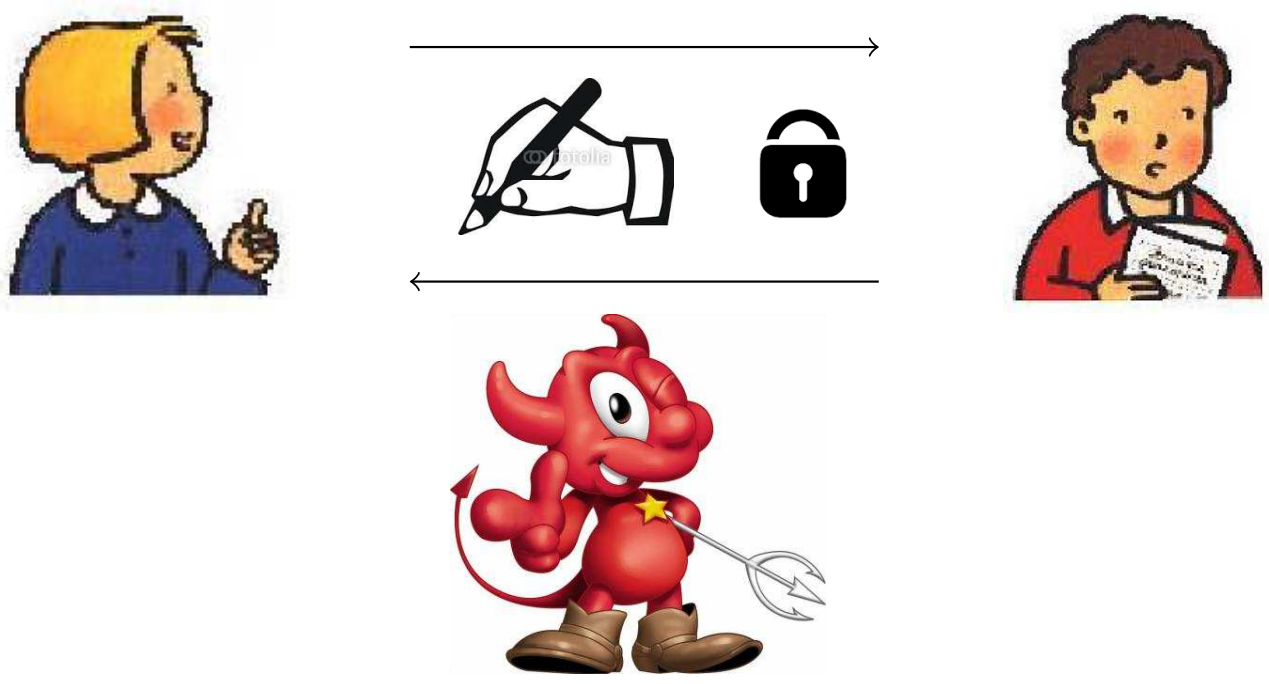
- Protocole : petit programme explicitant les messages échangés

Un protocole cryptographique : qu'est-ce que c'est ?



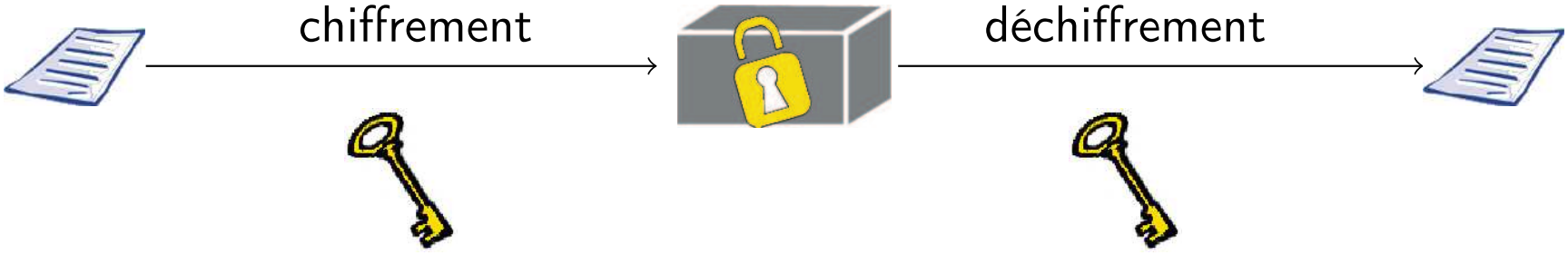
- Protocole : petit programme explicitant les messages échangés

Un protocole cryptographique : qu'est-ce que c'est ?

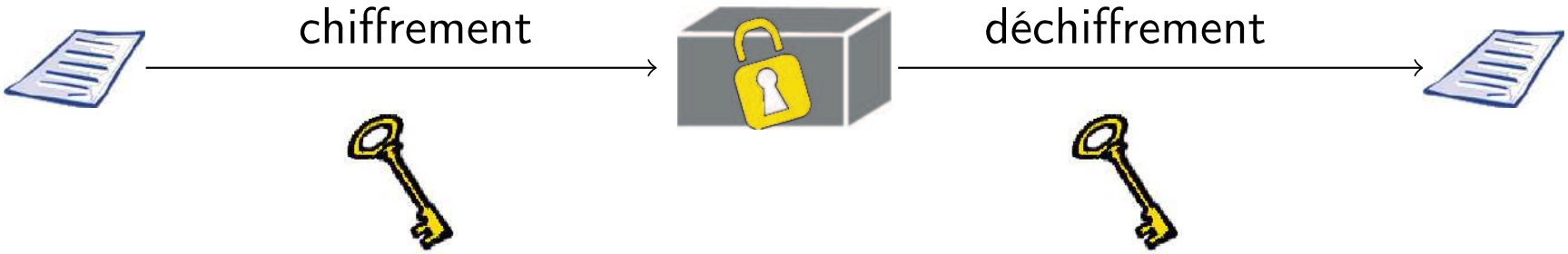


- Protocole : petit programme explicitant les messages échangés
- Cryptographique : utilisant des primitives cryptographiques (e.g. chiffrement symétrique, asymétrique, signature, . . .)

Chiffrement symétrique

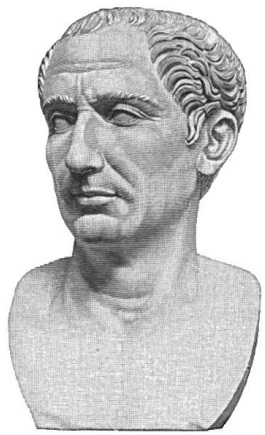


Chiffrement symétrique



scytale (400 av. JC)

César (50 av. JC)

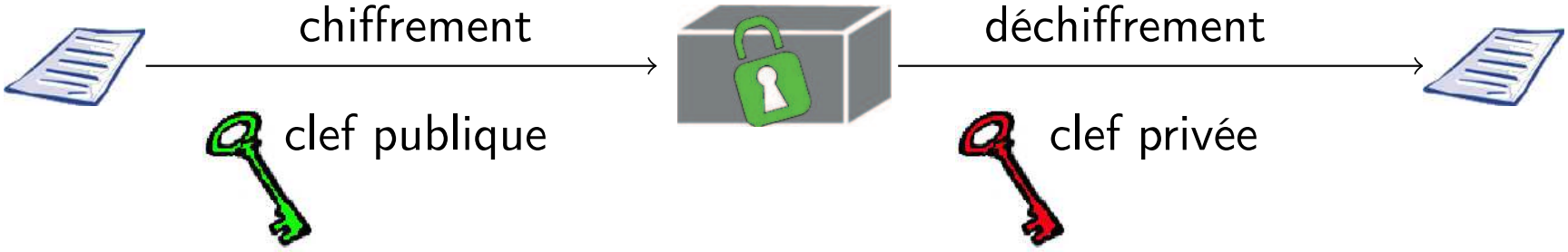


Enigma (1940)

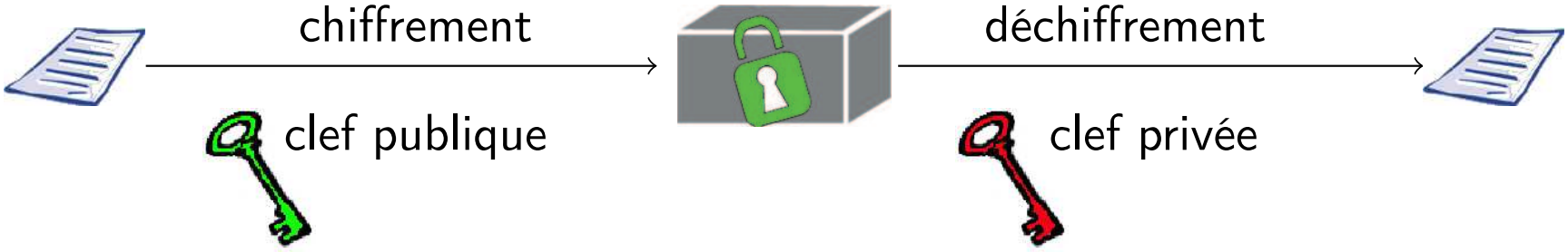
Quelques algorithmes plus récents :

- Data Encryption Standard (1977) ;
- Advanced Encryption Standard (2000).

Chiffrement asymétrique

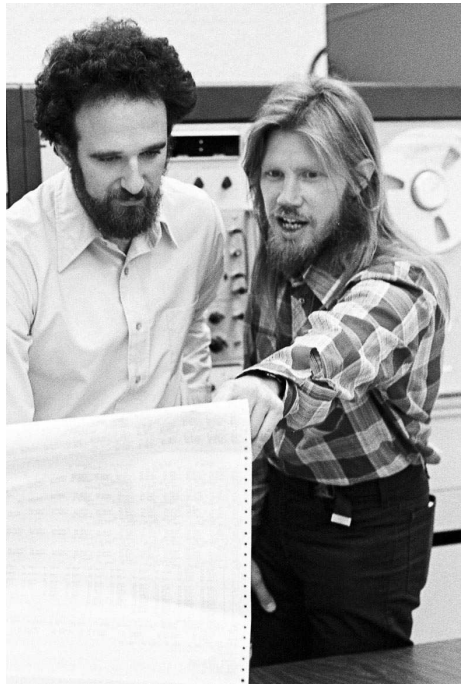


Chiffrement asymétrique



Quelques exemples :

- 1976 : 1er système
  W. Diffie et M. Hellman
  $\longrightarrow$ Prix Turing 2016
- 1977 : système RSA
  R. Rivest, A. Shamir, et L. Adleman



$\longrightarrow$ Ces systèmes sont toujours utilisés de nos jours.

Protocole de Denning Sacco (1981)

$\longrightarrow$ version simplifiée

$$\text{aenc}(\text{sign}(k_{AB}, \text{priv}(A)), \text{pub}(B))$$

Est-ce un bon protocole d'établissement de clefs ?

Protocole de Denning Sacco (1981)

$\longrightarrow$ version simplifiée

$$\text{aenc}(\text{sign}(k_{AB}, \text{priv}(A)), \text{pub}(B)) \longrightarrow$$

Est-ce un bon protocole d'établissement de clefs ? Non !

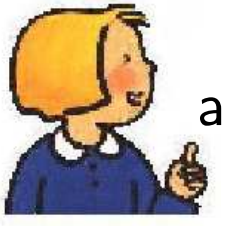
Protocole de Denning Sacco (1981)

$\longrightarrow$ version simplifiée

$$\text{aenc}(\text{sign}(k_{AB}, \text{priv}(A)), \text{pub}(B)) \longrightarrow$$

Est-ce un bon protocole d'établissement de clefs ? Non !

Description de l'attaque :



$$\text{aenc}(\text{sign}(k_{AC}, \text{priv}(A)), \text{pub}(C)) \longrightarrow$$

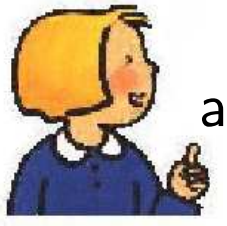
Protocole de Denning Sacco (1981)

$\longrightarrow$ version simplifiée

$$\text{aenc}(\text{sign}(k_{AB}, \text{priv}(A)), \text{pub}(B))$$

Est-ce un bon protocole d'établissement de clefs ? Non !

Description de l'attaque :



$$\text{aenc}(\text{sign}(k_{AC}, \text{priv}(A)), \text{pub}(C))$$

$\text{sign}(k_{AC}, \text{priv}(A))$

k_{AC}

$$\text{aenc}(\text{sign}(k_{AC}, \text{priv}(A)), \text{pub}(B))$$

Les protocoles utilisés de nos jours ...

... comportent de nombreuses failles qualifiées de logiques.

Connexion HTTPS Barghavan et al. 2015

Une attaque du type “homme du milieu“ permet de faire revivre un vieux mode de chiffrement.

$\longrightarrow$ environ 10% des sites sont vulnérables

`https://freakattack.com`

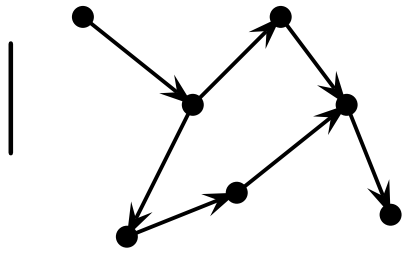
Passeport électronique Chothia et al. 2010

Des messages d'erreurs trop précis permettent de tracer le porteur d'un passeport français.

La vérification formelle appliquée aux protocoles cryptographiques

Est-ce que le protocole *satisfait* la propriété de sécurité ?

Modélisation



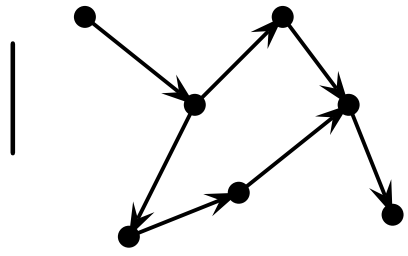
$\models$

φ

La vérification formelle appliquée aux protocoles cryptographiques

Est-ce que le protocole *satisfait* la propriété de sécurité ?

Modélisation



$\models$ φ

Deux tâches principales

1. Modélisation : protocoles, propriétés de sécurité, sans oublier notre adversaire !
2. Mise au point d'algorithmes de vérification.

Partie I

Modélisation

Deux grandes familles de modèles ...

... avec leurs avantages et leurs inconvénients.

Modèle dit calculatoire

- + messages et adversaire représentés précisément
- − preuves souvent manuelles, longues, et difficiles

Modèle dit **symbolique**

- − messages et adversaire sont représentés de façon abstraite
- + preuves automatiques

Deux grandes familles de modèles ...

... avec leurs avantages et leurs inconvénients.

Modèle dit calculatoire

- + messages et adversaire représentés précisément
- − preuves souvent manuelles, longues, et difficiles

Modèle dit **symbolique**

- − messages et adversaire sont représentés de façon abstraite
- + preuves automatiques

De nombreux résultats ont permis de rapprocher ces deux modèles.

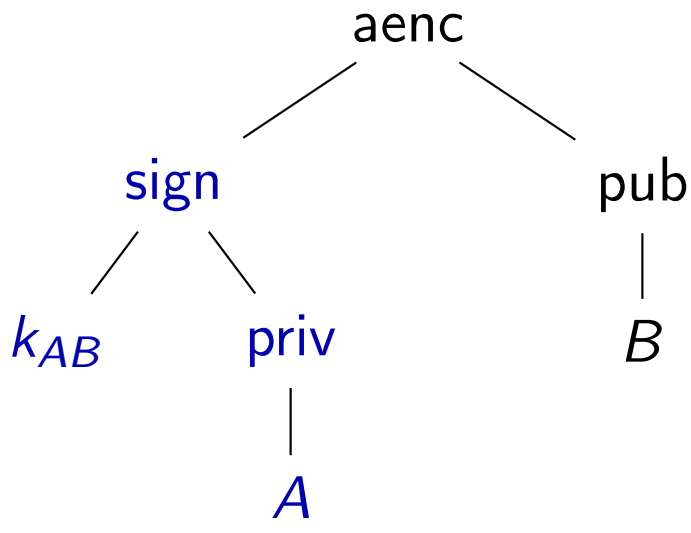
$\longrightarrow$ Abadi & Rogaway 2000

Les messages

Ils sont abstraits par des termes.

Retour sur le protocole de Denning Sacco

aenc(sign(k_{AB}, priv(A)), pub(B))



Le pouvoir de notre adversaire

« L'adversaire connaît le système » : C. Shannon ;
Il contrôle le réseau de communication.
$\longrightarrow$ lire, intercepter, et envoyer des messages.

Le pouvoir de notre adversaire

« L'adversaire connaît le système » : C. Shannon ;
Il contrôle le réseau de communication.
$\longrightarrow$ lire, intercepter, et envoyer des messages.

Il peut construire de nouveaux messages. Oui, mais lesquels ?

Chiffrement asymétrique :

$$\frac{x \quad \mathsf{pub}(y)}{\mathsf{aenc}(x, \mathsf{pub}(y))} \qquad \frac{\mathsf{aenc}(x, \mathsf{pub}(y)) \quad \mathsf{priv}(y)}{x}$$

Le pouvoir de notre adversaire

« L'adversaire connaît le système » : C. Shannon ;
Il contrôle le réseau de communication.
$\longrightarrow$ lire, intercepter, et envoyer des messages.

Il peut construire de nouveaux messages. Oui, mais lesquels ?

Chiffrement asymétrique :

$$\frac{x \quad \mathsf{pub}(y)}{\mathsf{aenc}(x, \mathsf{pub}(y))} \qquad \frac{\mathsf{aenc}(x, \mathsf{pub}(y)) \quad \mathsf{priv}(y)}{x}$$

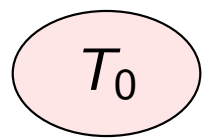
Signature

$$\frac{x \quad \mathsf{priv}(y)}{\mathsf{sign}(x, \mathsf{priv}(y))} \qquad \frac{\mathsf{sign}(x, \mathsf{priv}(y))}{x}$$

Exemple

$$T_0 = \{m,\ \mathsf{pub}(a)\}$$

Quels messages l'adversaire peut-il construire à partir de T_0 ?

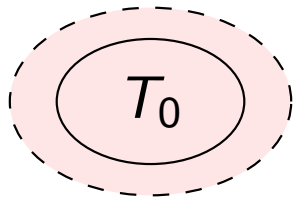


Exemple

$$T_0 = \{m,\ \mathsf{pub}(a)\}$$

Quels messages l'adversaire peut-il construire à partir de T_0 ?

$\mathsf{aenc}(m, \mathsf{pub}(a))$



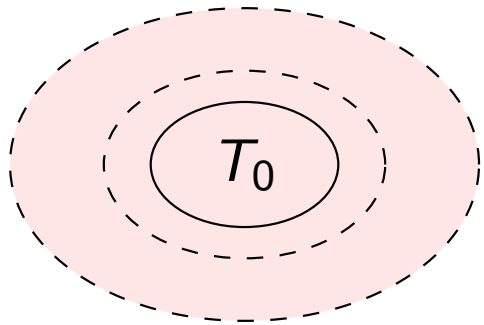
Exemple

$$T_0 = \{m,\ \mathsf{pub}(a)\}$$

Quels messages l'adversaire peut-il construire à partir de T_0 ?

$\mathsf{aenc}(m, \mathsf{pub}(a))$

$\mathsf{aenc}(\mathsf{aenc}(m, \mathsf{pub}(a)), \mathsf{pub}(a))$



Exemple

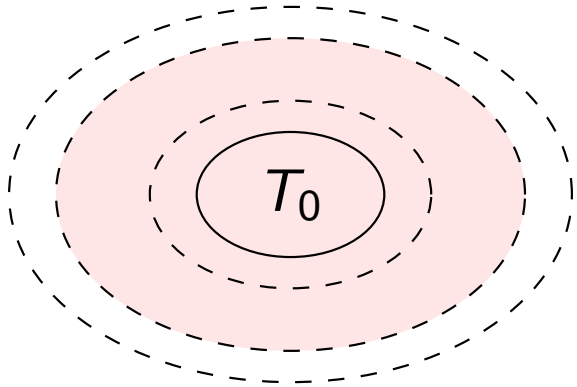
$$T_0 = \{m,\ \mathsf{pub}(a)\}$$

Quels messages l'adversaire peut-il construire à partir de T_0 ?

$\mathsf{aenc}(m, \mathsf{pub}(a))$

$\mathsf{aenc}(\mathsf{aenc}(m, \mathsf{pub}(a)), \mathsf{pub}(a))$

$\ldots$



$\longrightarrow$ beaucoup !

Le problème de déduction

Entrées : un ensemble fini de termes T et un terme u.

Sortie : Est-ce que u est dérivable à partir de T ?

Le problème de déduction

Entrées : un ensemble fini de termes T et un terme u.

Sortie : Est-ce que u est dérivable à partir de T ?

Exemple

- $T = \{\mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(C));\ \mathsf{priv}(C);\ \mathsf{pub}(B)\}$; et
- $u = \mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(B))$

Est-ce que u est dérivable à partir de T ?

Le problème de déduction

Entrées : un ensemble fini de termes T et un terme u.

Sortie : Est-ce que u est dérivable à partir de T ?

Exemple

- $T = \{\mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(C));\ \mathsf{priv}(C);\ \mathsf{pub}(B)\}$; et
- $u = \mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(B))$

Est-ce que u est dérivable à partir de T ?

$\longrightarrow$ Oui, on a $T \vdash u$.

$$\dfrac{\dfrac{\mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(C)) \quad \mathsf{priv}(C)}{\mathsf{sign}(k_{AC}, \mathsf{priv}(A))} \qquad \mathsf{pub}(B)}{\mathsf{aenc}(\mathsf{sign}(k_{AC}, \mathsf{priv}(A)), \mathsf{pub}(B))}$$

Le problème de déduction

Résultat

Le problème de déduction est décidable en temps polynomial.

$\longrightarrow$ pour les primitives classiques

Le problème de déduction

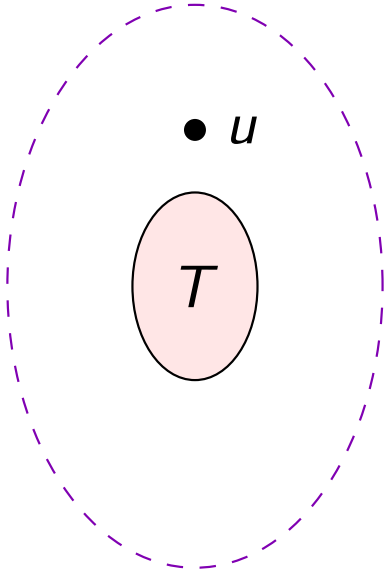
Résultat

Le problème de déduction est décidable en temps polynomial.

$\longrightarrow$ pour les primitives classiques

Algorithme

1. Calcul des sous-termes de T et de u ;



Le problème de déduction

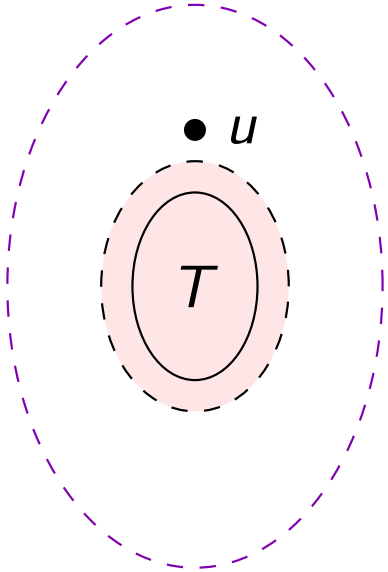
Résultat

Le problème de déduction est décidable en temps polynomial.

$\longrightarrow$ pour les primitives classiques

Algorithme

1. Calcul des sous-termes de T et de u ;
2. Saturation de T avec les sous-termes déductibles en une étape ;



Le problème de déduction

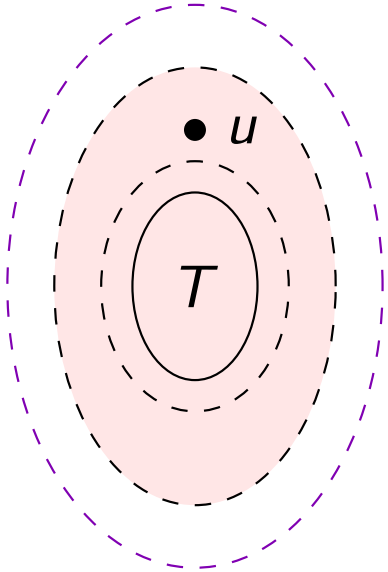
Résultat

Le problème de déduction est décidable en temps polynomial.

$\longrightarrow$ pour les primitives classiques

Algorithme

1. Calcul des sous-termes de T et de u ;
2. Saturation de T avec les sous-termes déductibles en une étape ;



Le problème de déduction

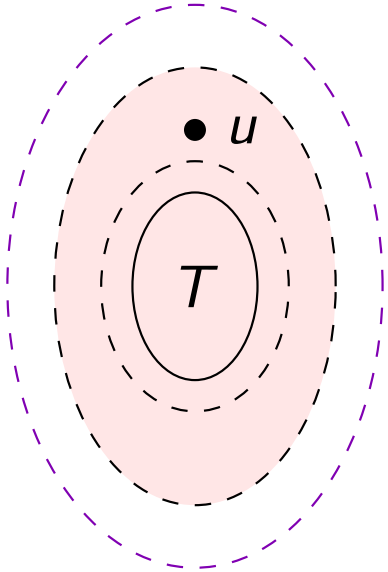
Résultat

Le problème de déduction est décidable en temps polynomial.

$\longrightarrow$ pour les primitives classiques

Algorithme

1. Calcul des sous-termes de T et de u ;
2. Saturation de T avec les sous-termes déductibles en une étape ;
3. Si u est atteint lors de la saturation alors répondre Oui ; sinon répondre Non.



Terminaison, correction, et complétude

Terminaison

Cet algorithme termine.

$\longrightarrow$ L'ensemble des sous-termes est fini.

Terminaison, correction, et complétude

Terminaison

Cet algorithme termine.

$\longrightarrow$ L'ensemble des sous-termes est fini.

Correction

Si l'algorithme répond Oui alors u est dérivable à partir de T.

$\longrightarrow$ Seuls des termes dérivables sont ajoutés par saturation.

Terminaison, correction, et complétude

Terminaison

Cet algorithme termine.

$\longrightarrow$ L'ensemble des sous-termes est fini.

Correction

Si l'algorithme répond Oui alors u est dérivable à partir de T.

$\longrightarrow$ Seuls des termes dérivables sont ajoutés par saturation.

Complétude (plus difficile)

Si u est dérivable à partir de T, alors l'algorithme répond Oui.

Terminaison, correction, et complétude

Terminaison

Cet algorithme termine.

$\longrightarrow$ L'ensemble des sous-termes est fini.

Correction

Si l'algorithme répond Oui alors u est dérivable à partir de T.

$\longrightarrow$ Seuls des termes dérivables sont ajoutés par saturation.

Complétude (plus difficile)

Si u est dérivable à partir de T, alors l'algorithme répond Oui.

Propriété de localité — Mc Allester 1993

Si $T \vdash u$ alors il existe une dérivation dont tous les noeuds sont étiquetés par des sous-termes de T ou de u.

Un protocole : qu'est-que c'est ?

C'est une liste d'instructions (out, in, new) à suivre pour chaque participant !

Retour sur le protocole de Denning Sacco

$$\xrightarrow{\text{aenc}(\text{sign}(k_{AB}, \text{priv}(A)), \text{pub}(B))}$$

Rôle Alice : $Alice(a, b) =$ new k;
out(aenc(sign(k, priv(a)), pub(b)))

Rôle Bob : $Bob(b, a) =$ in(aenc(sign(x, priv(a)), pub(b)))

Qu'est-ce qu'un protocole sûr ?

Vérifiabilité

Confidentialité

Absence de reçu

Non-traçabilité

Authentification

Équité

Anonymat

Qu'est-ce qu'un protocole sûr ?

Vérifiabilité

Confidentialité

Absence de reçu

Non-traçabilité

Authentification

Équité

Anonymat

Oui, mais qu'est-ce que cela signifie exactement ?

$\longrightarrow$ des définitions formelles n'existent pas toujours !

Qu'est-ce qu'un protocole sûr ?

Vérifiabilité

Confidentialité

Absence de reçu

Non-traçabilité

Authentification

Équité

Anonymat

Oui, mais qu'est-ce que cela signifie exactement ?

$\longrightarrow$ des définitions formelles n'existent pas toujours !

Confidentialité : Un agent malhonnête ne doit pas être en mesure de dériver la donnée confidentielle.

Authentification (faible) : Si un agent B termine son protocole en pensant avoir communiqué avec un agent A, alors A a bien joué le protocole (et a priori avec B).

Partie II

Vérification

Comment vérifier ces protocoles ?

Notre but :

- faire des preuves mathématiques rigoureuses ;
- d'une façon automatique.

Construire une machine à détecter les bugs !

Comment vérifier ces protocoles ?

Notre but :

- faire des preuves mathématiques rigoureuses ;
- d'une façon automatique.

Construire une machine à détecter les bugs !

A. Turing (1936)

Une telle machine n'existe pas ...

... même dans le cas particulier des protocoles cryptographiques.

Mais alors que faisons nous ?

Des procédures approchées pour la preuve de protocoles :

- outil ProVerif basé sur la résolution de clauses de Horn
  $\longrightarrow$ voir exposé V. Cortier (2015)
- outil Tamarin (avec un mode interactif)
  $\longrightarrow$ développé à l'ETH Zurich

Mais alors que faisons nous ?

Des procédures approchées pour la preuve de protocoles :

- outil ProVerif basé sur la résolution de clauses de Horn
  $\longrightarrow$ voir exposé V. Cortier (2015)
- outil Tamarin (avec un mode interactif)
  $\longrightarrow$ développé à l'ETH Zurich

Des procédures dédiées à la recherche d'attaques :

- outil SATMC basé sur les solveurs SAT
- outils OFMC et APTE basés sur la résolution de systèmes de contraintes

$\longrightarrow$ Ces procédures sont complètes pour certaines classes de protocoles et de scénarios (nombre borné de sessions).

Utilisation de solveurs SAT

$\longrightarrow$ voir cours de G. Berry et séminaire de L. Simon (16 mars 2016)

Solveur SAT : Un programme qui décide automatiquement si une formule de logique propositionnelle est satisfaisable.

Exemple : $$f = [x_1 \vee x_2] \wedge [\overline{x_1} \vee x_3]$$

La formule f est satisfaisable. Une valuation possible est :

$$x_1 \rightarrow \text{faux}, \quad x_2 \rightarrow \text{vrai}, \quad x_3 \rightarrow \text{vrai}$$

Utilisation de solveurs SAT

$\longrightarrow$ voir cours de G. Berry et séminaire de L. Simon (16 mars 2016)

Solveur SAT : Un programme qui décide automatiquement si une formule de logique propositionnelle est satisfaisable.

Exemple : $$f = [x_1 \vee x_2] \wedge [\overline{x_1} \vee x_3]$$

La formule f est satisfaisable. Une valuation possible est :

$$x_1 \to \text{faux}, \quad x_2 \to \text{vrai}, \quad x_3 \to \text{vrai}$$

Idée générale :

1. Transformer notre problème en une instance de SAT ;
2. Faire appel à des solveurs SAT (e.g. MiniSat, Glucose . . .) pour terminer le travail.

Utilisation de solveurs SAT

Quelques hypothèses sont nécessaires sur nos protocoles :

- typage fort : les messages reçus respectent précisément le format attendu ;

  $\longrightarrow$ pas toujours très réaliste

- scénarios de longueurs bornés :

  Est-ce qu'il existe une attaque en k étapes ?

$\longrightarrow$ Un codage SAT naif est alors envisageable
(une variable propositionnelle par fait et par étape)

Outil SATMC

Thèse de L. Compagna 2007

1. un calcul grossier des états accessibles via des techniques utilisées en planification ;
   $\rightarrow$ objectif : limiter le nombre de variables propositionnelles
2. une formule SAT encodant précisément le problème ;
3. appel à divers SAT solveurs (*e.g.* MiniSat).

Outil SATMC

Thèse de L. Compagna 2007

1. un calcul grossier des états accessibles via des techniques utilisées en planification ;
   $\rightarrow$ objectif : limiter le nombre de variables propositionnelles
2. une formule SAT encodant précisément le problème ;
3. appel à divers SAT solveurs (*e.g.* MiniSat).

Découverte d'une attaque sur le protocole Single Sign-On

A. Armando et al. 2011

Possibilité d'accèder aux différents comptes (e.g. GMail, Google Calendar) d'un utilisateur.

1. créer une application malhonnête ;
2. faire en sorte que l'utilisateur y accède.

Approche par résolution de contraintes (OFMC, APTE)

Accéder à l'infini : un rêve impossible ?

Approche par résolution de contraintes (OFMC, APTE)

Accéder à l'infini : un rêve impossible ?

Étape 1 :

Une exploration symbolique de toutes les traces possibles ;

L'infinité d'exécutions possible est représentée par un nombre fini de systèmes de contraintes.

$\longrightarrow$ cet ensemble peut s'avérer grand

Étape 2 :

Une procédure pour décider si un système de contraintes est satisfaisable.

Confidentialité via la résolution de contraintes

Étant donné une suite finie d'actions,

$$\mathsf{in}(u_1).\ \mathsf{out}(v_1).\ \mathsf{in}(u_2).\ \ldots.$$

$\longrightarrow$ u_i, v_i peuvent contenir des variables

on construit un système de contraintes :

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

Confidentialité via la résolution de contraintes

Étant donné une suite finie d'actions,

$$\text{in}(u_1).\ \text{out}(v_1).\ \text{in}(u_2).\ \ldots.$$

$\longrightarrow$ u_i, v_i peuvent contenir des variables

on construit un système de contraintes :

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

On appelle solution d'un système $\mathcal{C}$ une substitution σ telle que :

pour tout $T \overset{?}{\vdash} u$ dans $\mathcal{C}$, on a $u\sigma$ dérivable de $T\sigma$.

Retour sur le protocole de Denning Sacco

$$A \to B \quad : \quad \mathsf{aenc}(\mathsf{sign}(k_{AB}, \mathsf{priv}(A)), \mathsf{pub}(B))$$

Une suite d'actions à considérer : (il y en a plein d'autres)

$\mathsf{out}(\mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)))$; $\mathsf{in}(\mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b)))$.

Retour sur le protocole de Denning Sacco

$$A \to B \quad : \quad \mathsf{aenc}(\mathsf{sign}(k_{AB}, \mathsf{priv}(A)), \mathsf{pub}(B))$$

Une suite d'actions à considérer : (il y en a plein d'autres)

$\mathsf{out}(\mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)))$; $\mathsf{in}(\mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b)))$.

Un système de contraintes :

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} \mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b))$$

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} x$$

avec $T_0 = \{\mathsf{pub}(a),\ \mathsf{pub}(b);\ \mathsf{pub}(c);\ \mathsf{priv}(c)\}$.

Retour sur le protocole de Denning Sacco

$$A \to B \quad : \quad \mathsf{aenc}(\mathsf{sign}(k_{AB}, \mathsf{priv}(A)), \mathsf{pub}(B))$$

Une suite d'actions à considérer : (il y en a plein d'autres)

$\mathsf{out}(\mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)))$; $\mathsf{in}(\mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b)))$.

Un système de contraintes :

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} \mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b))$$

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} x$$

avec $T_0 = \{\mathsf{pub}(a),\ \mathsf{pub}(b);\ \mathsf{pub}(c);\ \mathsf{priv}(c)\}$.

Est-ce que ce système admet une solution ?

Retour sur le protocole de Denning Sacco

$$A \to B \quad : \quad \mathsf{aenc}(\mathsf{sign}(k_{AB}, \mathsf{priv}(A)), \mathsf{pub}(B))$$

Une suite d'actions à considérer : (il y en a plein d'autres)

$\mathsf{out}(\mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)))$; $\mathsf{in}(\mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b)))$.

Un système de contraintes :

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} \mathsf{aenc}(\mathsf{sign}(x, \mathsf{priv}(a)), \mathsf{pub}(b))$$

$$T_0;\ \mathsf{aenc}(\mathsf{sign}(k_{ac}, \mathsf{priv}(a)), \mathsf{pub}(c)) \overset{?}{\vdash} x$$

avec $T_0 = \{\mathsf{pub}(a),\ \mathsf{pub}(b);\ \mathsf{pub}(c);\ \mathsf{priv}(c)\}$.

Est-ce que ce système admet une solution ? Oui $x \to k_{ac}$

Le cas général

$\longrightarrow$ simplifier le système jusqu'à l'obtention de systèmes dont la satisfaisabilité est trivialement décidable.

Le cas général

$\longrightarrow$ simplifier le système jusqu'à l'obtention de systèmes dont la satisfaisabilité est trivialement décidable.

Qu'est-ce qu'une forme résolue ?

$$\mathcal{C} = \begin{cases} T_0 \stackrel{?}{\vdash} x_0 \\ T_0 \cup T_1 \stackrel{?}{\vdash} x_1 \\ \quad \ldots \\ T_0 \cup T_1 \ldots \cup T_n \stackrel{?}{\vdash} x_n \end{cases}$$

Un tel système a des solutions !

$\longrightarrow$ prendre $\sigma = \{x_0 \mapsto u_0, \ldots, x_n \mapsto u_0\}$ avec $u_0 \in T_0$.

Quelques règles de simplification

Décomposition

$$\mathcal{C} = \left\{ \begin{array}{c} \dots \\ T \overset{?}{\vdash} \mathsf{aenc}(u, \mathsf{pub}(v)) \\ \dots \end{array} \right. \quad \rightsquigarrow \quad \begin{array}{c} \dots \\ T \overset{?}{\vdash} u \\ T \overset{?}{\vdash} \mathsf{pub}(v) \\ \dots \end{array}$$

Quelques règles de simplification

Décomposition

$$\mathcal{C} = \left\{ \begin{array}{c} \dots \\ T \overset{?}{\vdash} \mathsf{aenc}(u, \mathsf{pub}(v)) \\ \dots \end{array} \right. \quad \rightsquigarrow \quad \begin{array}{c} \dots \\ T \overset{?}{\vdash} u \\ T \overset{?}{\vdash} \mathsf{pub}(v) \\ \dots \end{array}$$

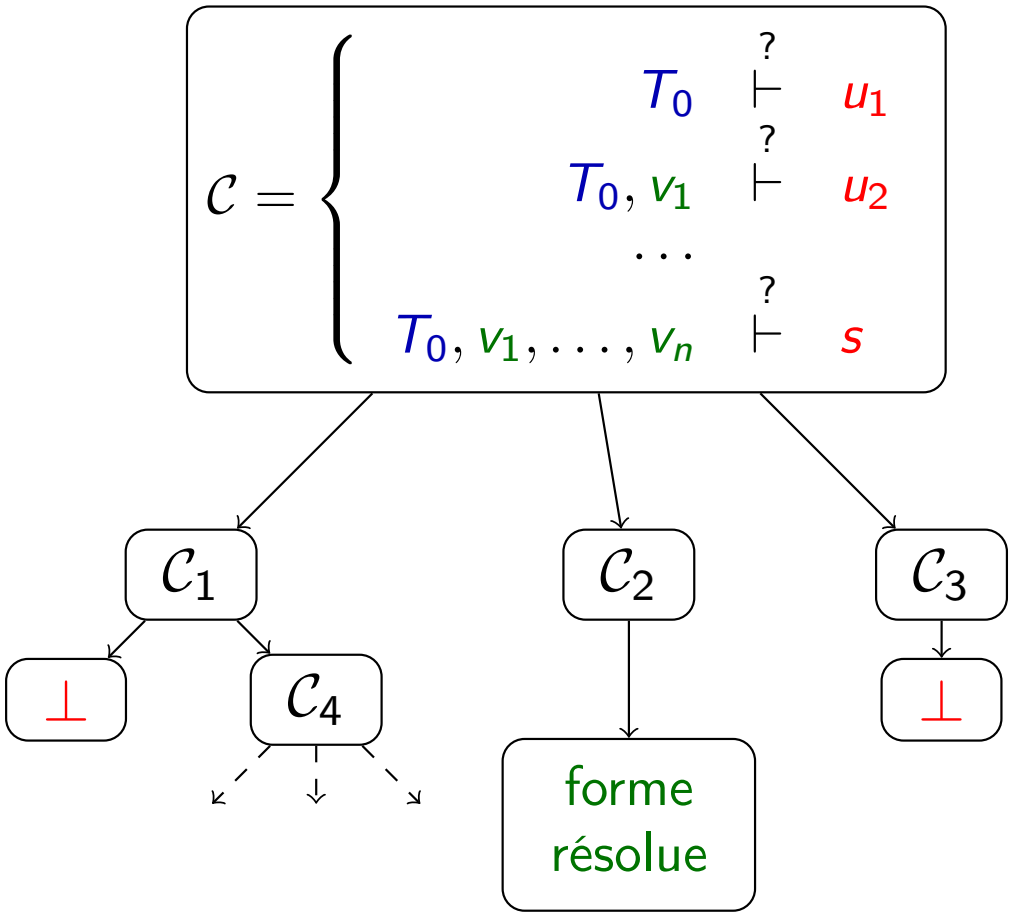
Unification

$$\mathcal{C} = \left\{ \begin{array}{c} \mathcal{C}' \\ T \overset{?}{\vdash} \mathsf{aenc}(u, \mathsf{pub}(v)) \end{array} \right. \quad \rightsquigarrow \quad \mathcal{C}'\sigma$$

avec σ unificateur (le plus général) entre $\mathsf{aenc}(u, \mathsf{pub}(v))$ et $t \in T$.

La procédure de résolution de contraintes

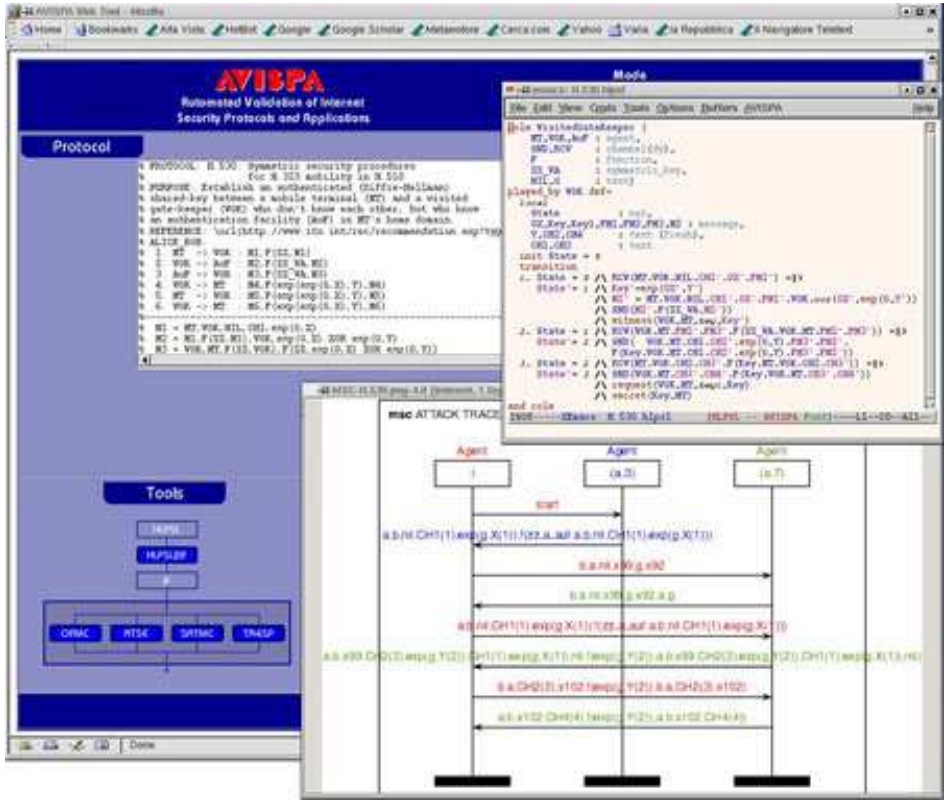
Millen & Shmatikov, 2001 ; Comon-Lundh et al, 2010



$\longrightarrow$ on obtient ainsi une représentation finie de toutes les solutions.

Outil de vérification OFMC

$\longrightarrow$ pour un nombre borné de sessions



Basin et al., 2005

Partie III

Retour sur le passeport

Passeport électronique

Un passeport électronique est un passeport contenant une puce RFID.

$\longrightarrow$ ils sont délivrés en France depuis l'été 2006.

Passeport électronique

Un passeport électronique est un passeport contenant une puce RFID.

$\longrightarrow$ ils sont délivrés en France depuis l'été 2006.

La puce RFID permet de stocker :

- les informations écrites sur le passeport,
- votre photo numérisée.

Passeport électronique

Un passeport électronique est un passeport contenant une puce RFID.

$\longrightarrow$ ils sont délivrés en France depuis l'été 2006.

La puce RFID permet de stocker :

- les informations écrites sur le passeport,
- votre photo numérisée.

Il est interrogeable à distance à l'insu de son propriétaire !

Passeport électronique

1ère génération
Aucun mécanisme de sécurité pour protéger
les informations personnelles

$\longrightarrow$ passeports émis entre 2004 et 2006 en Belgique

Passeport électronique

1ère génération
Aucun mécanisme de sécurité pour protéger les informations personnelles

$\longrightarrow$ passeports émis entre 2004 et 2006 en Belgique

2ème génération
Mise en place d'un mécanisme (BAC) pour protéger nos informations personnelles

$\longrightarrow$ passeports émis à partir de 2006 en France, en Belgique, ...

Et le respect de la vie privée dans tout ça ?

ISO/IEC standard 15408 : Un utilisateur doit pouvoir utiliser plusieurs fois un service ou une ressource sans permettre à un tiers de faire un lien entre ces différentes utilisations.

Et le respect de la vie privée dans tout ça ?

ISO/IEC standard 15408 : Un utilisateur doit pouvoir utiliser plusieurs fois un service ou une ressource sans permettre à un tiers de faire un lien entre ces différentes utilisations.

Et le respect de la vie privée dans tout ça ?

ISO/IEC standard 15408 : Un utilisateur doit pouvoir utiliser plusieurs fois un service ou une ressource sans permettre à un tiers de faire un lien entre ces différentes utilisations.

L'adversaire peut-il faire la différence entre :

1. une situation où un même passeport est utilisé plusieurs fois ;
2. une situation où chaque passeport est utilisé au plus une fois.

Arapinis et al., 2010

Équivalence via la résolution de contraintes

Étape 1 :

Une exploration symbolique de toutes les traces possibles ;

L'infinité d'exécutions possible est représenté par un nombre fini de systèmes de contraintes.

$\longrightarrow$ cet ensemble peut s'avérer grand

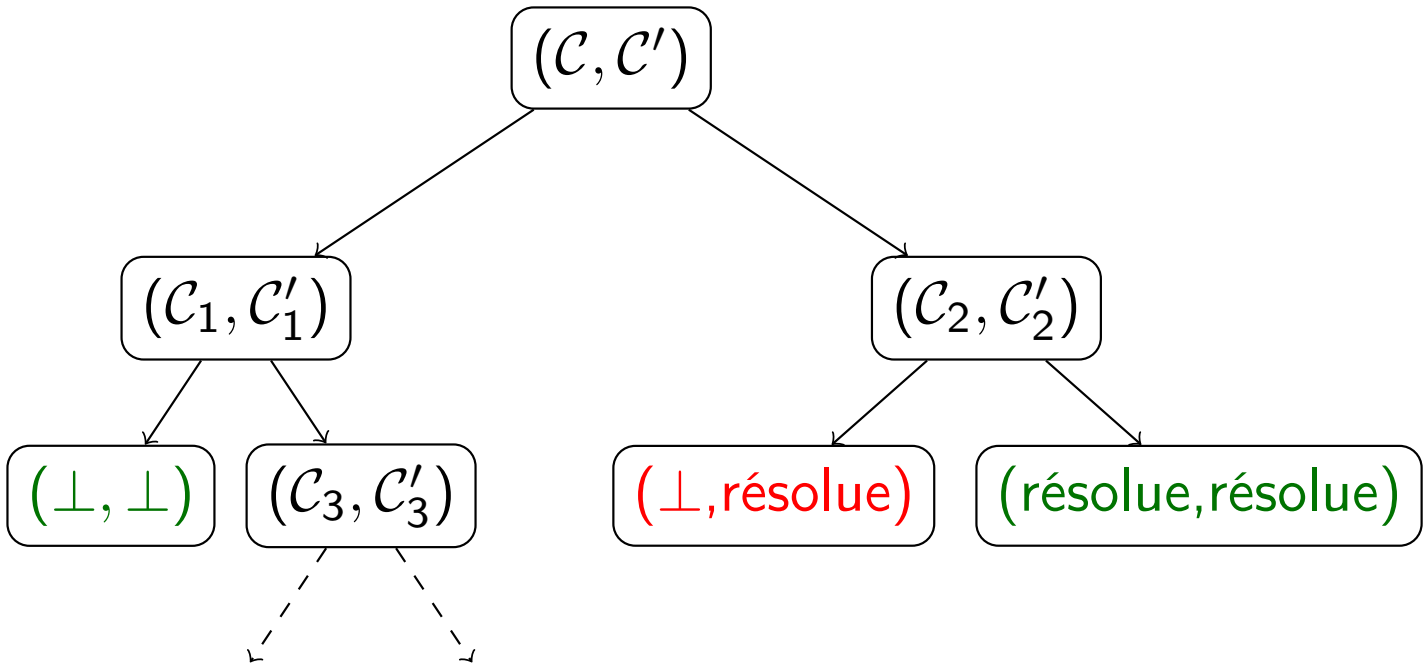
Étape 2 :

Une procédure pour décider si deux systèmes de contraintes sont équivalents (*i.e.* même ensemble de solutions).

Baudet 2005, Cheval et al 2010, ...

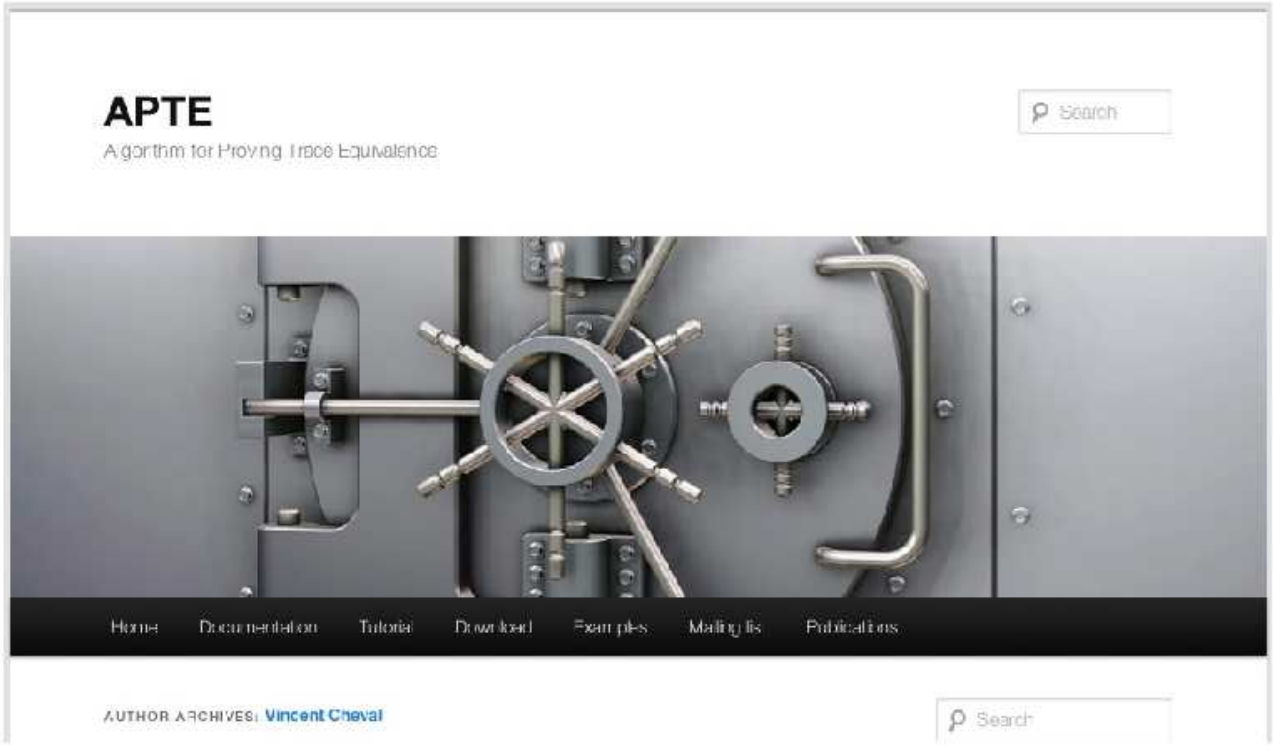
La procédure vue de loin

$\longrightarrow$ simplifier le système jusqu'à l'obtention de systèmes pour lesquels l'équivalence est trivialement décidable.



Outil APTE

$\longrightarrow$ développé par Vincent Cheval



Le passeport français est traçable !

L'attaquant écoute une session honnête entre Alice et le lecteur ...

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

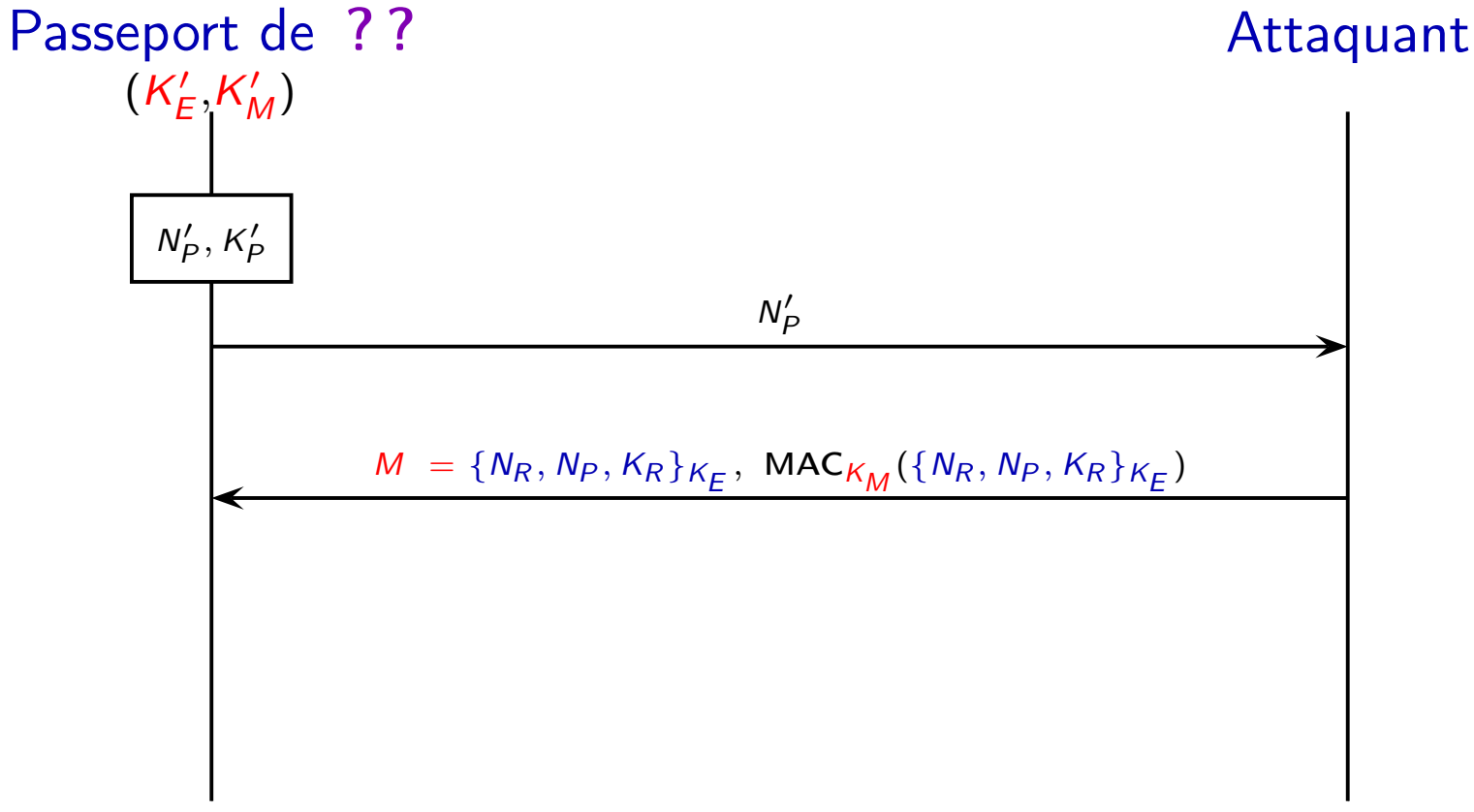
... puis rejoue M et analyse le message d'erreur obtenu.

Le passeport français est traçable !

L'attaquant écoute une session honnête entre Alice et le lecteur . . .

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

. . . puis rejoue M et analyse le message d'erreur obtenu.

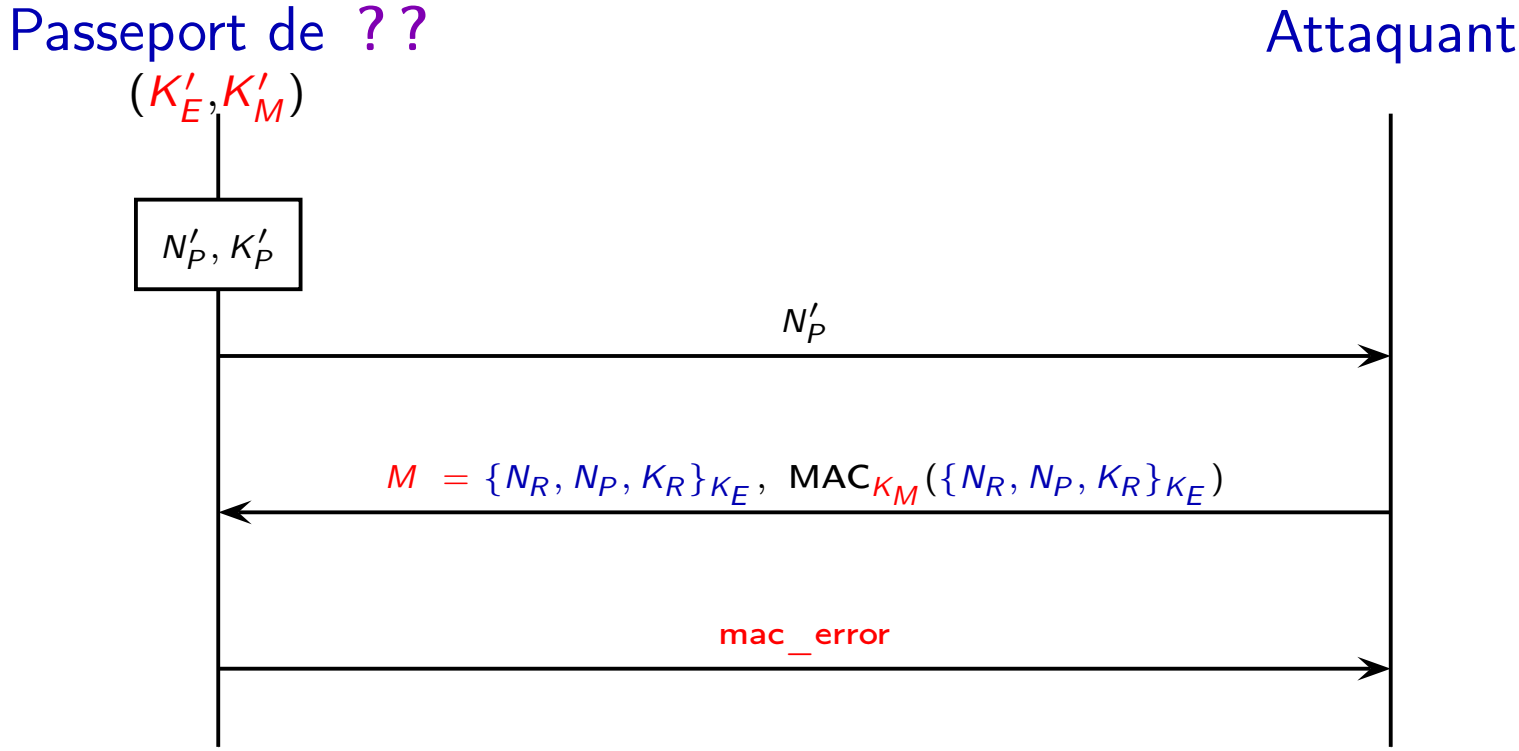


Le passeport français est traçable !

L'attaquant écoute une session honnête entre Alice et le lecteur . . .

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

. . . puis rejoue M et analyse le message d'erreur obtenu.



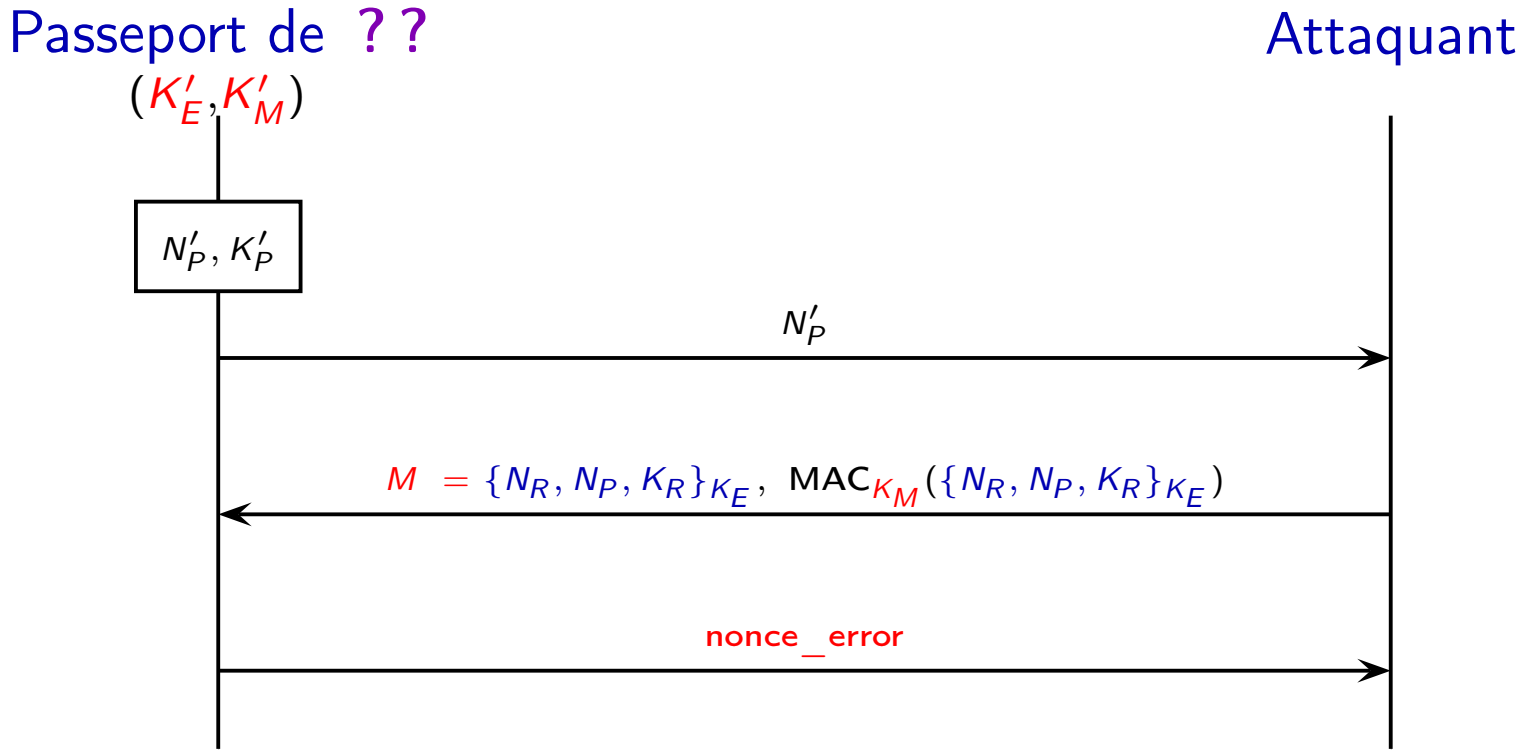
$$\Longrightarrow \quad K'_M \neq K_M \quad \Longrightarrow \quad ?? \text{ n'est pas Alice}$$

Le passeport français est traçable !

L'attaquant écoute une session honnête entre Alice et le lecteur . . .

$$M = \{N_R, N_P, K_R\}_{K_E},\ \mathrm{MAC}_{K_M}(\{N_R, N_P, K_R\}_{K_E})$$

. . . puis rejoue M et analyse le message d'erreur obtenu.



$$\Longrightarrow \quad K'_M = K_M \quad \Longrightarrow \quad ?? \text{ est Alice}$$

Conclusion

À retenir

Les protocoles cryptographiques sont :

- difficiles à concevoir et à analyser ;
- vulnérables aux attaques logiques.

Des primitives robustes, c'est bien . . .

. . . **mais ce n'est pas suffisant !**

À retenir

Les protocoles cryptographiques sont :

- difficiles à concevoir et à analyser ;
- vulnérables aux attaques logiques.

Il est important de s'assurer du bon fonctionnement de ces protocoles.

Ce que l'on sait faire :

- les propriétés de sécurité les plus classiques ;
- l'analyse de protocoles plutôt petits ;
- les primitives cryptographiques standard.

De nombreuses pistes à explorer

Au vu des applications qui voient le jour, **ce n'est pas suffisant !**

- nouveaux objectifs de sécurité
  $\longrightarrow$ anonymat, non traçabilité, proximité physique, ...
- propriétés algébriques
  $\longrightarrow$ chiffrement homomorphe, ou exclusif, ...
- passage à l'échelle
  $\longrightarrow$ une même application est généralement composée de plusieurs protocoles

Questions ?