Ces protocoles qui nous protègent ...

Stéphanie Delaune

Chargée de Recherche CNRS affectée au LSV
Équipe-Projet SecSI

ENS Cachan

- 17 départements d'enseignement: mathématiques, informatique, chimie, génie mécanique, sciences sociales, ...
- 14 laboratoires de recherche:

Laboratoire Spécification & Vérification

L'enseignement en informatique à l'ENS Cachan

$\longrightarrow$ formation tournée vers les **fondements de l'informatique**

L'enseignement en informatique à l'ENS Cachan

$\longrightarrow$ formation tournée vers les **fondements de l'informatique**

- Les bases: calculabilité et logique, algorithmique, complexité, langages formels, programmation, ...
- Des enseignements plus poussés: preuves assistées par ordinateur, démonstration automatique, bioinformatique, ...

L'enseignement en informatique à l'ENS Cachan

$\longrightarrow$ formation tournée vers les **fondements de l'informatique**

- Les bases: calculabilité et logique, algorithmique, complexité, langages formels, programmation, ...
- Des enseignements plus poussés: preuves assistées par ordinateur, démonstration automatique, bioinformatique, ...

« La science informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des téléscopes »

E. Dijkstra

La recherche au LSV

Ennemi public numéro 1: le bug !

La recherche au LSV

Ennemi public numéro 1: le bug !

$\longrightarrow$ accroître notre confiance dans les logiciels critiques

La recherche au LSV

Ennemi public numéro 1: le bug !

$\longrightarrow$ accroître notre confiance dans les logiciels critiques

- logiciel: texte relativement long écrit dans un langage spécifique et qui sera exécuté par un ordinateur
- critique: une défaillance peut avoir des conséquences désastreuses en termes humains ou économiques

Dans la vie quotidienne ...

Zune : le bug de l'an 2009 !

Il redémarre tout seul puis reste figé sur l'écran de démarrage.

Dans la vie quotidienne ... ou presque !

Sonde Mars Climate Orbiter - 26 septembre 1999

Perte de la sonde due ...

Dans la vie quotidienne ... ou presque !

Sonde Mars Climate Orbiter - 26 septembre 1999

Perte de la sonde due ... à un problème d'unité de mesure !

Équipe projet SECSI

Sécurité des Systèmes d'Information

- 4 permanents: H. Comon-Lundh, S. Delaune, J. Goubault-Larrecq, et G. Steel.




- 6 membres temporaires
- 4 doctorants

Protocoles cryptographiques

PayPal™

- petits programmes destinés à sécuriser nos communications (*e.g.* confidentialité, authentification)
- omniprésents dans notre vie quotidienne.

Protocoles cryptographiques

PayPal™

- petits programmes destinés à sécuriser nos communications (*e.g.* confidentialité, authentification)
- omniprésents dans notre vie quotidienne.

Protocoles cryptographiques

PayPal™

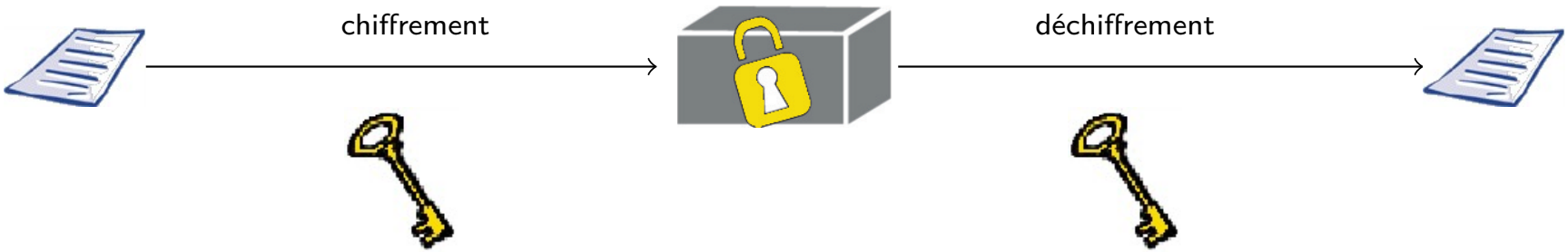
- petits programmes destinés à sécuriser nos communications (*e.g.* confidentialité, authentification)
- omniprésents dans notre vie quotidienne.

Nos informations personnelles sont en danger !

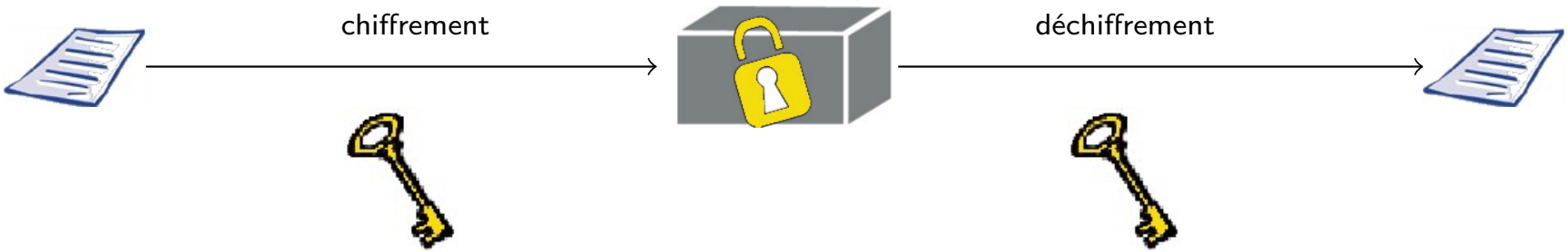
Le chiffrement

« KDXWH QRUPD QGLH »

Chiffrement symétrique



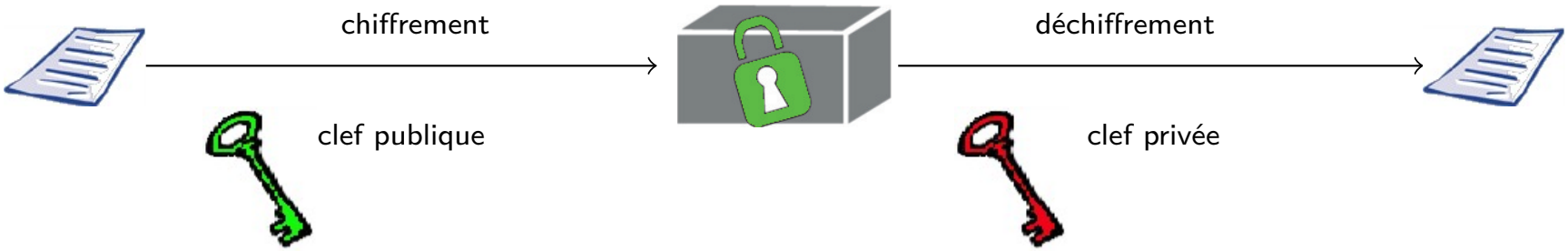
Chiffrement symétrique



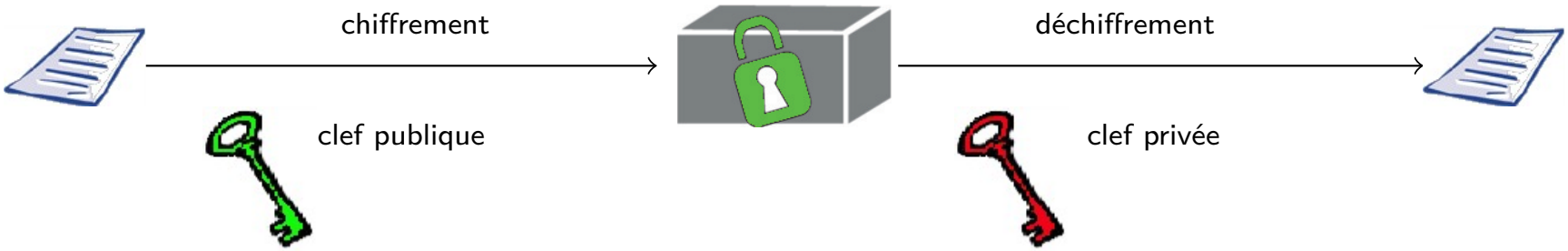
Quelques dates repères:

- 2000 avant J.-C.: traces de son utilisation par les Égyptiens
- 1920: machine Enigma
- 1977: Data Encryption Standard (DES)
- 2000: Advanced Encryption Standard (AES)

Chiffrement asymétrique (ou à clefs publiques)



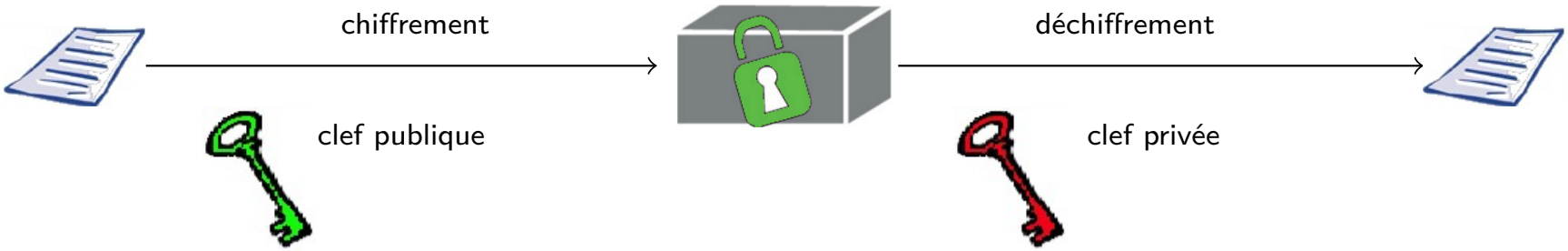
Chiffrement asymétrique (ou à clefs publiques)



1977: chiffrement RSA (encore utilisé à l'heure actuelle)

- cette méthode de chiffrement repose sur un problème mathématique bien connu: le problème de la factorisation.

Chiffrement asymétrique (ou à clefs publiques)



1977: chiffrement RSA (encore utilisé à l'heure actuelle)

- cette méthode de chiffrement repose sur un problème mathématique bien connu: le problème de la factorisation.

Tant que nous ne sommes pas capables de résoudre ce problème d'une façon efficace, la méthode de chiffrement RSA sera considérée sûre.

Mais chiffrer ne suffit pas toujours !

La carte bancaire

Le vote électronique

Le passeport électronique

Le protocole de paiement

- Le client Cl insère sa carte C dans le terminal T.
- Le marchand saisit le montant M de la transaction.
- Le terminal vérifie qu'il s'agit d'une « vraie carte ».
- Le client entre son code.
  Si $M \geq €100$, alors dans 20% des cas,
  - Le terminal contacte la banque B.
  - La banque donne (ou pas) son autorisation.

En détails (1/2)

4 acteurs: la Banque B , le Client Cl, la Carte C et le Terminal T

En détails (1/2)

4 acteurs: la Banque B , le Client Cl, la Carte C et le Terminal T

La **Banque possède**

- une clef privée – priv(B)
- une clef publique – pub(B)
- une clef symétrique secrète partagée avec la carte – K_{CB}

En détails (1/2)

4 acteurs: la Banque B , le Client Cl, la Carte C et le Terminal T

La **Banque possède**

- une clef privée – $\mathsf{priv}(B)$
- une clef publique – $\mathsf{pub}(B)$
- une clef symétrique secrète partagée avec la carte – K_{CB}

La **Carte** possède

- des données $Data$: nom du propriétaire, date d'expiration, ...
- la signature de ces données – $\{Data\}_{\mathsf{priv}(B)}$
- la clef K_{CB}, clef secrète partagée avec la banque.

En détails (1/2)

4 acteurs: la Banque B , le Client Cl, la Carte C et le Terminal T

La **Banque possède**

- une clef privée – $\mathsf{priv}(B)$
- une clef publique – $\mathsf{pub}(B)$
- une clef symétrique secrète partagée avec la carte – K_{CB}

La **Carte** possède

- des données $Data$: nom du propriétaire, date d'expiration, ...
- la signature de ces données – $\{Data\}_{\mathsf{priv}(B)}$
- la clef K_{CB}, clef secrète partagée avec la banque.

Le **Terminal possède**

- la clef publique de la banque – $\mathsf{pub}(B)$

En détails (2/2)

Le terminal T lit la carte C:

1. $C \rightarrow T : Data, \{Data\}_{priv(B)}$

En détails (2/2)

Le terminal T lit la carte C:

1. $C \rightarrow T$: $Data, \{Data\}_{priv(B)}$

Le terminal T demande le code:

2. $T \rightarrow CI$: $code?$
3. $CI \rightarrow C$: 1234
4. $C \rightarrow T$: code bon

En détails (2/2)

Le terminal T lit la carte C:

1. $C \rightarrow T$: $Data, \{Data\}_{priv(B)}$

Le terminal T demande le code:

2. $T \rightarrow CI$: *code?*
3. $CI \rightarrow C$: 1234
4. $C \rightarrow T$: code bon

Le terminal T demande l'autorisation à la banque B:

5. $T \rightarrow B$: *autorisation?*
6. $B \rightarrow T$: 45289
7. $T \rightarrow C$: 45289
8. $C \rightarrow T$: $\{45289\}_{K_{CB}}$
9. $T \rightarrow B$: $\{45289\}_{K_{CB}}$
10. $B \rightarrow T$: *ok*

Attaques sur la carte bleue

Initialement la sécurité été assurée par :

- cartes difficilement réplicables,
- secret des clefs et du protocole.

Attaques sur la carte bleue

Initialement la sécurité été assurée par :

- cartes difficilement réplicables,
- secret des clefs et du protocole.

Mais il y a des failles !

- le chiffrement n'est pas sûr (les clefs de 320 bits ne sont plus sûres);
- on peut faire des fausses cartes.

$\rightarrow$ "YesCard" fabriquées par Serge Humpich (1997).

La « YesCard »: Comment ca marche ?

Faille logique

$$
\begin{aligned}
1.C &\rightarrow T &&: \text{Data}, \{\text{Data}\}_{\text{priv}(B)} \\
2.T &\rightarrow CI &&: code? \\
3.CI &\rightarrow C &&: 1234 \\
4.C &\rightarrow T &&: ok
\end{aligned}
$$

La « YesCard »: Comment ca marche ?

Faille logique

$$
\begin{aligned}
1.C &\rightarrow T &&: \text{Data}, \{\text{Data}\}_{\text{priv}(B)} \\
2.T &\rightarrow CI &&: code? \\
3.CI &\rightarrow C &&: \star\ \star\ \star\star \\
4.C &\rightarrow T &&: ok
\end{aligned}
$$

La « YesCard »: Comment ca marche ?

Faille logique

$$
\begin{aligned}
1. C &\rightarrow T &&: \text{Data}, \{\text{Data}\}_{\text{priv}(B)} \\
2. T &\rightarrow CI &&: code? \\
3. CI &\rightarrow C &&: \star\ \star\ \star\star \\
4. C &\rightarrow T &&: ok
\end{aligned}
$$

Ajout d'une fausse signature sur une fausse carte

$$
\begin{aligned}
1. C &\rightarrow T &&: \text{XXX}, \{\text{XXX}\}_{\text{priv}(B)} \\
2. T &\rightarrow CI &&: code? \\
3. CI &\rightarrow C &&: \star\ \star\ \star\ \star \\
4. C &\rightarrow T &&: ok
\end{aligned}
$$

Corrections apportées

Utilisation de plus grands nombres pour sécuriser les cartes bancaires

$\longrightarrow$ 232 chiffres au lieu des 96 chiffres utilisés en 1997.

Corrections apportées

Utilisation de plus grands nombres pour sécuriser les cartes bancaires

$\longrightarrow$ 232 chiffres au lieu des 96 chiffres utilisés en 1997.

Europay, MasterCard et Visa (EMV) ont produit 3 nouveaux protocoles:

1. SDA: Static Data Authentication
   $\longrightarrow$ Ce système est le plus couramment utilisé.
2. DDA: Dynamic Data Authentication
   $\longrightarrow$ Ce système devrait permettre de lutter efficacement contre l'utilisation des "YesCards".
3. CDA: Combined Data Authentication

$\longrightarrow$ conçus en 2004 et déployés en 2006.

Le passeport électronique

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

Passeport électronique 1ère Génération

Aucun mécanisme de sécurité pour protéger les informations personnelles

$\longrightarrow$ possibilité de récupérer la signature manuscrite du porteur en interrogeant le passeport à distance

Passeport électronique — 1ère Génération

Aucun mécanisme de sécurité pour protéger les informations personnelles

$\longrightarrow$ possibilité de récupérer la signature manuscrite du porteur en interrogeant le passeport à distance

"Faille" découverte sur les passeports belges

Passeport émis entre 2004 et 2006 en Belgique

Passeport électronique

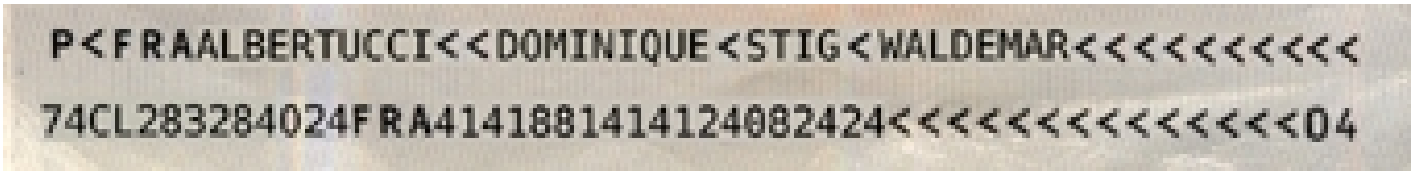
2ème Génération

Passeport émis à partir de 2006 en France, en Belgique, . . .

Passeport électronique — 2ème Génération

Passeport émis à partir de 2006 en France, en Belgique, . . .

⟶ mis en place d'un mécanisme (protocole BAC) qui permet de protéger nos informations personnelles



Avec ces deux lignes, on peut interroger la puce électronique.

Protocole BAC - objectif

Le Basic Access Control (BAC) protocole est un protocole d'établissement de clef qui doit assurer la protection de nos données personnelles ansi que la non traçabilité du passeport.

Protocole BAC - objectif

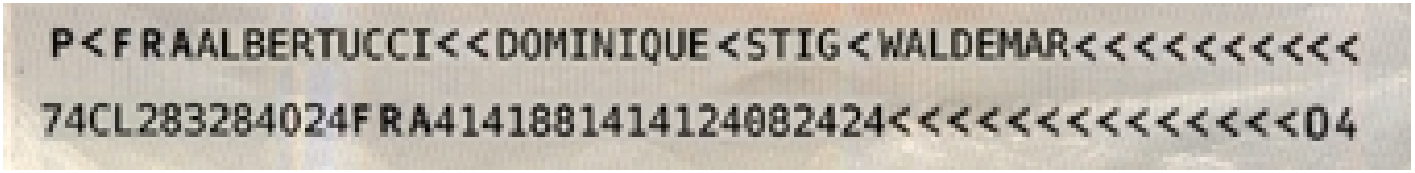
Le Basic Access Control (BAC) protocole est un protocole d'établissement de clef qui doit assurer la protection de nos données personnelles ansi que la non traçabilité du passeport.

ISO/IEC standard 15408

La non traçabilité a pour but d'assurer qu'un utilisateur peut utiliser plusieurs fois un service ou une ressource sans permettre à un tiers de faire un lien entre ces différentes utilisations.

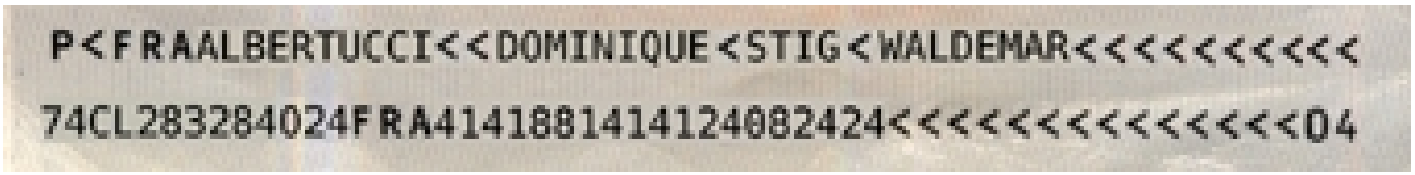
Protocole BAC - étape par étape

1. le lecteur dérive deux clefs K_E et K_M par lecture des lignes codées



Protocole BAC - étape par étape

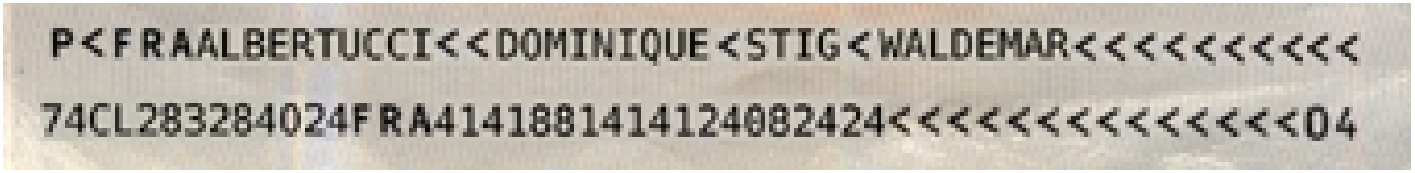
1. le lecteur dérive deux clefs K_E et K_M par lecture des lignes codées



2. Ensuite le passeport et le lecteur s'envoient des messages chiffrés pour établir une nouvelle clef K (utilisable qu'une seule fois);

Protocole BAC - étape par étape

1. le lecteur dérive deux clefs K_E et K_M par lecture des lignes codées



2. Ensuite le passeport et le lecteur s'envoient des messages chiffrés pour établir une nouvelle clef K (utilisable qu'une seule fois);

3. Ensuite, nos données confidentielles sont transmises chiffrées avec cette clef K.

Protocole BAC

Que faire si le protocole ne se déroule pas comme prévu ?

Protocole BAC

Que faire si le protocole ne se déroule pas comme prévu ?

Dans la description du protocole:

- il est mentionné que le passeport doit répondre à tous les messages qu'il reçoit (éventuellement avec un message d'erreur) mais ...

Protocole BAC

Que faire si le protocole ne se déroule pas comme prévu ?

Dans la description du protocole:

- il est mentionné que le passeport doit répondre à tous les messages qu'il reçoit (éventuellement avec un message d'erreur) mais ...
- ... ces messages d'erreurs ne sont pas précisés.

$\longrightarrow$ Il en résulte une implémentation différentes selon les nations

...

Protocole BAC

Que faire si le protocole ne se déroule pas comme prévu ?

Dans la description du protocole:

- il est mentionné que le passeport doit répondre à tous les messages qu'il reçoit (éventuellement avec un message d'erreur) mais ...
- ... ces messages d'erreurs ne sont pas précisés.

$\longrightarrow$ Il en résulte une implémentation différentes selon les nations

... et une **attaque sur le passeport français**.

passer les détails

Attaque sur le passeport Français

Étape 1: l'attaquant intercepte les messages échangés au cours d'une exécution du protocole avec le passeport d'Alice:

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

Attaque sur le passeport Français

Étape 1: l'attaquant intercepte les messages échangés au cours d'une exécution du protocole avec le passeport d'Alice:

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

Étape 2: l'attaquant rejoue ce message ultérieurement, le passeport en présence répondra:

1. soit error 6300: "échec lors la vérification du MAC"
2. soit error 6A80: "échec lors de la vérification du nonce".

Attaque sur le passeport Français

Étape 1: l'attaquant intercepte les messages échangés au cours d'une exécution du protocole avec le passeport d'Alice:

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

Étape 2: l'attaquant rejoue ce message ultérieurement, le passeport en présence répondra:

1. soit error 6300: "échec lors la vérification du MAC"
2. soit error 6A80: "échec lors de la vérification du nonce".

$\longrightarrow$ le message error 6A80 indique qu'il s'agit du passeport Alice.

Attaque sur le passeport Français

Étape 1: l'attaquant intercepte les messages échangés au cours d'une exécution du protocole avec le passeport d'Alice:

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

Étape 2: l'attaquant rejoue ce message ultérieurement, le passeport en présence répondra:

1. soit error 6300: "échec lors la vérification du MAC"
2. soit error 6A80: "échec lors de la vérification du nonce".

$\longrightarrow$ le message error 6A80 indique qu'il s'agit du passeport Alice.

Attaque découverte en 2010 par T. Chothia et V. Smirnov

Retour

Passeport électronique 3ème Génération

SAC : Passeport 3e génération

Une nouvelle dimension dans la sécurité du passeport électronique

Je cite: « Ce mécanisme offre des propriétés de sécurité supérieures à celle du mécanisme BAC et garantit ainsi une très haute protection de l'anonymat du porteur. Il assure les propriétés de: non liable, non transférable, et intraçable. »

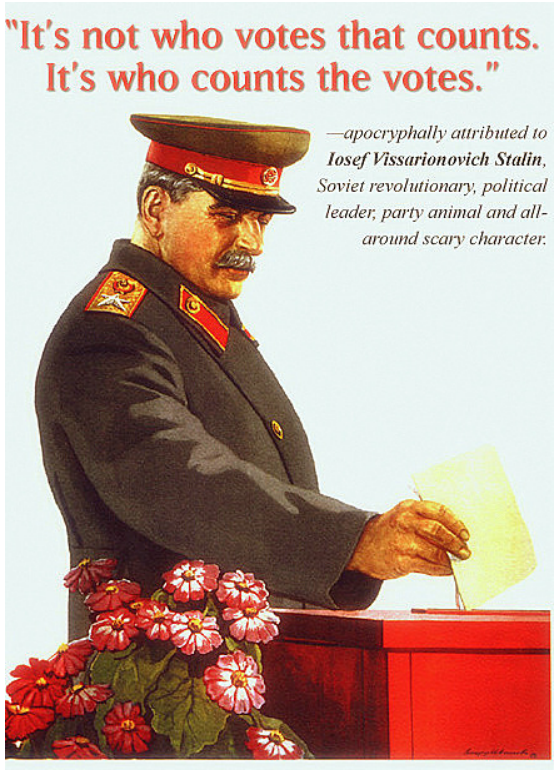
Vote électronique

Vote électronique

La démocratie est-elle en péril ?

Avantages:

- pratique: différents types de scrutins, possibilité de voter de chez soi, ...
- décompte efficace des bulletins.



... mais il est souvent **opaque** et **invérifiable** !

Deux grandes familles

Machines à voter

- utilisation des bureaux de vote / isoloirs;
- mécanisme d'authentification externe (*e.g.* carte d'identité)

Deux grandes familles

Machines à voter

- utilisation des bureaux de vote / isoloirs;
- mécanisme d'authentification externe (*e.g.* carte d'identité)

$\longrightarrow$ utilisées dans 82 communes - élections présidentielles 2007

Deux grandes familles

Machines à voter

- utilisation des bureaux de vote / isoloirs;
- mécanisme d'authentification externe (*e.g.* carte d'identité)

$\longrightarrow$ utilisées dans 82 communes - élections présidentielles 2007

Vote par Internet

- possibilité de voter de chez soi avec son ordinateur personnel;

Deux grandes familles

Machines à voter

- utilisation des bureaux de vote / isoloirs;
- mécanisme d'authentification externe (*e.g.* carte d'identité)

$\longrightarrow$ utilisées dans 82 communes - élections présidentielles 2007

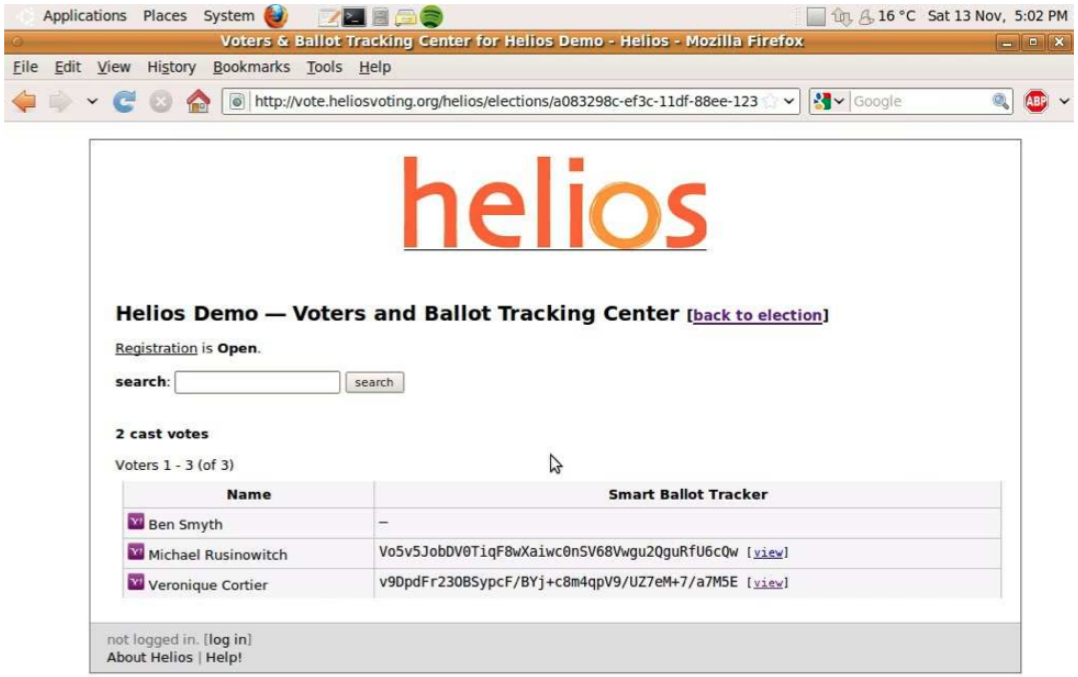
Vote par Internet

- possibilité de voter de chez soi avec son ordinateur personnel;

$\longrightarrow$ 11 députés à l'Assemblée nationale seront élus par voie électronique - élections législatives 2012

Helios

$\longrightarrow$ développé par Ben Adida *et al.*



$\longrightarrow$ utilisé lors de plusieurs élections: à l'UCL, à l'Université de Princeton, ...

Qu'est-ce qu'un bon protocole de vote ?

Vérifiabilité individuelle

Équité

Absence de reçu

Résistance à la coercition

Vérifiabilité universelle

Éligibilité

Anonymat

Qu'est-ce qu'un bon protocole de vote ?

Vérifiabilité individuelle

Équité

Absence de reçu **Résistance à la coercition**

Vérifiabilité universelle

Éligibilité

Anonymat

Est-ce qu'un bon protocole de vote existe ?

Qu'est-ce qu'un bon protocole de vote ?

Vérifiabilité individuelle

Équité

Absence de reçu **Résistance à la coercition**

Vérifiabilité universelle

Éligibilité

Anonymat

Est-ce qu'un bon protocole de vote existe ?

$\longrightarrow$ protocoles souvent complexes, utilisant des mécanismes cryptographiques « exotiques » et ne satisfaisant qu'un sous-ensemble des propriétés de sécurité ci-dessus.

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\mathsf{pub}(S)}$
Bob	$\{v_B\}_{\mathsf{pub}(S)}$
Chris	$\{v_C\}_{\mathsf{pub}(S)}$

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\mathsf{pub}(S)}$
Bob	$\{v_B\}_{\mathsf{pub}(S)}$
Chris	$\{v_C\}_{\mathsf{pub}(S)}$

$\{v_D\}_{\mathsf{pub}(S)}$ ←

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\mathsf{pub}(S)}$
Bob	$\{v_B\}_{\mathsf{pub}(S)}$
Chris	$\{v_C\}_{\mathsf{pub}(S)}$
David	$\{v_D\}_{\mathsf{pub}(S)}$

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\text{pub}(S)}$
Bob	$\{v_B\}_{\text{pub}(S)}$
Chris	$\{v_C\}_{\text{pub}(S)}$
David	$\{v_D\}_{\text{pub}(S)}$

Phase de comptage: utilisation du chiffrement homomorphique

$$\{v_A\}_{\text{pub}(S)} \times \{v_B\}_{\text{pub}(S)} \times \ldots = \{v_A + v_B + \ldots\}_{\text{pub}(S)}$$

$\longrightarrow$ Ainsi seul le résultat final sera déchiffré.

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\text{pub}(S)}$
Bob	$\{v_B\}_{\text{pub}(S)}$
Chris	$\{v_C\}_{\text{pub}(S)}$
David	$\{v_D\}_{\text{pub}(S)}$

Phase de comptage: utilisation du chiffrement homomorphique

$$\{v_A\}_{\text{pub}(S)} \times \{v_B\}_{\text{pub}(S)} \times \ldots = \{v_A + v_B + \ldots\}_{\text{pub}(S)}$$

$\longrightarrow$ Ainsi seul le résultat final sera déchiffré.

Un votant malhonnête pourrait tricher !

Protocole Helios (version simplifiée)

Phase de vote: (valeur 0 ou 1)

Tableau d'affichage	
Alice	$\{v_A\}_{\text{pub}(S)}$
Bob	$\{v_B\}_{\text{pub}(S)}$
Chris	$\{v_C\}_{\text{pub}(S)}$
David	$\{v_D\}_{\text{pub}(S)}$

Phase de comptage: utilisation du chiffrement homomorphique

$$\{v_A\}_{\text{pub}(S)} \times \{v_B\}_{\text{pub}(S)} \times \ldots = \{v_A + v_B + \ldots\}_{\text{pub}(S)}$$

$\longrightarrow$ Ainsi seul le résultat final sera déchiffré.

Un votant malhonnête pourrait tricher !

$\{v_D\}_{\text{pub}(S)}$ " + " preuve que v_D est égal à 0 ou 1

Protocole Helios

Vérifiabilité individuelle et universelle

Helios satisfait a priori les différentes formes de vérifiabilité.

Protocole Helios

Vérifiabilité individuelle et universelle

Helios satisfait a priori les différentes formes de vérifiabilité.

Anonymat, sans reçu, et résistance à la coercition

- Helios ne résiste pas aux formes de coercition les plus fortes
  $\longrightarrow$ il est possible d'obtenir un reçu de son vote

Protocole Helios

Vérifiabilité individuelle et universelle

Helios satisfait a priori les différentes formes de vérifiabilité.

Anonymat, sans reçu, et résistance à la coercition

- Helios ne résiste pas aux formes de coercition les plus fortes
  $\longrightarrow$ il est possible d'obtenir un reçu de son vote
- **Helios ne satisfait même pas l'anonymat !**
  $\longrightarrow$ il est possible de rejouer un message et de voter comme une autre votant de son choix (sans pour autant connaître la valeur de son vote)

Attaque découverte en 2011 par B. Smyth et V. Cortier

Comment vérifier ces protocoles ?

Les mathématiques et l'informatique à la rescousse !

Comment vérifier ces protocoles ?

Les mathématiques et l'informatique à la rescousse !

Notre but:

1. faire des preuves mathématiques rigoureuses,
2. d'une façon automatique.

« Construire une machine à détecter les bugs »

Comment vérifier ces protocoles ?

Les mathématiques et l'informatique à la rescousse !

Notre but:

1. faire des preuves mathématiques rigoureuses,
2. d'une façon automatique.

« Construire une machine à détecter les bugs »

1936: une telle machine n'existe pas (Alan Turing)

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

Mais alors, que faisons-nous ?

Le problème n'a pas de solution ....

Mais alors, que faisons-nous ?

Le problème n'a pas de solution ....

mais seulement dans le cas général

Mais alors, que faisons-nous ?

Le problème n'a pas de solution ....

mais seulement dans le cas général

Différentes pistes:

- résoudre le problème dans de nombreux cas intéressants,
- proposer des procédures approchées,

Mais alors, que faisons-nous ?

Le problème n'a pas de solution ....

mais seulement dans le cas général

Différentes pistes:

- résoudre le problème dans de nombreux cas intéressants,
- proposer des procédures approchées,

$\longrightarrow$ **Prix Turing** décerné à Joseph Sifakis en 2007 pour ses travaux sur le model-checking.

▸ passer les détails

La logique (du premier ordre) à la rescousse !

Un peu de vocabulaire:

- les termes représentent les objets, e.g. $\{s\}_{\mathsf{pub}(A)}$;
- les prédicats représentent les relations entre ces objets, e.g. $\mathcal{I}(\_)$;
- les atomes représentent les faits qui sont vrais, e.g. $\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$.

La logique (du premier ordre) à la rescousse !

Un peu de vocabulaire:

- les termes représentent les objets, e.g. $\{s\}_{\mathsf{pub}(A)}$;
- les prédicats représentent les relations entre ces objets, e.g. $\mathcal{I}(\_)$;
- les atomes représentent les faits qui sont vrais, e.g. $\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$.

Clauses de Horn:

$$\forall x_1 \cdot \ldots \forall x_n \cdot \; A_1; \ldots A_n \Rightarrow B$$

La logique (du premier ordre) à la rescousse !

Un peu de vocabulaire:

- les termes représentent les objets, e.g. $\{s\}_{\mathsf{pub}(A)}$;
- les prédicats représentent les relations entre ces objets, e.g. $\mathcal{I}(\_)$;
- les atomes représentent les faits qui sont vrais, e.g. $\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$.

Clauses de Horn:

$$\forall x_1 \cdot \ldots \forall x_n \cdot \; A_1; \ldots A_n \Rightarrow B$$

Exemple (clauses de Horn pour l'attaquant $\mathcal{C}_A$):

$$\forall x \cdot \forall y \cdot \quad \mathcal{I}(x);\, \mathcal{I}(\mathsf{pub}(y)) \quad \Rightarrow \quad \mathcal{I}(\{x\}_{\mathsf{pub}(y)})$$

$$\forall x \cdot \forall y \cdot \quad \mathcal{I}(\{x\}_{\mathsf{pub}(y)});\, \mathcal{I}(\mathsf{priv}(y)) \quad \Rightarrow \quad \mathcal{I}(x)$$

Un exemple pour illustrer

$$\xleftarrow{\{secret\}_{pub(A)}}$$

$$\xrightarrow{\{\{secret\}_{pub(A)}\}_{pub(B)}}$$

$$\xleftarrow{\{secret\}_{pub(B)}}$$

$$\longrightarrow \{\{secret\}_{pub(A)}\}_{pub(B)} = \{\{secret\}_{pub(B)}\}_{pub(A)}.$$

Un exemple pour illustrer

$\leftarrow \{secret\}_{pub(A)}$

$\{\{secret\}_{pub(A)}\}_{pub(B)} \rightarrow$

$\leftarrow \{secret\}_{pub(B)}$

Ce protocole ne marche pas! (problème d'authentification)

Un exemple pour illustrer

$\leftarrow \{secret\}_{pub(A)}$

$\{\{secret\}_{pub(A)}\}_{pub(B)} \rightarrow$

$\leftarrow \{secret\}_{pub(B)}$

Ce protocole ne marche pas! (problème d'authentification)

$\leftarrow \{secret\}_{pub(A)}$

$\{\{secret\}_{pub(A)}\}_{pub(I)} \rightarrow$

$\leftarrow \{secret\}_{pub(I)}$

Clauses de Horn pour le protocole

Protocole

$$
\begin{array}{lcl}
A & \rightarrow & B : \{s\}_{\mathsf{pub}(A)} \\
B & \rightarrow & A : \{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)} \\
A & \rightarrow & B : \{s\}_{\mathsf{pub}(B)}
\end{array}
$$

Clauses de Horn $\mathcal{C}_P$

$$
\begin{array}{rcl}
 & \Rightarrow & \mathcal{I}(\{s\}_{\mathsf{pub}(A)}) \\
\forall x \cdot \mathcal{I}(x) & \Rightarrow & \mathcal{I}(\{x\}_{\mathsf{pub}(B)}) \\
\forall x \cdot \mathcal{I}(\{x\}_{\mathsf{pub}(A)}) & \Rightarrow & \mathcal{I}(x)
\end{array}
$$

Clauses de Horn pour le protocole

Protocole			Clauses de Horn $\mathcal{C}_P$		
A	$\rightarrow$	$B : \{s\}_{\mathsf{pub}(A)}$		$\Rightarrow$	$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$
B	$\rightarrow$	$A : \{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)}$	$\forall x \cdot \mathcal{I}(x)$	$\Rightarrow$	$\mathcal{I}(\{x\}_{\mathsf{pub}(B)})$
A	$\rightarrow$	$B : \{s\}_{\mathsf{pub}(B)}$	$\forall x \cdot \mathcal{I}(\{x\}_{\mathsf{pub}(A)})$	$\Rightarrow$	$\mathcal{I}(x)$

Résultat: Le protocole est sûr si l'ensemble des formules codant:

- la capacité de déduction de l'attaquant;
- les règles du protocole; et
- la propriété de sécurité, e.g. $\neg\mathcal{I}(s)$

est satisfaisable.

$\longrightarrow$ **Question**: Est-ce que $\mathcal{C}_P \cup \mathcal{C}_A \cup \{\neg\mathcal{I}(s)\}$ est satisfaisable?

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$;

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$; $\mathcal{I}(\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)})$;

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$; $\mathcal{I}(\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)})$; $\mathcal{I}(\{\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)}\}_{\mathsf{pub}(B)})$;

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$; $\mathcal{I}(\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)})$; $\mathcal{I}(\{\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)}\}_{\mathsf{pub}(B)})$; ...

... mais la saturation ne termine pas (même sur des exemples simples)

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$; $\mathcal{I}(\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)})$; $\mathcal{I}(\{\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)}\}_{\mathsf{pub}(B)})$; ...

... mais la saturation ne termine pas (même sur des exemples simples)

Résolution ordonnée avec sélection

$\longrightarrow$ choisir « soigneusement » le litéral sur lequel on va faire la résolution

Comment tester la satisfaisabilité ?

Résolution

$\longrightarrow$ saturer l'ensemble de clauses avec toutes ses conséquences logiques

$\mathcal{I}(\{s\}_{\mathsf{pub}(A)})$; $\mathcal{I}(\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)})$; $\mathcal{I}(\{\{\{s\}_{\mathsf{pub}(A)}\}_{\mathsf{pub}(B)}\}_{\mathsf{pub}(B)})$; ...

... mais la saturation ne termine pas (même sur des exemples simples)

Résolution ordonnée avec sélection

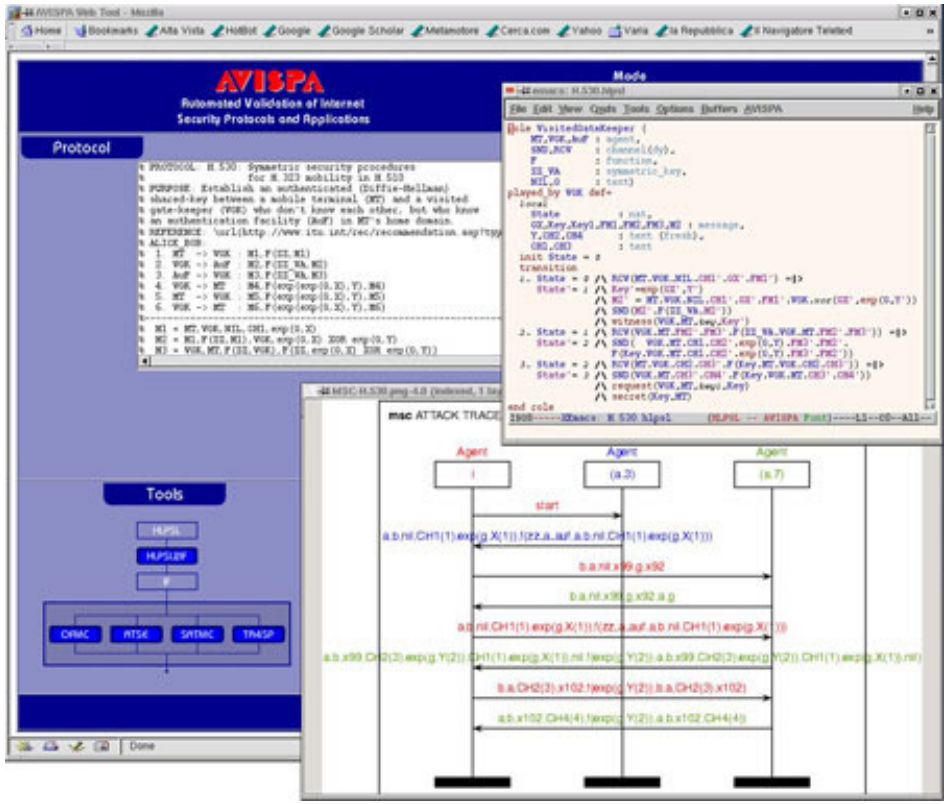
$\longrightarrow$ choisir « soigneusement » le litéral sur lequel on va faire la résolution

- on peut garantir la terminaison pour des classes intéressantes;
- ça termine souvent en pratique.

Retour

Outil de vérification AVISPA

Outil disponible en ligne: `http://www.avispa-project.org/`



$\longrightarrow$ Projet Européen (France, Italie, Allemagne, Suisse)

Conclusion

Les méthodes formelles permettent une bonne analyse des protocoles cryptographiques.

- découverte de failles à l'aide d'outil de vérification;
- des preuves formelles de sécurité peuvent être obtenues automatiquement (sans intervention humaine).

Conclusion

Les méthodes formelles permettent une bonne analyse des protocoles cryptographiques.

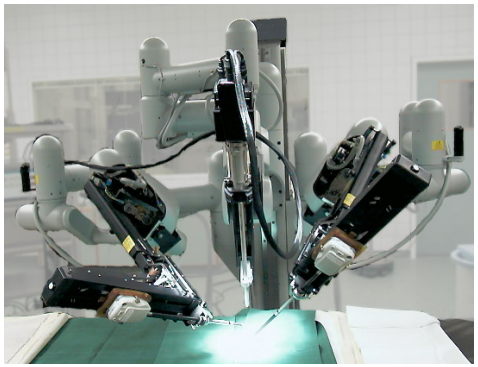
- découverte de failles à l'aide d'outil de vérification;
- des preuves formelles de sécurité peuvent être obtenues automatiquement (sans intervention humaine).

Il reste cependant beaucoup à faire:

- savoir analyser différentes propriétés de sécurité
  $\longrightarrow$ l'anonymat sous toutes ses formes!
- prendre en compte les propriétés mathématiques du chiffrement (e.g. chiffrement homomorphique)
- réaliser ses analyses formelles dans des modèles plus réalistes.

Il reste beaucoup à faire (suite)

Les logiciels sont en perpétuelle évolution ...



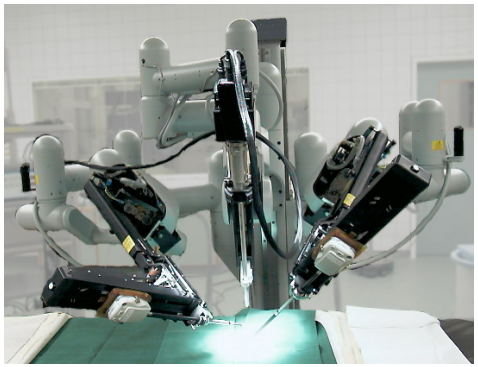
- en vue de leur amélioration,
- pour développer de nouvelles applications

  $\longrightarrow$ vote électronique, robot en chirurgie, ...

... et sont de plus en plus complexes.

Il reste beaucoup à faire (suite)

Les logiciels sont en perpétuelle évolution ...



- en vue de leur amélioration,
- pour développer de nouvelles applications

  $\longrightarrow$ vote électronique, robot en chirurgie, ...

... et sont de plus en plus complexes.

L'informatique est une **discipline** très vaste et en plein essor.

- informatique de la vérification,
- bioinformatique,
- exploration de données, ...

Enseigner l'informatique au lycée apparaît comme une nécessité, du fait de la place de cette discipline aussi bien dans notre économie et dans notre société que parmi les outils qui nous permettent de comprendre le monde.

« Quelle informatique enseigner au lycée ? », Gilles Dowek, Mars 2005.