Prouver la sécurité informatique : la logique à la rescousse

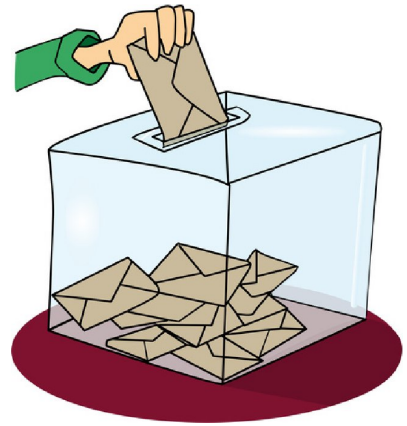
Véronique Cortier [1]

Mars 2015

1. LORIA - CNRS, INRIA, Université de Lorraine

Securité informatique

De nombreux actes de la vie quotidienne sont maintenant numériques.



Enjeux de sécurité : confidentialité, authentification, respect de la vie privée, équité, non-répudiation, . . .

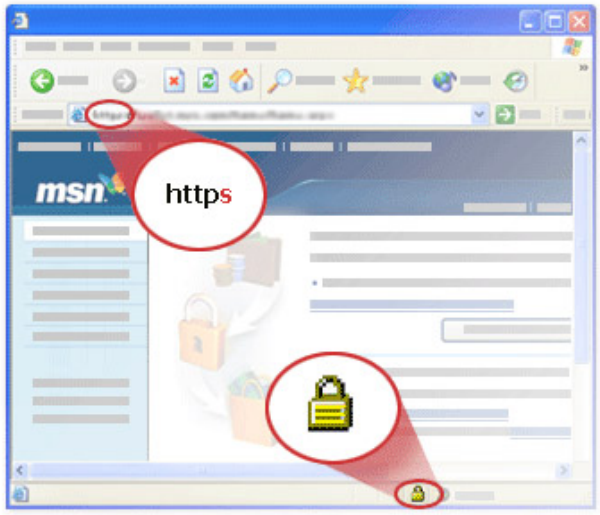
Protocoles cryptographiques

Protocole : description du comportement de chaque participant / entité pour arriver au but désiré (e.g. échanger une information confidentielle)

Cryptographie : techniques de codage et décodage des messages

- chiffrement
- signature
- hachage
- ...

Exemple : HTTPS



- TLS : Transport Security Layer, depuis 1994
- de nombreuses implémentations : OpenSSL, SecureTransport, JSSE, ...
- De nombreux bugs et attaques, des corrections chaque année

Dépassement de mémoire

`Go to fail`

Vérifications manquantes (MACs, signatures, ...)

FREAK attack - fin février 2015

Bhargavan et al.

Partie I

Connaître son ennemi

Ou comment apprendre à devenir paranoïaque.

Que peut faire un attaquant ?

Le réseau est public (Internet, Wifi, GSM)
$\rightarrow$ les messages peuvent être lus

Que peut faire un attaquant ?

Le réseau est public (Internet, Wifi, GSM)
$\rightarrow$ les messages peuvent être lus

Pas de sécurité par l'obscurité !

"Il faut que [le système de chiffrement] n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi."

Principes de Kerckhoffs, 1883

$\rightarrow$ L'attaquant connaît le système.

Comment échanger un secret ?

Brique de base : chiffrement symétrique

Comment échanger un secret ?

Brique de base : chiffrement symétrique

Ici, chiffrement commutatif

$$\{\{\text{Bonjour}\}_{k_1}\}_{k_2} = \{\{\text{Bonjour}\}_{k_2}\}_{k_1}$$

Échange d'un secret avec du chiffrement commutatif

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

Échange d'un secret avec du chiffrement commutatif

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\leftarrow \left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}}$$

Échange d'un secret avec du chiffrement commutatif

$$\xrightarrow{\{\text{pin} : 3443\}_{k_{\text{alice}}}}$$

$$\xleftarrow{\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}}}$$

$$\xrightarrow{\{\text{pin} : 3443\}_{k_{\text{bob}}}}$$

Puisque $\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}} = \left\{\{\text{pin} : 3443\}_{k_{\text{bob}}}\right\}_{k_{\text{alice}}}$

Échange d'un secret avec du chiffrement commutatif

$$\xrightarrow{\{\text{pin} : 3443\}_{k_{\text{alice}}}}$$

$$\xleftarrow{\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}}}$$

$$\xrightarrow{\{\text{pin} : 3443\}_{k_{\text{bob}}}}$$

$\rightarrow$ Une attaque est possible ! (manque d'authentification)

Échange d'un secret avec du chiffrement commutatif

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}} \longleftarrow$$

$$\{\text{pin} : 3443\}_{k_{\text{bob}}} \longrightarrow$$

$\rightarrow$ Une attaque est possible ! (manque d'authentification)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{intrus}}} \longleftarrow$$

$$\{\text{pin} : 3443\}_{k_{\text{intrus}}} \longrightarrow$$

Un attaquant actif

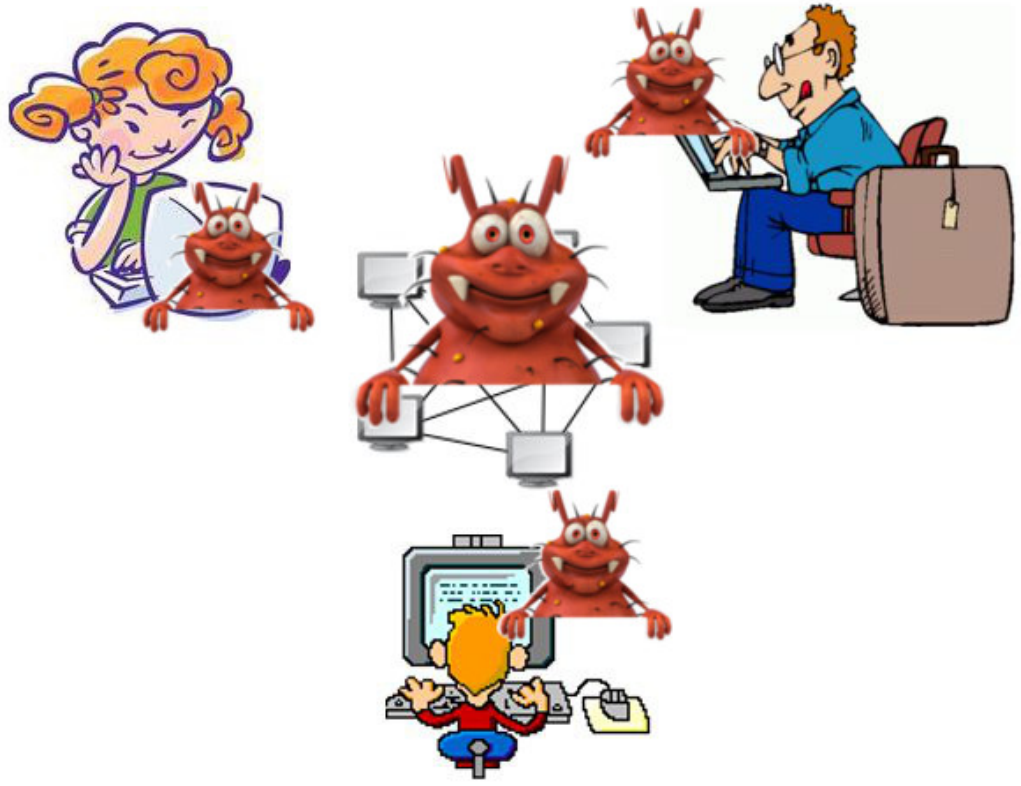
Des utilisateurs malhonnêtes peuvent :

- lire les messages envoyés,
- intercepter certains messages,
- construire et envoyer des messages,
- prendre part aux protocoles.

Votre ordinateur est-il sûr ?

- virus
- bugs
- cheval de Troie

Votre ordinateur est-il sûr ?



- virus
- bugs
- cheval de Troie

API de sécurité

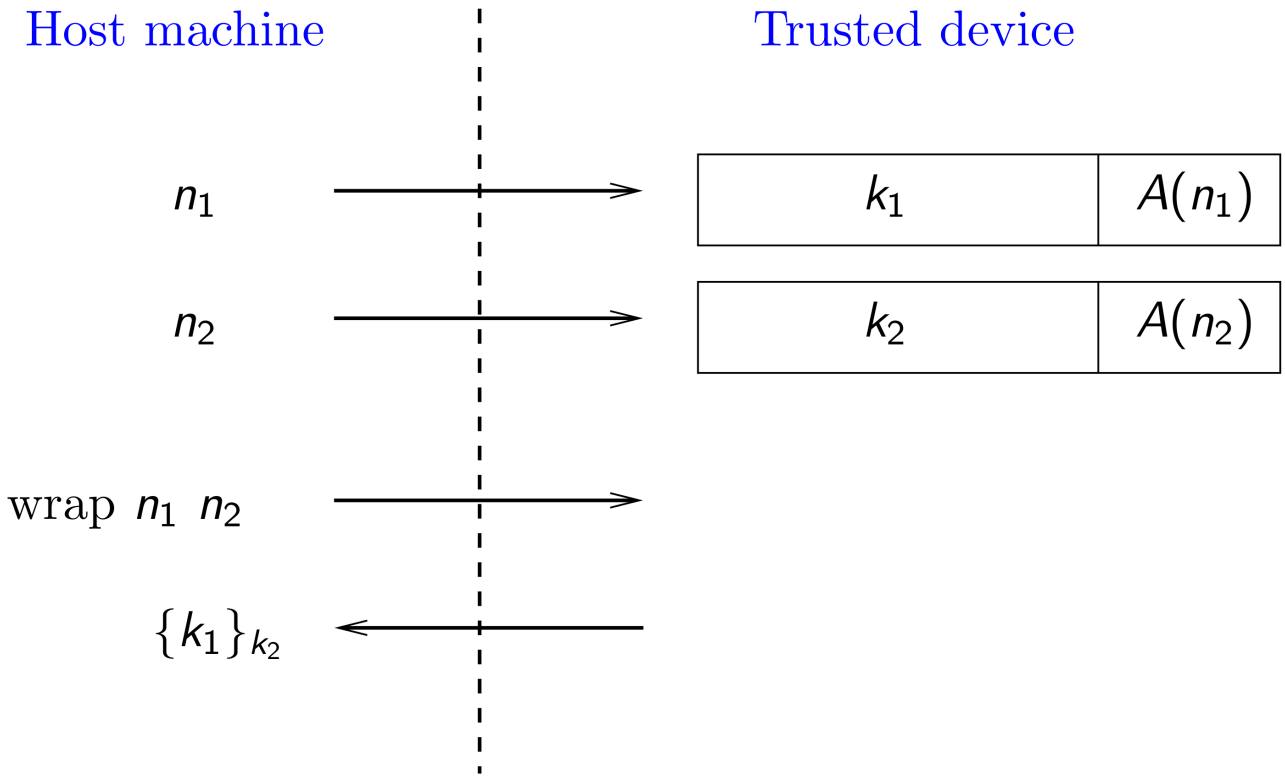
Machine hôte

Dispositif de sécurité

API de sécurité

Objectif : Assurer la sécurité des données stockées dans le dispositif, même s'il est connecté à une machine non fiable.

Exemple de fonctionnement : PKCS#11



De nombreuses attaques

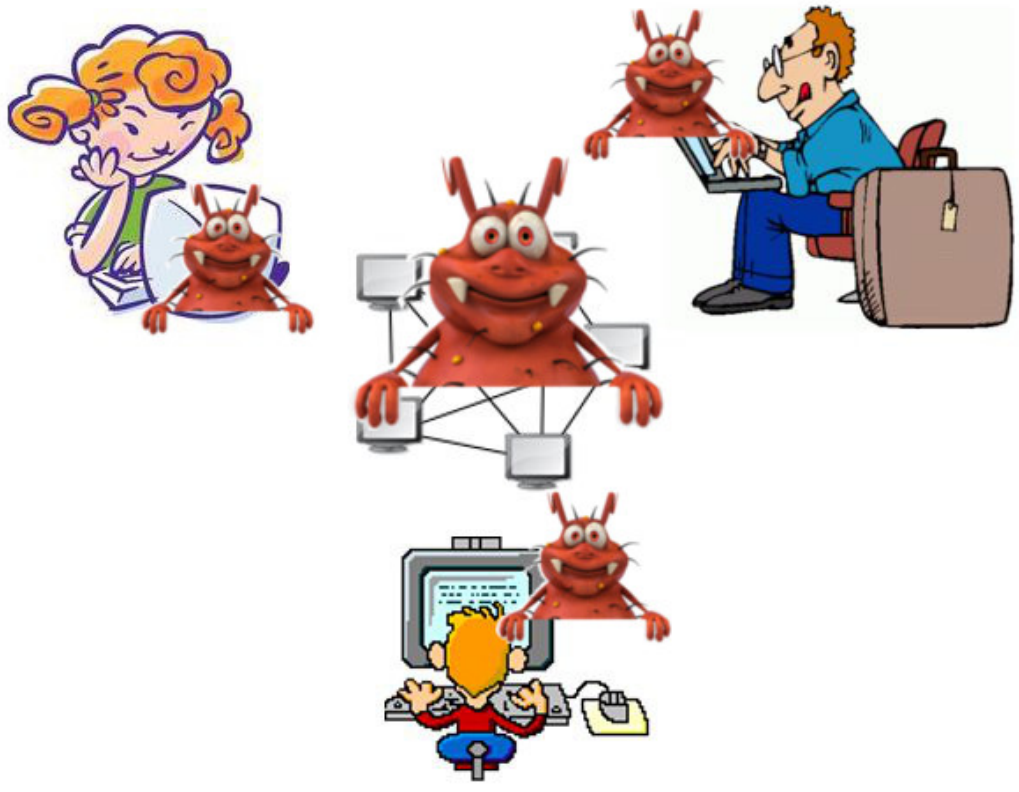
	Device		Supported Functionality						Attacks found					
	Company	Model	sym	asym	cobj	chan	w	ws	a1	a2	a3	a4	a5	mc
USB	Aladdin	eToken PRO	✓	✓	✓	✓	✓	✓	✓	✓				a1
	Athena	ASEKey	✓	✓	✓									
	Bull	Trustway RCI	✓	✓	✓	✓	✓	✓	✓	✓				a1
	Eutron	Crypto Id. ITSEC		✓	✓									
	Feitian	StorePass2000	✓	✓	✓	✓	✓	✓		✓	✓	✓		a3
	Feitian	ePass2000	✓	✓	✓	✓	✓	✓		✓	✓	✓		a3
	Feitian	ePass3003Auto	✓	✓	✓	✓	✓	✓		✓	✓	✓		a3
	Gemalto	Smart Enterprise Guardian		✓		✓								
	MXI Security	Stealth MXP Bio	✓	✓		✓								
	SafeNet	iKey 2032	✓	✓	✓		✓							
	Sata	DKey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	a3
Card	ACS	ACOS5	✓	✓	✓	✓								
	Athena	ASE Smartcard	✓	✓	✓									
	Gemalto	Cyberflex V2	✓	✓	✓		✓	✓		✓				a2
	Gemalto	SafeSite Classic TPC IS V1		✓		✓								
	Gemalto	SafeSite Classic TPC IS V2	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	a3
	Siemens	CardOS V4.3 B	✓	✓	✓		✓					✓		a4
Soft	Eracom	HSM simulator	✓	✓		✓	✓	✓	✓	✓		✓		a1
	IBM	opencryptoki 2.3.1	✓	✓	✓	✓	✓	✓	✓	✓		✓		a1

Attaques découvertes à l'aide d'un outil de model-checking

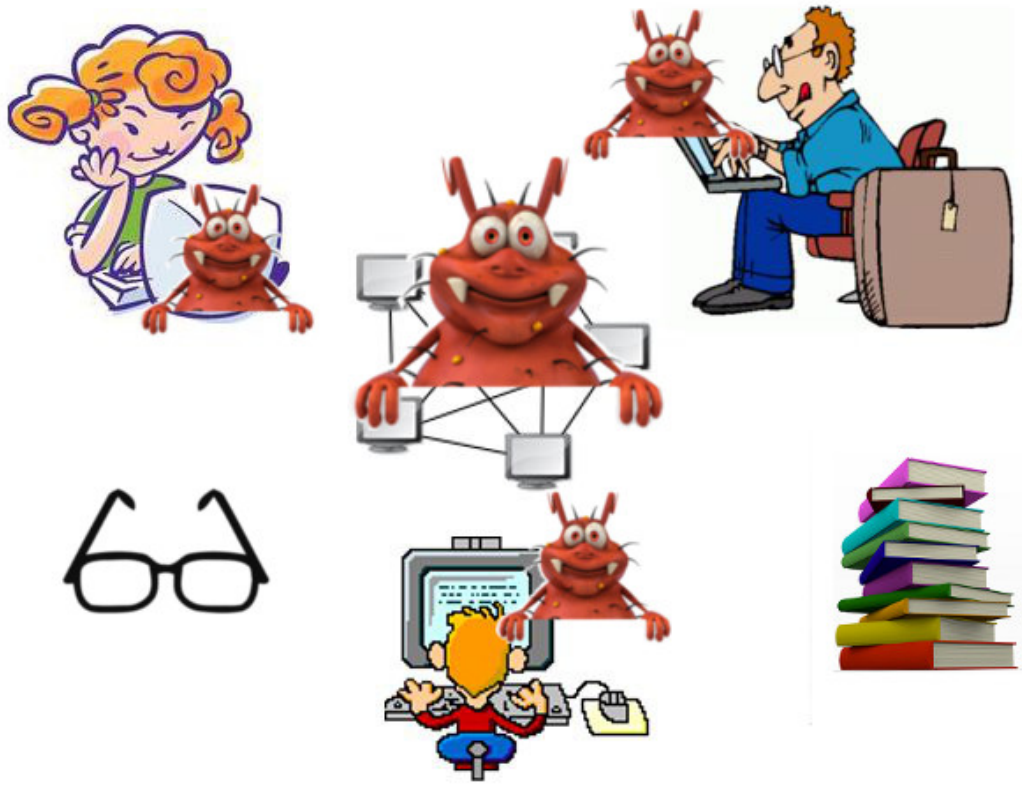
[Graham Steel et al.]

Récapitulatif : un attaquant potentiellement très puissant

Récapitulatif : un attaquant potentiellement très puissant



Récapitulatif : un attaquant potentiellement très puissant



Faut-il s'attendre à pire ?

Attaques par canaux cachés

Un détour : Exponentiation rapide

Comment calculer 7^{42} ?

1. $7 \times 7 \times \cdots \times 7 \rightarrow 42$ opérations

Attaques par canaux cachés

Un détour : Exponentiation rapide

Comment calculer 7^{42} ?

1. $7 \times 7 \times \cdots \times 7 \rightarrow 42$ opérations
2. On remarque que 42 s'écrit 10101 en binaire *i.e.*

$$42 = 2^5 + 2^3 + 2$$

- Par carrés successifs, on calcule $7, 7^2, 7^{2^2}, 7^{2^3}, 7^{2^4}, 7^{2^5}$.
- Puis on multiplie $7^{42} = 7^{2^5} \times 7^{2^3} \times 7$
- Soit 5 carrés et 2 multiplications seulement !

Chiffrement RSA

Chiffrement RSA de m par k : m^k où

$$k \approx 2^{2048} \approx 10^{617}$$

Calcul de m^k : 2048 carrés et au plus 2048 multiplications.

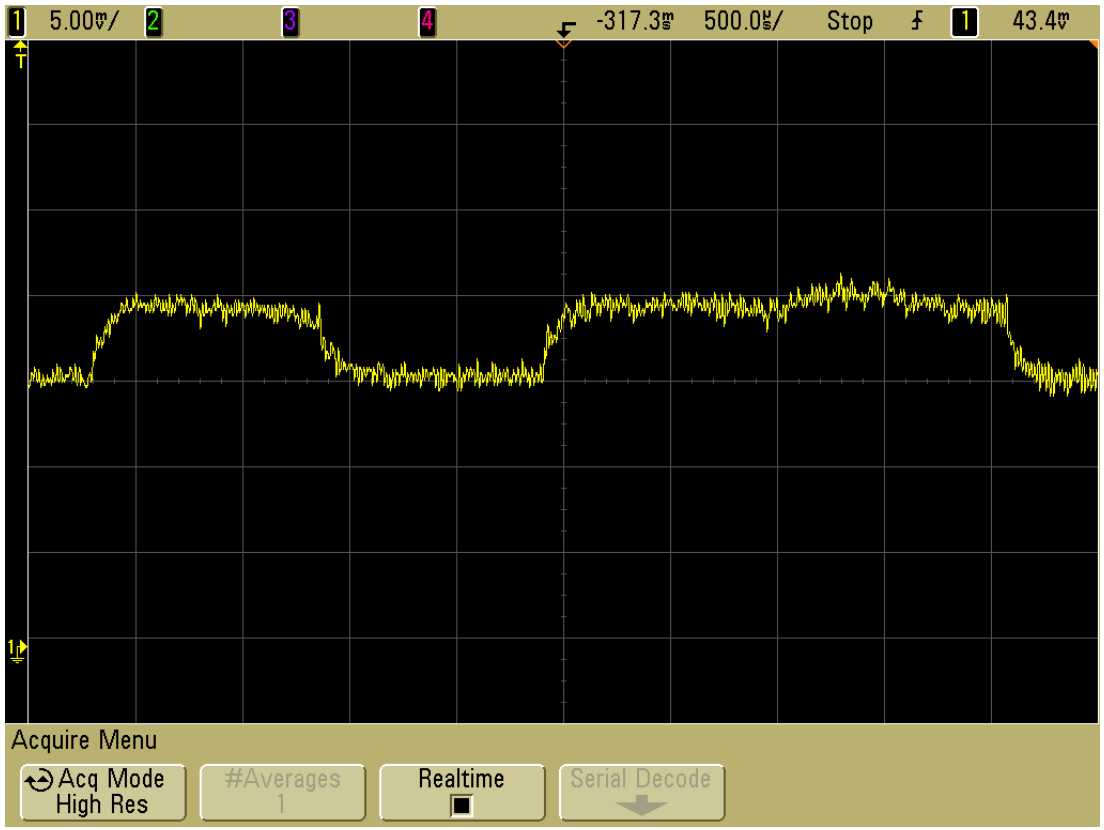
Chiffrement RSA

Chiffrement RSA de m par k : m^k où

$$k \approx 2^{2048} \approx 10^{617}$$

Calcul de m^k : 2048 carrés et au plus 2048 multiplications.

Difficulté : Un carré est plus "facile" qu'une multiplication

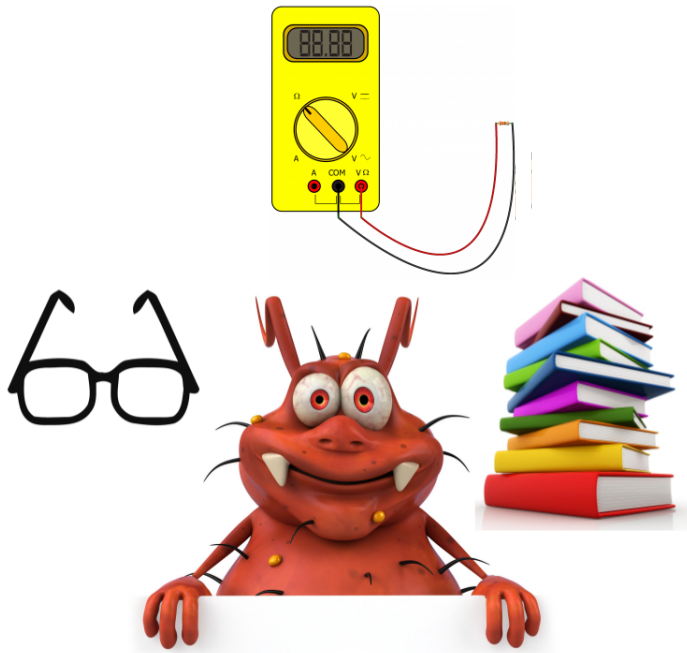


Différents canaux cachés

- ...

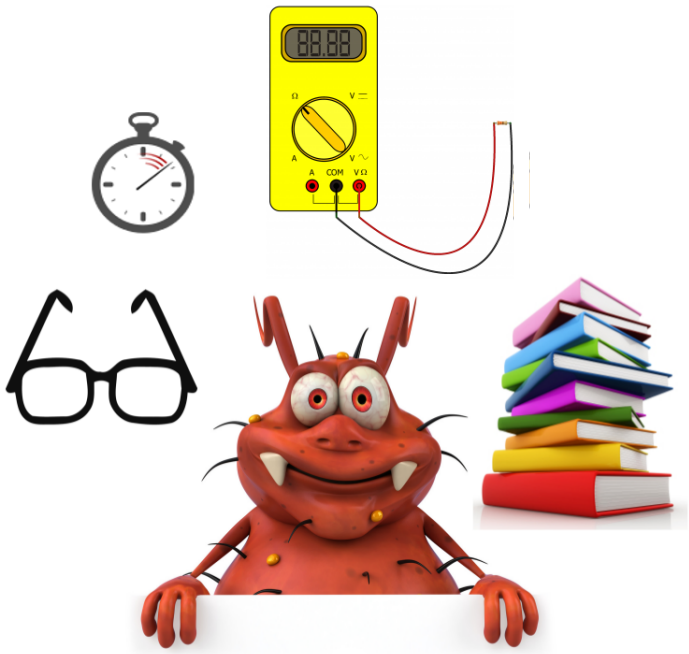
Différents canaux cachés

- Mesure de consommation
-
-
-
-
- ...



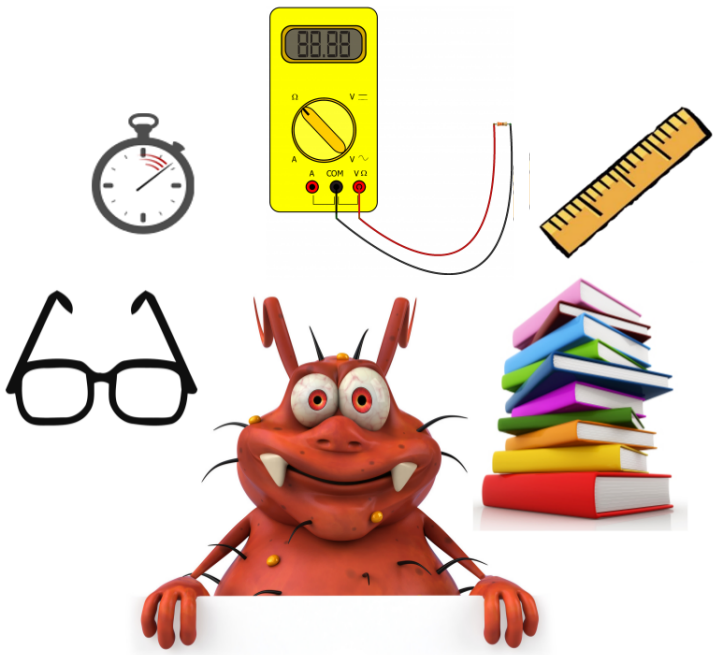
Différents canaux cachés

- Mesure de consommation
- Temps d'exécution
-
-
-
- ...



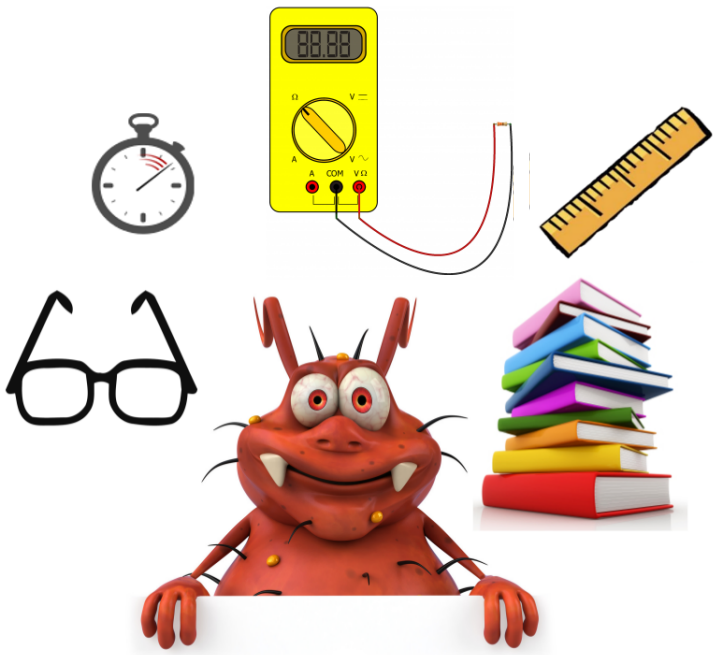
Différents canaux cachés

- Mesure de consommation
- Temps d'exécution
- Longueurs des messages
-
-
- ...



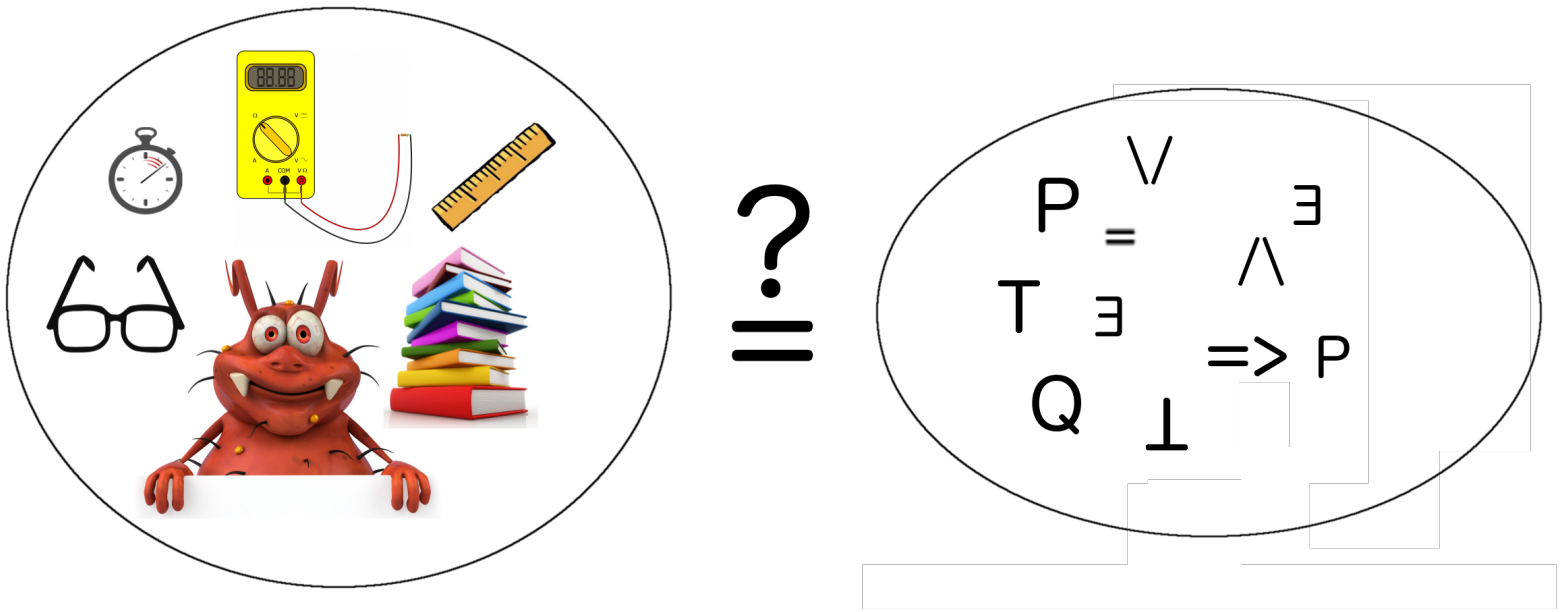
Différents canaux cachés

- Mesure de consommation
- Temps d'exécution
- Longueurs des messages
- Rayonnement magnétique
- Bruit (composants, imprimante, . . .)
- . . .



Partie II

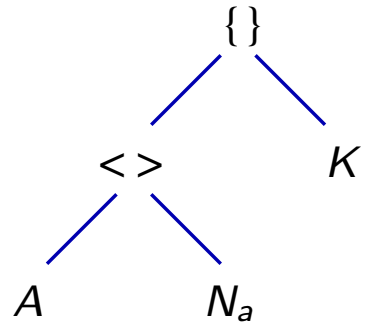
Modèles et techniques d'analyse



Messages

Les messages sont représentés par des termes (= graphes étiquetés).

Exemple :
Le message $\{\langle A, N_a \rangle\}_K$ est représenté par :



$\rightarrow$ Seule la structure des messages est conservée

Attaquant

Les calculs de l'attaquant peuvent être représentés par des formules logiques.

$\forall x \forall y$	$I(x), I(y)$	$\Rightarrow$	$I(\{x\}_y)$	chiffrement
$\forall x \forall y$	$I(\{x\}_y), I(y)$	$\Rightarrow$	$I(x)$	déchiffrement
$\forall x \forall y$	$I(x), I(y)$	$\Rightarrow$	$I(< x, y >)$	concaténation
$\forall x \forall y$	$I(< x, y >)$	$\Rightarrow$	$I(x)$	1ère projection
$\forall x \forall y$	$I(< x, y >)$	$\Rightarrow$	$I(y)$	2ème projection

Représentation d'un protocole en formules logiques

secret : 3443

$\{\text{secret} : 3443\}_{k_{\text{alice}}}$ →

$\leftarrow \{\{\text{secret} : 3443\}_{k_{\text{alice}}}\}_{k_{\text{bob}}}$

$\{\text{secret} : 3443\}_{k_{\text{bob}}}$ →

secret : 3443

Les actions du protocole sont représentées par des implications logiques.

$$\begin{array}{lrcl} & & \Rightarrow & I(\{\text{secret}\}_{k_a}) \\ \forall x & I(x) & \Rightarrow & I(\{x\}_{k_b}) \\ \forall x & I(\{x\}_{k_a}) & \Rightarrow & I(x) \end{array}$$

La sécurité revient à la cohérence d'une théorie

sûr ?

$$\rightsquigarrow$$

$$
\begin{array}{lrcl}
\forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(\{x\}_y) \\
\forall x \forall y & I(\{x\}_y), I(y) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(y)
\end{array}
$$

$$
\begin{array}{lrcl}
 & & & I(\{\text{secret}\}_{k_a}) \\
\forall x & I(x) & \Rightarrow & I(\{x\}_{k_b}) \\
\forall x & I(\{x\}_{k_a}) & \Rightarrow & I(x)
\end{array}
$$

La sécurité revient à la cohérence d'une théorie

sûr ?

$$\rightsquigarrow$$

$$\begin{array}{lrcl} & & & \neg I(\text{secret}) \\ \forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) \\ \forall x \forall y & I(x), I(y) & \Rightarrow & I(\{x\}_y) \\ \forall x \forall y & I(\{x\}_y), I(y) & \Rightarrow & I(x) \\ \forall x \forall y & I(< x, y >) & \Rightarrow & I(x) \\ \forall x \forall y & I(< x, y >) & \Rightarrow & I(y) \end{array}$$

$$\begin{array}{lrcl} & & & I(\{\text{secret}\}_{k_a}) \\ \forall x & I(x) & \Rightarrow & I(\{x\}_{k_b}) \\ \forall x & I(\{x\}_{k_a}) & \Rightarrow & I(x) \end{array}$$

N'aboutit pas à une contradiction ?

(i.e. théorie cohérente ?)

Comment savoir si un ensemble de formules est cohérent ?

Programme de Hilbert (1928)
"Entscheidung Problem"

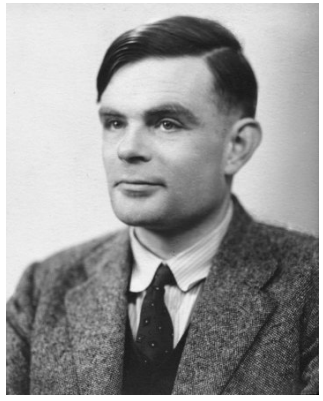
David Hilbert

Comment savoir si un ensemble de formules est cohérent ?

Programme de Hilbert (1928)
"Entscheidung Problem"

David Hilbert

C'est indécidable ! (1936)
$\rightarrow$ Il n'y a pas d'algorithme qui répond à cette question.



Alan Turing

(Tout cela sans ordinateur)

Revenons à nos moutons

sûr ?

Tout cela pour rien ?

$\neg I(\text{secret})$

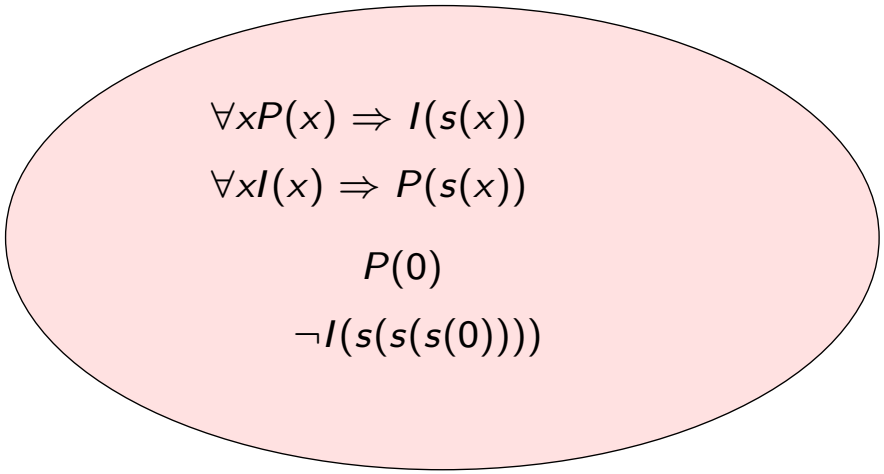
$$
\begin{array}{lrcl}
\forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(\{x\}_y) \\
\forall x \forall y & I(\{x\}_y), I(y) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(y) \\
 & & & I(\{\text{secret}\}_{k_a}) \\
\forall x & I(x) & \Rightarrow & I(\{x\}_{k_b}) \\
\forall x & I(\{x\}_{k_a}) & \Rightarrow & I(x)
\end{array}
$$

N'aboutit pas à une contradiction ?

(i.e. théorie cohérente ?)

Une technique classique : la résolution

Idée : on ajoute des conséquences logiques ...



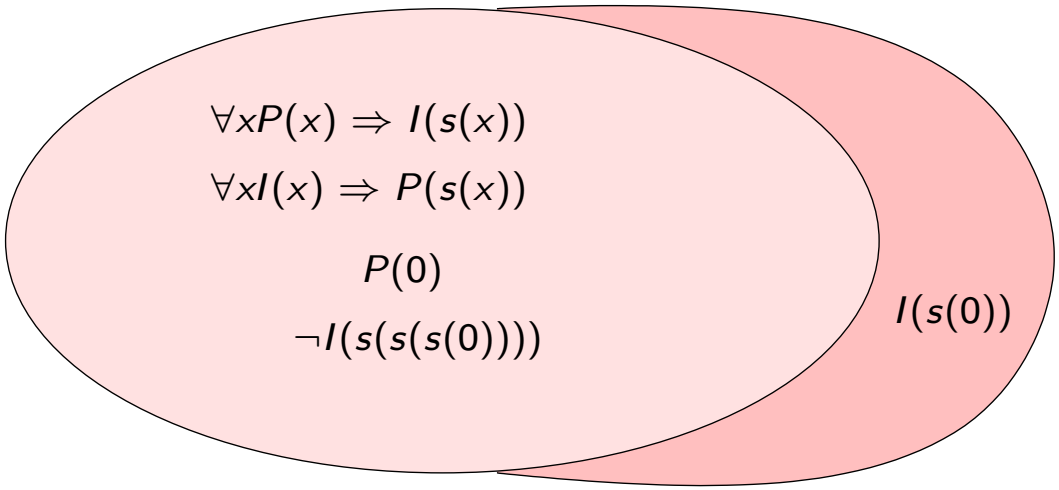
... jusqu'à trouver une contradiction.

Il faut une méthode (stratégie) :

- correcte : les formules ajoutées sont bien des conséquences
- complète : qui trouve la contradiction (si elle existe)
- en un nombre fini d'étapes (fragment décidable)

Une technique classique : la résolution

Idée : on ajoute des conséquences logiques ...



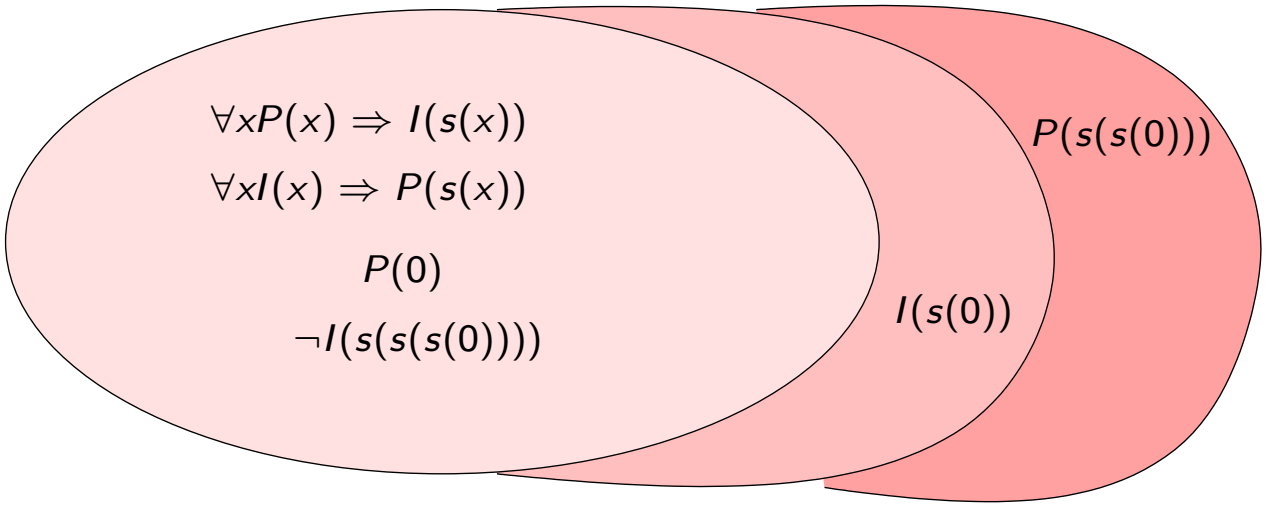
... jusqu'à trouver une contradiction.

Il faut une méthode (stratégie) :

- correcte : les formules ajoutées sont bien des conséquences
- complète : qui trouve la contradiction (si elle existe)
- en un nombre fini d'étapes (fragment décidable)

Une technique classique : la résolution

Idée : on ajoute des conséquences logiques ...



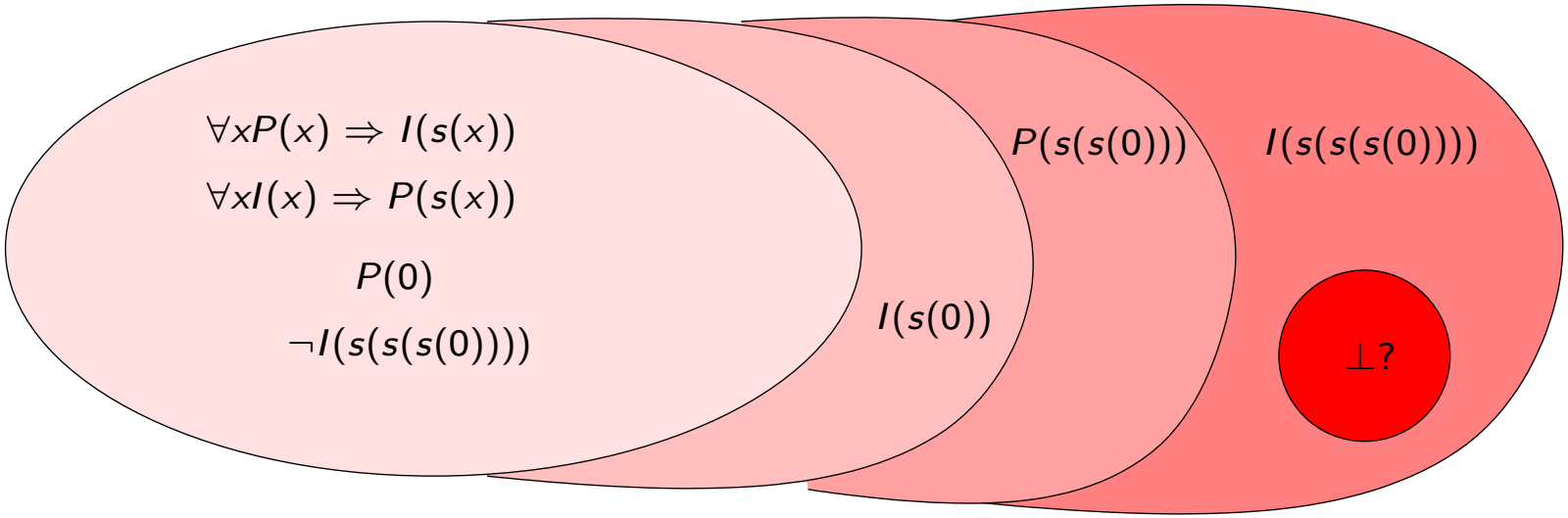
... jusqu'à trouver une contradiction.

Il faut une méthode (stratégie) :

- correcte : les formules ajoutées sont bien des conséquences
- complète : qui trouve la contradiction (si elle existe)
- en un nombre fini d'étapes (fragment décidable)

Une technique classique : la résolution

Idée : on ajoute des conséquences logiques ...



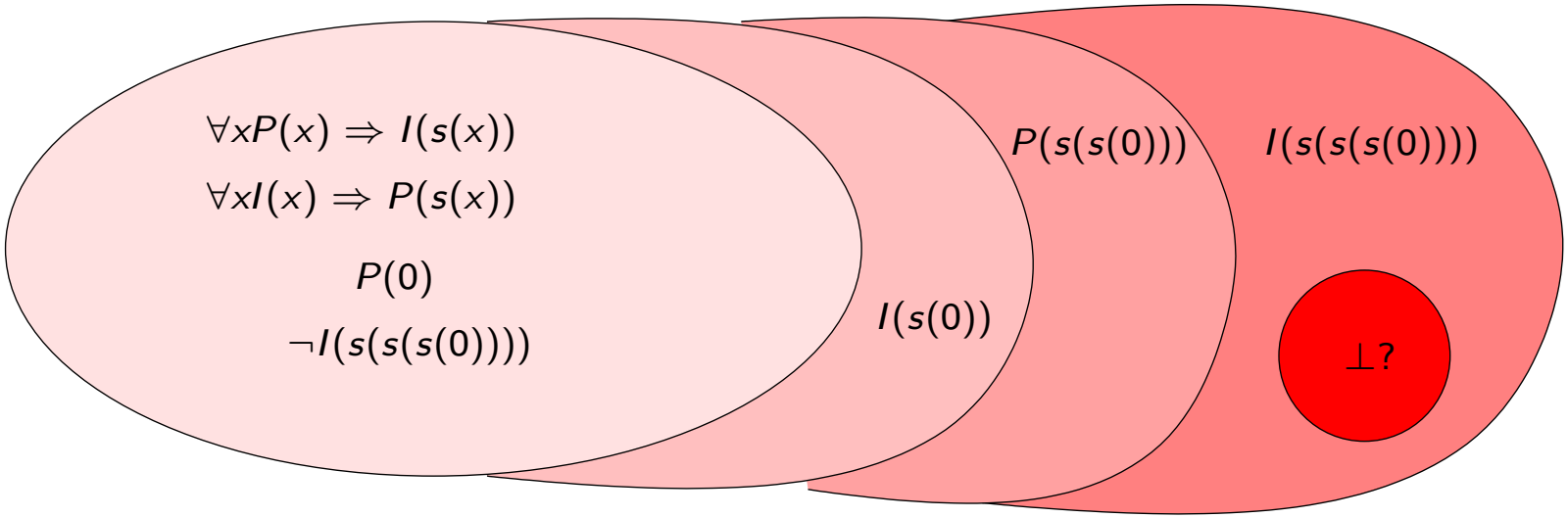
... jusqu'à trouver une contradiction.

Il faut une méthode (stratégie) :

- correcte : les formules ajoutées sont bien des conséquences
- complète : qui trouve la contradiction (si elle existe)
- en un nombre fini d'étapes (fragment décidable)

Une technique classique : la résolution

Idée : on ajoute des conséquences logiques ...



... jusqu'à trouver une contradiction.

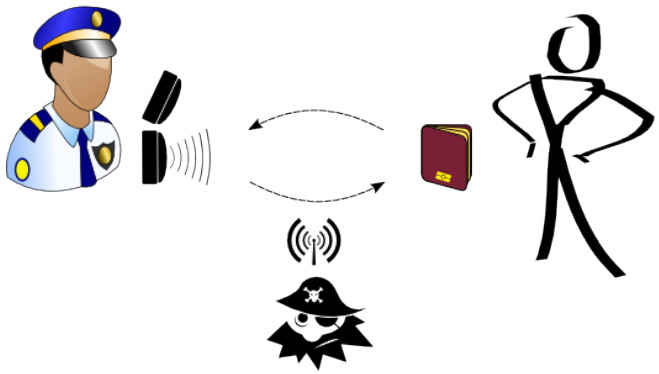
Il faut une méthode (stratégie) :

- correcte : les formules ajoutées sont bien des conséquences
- complète : qui trouve la contradiction (si elle existe)
- en un nombre fini d'étapes (fragment décidable)

$\rightarrow$ ProVerif, logiciel développé par Bruno Blanchet

Propriétés d'anonymat, vie privée

Peut-on tracer le porteur d'un passeport biométrique ?

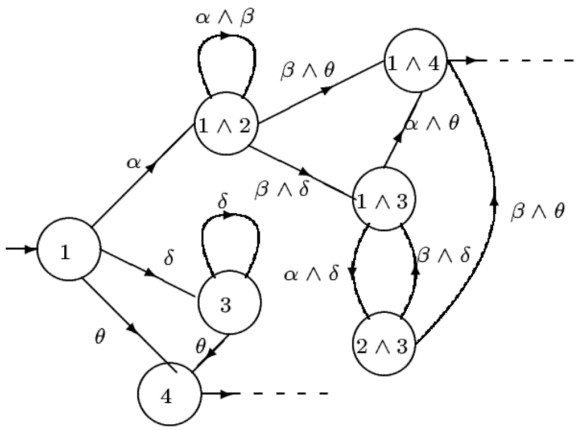


$$\text{Pass}(A) \approx \text{Pass}(B)$$

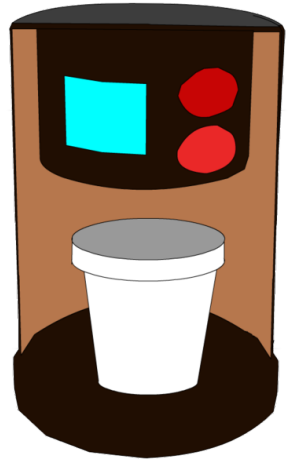
Confidentialité des votes : est-ce que mon vote est secret ?

$$\text{Alice}(1) \mid \text{Bob}(0) \approx \text{Alice}(0) \mid \text{Bob}(1)$$

Equivalence - historique



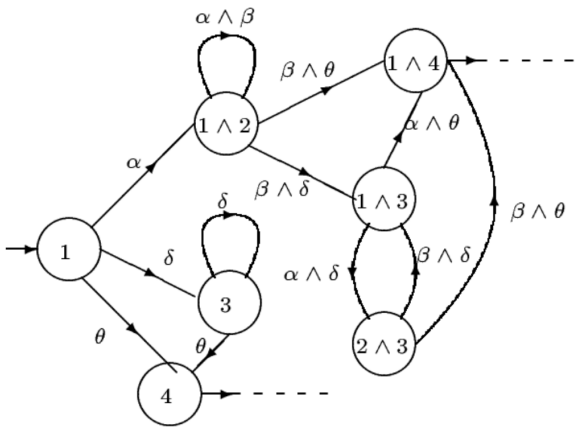
$\approx$



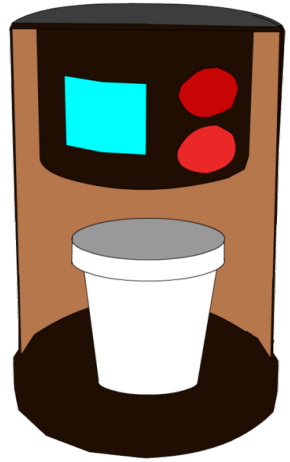
CCS, pi-calcul : années 80, Robin Milner

- communication et concurrence
- action atomique
- équivalence observationelle : est-ce que deux systèmes sont identiques, du point de vue d'un utilisateur ?

Equivalence - historique



$\approx$



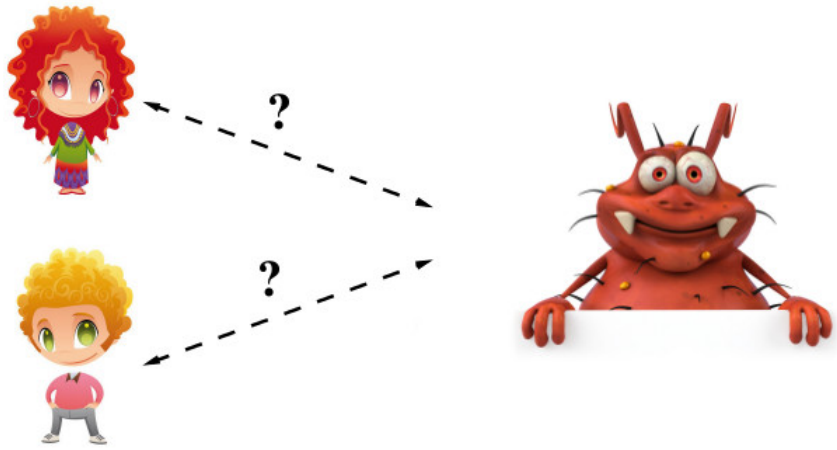
CCS, pi-calcul : années 80, Robin Milner

- communication et concurrence
- action atomique
- équivalence observationelle : est-ce que deux systèmes sont identiques, du point de vue d'un utilisateur ?

pi-calcul appliqué : Martin Abadi et Cédric Fournet, 2001

- extension du pi-calcul
- les messages ne sont plus des atomes mais des termes

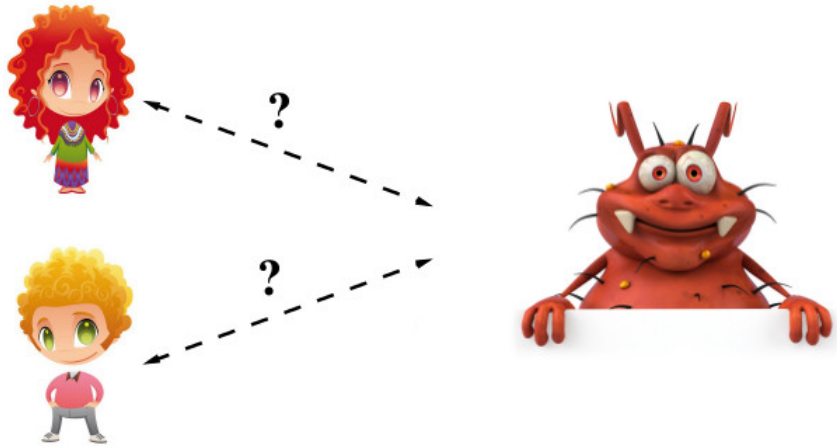
Equivalence - Intuition



Équivalence statique : Deux suites de messages sont équivalentes si l'attaquant ne peut pas mettre au point un test qui les distingue.

$$\{^{\mathsf{enc}(A,k)}/_{x_1},\ ^{k}/_{x_2}\} \overset{?}{\approx} \{^{\mathsf{enc}(B,k)}/_{x_1},\ ^{k}/_{x_2}\}$$

Equivalence - Intuition

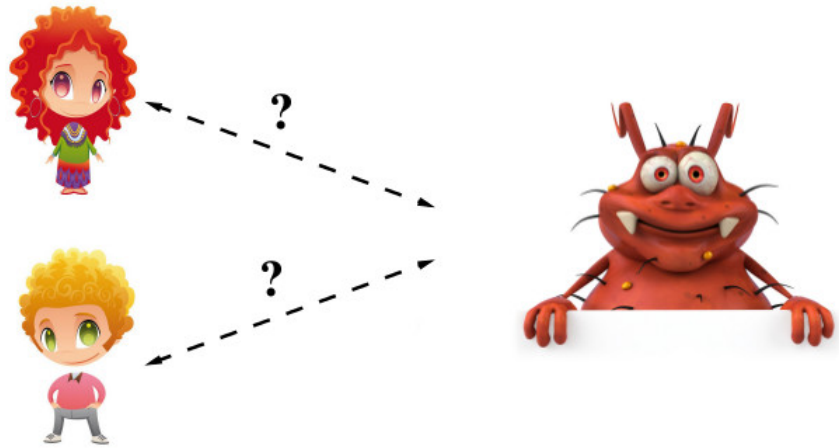


Équivalence statique : Deux suites de messages sont équivalentes si l'attaquant ne peut pas mettre au point un test qui les distingue.

$$\{^{\mathsf{enc}(A,k)}/_{x_1}, \ ^{k}/_{x_2}\} \not\approx \{^{\mathsf{enc}(B,k)}/_{x_1}, \ ^{k}/_{x_2}\}$$

Car $\mathsf{dec}(x_2, x_1) = A$ est vraie à gauche et non à droite.

Equivalence - Intuition



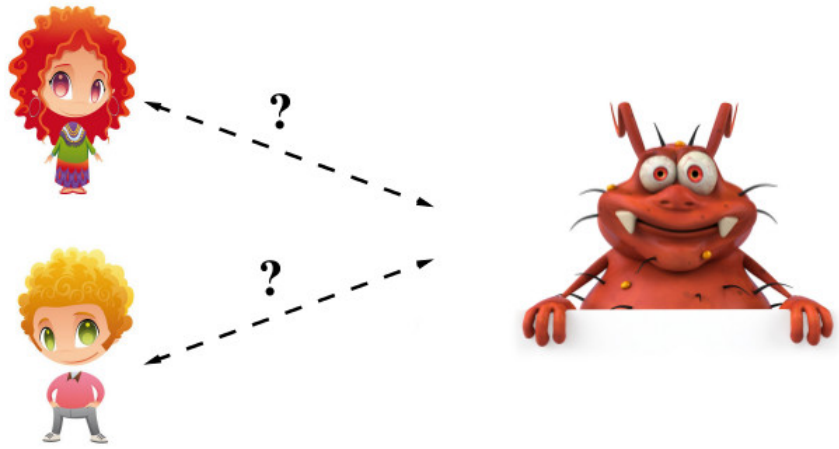
Équivalence statique : Deux suites de messages sont équivalentes si l'attaquant ne peut pas mettre au point un test qui les distingue.

$$\{^{\mathsf{enc}(A,k)}/_{x_1}, \ ^{k}/_{x_2}\} \not\approx \{^{\mathsf{enc}(B,k)}/_{x_1}, \ ^{k}/_{x_2}\}$$

Car $\mathsf{dec}(x_2, x_1) = A$ est vraie à gauche et non à droite.

Équivalence de traces : $P \approx Q$
si pour toute séquence d'actions identiques du point de vue de l'attaquant, les suites de messages résultantes de P et Q sont en équivalence statique.

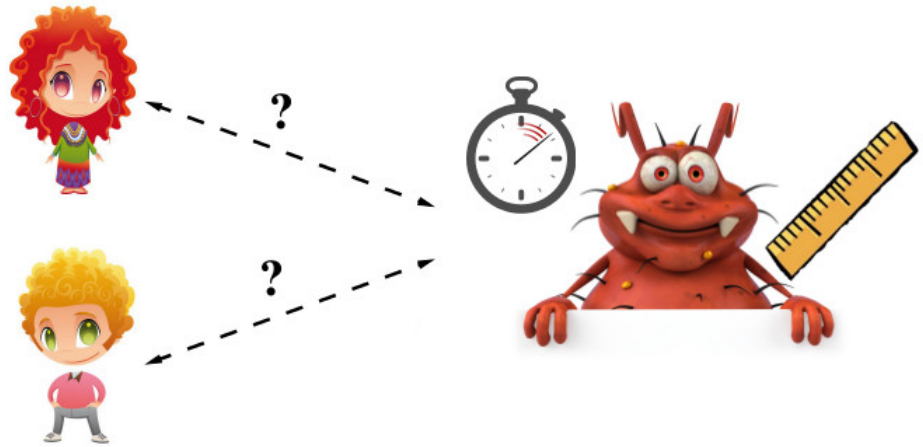
Equivalence - procédures de décision



Nombre borné d'exécutions

- outils APTE, Akiss, SPEC

Equivalence - procédures de décision

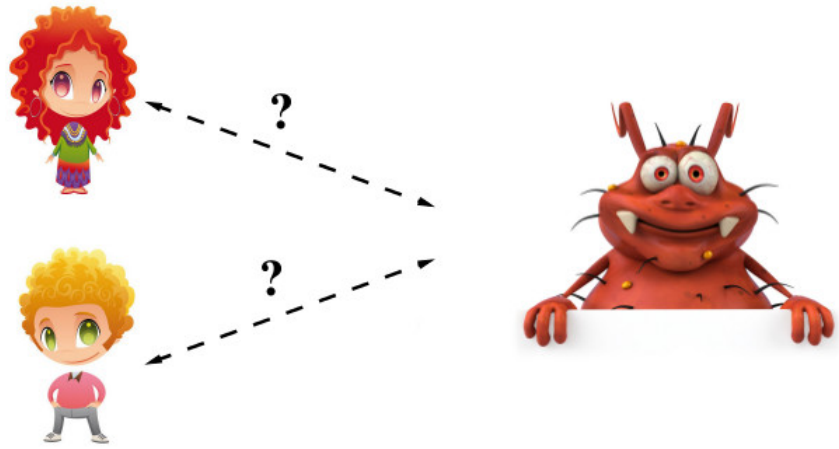


Nombre borné d'exécutions

- outils APTE, Akiss, SPEC
- avec temps et longueurs : Vincent Cheval

$$\ell(f(t_1, \ldots, t_n)) = P_f(\ell(t_1), \ldots . \ell(t_n)) \qquad P_f \in \mathbb{Z}[X_1, \ldots, X_n]$$

Equivalence - procédures de décision



Nombre borné d'exécutions

- outils APTE, Akiss, SPEC
- avec temps et longueurs : Vincent Cheval

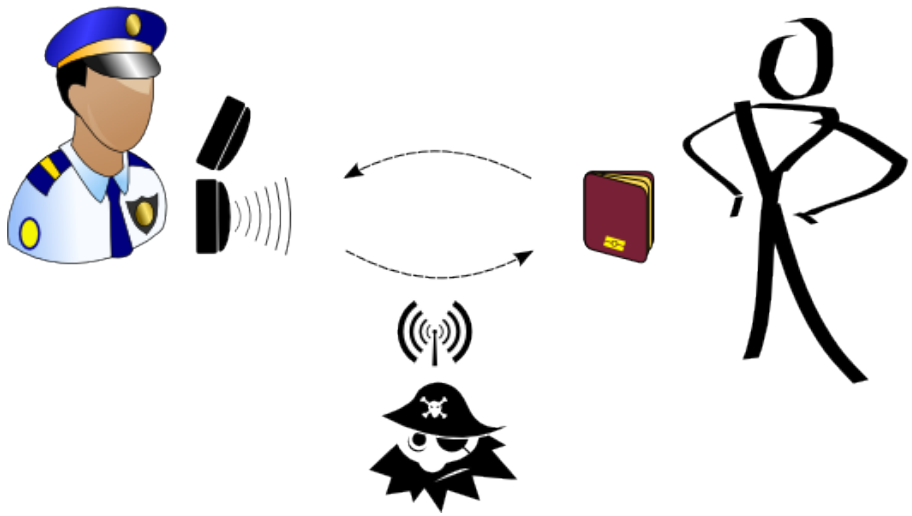
$$\ell(f(t_1,\ldots,t_n)) = P_f(\ell(t_1),\ldots.\ell(t_n)) \qquad P_f \in \mathbb{Z}[X_1,\ldots,X_n]$$

Nombre non borné d'exécutions

- de récents algorithmes de décision thèse Rémy Chrétien
- outil ProVerif (terminaison non garantie)

Cas d'étude : passeport biométrique

Protocole "Passive Authentication"



Une nouvelle attaque permettant de tracer un utilisateur même si les données sont chiffrées

- Testée
- Repose sur une faiblesse classique : messages de taille variable (e.g. image jpeg)
- Peut être corrigée avec du remplissage

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

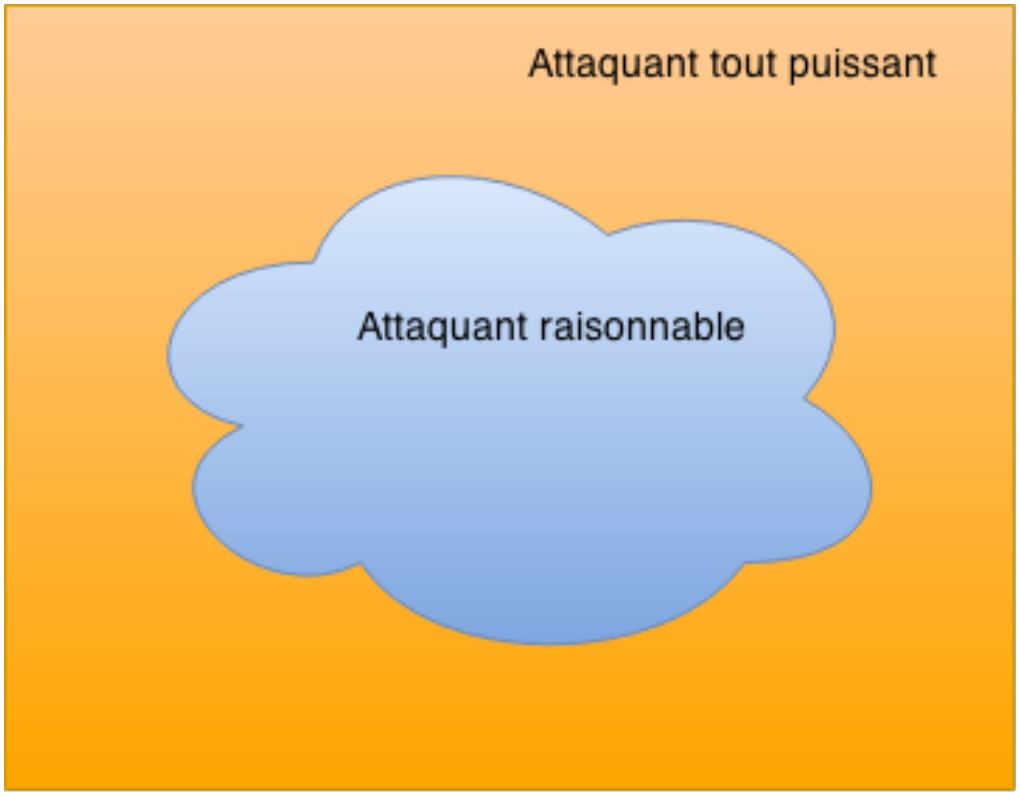
Gergei Bana et Hubert Comon-Lundh

Pas d'hypothèse

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

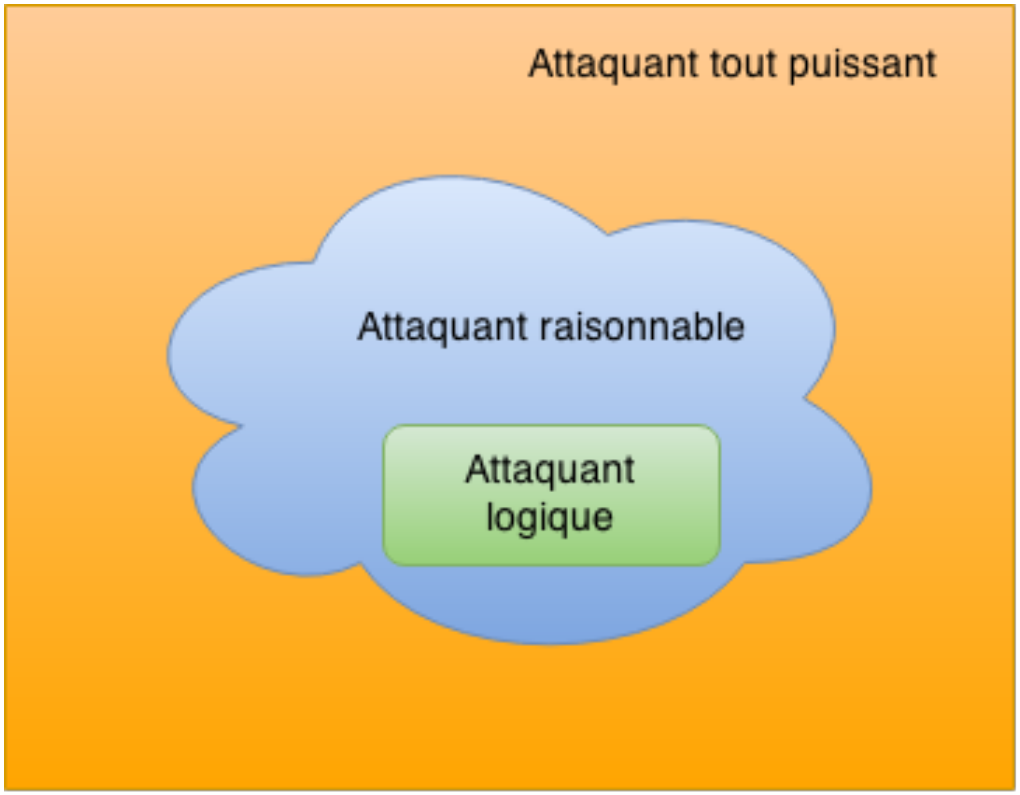
Gergei Bana et Hubert Comon-Lundh



Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh

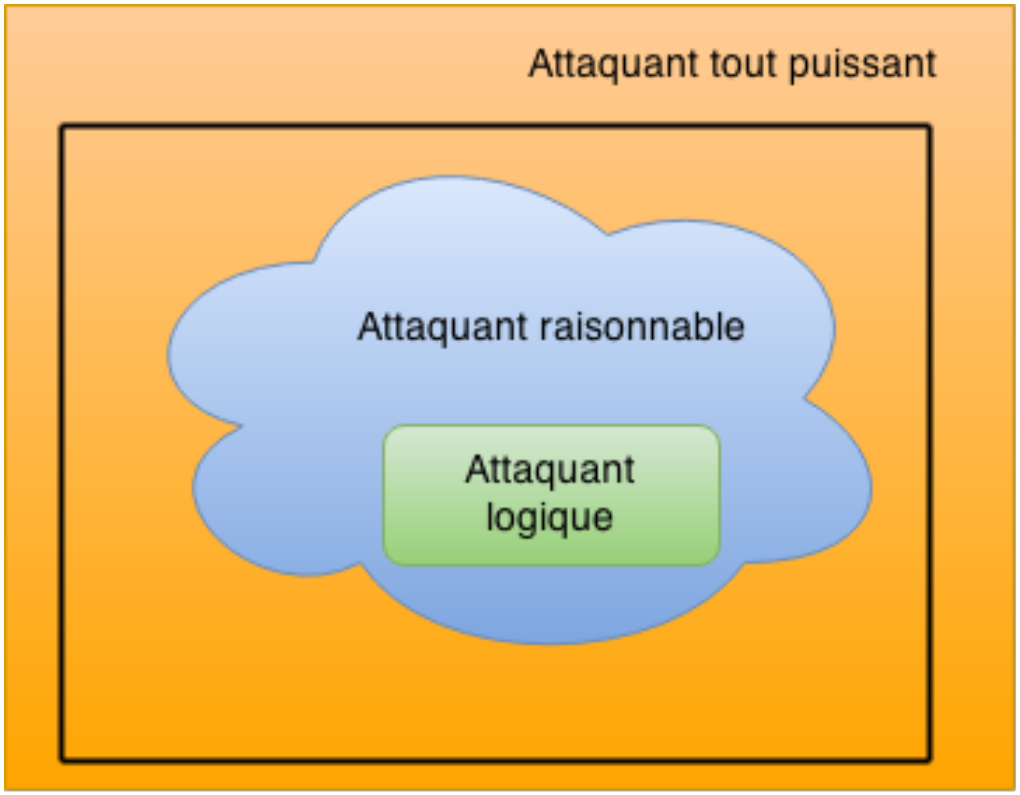


$$I(x), I(y) \Rightarrow I(\{x\}_y)$$
$$I(\{x\}_y), I(y) \Rightarrow I(x)$$

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh



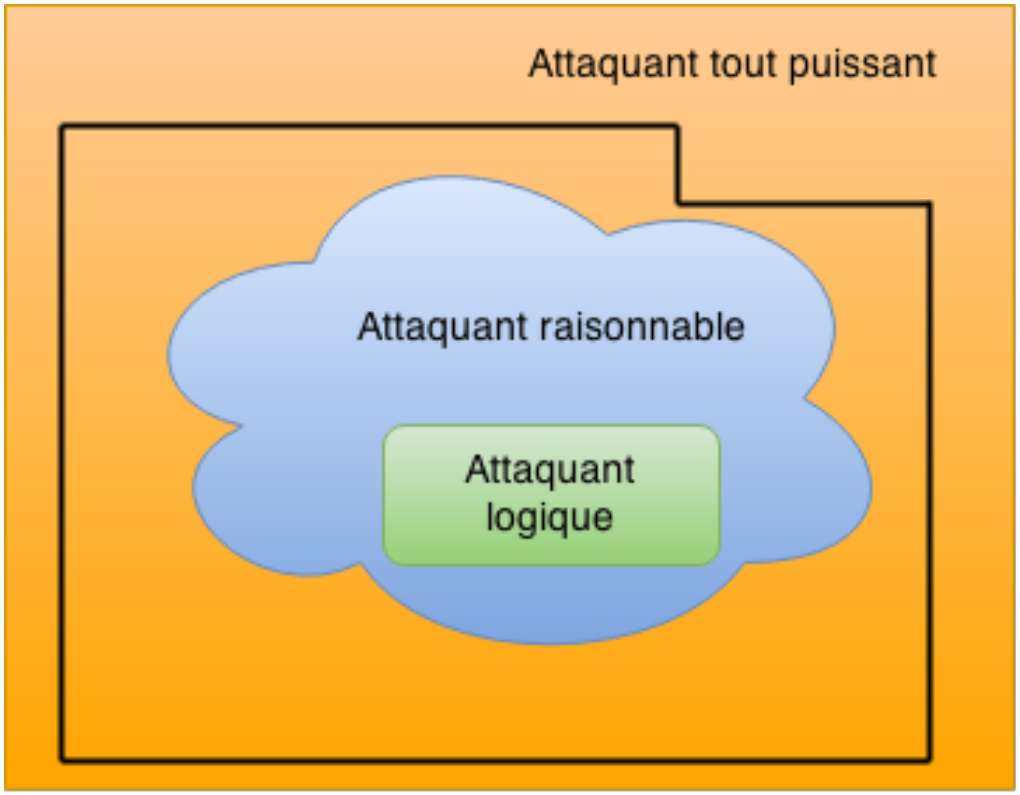
axiome 1

Tout ce qui n'est pas explicitement interdit est autorisé.

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh



axiome 1

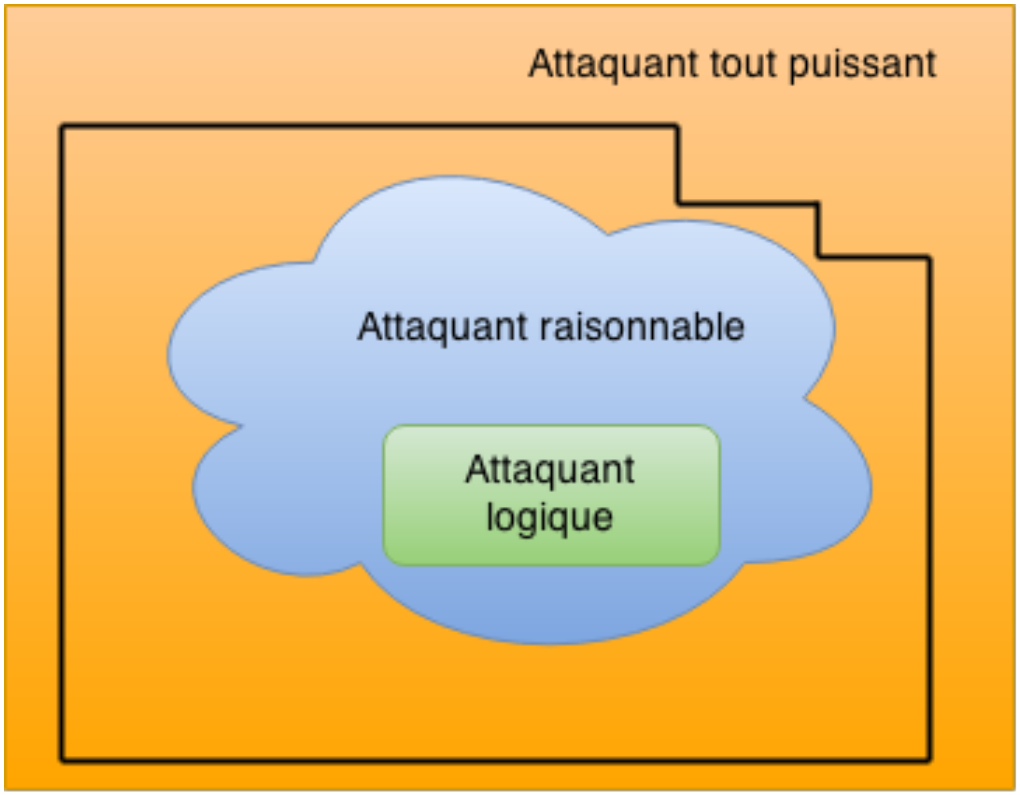
axiome 2

Tout ce qui n'est pas explicitement interdit est autorisé.

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh



axiome 1

axiome 2

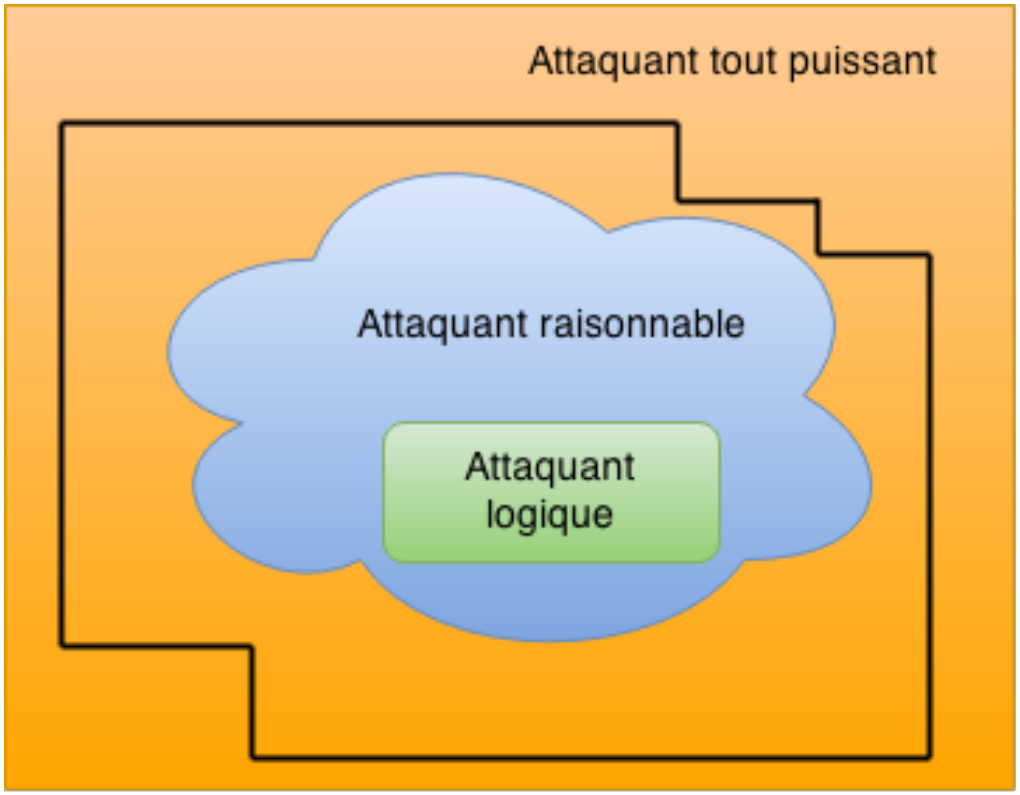
axiome 3

Tout ce qui n'est pas explicitement interdit est autorisé.

Et si on ne connait pas son attaquant ?

Un modèle symbolique permissif

Gergei Bana et Hubert Comon-Lundh



axiome 1
axiome 2
axiome 3
axiome 4

Tout ce qui n'est pas explicitement interdit est autorisé.

Axiomes : exemples

$\phi \triangleright m$: l'attaquant calcule m à partir de ϕ.

Pas de télépathie : on ne peut pas deviner une clef K non utilisée.

$$\phi \not\triangleright K \quad \| \quad K \not\sqsubseteq \phi$$

Axiomes : exemples

$\phi \triangleright m$: l'attaquant calcule m à partir de ϕ.

Pas de télépathie : on ne peut pas deviner une clef K non utilisée.

$$\phi \not\triangleright K \quad \| \quad K \not\sqsubseteq \phi$$

Secret (axiome correct si chiffrement fort - IND-CCA) :
le chiffrement d'un message m n'apprend rien sur m.

$$(\phi; \{m\}^r_{pk(A)}) \triangleright m \quad \rightarrow \quad \phi \triangleright m \quad \| \quad sk(A) \not\sqsubseteq (\phi; m)$$

Axiomes : exemples

$\phi \triangleright m$: l'attaquant calcule m à partir de ϕ.

Pas de télépathie : on ne peut pas deviner une clef K non utilisée.

$$\phi \not\triangleright K \quad \| \quad K \not\sqsubseteq \phi$$

Secret (axiome correct si chiffrement fort - IND-CCA) :
le chiffrement d'un message m n'apprend rien sur m.

$$(\phi; \{m\}^r_{pk(A)}) \triangleright m \quad \rightarrow \quad \phi \triangleright m \quad \| \quad sk(A) \not\sqsubseteq (\phi; m)$$

Axiome sur l'implémentation :
il est impossible de confondre une clef avec une concaténation.

$$\langle a, K \rangle \neq K'$$

Axiomes : exemples

$\phi \triangleright m$: l'attaquant calcule m à partir de ϕ.

Pas de télépathie : on ne peut pas deviner une clef K non utilisée.

$$\phi \not\triangleright K \quad \| \quad K \not\sqsubseteq \phi$$

Secret (axiome correct si chiffrement fort - IND-CCA) :
le chiffrement d'un message m n'apprend rien sur m.

$$(\phi; \{m\}^r_{pk(A)}) \triangleright m \quad \rightarrow \quad \phi \triangleright m \quad \| \quad sk(A) \not\sqsubseteq (\phi; m)$$

Axiome sur l'implémentation :
il est impossible de confondre une clef avec une concaténation.

$$\langle a, K \rangle \neq K'$$

Sûreté du protocole = incohérence du modèle

Automatisation (SCARY) : thèse de Guillaume Scerri

De nombreuses pistes encore à explorer (1/3)

Vote électronique

- propriétés d'équivalence
- primitives non standards :
  - chiffrement homomorphe
  - preuves à connaissance nulle
- pas de confiance en les autorités

De nombreuses pistes encore à explorer (1/3)

Vote électronique

- propriétés d'équivalence
- primitives non standards :
  - chiffrement homomorphe
  - preuves à connaissance nulle
- pas de confiance en les autorités
- ni en son propre ordinateur (virus, ...)

?

De nombreuses pistes encore à explorer (2/3)

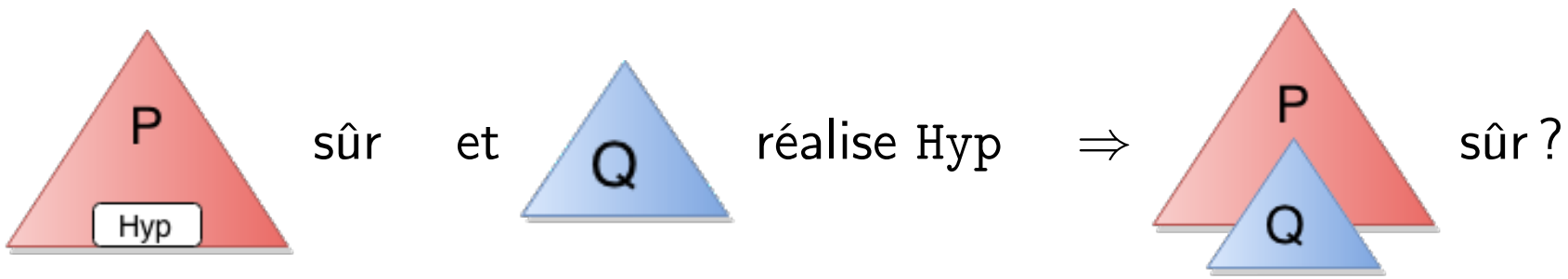
Raffinement sûr

- Qu'est-ce qu'un bon canal sûr ?
  Exemple : TLS
- Qu'est-ce qu'un bon canal authentifié ?
  Exemple : authentification par mot de passe

De nombreuses pistes encore à explorer (2/3)

Raffinement sûr

- Qu'est-ce qu'un bon canal sûr ?
  Exemple : TLS
- Qu'est-ce qu'un bon canal authentifié ?
  Exemple : authentification par mot de passe

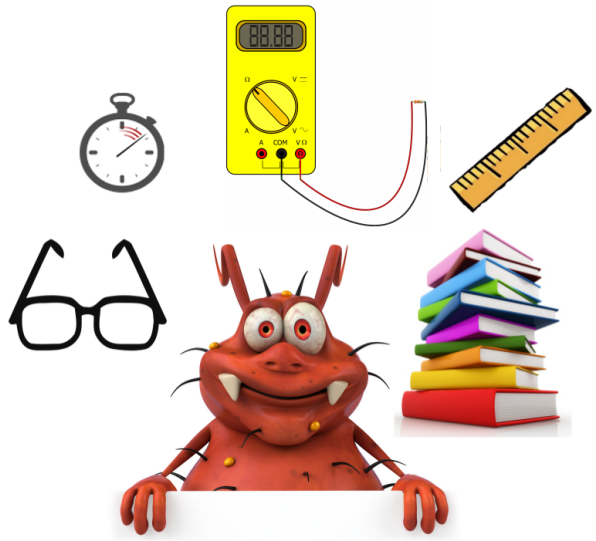


→ Crucial pour une conception modulaire

De nombreuses pistes encore à explorer (3/3)

Attaques par canaux cachés

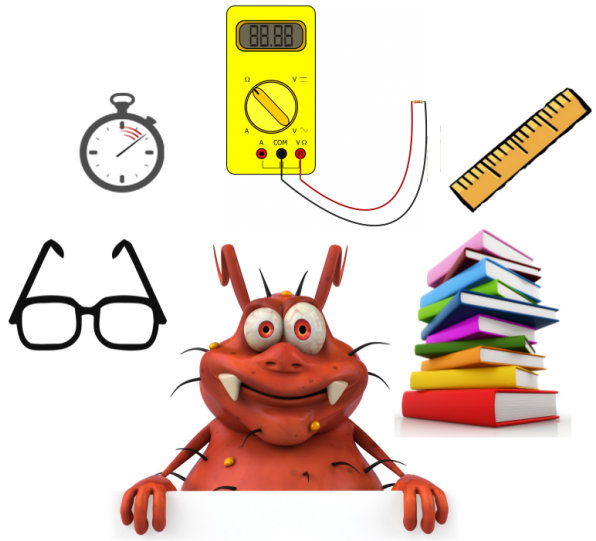
- étude de l'implémentation
- objectif "temps constant"
- pas de branchement sur des valeurs secrètes (e.g. if $k = 1$ then ...)



De nombreuses pistes encore à explorer (3/3)

Attaques par canaux cachés

- étude de l'implémentation
- objectif "temps constant"
- pas de branchement sur des valeurs secrètes (e.g. if $k = 1$ then ...)

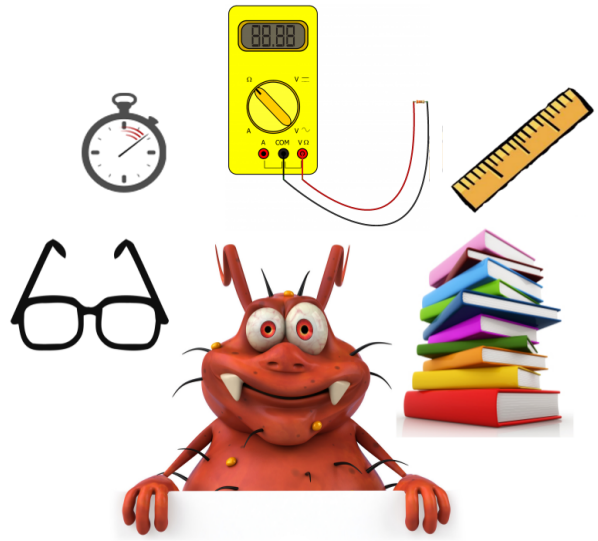


$\rightarrow$ Un combat contre le compilateur et le processeur qui cherchent à optimiser (mise en cache, prédiction, etc.) !

De nombreuses pistes encore à explorer (3/3)

Attaques par canaux cachés

- étude de l'implémentation
- objectif "temps constant"
- pas de branchement sur des valeurs secrètes (e.g. if $k = 1$ then ...)



$\rightarrow$ Un combat contre le compilateur et le processeur qui cherchent à optimiser (mise en cache, prédiction, etc.) !

Modèles plus précis

- Preuves des primitives cryptographiques (CryptoVerif, EasyCrypt, ...)
- Études des implémentations (F^*, ...)

À retenir

- Les protocoles sont difficiles à concevoir
- ... et tout autant à analyser !

À retenir

- Les protocoles sont difficiles à concevoir
- ... et tout autant à analyser !
- Un très beau terrain de jeux pour la logique et la vérification
- avec de nombreux outils efficaces :
  ProVerif, Avispa, Scyther, ...



Quelques références bibliographiques (1/5)

Analyse symbolique de protocoles de sécurité

Deux ouvrages qui présentent les techniques les plus classiques :

- *Formal Models and Techniques for Analyzing Security Protocols*. Véronique Cortier and Steve Kremer, editors. Cryptology and Information Security Series 5, IOS Press, 2011.
- *Formal Models and Techniques for Analyzing Security Protocols : A Tutorial* (notes de cours). Véronique Cortier and Steve Kremer. Foundations and Trends in Programming Languages, 2014.

Sur les protocoles en général

- B. Schneier. *Applied Cryptography Second Edition : protocols, algorithms, and source code in C*, J. Wiley & Sons, Inc. publisher, 1996.
- A. J. Menezes and P. C. van Oorschot and S. A. Vanstone. *Handbook of applied cryptography*, CRC Press publisher, 1997.

Quelques références bibliographiques (2/5)

Attaque FREAK

- *A Messy State of the Union : Taming the Composite State Machines of TLS*. Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cedric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, Jean Karim Zinzindohoue. IEEE Symposium on Security and Privacy S&P 2015

API de sécurité

- *Attacking and Fixing PKCS#11 Security Tokens*. M. Bortolozzo, M. Centenaro, R. Focardi and G. Steel. CCS 2010.
- *A Generic Security API for Symmetric Key Management on Cryptographic Devices*. Véronique Cortier and Graham Steel. Information and Computation,, 2014.
- *Automated Analysis of Security Protocols with Global State*. Steve Kremer and Robert Künnemann. IEEE Symposium on Security and Privacy S&P 2014

Quelques références bibliographiques (3/5)

Procedures basées sur les clauses de Horn, nombre non borné de sessions

- B. Blanchet. *An Efficient Cryptographic Protocol Verifier Based on Prolog Rules*. CSFW 2001.
- H. Comon, V. Cortier. *Tree automata with one memory set constraints and cryptographic protocols*. Theoretical Computer Science 2005.

Résolution de contraintes, nombre borné de sessions

- J. K. Millen, V. Shmatikov. *Constraint solving for bounded-process cryptographic protocol analysis*. ACM Conference on Computer and Communications Security 2001.
- H. Comon-Lundh, V. Shmatikov. *Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or*. LICS 2003
- Michaël Rusinowitch, Mathieu Turuani. *Protocol Insecurity with Finite Number of Sessions is NP-Complete*. CSFW 2001.

Quelques références bibliographiques (4/5)

pi-calcul appliqué

- *Mobile values, new names, and secure communication*. Martín Abadi, Cédric Fournet. POPL 2001

Procédures pour l'équivalence, nombre borné de sessions

- *Trace Equivalence Decision : Negative Tests and Non-determinism*. Vincent Cheval, Hubert Comon-Lundh and Stéphanie Delaune. CCS 2011.
- *Automated verification of equivalence properties of cryptographic protocols*. Rohit Chadha, Stefan Ciobaca, and Steve Kremer. ESOP 2012.
- *Lengths may break privacy - or how to check for equivalences with length*. Vincent Cheval, Véronique Cortier, and Antoine Plet. CAV 2013.

Procédures pour l'équivalence, nombre non borné de sessions

- *Automated verification of selected equivalences for security* protocols. Bruno Blanchet, Martín Abadi, Cédric Fournet. J. Log. Algebr. Program. 2008
- *From security protocols to pushdown automata*. Rémy Chrétien, Véronique Cortier, and Stéphanie Delaune. ICALP 2013.
- *Typing messages for free in security protocols : the case of equivalence properties*. Rémy Chrétien, Véronique Cortier, and Stéphanie Delaune. CONCUR 2014.

Quelques références bibliographiques (5/5)

Information flow (très incomplet !)

- *Security policies and security models*, Joseph A. Goguen and José Meseguer. i IEEE Symp. on Security and Privacy S&P 1982.
- *Language-Based Information-Flow Security*. Andrei Sabelfeld and Andrew C. Myers. IEEE Journal on Selected Areas in Communications, 2003.

Modèle intrus permissif

- *Towards Unconditional Soundness : Computationally Complete Symbolic Attacker*. Gergei Bana and Hubert Comon-Lundh. POST 2012.
- *A Computationally Complete Symbolic Attacker for Equivalence Properties*. Gergei Bana and Hubert Comon-Lundh. CCS 2014.
- *Tractable inference systems : an extension with a deducibility predicate*. Hubert Comon-Lundh, Véronique Cortier, and Guillaume Scerri. CADE 2013.