

Voiture connectée...

Charlie Miller et Chris Valasek (BlackHat2015) :
prise de contrôle à distance d'une Jeep



TOUTE L'ACTUALITÉ / SÉCURITÉ
Chrysler rappelle 1,4 million de voitures exposées à un piratage à distance
, publié le 27 Juillet 2015

- Cause : de nombreux services non sécurisés en écoute sur internet
- Combien de systèmes (pacemakers, voitures, avions, usines., réseaux de distribution..) reposent de manière critique sur du logiciel pour fonctionner ?

MS17-010 : la menace des rançongiciels [O. Levillain]

En mai et juin 2017



- Deux attaques très médiatisées par des rançongiciels
 - exploitation d'une vulnérabilité critique dans Windows
 - ... sur un service qui ne devrait pas être exposé
 - ... pour lequel un correctif est disponible depuis mars
- Pourquoi la sécurité semble-t-elle un échec?

Cours 5MMSIC – WMMBESIC

Sécurité Informatique et confidentialité

Jean-Louis ROCH

- Objectif: un aperçu général et des approfondissements techniques sur les éléments principaux qui définissent la sécurité d'un système informatique, avec un accent particulier sur les certificats et la dimension intégrité et confidentialité des données.
- Documents de référence: documents de l'ANSSI (tenus à jour)
 - Transparents : **Contenu pédagogique CyberEdu**
 - <https://www.ssi.gouv.fr/administration/formations/cyberedu/contenu-pedagogique-cyberedu/>
 - **Guides « Bonnes pratiques »** <https://www.ssi.gouv.fr/particulier/bonnes-pratiques/>
 - « **Renforcer la sécurité de son système d'information en 42 mesures** »
https://www.ssi.gouv.fr/uploads/2017/01/guide_hygiene_informatique_anssi.pdf
 - **Référentiel Général de Sécurité (RGS)** publié le 1^{er} juillet 2014.

Plan du cours

http://chamilo.grenoble-inp.fr/main/course_home/course_home.php?cidReq=ENSIMAGWMMBESIC

Calcul note = 50% Examen + 50% MAX (Devoir, Examen)

1. Principes de base sur la sécurité des systèmes . [Jean-Louis Roch]
2. Primitives cryptographiques pour la Confidentialité, l'Authentification, l'Intégrité, la Non-répudiation (CAIN) [Jean-Louis Roch]
 - TP non surveillé : ssh et certificats [Jean-Louis Roch]
3. Présentation du devoir: Analyse d'une faille et déploiement d'une machine virtuelle - Docker et Vagrant [Sébastien Viardot]
 - TP contrôle continu (50% de la note) : Sujet (TP1_Docker_Vagrant.tgz) [Sébastien Viardot]
4. Certificats. Architectures de confiance [Jean-Louis Roch]
5. Blockchain [François Launay, IBM Systems hardware France – IBM Z]
Blockchain vs PKI [Jean-Louis Roch]
6. RGPD -- Emails et fuites d'information : comment se protéger [Cédric Lauradoux]
7. Confiance dans son système d'information: bonnes pratiques . [Jean-Louis Roch]
8. Délégations de calcul, preuves interactives et certificats de résultats [Jean-Louis Roch]

Ce qu'on a vu aujourd'hui

- 1. Les enjeux de la sécurité des S.I.**
- 2. Les besoins de sécurité : critères DICP**
- 3. Notions de vulnérabilité, menace, attaque**
- 4. Panorama de quelques menaces**
- 5. Le droit des T.I.C. et l'organisation de la sécurité en France**

Exemple: Guide sur la sécurité des mots de passe (ANSSI 25/05/2012)

- <https://www.ssi.gouv.fr/particulier/guide/mot-de-passe/>
- La robustesse d'un mot de passe dépend en général d'abord de sa complexité, mais également de divers autres paramètres, expliqués en détail dans le document « Recommandations de sécurité relatives aux mots de passe ».
- Si vous souhaitez une règle simple : choisissez des mots de passe d'au moins 12 caractères de types différents (majuscules, minuscules, chiffres, caractères spéciaux).
Deux méthodes pour choisir vos mots de passe :
 - La méthode phonétique : « J'ai acheté huit cd pour cent euros cet après-midi » deviendra ght8CD%E7am ;
 - La méthode des premières lettres : la citation « un tien vaut mieux que deux tu l'auras » donnera 1tvmQ2tl'A.

Vulnérabilités : <https://cve.mitre.org/>

https://cve.mitre.org

Rechercher

Search CVE List | Download CVE | Update an ID | Request a CVE ID | Data Feed

Follow CVE  

Common Vulnerabilities and Exposures
The Standard for Information Security Vulnerability Names

Home | CVE IDs | About CVE | CVE in Use | Community & Partners | Blog | News | Site Search

TOTAL CVE IDs: 91302

Become a CNA
[Click for process, documentation & more](#)

Request a CVE ID
[Click for CNAs, MITRE request form, guidelines, & more](#)

Update info in a CVE ID
[Click for MITRE request form, guidelines & more](#)

CVE List downloads
Available in [xml](#), [CVRF](#), [txt](#), & comma-separated

CVE content data feed
Available via [CVEnew Twitter Feed](#)

CNA Participation Growing Worldwide



Latest CVE News

- ♦ [ASUSTOR Added as CVE Numbering Authority \(CNA\)](#)
- ♦ [CVE Replaces "CVE Compatibility Program" with CVE Compatibility Guidance Document](#)
- ♦ [Minutes from CVE Board Teleconference Meeting on September 20 Now Available](#)
- ♦ [Forcepoint Added as CVE Numbering Authority \(CNA\)](#)

[More >>](#)

CVE Blog

Become a CVE Numbering Authority

[CVE Numbering Authorities](#), or "CNAs," are how the [CVE List](#) is built. Every [CVE ID](#) added to the list is assigned by a CNA.

The majority of CNAs are currently software vendors that assign CVE IDs to issues in their own products, but many vulnerability researchers and third-party coordinators also participate by assigning CVE IDs to issues in third-party products per their specified scopes of coverage.

As of today, there are [77 total CNAs](#) participating in the CVE program from around the world with [14 countries](#) now represented.

Please consider [joining us](#) as a CNA ...

New CVE IDs

Tweets by @CVEnew

 **CVE** @CVEnew
CVE-2017-15185 plugins/ogg.c in Libmp3split 0.9.2 calls the libvorbis vorbis_block_clear function with ... [bit.ly/2y4ZQV](#)
[Love](#) [Reply](#) [2h](#)

 **CVE** @CVEnew
CVE-2017-14973 IDentidcard Two-Reader Controller Configuration Manager 1.18.8 (396) is vulnerable to Stored ... [bit.ly/2wHZxL](#)
[Love](#) [Reply](#) [2h](#)

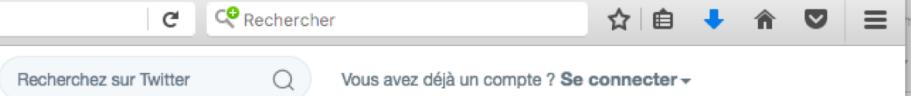
 **CVE** @CVEnew
CVE-2017-14972 InFocus Mondopad 2.2.08 is vulnerable to authentication bypass when accessing uploaded files by ... [bit.ly/2fX9pXu](#)
[Love](#) [Reply](#) [2h](#)

 **CVE** @CVEnew
CVE-2017-14973 IDentidcard Two-Reader Controller Configuration Manager 1.18.8 (396) is vulnerable to Stored ... [bit.ly/2wHZxL](#)
[Love](#) [Reply](#) [2h](#)

CVE tweets

Twitter, Inc. (US) | https://twitter.com/CVEnew/

Accueil À propos Rechercher sur Twitter Vous avez déjà un compte ? Se connecter



Tweets 9 352 Abo 2 Abo 3 873 Suivre

CVE
@CVEnew
Account maintained by the CVE Team to update the community on new CVE IDs. Send CVE questions via the form at cveform.mitre.org.
Inscrit en janvier 2017

Tweets **Tweets & réponses**

CVE @CVEnew · 2 h
CVE-2017-15185 plugins/ogg.c in Libmp3split 0.9.2 calls the libvorbis vorbis_block_clear function with ... bit.ly/2y42ZQV

CVE @CVEnew · 2 h
CVE-2017-14973 IDentocard Two-Reader Controller Configuration Manager 1.18.8 (396) is vulnerable to Stored ... bit.ly/2wHZ2xL

CVE @CVEnew · 2 h
CVE-2017-14972 InFocus Mondopad 2.2.08 is vulnerable to authentication bypass when accessing uploaded files by ... bit.ly/2fX9pXu

CVE @CVEnew · 2 h
CVE-2017-14971 Infocus Mondopad 2.2.08 is vulnerable to a Hashed Credential Disclosure vulnerability. The ... bit.ly/2y3yOti

CVE @CVEnew · 6 oct.
CVE-2015-2673 The ec_ajax_update_option and ec_ajax_clear_all_taxrates functions in ... bit.ly/2fRhXIE

Nouveau sur Twitter ?
Inscrivez-vous maintenant pour obtenir votre fil d'actualités personnalisé !

S'inscrire

Tendances : Monde

- #Estamos Unidos Mexicanos 160 k Tweets
- #FelizLunes 7 884 Tweets
- #HIAC 240 k Tweets
- #EyAmerica 7 401 Tweets
- #MasterChefMx 41,9 k Tweets
- صلاح الخبر 17,4 k Tweets
- Delhi-NCR 3 235 Tweets
- Los Ángeles Azules 6 032 Tweets

CVE-ID	
CVE-2017-15185 Learn more at National Vulnerability Database (NVD) <ul style="list-style-type: none">• Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings	
Description	
plugins/ogg.c in Libmp3splt 0.9.2 calls the libvorbis vorbis_block_clear function with uninitialized data upon detection of invalid input, which allows remote attackers to cause a denial of service (application crash) via a crafted file.	
References	
<p>Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.</p> <ul style="list-style-type: none">• EXPLOIT-DB:42399• URL:https://www.exploit-db.com/exploits/42399/• MISC:http://seclists.org/fulldisclosure/2017/Jul/82• MISC:https://anonscm.debian.org/cgit/users/ron/mp3splt.git/commit/?id=18f018cd774cb931116ce06a520dc0c5f9443932• MISC:https://lists.debian.org/debian-lts/2017/09/msg00115.html	
Assigning CNA	
MITRE Corporation	
Date Entry Created	
20171008	Disclaimer: The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20171008)	

<https://ensimag.skillsnetwork.site>

s [YT](#) Convertisseur You... [Scopia Desktop](#)

Get up to \$1200 value in free IBM Cloud services when taking our courses!

Grenoble INP ENSIMAG

Register Sign in

ENSIMAG - IBM Learning Portal

Practical skills from two technology leaders!



Get up to \$1200 value in free IBM Cloud services when taking our courses!

Gre
ENS



LEARNING PATHS

COURSES

COMPETITIONS

EVENTS

CODE PATTERNS



Jean-Louis Roch

Please pick a specialization below to get started.



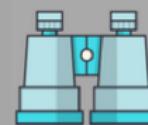
Data Science Essentials



Applied Data Science with Python



Data Engineer Essentials



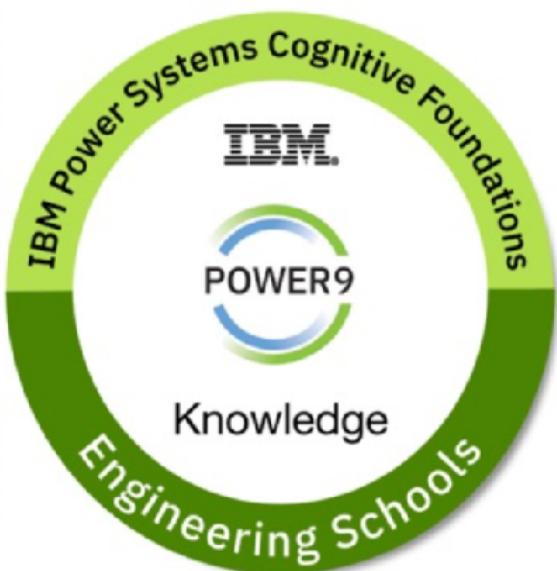
Machine Learning Essentials



Developing bots



Lightbend Reactive Architecture:
Foundations



Type: Validation

Level: Foundational

Time: Hours

IBM Power Systems Cognitive for Students

Issued By [IBM](#)

This badge attests that the holder has a good foundational understanding of Artificial Intelligence, Machine Learning, Deep Learning and Power AI & AI Vision.

Demonstrated Skills

AI VISION

Artificial Intelligence

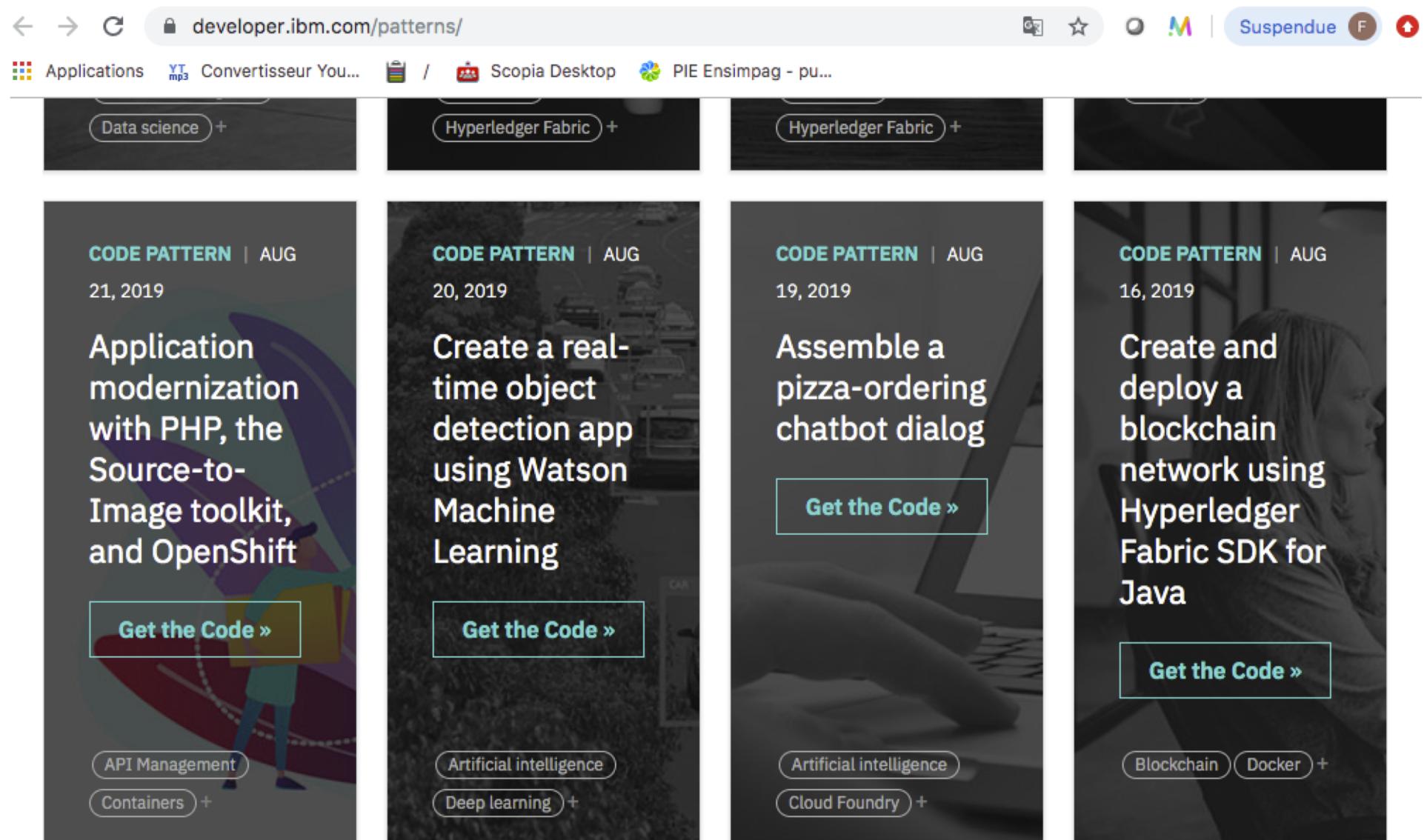
Machine Learning

PowerAI

Earning Criteria

- ❑ This badge is intended for students of Engineering Schools with practice on PowerAI, PowerAI Vision & Deep Learning and who have an understanding of the benefits that Power brings to artificial Intelligence.
- ❑ Attend a one day in-person session at an IBM facility hosting sessions in topics related to Machine Learning, Artificial Intelligence, Power AI & AI Vision.
- ❑ Pass the end-of-session quiz.

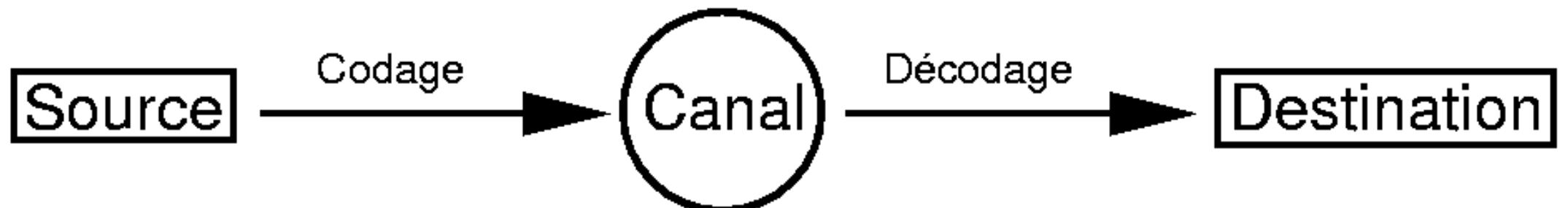
<https://developer.ibm.com/patterns/>



The screenshot shows a web browser window displaying the IBM Developer Patterns page. The URL in the address bar is <https://developer.ibm.com/patterns/>. The page features a grid of four code pattern cards:

- CODE PATTERN | AUG 21, 2019**
Application modernization with PHP, the Source-to-Image toolkit, and OpenShift
[Get the Code »](#)
API Management, Containers
- CODE PATTERN | AUG 20, 2019**
Create a real-time object detection app using Watson Machine Learning
[Get the Code »](#)
Artificial intelligence, Deep learning
- CODE PATTERN | AUG 19, 2019**
Assemble a pizza-ordering chatbot dialog
[Get the Code »](#)
Artificial intelligence, Cloud Foundry
- CODE PATTERN | AUG 16, 2019**
Create and deploy a blockchain network using Hyperledger Fabric SDK for Java
[Get the Code »](#)
Blockchain, Docker

Notion de code



- Le code doit répondre à différents critères :
 - Rentabilité : compression des données.
 - Sécurité de l'information : cryptage, authentification, etc.
 - Tolérance aux fautes : correction/détection d'erreurs.

Fondation: théorèmes de Shannon 1948

- **Compression** : « Pour toute source X d' entropie $H(x)$ on peut trouver un code dont la longueur moyenne s' approche de $H(X)$ d' aussi prêt que l' on veut »
 - Algorithme d' Huffman, extensions de sources
- **Correction d' erreurs** : « Pour tout canal on peut toujours trouver une famille de codes dont la probabilité d' erreur après décodage tend vers 0 »
- **Cryptage** : « si un chiffrement est parfait, alors il y a au moins autant de clefs possibles que de messages »
 - Un cryptanalyste doit obtenir au moins $H(M)$ informations pour retrouver M
 - version moderne: indistinguishable

À quoi sert la cryptographie (CAIN) ?

- Confidentialité des informations stockées ou manipulées
 - Seuls les utilisateurs autorisés peuvent accéder à l' information
- Authentification des utilisateurs
 - L' utilisateur est-il ou non autorisé ? Pour quelle action ?
- Intégrité des informations stockées ou manipulées
 - Contre l' altération des données
- Non-répudiation des informations
 - Empêcher un utilisateur de se dédire

CONFIDENTIALITE

Cryptographie

- Information chiffrée

Connaissance de l' existence de l' information

\neq

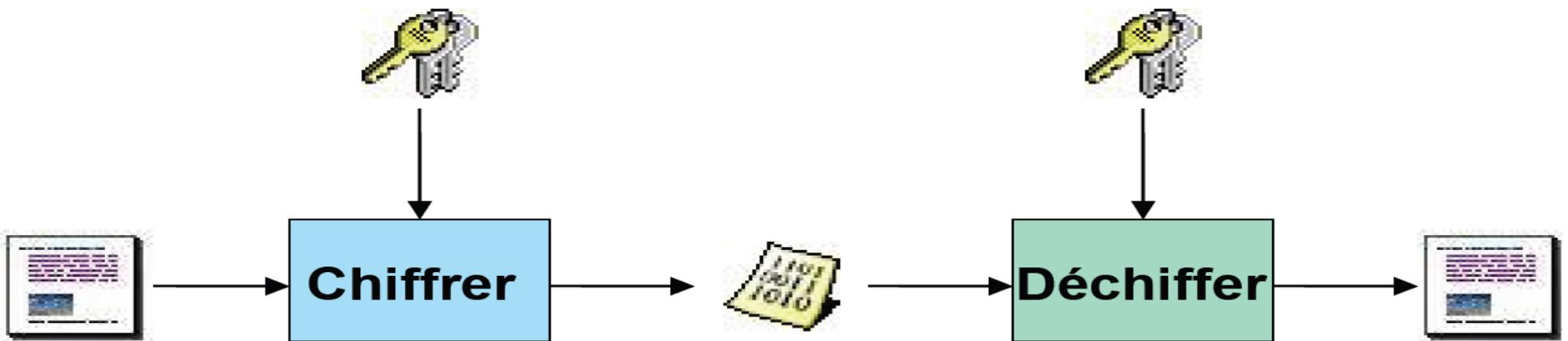
Connaissance de l' information

- Objectif

- Permettre à **Alice** et **Bob** de communiquer sur un canal peu sûr
 - Réseau informatique, téléphonique, etc.
- **Oscar** ne doit pas comprendre ce qui est échangé



Algorithmes de cryptographie



- Propriétés théoriques nécessaires :

1. Confusion

Aucune propriété statistique ne peut être déduite du message chiffré

2. Diffusion

Toute modification du message en clair se traduit par une modification complète du chiffré

Cryptographie moderne

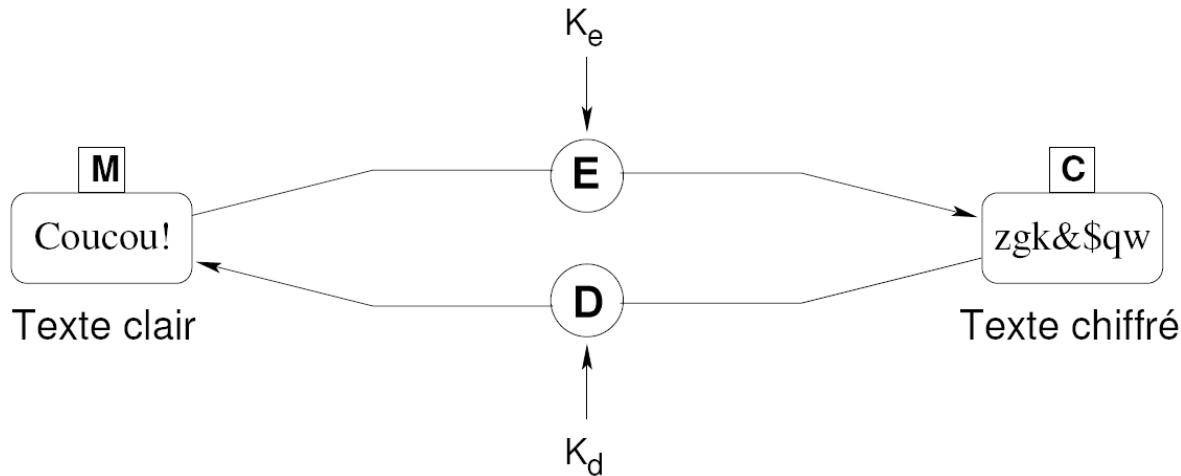
- Principes de Auguste Kerckhoffs (1883)
 1. La sécurité repose sur le secret de la clef et non sur le secret de l' algorithme
 - Canal +, Cartes Bleues, PS4 !!!
 2. Le déchiffrement sans la clef doit être impossible Trouver la clef à partir du clair et du chiffré est impossible
(à l' échelle humaine)
- Cœur: **fonction à sens unique**
 - "impossible" à inverser (sauf si on a la « clef »)

=> **Fonction à sens unique** (paramétrée par une clef)

Terminologie

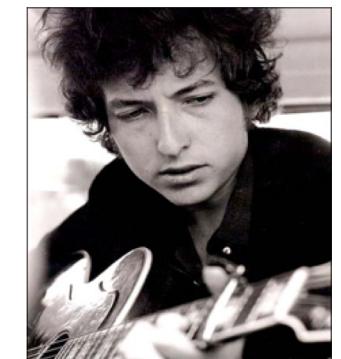
- Texte clair
 - information qu' Alice souhaite transmettre à Bob
- Chiffrement
 - processus de transformation d'un message M de telle manière à le rendre incompréhensible
 - Fonction de chiffrement E
 - Génération d'un chiffre (message chiffré) $C = E(M)$
- Déchiffrement
 - processus de reconstruction du message clair à partir du message chiffré
 - Fonction de déchiffrement D
 - $D(C) = D(E(M)) = M$ (E est injective et D surjective)

Types de cryptographie



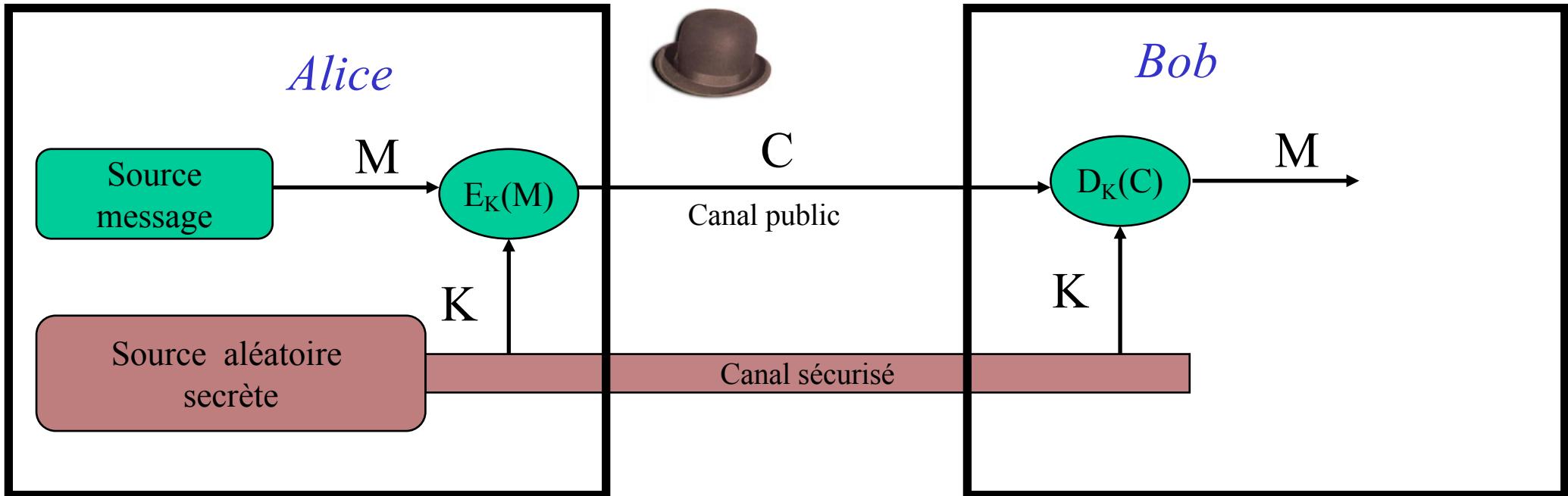
- En pratique E et D sont paramétrées par des clefs K_d et K_e
- Deux grandes catégories de systèmes cryptographiques
 - Systèmes à clefs secrètes (symétriques) : $K_e = K_d = K$
 - Systèmes à clefs publiques (asymétriques) : $K_e \neq K_d$
- Deux types de fonctionnement
 - Par flot : chaque nouveau bit est manipulé directement
 - Par bloc : chaque message est découpé en blocs

Cryptographie symétrique



Chiffrement symétrique

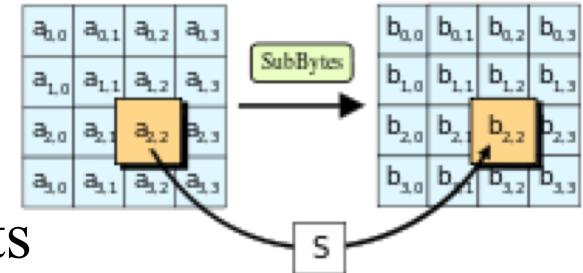
- Définition : chiffrement **parfait** ssi l'attaquant n'a **aucune information** sur M



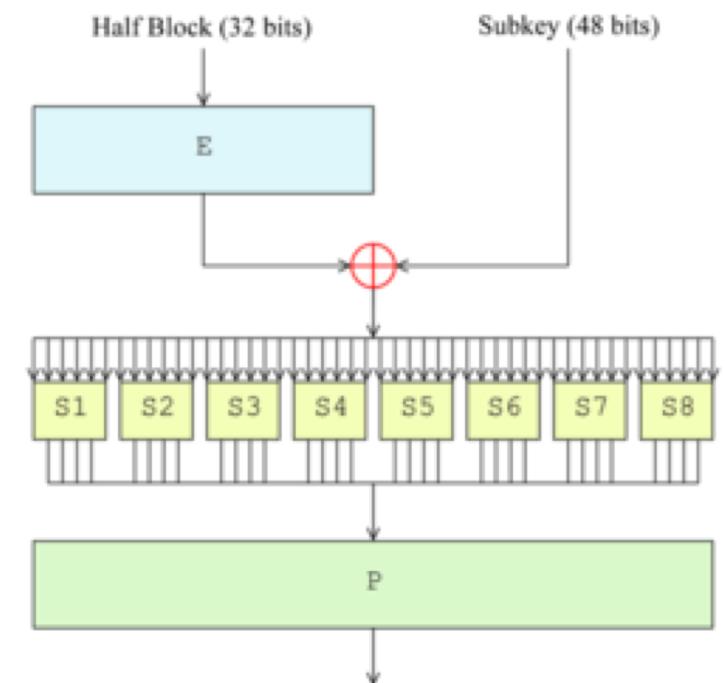
- Définition : le chiffrement est parfait ssi : $\forall C$ chiffré $\forall M_1 \neq M_2$ deux clairs :
 $\text{Prob}_K [E_K(M_1) = C] = \text{Prob}_K [E_K(M_2) = C]$ (*indistingualibilité*)
- Condition suffisante** : $\forall C$ chiffré $\forall M$ clair $\exists ! K$ clef : $E_K(M) = C$
- Condition nécessaire** : la clef secrète est au moins aussi longue que le clair
- Exercice: 1) exemple d'un tel chiffre ?
2) pourquoi compresser le clair avant chiffrement ?

Chiffrement à clef secrète: DES et AES

- AES : standard actuel (oct 2000)
 - Advanced Encryption Standard www.nist.gov/AES
 - Corps à 256 éléments : F256
 - Taille paramétrable de clefs: 128, 192 ou 256 bits



- DES [1972, IBM] Data Encryption Standard
 - Exemple : Crypt unix
 - Principe : chiffrement par bloc de 64 bits
 - clef 64 bits \Rightarrow 56 bits utiles + 16 transformations/rondes
 - Chaînage entre 2 blocs consécutifs
 - Attaque brutale : $2^{56} = 64.10^{15}$
 - $1000 \text{ PCs} * 10^9 \text{ Op/s} * 64000\text{s} = 20\text{h} \quad !!!!$
 - Contre mesures: augmenter l'entropie de la clef
 - Double DES (????), Triple DES,...
 - « Salt »: exemple Unix crypt
 - 24 bits si bit $i=1$, permutation bit i et bit $i+24$)

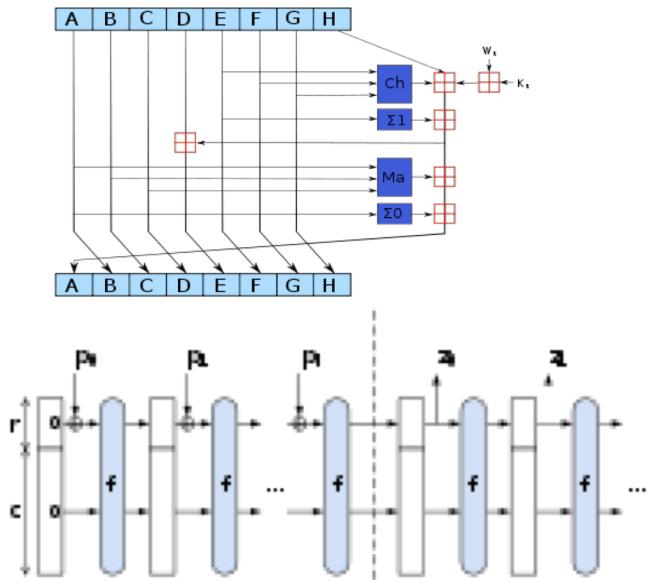


Application : stockage des mots de passe

- Principe: seul l'utilisateur devrait connaître son mot de passe
 - Pourquoi ?
- Pour chaque utilisateur, stockage de l'empreinte (hash) du mot de passe :
 - Salage: suffixe ajouté au mot de passe (dynamique)
 - Stockage d'une table des mots de passe encodés :
(identifiant ; salage ; hachage(mot de passe || salage))
- Exercice:
 - Quelle résistance aux attaques forces brutes d'un attaquant qui aurait obtenu la table?

Fonction de Hachage

- Fonction h : $h(M) = \text{résumé du message } M \text{ sur un nombre fini de bits}$ (exemple 256 bits)
- Hachage cryptographique (« empreinte »)
 - résiste aux collisions: il doit être calculatoirement impossible de trouver 2 messages $M \neq M'$ tel que $h(M)=h(M')$
- SHA : Secure Hash Algorithm
 - SHA-0, SHA-1 (MD5)
 - SHA-2 [2001] : tailles blocs 256 ou 512 bits
 - SHA-3 [2012] (Keccak)



Fonction à sens unique ? exemple: «Logarithme discret »

- $(G, *)$: un groupe fini cyclique d'ordre n ; g un générateur de G
 - $G = \{ g^i ; i = 0, \dots, n-1 \}$ Exemple ?
- **Exponentielle** : $\text{Exp} : \{ 0, \dots, n-1 \} \rightarrow G$ définie par $\text{Exp}(i) = g^i$

Coût du calcul de $\text{Exp}(i)$ = $O(\log(i)) = O(\log |G|)$ [algorithme?]

Exemple : $5^{11} [7] = ((5^2)^2 5)^2 5 = ((4)^2 5)^2 5 = (2.5)^2 5 = 2.5 = 3$

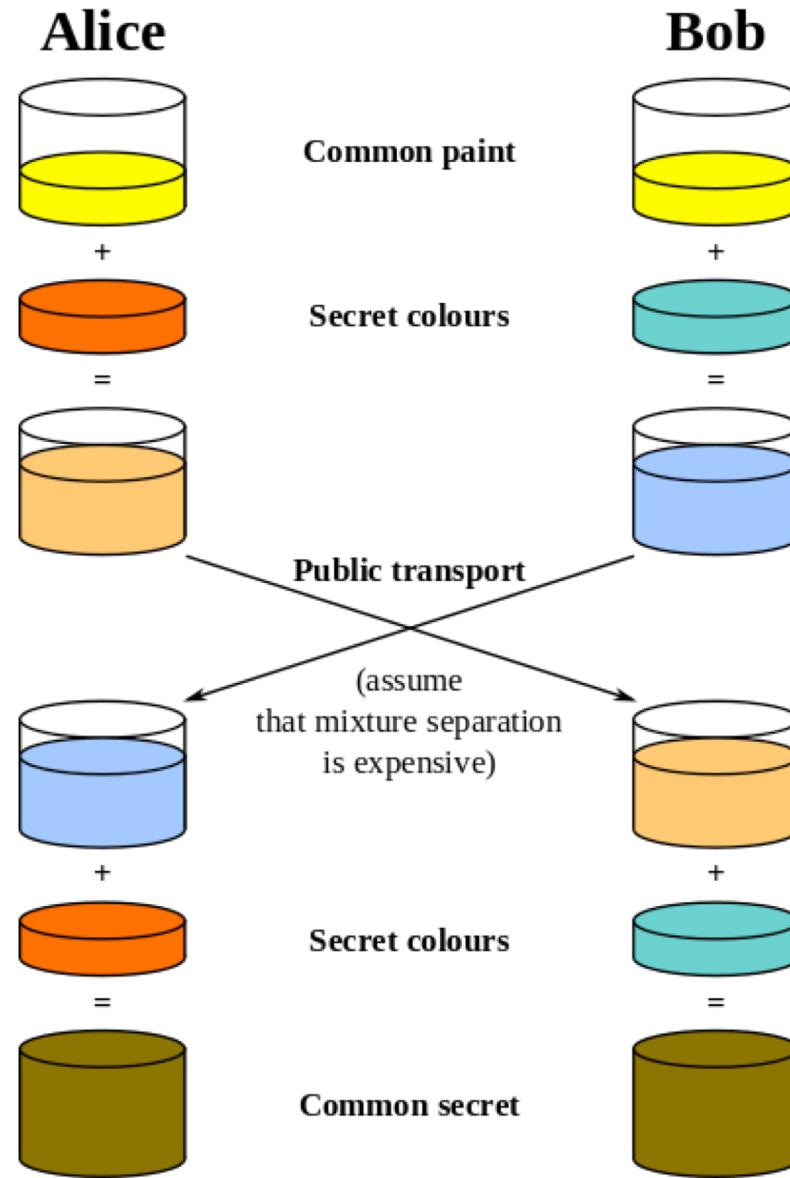
- **Logarithme** : $\text{Log} : G \rightarrow \{ 0, \dots, n-1 \}$ définie par $\text{Log}(x) = i$ avec $x = g^i$
Exemple : trouver x / $6^x = 8 [11]$ ($\angle = x : \text{équation}$)
- **Coût du calcul de $\text{Log}(i)$ = $\Omega(n^{0.5})$** (pour G abstrait) [Shanks] [algorithme?]
 - Sur $G = \mathbb{Z}_p^*$: prouvé « polynomialement plus coûteux » que la factorisation d'entiers`
- **Conjecturé difficile pour certains groupes** :
 - Très utilisé en cryptographie asymétrique : ex El Gamal, ECDLP
 - Mais **attention** : certaines instances faciles
 - Question : Existe-t-il une fonction à sens unique "programmable" ?

Partage de clef secrète

- De Diffie-Hellman à Internet Key Exchange (IKE)
 - Diffie Hellman Station-To-Station

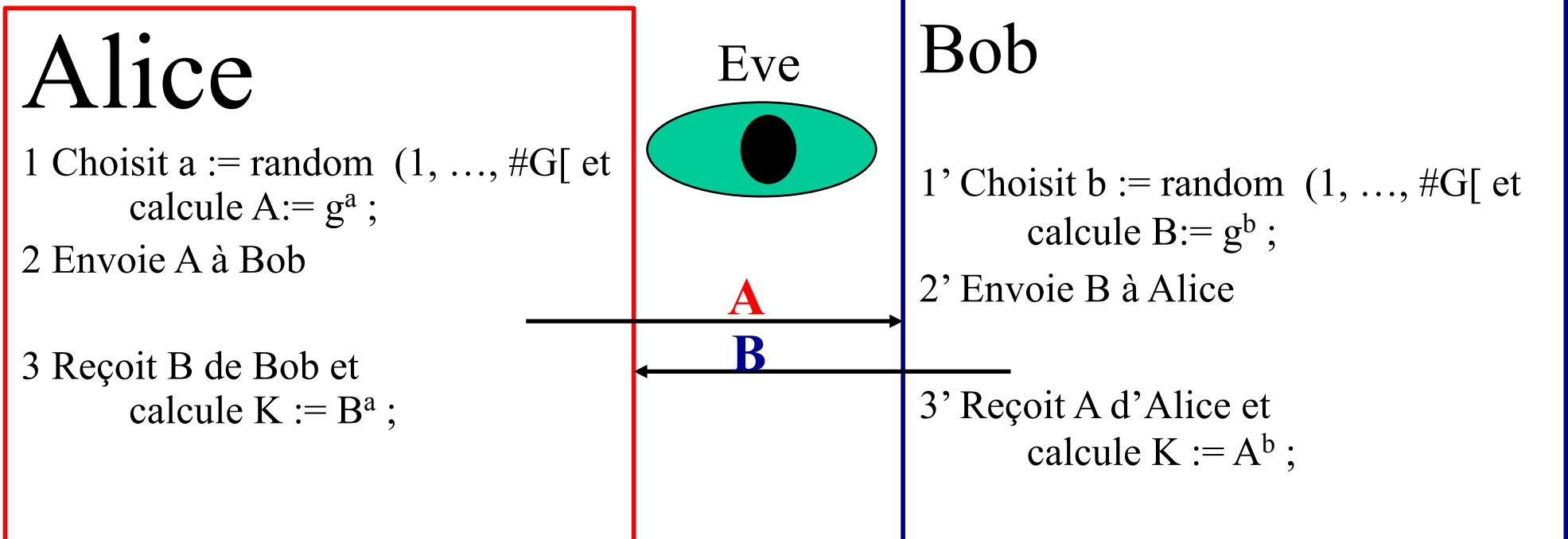
Echange de clefs « Diffie-Hellman »

[http://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Diffie-Hellman_Key_Exchange.svg/399px-Diffie-Hellman_Key_Exchange.svg.png]



Echange de clefs « Diffie-Hellman »

Paramètres publics: groupe cyclique G et un générateur g



- Sécurité: difficulté calculatoire du log discret dans des groupes cycliques de grande taille bien choisis.
- Faiblesse: Attaque «Man-in-the-middle»
 - Contre-mesure?

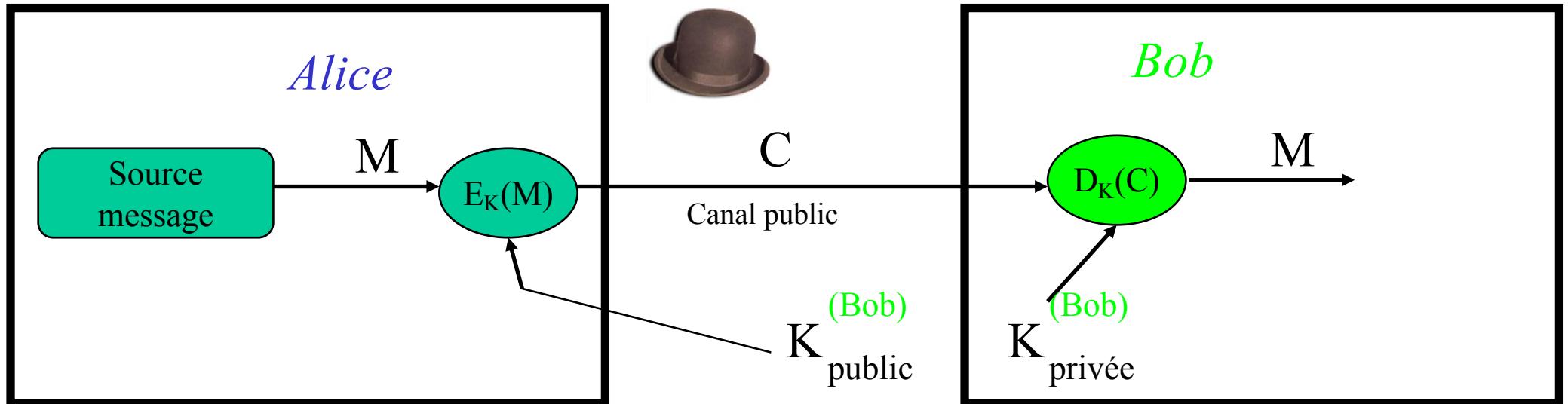
Cryptographie à clef publique (Cryptographie asymétrique)

Fonction à sens unique **chausse trappe**



Chiffrement asymétrique

- Asymétrique = boîte à lettres : C contient toute l'information sur M



- Définition : parfait si, connaissant C et la fonction E, alors il est **calculatoirement impossible** de calculer M
 - Humain: 10^{10} ordis (!!)* 10^{15} op/sec (!!)* 128 bits (!)* 3000 ans < $10^{39} = 2^{130}$ op binaires
 - Terre : 10^{50} opérations =
 - Univers : $10^{125} = 2^{420}$ opérations « physiques » élémentaires depuis le bing bang
- Les fonctions E et D = E^{-1} doivent vérifier :
 - $X = E(M)$: facile à calculer : coût ($E(M)$) = « linéaire » en la taille de M
 - $M = D(X)$: calculatoirement impossible
- Existe-t-il de telles fonctions **à sens unique chausse-trappe** ???

Un exemple de chiffrement asymétrique: Elgamal

1. Chiffrement
2. Déchiffrement
3. Cryptanalyse
4. Signature

Un groupe fini G cyclique d'ordre n :

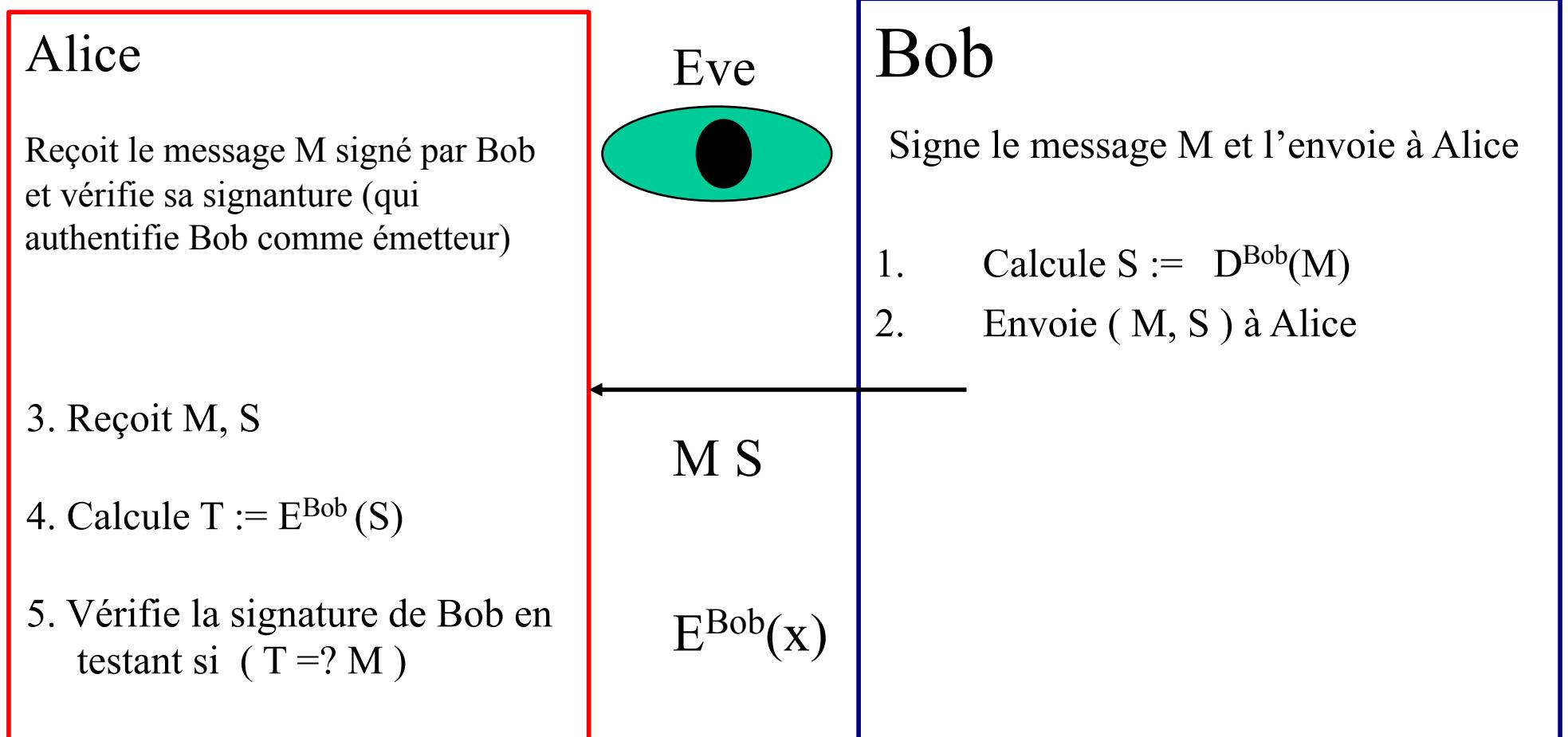
- avec n éléments et avec un élément g générateur :
$$G = \{g^i, i=0..p-1\} \quad \text{et} \quad g^n = 1_G$$
- Exemple: $G = \mathbb{Z}_p^*$
- Chaque élément de G est représentable par $\lceil \log_2 n \rceil$ bits
- Chaque message de $\lceil \log_2 n \rceil$ bits est représentable par un elt de G

Chiffrement Elgamal

- Bob: clef secrète: b random; clef publique = $B=g^b$ public (G, g, B)
 - **Question:** comment Bob calcule-t-il la clef publique B et à quel coût ?
- Chiffrement par Alice: a random; $E_A(m)=(c_1, c_2)$ avec $c_1=g^a$ et $c_2=m \cdot B^a$
 - **Questions:**
 - quelle est la longueur du chiffré?
 - pourquoi Alice choisit-elle a aléatoirement ?
- Déchiffrement par Bob: $D_A(c_1, c_2) = c_2 * c_1^{-b} = m$
- Hypothèse de sécurité ?
 - étant donné (g, g^a, g^b) , il est **calculatoirement impossible** de calculer g^{ab}
 - **Exercice:**
 - Si Eve sait casser ElGamal, sait-elle calculer le log discret?
 - Pourquoi Alice choisit-t-elle a aléatoirement ?
 - Peut-elle utiliser le même a deux fois ?

Signature par cryptographie asymétrique

[Hypothèse: $E^{Bob}D^{Bob}(M) = D^{Bob}E^{Bob}(M) = M$]

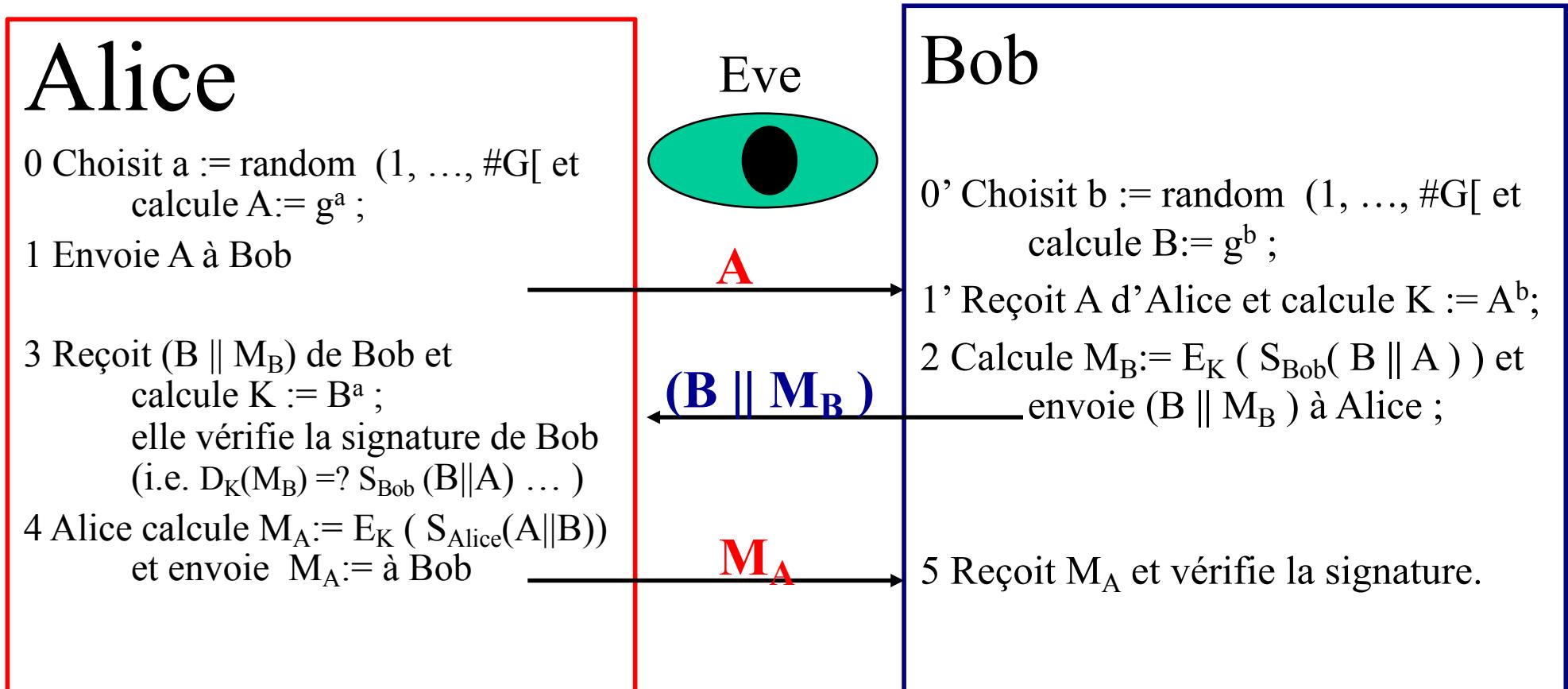


Exemple: passeport signé par une autorité (*tiers de confiance*).

Remarque: Bob peut se limiter à signer un résumé de M -- $hash(M)$ --

Exemple: Diffie-Hellman signé: Protocole STS Station-to-Station

Paramètres publics: groupe cyclique G et un générateur g
+ paires de clefs de signature asymétrique pour chaque participant



- Construction de clef commune + authentification
- Résiste attaque «Man-in-the-middle»

STS : variantes

- STS de base
 - (1) Alice → Bob : g^a
 - (2) Alice ← Bob : $g^b, E_K(S_{Bob}(g^b \parallel g^a))$
 - (3) Alice → Bob : $E_K(S_{Alice}(g^a \parallel g^b))$
- STS complet: si on ne connaît ni le groupe ni les clefs publiques, alors G, g et les clefs publiques sont transmises
(les clefs sont vérifiées via un certificat avec un tiers de confiance)
 - (1) Alice → Bob : G, g, g^a
 - (2) Alice ← Bob : $g^b, Certif_{Bob}, E_K(S_{Bob}(g^b \parallel g^a))$
 - (3) Alice → Bob : $Certif_{Alice}, E_K(S_{Alice}(g^a \parallel g^b))$
- STS restreint pour authentification :
 - (1) Alice → Bob : a
 - (2) Alice ← Bob : b, $S_{Bob}(b \parallel a)$
 - (3) Alice → Bob : $S_{Alice}(a \parallel b)$

Sécurité

- Modèles d'attaque considérés:
 1. Texte chiffré connu : seul C est connu d' Oscar
 2. Texte clair connu : Oscar a obtenu C et le M correspondant
 3. Texte clair choisi : pour tout M, Oscar peut obtenir le C
 4. Texte chiffré choisi : pour tout C, Oscar peut obtenir le M
- Garantir la confidentialité
 - Impossible de trouver M à partir de E(M)
 - Impossible de trouver la méthode de déchiffrement D à partir d' une séquence $\{M_1, \dots, M_k, E(M_1), \dots, E(M_k)\}$

Algorithmes d' attaque

1. Attaque exhaustive (par force brute)

- Énumérer toutes les valeurs possibles de clefs
- 64 bits → 2^{64} clefs = $1.844 \cdot 10^{19}$ combinaisons
 - Un milliard de combinaisons/seconde → 1 an sur 584 machines

2. Attaque par séquences connues

- Deviner la clef si une partie du message est connue
ex: en-têtes de standard de courriels

3. Attaque par séquences forcées

- Faire chiffrer par la victime un bloc dont l'attaquant connaît le contenu, puis on applique l'attaque précédente ...

4. Attaque par analyse différentielle

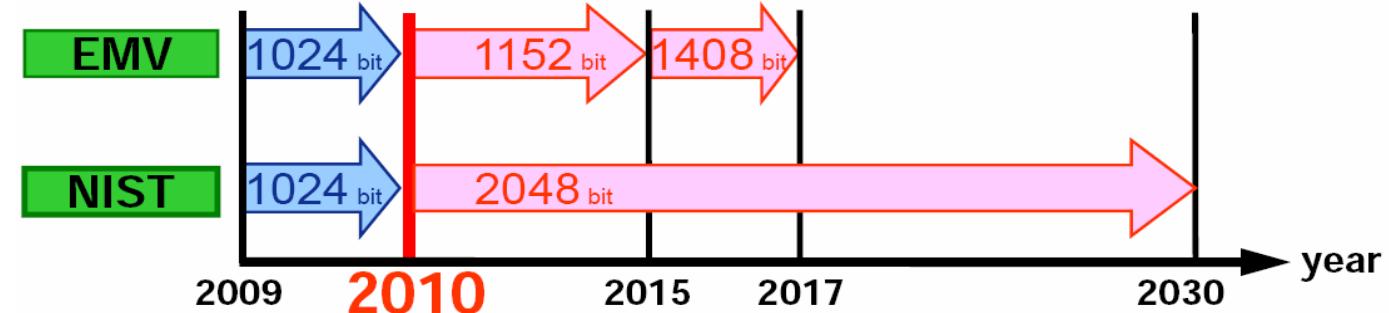
- Utiliser les faibles différences entre plusieurs messages (ex: logs) pour deviner la clef
- etc

Attaques - quelques chiffres

- La résistance d'un chiffrement dépend du nombre d'opérations requis pour le casser sans connaître le secret
- #opérations effectuées par l'univers depuis le big-bang : 10^{123}
 - Nombre particules dans l'univers : 10^{100}
- Echelle de Borel
 - 10^{10} = échelle humaine : attaque humaine
 - 10^{20} = échelle terrestre : attaque terrestre
 - 10^{100} = échelle cosmique : attaque cosmique
 - $>10^{100}$ = attaque super-cosmique
- Taille de clef et attaque exhaustive :
 - Clé de 128 bits aléatoire : $2^{128} = 10^{36}$
 - Clé de 256 bits aléatoire : $2^{256} = 10^{75}$
 - Clé de 512 bits aléatoire : $2^{512} = 10^{150}$
 - Mais attention aux failles !!!

Recommendations EMV, NIST

[<https://www.keylength.com/>]



Recommendations ANSSI

Date	Symmetric	Factoring Modulus	Discrete Logarithm Key	Group	Elliptic Curve GF(p)	Curve GF(2 ⁿ)	Hash
2014 - 2020	100	2048	200	2048	200	200	200
2021 - 2030	128	2048	200	2048	256	256	256
> 2030	128	3072	200	3072	256	256	256

Recommendations BSI

Date	Symmetric	Factoring Modulus	Discrete Logarithm Key	Group	Elliptic Curve	Hash
2017 - 2022	128	2000	250	2000	250	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512
> 2022	128	3000	250	3000	250	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512

Compromis performance / sécurité

- Echange de clefs (ex: STS)
- Communication par chiffrement symétrique (AES)
 - Chiffrement espace mémoire / disque : XTS
- Exemple: PGP

AUTHENTIFICATION

Application du chiffrement asymétrique: Authentification



- Alice peut authentifier Bob en lui soumettant un challenge (dont elle connaît la solution) :
 - Bob, grâce à sa clef secrète, peut résoudre le challenge
 - Exercice:
 - illustrer sur un chiffrement asymétrique abstrait
 - illustrer sur ElGamal
- « **Zero-knowledge** » : protocoles à divulgation nulle de connaissance
 - Celui qui authentifie n'apprend rien d'autre que le fait que l'autre possède la clef secrète
 - Quelqu'un qui observe n'apprend rien!
 - Est-ce vrai pour ElGamal ?

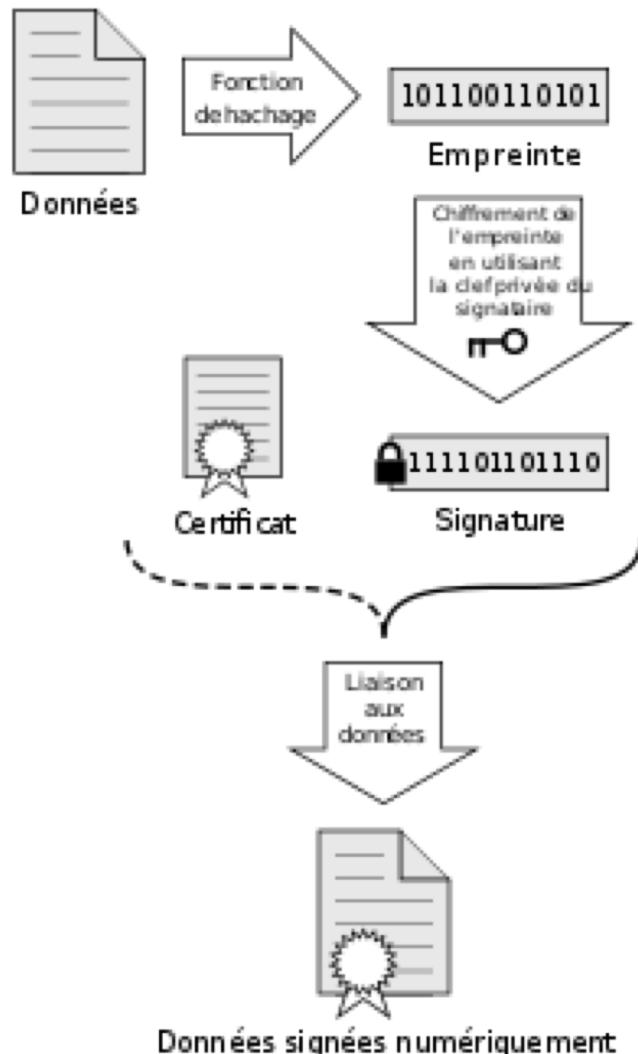


Signature Elgamal

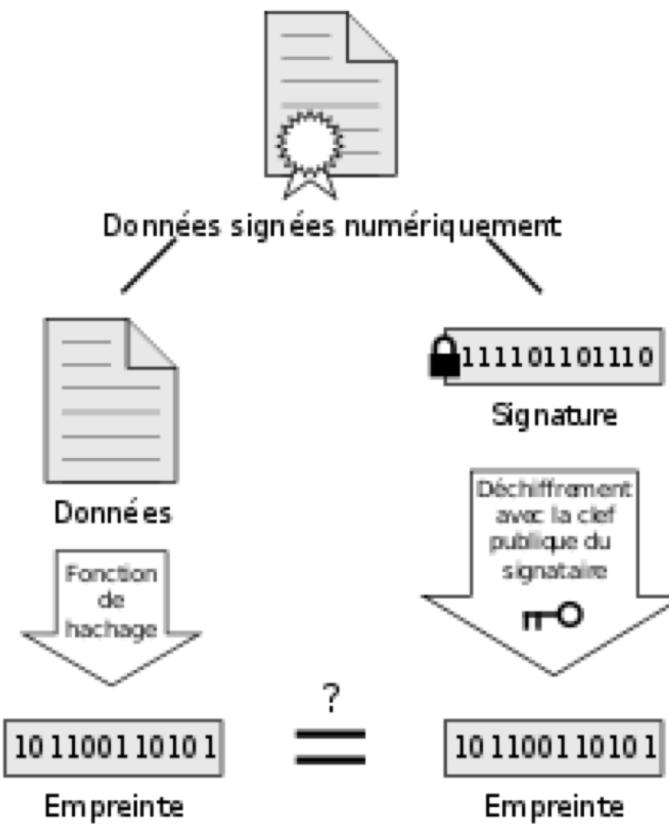
- Dans $G = \mathbb{Z}_p^*$ avec p premier (groupe cyclique d'ordre $p-1$ de générateur g)
- Alice signe le message m
- Clef secrète et publique d'Alice:
 - $A = g^a \text{ mod } p$, clef publique = (p, g, A) clef secrète= a (avec $1 < a < p-1$)
- Signature: Alice choisit un entier k secret tel que $\text{pgcd}(k, p-1) = 1$
 - $r = g^k$
 - $s = (\text{Hash}(m) - a.r) \cdot k^{-1} \text{ mod } p-1$ (si $s=0$, recommencer!)
 - Signature = (r, s) ; et donc message signé = (m, r, s)
- Vérification que (m, r, s) a bien été signé par Alice :
 - $g^{\text{Hash}(m)} =? A^r \cdot r^s \text{ mod } p-1$
 - Exercice: prouver que la vérification est correcte
- Exercice: comment usurper la signature d'Alice?

Signature numérique

Signature



Vérification



Si les empreintes sont identiques, la signature est valide

- https://fr.wikipedia.org/wiki/Signature_num%C3%A9rique

Signature DSA, ECDSA

- Algorithme de signature standard
- Génération de clefs (clef secrète + clef publique)
 - Signature avec la cle privée
 - Seul le hash $H(m)$ du message m est signé
 - Vérification de la signature avec la cle publique
 - Initialement défini sur Z_p^* ;
mais d'autres groupes sont possibles (exemple: EC DSA)
 - Affecte la génération des clefs (et leurs tailles = paramètre de sécurité)

Cf http://fr.wikipedia.org/wiki/Digital_Signature_Algorithm

Ce qu'on a vu aujourd'hui

- 1. Vulnérabilités - CVE.**
- 2. Code – Confidentialité, Authentification, Intégrité, Non-répudiation**
- 3. Chiffrement symétrique (coffre-fort)**
- 4. Chiffrement asymétrique (boite aux lettres)**
- 5. Echange de clef sécurisé (STS)**
- 6. Signature électronique**
- 7. Hachage: applications (authentification, intégrité, ...)**