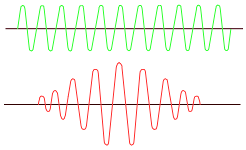
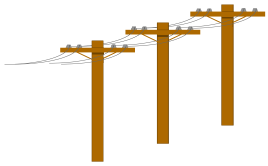
JLS CONSEIL





Sécurité des Transactions
Protection de l’information

Bases mathématiques pour la sécurité informatique

Première partie Principes de base

Jean-Luc Stehlé
EPITA FMSI ING 1
25Avril 2013

Jean-Luc.Stehle@NormaleSup.org

JLS CONSEIL

Planning des cours

Date	Code Cours	Horaire	Nb heures	Lieu	Promo	Cursus	Activité	Enseignant	N° de Salle	SEM	Intitulé du cours
jeudi 25 avril 2013	FMSI	14h00 à 17h00	3,0	KREMLIN	2015	ING1	COURS	STEHLE.Jean-Luc	amphi 4	S2	Fondements mathématiques pour la sécurité informatique
jeudi 30 mai 2013	FMSI	14h00 à 17h00	3,0	KREMLIN	2015	ING1	COURS	STEHLE.Jean-Luc	amphi 1	S2	Fondements mathématiques pour la sécurité informatique
jeudi 6 juin 2013	FMSI	14h00 à 17h00	3,0	KREMLIN	2015	ING1	COURS	STEHLE.Jean-Luc	amphi 4	S2	Fondements mathématiques pour la sécurité informatique
jeudi 13 juin 2013	FMSI	14h00 à 17h00	3,0	KREMLIN	2015	ING1	COURS	STEHLE.Jean-Luc	amphi 4	S2	Fondements mathématiques pour la sécurité informatique

JLS CONSEIL

Bases mathématiques pour la sécurité de l'information

- **Sensibilisation générale**
- **Protocoles classiques** (Diffie Helmann, RSA…)
- **Compléments d'arithmétique modulaire**
- **Bases mathématiques pour l'AES**
- **Courbes elliptiques**

Jean-Luc.Stehle@NormaleSup.org

JLS CONSEIL

BIBLIOGRAPHIE

- **B. Beckett** Introduction aux méthodes de la cryptologie *Masson1990*
- **G. Brassard** Cryptologie contemporaine *Masson1993*
- **G. Konheim** Cryptography : A primer *John Wiley 1981*
- **E. Kranakis** Primality and Cryptography *John Wiley 1986*
- **D.E.R. Denning** Cryptography and data security *Addison Wesley 1983*
- **X. Marsault** **Compression et cryptage en informatique** ***Hermès Paris 1992***
- **G. Robin** Algorithmique et cryptographie *Ellipses Paris 1991*
- **B. Schneier** **Applied cryptography** ***John Wiley 1993***
  Cryptologie Appliquée ***Thomson Publishing 1997***
- **M.R. Schroeder** Number Theory in Science and Communication with applications in Cryptography, Physics, Digital Information, Computing,... *Springer 1986*
- **J.H. Van Lint** **Introduction to Coding Theory** ***Springer 1982***

JLS CONSEIL

BIBLIOGRAPHIE (Suite)

Revue « Pour la Science »

Dossier spécial N° 36

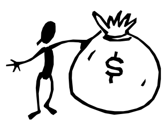
Juillet/Octobre 2002

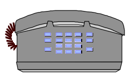
Excellente synthèse de l'état de l'art actuel

JLS CONSEIL

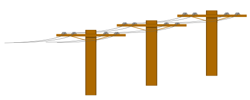
Quelques exemples instructifs

- Distributeurs de billets
- Changeurs de devises
- Piratage du téléphone

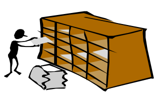





- Ralentissement de SWIFT



- Fichiers de malades

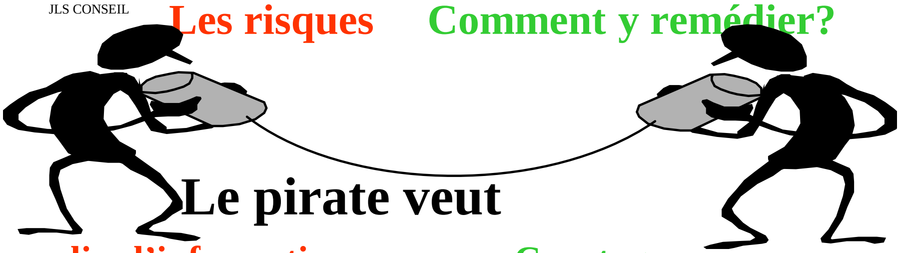


- Commerce électronique

JLS CONSEIL

Sécurité des transactions

Les risques   Comment y remédier?



Le pirate veut

- **lire l'information** — **Cryptage**
- **modifier l'information** — **Scellement**
- **se faire passer pour l'interlocuteur** — **Authentification**, **Signature électronique**

JLS CONSEIL

Sécurité des transactions

Les risques   Comment y remédier?

- **L'émetteur renie sa parole** — **Non répudiation**

- **Les deux interlocuteurs sont-ils bien d'accord ?**

Problème des accusés de réception...

JLS CONSEIL

Le problème des généraux byzantins


Armée A


Armée B


Armée C

A<C

B<C

A+B>C

Comment communiquer entre A et B en assurant le consensus ?

- Problème de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack de l'Ack.........
- **Problème des « connaissances communes »**

© Aurélien Leteinturier

JLS CONSEIL

Le Scellement

MESSAGE	Sceau

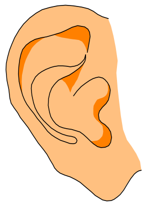
- **Sceau = f(Message, clé de scellement)**

  Assure que le message n'a pas été modifié

- **Problème de la non répudiation**

JLS CONSEIL

Divers types d'attaque

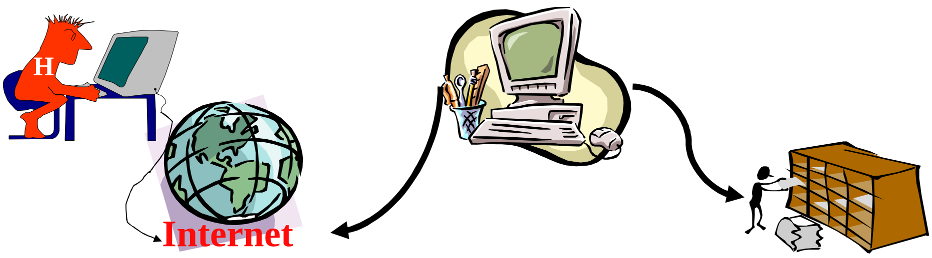


- **Écoute passive**
- **Écoute avec partie du clair connu**
- **Le pirate peut envoyer des messages de son choix**

- **Le pirate se fait passer pour l'interlocuteur**

JLS CONSEIL

Le problème de la « tête de pont »

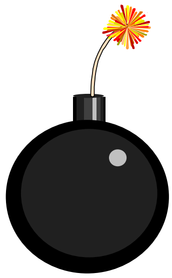


Il faudrait autant de réseaux physiques différents qu'il y a de projets ou d'applications différentes

JLS CONSEIL

Divers types d’attaque : le pirate peut injecter

- Un virus
- Un cheval de Troie
- Une bombe logique

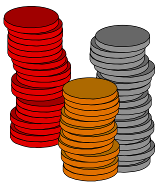


JLS CONSEIL

Principes de base de la protection (1)

Le pirate dispose de toutes les ressources de la technique

- Il est prêt à y mettre le prix
- Chiffrer
  - Coût du piratage
  - Bénéfice pour le pirate
  - Prix à payer pour la protection



JLS CONSEIL

Principes de base de la protection (2)

On est à la merci d'une faille humaine

- **Tout homme a un prix**
  - Complicités internes
  - **Chantage** *(La carotte ou le bâton)*

- **Pourquoi l'employé trahit-il ?**
  - pour s'enrichir
  - par malveillance, vengeance,...
  - par jeu

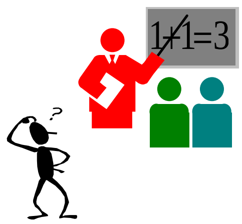
Profil psychologique et socioprofessionnel des employés ayant la possibilité de trahir

JLS CONSEIL

Principes de base de la protection (3)

On est à la merci des progrès

- **de la technique**
- **des mathématiques**



JLS CONSEIL

Principes de base de la protection (4)

Le pirate a de la chance

- **S'il y a une faille, il la trouvera**
- **Rechercher le maillon faible**

  L'information est-elle piratable avant cryptage ?

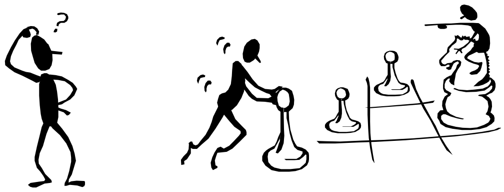
JLS CONSEIL

Principes de base de la protection

Approche globale de la sécurité

Jeu d'échec

- **Le pirate a les blancs**
- **Il faut prévoir d'avance sa stratégie**



JLS CONSEIL

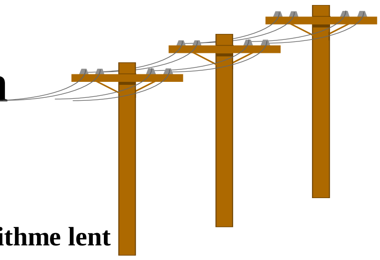
Principes de base de la protection

La solidité d'une chaîne est celle de son maillon le plus faible

JLS CONSEIL

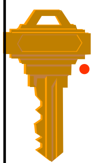
Les éléments de la sécurité

1. Communication



- **Authentification**

  On peut accepter un algorithme lent



- **Partage d'une clé de codage**

  doit être modifiée souvent

- **Codage proprement dit**

  nécessite un algorithme rapide

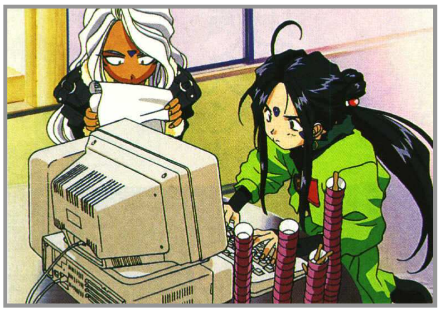
Document destiné uniquement aux élèves et aux enseignants de l'EPITA. L'auteur vous remercie d'avance de ne pas diffuser ce document

JLS CONSEIL

Les éléments de la sécurité

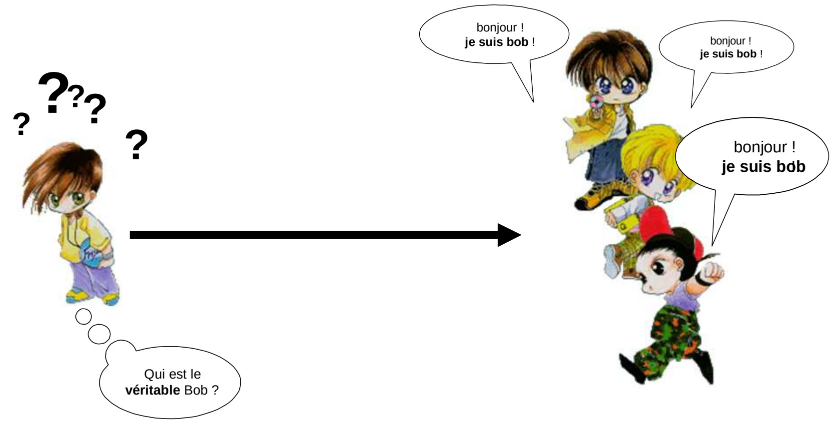
2. Contrôle

- **Accusé de réception**
- **Scellement**
- **Non répudiation**



JLS CONSEIL

Authentification des interlocuteurs

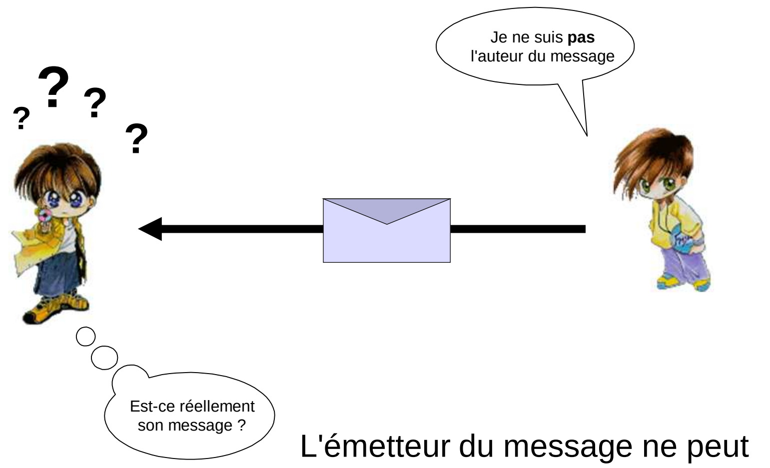


Être sûr de l'identité de l'interlocuteur

© Aurélien Leteinturier

JLS CONSEIL

Non répudiation



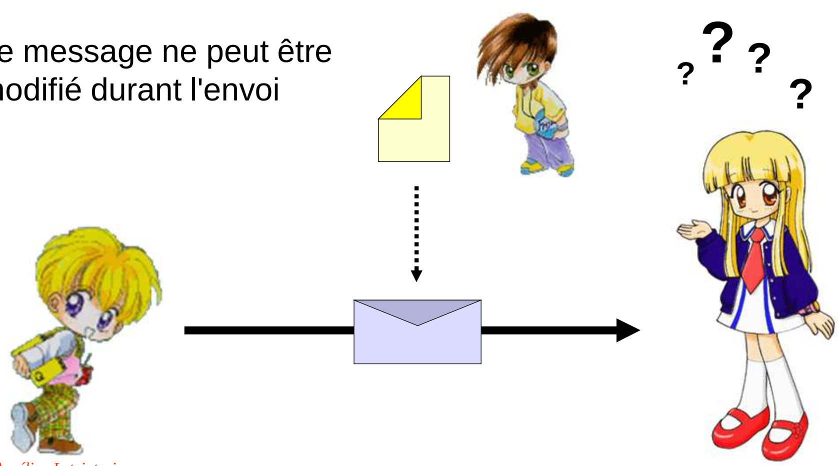
L'émetteur du message ne peut nier en être l'auteur

© Aurélien Leteinturier

JLS CONSEIL

Sceau / signature

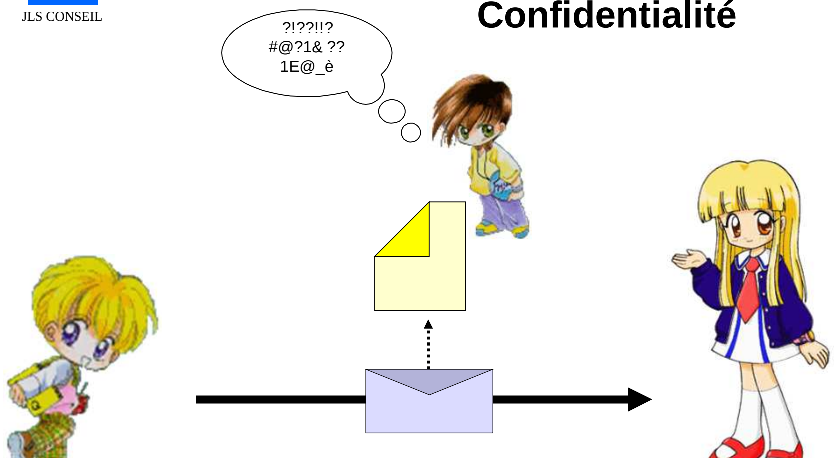
Le message ne peut être modifié durant l'envoi



© Aurélien Leteinturier

JLS CONSEIL

Confidentialité



Le message ne peut être consulté pendant le transfert

© Aurélien Leteinturier

JLS CONSEIL

Les éléments de la sécurité

3. Aspects légaux

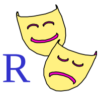
- **Législations contraignantes**
- **Trouver le juste compromis entre**
  - Assurer la sécurité des transmissions «honnêtes»
  - Empêcher qu'un système trop sécurisé permette
    - trafics divers…
    - blanchiment d'argent sale
    - Réseaux pédophiles
    - Terrorisme
    - …

Justice
Liberté
Sécurité

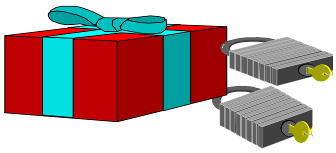
JLS CONSEIL

Exemples de codes de cryptage

- **Codes historiques**
  César, Vigenère
- Masques XOR
- Multicanaux



- **DES** **Data Encryption Standard**
  - Blocs de 64 bits
  - Standard USA depuis 1977
  - Algorithme symétrique (même clé de part et d'autre)
- **Méthode du colis à deux cadenas**
- **AES**
- **RSA**



JLS CONSEIL

Code de cryptage

- **Le problème de la backdoor**
  - **Un algorithme ne devrait être utilisé qu'après que toute la communauté des cryptographes ne l'ait validé**
    - Recherche de failles
    - Recherche de backdoor
  - **Et encore…**

Histoire de la cryptographie : Chiffre de César et dérivés

JLS CONSEIL

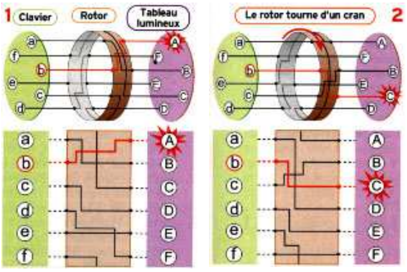
- On peut rendre le système de César plus robuste par création d'un alphabet mélangé : **ABCDEFGHIJKLMNOPQRSTUVWXYZ**
  MPLOKINJUBHYVGTCFRXDEWSZQA

Un tel système fut utilisé à la Renaissance. Il y a dans ce cas 26! soit 4.10^{26} clés possibles, la difficulté étant de les retenir :

© Franquin

Histoire de la cryptographie : La machine Enigma

JLS CONSEIL



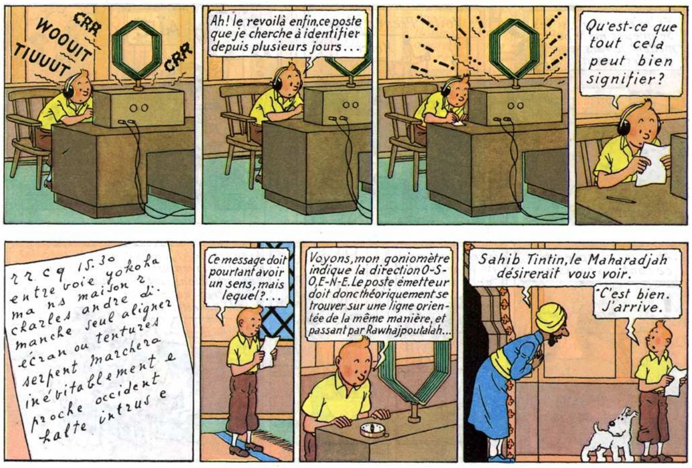
Le crypt(1) d'UNIX fonctionne encore sur ce principe

Utilisée par l'Allemagne Nazie : 3 rotors avec alphabet mélangé câblé changé à chaque caractère par rotation des rotors. Equivalent à un masque de période 26*26*26 = 17576.

Fut cryptanalysée avec succès par les alliés…

JLS CONSEIL

Cryptographie vs Stéganographie



© Hergé

JLS CONSEIL

Stéganographie

Stéganographie (du Grec steganos : couvert) : message caché (par un procédé secret) dans un autre d'apparence anodine

Un cryptogramme n'a pas une apparence anodine, c'est en général un inextricable charabia (gibberish).

© Hergé

JLS CONSEIL

Un exemple littéraire

d'après G. Sand

Je suis émue de vous dire que j'ai bien compris l'autre soir que vous aviez toujours une folle envie de me faire danser, je garde le souvenir de votre baiser et je voudrais bien que ce soit la preuve que je puisse être aimée par vous, je suis prête à vous montrer mon affection toute désintéressée et sans calcul, et si vous voulez me voir ainsi vous dévoiler sans nul artifice mon âme toute nue, veuillez me faire une visite. Nous causerons franchement en ami

Je vous prouverai que je suis la femme sincère et capable de vous offrir l'affection la plus profonde comme la plus étroite amitié : en un mot, la meilleure épouse que vous puissiez rêver. Puisque votre âme est libre, pensez que la détresse où j'habite est bien longue, bien dure et souvent bien difficile à vivre et me cause une peine très grosse. Accourez bien vite et venez me la faire oublier. A l'amour, je vais me soumettre.

© George Sand

JLS CONSEIL

Un exemple de cryptogramme

d'après G. Sand

Je suis émue de vous dire que j'ai
bien compris l'autre soir que vous aviez
toujours une folle envie de me faire
danser, je garde le souvenir de votre
baiser et je voudrais bien que ce soit
la preuve que je puisse être aimée
par vous, je suis prête à vous montrer mon
affection toute désintéressée et sans cal-
cul, et si vous voulez me voir ainsi
vous dévoiler sans nul artifice mon âme
toute nue, veuillez me faire une visite.
Nous causerons franchement en ami

Je vous prouverai que je suis la femme
sincère et capable de vous offrir l'affection
la plus profonde comme la plus étroite
amitié : en un mot, la meilleure épouse
que vous puissiez rêver. Puisque votre
âme est libre, pensez que la détresse où j'ha-
bite est bien longue, bien dure et souvent
bien difficile à vivre et me cause une peine
très grosse. Accourez bien vite et venez me
la faire oublier. A l'amour, je vais me sou-
mettre.

© George Sand

JLS CONSEIL

Les codes asymétriques

RSA : Système asymétrique

A dispose d'une clé publique et d'une clé privée

- Tout le monde peut envoyer un message confidentiel que seul A peut lire

  Cryptage

- A peut signer : tout le monde peut vérifier sa signature

  Authentification

- **Semble le meilleur actuellement**

JLS CONSEIL

Les codes asymétriques

Inconvénients de RSA

- **Lent**
- **A la merci des progrès de mathématiques**
- **Problème de génération/gestion des clés**

JLS CONSEIL

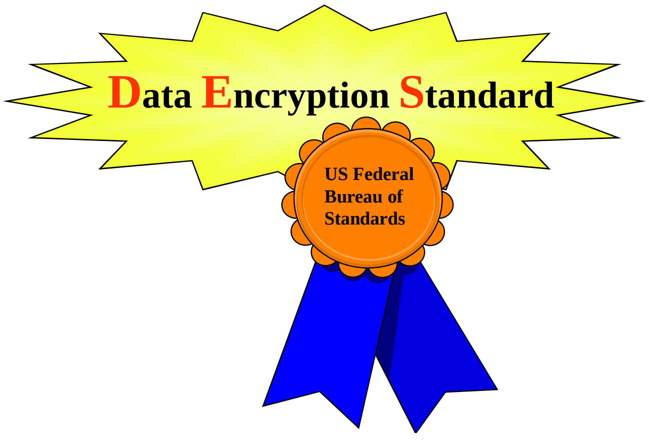
Conclusion

- Nombreux algorithmes, nombreuses techniques
- Difficultés de mise en place
  - Gestion des clés
  - Failles humaines
- Sécurité = approche globale
  - Traquer **toutes** les failles du système
  - Approche globale

JLS CONSEIL

Compléments sur le DES

Le « Standard » ancien

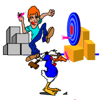


DES : Data Encryption Standard

JLS CONSEIL

- **1977 US Federal Bureau of Standards**

- *non autorisé pour le secret défense aux USA*
- **Facile à implémenter**
  - 300 lignes Fortran
- **Lent si implémentation logicielle**
- **Rapide si implémentation Hard**
  - 100 Mbits/s en 1993
- **Algorithme public**
- **Blocs de 64 bits (8 octets)**
- **Clés à 56 bits, symétrique**



- **Seule attaque connue**
  - Essai de toutes les $2^{56} = 7.2\ 10^{16}$ clés par recherche exhaustive
- **Vulnérabilité**
  - La gestion des clés
  - Clés «faibles»

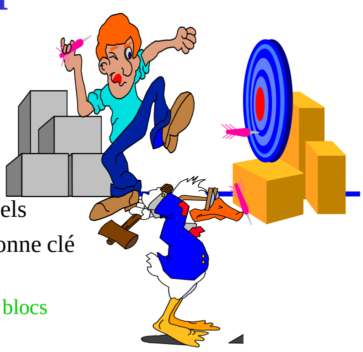
Améliorations

- **Triple DES**
  - $c = Des_{k1}(Des_{k2}^{-1}(Des_{k1}(m)))$
  - $m = Des_{k1}^{-1}(Des_{k2}(Des_{k1}^{-1}(c)))$
- **Mode CBC**
  - **Cipher Block Chaining**
  - $m = m_1 m_2 m_3 \ldots m_n \ldots$
  - $c_i = Des_k(m_i \oplus c_{i-1})$
  - $m_i = c_{i-1} \oplus Des_k^{-1}(c_i)$

DES : Data Encryption Standard

JLS CONSEIL

- ***Casser le DES ?***



  - **Recherche exhaustive**
    - $2^{56} = 72 \times 10^{15}$ clés à tester
    - A la porté des moyens de calcul actuels
    - Comment savoir qu'on a trouvé la bonne clé
      - Attaque à clair connu
      - D'autant plus facile qu'on a plus de blocs
  - **Attaques sans cassage de code**
    - ***Blocs rejoués***
    - ***Ajouter un MAC ?***

Banque émetteur	15 blocs
Banque bénéficiaire	15 blocs
Nom déposant	6 blocs
Numéro compte	2 blocs
Montant du dépôt	1 bloc

DES : Data Encryption Standard

JLS CONSEIL

- Permutation initiale PI
- 16 itérations

  $G_i = D_{i-1}$

  $D_i = G_{i-1} \oplus f(D_{i-1}, K_i)$

- Permutation inverse PI^{-1}

DES^{-1} : Il suffit de prendre les clés dans l'autre sens (décalages à droite)

- Sélection des clés K_i (48 bits)

  Clé initiale (56 bits) Permutation PC1

  ==> (g_0,d_0) (28 bits + 28 bits)

  Décalages gauches successifs sur g et d, de $k_i \in [1;2]$

  ==> (g_i,d_i) $(g_{16},d_{16}) = (g_0,d_0)$

  Permutations avec oubli ==>48 bits

  K_i = PC2 (g_i,d_i)

- Fonction f

  fonction d'extension E : D_{i-1} (32 bits) ==> 48 bits

  $\oplus$ K_i (48 bits) ==> 8 blocs de 6 bits

  On leur applique les Sboxes

  (8 boites noires : tableaux 4x16 de 4 bits)

  Bit 1 et 6 = Numéro de ligne Bit 2 à 5 = Numéro de colonne

  On lit 4 bits dans la Sbox ==> **32 bits résultats**

Cf. Robin ou Beckett ou Schneier

Génération d'une fonction inversible : Le schéma de Feistel

JLS CONSEIL

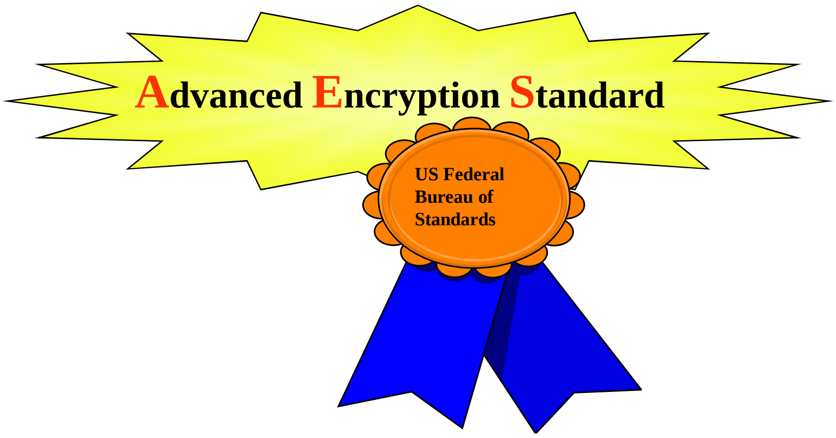
- **Étant donnée une fonction sur n bits, on peut en déduire une fonction inversible sur 2n bits**
  - $G_i = D_{i-1}$
  - $D_i = G_{i-1} \oplus f(D_{i-1}, K_i)$

  Cette fonction a pour fonction inverse
  - $D_{i-1} = G_i$
  - $G_{i-1} = D_i \oplus f(G_i, K_i)$
- **DES est un schéma de Feistel à 16 étapes**
- ***De nombreux algorithmes de cryptage sont basés sur les schémas de Feistel***

L'AES

JLS CONSEIL

Le nouveau « Standard »



L'AES appelé aussi Rijndael

JLS CONSEIL

Conçu par Vincent Rijmen et Joan Daemen
Université de Leuven Belgique

Choisi le 2 octobre 2000 par le NIST
(National Institute of Standards and Technology)

- **Plus rapide que le DES**
- **Blocs de 128 bits**
- **Clés de 128, 192 ou 256 bits**

- **Pour le moment aucune faille connue**

L'AES — Principe de base

JLS CONSEIL

Le corps de base est un corps fini K à 256 éléments isomorphe à $\{0;1\}[X] / (X^8 + X^4 + X^2 + X + 1)$

Un octet représente un élément de K

- Addition identique à XOR bit à bit
- Multiplication = multiplication de polynômes suivi d'une division euclidienne

Chaque bloc (128 bits = 16 octets) s'écrit comme une matrice (4,4) dont les éléments sont des octets

L'algorithme lui-même

- *On additionne une clé secrète au bloc*
- *On effectue 10 itérations, chacune ayant 4 étapes*

L'AES — Une itération = 4 étapes

JLS CONSEIL

Un bloc est une matrice (4,4) formée d'éléments de K (octets)

1. Transformation non linéaire S appliquée à chaque octet

2. Décalage des lignes

Permutations circulaires (0,1,2,3) vers la gauche

3. Brouillage des colonnes

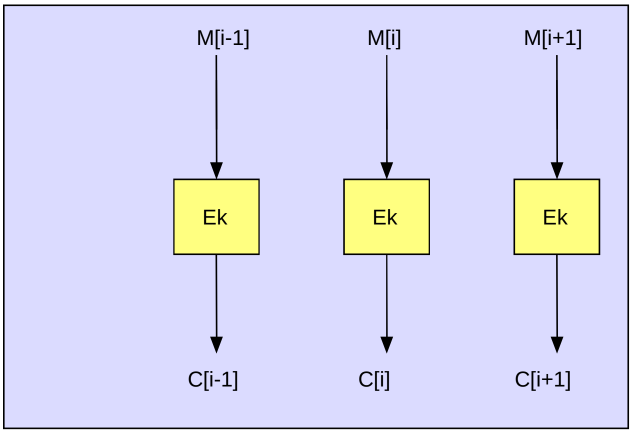
Multiplication d'une colonne par une matrice (4,4) dont les coefficients sont pris dans {1,2,3}

4. Addition d'une clé de tour (16 octets)

La clé de tour dépend de la clé secrète et est variable d'un tour à l'autre

JLS CONSEIL

Mode « standard » EBC

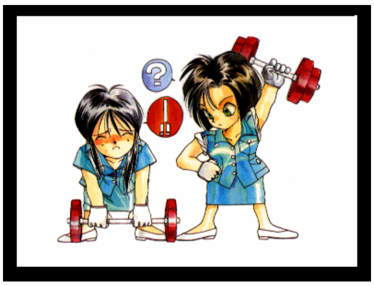


Risque : l'attaquant peut se constituer un carnet de codage = dictionnaire (Electronic code book)
Possibilité d'attaque par bloc rejoué

JLS CONSEIL

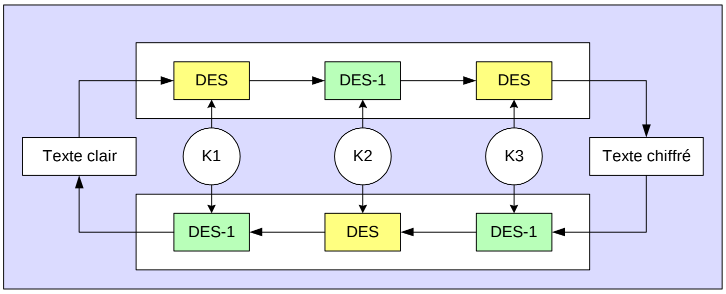
Amélioration du DES

- ***Rendre le chiffrement plus fort***



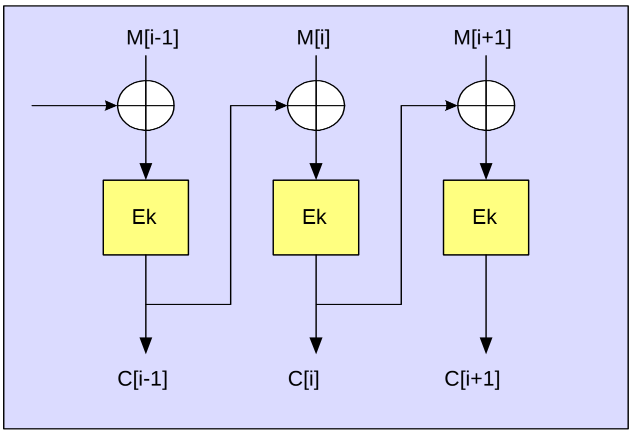
- Triple DES
- Chaînage des algorithmes par blocs (CBC, CTS, CTR…)

Triple DES



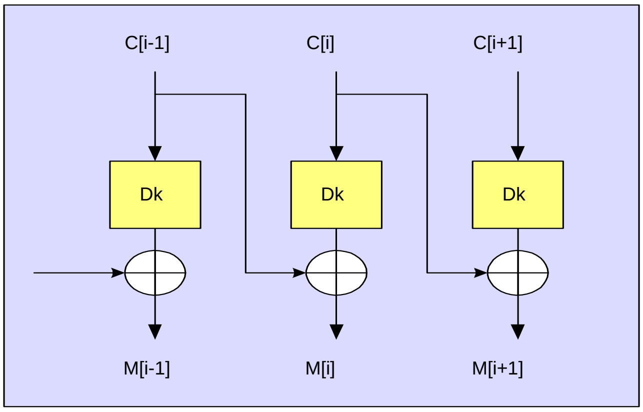
- K1 peut être égal à K3, l'algorithme est alors plus facilement réversible

Mode chaîné CBC



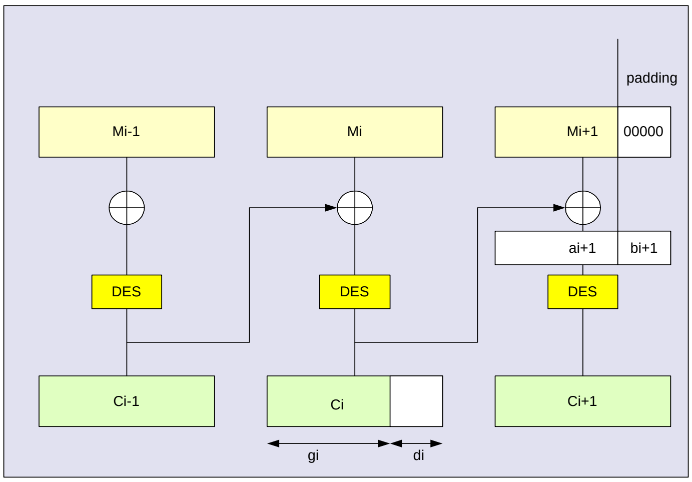
JLS CONSEIL

Mode chaîné CBC



JLS CONSEIL

Mode chaîné CTS



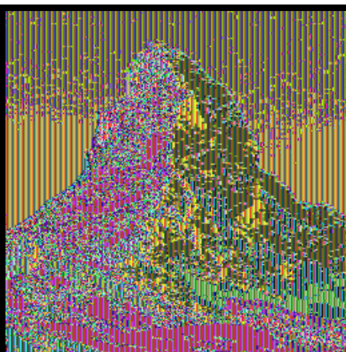
Mode CTR

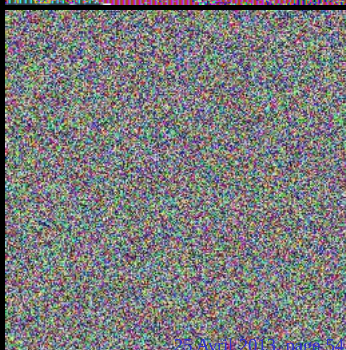
JLS CONSEIL

- **On chiffre un compteur, le résultat du chiffrement est XORé avec le texte à chiffrer/déchiffrer**
  - **Chiffrement = déchiffrement**
  - **Pratique pour le chiffrement de supports à accès direct**
    - ***Inutile de tout lire pour déchiffrer un secteur***

Masque[n] = AES_K(f(n))
CT[n] = PT[n] $\oplus$ Masque[n]

Tous ces modes de chaînage sont valables pour tous les algorithmes de chiffrement par blocs





Chiffrement d'une image en AES
En mode EBC
Avec chaînage

© Wikipedia

JLS CONSEIL

Les services d'e-banking

Banque à domicile

- **Problèmes de l'authentification de l'utilisateur**
  - ☞ **Comment éviter qu'un pirate se fasse passer pour le client ?**
  - ☞ Madame Michu a des compétences limitées en matière de sécurité informatique
  - ☞ Le pirate peut facilement pirater les données d'authentification du client
    - ➢ Spyware espionnant les frappes clavier
    - ➢ Attaques par phishing

JLS CONSEIL

Les services d'e-banking

Détection du phishing

- **Le phishing**
  - ☞ Le pirate simule le faux site bancaires du client
  - ☞ Il reroute le client vers ce site par des mails piégés
- **Comment détecter les phishing ?**
  - ☞ **Ne jamais répondre à des demandes envoyées via Internet**

Phishing-Mail vom 4. Juni 2005 /23:17

Von: PostFinance <DeclanAcker@postfinance.ch>
Datum: 4. Juni 2005 23:17:10 GMT+02:00
an: vorname.name@bluewin.ch
Betreff: PostFinance Client - vorname.name@bluewin.ch

Dear PostFinance Customer,

This email was sent by the PostFinance server to verify your e-mail address. You must complete this process by clicking on the link below and entering in the small window your PostFinance online access details. This is done for your protection - because some of our members no longer have access to their email addresses and we must verify it. To verify your e-mail address, click on the link below:

http://www.postfinance.ch/OQgfy7USKaSZEQsx43N3c8ZAK9vKdxKRsBOkQc7S6mf1tYaiP42wjyg1s9g4j58s

---

Hinter dem Link ist folgendes versteckt:
http://www.google.ca/url?q=http://go.msn.com/HML/6/8.asp?target=http://%09%7163%71cd%69%2E%64%41%%2E%%%%09%%52%09%%%75%%09/

JLS CONSEIL

Les services d'e-banking

Détection du phishing

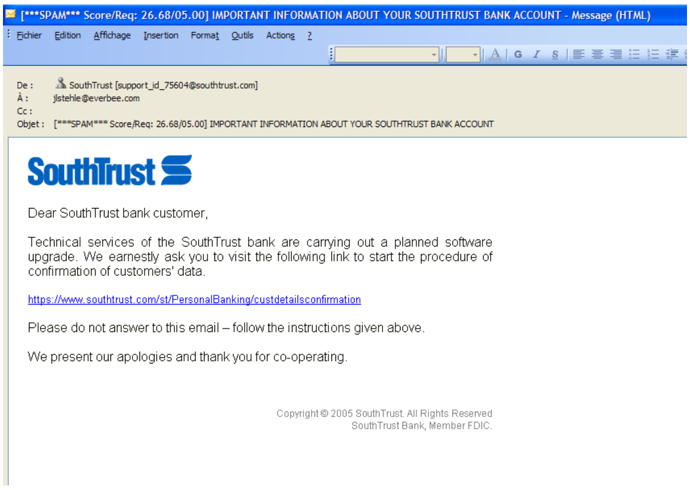
- **La zone sur laquelle on demande de cliquer est une image**
  - ☞Modification du pointeur de la souris
  - ☞Derrière l'image se cache un site pirate
- **Il y a souvent des textes cachés (blanc sur blanc)**
  - ☞Permet de bypasser certains détecteurs de spam
  - ☞Lisible si on les sélectionne à la souris

Quelques exemples

JLS CONSEIL

Les services d'e-banking

Détection du phishing



JLS CONSEIL

Les services d'e-banking

Banque à domicile Solutions pour authentifier l'utilisateur

- **Login / Password**
  - ☞ Très facile à attaquer par spyware
- **Login / Password entré à la souris sur une mire aléatoire**
  - ☞ **Solution BNP Paribas**
  - ☞ **Attaque possible : le spyware doit récupérer la mire et tous les mouvements de la souris**
- **Après Login / Password , utilisation d'un mot de passe à usage unique**
  - ☞ **Ancienne solution PostFinance Suisse**
  - ☞ **Le client reçoit une petite carte contenant 100 mots de passe, et après l'authentification Login/Password standard, on lui demande un des 100 mots de passe de sa carte**
  - ☞ **Attaque possible : le pirate doit photocopier la carte à l'insu du client**

☞ **Après Login / Password , Défi/Réponse avec calcul utilisant un pincode.**
SOLUTION ACTUELLEMENT OPTIMALE

JLS CONSEIL

Les services d'e-banking

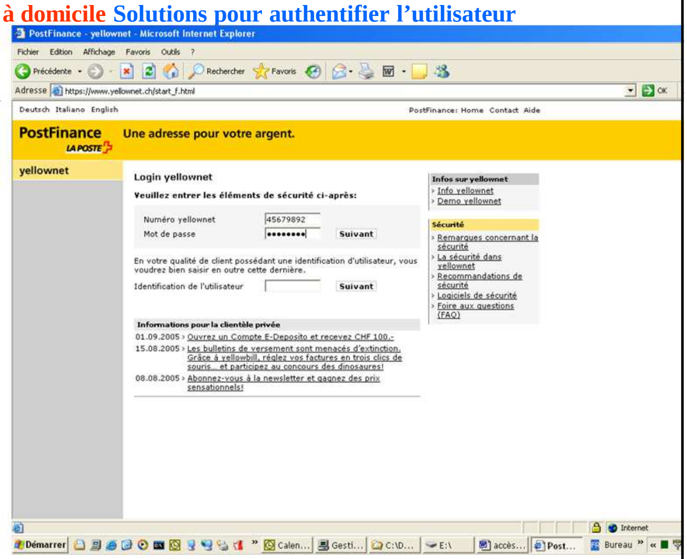
Banque à domicile Solutions pour authentifier l'utilisateur

Login / Password entré à la souris sur une mire aléatoire

JLS CONSEIL

Les services d’e-banking

Banque à domicile Solutions pour authentifier l’utilisateur



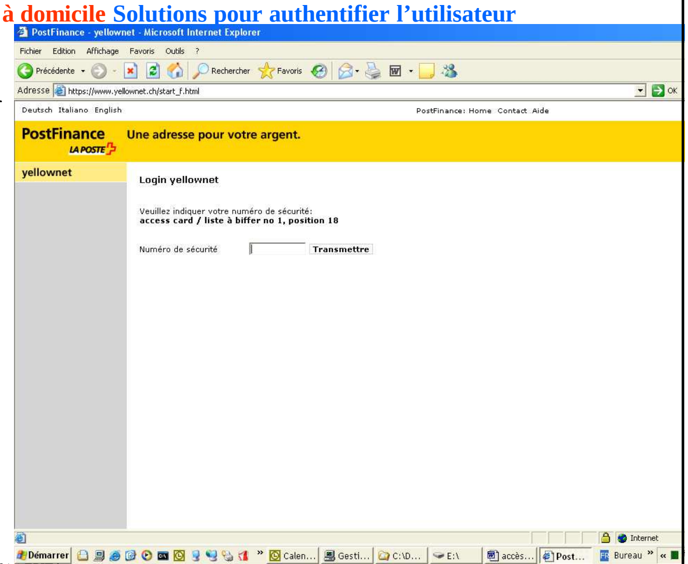
Mot de passe à usage unique

Phase 1 : Préauthentification

JLS CONSEIL

Les services d’e-banking

Banque à domicile Solutions pour authentifier l’utilisateur

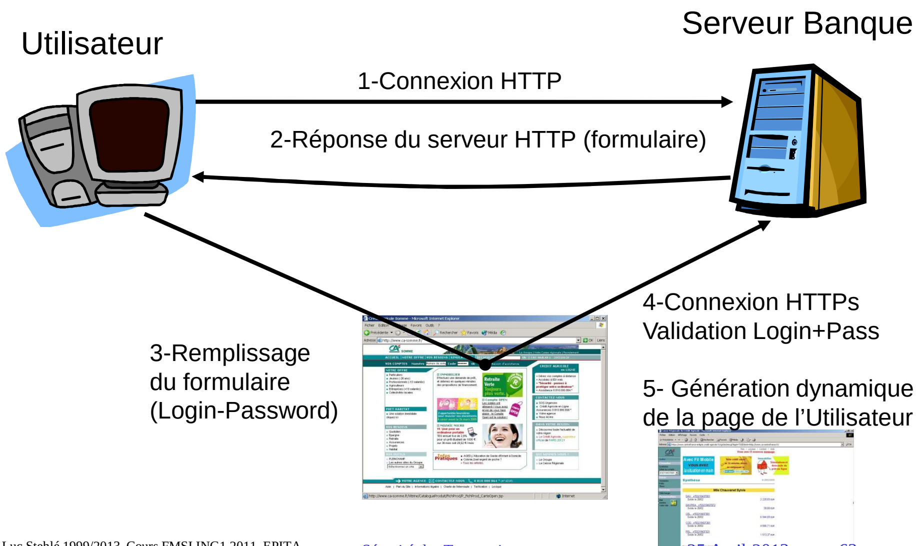


Mot de passe à usage unique

Phase 2 : Authentification forte

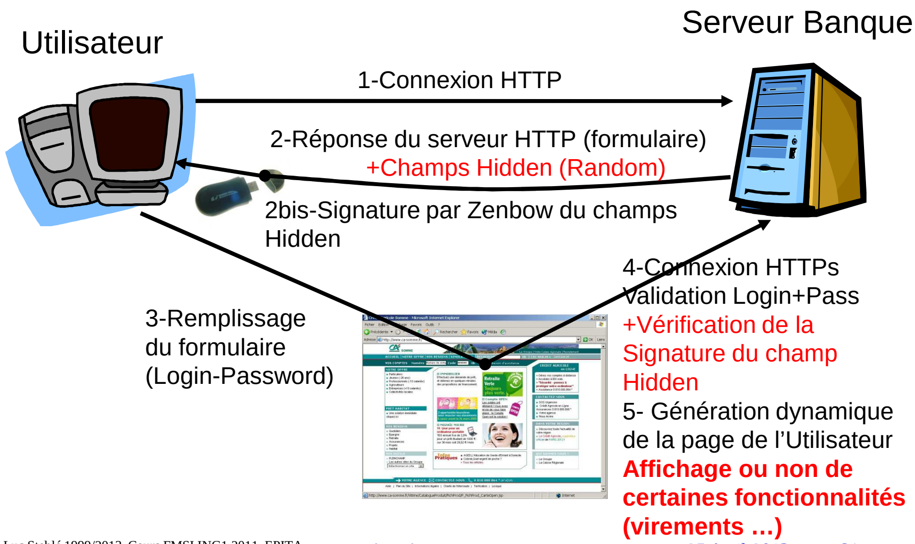
JLS CONSEIL

Principe général de la connexion
Protocole actuel



JLS CONSEIL

Principe général de la connexion
Technologie Everbee

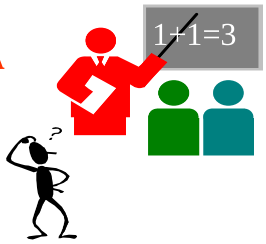


JLS CONSEIL

Et maintenant les détails mathématiques sur

Le logarithme discret
Les nombres premiers
Le théorème d'Euler et RSA

Fonctions à sens unique



JLS CONSEIL

Arithmétique modulaire

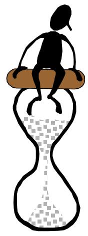
- **De nombreux systèmes cryptographiques sont basés sur l'arithmétique modulaire**
  - **Logarithme discret**
  - **Théorèmes de Fermat et d'Euler**
  - ***Théorème de Bezout***
  - **RSA**
    - **Propriétés des nombres premiers**
    - **Tests de primalité**
- **Développer des algorithmes efficaces en arithmétique modulaire**

Arithmétique modulo N

JLS CONSEIL

N est très grand : 1024 bits ($\cong 10^{300}$) , 2048, …, 4096 bits

- **Addition** — **Facile (Temps en Log N)**
- **Multiplication** — **Facile mais plus long (Log^2 N)**
- **Division** — **Plus difficile (Log^3 N)** ***avec Bezout/Euclide généralisé***
- **Réduction modulo N** — **Il existe des algorithmes de complexité équivalente à celle de la multiplication**
- **Puissance a^b** — **Facile (Log^3 N)** ***(écrire b en binaire)***



Problème du logarithme Discret

Pas d'algorithme rapide

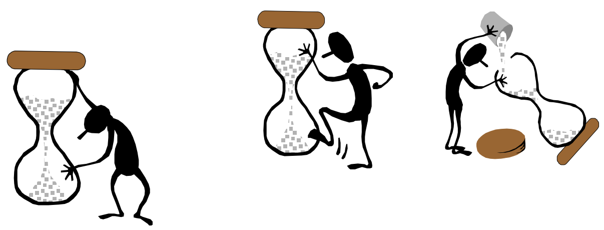
$a^x \equiv b$ [N] ?

Problème du logarithme Discret

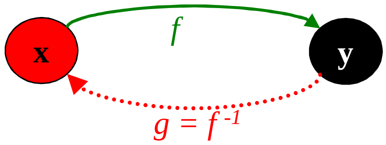
JLS CONSEIL

$$a^x \equiv b\ [N]$$

- **Pour N grand (10^{300}), le calcul de x connaissant a et b nécessite un temps supérieur à l'âge de l'univers**



L'exponentiation modulaire est une fonction à sens unique

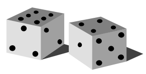


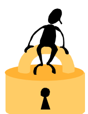
a et N sont connus et publics

- **$y = f(x) = a^x$ (Mod N)**
- **$x = g(y)$ est la solution, en arithmétique modulo N de l'équation $a^x = y$ (Logarithme discret en base a)**

Diffie Hellman :
Échange de clé sur un réseau public

- **N et g $\in$ ($Z/_{NZ}$)*** (Problème du choix de g)
- **Alice calcule a aléatoire, envoie g^a[N]**
- **Bob calcule b aléatoire, envoie g^b[N]**
- **Les deux peuvent calculer g^{ab}[N]**
- **Le pirate connaît g g^a [N] g^b [N]**

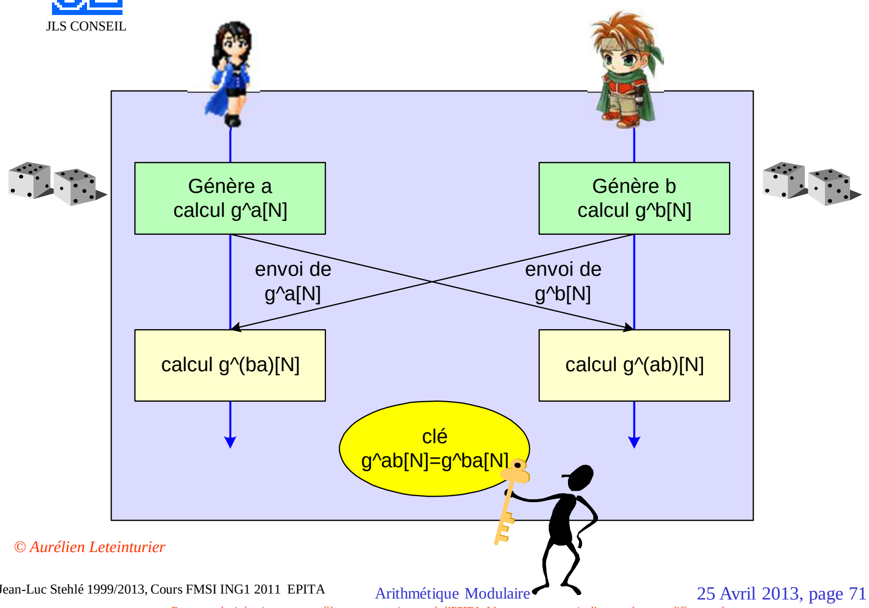




mais ne peut pas calculer g^{ab} [N]

JLS CONSEIL

Protocole de Diffie-Hellman



© Aurélien Leteinturier

JLS CONSEIL

Failles / Backdoors dans le problème du logarithme discret

- Groupes faibles
- Groupes choisis intentionnellement pour créer un backdoor
  - ➢ Utiliser des groupes « aléatoires »
  - ➢ Les groupes d'Oakley

    Basés sur les décimales de π

Les groupes d'Oakley

- **Groupe 1 : 768 bits**

$2^{768} - 2^{704} - 1 + 2^{64} \times [149686+2^{638}\pi] =$

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF
```


Les groupes d'Oakley

- **Groupe 2 : 1024 bits**

$2^{1024} - 2^{960} - 1 + 2^{64} \times [129093+2^{894}\pi] =$

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF
```

JLS CONSEIL

Les groupes d'Oakley

- **Groupe 5 : 1536 bits**

$2^{1536} - 2^{1472} - 1 + 2^{64} \times [741804+2^{1406}\pi] =$

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFF FFFFFFFF
```


JLS CONSEIL

Les groupes d'Oakley

- **Principe de construction** (exemple 768 bits)

  - $(2^{64}-1) \times (2^{704}+1) = 2^{768} - 2^{704} + 2^{64} - 1 =$

```
FFFFFFFF FFFFFFFF 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF
```

  - **On complète par le développement binaire de π entre les bits 704 = 768-64 et 64**

```
                  C90FDAA2 2168C234 C4C6628B 80DC1CD1  29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245  E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D  C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D  670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B
E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9  DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510
15728E5A 8AAAC42D AD33170D 04507A33 A85521AB DF1CBA64  ECFB8504 58DBEF0A 8AEA7157 5D060C7D B3970F85 A6E1E4C7
ABF5AE8C DB0933D7 1E8C94E0 4A25619D CEE3D226 1AD2EE6B  F12FFA06 D98A0864 D8760273 3EC86A64 521F2B18 177B200C
BBE11757 7A615D6C 770988C0 BAD946E2 08E24FA0 74E5AB31  43DB5BFC E0FD108E 4B82D120 A9210801 1A723C12 A787E6D7
88719A10 BDBA5B26 99C32718 6AF4E23C 1A946834 B6150BDA  2583E9CA 2AD44CE8 DBBBC2DB 04DE8EF9 2E8EFC14 1FBECAA6
287C5947 4E6BC05D 99B2964F A090C3A2 233BA186 515BE7ED  1F612970 CEE2D7AF B81BDD76 2170481C D0069127 D5B05AA9
93B4EA98 8D8FDDC1 86FFB7DC 90A6C08F 4DF435C9 34028492  36C3FAB4 D27C7026 C1D4DCB2 602646DE C9751E76 3DBA37BD
F8FF9406 AD9E530E E5DB382F 413001AE B06A53ED 9027D831  179727B0 865A8918 DA3EDBEB CF9B14ED 44CE6CBA CED4BB1B
DB7F1447 E6CC254B 33205151 2BD7AF42 6FB8F401 378CD2BF  5983CA01 C64B92EC F032EA15 D1721D03 F482D7CE 6E74FEF6
D55E702F 46980C82 B5A84031 900B1C9E 59E7C97F BEC7E8F3  23A97A7E 36CC88BE 0F1D45B7 FF585AC5 4BD407B2 2B4154AA
CC8F6D7E BF48E1D8 14CC5ED2 0F8037E0 A79715EE F29BE328  06A1D58B B7C5DA76 F550AA3D 8A1FBFF0 EB19CCB1 A313D55C
DA56C9EC 2EF29632 387FE8D7 6E3C0468 043E8F66 3F4860EE  12BF2D5B 0B7474D6 E694F91E 6DBE1159 74A3926F 12FEE5E4
38777CB6 A932DF8C D8BEC4D0 73B931BA 3BC832B6 8D9DD300  741FA7BF 8AFC47ED 2576F693 6BA42466 3AAB639C 5AE4F568
3423B474 2BF1C978 238F16CB E39D652D E3FDB8BE FC848AD9  22222E04 A4037C07 13EB57A8 1A23F0C7 3473FC64 6CEA306B
4BCBC886 2F8385DD FA9D4B7F A2C087E8 79683303 ED5BDD3A  062B3CF5 B3A278A6 6D2A13F8 3F44F82D DF310EE0 74AB6A36
4597E899 A0255DC1 64F31CC5 0846851D F9AB4819 5DED7EA1  B1D510BD 7EE74D73 FAF36BC3 1ECFA268 359046F4 EB879F92
4009438B 481C6CD7 889A002E D5EE382B C9190DA6 FC026E47  9558E447 5677E9AA 9E3050E2 765694DF C81F56E8 80B96E71
```

  - **On recherche le premier nombre p supérieur à cela, premier tel que (p-1)/2 soit lui-même premier** *(nombre de Sophie Germain)* **et dont les 64 bits de poids faible soient à 1**

  - **D'où le résultat** $2^{768} - 2^{704} - 1 + 2^{64} \times [149686+2^{638}\pi]$

La génération des nombres aléatoires

JLS CONSEIL

- **Indispensables pour les schémas de Diffie-Hellman**
- **Utilisés dans les protocoles IPSec**
- **Peuvent être utilisés pour créer des masques XOR**
- **Utilisés pour les générations automatiques de clés**
  - **L'exemple de PGP**

    La génération de nombres aléatoires introduit une faille dans la méthode

La génération des nombres aléatoires

JLS CONSEIL

- **La suite des aléas générés doit ressembler à une suite « au hasard »**

  Équiprobabilité de tous les motifs possibles

  Tests statistiques
- **La suite des aléas générés doit être parfaitement imprévisible pour le pirate**
- **Notion d'entropie des nombres aléatoires**

La génération des nombres aléatoires

JLS CONSEIL

- **Générateurs aléatoires purs :**
  basés sur un phénomène aléatoire
  - *Bruits de fond d'un circuit électronique*
  - *Diode Zener au point d'instabilité*
  - *Trafic sur un réseau informatique*
  - *Checksum de la mémoire vive*

- **Générateurs pseudo-aléatoires :**
  basés sur un algorithme mathématique
  parfaitement déterministes

Les générateurs pseudo-aléatoires

JLS CONSEIL

- **Notion d'automate fini**
  - **État de l'automate**
  - **Algorithme fournissant**
    - *Résultat pour calculer le prochain nombre aléatoire*
    - *Nouvel état*
  - **Périodique**

- **Exemple :** ***« The bad and dirty generator »***

  $x \leftarrow x \times 7^5 \;\; [\,Mod\ M_{31}]$

  $M_{31} = 2^{31}-1$: Nombre premier de Mersenne

- ***La connaissance d'un résultat permet de calculer intégralement tout le passé et tout le futur***

JLS CONSEIL

Les générateurs pseudo-aléatoires

- « Camoufler » l'état du générateur
  - *Appliquer un DES_k au résultat du générateur*
  - *Utiliser un shuffle*

- Mettre en parallèle plusieurs générateurs désynchronisés pour augmenter la périodicité
  - *Générateur de période très grande (> âge de l'univers)*
  - *Perturbation par des aléas vrais*

- Brevets Everbee sur les générateurs aléatoires

JLS CONSEIL

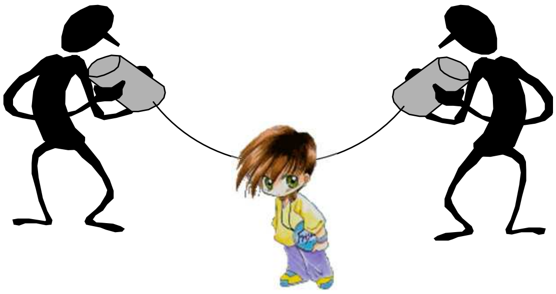
Diffie Hellman :
Échange de clé sur un réseau public

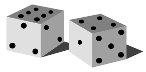
Peut être attaqué par le « man in the middle »

JLS CONSEIL

Diffie Hellman :
Échange de clé sur un réseau public

Peut être attaqué par le « man in the middle »



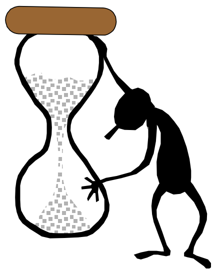


Parade : Sécuriser l'échange D-H par un chiffrement asymétrique

JLS CONSEIL

Complexité algorithmique

Temps de calcul en fonction du nombre n de bits des données



Algorithmes polynomiaux
- Linéaires (addition)
- Quadratiques (multiplication)
- Cubiques (exponentiation)

Algorithmes exponentiels
- Casser le Log discret par attaque brutale
  (essayer successivement tous les exposants)

Algorithmes subexponentiels
- $L_n(\gamma,c) = O(\exp(c\ n^{\gamma}\ \ln(n)^{1-\gamma}))$
- $\gamma=1$: exponentiels $\gamma=0$: polynomiaux

Log discret : $\gamma=1/2$ et depuis les théories du crible numérique de Lenstra (1993) $\gamma=1/3$

JLS CONSEIL

A propos des ordres de grandeur

Hypothèse : processeur à 1 GHz

- **1 seconde = 10^9 nanosecondes ≈ 2^{30} nanosecondes**
- **1 jour = 86400 secondes ≈ 2^{16} secondes.**
  - **1 an = 365.25 jours = 31 557 600 secondes ≈ 2^{25} secondes ≈ 2^{55} nanosecondes**
    - **Âge de l'univers ≈ 20 milliards d'années ≈ 6.10^{17} ≈ 2^{59} secondes ≈ 2^{89} nanosecondes**

JLS CONSEIL

A propos des ordres de grandeur

Application cryptographiques (processeur à 1GHz)

	Nombre de tops d'horloge	Log base 10	Log base 2
1 seconde	1.000E+09	9.000	29.897
1 jour	8.640E+13	13.937	46.296
1 an	3.154E+16	16.499	54.808
Grid pendant 1 jour (un milliard de CPU à 1GHz)	8.640E+22	22.937	76.193
Grid pendant 1 an (un milliard de CPU à 1GHz)	3.154E+25	25.499	84.705
Grid pendant 20 milliards d'années (un milliard de CPU à 1GHz)	6.307E+35	35.800	118.924
Grid pendant 20 milliards d'années (un milliard de CPU à 10TFlops)	6.307E+39	38.800	128.890

Rappel :

DES = 56 bits
Triple DES = 112 bits
Algo actuels = 128 bits

JLS CONSEIL

Accélérateurs d'attaques

On se propose de casser le Log discret

– **Déterminer a connaissant g^a [modulo N]**

- **Si on a le temps**
  - **Essayer tous les a possibles**
  - **En moyenne il faudra N/2 essais**
- **Si on a la mémoire**
  - **Calculer une fois pour toutes tous les g^a [modulo N]**
  - **Les stocker en mémoire (fichiers indexés…)**
  - **Pour chaque nouveau g^a consultation de table**

JLS CONSEIL

Accélérateurs d'attaques

On se propose de casser le Log discret

– **Déterminer a connaissant g^a [modulo N]**

- **Attaque mixte**
  - **Précalculer et stocker g^a [modulo N]**
    pour T valeurs de a équiréparties ***(distantes de S avec S.T=N)***
  - **Etant donné un g^a avec a inconnu,**
    s'il n'est pas dans la table, multiplier par g et réitérer
    jusqu'à trouver un élément présent dans la table
    En moyenne S/2 calculs
  - ***Exercice : Quel est le γ?***
- **Exemple numérique : $N \approx 10^{18} \approx 2^{60}$**
  - **Précalculer et stocker 10^9 valeurs : Quelques gigas de mémoire**
  - **En moyenne un demi milliard de calculs suffisent…**
- **Les ressources nécessaires au pirate croissent comme la racine carrée de N**

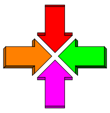
Applications du Logarithme Discret

- **Il est impossible d'inverser en un temps raisonnable l'exponentiation modulaire**

  Protocoles d'échanges de clés (Diffie Hellman)

  Chiffrement asymétrique (El Gamal)

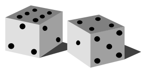
  Protocoles d'authentification



Système de El Gamal

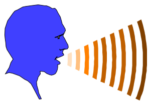
- **Système de chiffrement asymétrique**

  Tout le monde peut envoyer un message secret à A

  N et g sont publics ***(N premier, g générateur de Z/NZ)***

  A choisit a aléatoire secret et publie g^a [Mod N]

  – **g^a [Mod N] est la clé publique de A**

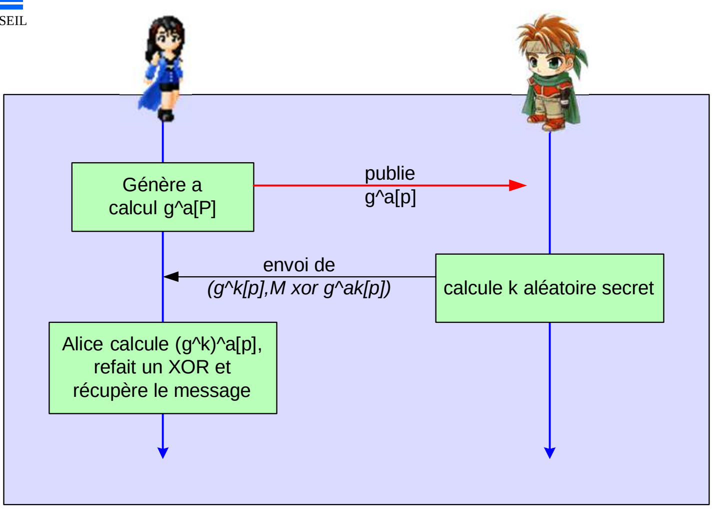
  Pour envoyer un message M à A,
  B génère un k aléatoire secret, et envoie
  (g^k[Mod N], M ⊕ g^{ak}[Mod N])

  A calcule $(g^k)^a$ [Mod N] et refait un ⊕ pour retrouver M

- **Le pirate connait g [Mod N], g^k [Mod N] et g^a [Mod N] mais il n'a aucun moyen de retrouver g^{ak} [Mod N]**

Système de El Gamal

JLS CONSEIL



© Aurélien Leteinturier

JLS CONSEIL

Un protocole d'authentification

- **Protocole de défi/réponse** **(Similaire à El Gamal)**
  - **N et g sont publics** ***(N premier, g générateur de Z/NZ)***
  - **A choisit a aléatoire secret et publie g^a (Mod N)**
    - **a est la clé secrète de A**
    - **g^a (Mod N) est la clé publique de A**
  - **Pour authentifier A,**
    - **B génère un k aléatoire secret,**
    - **calcule g^k**
    - **envoie g^k à A (c'est le *défi*)**
  - **A calcule $(g^k)^a$ et le renvoie à B (c'est la *réponse* au défi)**

  - **Le pirate connait g , g^k [Mod N] et g^a [Mod N]**

mais il n'a aucun moyen de retrouver g^{ak} [Mod N]

- **Seul quelqu'un connaissant a pouvait répondre correctement au défi**

Protocole de Défi / Réponse

JLS CONSEIL

- **Pour authentifier A, on lui envoie un défi**
- **Seul quelqu'un connaissant la clé secrète de A peut calculer rapidement la réponse**
  - **Le calcul de la réponse nécessite une exponentiation modulaire**
- **Un pirate ne connaissant pas la clé secrète doit faire un calcul très long**
  - **Il faut résoudre le logarithme discret**
- **A dispose d'une puissance de calcul limitée**
  - **carte à puce, carte SIM de téléphone mobile**
- **Le pirate dispose de moyens très importants**
  - **organisation criminelle puissante**

Bonnes fonctions à sens unique ?

JLS CONSEIL

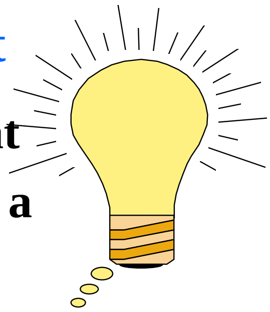
- **C'est un des enjeux des recherches actuelles**
  Fonction directe rapide à calculer sur des processeurs à très faible puissance
  Fonction inverse impossible à calculer même avec des ressources puissantes
- **En tenant compte des possibles évolutions de la technologie et des puissances de calcul**
- ***Utilisation de courbes elliptiques sur un corps fini***
- ***Groupes de Jacobi des courbes hyperelliptiques***

Théorème de Fermat

JLS CONSEIL

« Petit » théorème de Fermat

Pour p premier, $a \neq 0$ [p] , on a



$$a^{p-1} \equiv 1 \ [p]$$

Démonstration : *L'ordre du sous-groupe des puissances de a divise l'ordre du groupe multiplicatif des éléments inversibles dans Z/pZ, qui a p-1 éléments car p est premier.*

Application du théorème de Fermat

JLS CONSEIL

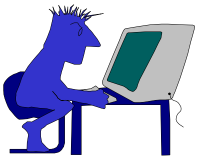
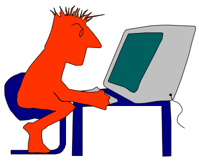
- **Pour c premier à p-1, on calcule d inverse de c modulo p-1 (Bezout)**
- **Pour $cd \equiv 1$ [p-1] on a, pour tout a** (y compris 0 [p])
  $$(a^c)^d \equiv (a^d)^c \equiv a \quad [p]$$
- *Deux exponentiations modulaires réciproques l'une de l'autre*

$$\begin{tikzcd} x \arrow[r, bend left, "f"] & y \arrow[l, dotted, bend left, "g = f^{-1}"] \end{tikzcd}$$

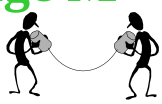
f : élever à la puissance c
g : élever à la puissance d

Système de Massey - Omura
Initialisation

- **p premier public très grand (> 10^{120})**
- **A choisit c_A et d_A secrets avec $c_A.d_A \equiv 1$ [p-1]**
- **B choisit c_B et d_B secrets avec $c_B.d_B \equiv 1$ [p-1]**

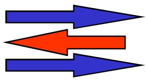
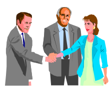
Système de Massey - Omura
Transmission d'un message M



- A calcule et transmet M^{c_A} ,
- B l'élève à la puissance c_B et renvoie $M^{c_A c_B}$
- A l'élève à la puissance d_A obtient $M^{c_A c_B d_A} \equiv M^{c_B}$ qu'il transmet
- élève à la puissance d_B et retrouve M

N.B. : Tous les calculs sont modulo p

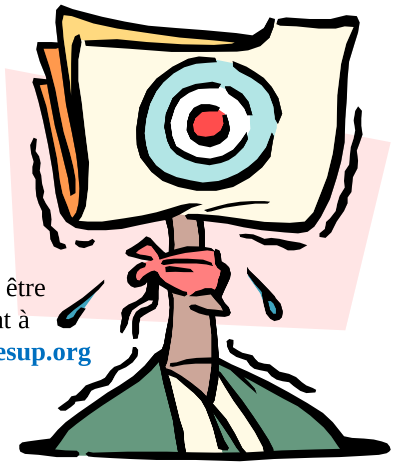
- **Trois échanges**

- **nécessite une *authentification préalable***
- **Protocole de valise à deux cadenas**

Questions?

JLS CONSEIL



Les questions peuvent être adressées directement à **jean-luc.stehle@normalesup.org**