Grenoble INP ENSIMAG

Sécurité: cryptographie

Menaces et solutions
Chiffrement
Authentification, certificats
Signature, Mots de passe
(Ssh et SSL)

Contenu du chapitre Crypto

- Terminologie, menaces et remèdes
- Chiffrement
  - Cryptographie classique dans Z/26Z: substitution, permutation
  - Chiffrement symétrique (clé secrète)
  - Chiffrement asymétrique (clé publique)
  - Gestion des clés, création de clé de session (Diffie-Hellman)
- Authentification
  - Secret partagé (mot de passe, multifacteurs)
  - par clé publique
  - Certificats
- Hachage cryptographique, signature
- Applications:
  - Stockage de mots de passe
  - ssh, SSL

Menaces & solutions

Ennemi
cote 500

Lieutenant X

Chef de bataillon

Lieutenant Y

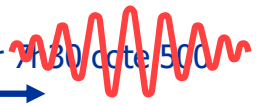
1 Lt X: Artillerie demandée pour 7h30 cote 500

2 *Lt X*: Nous contrôlons la situation

3 Lt X: Artillerie demandée pour 8h cote 200

4 Lt Y: Artillerie demandée pour 7h30 cote 500

5 Cdt: Exécutez les prisonniers



6 Lt X: Artillerie demandée pour 7h30 cote 500

Attaque	Propriété
Ecoute	Confidentialité
Usurpation	
Corruption	
Rejeu, Faux émetteur	
Dénégation	
Brouillage	

Menaces & solutions

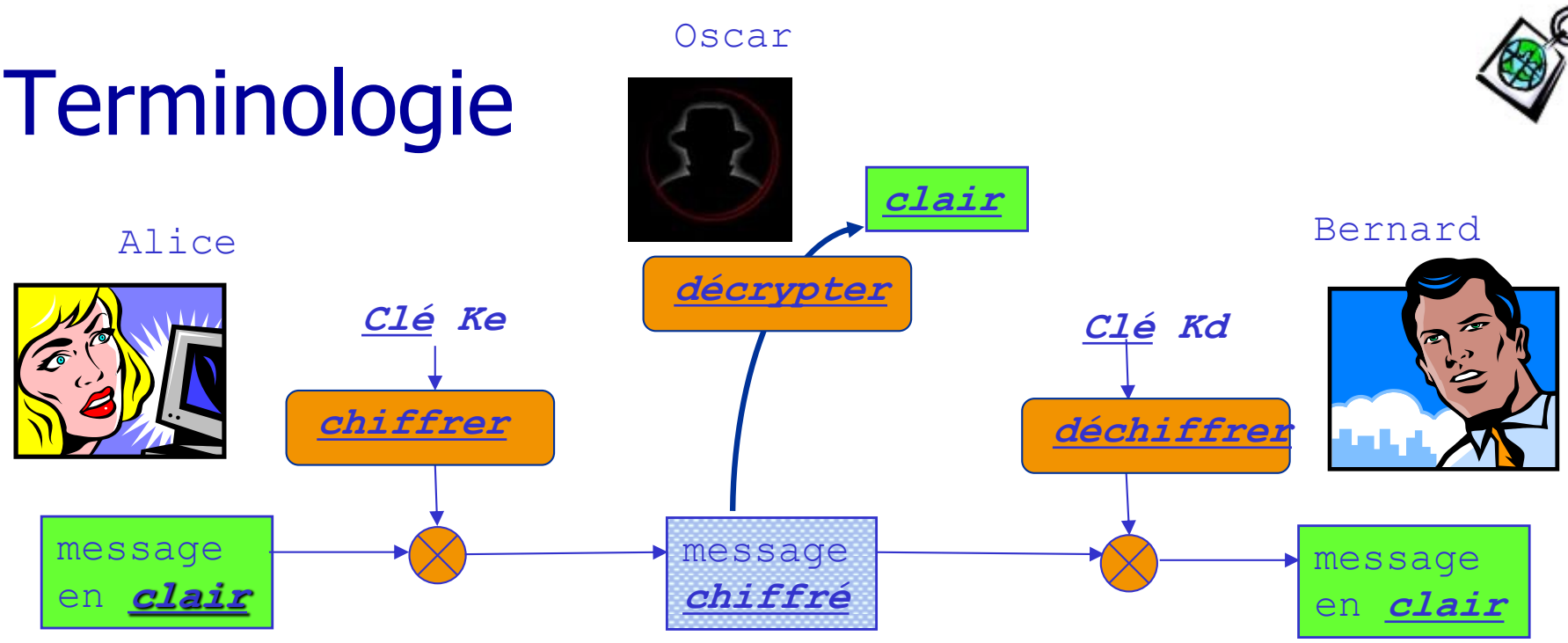
Attaque	Propriété	Solution	Méthodes
Ecoute	Confidentialité	Chiffrement	Algos crypto
Usurpation			
Corruption			
Rejeu, Faux émetteur			
Dénégation			
Brouillage			

Grenoble INP ENSIMAG

Chiffrement

Les principes

Terminologie



- Chiffrer (~~crypter~~) Déchiffrer

  $C=E_{Ke}(M)$ $M=D_{Kd}(C)$

- Cryptanalyse: décrypter sans connaître la clé Kd
  - Attaque à texte chiffré connu: C
  - Attaque à texte clair connu: couples (M,C), trouver Kd
  - Attaque à texte clair choisi: Oscar peut fournir M et observer C

Chiffrement: Substitution monoalphabétique (ou monosymbolique)

- Substitution E: bijection de {A..Z} -> {A..Z}
  - Chaque lettre x remplacée par E(x)
  - {A..Z} assimilé à Z/26Z
- Chiffrement par décalage
  - E(x) = x + d (mod 26)
  - Par ex: (d=3): JULESCESAR -> MXOHVFHVDU
- Chiffrement affine
  - E(x) = a x + b (mod 26) (a inversible, premier avec 26)
  - M = D(C) = a^{-1}(C-b) (mod 26)
- Substitution quelconque (clé=26 lettres)
  - Lettres: `ABCDEFGHIJLKMNOPQRSTUVWXYZ`
  - Clé: `UCPKIAVZBHWRXYDLGOJNELFQMS`

Cryptanalyse

- Force brute (recherche exhaustive)
  - Chiffrement par décalage: trivial
    - 25 essais max, 13 en moyenne
  - Substitution quelconque: impraticable
    - $26! \sim 4.10^{26} \sim 2^{90}/3$
    - Ex: machine faisant 10^9 opérations/sec.
      - 2^{46} op/j, 2^{55} op/an, 2^{90} op/15 milliards ans (âge univers)
- Analyse Fréquentielle (*Al-Kindi, IXè siècle*)
  - Fréquence des lettres en français: ESAINTURLO…
  - E: 18%, S: 9%, A: 7,6%, I: 7,5% … K = 0,05%
  - Très efficace pour décrypter substitution (polynomial, quasi linéaire)

Plus généralement, tout biais de répartition est source d'attaques

Substitution polyalphabétique

- Chiffrement (par le carré) de Vigenère (1586)
  - Décalage différent pour chaque lettre du clair
  - Décalages définis par mot clé, de longueur N, répété

Clair	**I**	**L**	**A**	**D**	**O**	**R**	**E**	**L**	**E**	**S**	**C**	**O**	**U**	**R**	**S**	**D**	**E**	**R**	**E**	**S**	**E**	**A**	**U**	**X**
Clé	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*
Décalage	4	13	18	8	12	0	6	4	13	18	8	12	0	6	4	13	18	8	12	0	6	4	13	18
Chiffré	M	Y	S	L	A	R	K	P	R	K	K	A	U	X	W	Q	W	Z	Q	S	K	E	H	P

- Résiste 3 siècles, décrypté par Babbage (1854) & Kasiski (1863)
  - Phase 1: trouver la longueur de la clé (N=|clé|) par trigrammes
    - Trigrammes répétés du chiffré très probablement alignés avec ceux du clair
    - Espacement=multiple de N => trouver N comme PGCD des espacements
  - Phase 2: analyse fréquentielle sur chacun des N sous-textes
    sous-texte$_r$(i)=texte(r+N.i)
- Généralisé par chiffrement de Vernam (one-time pad)

|clé|: longueur de clé

Transposition (permutation)

- Les symboles (lettres) du texte d'origine sont permutées. Exemple: rang=ordre alphabétique des lettres de la clé

Clair	**I**	**L**	**A**	**D**	**O**	**R**	**E**	**L**	**E**	**S**	**C**	**O**	**U**	**R**	**S**	**D**	**E**	**R**	**E**	**S**	**E**	**A**	**U**	**X**
Clé	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*	*I*	*M*	*A*	*G*	*E*	*N*	*S*
Rang	2	6	7	4	5	1	3	2	6	7	4	5	1	3	2	6	7	4	5	1	3			
Chiffré	R	I	E	D	O	L	A	U	L	R	C	O	E	S	S	S	E	R	E	D	A	A	U	X

  - Déjoue l'attaque fréquentielle simple, mais cassable par analyse digrammes, et recherche longueur clé.
- Améliorable par permutations multiples etc.

Les chiffrements modernes combinent souvent substitution et permutation,

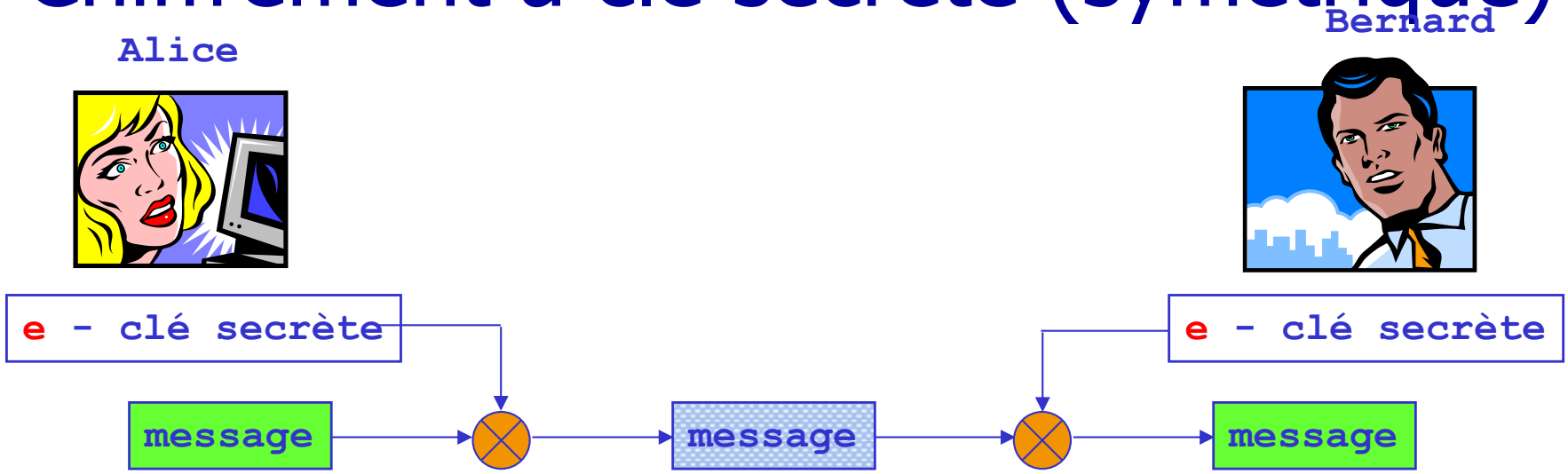
dans $Z/2^nZ$ (blocs de n bits)

Grenoble INP ENSIMAG

Chiffrement
Principaux algorithmes

Chiffrement symétrique: flot, bloc
Chiffrement asymétrique (à clé publique)

Chiffrement à clé secrète (symétrique)



- Algorithme à clé secrète (DES, AES,…)
  - **message chiffré c = $f_e(m)$**
  - **message déchiffré = $f_e^{-1}(c)$**
- Les deux côtés doivent connaître la clé
- Chiffrement efficace

Principaux algorithmes symétriques

- Chiffrement par flot: traité bit par bit (substitution par XOR)
  - (Vernam), RC4, A5/1
  - Génération flux pseudo-aléatoire à partir de la clé
- Chiffrement par bloc: m coupé en blocs de bits
  - DES, 3DES, AES, RC6…
  - Taille du bloc: ex 64 (DES) ou 128 bits (AES)
  - Taille de clé: ex 56 (DES), 128-256 bits AES (force brute > 2^{128} op)
    - A partir clé -> génération de t sous-clés
  - Algos itératifs: répétition t tours, paramétrés par sous-clé, combinant substitutions et permutations

Chiffrement de Vernam (masque jetable)

- Gilbert Vernam, 1917 + J. Mauborgne (aléa)
- ~Vigenère avec |clé|=|message| et clé aléatoire
- Informatique: XOR entre bits clé et bits mess.

Clair	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**
⊕ *Clé*	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*0*	*0*	*1*	*1*	*1*	*0*	*1*	*1*	*0*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*0*
Chiffré	0	1	1	1	1	0	1	1	0	0	1	0	0	0	0	0	1	1	0	1	1	1	0	1

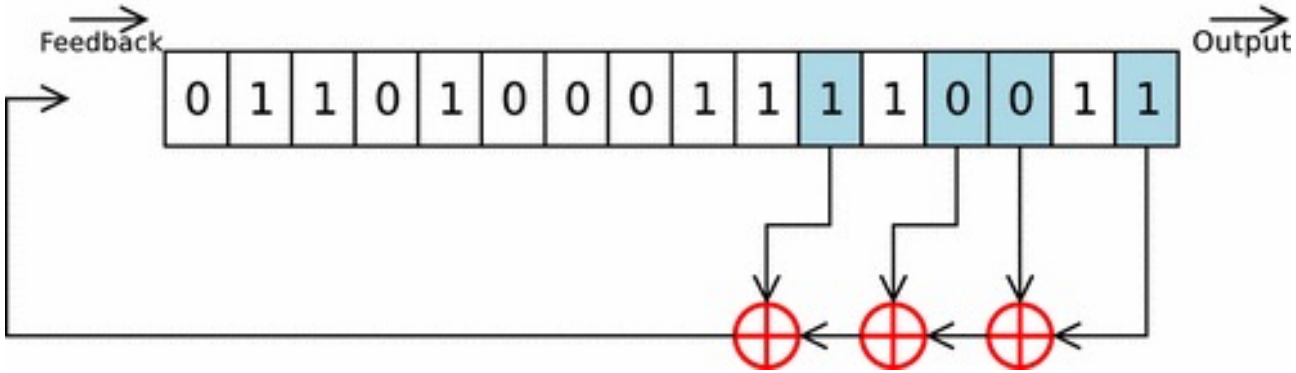
- Prouvé sûr (indécryptable sans la clé) si:
  - Clé aléatoire, aussi longue que le texte, jamais réutilisée
  - Vrai aléa produit par des processus physiques (ex: bruit thermique, phénomènes quantiques)
- Difficulté de mise en œuvre (clé des 2 côtés)

Génération de clés de substitution

- Chiffrement Vernam (clé à usage unique)
  - sécurité parfaite, si aléa parfait
  - Pb: clé très longue, en 2 copies (exp-dest)
  - lourd: réservé aux ambassades et services secrets
- Approximation: suite pseudo-aléatoire
  - engendré par calcul « chaotique » à partir de clés paramètres (ou morceaux d'une clé longue)
  - cycle (taille de la clé de substitution) très long en combinant des sous-cycles de longueurs premières entre elles

Electronique binaire: Registres à décalage

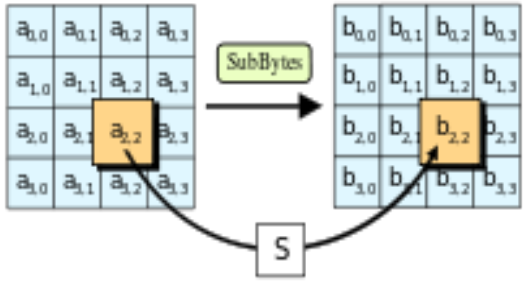
- Chiffrement par flot: on crée une suite pseudo-aléatoire de bits par un registre à décalage initialisé par une clé.



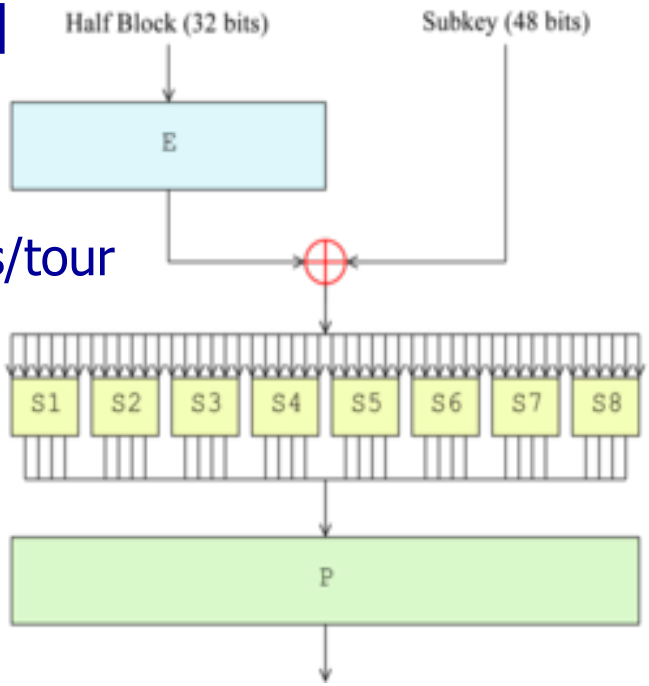
- Les bits sortants sont additionnés au texte (~Vernam avec pseudo-aléa).
- Ex: algo de chiffrement A5/1 de téléphonie GSM combine 3 registres (19, 22, 23 bits)
- Base mathématique: polynômes sur corps finis

Chiffrement par blocs: DES et AES

- AES : standard actuel (oct 2000)
  - Advanced Encryption Standard www.nist.gov/AES
  - Par blocs : 128 bits (16 octets).
  - Clef secrète : 128, 192 ou 256 bits
  - Corps à 256 éléments : F256



- DES [1972, IBM] Data Encryption Standard
  - Exemple : Crypt unix
  - Principe : bloc de 64 bits , clef de 64 bits
    - clef 64 bits => 56 bits utiles + 16 transformations/tour
  - Attaque brutale : 2^56 = 64.10^15
    - 1000 PCs * 10^9 Op/s * 64000s = 20h !!!!
- 3DES (1998, IBM)
  - Enchaîne 3 DES, avec 2 ou 3 clés différentes



Chiffrement à clé publique

e clé de Bernard
connue d'Alice

e - clé publique

d - clé privée

m

c

message

message

message

En rouge:
secret

- Chiffrement inversible: $c = f(e,m)$, $m = g(d,c)$
- Fonction mathématique f telle que connaître $f(e,.)$ ne permet pas de déduire f^{-1} i.e. $g(d,.)$.
  - Plus précisément: f et g connus, mais on ne peut retrouver d à partir de e.

    une clé de chiffrement (propre à Bernard mais rendue) publique
    une clé de déchiffrement (propre à Bernard mais rendue) secrète
- Chiffrement lent: opérations mathématiques nécessitant des calculs complexes

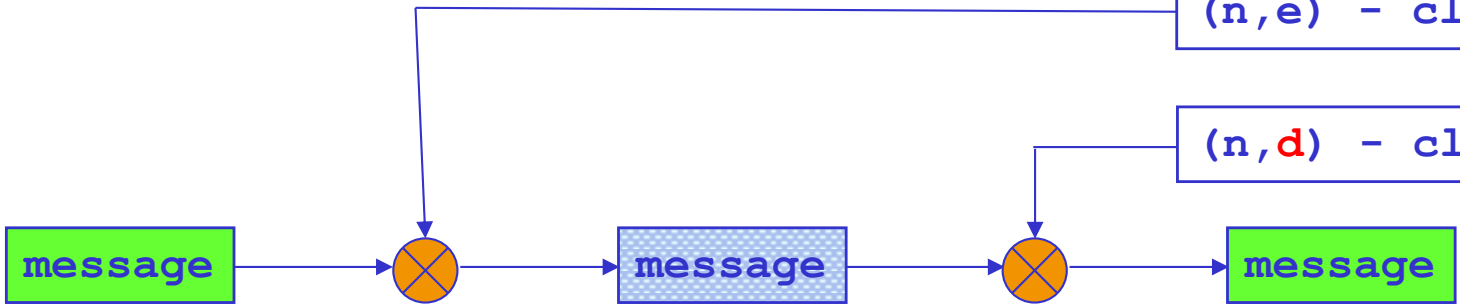
Bases arithmétiques et complexité

pour le chiffrement à clé publique

- Base: corps finis, en particulier $\mathbf{Z}/p\mathbf{Z}$
  - p premier, très grand (ex. nombre à 1024 bits, donc 309 chiffres décimaux)
- Problème « facile » sur nombre n de k bits: si $\exists$ algo de complexité polynomiale en k
  - n est-il premier ? (tests de primalité)
  - calculer $a \times b \bmod n$, $a^b \bmod n$
- Problème « difficile »: algos exponentiels en k
  - factoriser n (*cf algo RSA*)
  - logarithme discret (*algo El Gamal*): connaissant a premier avec n et b, trouver x tel que $b = a^x \bmod n$

Algorithme RSA (1977)

(n,e) clé de Bernard
connue d'Alice



- Algorithme RSA : p & q premiers, n=p.q, z=φ(n)=(p-1).(q-1); e premier avec z; e.d = 1 mod z
  - **message chiffré c = (m^e mod n)**
  - **message déchiffré m = (c^d mod n)**
- Propriété fondamentale: **$(m^e)^d$ mod n = m**
  - Selon théorème Euler-Fermat: $a^n=1 \Rightarrow a^{\varphi(n)}=1 \bmod n$

Connaître n ne permet pas de trouver p et q

Chiffrement d'El Gamal (1984)

Repose sur logarithme discret dans un groupe cyclique G, de générateur g

- Bernard:
  - Choisit **b** aléatoire, qui sera sa clé privée (secrète)
  - Calcule : $B=g^b$ et publie (G, g, B) = clé publique
- Chiffrement par Alice d'un message m :
  - Choisit **a** aléatoire ("nonce"= Number Once)
  - Calcule $c_1=g^a$ et $c_2=m.B^a$ (multiplication dans le groupe G)
  - Envoie à Bernard le chiffré $E_A(m)=(c_1, c_2)$
- Déchiffrement par Bernard
  - $D(c_1, c_2) = c_2.c_1^{-b} = m.g^{ba}.\ g^{-ab} = m$

Chiffrement

- Chiffrement symétrique : clés secrètes
  - combinent substitutions (avec registres à décalages) et permutations
  - facilement implantables en matériel: algorithmes rapides
  - adapté au chiffrement des flux d'informations
  - Exemples d'algorithmes: 3DES, AES, A5/1, A5/3
- Chiffrement asymétrique: RSA, courbes elliptiques
  - une clé de chiffrement (propre à Bernard mais rendue) publique
  - une clé de déchiffrement (propre à Bernard mais rendue) secrète
  - algorithmes lents
  - adapté à l'établissement de connexion, et à l'authentification

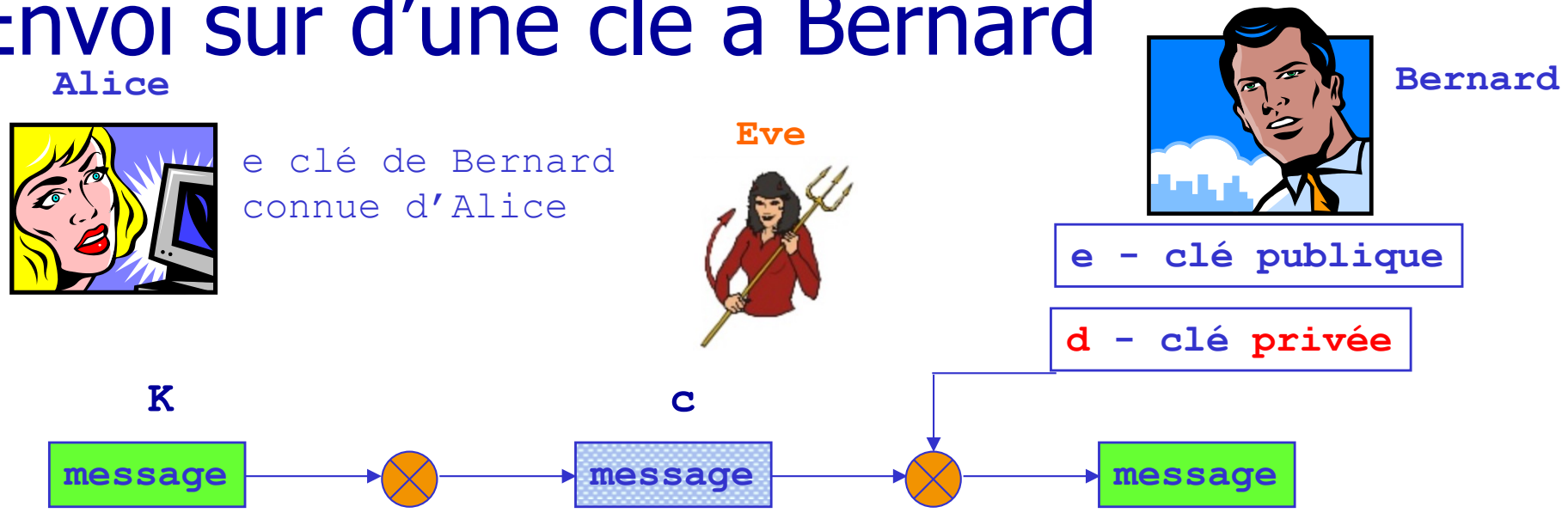
Gestion des clés

- La sécurité du chiffre repose sur celle des clés

- Clé secrète: il faut faire parvenir la clé aux interlocuteurs
  - Porteur spécial (militaires, ambassades…)
  - Envoyer de nouvelles clés par le canal chiffré avec les anciennes clés (tant que celles-ci ne sont pas compromises) ou par un autre canal chiffré
  - Alice peut envoyer une clé secrète avec K_{pub}(Bernard)
  - **Création en ligne de clé secrète: voir Diffie-Hellman**

- Clé publique:
  - Certificats (cf plus loin), gérés au sein d'une IGC: Infrastructure de Gestion des Clés (PKI en anglais)

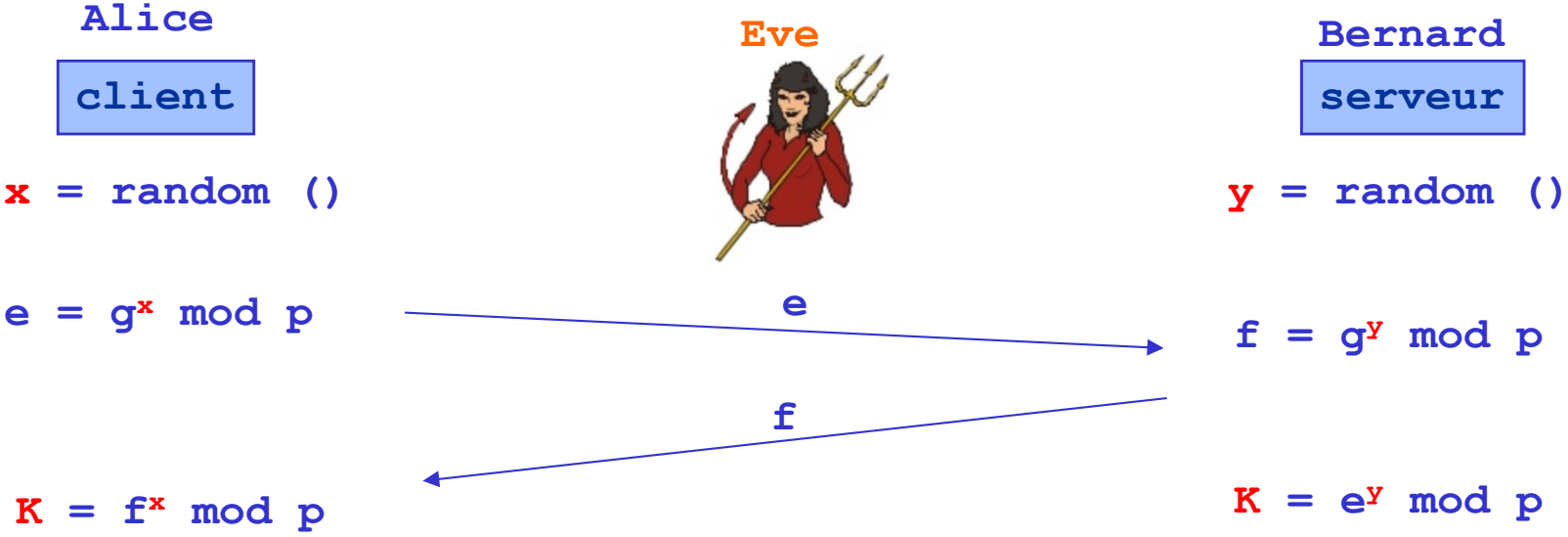
Envoi sûr d'une clé à Bernard



- Alice envoie : `c = f(e,K)`
- Bernard peut retrouver `K= g(d,c)`
- Eve ne peut retrouver `K` même si elle connaît `e`, car elle ne connaît pas `d`

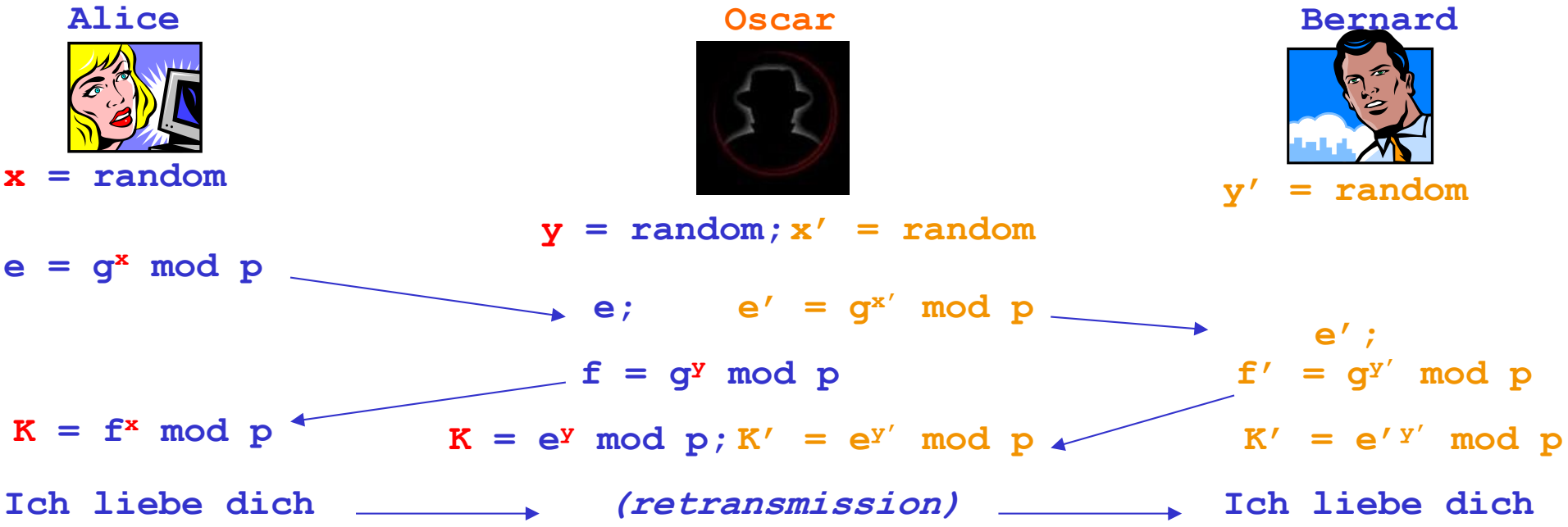
Mais si Alice ne connaît pas la clé `e` de Bernard, et n'a aucun canal confidentiel avec Bernard, comment peuvent-ils s'échanger une clé secrète sans qu'Eve ne puisse la connaître ?

Création de clé secrète (Diffie-Hellman 1976)



- **Paramètres publics (envoyés avec e)**
  - **p: grand nombre premier**
  - **g: générateur de Z/pZ**
- On choisit **x** et **y** tels que: $1 < x, y < (p - 1) / 2$
- Clé $K = g^{xy} \bmod p$
  $= (g^x \bmod p)^y \bmod p = (g^y \bmod p)^x \bmod p$

Attaque MitM (homme-au-milieu)



- Eve ne fait qu'écouter. Mais Oscar peut intercepter les messages et en injecter.
- Oscar peut se faire passer pour Bernard auprès d'Alice, et pour Alice auprès de Bernard
- Il crée une clé secrète avec chacun
- Il retransmet ensuite tous les messages: Alice a l'impression de parler secrètement à Bernard, mais en fait Oscar voit tout passer.
- Solution: *authentification*

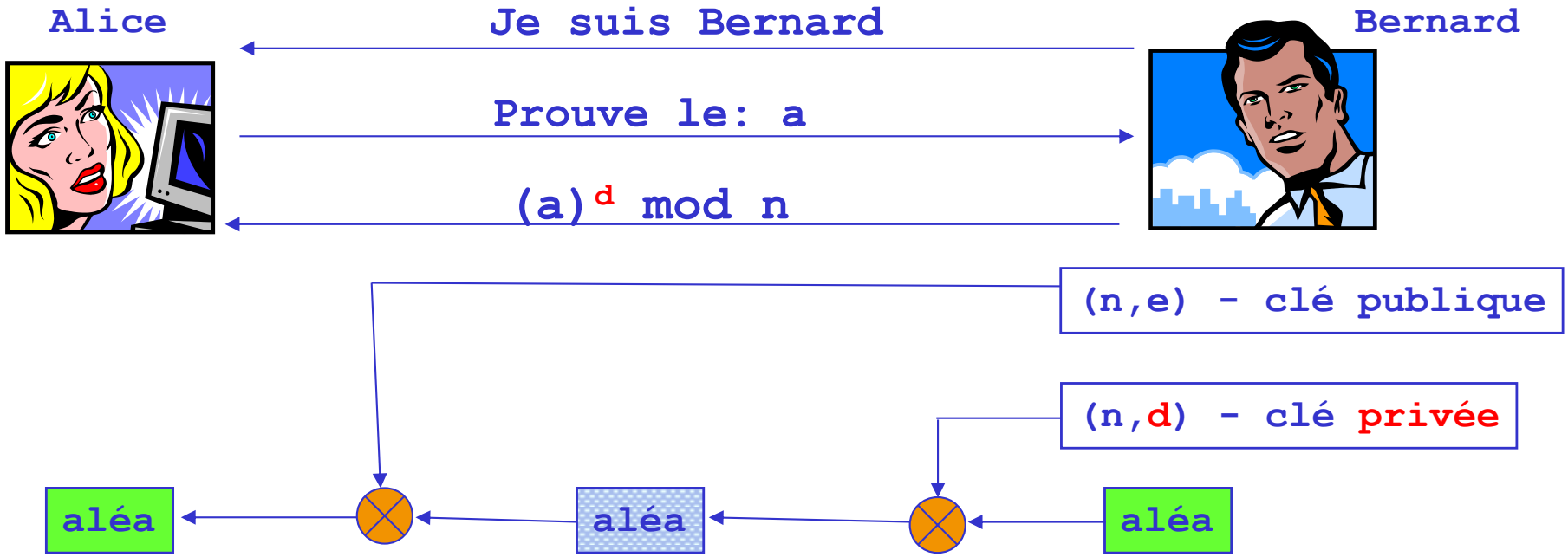
Contenu du chapitre Crypto

- Terminologie, menaces et remèdes
- Chiffrement **confidentialité**
  - Cryptographie classique dans Z/26Z: substitution, permutation
  - Chiffrement symétrique (clé secrète)
  - Chiffrement asymétrique (clé publique)
  - Gestion des clés, création de clé de session (Diffie-Hellman)
- Authentification **Légitimité (authenticité)**
  - Secret partagé (mot de passe, multifacteurs)
  - par clé publique
  - Certificats
- Hachage cryptographique, signature **intégrité**
- Applications:
  - Stockage de mots de passe
  - ssh, SSL

Authentification classique

- Secret partagé
  - Mot de passe
    - Sensible au rejeu
  - (Défense contre rejeu): défi-réponse
    - Chef de bataillon à LtX: nombre `a` aléatoire
    - LtX répond: `f(e,a)` où `e` secret partagé
    - Le chef de bataillon compare avec son propre calcul
- Authentification multifacteur, canaux séparés
  - LtX contacte son chef par radio; le chef rappelle sur téléphone fixe
  - Bancaire: demande virement par Internet; confirmer avec code reçu par SMS

Authentification à clé publique



- Alice envoie un défi aléatoire `a`, à usage unique (nonce)
- Bernard le chiffre avec `d`: $(a)^d \bmod n$
- Alice vérifie que $(a^d)^e \bmod n = a$

Notations: protocoles cryptographiques (Dolev-Yao 1983, BAN 89)

Alice

```
B->A:  {B}
A->B:  {N}

B->A:  {B;N}Kd(B)
```

Bernard

- {m} : Message portant l'information m
  - N : nonce (nombre aléatoire utilisé 1 seule fois)
- $\{m\}_K$: message m chiffré avec la clé K
- Ke(B) : clé de chiffrement de l'acteur B; aussi notée K_B^+
- Kd(B) : clé de déchiffrement de l'acteur B; notée K_B^-
  - *e : encryption; d = decryption*
- K_{AB} : clé symétrique partagée entre A et B.
- On peut composer: $\{\{m;A;date\}_{Ke(A)} ; B\}_{Kd(B)}$

Authentification de serveur Web

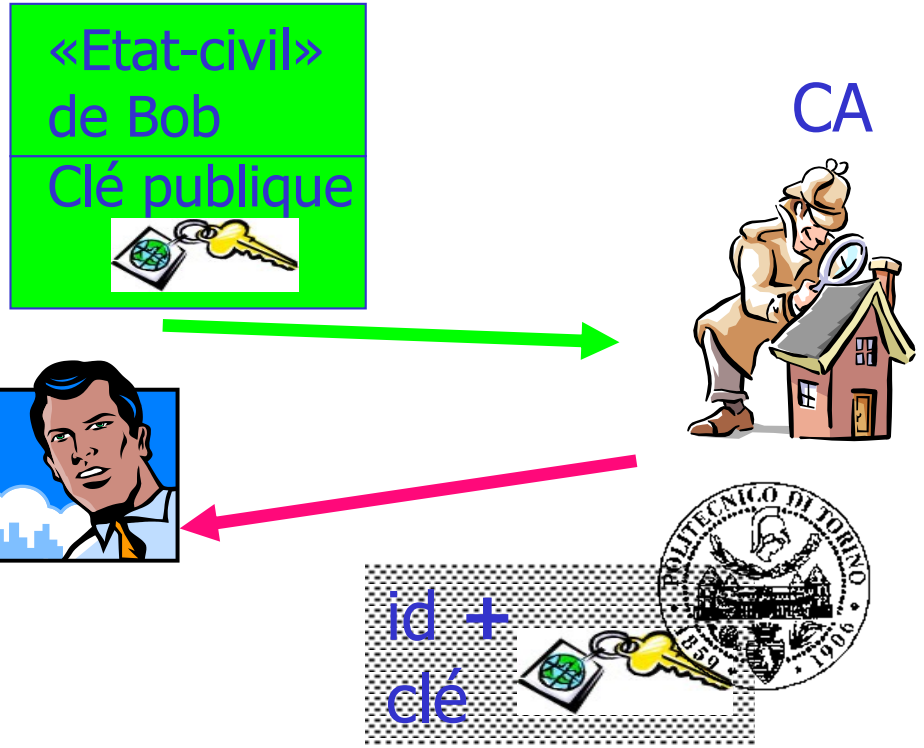
- Le navigateur (~Alice) doit s'assurer de l'identité du serveur (~Bob: c'est aux USA :-)

Pb: comment être sûr de la clé publique de Bob ?

- Notion de certificat:
  - des organismes appelés « Autorités de Certification » (CA) vont authentifier des clés publiques
  - Bob enverra à Alice le certificat, confirmation de l'association entre son identité et sa clé publique, le tout signé avec la clé privée de CA.

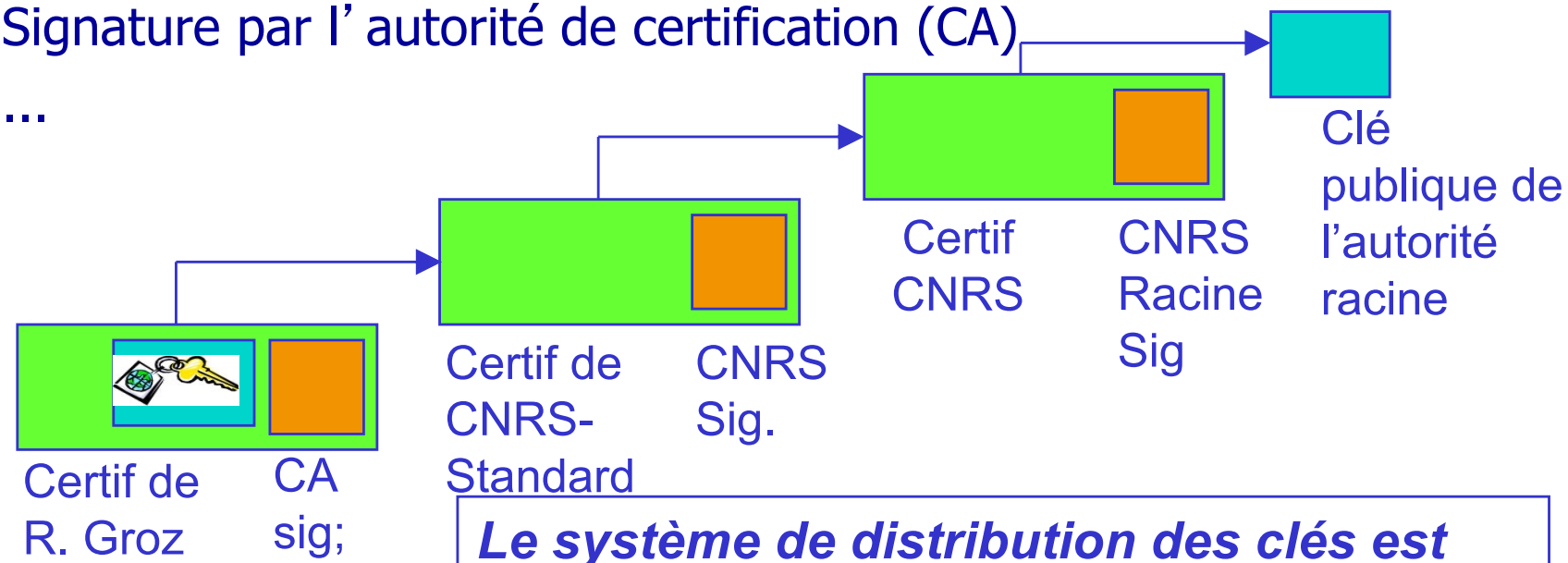
Authentification par certificats

- L'autorité (CA) applique sa signature (clé privée) à un couple: (Identité de Bob, Clé pub. de Bob)
- Bob envoie ce certificat à Alice lorsqu'elle le contacte
- Alice vérifie la signature du certificat avec la clé pub. du CA: elle est sûre de la clé publique de Bob.



Chaînes de certificats

- Une CA peut elle-même être certifiée par une CA de plus haut niveau
- Contenu de chaque certificat (standard X.509)
  - No de série (unique pour ce certif), algo de signature
  - Nom du détenteur et sa clé publique
  - Signature par l'autorité de certification (CA)
  - ...



Le système de distribution des clés est appelé PKI (Public Key Infrastructure)

Contenu du chapitre Crypto

- Terminologie, menaces et remèdes
- Chiffrement **confidentialité**
  - Cryptographie classique dans Z/26Z: substitution, permutation
  - Chiffrement symétrique (clé secrète)
  - Chiffrement asymétrique (clé publique)
  - Gestion des clés, création de clé de session (Diffie-Hellman)
- Authentification **Légitimité (authenticité)**
  - Secret partagé (mot de passe, multifacteurs)
  - par clé publique
  - Certificats
- Hachage cryptographique, signature **intégrité**
- Applications:
  - Stockage de mots de passe
  - ssh, SSL

Hachage cryptographique

- m: message aussi long qu'on veut, dans {0,1}*
- H {0,1}* -> $\{0,1\}^n$
  - H fonction non inversible (car non injective)
  - Pré-image difficile à calculer
  - Difficile de trouver m' tel que H(m')=H(m) (résistance aux collisions)
- H(m): *empreinte* de m (ou « *condensat* »)

- Algorithmes crypto de hachage: [MD5 (128 bits), SHA-1 (160 bits), trop faibles], SHA-256 (256 bits)…

Utilisation du hachage

- Intégrité: Message Authentication Code (MAC)
  - Alice envoie {m;H(m,s_{AB})}, où s_{AB} = *secret partagé*
  - H(m,s) = H(f(m,s)), f combine m & s (concaténation, addition..)
- Intégrité par Signature (avec *clés asymétriques*)
  - Alice envoie {m; $\{H(m)\}_{Kd(A)}$}
- Vérification qu'une clé publique est la bonne
  - Cf ssh et TP.
- Stockage de mot de passe p
  - Stocker H(p), et comparer H(saisie) avec H(p), cf plus loin
- ...

Contenu du chapitre Crypto

- Terminologie, menaces et remèdes
- Chiffrement **confidentialité**
  - Cryptographie classique dans Z/26Z: substitution, permutation
  - Chiffrement symétrique (clé secrète)
  - Chiffrement asymétrique (clé publique)
  - Gestion des clés, création de clé de session (Diffie-Hellman)
- Authentification **Légitimité (authenticité)**
  - Secret partagé (mot de passe, multifacteurs)
  - par clé publique
  - Certificats
- Hachage cryptographique, signature **intégrité**
- Applications:
  - Stockage de mots de passe
  - ssh, SSL

Stockage des mots de passe sur serveur

- Ne jamais stocker les mdp (p) en clair !
  - Stocker une version chiffrée: $c_{ref} = f(p_{ref})$
  - Lors de l'authentification, comparer $f(p_{saisi})$ et c_{ref}
- Stockage de hachage crypto: $f=H$?
  - Pré-image difficile => on ne peut retrouver le mdp
  - Résistance aux collisions: mais
    - Danger si deux comptes ont le même mdp
    - Ou si nombreux c_{ref} disponibles
- Solution: salage; hachage(mdp+sel)
  - A la création du compte, on calcul sel (nombre) aléatoire.
  - Table stocke: $(id, sel, H(p_{ref}+sel))$

  Unix stocke $sel, H(p_{ref}+sel)$ dans une même chaîne

Connexions sur une machine distante

- Login sur une machine distante
  - Pour pouvoir y lancer des commandes: connexion à un « shell »
  - Solution sécurisée: la commande ssh
- Connexion à des applications en réseau:
  - Messagerie (BAL distante)
  - Site Web
  - Solution sécurisée: protocole cryptographique TLS

Cf cours IRC: AP_cnx

En guise de conclusion

Menaces sur la société du numérique

- Hacker fou: anecdotique (mais médiatisé)
- Cyber-criminalité
  - cible entreprises: VPC, données sensibles (hôpitaux…)
  - pas toujours médiatisée (iceberg: attaques Amazon, Sony) mais très importante
  - sociétés de « protection », chantage etc
  - attaque des particuliers: 1. pour escroquerie ou chantage; 2. pour enrôler dans botnets
- Etats (domaine militaire) + terrorisme
  - Russie-Estonie (avril 2007), blackout Ukraine (déc 2015)
  - Corée du Nord (+100 hackers/an, attaque sur Sony 11/2014)
  - Attaque terroriste sur TV5 Monde (avril 2015)
  - Attaques par virus (Stuxnet, Duqu, centres drones US...)
  - USA: NSA et écoutes
  - Influence d'opinion, désinformation (via réseaux sociaux)…

Terminologie des maliciels

- Virus: pgm capable de se recopier lors d'un échange entre machines (ou via support amovible)
  - Contient « charge utile (malicieuse) » et code pour se reproduire et reconnaître et s'insérer dans le système
  - Mutations et virus polymorphes (pour échapper à la reconnaissance par anti-virus)
- Ver (Worm): forme de virus, code auto-réplicant se propageant (spontanément) par le réseau
- Cheval de Troie (Trojan): programme « cadeau » proposé au téléchargement et remplissant une fonction utile – mais cachant des fonctions nocives
- Porte dérobée (Backdoor): entrée de réseau ou de programme cachée dans un système pour permettre de s'y connecter subrepticement.

Terminologie des maliciels

- Rootkit: ensemble d'outils installés dans un système permettant d'en prendre le contrôle à distance, et dissimulé parmi les fichiers d'un système (en général capable de se réparer si on n'en détecte qu'une partie)
- (Capteur clavier) Keylogger: (partie de) programme espionnant les touches saisies sur un clavier
- Rançongiciel (Ransomware): programme chiffrant les données d'une victime et demandant une rançon pour donner la clé de déchiffrement
- Botnets (ou « bots »): ensemble de machines ayant une porte dérobée contrôlée depuis un ordinateur distant pour lancer simultanément des attaques massives depuis un grand nombre de sources (DDoS)

Complexité croissante des maliciels

- Polymorphisme: mutations aléatoires du code (comme le virus de la grippe) rendant difficile la détection
  - Milliers de variantes à partir d'une même souche pour tromper les anti-virus
- Maliciels chiffrés et empaquetés en « vagues »
  - Le code initial du virus (C0) ne fait rien d'autre que de se réécrire en un code C1, qui lui même ne contient pas le code malicieux mais se réécrit en code C2 et le vrai code malicieux est Cn (n~10).
- Code binaire de processeurs « inconnus »
  - Exécuté par une VM chiffrée
- Se cachent si détectent qu'ils sont observés

Risques élevés

- Les principes et ressorts de la malveillance sont vieux comme le monde:
  - Escroquerie, chantage, incapacitation, désinformation
  - Ingénierie Sociale (jouer sur le facteur humain, maillon faible de la sécurité)
- Mais les moyens techniques vont à la vitesse des ordinateurs et d'algorithmes optimisés
  - Ex: craquage de milliers de mot de passe / seconde, déni de service par envoi de Térabits/s
- Un seul clic étourdi sur un lien dangereux peut compromettre votre ordinateur ou smartphone – à votre insu (installation de rootkit, rançongiciel, keylogger etc)
- Même des sites militaires secrets ont été compromis (Stuxnet, centre drones USA…)

Menaces dans le (cyber?)-espace

- Virus, ver; programmes auto-réplicants
  - danger planétaire.
  - Solutions: détection (a posteriori), non exéc de pgms téléchargés…
- Intrusions
  - vols de données, prise de contrôle
  - contrôle d'accès, pare-feux, IDS (Intrusion Detection Systems)
- Déni de service (DoS): saturation de serveurs
- Botnets
  - 25 à 30% des ordinateurs enrôlés

Menaces planétaires

- 1960-1980 Guerre froide, la peur nucléaire
  - Missiles pouvant frapper le monde en 30 minutes
- 2000 ->
  - Cyber-attaque ou virus pouvant paralyser le monde en quelques secondes
  - Tout est raccordé à l'Internet: énergie (centrales nucléaires), transports, eau, automates industriels et agricoles…
    - Et même les sites militaires sensibles non connectés peuvent être attaqués (cf Stuxnet, usines nucléaires Iran, centre de contrôle des drones américains…)
  - Mafias, terroristes et états ennemis…

Les chevaliers du XXIème siècle

Face à des menaces grandissantes

Des experts prêts à protéger l'humanité: VOUS !

Engagez-vous, venez nous rejoindre

- Vos enseignants chercheurs en sécurité
  - Marie-Laure Potet (analyse de code, vulnérabilités., contre-mesures)
  - Roland Groz (réseaux, test, reverse Machine Learning)
  - Jean-Louis Roch (codage, cryptographie)
  - François Cayre (réseaux, image, tatouage…)
  - Maciej Korczynski, Andrzej Duda, Franck Rousseau (réseaux, botnets, IoT)
  - Régis Leveugle (attaques physiques sur circuits)
  - Stéphane Mocanu (protocoles industriels)
- UGA, INRIA: Philippe Elbaz-Vincent (maths), Florent Autréau (audit, forensics), Cédric Lauradoux (privacy)…
- Master cybersécurité (sécurité, crypto):
  - spécialisation 3A après une 2A filière SEOC (signal, code, sécu), MMIS (TS) ou ISI (avec cours code)

Recherches en sécurité

- Attaques physiques sur circuits (laser etc…)
- Pots de miel (typologie des attaques)
- Détection d'intrusions IDS (à base de Traitement de Signal avancé ou de Machine Learning etc)
- Contrôle d' accès (vérification, déploiement)
- O/S sécurisés et techniques d' exécution sûres
- Analyse de vulnérabilités du code
- Fuzz testing (frelatage)
- Preuves de protocoles cryptographiques
- …

Il y en a pour tous les goûts, des maths pures (théorie des nombres) à la rétro-ingénierie de code binaire

SecurIMAG

- Club étudiant très actif: jeudis 17h
- Ouvert à tous (et à toutes)
  - étudiants Ensimag, Phelma, UGA,
  - thésards en sécurité...
- Présentations informelles entre étudiants + conférences par des professionnels
  - Alexandra Ruiz (juriste, LEXSI): aspects juridiques
  - Nicolas Ruff (Google): (EADS… sécurité Windows)
  - Denis Pélanchon (A-Axis): intelligence économique
  - Fabrice Desclaux (CEA-DAM): rétro-ingénierie de binaires
- https://securimag.org/wp/

Bilan chapitre Crypto: notions essentielles

- Terminologie de la cryptographie
  - propriétés: confidentialité, intégrité, disponibilité...
  - clair, chiffrer, déchiffrer, décrypter, substitution, permutation
- Chiffrements à clés publiques / secrètes
  - Chiffrement symétrique, asymétrique
  - Chiffrement symétrique pour flux, cryptographie asymétrique pour authentification
- Authentification à clé publique
  - Certificats pour les clés publiques
  - Pbs de distribution des clés, schéma de Diffie-Helmann
- Hachage et signature
- Mécanismes de sécurité de ssh, et TLS

Référence humoristique

R. GROZ