Sécurité des SI

Xavier Delannoy

Qu'est-ce que la sécurité ?

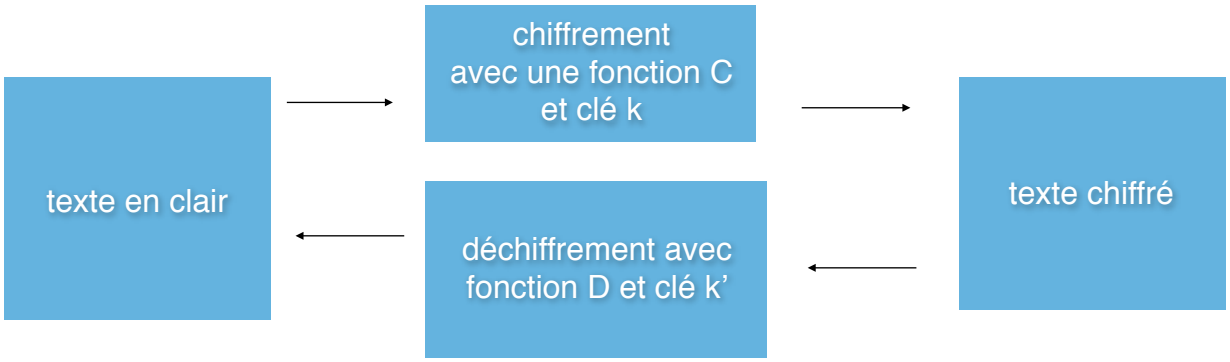
Plusieurs approches, de différents niveau de maturité :

- niveau 0 , c'est du réseau.
- niveau 1 , c'est de la technique ,quand on programme aussi,
- niveau 2 , c'est aussi dans les processus
- dernier niveau, pourquoi fait-on de la sécurité, s'il y a un risque ! La sécurité c'est la réponse aux risques. Des données publiques n'ont pas à être protégées.

Problème de cloud. Est-ce que ça vous embête de stocker des cartes de crédit sur le compte Apple ? oui, non… petit débat.
La sécurité, c'est aussi un problème de l'utilisateur et pas uniquement du RSSI. En voiture, c'est bien au conducteur d'être prudent. De plus, la sécurité est très transverse…! Et est complexe (firewall, base de données, réseaux, …)
On s'intéresse à la sécurité du data processing : sécurité des données, des traitements, des échanges, en commençant par des éléments de cryptographie, de façon pragmatique, sans entrer dans les algorithmes et les méthodes mathématiques en informatique de gestion.

Première partie : Éléments de cryptographie

Quelques éléments de terminologie



(C,D) = crypto système, le chiffre
On peut aussi essayer de casser le texte chiffré.

Attaque texte en clair connu

Hypothèse : l'attaquant dispose de plusieurs couples (texte clair , texte chiffré) : (T1,X1), (T2,X2) On a X3 et on veut T3. Cela a permis à Champollion avec la pierre de rosette, de déchiffrer les hiéroglyphes.

Texte chiffré seul

on a le texte chiffré seul X1 et les fonctions C et D.
Bruteforce : i=1, si Dp(X1) et on voit si c'est un texte intelligible. si c'est ok, on arrête, sinon on passe à i+1.
On peut compliquer les choses pour l'attaquant en changeant les fréquences, en compressant le fichier avant de le crypter (même compresser plusieurs fois) afin de plomber ses temps de calculs. Par exemple, le chiffrement utilise une clé hachée plusieurs fois.

La sécurité

Les points importants de la sécurité :

Disponibilité : ça marche

Intégrité :
Confidentialité
Traçabilité
Preuve, non répudiation. (statique ou dynamique)

HIPS, outil utilisé en sécurité qui permet de bloquer une attaque quand elle apparait. Elle a des signatures d'attaque pour reconnaitre des comportements non standards, des applications qui sortent de leur espace mémoire.

L'antivirus lui recherche des patterns de codes. cependant 70% des malwares sont utilisés sur des cibles uniques et précises. Un antivirus fait du gros œuvre et détecte le bruit de fond. Il fait cependant bien parti de l'arsenal.

La mauvaise sécurité, c'est de mettre un « boîtier » pour chaque problème. Shellshock, Heartbleed, etc…

Il est primordial de désactiver tous les services inutiles, comme un driver usb par exemple. Aujourd'hui, l'avantage est du côté de l'attaquant qui utilise une seule faille, quand la RSSI doit protéger toutes les failles. À la douane, il y a des attaques quotidiennes, et particulièrement les veilles de vacances, de pont etc… Le vrai problème, quand on détecte une attaque, si c'est un antivirus ou un HiPS ça va. Alors que si on trouve quelque chose en fouillant les logs…. ça peut être plus difficile. L'idéal, c'est le réseau distinct ! Mais cela n'est pas possible pour tout le monde.

Retour sur les éléments fondamentaux du chiffrement : Propriétés constitutives d'un bon chiffre

Efficacité : exemple, lorsque les américains ont lancés un concours pour savoir quel chiffre symétrique pourrait remplacer le DES, à utiliser dans l'administration et IBM a été retenu avec lucypher. Puis le DES a été cassé en une journée en force brute et a alors été remplacé, non pas par la plus robuste mais celle qui avait le meilleur compromis entre vitesse et robustesse. Ce chiffre est l'actuel AES.

Confusion : la relation entre d'une part le texte en clair et la clé, et d'autre part le texte chiffré doit être aussi difficile que possible à établir.

Exemple:
(ABC, BCD)
(LMN, MNO)
Peu de confusion, clé k égale à 1.

Diffusion : une modification, même mineure du texte en clair doit se traduire par une modification très importante du texte chiffré.

une bonne diffusion par exemple ABC -> EFG et ABD -> XTA.

Principe de Kerckhoffs

La difficulté de casser un texte chiffré ne doit pas dépendre du secret du cryptosystème mais du secret des clés.

Ce principe apparaît parmi les 6 « desiderata de la cryptographie militaire » énoncés par Kerckhoffs dans son traité, qui sont :

1. Le système doit être matériellement, sinon mathématiquement indéchiffrable ;
2. Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;
3. La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;
4. Il faut qu'il soit applicable à la correspondance télégraphique ;
5. Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;
6. Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

Kerckhoffs insiste sur les 3 premiers desiderata, qui sont véritablement originaux à son époque, les 3 derniers n'étant alors pas contestés. Ce qui est appelé aujourd'hui « principe de Kerckhoffs » est essentiellement le second.

« Le principe de Kerckhoffs s'applique au-delà des chiffres et des codes, c'est-à-dire aux systèmes de sécurité en général : tout secret est en fait un point de cassure possible. Par conséquent, le secret est une cause première de fragilité, donc cela même peut amener un système à un effondrement catastrophique. À l'inverse, l'ouverture amène la ductilité. »

Vulnérabilités

De conception
D'implémentation
D'exploitation
Humaines

Typologie des chiffres

Les chiffres symétriques k = k' : ceux par bloc et ceux continue.
Les chiffres asymétriques k ≠ k'

LES CHIFFRES SYMÉTRIQUES PAR BLOC

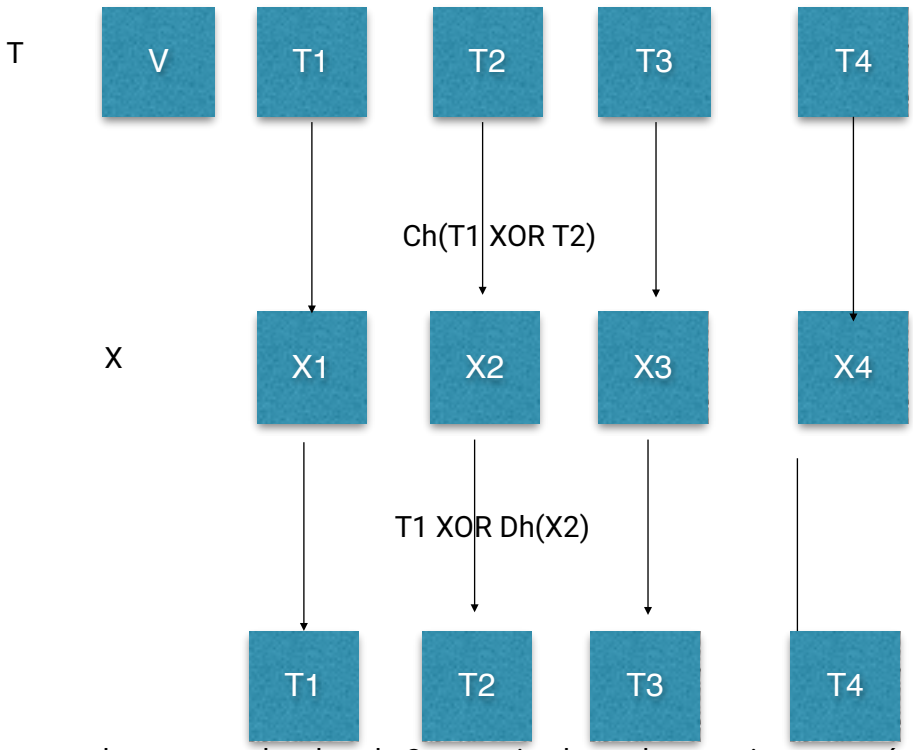
Une liste non-exhaustive de chiffrements par bloc :

- DES, l'ancêtre conçu dans les années 1970, a été passablement étudié
- AES, le remplaçant de DES
- Blowfish, Serpent et Twofish, des alternatives à AES

Le principe

On découpe le texte par bloc et chacun des blocs doit être chiffré
Problème : il y a un invariant, un message identique sera chiffré identiquement. T2 XOR T3. le XOR du XOR est l'identité. T1 XOR Ch(X2) = T1 XOR Dh(Ch(T1 XOR T2)) = T1 XOR T1 XOR T2= T2.
V est le vecteur d'initialisation.

Cypher Bloc Chaining (CBC)



L'asymétrique est beaucoup plus lourd. On essaie donc de revenir au symétrique rapidement, même si on initialise la communication en asymétrique

LES CHIFFRES SYMÉTRIQUES EN CONTINU,

Les chiffrements en continu utilisent un changement de la clé en permanence.

One time pad (le masque jetable) système aujourd'hui le plus robuste. Le one time pad est aussi long que le texte à chiffrer.

1	0	1	0
1	1	0	1
0	1	1	0

xor

1	0	1	1
1	1	0	0
0	1	0	0

=

0	0	0	1
0	0	0	1
0	0	1	0

Pour chiffrer on calcule donc C = A ⊕ B. Le résultat C est le chiffré de A. L'opération est effectuée pour chaque bit du clair avec le bit correspondant de la clé.
Le déchiffrement s'effectue en combinant le chiffré C avec le bit de clé B par la simple opération : C ⊕ B. Il se trouve qu'elle fait retrouver le clair A, comme nous allons le montrer.
Remarquons que l'opération XOR possède les deux propriétés suivantes :

A ⊕ A = 0
A ⊕ 0 = A

ce qu'on vérifie facilement avec le tableau ci-dessus, en considérant les deux valeurs possibles de A, qui sont 0 ou 1.
Le calcul de déchiffrement peut donc s'écrire :

C ⊕ B = (A ⊕ B) ⊕ B = A ⊕ (B ⊕ B) = A ⊕ 0 = A

Il fait bien retrouver le bit de clair A.

Expérimentation concrète d'un algorithme symétrique continu sur un cas concret.

RC4
Keystream ; la clé change au fur et à mesure de l'algo
Ici, on recopie la cel jusqu'à l'index 255.

```
index1=-1 index2=0
for (int counter=0; counter(TC.length); counter++){
      //index suivant
      index1=(index1+1) mod ksize;
      index2=(index2+ks[index1]) mod ksize

      //permutation
      swap=ks[index1];
      ks[index1]=ks[index2];
      ks[index2]=swap;

      //xor
      XORindex=(ks[index1]+ks[index2]) mod ksize
      TC[counter]=TC[counter] XOR  ks[XORindex]

}
```

```
index1=(-1+1) mod 255 = 0
index2=(0 +ks[index1]) mod 255
index2=(0 +ks[0]) mod 255
index2=(0 +S) mod 255
index2=83 mod 255

//permutation
swap=ks[index1]
swap=ks[0];
swap=S;
swap=83;

ks[0]=ks[index2];
ks[0]=ks[83];
ks[0]=ks[83];
ks[0]=T;
ks[0]=84;

ks[1]=swap;
ks[1]=83;

//xor
XORindex=(ks[0]+ks[1]) mod 255
XORindex=(84+83) mod 255
XORindex= 167 mod 255
TC[0]=TC[counter] XOR  ks[XORindex]
TC[0]=TC[0] XOR  ks[167]
TC[0]=M XOR  T
TC[0]=77 XOR 84
TC[0]=25 = EM (end of medium)

65 XOR 69 = 4
83 XOR 84 = 7
```

TC

M	A	S	T	E	R
77	65	83	84	69	82

CLÉ

S	E	C	R	E	T

KS KeyStream

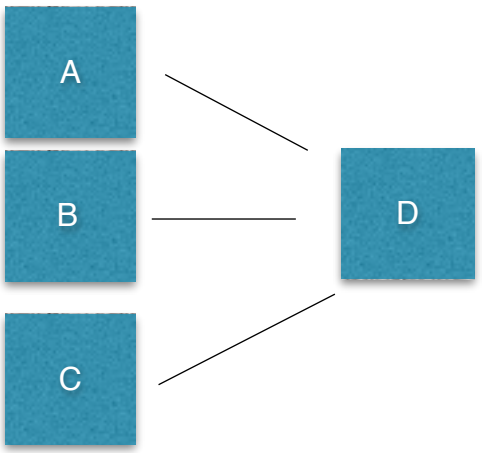
counter	0	1	2	3	4	5	6	7	8	…	83	…	149	…	152	…	219
0	S	E	C	R	E	T	S	E	C	….	T	…	T	…	C	…	R
1	T										S						

Exemple de chiffre symétrique par bloc : DES Digital Encryption System

Date des années 73 par un concours de la NSA, devenu un standard dans l'administration américaine. si on peut aller 1000 fois plus vite, il faut ajouter 10 bits à la clé pour contrer car 2^10 > 1000.
sbox
pbox
Message est découpé en blocs de 64 bits
clé est découpée en 16 clé de 48 bits

Chiffres Asymétriques

pub(priv(M))=priv(pub(M)=M
avec 4 personnes, il faut 6 clés symétriques mais seulement 4 clés asymétriques. on s'affranchit avec l'asymétrique du problème de diffusion de la clé.



RSA

choisir deux entiers p et q ; n=pq
choisir e tel qu'il soit premier à (p-1)(q-1)
(n,e) clé publique
choisir d tel que le quotient (ed-1)/(p-1)(q-1) soit un entier
(n,d) clé privée

Exemple
p=3
q=5
n=15

e = 11
d=3
ed-1 = 32
(p-1)(q-1)=8

Xi = Ti ^e mod n

T = SECRET soit 83 69 67 82 69 84

8^11 mod 15 = X1 = 2
3^11 mod 15 = X2 = C
6^11 mod 15 = X3 = 6
9^11 mod 15 = X4 = 9

on remarque un invariant car les clés sont petites.

2^3 mod 5 = 8

L'attaquant connait e et n. il faut trouver p et q et donc factoriser n. ce qui est très difficile avec des nombres premiers.

hashcode

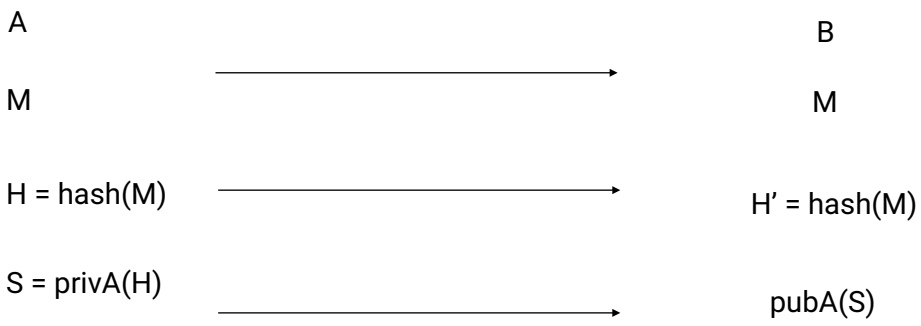
md-5 (peu fiable)
sha-1 (commence à montrer des faiblesses)
sha-2
sha-3

Ce type de fonction est intéressante quand il y a peu de collisions.

Confiance dans les échanges électroniques

Authentification
Confidentialité : l'information est cryptée et si elle est
Principe de non répudiation dynamique : lettre recommandée, atteste qu'il y a eu un échange entre A et B. Les partenaires ne peuvent pas nier leur participation.
Principe de non répudiation statique : signature, on ne peut nier avoir envoyé ou avoir eu connaissance d'un envoi.
Intégrité : on s'assure qu'un contenu n'a pas été modifié pendant le transfert. Détecter que ce qui a été reçu est bien ce qui a été envoyé.

authentification : A envoie M à B.
B y applique sa clé privée
A applique la clé publique de B et vérifie qu'il a bien M.
Fonctionnellement parlant, le hash n'est pas nécessaire et permet une réduction de la taille du message. pour calculer S, on a besoin de la clé privée de A et donc cette procédure permet une signature électronique, garantie la non répudiation du message.



Comment A peut-il s'assurer que la clé publique de B est bien celle de B ?
Il faut faire intervenir quelqu'un en plus, quelqu'un en qui A a confiance. Il faut une autorité de confiance. Le certificat est un fichier qui contient une identité et une clé. ET sa signature Ce certificat doit être protégé lui même.
Le certificat possède une période de validité. Le certificat permet une signature, authentification et chiffrement.
On est A, on reçoit le certificat de l'identité de B.
pubAC(S) = hash (C)
Concept de révocation de certificat à revoir. (lorsqu'un certificat arrive à fin de validité)
CRL distribution point

Autorité de certificat
PSCE Prestataire de services de certification électronique (PSCE)

AC et AE

Certinomis, prestataire de services de certification de la poste

Politique de sécurité du système d'information

Mythe des password cracker en résultat. La sécurité aujourd'hui c'est de l'argent. On peut payer pour des pentests.

Etat de la menace.

étude annuelle faite par verizon.
Événement : anormalité. Il faut la qualifier, étudier la normalité a posteriori.
Mythe de la sophistication des attaques.
faux, les attaquant vont au plus simple. Ils utilisent ce qui est simple et qui marche tant que ça marche. l'objectif n'es plus le challenge technique mais d'arriver rapidement à l'objectif qui n'est pas technique : gain financier, espionnage…
Pas de nouvelles menaces sur les mobiles. les techniques usuelles marchent…!
La création de nouvelles menaces sur les mobiles s'est ralentie car les auteurs préfèrent améliorer celles déjà existantes.
84% des infractions sont réalisées en quelques minutes ou quelques heures
80% des attaques de type injection SQL sont automatisées (havij, sqlmap,…)
le niveau de difficulté des infractions augmente mais reste simple.

Les attaques sophistiquées combinent des techniques simples
souvent en 5phases
1-reconnaissance : ingénierie sociale
2-incursion
3-Découverte
4-Capture
5-Exfiltration

autre mythe : les centres de production sont la cible privilégiée des attaquants
l'informatique locale est un des vecteurs d'attaque significatif pour s'introduire dans le système d'information
25% des infractions impliquent un ordinateur de bureau
22= des infractions impliquent un portable
22% des infractions impliquent un serveur de fichier

28% des attaques reposent sur un serveur web compromis
la passerelle xxl a bloqué 430486 accès à des pages web infectées en septembre 2013
Une application web est attaquée en moyenne 4fois par mois certaines attaquées en permanence.

mythe
Les attaques sur les systèmes android croissent mais son peu répandues dans l'absolu
réalité
en autsralie et aux etats unis les systèmes anordi ont une probabilité d'être attaqué supérieure aux systèmes windows. plus ou moins équivalenten france, allemagne et pays bas
80% des malwares pour mobile ciblent le système d'exploitation android
73% de part de marché et 83% des par de marché.
les malwares android utilisent désormais les mêmes techniques que sur windows
Les antivirus pour mobiles bouffent l'autonomie.
180 failles trouvées chez iOS en 2013 et 3 chez Android

mythe
les données sont piratées durant leur transport sur internet

réalité
très peu d'infractions sur des données en transit sur les réseaux
les donnés sont principalement vulnérables au repos dans les bases de donnés et les serveurs de fichiers
en cours de traitent dans les serveurs application

Sécuriser un lien est relativement facile. voir la démarche de Galiloe qui ci

mythe
les attaques sont visibles (si je suis piraté, je le sais)

réalité
les attaques sont furtives
69% des infractions sont repérées par un tiers indirectement
66% des infractions restent ignorées pendant des mois (ce qui augmente l'impact
4% des infractions sont détectées après plusieurs années

mythe
Les attaques exploitent des systèmes non pactisé (correctifs de sécurité non appliqués)
réalité
les correctifs de sécurité permettent de se protéger des attaques opportunistes (50% des attaques) mais….
pas des attaques basées sur des 0-day : vulnérabilité exploitée mais das de correctif disponible
14 vulnérabilités publiées 0 day en 2012
non publiées
forte hausse de l'exploitation des 0-day en 2013

Certains logiciels courants sont mal développés et contiennenttbeaucoup de vulnérabilités et donc un fort potentiel de 0-day
Adobe reader : 58 vuln critiques depuis le début de l'année
Par comparaison, Microsoft word : 15 vulnérabilités critiques depus le début de l'année.

comment trouver une vulnérabilité ?
souvent en faisant du fusing (plein de données dans tous les sens pour créer des erreurs)

les attaques exploitent des vulnérabilités

spear phishin est ciblé (APT) 91% des campagne en plus par rapport entre 2012 et 2013
472% de campagnes de plus entre 2011 et 2013

les administrations sont particulièrement visées
de mieux en mieux ciblées : de moins en moins de mail pour une attaque (-76% entre 2012 et 2013)
De plus en plus discret :
mails étalés sur une période plus longue : de trois jour en 2012 à 8 jours en 2013

Les mails de spearfishing peuvent renvoyer vers un site infecté ou contenir directement une charge malveillante dans un document joint.
Format des documents joints. exe reste une valeur sure

pdf, doc, xls en baisse
.class et jpeg en hausse

Etapes d'une attaque :
l'intrusion . l'attaquant prend pied sur un système informatique:
la persistance . l'attaquant met en place les moyens lui permettant de régulièrement se connecter en toute discrétion à ce système

drive by download attack
hacking as a service
responsabilité pénale : l'hébergeur d'un serveur de redirection est responsable

l'exploitation des vulnérabilités des applications sur linux est rapidement intégrée par les antiviraux par exemple la backdoor darkleech ou shell bash.

la sécurité du poste de travail es rassurée par sont antiviraux
75% des malwares ne sont trouvés que dans une seule organisation.

Retour sur les certificats
identité + clé publique + date de validité + signature (établie avec des éléments non fraudantes), attributs des certificats : keyusage

signature par A des données D est : privA(hash(D))