Introduction à la cryptographie

Jean Goubault-Larrecq

Séminaire Regards Croisés

Outline

Outline

Le chiffrement

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Le chiffrement

QUIDITCRY
PTOGRAPHI
EDITCHIFF
REMENTETC
EDEPUISLA
PLUSHAUTE
ANTIQUITE

(C’est plus clair comme ça?)

La scytale

(utilisée par les anciens spartiates.)

Les premiers usages de la cryptographie

1. Ammien Marcellin (330-395), XVIII, III, 2-3, relate un épisode qui s'est déroulé sous le règne de Constance, à Sirmium, au retour de l'expédition contre les Sarmates pendant l'hiver 358-359. Le commandant de l'infanterie Barbation et son épouse Assyria eurent la tête tranchée suite à la découverte d'une missive chiffrée.
2. Ammien Marcellin, XVIII, VI, 7, expose la fuite des Romains à l'approche des Perses et leur arrivée dans la ville d'Amida en Mésopotamie, sous l'empereur Julien (360-363): "nous découvrîmes, à l'intérieur d'un fourreau, un parchemin recouvert par les caractères de signes secrets confiés par Procope à notre intention."

Source: `http://bcs.fltr.ucl.ac.be/FE/07/CRYPT/Crypto07-28.html`.

La cryptographie militaire

JOURNAL
DES
SCIENCES MILITAIRES.

Janvier 1883.

LA CRYPTOGRAPHIE MILITAIRE.

« La cryptographie est un auxiliaire puissant de la tactique militaire. »
(Général LEWAL, *Études de guerre.*)

I.

LA CRYPTOGRAPHIE DANS L'ARMÉE

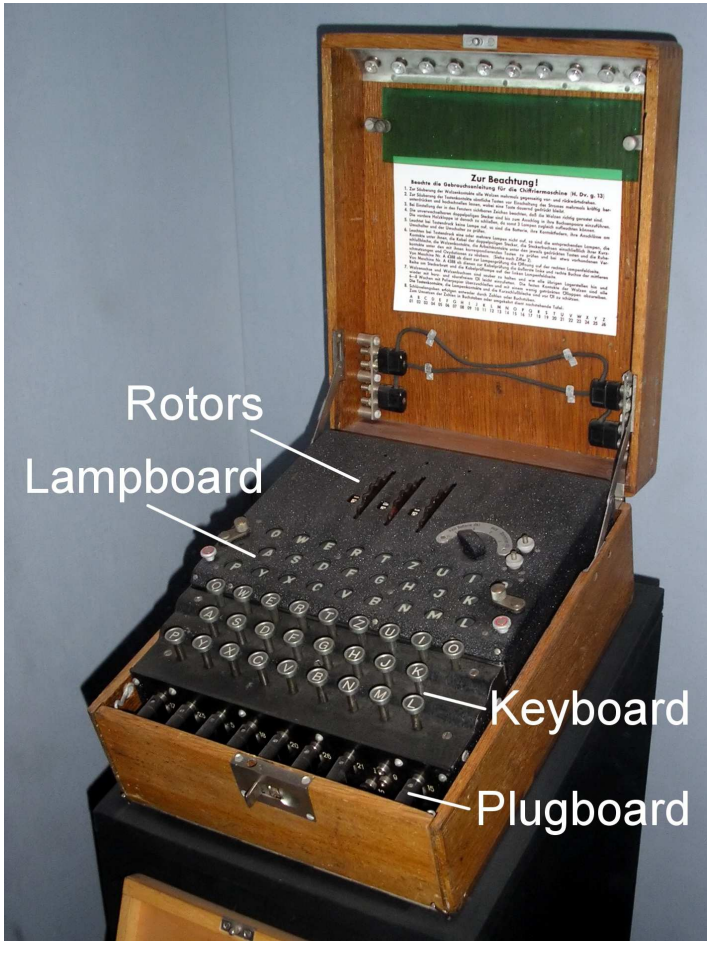
A. Notions historiques.

La *Cryptographie* ou l'*Art de chiffrer* est une science vieille comme le monde ; confondue à son origine avec la télégraphie militaire, elle a été cultivée, dès la plus haute antiquité, par les Chinois, les Perses, les Carthaginois ; elle a été enseignée dans les écoles tactiques de la Grèce, et tenue en haute estime par les plus illustres généraux romains [1].

Depuis la modeste scytale des Lacédémoniens et les *trucs* inventés ou rapportés par Æneas-le-Tacticien [2], jusqu'au fameux

Auguste Kerckhoffs, 1883

La machine Enigma



Allemagne, 1939-45.

Aujourd'hui—l'e-commerce

Applications modernes de la cryptographie

- E-commerce (cf. ci-dessus);
- Vote électronique (à venir en France, déjà installé en Estonie);
- Banques: cartes à puce, distributeur bancaires (HSM);
- Mises à jour automatiques de logiciels;
- Signature électronique de documents;
- et bien d'autres.

Ce cours

1. Dans ce cours, nous verrons les principes de base de la cryptographie.
2. Stéphanie Delaune vous présentera ceux du domaine plus large de la sécurité des protocoles cryptographiques.
3. Frédéric Grosshans vous parlera ensuite de cryptographie quantique.

Outline

Un peu de vocabulaire

- Chiffrement: $M, K \mapsto \{M\}_K$
  où $M =$ message *en clair*, $K =$ clé,
  $\{M\}_K =$ message *chiffré*;
- Déchiffrement: $\{M\}_K, K^{-1} \mapsto M$;
  (La clé de déchiffrement K^{-1} sera égale à K dans le cas de chiffrements symétriques.)
- Décryptage: retrouver M (ou K) à partir de $\{M\}_K$, sans K^{-1}.

Note 1: on ne dit pas “codage”, “encodage”, “décodage”, “encryption”, “décryption”.

Note 2: le chiffrement et le déchiffrement sont des activités “normales”, le décryptage est une activité de pirate (ou de cryptanalyste).

Le chiffrement de César (variante)

Chiffrement par substitution alphabétique: A$\mapsto$N, B$\mapsto$O, etc.

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Le chiffrement de César (variante)

Chiffrement par substitution alphabétique: A$\mapsto$N, B$\mapsto$O, etc.

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, ...

Le chiffrement de César (variante)

Ici, G, R, V (8), H (5), N, C (4), etc.

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, . . .

Le chiffrement de César (variante)

Ici, G, R, V (8), H (5), N, C (4), etc.

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, ...
Supposons que G remplace E: décalage=+2.

Le chiffrement de César (variante)

Ici, G, R, V (8), H (5), N, C (4), etc.

```
BFTOTENCJ
AEZRCLAST
OPTENSTQQ
CPXPYEPEN
POPAFTDWL
AWFDSLFEP
LYETBFTEP
```

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, ...
Supposons que G remplace E: décalage=+2. Non.

Le chiffrement de César (variante)

Ici, G, R, V (8), H (5), N, C (4), etc.

DHVQVGPEL
CGBTENCUV
RQVGPUVSS
ERZRAGRGP
RQRCHVFYN
CYHFUNHGR
NAGVDHVGR

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, . . .
Essayons R pour E: décalage=+13.

Le chiffrement de César (variante)

Ici, G, R, V (8), H (5), N, C (4), etc.

```
QUIDITCRY
PTOGRAPHI
EDITCHIFF
REMENTETC
EDEPUISLA
PLUSHAUTE
ANTIQUITE
```

Cryptanalyse: schéma faible [AlKindi, 801-873].
Idée: si le test est en Français, les lettres classées par fréquence décroissante sont E, A, I, S, T, ...
Essayons R pour E: décalage=+13. Ça marche.

Le chiffrement de César

1. La clé était donc le décalage 13, ici.
2. Le chiffrement de César utilisait comme clé 3. (C'est bon pour les messages "secrets" des jeux d'un magazine pour enfants...)
3. Il n'y a que 26 clés possibles $\Rightarrow$ facilement énumérables sur un ordinateur moderne (même pas besoin de statistiques).
4. Mais, plus subtilement, un programme de décryptage doit aussi pouvoir décider quand il a trouvé la bonne clé. Exercice: comment peut-on faire? (Penser statistiques...)
5. Amélioration: la fréquence des bigrammes est ES>LE>EN>DE>RE>NT> ...

Le chiffrement de Vigenère (1523-1596)

1. Chiffrement polyalphabétique.
2. On utilise un autre message (court) comme clé K.
3. Principe:

$$\text{lettre } i \text{ de } \{M\}_K = \text{lettre } i \text{ de } M + \text{lettre } (i \mod |K|) \text{ de } K, \mod 26.$$

Clair:	J ADORE ECOUTER LA RADIO TOUTE LA JOURNEE
Clé:	M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
Chiffré:	V UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY

Déchiffrement: exercice. Cryptanalyse?

Note: certaines lettres différentes sont chiffrées pareil, certaines lettres identiques sont chiffrées différemment.

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé…

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.

Clair:	J	A	D	O	R	E	E	C	O	U	T	E	R	L	A	R	A	D	I	O	T	O	U
Clé:	M	U	S	I	Q	U	E	M	U	S	I	Q	U	E	M	U	S	I	Q	U	E	M	U
Chiffré:	V	U	V	W	H	Y	I	O	I	M	B	U	L	P	M	L	S	L	Y	I	X	A	O

Clair:	T	E	L	A	J	O	U	R	N	E	E
Clé:	S	I	Q	U	E	M	U	S	I	Q	U
Chiffré:	L	M	B	U	N	A	O	J	V	U	Y

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.

Clair:	J ADORE ECOUTER LA RADIO TOUTE LA JOURNEE
Clé:	M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
Chiffré:	V UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.

Clair:	J	A	D	O	R	E	E	C	O	U	T	E	R	L	A	R	A	D	I	O	T	O
Clé:	M	U	S	I	Q	U	E	M	U	S	I	Q	U	E	M	U	S	I	Q	U	E	M
Chiffré:	V	U	V	W	H	Y	I	O	I	M	B	U	L	P	M	L	S	L	Y	I	X	A

Clair:	U	S	E	C	O	U	T	E	R	L	A	R	A	D	I	O	T	O	U	T	E	L
Clé:	S	I	Q	U	E	M	U	S	I	Q	U	E	M	U	S	I	Q	U	E	M	U	S
Chiffré:	M	A	U	W	S	I	N	W	Z	B	U	V	M	X	A	W	J	I	Y	F	Y	D

Clair:	A	J	O	U	R	N	E	E														
Clé:	I	Q	U	E	M	U	S	I														
Chiffré:	I	Z	I	Y	E	H	W	M														

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.

Clair:	J		A	D	O	R	E		E	C	O	U	T	E	R		L	A		R	A	D	I	O		T	O	U	T	E		L	A		J	O	U	R	N	E	E
Clé:	M		U	S	I	Q	U		E	M	U	S	I	Q	U		E	M		U	S	I	Q	U		E	M	U	S	I		Q	U		E	M	U	S	I	Q	U
Chiffré:	V		U	V	W	H	Y		I	O	I	M	B	U	L		P	M		L	S	L	Y	I		X	A	O	L	M		B	U		N	A	O	J	V	U	Y

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.

Clair:	J ADORE ECOUTER LA RADIO TOUTE LA JOURNEE
Clé:	M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
Chiffré:	V UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
KQOWEFVJPUJUUNUKGLMEKJINMWUXFQMKJBGWRLFNFGHUDWUUMBSVLPS
NCMUEKQCTESWREEKOYSSIWCTUAXYOTAPXPLWPNTCGOJBGFQHTDWXIZA
YGFFNSXCSEYNCTSSPNTUJNYTGGWZGRWUUNEJUUQEAPYMEKQHUIDUXFP
GUYTSMTFFSHNUOCZGMRUWEYTRGKMEEDCTVRECFBDJQCUSWVBPNLGOYL
SKMTEFVJJTWWMFMWPNMEMTMHRSPXFSSKFFSTNUOCZGMDOEOYEEKCPJR
GPMURSKHFRSEIUEVGOYCWXIZAYGOSAANYDOEOYJLWUNHAMEBFELXYVL
WNOJNSIOFRWUCCESWKVIDGMUCGOCRUWGNMAAFFVNSIUDEKQHCEUCPFC
MPVSUDGAVEMNYMAMVLFMAOYFNTQCUAFVFJNXKLNEIWCWODCCULWRIFT
WGMUSWOVMATNYBUHTCOCWFYTNMGYTQMKBBNLGFBTWOJFTWGNTEJKNEE
DCLDHWTVBUVGFBIJG
```

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
KQOWEFVJPUJUUNUKGLMEKJINMWUXFQMKJBGWRLFNFGHUDWUUMBSVLPS
NCMUEKQCTESWREEKOYSSIWCTUAXYOTAPXPLWPNTCGOJBGFQHTDWXIZA
YGFFNSXCSEYNCTSSPNTUJNYTGGWZGRWUUNEJUUQEAPYMEKQHUIDUXFP
GUYTSMTFFSHNUOCZGMRUWEYTRGKMEEDCTVRECFBDJQCUSWVBPNLGOYL
SKMTEFVJJTWWMFMWPNMEMTMHRSPXFSSKFFSTNUOCZGMDOEOYEEKCPJR
GPMURSKHFRSEIUEVGOYCWXIZAYGOSAANYDOEOYJLWUNHAMEBFELXYVL
WNOJNSIOFRWUCCESWKVIDGMUCGOCRUWGNMAAFFVNSIUDEKQHCEUCPFC
MPVSUDGAVEMNYMAMVLFMAOYFNTQCUAFVFJNXKLNEIWCWODCCULWRIFT
WGMUSWOVMATNYBUHTCOCWFYTNMGYTQMKBBNLGFBTWOJFTWGNTEJKNEE
DCLDHWTVBUVGFBIJG
```

On repère les répétitions de motifs les plus longs...

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
KQOWEFVJPUJUUNUKGLMEKJINMWUXFQMKJBGWRLFNFGHUDWUUMBSVLPS
NCMUEKQCTESWREEKOYSSIWCTUAXYOTAPXPLWPNTCGOJBGFQHTDWXIZA
YGFFNSXCSEYNCTSSPNTUJNYTGGWZGRWUUNEJUUQEAPYMEKQHUIDUXFP
GUYTSMTFFSHNUOCZGMRUWEYTRGKMEEDCTVRECFBDJQCUSWVBPNLGOYL
SKMTEFVJJTWWMFMWPNMEMTMHRSPXFSSKFFSTNUOCZGMDOEOYEEKCPJR
GPMURSKHFRSEIUEVGOYCWXIZAYGOSAANYDOEOYJLWUNHAMEBFELXYVL
WNOJNSIOFRWUCCESWKVIDGMUCGOCRUWGNMAAFFVNSIUDEKQHCEUCPFC
MPVSUDGAVEMNYMAMVLFMAOYFNTQCUAFVFJNXKLNEIWCWODCCULWRIFT
WGMUSWOVMATNYBUHTCOCWFYTNMGYTQMKBBNLGFBTWOJFTWGNTEJKNEE
DCLDHWTVBUVGFBIJG
```

On calcule leurs distances: 95, 200, 190, 80, 45, 90.
Pgcd = 5 — souvent un multiple de $|K|$ (et même $= |K|$).

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
SFUHENKPAURJAYUSVRXESYOYMEJDQQUZPMGEGRQNNVNFDEJAXBAKRAS
VRSFESFIEEALXPESDEDSQLIEUIMEZTIEDALEETECODPMGNFNEDEMOKA
GVLQNAMIDEGCIESAETEURCEEGOLFRREJAYERJABEIEEXESFNFILJDQP
OJEESUILQSPCAZCHVSCUETEEROZSPELRZGRMRLMDRFIFSEKHANTVUJL
AZSEENKPUTELSQMEETXEUISSRAEDQSAZLQSBCAZCHVSOOMDEPESRVUR
OESFRAZNQRATOFEDVUJCEMOKAGVUDAICEOOMDEULEJTSAUTHQETMEGL
ECUUNAXUQREJINEALQGILVSFCODICUEVTXAIULGNAXAOESFNNECRVQC
UEBDULVGGEUCEXAUKRQMIDEQNBFIFANKLUNFZRYEQLIHOLRIFLEGOQT
EVSFSEDBXABCEMUPIIZCEUEENUVEEQUZHMNTVLMTEDPQTEVTEERZTPE
LRROHEIBMUDVLMIRV
```

Analyse de César: décalages 18, 11, 20, 15, 0.

Cryptanalyse du chiffrement de Vigenère

1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
SOUHENTPAURSAYUSERXESHOYMESDQQUIPMGEPRQNNENFDESAXBATRAS
VASFESOIEEAUXPESMEDSQUIEUIVEZTINDALENTECOMPMGNONEDEVOKA
GELQNAVIDEGLIESANTEURLEEGOUFRRESAYERSABEINEXESONFILSDQP
OSEESURLQSPLAZCHESCUECEEROISPELAZGRMALMDROIFSETHANTEUJL
AISEENTPUTEUSQMENTXEURSSRANDQSAILQSBLAZCHESOOMMEPESAVUR
ONSFRAINQRACOFEDEUJCEVOKAGEUDAILEOOMMEULESTSAUCHQETVEGL
ELUUNAGUQRESINEAUQGILESFCOMICUEETXAIDLGNAGAOESONNECAVQC
UNBDULEGGEULEXAUTRQMIMEQNBOIFANTLUNFIRYEQUIHOLAIFLEPOQT
EESFSEMBXABLEMUPRIZCEDEENUEEEQUIHMNTELMTEMPQTEETEERITPE
LAROHERBMUDELMIRE
```

Analyse de César: décalages 18, 2, 20, 15, 0.

Cryptanalyse du chiffrement de Vigenère

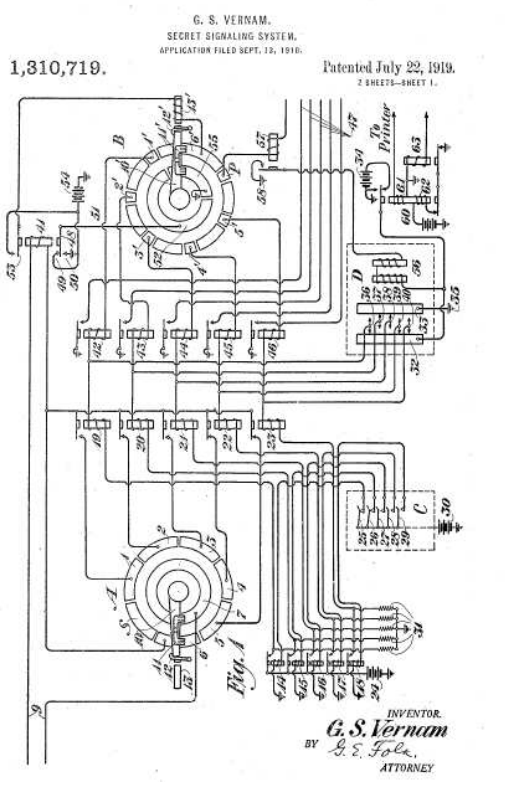
1. Si l'on connaît la longueur $|K|$ de la clé...
   alors c'est juste $|K|$ chiffrements de César entrelacés.
2. Et l'on peut retrouver $|K|$ avec forte probabilité [Babbage1854, Kasiski1863].

```
SOUVENTPOURSAMUSERLESHOMMESDEQUIPAGEPRENNENTDESALBATROS
VASTESOISEAUXDESMERSQUISUIVENTINDOLENTSCOMPAGNONSDEVOYA
GELENAVIREGLISSANTSURLESGOUFFRESAMERSAPEINELESONTILSDEP
OSESSURLESPLANCHESQUECESROISDELAZURMALADROITSETHONTEUXL
AISSENTPITEUSEMENTLEURSGRANDESAILESBLANCHESCOMMEDESAVIR
ONSTRAINERACOTEDEUXCEVOYAGEURAILECOMMEILESTGAUCHEETVEUL
ELUINAGUERESIBEAUQUILESTCOMIQUEETLAIDLUNAGACESONBECAVEC
UNBRULEGUEULELAUTREMIMEENBOITANTLINFIRMEQUIVOLAITLEPOET
EESTSEMBLABLEAUPRINCEDESNUEESQUIHANTELATEMPETEETSERITDE
LARCHERBAUDELAIRE
```

Analyse de César: décalages 18, 2, 20, 1, 0.

Le masque jetable (one-time pad, OTP)

Gilbert S. Vernam.
Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications.
J. Am. Inst. Elec. Engineers, v. XLV:109–115, 1926.



Comme Vigenère, mais:

- clé aussi longue que le clair;
- clé utilisée seulement une fois;
- clé aléatoire, tirée uniformément [J. O. Mauborgne];
- clé secrète.

Le masque jetable (one-time pad, OTP)

Clair M:	SOUVENTPOURSAMUSERLESHOMMESDEQUIPAGE
Clé K:	LOUOOKCOOFOWJZUYFKUNZSTYJRLUXJTZEXHR
Chiffré $\{M\}_K$:	DCOJSXVDCZFOJLOQJBFRRZHKVVDXBZNHTXNV

- Chiffrement: $\{M\}_K[i] = M[i] \oplus K[i]$;
- Déchiffrement: $M[i] = \{M\}_K[i] \ominus K[i]$.

Note: on utilise souvent le ou exclusif pour $\oplus$ plutôt que l'addition mod 26.

Cryptanalyse du masque jetable

Le masque jetable est inconditionnellement sûr:

Théorème

Soit a la taille de l'alphabet Σ (e.g., 26). La distribution des clairs, même connaissant le chiffré, est la distribution uniforme:

$$Pr_K[M, \{M\}_K = M' \mid M'] = \frac{1}{a^{|M|}}$$

Démonstration. *Considérons des M de longueur $|M| = n$. Pour un chiffré fixé M', il y a bijection entre les K et les M tels que $\{M\}_K = M'$.*
Donc $Pr_K[M, \{M\}_K = M' \mid M'] = Pr_K[K \mid M'] = \frac{1}{a^n}$. □
Note: le chiffré est lui aussi aléatoire uniforme!

Sûreté inconditionnelle?

. . . mais ça ne veut pas dire qu'il n'y a pas d'attaque!

- si K n'est pas secrète;
- si K a un biais statistique;

  voir suite (théorie de Shannon).
- si on utilise K plusieurs fois;

  $\{M_1\}_K, \{M_2\}_K \vdash M_1 \ominus M_2$—statistique connue.
- si les attaquants sont actifs;
- et puis, des attaques contre l'intégrité des messages.

Attaquants passifs, actifs

- Eve (passif): écoute les messages échangés entre A et B;
- Charlie (actif): + fabrique de nouveaux messages, interagit avec A et B, reroute les messages. . .

Vous verrez avec Stéphanie Delaune que certains protocoles cryptographiques sont attaquables même en présence de cryptographie inconditionnellement sûre.

Intégrité

Exemple (incomplet). Le protocole de Needham-Schroeder à clés publiques [NS78] est erroné:

$$\begin{aligned} A &\rightarrow B: \quad \{N_a, A\}_{K_b} \\ B &\rightarrow A: \quad \{N_a, N_b\}_{K_a} \\ A &\rightarrow B: \quad \{Nb\}_{K_b} \end{aligned}$$

Intégrité

Exemple (incomplet). Le protocole de Needham-Schroeder-Lowe à clés publiques [Lowe95] est réparé:

$$\begin{array}{lcll} A & \rightarrow & B: & \{N_a, A\}_{K_b} \\ B & \rightarrow & A: & \{N_a, N_b, B\}_{K_a} \\ A & \rightarrow & B: & \{Nb\}_{K_b} \end{array}$$

Intégrité

Exemple (incomplet). Le protocole de Needham-Schroeder-Lowe à clés publiques [Lowe95] est réparé:

$$\begin{array}{lcll} A & \to & B: & \{N_a, A\}_{K_b} \\ B & \to & A: & \{N_a, N_b, B\}_{K_a} \\ A & \to & B: & \{Nb\}_{K_b} \end{array}$$

Avec un masque jetable (pourtant inconditionnellement sûr):

$$B \to C(A): \quad \{N_a, N_b, B\}_{K_a} = \left\{ \begin{array}{cccc} & N_a, & N_b, & B \\ \oplus & K_a^1, & K_a^2, & K_a^3 \end{array} \right.$$

$$C \text{ ajoute} \qquad 0, \quad 0, \quad C \ominus B$$

$$C \to A: \qquad \{N_a, N_b, C\}_{K_a} = \left\{ \begin{array}{cccc} & N_a, & N_b, & C \\ \oplus & K_a^1, & K_a^2, & K_a^3 \end{array} \right.$$

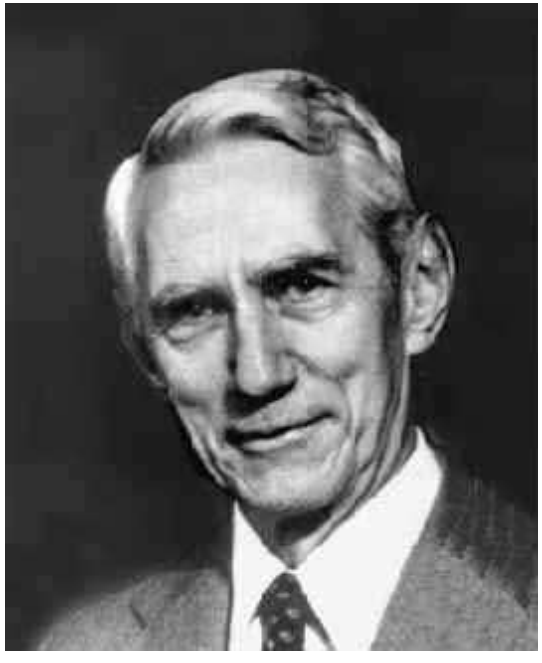
ce qui annule la réparation.

Note: C n'a jamais eu besoin de rien décrypter!

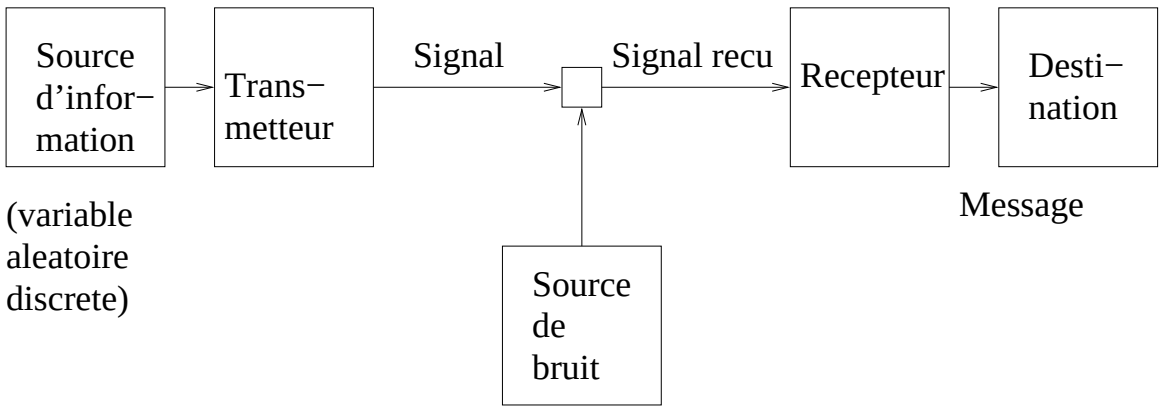
Outline

Entropie statistique

Claude E. Shannon.
A Mathematical Theory of Communication.
Bell System Technical Journal, 27:379–423, 623–656, 1948.



Entropie de Shannon



Definition

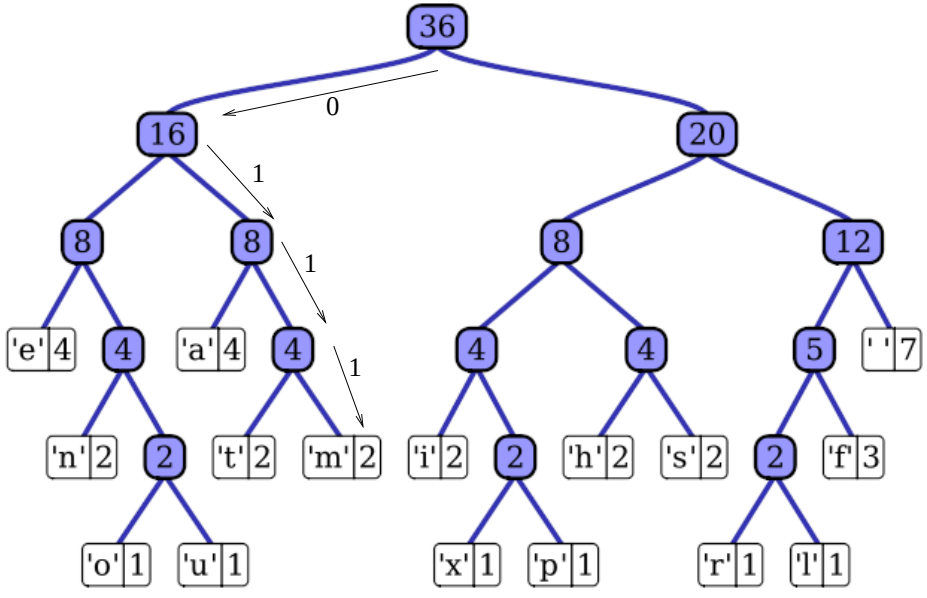
Soit X une v.a. discrète sur Σ, et $p_a = Pr[X = a]$, alors:

$$H(X) = -\sum_{a \in \Sigma} p_a \log p_a$$

Mesure la "quantité d'information" contenue dans la source X.
Note: log = log base 2: $H(X)$ se mesure en *bits*.

Compression de Huffman

Soit un texte, où $p_{\mathtt{a}} = \frac{4}{36}$, $p_{\mathtt{f}} = \frac{3}{36}$, $p_{\mathtt{r}} = p_{\mathtt{l}} = \frac{1}{36}$, etc.



On code chaque lettre par la suite de bits représentant son chemin d'accès (0=gauche, 1=droite) depuis la racine:

m	a	t	h	e	u	x
0111	010	0110	1010	000	00111	10010

Compression de Huffman

Théorème

Pour toute longueur $n \in \mathbb{N}$,

$$H(X) \leq \frac{1}{n} E(\#\text{bits dans } code(X^n)) < H(X) + 1$$

Ex.1: $X =$ distribution uniforme sur Σ.

$$H(X) = -\sum_{a \in \Sigma} \frac{1}{|\Sigma|} \log \frac{1}{|\Sigma|} = \log |\Sigma|$$

Arbre de Huffman = arbre binaire complet de profondeur $\log |\Sigma|$.

Ex.2: (dégénéré) $X = \delta_a$ sur une lettre a.

$$H(X) = -1 \ \log 1 = 0$$

Et $code(a^n) =$ mot vide, longueur 0.

(il faudra coder la longueur n séparément pour désambiguer...)

Compression

En général, soit la capacité d'un canal (=code)
$C = \lim_{n \to +\infty} \frac{\log N(n)}{n}$, où $N(n) =$ nombre de messages codés possibles de n bits.

Théorème [Shannon 48, Thm.9]

1. Il n'y a pas de code tel que $C > H$;
2. Pour tout $\epsilon > 0$, il existe un code tel que $C < (1 + \epsilon)H$.

(Pour 2, Huffman par exemple, qui est optimal, ou Lempel-Ziv.)
Moralité: $H(X)$ est le nombre minimal de bits par lettre nécessité en moyenne par un code (en l'absence de bruit).

Propriétés de l'entropie

Théorème

1. $H(X) \geq 0$, égalité ssi X est un Dirac.
2. $H(X) \leq \log |\Sigma|$, égalité ssi X uniforme.
3. Soit $H(X,Y) = -\sum_{a \in \Sigma, b \in \Sigma'} p_{ab} \log p_{ab}$ l'entropie *mutuelle*, alors
$$H(X,Y) \leq H(X) + H(Y),$$
égalité si X et Y indépendantes (i.e., si $p_{ab} = p_a p_b$ pour tous a, b).

Inégalité de Locatelli-Mabon

Proposition

Soit X de loi $(p_a)_{a\in\Sigma}$. Pour toute loi p_a, $a \in \Sigma$,

$$H(X) \leq -\sum_{a\in\Sigma} p_a \log q_a$$

Démonstration:

$$\begin{aligned} H(X) + \sum_{a\in\Sigma} p_a \log q_a &= \sum_{a\in\Sigma} p_a \log\left(\frac{q_a}{p_a}\right) = \sum_{a\in\Sigma} p_a \ln\left(\frac{q_a}{p_a}\right) \frac{1}{\ln 2} \\ &\leq \sum_{a\in\Sigma} p_a \left(\frac{q_a}{p_a} - 1\right) \frac{1}{\ln 2} \\ &= \left(\sum_{a\in\Sigma} q_a - \sum_{a\in\Sigma} p_a\right) \frac{1}{\ln 2} = 0 \quad \square \end{aligned}$$

Propriétés de l'entropie (suite)

De $H(X) \leq -\sum_{a\in\Sigma} p_a \log q_a$, on déduit:

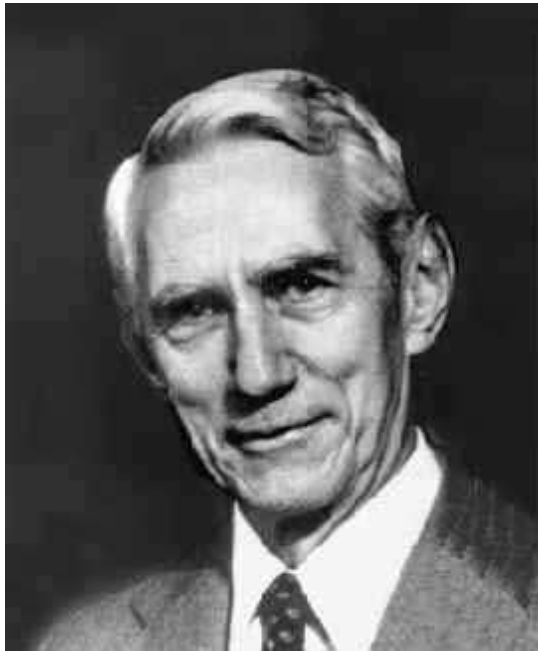
- $H(X) \leq \log n$, où $n = |\Sigma|$: prendre $q_a = \frac{1}{n}$.
- $H(X) < \log n$ sauf si loi uniforme.
  Démonstration: si $(p_a)_{a\in\Sigma}$ pas uniforme, on peut trouver b, c avec $p_b > p_c$.
  Pour $\epsilon > 0$ assez petit,
  $p_b \log(1+n\epsilon) + p_c \log(1-n\epsilon) \sim \frac{(p_b-p_c)n\epsilon}{\ln 2} > 0$, donc pour $q_b = \frac{1}{n}+\epsilon$, $q_c = \frac{1}{n}-\epsilon$, $q_a = \frac{1}{n}$ $(a \neq b, c)$:

$$\begin{aligned} H(X) &\leq -p_b \log q_b - p_c \log q_c - \sum_{a\notin\{b,c\}} p_a \log \frac{1}{n} \\ &= -p_b \log(1+n\epsilon) - p_c \log(1-n\epsilon) + \log n < \log n. \end{aligned}$$

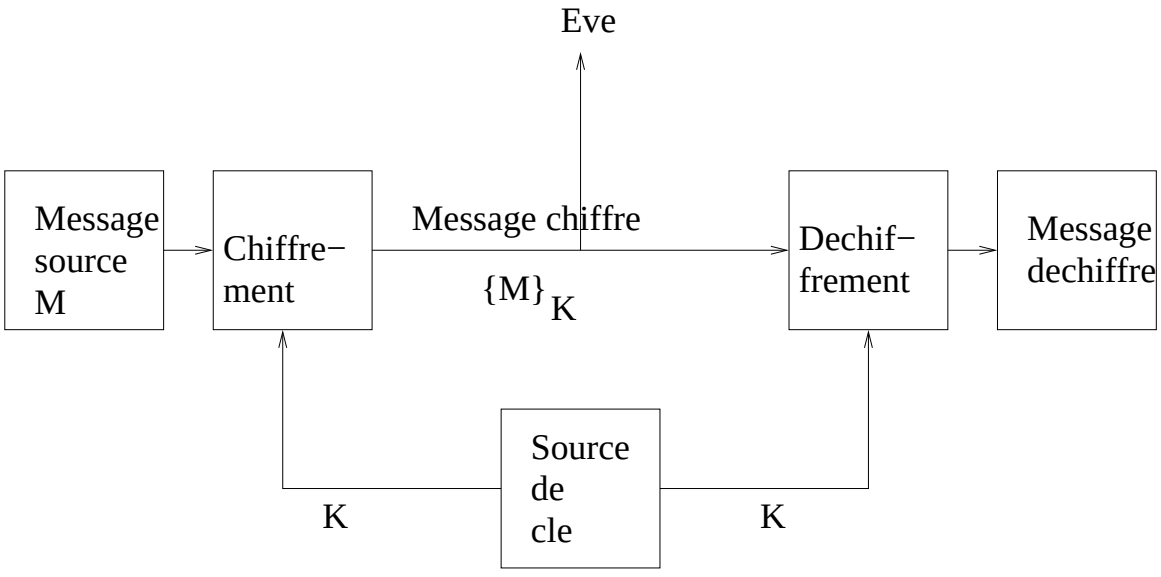
□

Théorie du chiffrement

Claude E. Shannon.
Communication Theory of Secrecy Systems.
Bell System Technical Journal, 28(4):656–715, 1949.



Théorie de l'information et chiffrement



On va mesurer le nombre de bits que l'on ne connaît pas de la source $X = M$, connaissant le chiffré $Y = \{M\}_K$ par l'entropie conditionnelle.

Entropie conditionnelle

Soit X la v.a. source (clairs), Y la v.a. destination (chiffrés).

Definition (Entropie conditionnelle)

Soit $H(X \mid Y = b)$ l'entropie de la loi $X \mid Y = b$, i.e., $-\sum_{a \in \Sigma} p(a \mid b) \log p(a \mid b)$, où $p(a \mid b) = Pr[X = a \mid Y = b]$.
L'entropie conditionnelle est:

$$H(X \mid Y) = \sum_{b \in \Sigma'} p_b H(X \mid Y = b)$$

Mesure le degré d'ambiguïté sur X, ayant observé Y.

Théorème

$$H(X \mid Y) = H(X, Y) - H(Y)$$

Entropie conditionnelle

Ex.1 Si X et Y indépendantes, $H(X \mid Y) = H(X)$: tous les bits de X sont irrécupérables.

Ex.2 Si $X = f(Y)$ (f fonction de décodage, sans erreur), alors $H(X,Y) = -\sum_{b \in \Sigma'} p_{f(b)b} \log p_{f(b)b} = H(Y)$, donc $H(X \mid Y) = 0$: tous les bits de X sont récupérables.

Ex.3 Si $Y = g(X)$ (fonction déterministe, sans erreur), alors $H(X,Y) = H(X)$, donc $H(X \mid Y) = H(X) - H(Y)$.

Ex.4 Si $Y = g(X)$, g injective, $H(X \mid Y) = 0$, on peut récupérer tous les bits de X sachant Y.

Propriétés de l'entropie conditionnelle

Théorème

1. $H(X) \geq H(X \mid Y)$, égalité ssi X et Y indépendantes;
2. $H(X,Y) = H(Y) + H(X \mid Y) = H(X) + H(Y \mid X)$.

Démonstration.

1. $H(Y) \geq 0$, égalité ssi $H(X,Y) = H(X) + H(Y)$.
2. Définition + $H(X,Y) = H(Y,X)$. □

Ce que fait Eve

À l'étape n, Eve a vu le préfixe Y_n de longueur n du chiffré $Y = \{M\}_K$, et cherche à retrouver $X = M$.

- Initialement ($n = 0$), $H(X \mid Y_0) = H(X)$.
  Pas de stratégie meilleure que de parier sur les clairs les plus fréquents.
- Ensuite, $H(X) \geq H(X \mid Y_1) \geq H(X \mid Y_n) \geq \ldots$
- S'il existe n tel que $H(X \mid Y_n) = 0$, alors $X = M$ est déterminé de façon unique en fonction de Y_n.

Ce que fait Eve

À l'étape n, Eve a vu le préfixe Y_n de longueur n du chiffré $Y = \{M\}_K$, et cherche à retrouver $X = M$.

- Initialement ($n = 0$), $H(X \mid Y_0) = H(X)$.
  Pas de stratégie meilleure que de parier sur les clairs les plus fréquents.
- Ensuite, $H(X) \geq H(X \mid Y_1) \geq H(X \mid Y_n) \geq \ldots$
- S'il existe n tel que $H(X \mid Y_n) = 0$, alors $X = M$ est déterminé de façon unique en fonction de Y_n.
- Plus intelligent: s'il existe n tel que $H(X \mid Y_n) \leq k$ (faible), alors on n'a que 2^k bits de X à énumérer en moyenne pour trouver le bon étant donné Y (par force brute).

Forces et faiblesses de l'approche de Shannon

+ Théorie mathématique quantifiant exactement le niveau de sécurité d'un schéma de chiffrement.
  - Masque jetable: $H(X \mid Y_n) = H(X)$ pour tout n (inconditionnellement sûr).
  - Permet d'analyser la taille n à partir de laquelle $H(X \mid Y_n)$ devient trop faible, et où il faut changer la clé [Shannon49].
– Ne traite que d'attaquants passifs.
– Ne traite que du secret.
– Ignore la difficulté calculatoire du déchiffrement/décryptage.

Outline

Difficulté calculatoire

La théorie de Shannon suppose que s'il existe un unique X tel que $f(X) = Y$ (Y donné), alors on peut le retrouver.

Calculatoirement, c'est irréaliste.

Etude de cas: X, Y de n bits. Il suffit d'énumérer tous les X, et de tester si $f(X) = Y$ jusqu'à ce qu'on trouve celui qui convient.

Temps: $O(2^n)$, en supposant le calcul de f en temps constant.

Difficulté calculatoire: complexité

Supposons qu'on énumère un X chaque picoseconde.

n	10	20	30	40	50	60	70	80	90	100
Temps polynomial										
n^2	0,1ns	0,4ns				...				10ns
n^3	1ns	8ns				...				1μs
n^4	10ns	0,16μs				...				0,1ms
n^6	1μs	64μs				...				1s
Temps exponentiel										
2^n	1 ns	1 μs	1 ms	1 s	18,8 min	13,3 j	37,4 ans	38,3 k.ans	39,3 M.ans	2,7 u

Difficulté calculatoire: complexité

Supposons qu'on énumère un X chaque picoseconde.

n	10	20	30	40	50	60	70	80	90	100
Temps polynomial										
n^2	0, 1ns	0, 4ns				...				10ns
n^3	1ns	8ns				...				1μs
n^4	10ns	0, 16μs				...				0, 1ms
n^6	1μs	64μs				...				1s
Temps exponentiel										
2^n	1 ns	1 μs	1 ms	1 s	18, 8 min	13, 3 j	37, 4 ans	38, 3 k.ans	39, 3 M.ans	2, 7 u

Dogme

- temps polynomial = tractable,
- temps exponentiel=intractable.

Fonctions à sens unique

La cryptographie (classique) moderne est fondée sur la notion:

Definition (Fonction à sens unique)

$f : \Sigma^* \to \Sigma^*$ est à sens unique ssi:

- $f(x)$ est calculable en temps **polynomial** (en $|x|$);
  (Il est facile de chiffrer.)
- il n'existe pas d'algorithme en temps **polynomial** **randomisé** calculant un x tel que $f(x) = y$.
  (Il est difficile de décrypter.)

Fonctions à sens unique à trappe

Oups, en fait sur:

Definition (Fonction à sens unique à trappe)

$f : \Sigma^* \times \Sigma^* \to \Sigma^*$ est à sens unique et à trappe ssi:

- $f(x,k)$ est calculable en temps **polynomial** (en $|x| + |k|$);
  (Il est facile de chiffrer.)
- pour tout k, il existe g_k telle que $f(g_k(y),k) = y$ pour tous $y = f(x,k)$, calculable en temps **polynomial**;
  (Il est facile de déchiffrer.)
- il n'existe pas d'algorithme en temps **polynomial randomisé** calculant un x tel que $f(x,k) = y$.
  (Il est difficile de décrypter.)

Outline

Chiffrement symétrique

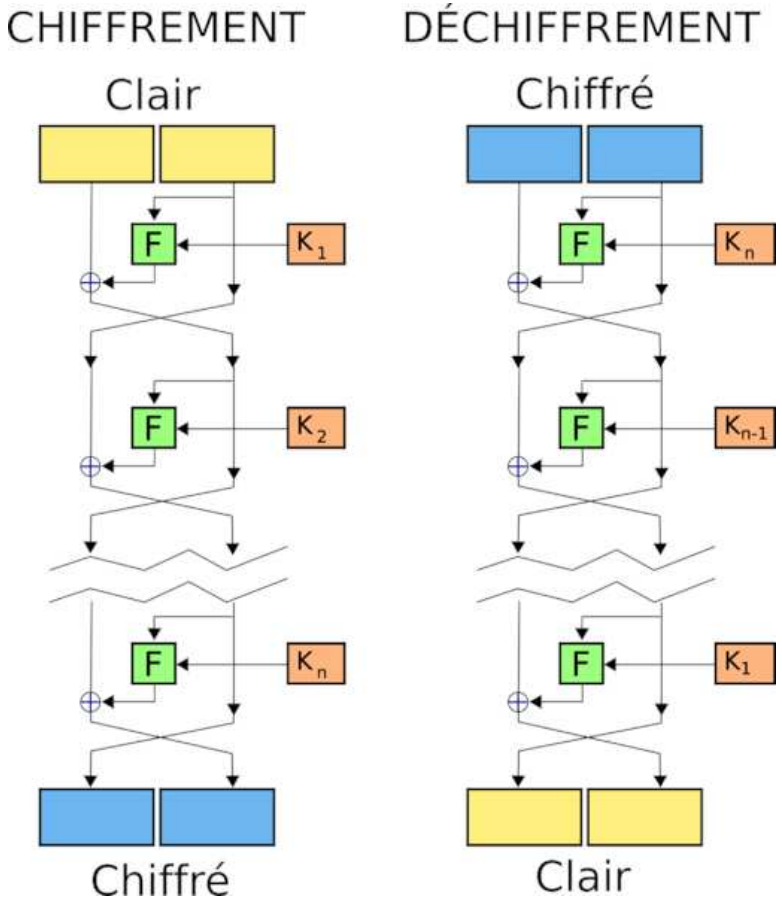
On dit qu'un schema de chiffrement f est symétrique si $g_k(y)$ est une fonction $g(y,k)$—en temps **polynomial**.
Autrement dit, si on n'a besoin que de la clé pour déchiffrer ($k^{-1} = k$).

Exemples:

- DES [IBM, NSA76] (considéré cassé aujourd'hui);
- Triple DES [Tuchman99];
- AES, a.k.a., Rijndael [DR00, NIST00].
- xTEA [NeedhamWheeler97].

Note: aucun n'est prouvé à sens unique, mais sont conçus pour résister à des familles d'attaques connues.

Chiffrements de Feistel



- $K_1, \ldots, K_n$ obtenues par expansion de la clé K ;
- F donnée typiquement par des tables (S-boxes);
- $\oplus$ opération de groupe (XOR);
- chiffre des blocs de taille fixe (ex: 2×32 bits).

xTEA: le plus court (pas Feistel)

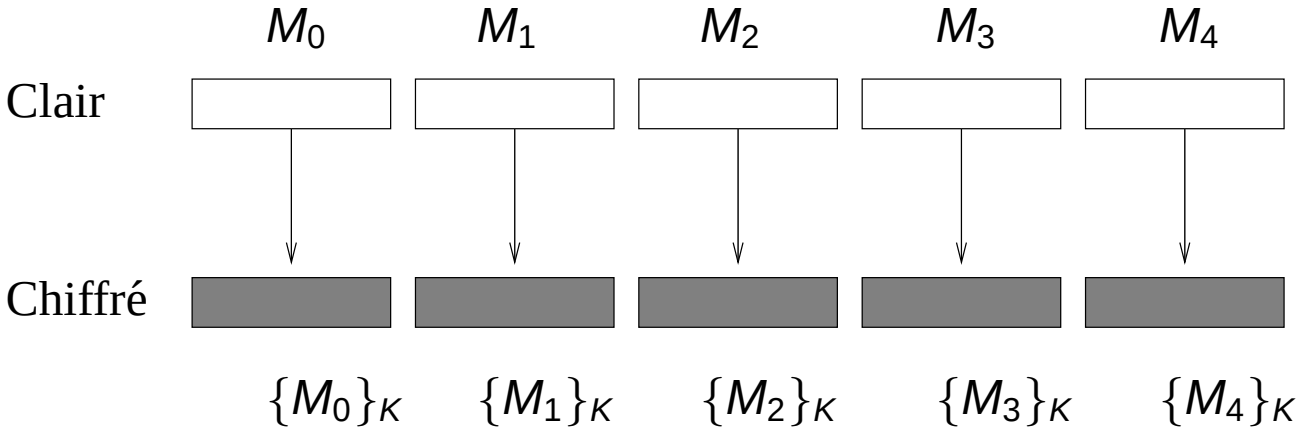
```
void encipher(unsigned int num_rounds, unsigned long* v, unsigned long* k) {
    unsigned long v0=v[0], v1=v[1], i;
    unsigned long sum=0, delta=0x9E3779B9;
    for(i=0; i<num_rounds; i++) {
        v0 += ((v1 << 4 ^ v1 >> 5) + v1) ^ (sum + k[sum & 3]);
        sum += delta;
        v1 += ((v0 << 4 ^ v0 >> 5) + v0) ^ (sum + k[sum>>11 & 3]);
    }
    v[0]=v0; v[1]=v1;
}

void decipher(unsigned int num_rounds, unsigned long* v, unsigned long* k) {
    unsigned long v0=v[0], v1=v[1], i;
    unsigned long delta=0x9E3779B9, sum=delta*num_rounds;
    for(i=0; i<num_rounds; i++) {
        v1 -= ((v0 << 4 ^ v0 >> 5) + v0) ^ (sum + k[sum>>11 & 3]);
        sum -= delta;
        v0 -= ((v1 << 4 ^ v1 >> 5) + v1) ^ (sum + k[sum & 3]);
    }
    v[0]=v0; v[1]=v1;
}
```

Modes de chiffrement

Mode

Mode de chiffrement=algorithme convertissant un algo de chiffrement par blocs en un algo de chiffrement de blocs de taille arbitraire.

Ex: ECB (Electronic Code Book).



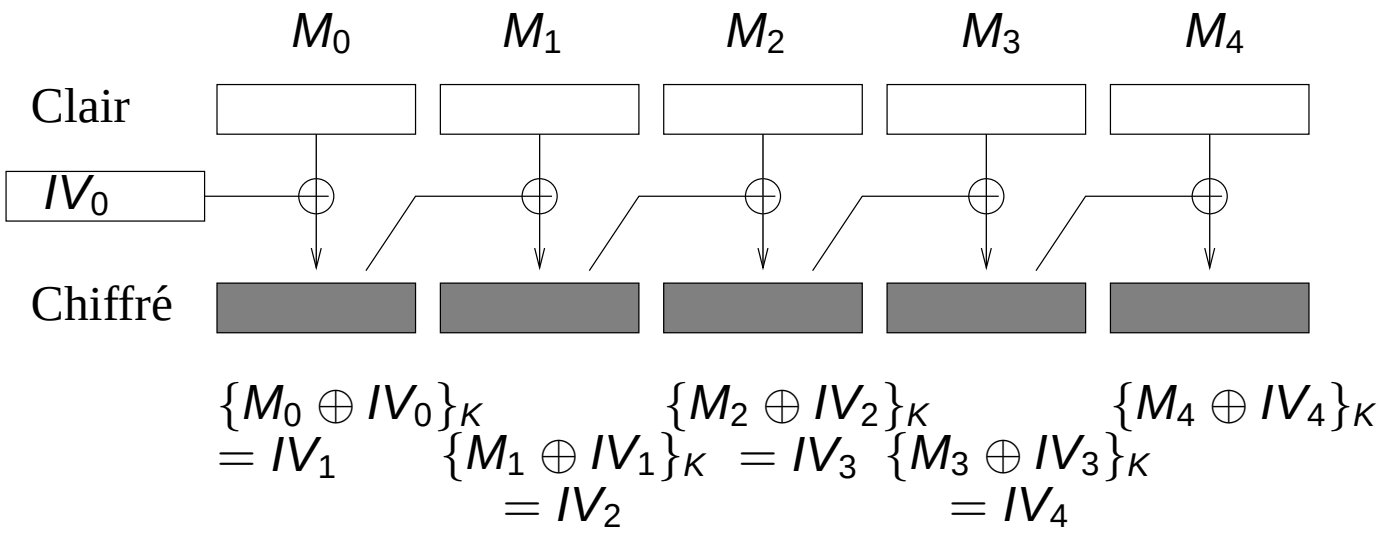
Note: ECB a un problème d'intégrité: lequel?

Modes de chiffrement

Mode

Mode de chiffrement=algorithme convertissant un algo de chiffrement par blocs en un algo de chiffrement de blocs de taille arbitraire.

Ex: CBC (Cipher Block Chaining)



Outline

Génération de clé partagée

- A et B veulent fabriquer une clé K fraîche;
- Personne d'autre ne doit connaître K !

Scénarios:

Génération de clé partagée

- A et B veulent fabriquer une clé K fraîche;
- Personne d'autre ne doit connaître K !

Scénarios:

- A fabrique K, l'envoie à B chiffrée. . . avec quelle clé?

Génération de clé partagée

- A et B veulent fabriquer une clé K fraîche;
- Personne d'autre ne doit connaître K !

Scénarios:

- A fabrique K, l'envoie à B chiffrée. . . avec quelle clé?
- Un camion de la Brinks?

Génération de clé partagée

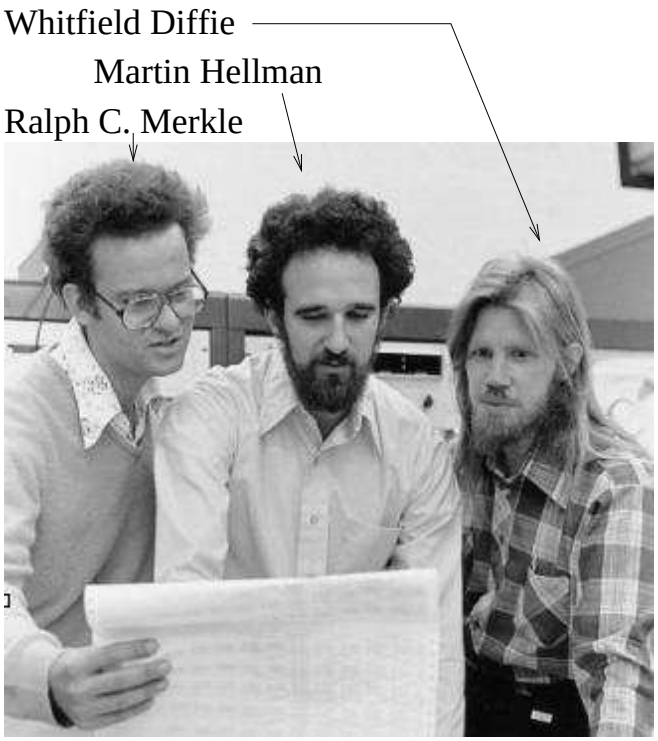
- A et B veulent fabriquer une clé K fraîche;
- Personne d'autre ne doit connaître K !

Scénarios:

- A fabrique K, l'envoie à B chiffrée. . . avec quelle clé?
- Un camion de la Brinks?
- Le protocole de Diffie-Hellman [1978].

Le protocole de Diffie-Hellman(-Merkle)

Whitfield Diffie, Martin E. Hellman.
New Directions in Cryptography.
IEEE Trans. Inf. Theory, IT-22:644-654, 1976.



Le protocole de Diffie-Hellman(-Merkle)

- Choisir un groupe cyclique ($\mathbb{Z}/p\mathbb{Z}^*$, p premier grand), et g un générateur du groupe: $\{g^n \mid n = 0 \dots p-2\} = \mathbb{Z}/p\mathbb{Z}^*$.
  Ex: $p = 5$, $g = 2$: $g^0 = 1$, $g^1 = 2$, $g^2 = 4$, $g^3 = 3$.
- Représenter des blocs M de $k \sim \log p$ bits comme des nombres écrits en binaire.

A	$\to$	B :	g^{x_A}	mod p	x_A aléatoire dans $0 \dots p-2$
B	$\to$	A :	g^{x_B}	mod p	x_B aléatoire dans $0 \dots p-2$

La clé partagée est $K = g^{x_A \cdot x_B}$.

Exercice: comment A et B calculent-ils K efficacement?

(Note: l'exponentiation mod p s'effectue en temps polynomial.)

Logarithmes discrets

Definition

Le logarithme discret $\log_g y$ de $y \in \mathbb{Z}/p\mathbb{Z}^*$ est l'unique $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ tel que $g^x = y \mod p$.

(Unique car g *générateur* du groupe $\mathbb{Z}/p\mathbb{Z}^*$.)

DL Cassé $\Rightarrow$ DH Cassé

Si l'on sait calculer $\log_g y$ efficacement, alors DH est cassé.

Démonstration: l'attaquant peut calculer $x_A = \log_g(g^{x_A})$, $x_B = \log_g(g^{x_B})$, puis la clé "secrète" $g^{x_A \cdot x_B}$. □

Difficulté du logarithme discret

Conjecture

Il n'existe pas d'algorithme en temps polynomial randomisé calculant $\log_g$ avec forte probabilité.

- Meilleur algorithme générique: pas-de-bébé-pas-de-géant, temps $O(\sqrt{p})$ (**exponentiel**!).

Le modèle générique

Victor Shoup.
Lower bounds for discrete logarithms and related problems.
Eurocrypt'97. Springer-Verlag LNCS 1233, 256-266, 1997.

Le modèle générique

Definition

Un algorithme est générique ssi les seuls traitements qu'il effectue sur les éléments du groupe $\mathbb{Z}/p\mathbb{Z}^*$ sont les opérations de groupe: 1, $\times$, $\_^{-1}$.

(A l'air raisonnable. . . mais interdit de voir $x \in \mathbb{Z}/p\mathbb{Z}^*$ comme un entier entre 0 et $p-1$, calculer dessus dans $\mathbb{Z}$, puis projeter mod p, par exemple.)

Le modèle générique

Theorem (Shoup 1997)

La sécurité de Diffie-Hellman est équivalente à la difficulté de casser le logarithme discret, dans le modèle générique.

Plus précisément, on montre que DDH nécessite $\Omega(\sqrt{q})$ opérations en moyenne, où q est le plus grand diviseur premier de $p-1$, et DDH est:
ENTRÉE: $g^{x_A}, g^{x_B}, r \mod p$;
SORTIE: $r = g^{x_A \cdot x_B} \mod p$?

Présentations de groupe

- A isomorphisme près, il n'existe qu'un groupe cyclique d'ordre $p-1$.
- Mais la présentation du groupe est importante calculatoirement.

Ex: $x \mapsto g^x \mod p$ est un isomorphisme de $\mathbb{Z}/(p-1)\mathbb{Z}$ sur $\mathbb{Z}/p\mathbb{Z}^*$:

- $g^x \mod p$ facile à calculer;
- l'inverse $\log_g y$ difficile à calculer.

Il est donc a priori intéressant d'explorer d'autres présentations du groupe cyclique d'ordre $p-1$.

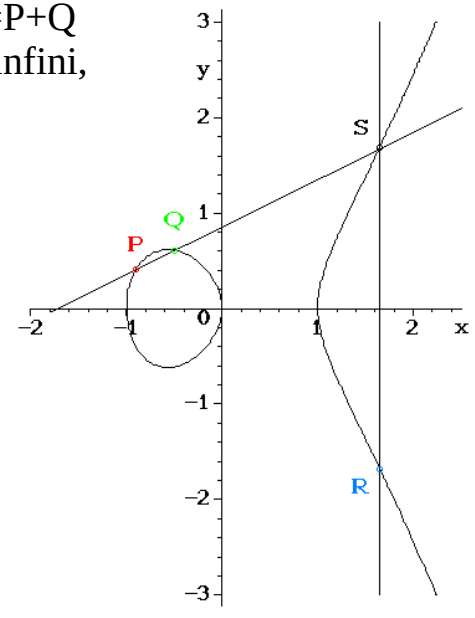
Courbes elliptiques

La version classique [Weierstrass]. Le lieu des points

$$y^2 = x^3 + ax + b$$

dans $\mathbb{R}^2$ + un point à l'infini.
(Avec $\Delta = -16(4a^3 + 27b^2) \neq 0$.)

Calcul de R=P+Q
(0=point a l'infini,
−R=S.)



Theorem

Toute courbe elliptique est un groupe commutatif.

Courbes elliptiques

Les formules explicites:

$$\begin{array}{rclcrcl} x_S & = & s^2 - x_P - x_Q & & x_R & = & x_S \\ y_S & = & y_P + s(x_P - x_S) & & y_R & = & -y_S \end{array}$$

où s (slope) $= \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} & \text{si } x_P \neq x_Q \\ \frac{3x_P^2 + a}{2y_P} & \text{si } x_P = x_Q, y_P = y_Q \end{cases}$

(si $x_P = x_Q$, $y_P \neq y_P$, alors $y_Q = -y_P$ et la somme est le point à l'infini.)

s'interprètent dans n'importe quel corps, pas forcément $\mathbb{R}$...

par exemple $\mathbb{Z}/p\mathbb{Z}$, ou plutôt un corps de Galois $\mathbb{F}_q$ (par ex., $q = 256$).

(...en caractéristique 2, l'équation de base est plus compliquée).

Diffie-Hellman par courbes elliptiques

EC-DH

On choisit une courbe elliptique sur $\mathbb{F}_q$, un point P d'ordre n suffisamment grand. Notons $x.P = \underbrace{P + P + \ldots + P}_{x \text{ fois}}$.

A	$\rightarrow$	B :	$x_A.P$	x_A aléatoire dans $0 \ldots n-1$
B	$\rightarrow$	A :	$x_B.P$	x_B aléatoire dans $0 \ldots n-1$

La clé partagée est le point $x_A x_B.P$.

Note: Une courbe elliptique sur $\mathbb{F}_q$ n'est pas en général un groupe cyclique, mais $\mathbb{Z}.P$ en est un sous-groupe cyclique.
Sécurité: difficulté du log discret sur la courbe, i.e., étant donné $n.P$, il est difficile de retrouver n + modèle générique plus réaliste ici.

Quelques remarques

- On considère qu'on obtient un niveau de sécurité équivalent avec moins de bits de clé si on utilise une (bonne) courbe elliptique que $\mathbb{Z}/p\mathbb{Z}^*$.
- Comment calcule-t-on un bon nombre aléatoire?

Outline

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

198 089 357 218 068 216 517 260 685 773 624 326 765

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

198 089 357 218 068 216 517 260 685 773 624 326 765

324 073 163 420 648 383 635 925 543 613 609 217 851

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

198 089 357 218 068 216 517 260 685 773 624 326 765

324 073 163 420 648 383 635 925 543 613 609 217 851

232 992 305 167 558 465 512 230 572 597 182 844 496

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

198 089 357 218 068 216 517 260 685 773 624 326 765
324 073 163 420 648 383 635 925 543 613 609 217 851
232 992 305 167 558 465 512 230 572 597 182 844 496
285 921 562 481 663 339 762 468 133 626 338 094 559

Nombres aléatoires

On souhaite fabriquer une suite de nombres aléatoires (uniforme) à n bits (ici $n = 128$):

198 089 357 218 068 216 517 260 685 773 624 326 765
324 073 163 420 648 383 635 925 543 613 609 217 851
232 992 305 167 558 465 512 230 572 597 182 844 496
285 921 562 481 663 339 762 468 133 626 338 094 559
311 193 852 387 359 001 535 265 578 684 927 465 138

. . .

Sources de nombres aléatoires

Sources physiques:

- Sources de bruit blanc (classiques);
  Débit faible, et ayant souvent un biais statistique.
  Ex: `/dev/random` produit $\sim$ 160 bits par seconde.
  (Expérience sur 60 sec., Dell D620, Intel Core 2 T7200, 2GHz.)
- Fluctuations quantiques;
  Débit important: $\sim 100 Mb/s$ pour quelques centaines d'euros en 2010 (voir F. Grosshans).

Sources de nombres aléatoires

Sources physiques:

- Sources de bruit blanc (classiques);
  Débit faible, et ayant souvent un biais statistique.
  Ex: `/dev/random` produit $\sim$ 160 bits par seconde.
  (Expérience sur 60 sec., Dell D620, Intel Core 2 T7200, 2GHz.)
- Fluctuations quantiques;
  Débit important: $\sim 100 Mb/s$ pour quelques centaines d'euros en 2010 (voir F. Grosshans).

Definition

Un générateur pseudo-aléatoire est un algorithme prenant en entrée une *graine* s (vraiment aléatoire) et sortant une suite $x_0(s), x_1(s), \ldots, x_n(s)$ de nombres "ayant les mêmes propriétés qu'une suite de nombres aléatoires".

Générateurs pseudo-aléatoires

- Le but est d'améliorer le débit:
  - Tirer une graine s_0 au hasard;
    Emettre $x_0(s_0), x_1(s_0), \ldots, x_{1000}(s_0)$.
  - Tirer une graine s_1 au hasard;
    Emettre $x_0(s_1), x_1(s_1), \ldots, x_{1000}(s_1)$.
  - Etc.

Ex: (générateur linéaire congruentiel, ou LCG)

$$x_n(s) = f^n(s), \quad f(x) = ax + b \mod m$$

Le LCG: pourquoi pas?

$$x_{n+1} = ax_n + b \mod m$$

avec $\text{pgcd}(b, m) = 1$, $a - 1$ divisible par tous les facteurs premiers de m, $a - 1$ multiple de 4 ssi m multiple de 4.

[Knuth, ACP vol.2, 1997]

Sous ces conditions, le générateur a période pleine: $(x_n)_{n=0,\ldots,m-1}$ énumère $\{0, \ldots, m-1\}$.

Ceci est **nécessaire.** Si $(x_n)_{n\in\mathbb{N}}$ doit "avoir les mêmes propriétés qu'une suite aléatoire", alors pour tout i, $Freq[x_n = i] = Pr_r[r = i] = \frac{1}{m} \neq 0$.

Le LCG: analyse fréquentielle

[Knuth, ACP vol.2, 1997]

Le LCG a une distribution empirique uniforme:

$$Freq[x_n = i] = \frac{1}{m}$$

Il passe d'autre part toute une série de tests statistiques (χ^2, Student, etc.)

Le LCG: analyse fréquentielle

[Knuth, ACP vol.2, 1997]

Le LCG a une distribution empirique uniforme:

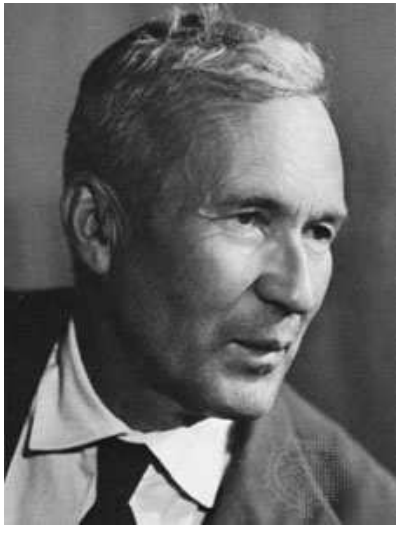
$$Freq[x_n = i] = \frac{1}{m}$$

Il passe d'autre part toute une série de tests statistiques (χ^2, Student, etc.)
Mais a certaines anomalies statistiques:

Theorem (Marsaglia)

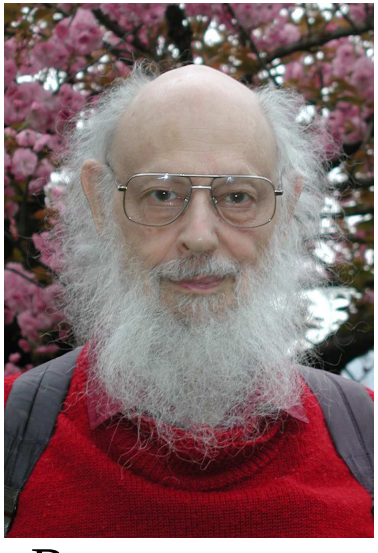
Si l'on tire des points dans $[0,1]^n$ *en utilisant le LCG, ces points appartiendront à l'union d'au plus* $m^{1/n}$ *hyperplans.*

Qu'est-ce que l'aléatoire?



Andrei Nikolaievitch Kolmogorov

Gregory Chaitin



Ray Solomonoff

Per Martin−Lof

Qu'est-ce que l'aléatoire?

Theorem (Kolmogorov-Chaitin-Solomonoff)

Une suite $(x_n)_{n\in\mathbb{N}}$ est aléatoire ssi l'une des deux propriétés équivalentes tient:

- *La suite est incompressible: il n'y a aucun algorithme $\mathcal{A}_n$ calculant x_n qui soit significativement plus court que x_n.*
- *La suite est imprédictible: il n'y a aucun algorithme $\mathcal{A}_n$ qui, étant donnés $x_0, x_1, \ldots, x_{n-1}$, calcule x_n.*

(Formulation précise et détails techniques omis.)

Le LCG: un très mauvais générateur

Fait

Le LCG:

$$x_{n+1} = ax_n + b \mod m$$

est facile à prédire... donc très loin de ressembler à une source aléatoire.

Démonstration. Connaissant x_0, x_1, et x_2,

$$a = \frac{x_2 - x_1}{x_1 - x_0} \qquad b = x_2 - ax_1$$

Maintenant l'intrus connaît x_3, x_4, etc. □

Application à Diffie-Hellman

A	$\rightarrow$	B :	g^{x_A}	mod p	x_A fourni par un LCG
B	$\rightarrow$	A :	g^{x_B}	mod p	x_B fourni par un LCG

- Si un attaquant prend connaissance des premières valeurs du LCG, il peut reconstruire x_A, x_B, donc la clé “secrète” $g^{x_A.x_B} \mod p$.

Application à Diffie-Hellman

A	$\rightarrow$	B :	g^{x_A}	mod p	x_A fourni par un LCG
B	$\rightarrow$	A :	g^{x_B}	mod p	x_B fourni par un LCG

- Si un attaquant prend connaissance des premières valeurs du LCG, il peut reconstruire x_A, x_B, donc la clé “secrète” $g^{x_A.x_B} \mod p$.
- Même sans ça, $g^{ax_n+b} = (g^{x_n})^a.g^b \mod p \ldots$ (le $\mod m$ dans l’exposant est un peu plus compliqué à traiter).

Ressembler à l'aléatoire

Definition

Un générateur pseudo-aléatoire $b_n(s)$ est cryptographiquement sûr ssi pour tout algorithme en temps polynomial randomisé $\mathcal{A}(x, r)$ (x entrée, r suite aléatoire utilisée par $\mathcal{A}$),

$$Pr_{r,s}[\mathcal{A}(x, r) \neq \mathcal{A}(x, b_0(s)b_1(s)\ldots b_n(s)\ldots)]$$

est négligeable (inférieure asymptotiquement à tout inverse de polynôme, lorsque $|x| \to +\infty$).

I.e., on peut remplacer l'aléa r par la suite pseudo-aléatoire $b_0(s)b_1(s)\ldots b_n(s)\ldots$ sans qu'aucun attaquant (en temps polynomial) ne voie de différence exploitable.
Note: En particulier, $b_n(s)$ est imprédictible par une machine en temps polynomial.

Le générateur de Blum-Micali

Soit $N = pq$, produit de deux grands nombres premiers (disons 128 bits). Soit n entier assez grand.
À partir d'un nombre aléatoire s, poser

$$x_n = s \qquad x_{i-1} = x_i^2 \mod N \quad (i = N..2)$$

On tire ensuite $x_1, x_2, x_3, ..., x_n$ dans l'ordre inverse.

Theorem (Blum-Micali 1984)

Si ceci est prédictible en temps polynomial, alors on sait factoriser N.

Conséquence: prédire Blum-Micali est au moins aussi difficile que factoriser de grands nombres produits de deux premiers, un problème réputé dur.

(. . . sauf sur une machine quantique [Shor 1994].)

Variante: $x_{i-1} = g^{x_{i_2}} \mod p$, au moins aussi difficile que le log discret.

Outline

Chiffrement asymétrique

- Chiffrement asymétrique: la clé de déchiffrement K^{-1} n'est plus la même que celle de chiffrement K.
- Principe proposé par [DiffieHellman76] (encore).
- À l'époque, aucune réalisation pratique connue.

Chiffrement asymétrique et partage de clés

Le chiffrement asymétrique résout (en principe) le problème de partage de clés — en n'en ayant plus besoin.

- B fabrique un couple de clés K_B^{-1} (privée), K_B (publique — diffusée à tout le monde).
- Pour chiffrer M:

$$A \to B : \{M\}_{K_B}$$

- Seul B ayant K_B^{-1} peut déchiffrer: confidentialité.

Chiffrement asymétrique et partage de clés

Le chiffrement asymétrique résout (en principe) le problème de partage de clés — en n'en ayant plus besoin.

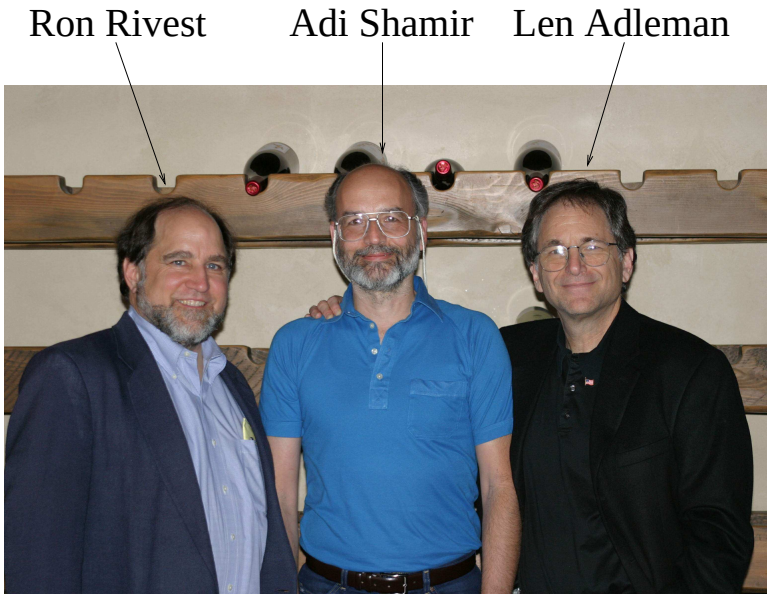
- B fabrique un couple de clés K_B^{-1} (privée), K_B (publique — diffusée à tout le monde).
- Pour chiffrer M:

$$A \rightarrow B : \{M\}_{K_B}$$

- Seul B ayant K_B^{-1} peut déchiffrer: confidentialité.
- **Note:** plus besoin de partager une clé commune — mais besoin de distribuer K_B de façon authentique.
- **Note:** n'importe qui peut chiffrer avec $K_B \Rightarrow$ aucune authentification.

Le chiffrement RSA

Ron Rivest, Adi Shamir, Len Adleman.
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.
Comm. ACM, 21(2):120-126, 1978



La fonction ϕ d'Euler

Definition (Totient)

La fonction totient d'Euler est $\phi(n) =$ nombre de k, $1 \leq k \leq n$ tels que $\text{pgcd}(k, n) = 1$.

Proposition

- $\phi(p) = p - 1$ si p premier;
- $\phi(p^k) = (p-1)p^{k-1}$ si p premier, $k \geq 1$;
- $\phi(p_1^{k_1} p_2^{k_2} \dots p_n^{k_n}) = \prod_{i=1}^{n} (p_i - 1) p_i^{k_i - 1}$.

En particulier, si $N = pq$ produit de deux nombres premiers, $\phi(N) = (p-1)(q-1)$.

Le petit théorème de Fermat

Version initiale: si p premier, alors $x^p = x \mod p$ pour tout x.
En particulier, $x^{p-1} = 1 \mod p$ pour tout $x \neq 0 \mod p$.

Theorem (Fermat-Euler)

Pour tout $N \geq 1$, pour tout x avec $pgcd(x, N) = 1$,

$$x^{\phi(N)} = 1 \mod N.$$

En particulier, $x^{(p-1)(q-1)} = 1 \mod pq$ (p, q premiers) dès que ni p ni q ne divise x.
Note: $\phi(N)$ est difficile à calculer en général.
On peut calculer $\phi(N)$ si on peut factoriser N, mais factoriser est considéré comme difficile.

Le chiffrement RSA

Génération de clés: tirer deux grands nombres premiers $p \neq q$. $N = pq$, $\phi(N) = (p-1)(q-1)$.
Tirer $d \in [2, \phi(N)[$ premier avec $\phi(N)$.
Calculer e tel que $de = 1 \mod \phi(N)$ (Bezout).
Clé *publique* $K_B = (N, e)$ *privée* $K_B^{-1} = d$.

- Chiffrement: $\{M\}_{K_B} = M^e \mod N$.
- Déchiffrement: $dec(M', K_B^{-1}) = M'^d \mod N$.

Proposition

$dec(\{M\}_{K_B}, K_B^{-1}) = M$.

Démonstration: $dec(\{M\}_{K_B}, K_B^{-1}) = (M^e)^d = M^{ed} \mod N$. Or $ed = 1 \mod \phi(N)$, donc $M^{ed} = M^1 = M \mod N$. □

RSA sur un exemple

$p = 11\,508\,288\,272\,609\,080\,559$, $q = 16\,540\,558\,193\,182\,354\,801$ (64 bits).
$N = 190\,353\,511\,877\,008\,536\,544\,166\,732\,523\,129\,413\,759$
$\phi(N) = 190\,353\,511\,877\,008\,536\,516\,117\,886\,057\,337\,978\,400$
$d = 68\,551\,770\,307\,036\,473\,070\,080\,515\,976\,000\,236\,897$,
$e = 71\,313\,093\,839\,083\,143\,987\,818\,093\,247\,265\,904\,033$.

- Clair $M =$ "`sol:1;2;1; 08:05`" =
  $153\,439\,759\,577\,269\,640\,773\,274\,743\,596\,375\,486\,517$.
- Chiffré $M' = M^e \mod N$
  $= 56\,605\,557\,572\,603\,317\,510\,437\,246\,813\,714\,386\,816$
  = "`*\149Õv2;~¾\130]f5\^Rö\Ô\128`".
- On déchiffre: $M'^d \mod N =$
  ...

RSA sur un exemple

$p = 11\,508\,288\,272\,609\,080\,559$, $q = 16\,540\,558\,193\,182\,354\,801$ (64 bits).
$N = 190\,353\,511\,877\,008\,536\,544\,166\,732\,523\,129\,413\,759$
$\phi(N) = 190\,353\,511\,877\,008\,536\,516\,117\,886\,057\,337\,978\,400$
$d = 68\,551\,770\,307\,036\,473\,070\,080\,515\,976\,000\,236\,897$,
$e = 71\,313\,093\,839\,083\,143\,987\,818\,093\,247\,265\,904\,033$.

- Clair $M =$ "`sol:1;2;1; 08:05`" = $153\,439\,759\,577\,269\,640\,773\,274\,743\,596\,375\,486\,517$.
- Chiffré $M' = M^e \mod N$
  $= 56\,605\,557\,572\,603\,317\,510\,437\,246\,813\,714\,386\,816$
  = "`*\149Õv2;~¾\130]f5\^Rö\Ô\128`".
- On déchiffre: $M'^d \mod N =$
  $\ldots 153\,439\,759\,577\,269\,640\,773\,274\,743\,596\,375\,486\,517 = M$.

Sécurité de RSA

- Plusieurs attaques connues: Wiener (si $d < \frac{1}{3}N^{1/4}$), Coppersmith (si e est petit), Håstad (par broadcast, si plusieurs participants ont la même clé publique), Franklin-Reiter, Boneh-Durfee-Frankel (si $e < \sqrt{N}$, N a n bits, et on connaît $n/4$ bits de d, on peut retrouver les autres), etc.
- RSA-OAEP: rend RSA sûr contre n'importe quel attaquant passif en temps polynomial randomisé, sous l'hypothèse RSA (on ne peut pas retrouver efficacement d [privée] à partir de N, e).
- En pratique, cassé si clés $\leq$768 bits, considéré sûr à partir de 1024 bits.

Outline

Authentication

- A souhaite prouver son identité à B.
- A souhaite prouver à B qu'il possède un droit sur une ressource.
- A souhaite prouver à B que le message M que B a reçu vient bien de A (et récemment).

$$A \rightarrow B : M$$

- ...

De nombreuses définitions de l'authentication (avec intégrité dans le 3ème cas).

Authentication simple

A souhaite prouver à B qu'il possède un droit sur une ressource. (Ex: login.)
Solution usuelle: A envoie à B un mot de passe.
Problèmes:

- Mots de passe faibles (facilement devinables).
- Rejeu des mots de passe (peut être évité par exemple par l'utilisation de mots de passe à usage unique).
- Etc.

Authentication à clé publique: signature

Encore Diffie et Hellman (1978). Signature RSA:

- A fabrique un couple de clés K_A^{-1} (privée), K_A (publique — diffusée à tout le monde).
- Pour signer M:

$$A \to B : M, \{M\}_{K_A^{-1}}$$

(Rappel: c'était $\{M\}_{K_B}$ pour chiffrer.)

- B reçoit M, M', et accepte si $dec(M', K_A) = M$: M' a forcément été créé avec la clé que seul A possède.
- Gère l'authentication simple, avec $M = \mathsf{OK}$ par ex.

Authentication à clé publique: signature

Encore Diffie et Hellman (1978). Signature RSA:

- A fabrique un couple de clés K_A^{-1} (privée), K_A (publique — diffusée à tout le monde).
- Pour signer M:

$$A \to B : M, \{M\}_{K_A^{-1}}$$

(Rappel: c'était $\{M\}_{K_B}$ pour chiffrer.)

- B reçoit M, M', et accepte si $dec(M', K_A) = M$: M' a forcément été créé avec la clé que seul A possède.
- Gère l'authentication simple, avec $M = \mathsf{OK}$ par ex.
- **Note:** c'est A qui crée les clés, pas B comme dans le chiffrement.
- **Note:** le message M est loin d'être confidentiel!

Signature RSA

En pratique, chiffrer M avec K_A^{-1} est trop long. On utilise une fonction de hachage h:

- $h(M)$ est un message court (quelques centaines de bits);
- Avec forte probabilité, $h(M) \neq h(N)$ (pas de collision);
- Il est difficile de calculer M tel que $h(M) = M'$ (résistance aux préimages);
- Il est difficile de calculer $N \neq M$ tel que $h(N) = h(M)$ (résistance aux secondes préimages).

Pour signer:

$$A \rightarrow B : M, \{M\}_{K_A^{-1}}$$

B reçoit M, M' et accepte si $dec(M', K_A) = h(M)$.

La signature Elgamal

Taher Elgamal.
A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.
IEEE Trans. Inf. Theory, 31(4):469-472, 1985.

La signature Elgamal

- N'est pas comme la signature RSA un chiffrement détourné.
  En particulier, ne permet pas de récupérer le message signé M.
  On notera $[M]_K$ la signature de M avec K, plutôt que $\{M\}_K$.
- Fondé sur la difficulté de calculer le logarithme discret.
  (dans $\mathbb{Z}/p\mathbb{Z}^*$, ou sur une courbe elliptique.)
- Base de l'algorithme de signature standard DSA (Digital Signature Algorithm).

La signature Elgamal

- Choisir un groupe cyclique ($\mathbb{Z}/p\mathbb{Z}^*$, p premier grand), et g un générateur du groupe.

			Création des clés:
A	$\rightarrow$	tous :	$g^x \mod p$ (x aléatoire) — clé secrète $K_A^{-1} = x$, — clé publique $K_A = g^x \mod p$,
			Signature:
A	$\rightarrow$	B :	$g^k, (h(M) - x.g^k).k^{-1} \mod p$ (k aléatoire uniforme, premier avec $p-1$) — tout ceci est la signature $[M]_{K_A^{-1}}$

B reçoit r, s et vérifie que $g^{h(M)} = K_A^r.r^s \mod p$, sachant M.

Chiffrer et signer

Folklore

Si on veut préserver à la fois la confidentialité et l'authentication, il faut signer puis chiffrer:

$$A \to B : \{M, [h(M)]_{K_A^{-1}}\}_{K_B}$$

(K_B publique, K_A^{-1} clé de signature privée.)

Exercice: Voyez-vous une attaque sur le schéma inverse:

$$A \to B : \{M\}_{K_B}, [h(\{M\}_{K_B})]_{K_A^{-1}}?$$

Chiffrer et signer

Folklore

Si on veut préserver à la fois la confidentialité et l'authentication, il faut signer puis chiffrer:

$$A \to B : \{M, [h(M)]_{K_A^{-1}}\}_{K_B}$$

(K_B publique, K_A^{-1} clé de signature privée.)

Exercice: Voyez-vous une attaque sur le schéma inverse:

$$A \to B : \{M\}_{K_B}, [h(\{M\}_{K_B})]_{K_A^{-1}}?$$

L'attaquant remplace la signature par la sienne propre, $[h(\{M\}_{K_B})]_{K_C^{-1}}$, et peut donc prétendre à être authentifié comme l'émetteur de M.

Chiffrer et signer

Folklore. . . mais Insuffisant!

Si on veut préserver à la fois la confidentialité et l'authentication, il faut signer puis chiffrer:

$$A \to B : \{M, [h(M)]_{K_A^{-1}}\}_{K_B}$$

(K_B publique, K_A^{-1} clé de signature privée.)

Exercice: Voyez-vous une attaque sur le schéma inverse:

$$A \to B : \{M\}_{K_B}, [h(\{M\}_{K_B})]_{K_A^{-1}}?$$

L'attaquant remplace la signature par la sienne propre, $[h(\{M\}_{K_B})]_{K_C^{-1}}$, et peut donc prétendre à être authentifié comme l'émetteur de M.

Signer-puis-chiffrer: attaques [Davis 2001]

Attaque man-in-the-middle (C est l'attaquant):

$$
\begin{array}{lcll}
C & \rightarrow & A: & \text{Hello} \\
A & \rightarrow & C: & \{\text{mon idée}\}_{K_C}, [h(\{\text{mon idée}]_{K_C})]_{K_A^{-1}} \\
 & & & \text{— } A \text{ essaie de convaincre } C \\
C & \rightarrow & B: & \{\text{mon idée}\}_{K_B}, [h(\{\text{mon idée}]_{K_B})]_{K_C^{-1}} \\
 & & & \text{— } C \text{ convainc } B \text{ que c'est son idée...}
\end{array}
$$

$\Rightarrow$ attaque sur l'authentication et sur la confidentialité.

Chiffrer-puis-signer: attaques [Davis 2001]

Encore une attaque man-in-the-middle:

$$\begin{aligned} A &\rightarrow C: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_C} \\ C &\rightarrow B: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_B} \end{aligned}$$

où $M =$ “je vous aime”.

(Embarrassant, non?)

Chiffrer-puis-signer: attaques [Davis 2001]

Encore une attaque man-in-the-middle:

$$\begin{aligned} A &\rightarrow C: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_C} \\ C &\rightarrow B: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_B} \end{aligned}$$

où $M =$ plans secrets.

(C un traître, B un concurrent)

Chiffrer-puis-signer: attaques [Davis 2001]

Encore une attaque man-in-the-middle:

$$\begin{aligned} A &\rightarrow C: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_C} \\ C &\rightarrow B: \quad \{M, [h(M)]_{K_A^{-1}}\}_{K_B} \end{aligned}$$

où $M =$ “reconnaissance de dette: 1000 euros”.
(puis C se plaindra que A n’honore pas ses dettes.)

Outline

Conclusion

- Notre modèle de sécurité (ce que les différents acteurs peuvent faire) était erroné (si tant est que nous en ayons eu un).
- Utiliser la cryptographie à bon escient pour assurer une propriété de sécurité (confidentialité, authentication, autre) est le sujet des protocoles cryptographiques: voir cours prochain avec Stéphanie Delaune.
- La cryptographie est beaucoup plus riche encore: intégrité et MACs, preuves à connaissance nulle, bit commitment, oblivious transfer, calcul homomorphique [Chambers2009], etc.