Organisation

Quelques mots de présentation

- Maxime Bombar
  maxime.bombar@math.u-bordeaux.fr
- Bureau 382, bâtiment A33 - IMB
- maximebombar.fr

Il n'y a pas de questions stupides !

Organisation du Cours

- CM les Mardis de 14h à 15h20 - A29 / Salle 101 (sauf exception)
- TD/TP de 15h30 à 18h20 - CREMI salle 009 (aujourd'hui 103)
- TP en SageMath: Objectif, amusez vous dans ces TPs.

- Du contrôle continu: 50% de la note
  - Challenges de cryptanalyse: Il est important de pratiquer.
  - Un (petit) DM ou DS à la moitié du semestre.
- Un examen final (3h) en décembre, sur papier: 50% de la note.

Lectures Complémentaires (Librement Accessibles)

- A. Canteaut - Lecture Notes on Cryptographic Boolean Functions
- A. Canteaut - Lecture Notes on ECC and their Applications to Symmetric Crypto
- D. Boneh, V. Shoup - A Graduate Course in Applied Cryptography
- C. Swenson - Modern Cryptanalysis: Techniques for Advanced Code Breaking.
- Des notes sur la théorie de l'information: Par exemple E. Berardini, G. Zémor.
- Le poly du cours jusqu'en 2023-2024: G. Castagnos - Cryptanalyse

Autres Lectures Complémentaires


G. Zémor - Cours de Cryptographie


D. Vergnaud - Exercices et Problèmes de Cryptographie


A. Joux - Algorithmic Cryptanalysis

Cours I: Introduction à la Cryptanalyse

Maxime Bombar

Objectifs du Cours

- Comprendre les principes de **design cryptographiques**
- Culture générale des techniques modernes en cryptanalyse.
- Développer la capacité de lire de véritables articles de recherche
  $\rightarrow$ Visitez `eprint.iacr.org`.
- Mettre en application ces techniques sur de la crypto "de la vraie vie"
  - Dans ce cours (TP)
  - Dans vos stages
  - Dans vos futurs jobs: Recherche ou entreprise (ou les deux, cf Ciffre)
- Mises en garde:
  - N'implémentez pas votre propre crypto vous même (mais cryptanalyse OK).
  - Plus efficace ne signifie pas forcément plus sûr.

Remarques Importantes

- En réalité, la plupart des attaques sont **Impraticables**. Et c'est connu!
- Elles peuvent nécessiter des ressources monstrueuses (temps/mémoire ...)
- Elles peuvent faire des hypothèses sur les conditions de l'attaque ou utiliser des informations extérieures (par exemple side-channel).
- Le but du cryptanalyste est de donner des estimations de la sécurité.
- **Tout le monde** utilise les primitives cryptographiques dans des protocoles plus avancés. On ne peut pas se permettre la moindre faille.

$$\text{Confiance}(T) = \int_{t=0}^{T} \text{cryptanalyse}(t)\, dt$$

Quelques Ordres de Grandeur

On considère aujourd'hui qu'une primitive est sécurisée si les attaques nécessitent au moins 2^{128} opérations. On parle de 128 **bits de sécurité**.

- $2^{10} = 1024 \approx 1000$
- $2^{20} \approx 10^6$
- $2^{30} \approx 10^9$
- 1 CPU laptop $\approx$ 4 GHz $\approx 2^{32}$ opérations par seconde.
- Réseau Bitcoin total: $\approx 2^{60}$ évaluations de SHA-256 par seconde, soit $\approx 2^{85}$ par an.
- Evaluer 2^{128} hash demanderait plus d'énergie que celle pour vaporiser toute l'eau sur Terre.

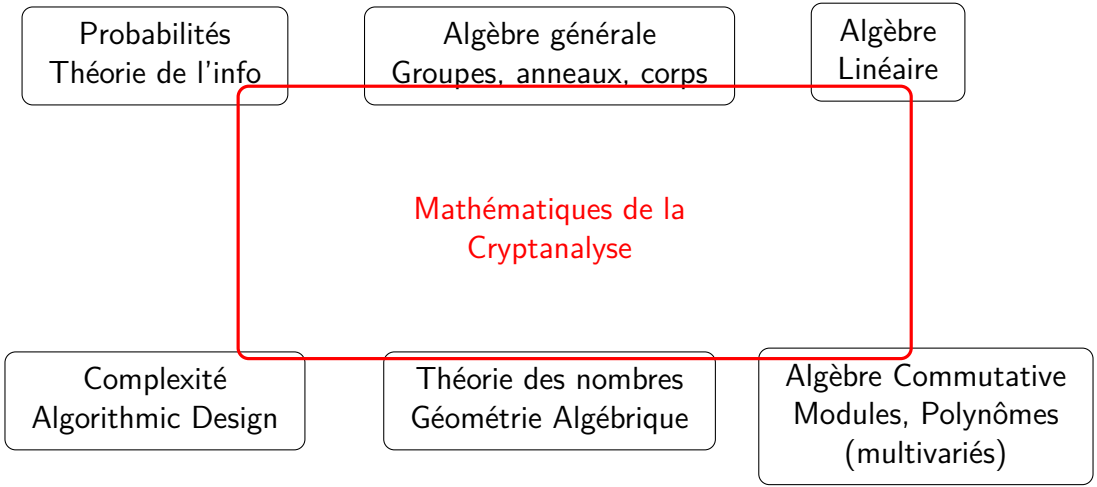
Contenu du cours

- Chiffrement par blocs
- Chiffrement par flots
- Fonctions de hachage

- Cryptanalyse Linéaire
- Cryptanalyse Différentielle
- Cryptanalyse Algébrique

- Méthodes de réduction de réseaux

Outils Mathématiques



Principe de Kerckhoffs

Auguste Kerckhoffs
(1835-1903)

- L'algorithme du cryptosystème ne doit pas être secret.
  $\rightarrow$ Le cryptanalyste connaît l'algorithme.
- Seule la clé doit être secrète.
  $\rightarrow$ La clé détermine une instance particulière du cryptosystème.

Types de Cryptanalyse

Clair Connu
Retrouve la clé à partir de (nombreux) couples *(clair, chiffré)*.

Chiffrés Choisis
L'attaquant a accès à un oracle pour chiffrer des **messages de son choix**.
Permet des attaques **adaptatives**.

Canaux Auxiliaires
Utilise de l'information supplémentaire (consommation énergétique, injection de fautes...)
cf: UE Cartes à Puces

Cryptanalyse quantique
Shor, Grover
cf: UE Algo Arithmétiques

Préliminaires

Un rêve: le chiffrement parfait

Théorie de Shannon

Def: Un système de chiffrement est un triplet de variables aléatoires $(M, K, C) \in \mathcal{M} \times \mathcal{K} \times \mathcal{C}$ tel que

- M et K sont indépendantes
- $H(M|K, C) = 0$ où $H(\cdot|\cdot)$ désigne l'**entropie conditionnelle** est l'incertitude que l'on a sur M connaissant K et C.

Autrement dit, le déchiffrement est toujours unique.

Def: Un chiffrement (M, K, C) est dit **parfait** lorsque $H(M|C) = H(M)$.

Exemple de chiffrement parfait?

Le One-Time-Pad (ou chiffrement de Vernam)

Dans le chifrement One-Time Pad, $\mathcal{M}, \mathcal{K}, \mathcal{C}$ sont identifiés à un même groupe abélien G. Pour une clé $K \in G$, et un message $M \in G$, le chiffré est

$$C = E_K(M) \stackrel{\text{def}}{=} M + K.$$

Prop. Le *One-Time Pad* est un chiffrement parfait.

Théorème de Shannon pour le Chiffrement

Problème: Si (M, K, C) est un système de chiffrement parfait, alors $H(K) \geqslant H(M)$. En pratique, la taille des clés doit être au moins aussi grande que les messages.

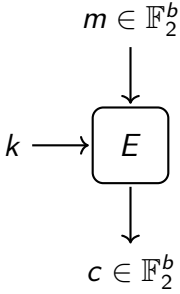
Cryptographie en Pratique

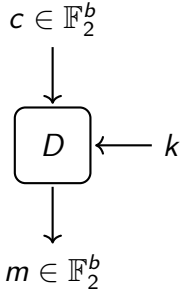
- Un chiffrement parfait donne une garantie de sécurité statistique, et **inconditionnelle**.
- En pratique, un adversaire est **limité en ressources** (temps, mémoire etc...)
- Peut-on produire des chiffrements **efficaces** et **sécurisés** dans cette limitation?

Chiffrement parfait ne veut pas dire résiste à la cryptanalyse. Le OTP est par exemple vulnérable à une attaque à clairs connus.

Rappel: Chiffrement par Blocs

Def: Un chiffrement par blocs de taille de blocs b et de taille de clé n est une famille de 2^n bijections de $E_k : \mathbb{F}_2^b \to \mathbb{F}_2^b$, indexées par un élément $k \in \mathbb{F}_2^n$. On note en général D_k l'application inverse (de déchiffrement) $D_k \stackrel{\text{def}}{=} E_k^{-1}$.





Chiffrements par Blocs (Cont'd)

- Les chiffrements par blocs sont des **primitives fondamentales**. Leurs constructions sont à la base d'autres primitives comme de nombreuses fonctions de hachage par exemple.
- La cryptanalyse de chiffrements par blocs peut s'adapter pour d'autres constructions (par exemple la **cryptanalyse différentielle** dans quelques semaines).

Un chiffrement par blocs seul ne forme pas un réel algorithme de chiffrement sécurisé. Il s'utilise dans un **mode d'opération** (CBC, Authenticated Encryption, Hachage...)

La sécurité d'un chiffrement par blocs dépend du mode dans lequel il est utilisé.

Cryptanalyse Chiffrement par Blocs

- **Retrouver la clé**: Étant donnés une liste de couples clair-chiffrés (P_i, C_i), trouver la/les clés K telles que $C_i = E_K(P_i)$ pour tout i.
- **Distingueur**: En général, on modélise un chiffrement par blocs sécurisé comme une permutation aléatoire de $\mathbb{F}_2^b$. Une attaque par distingueur consiste à invalider cette propriété.
- **Information partielle**: Peut-on prédire quelques bits du clair à partir du chiffré?

Les attaques par recouvrement de clé reposent souvent sur un distingueur sur moins de tours: Si la clé est mauvaise, E_K doit se comporter comme une permutation aléatoire, alors que si K est la bonne, ce n'est plus le cas.

Plus précis plus tard dans le cours.

Grands Principes de Sécurité (Shannon)

- **Confusion:** Il ne doit pas y avoir de lien évident entre les bits de M, C et K.
- **Diffusion:** Un changement de 1 bit de M ou de K doit provoquer un changement radical de C.

OTP ne respecte pas ces principes.

Pour des raisons d'efficacité, un chiffrement par blocs est en général la composition de nombreuses fonctions (dites **fonctions de tour**) opérant sur des espaces beaucoup plus petits mais qui vont chacune assurer ces deux propriétés. En général, les couches **linéaires** (*e.g.* rotations de bits) participent à la diffusion, et les couches **non-linéaires** (*e.g.* S-Box) apportent la confusion.

Grandes Familles de Chiffrements par Blocs

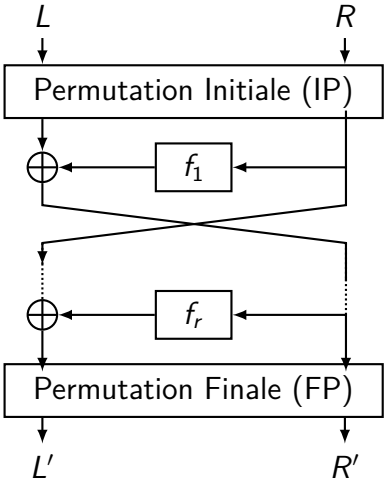
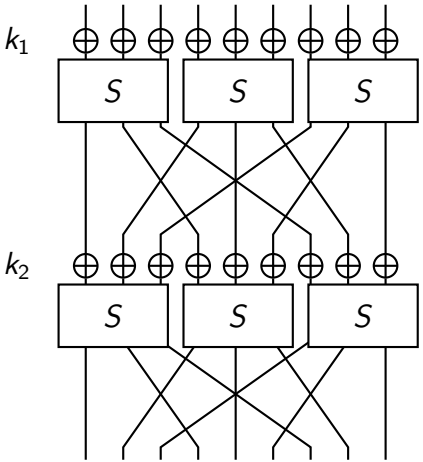


Schéma de Feistel (ex: DES)



Réseau SPN (ex: AES)

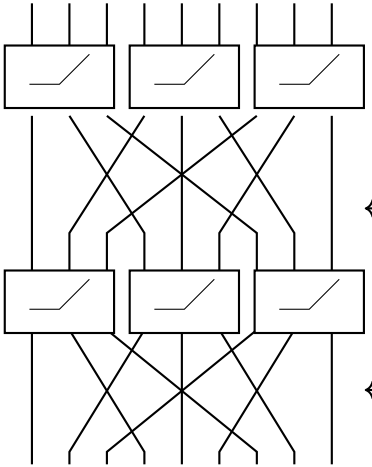
Applications de la Cryptanalyse ..... au-delà de la Crypto?

Culture Générale Scientifique

Un chiffrement par blocs de type SPN (Réseau de Substitution Permutation) est une suite de couches linéaires globales (opérant sur tous les bits) et non linéaires locales (opérant sur quelques bits seulement).

Connaissez-vous autre chose qui utilise le même principe?

Applications de la Cryptanalyse ..... au-delà de la Crypto?



$\longleftarrow$ $\mathbf{W_1}$: Poids (réels/floats)

$\longleftarrow$ $\mathbf{W_2}$: Poids (réels/floats)

Réseaux de Neurones!

Le problème de retrouver les poids d'un DNN est similaire à celui de retrouver la clé d'un SPN!

Polynomial Time Cryptanalytic Extraction of Deep Neural Networks in the Hard-Label Setting

Nicholas Carlini[1], Jorge Chávez-Saab[2], Anna Hambitzer[2], Francisco Rodríguez-Henríquez[2], and Adi Shamir ✉[3]

[1] Google DeepMind nicholas@carlini.com
[2] Cryptography Research Center, Technology Innovation Institute
{jorge.saab,anna.hambitzer,francisco.rodriguez}@tii.ae
[3] Weizmann Institute
adi.shamir@weizmann.ac.il

Techniques Génériques de Cryptanalyse

Attaque Générique: Force Brute

- Attaque à clairs connus $(P_1, C_1), \ldots, (P_r, C_r)$
- Itère sur toutes les clés jusqu'à avoir $C_1 = E_K(P_1)$.
- Attention aux faux positifs!! Utiliser les autres clairs-chiffrés.

Soit $(P, C) \in \mathbb{F}_2^b \times \mathbb{F}_2^b$ tel que $C = E_{K^\star}(P)$ pour une certaine clé $K^\star \in \mathbb{F}_2^n$. Combien d'autres clés K existe-t-il telles que $C = E_K(P)$?

Faux Positifs

Hypothèse: On modélise un chiffrement par blocs E_K comme une permutation aléatoire de $\mathbb{F}_2^b$.

On a une liste de couples (P, C) avec $C = E_{K^\star}(C)$ et $K \leftarrow \mathbb{F}_2^n$.

- On déchiffre C avec une clé K pour obtenir $P' = D_K(C)$. Si K n'est pas la bonne clé, alors P' est uniforme dans $\mathbb{F}_2^b$ et donc la probabilité que K passe le test est $\mathbb{P}(P' = P) = \frac{1}{2^b}$.
- Avec r paires indépendantes, la probabilité qu'une mauvaise clé passe tous les tests tombe donc à 2^{-rb}.

Voir TD1.

Nombre Moyen de Tests

On note T la variable aléatoire qui définit le nombre d'essais de clés K avant de trouver la bonne clé $K^\star$. On tire les clés uniformément sans remise: autrement dit on tiré un ordre uniformément sur toutes les clés.

La probabilité que la bonne clé soit la k-ème est uniforme dans $\mathbb{F}_2^n$. Si T désigne le nombre de tirages avant d'obtenir $K^\star$, on a donc que T est uniformément distribuée dans $\{1, \ldots, 2^n\}$

En moyenne, on doit donc faire $\mathbb{E}(T)$ tirages, soit

$$\mathbb{E}(T) = \sum_{k=1}^{2^n} k \cdot \mathbb{P}(T = k) = 2^{-n} \sum_{k=1}^{2^n} k = 2^{-n} \frac{2^n(2^n+1)}{2} = \frac{2^n+1}{2} \approx 2^{n-1}.$$

Remarques sur la Force Brute

- Pas si facile que vous ne l'imaginez en pratique.
- Les optimisations pour la force brute ne sont pas les mêmes que pour le chiffrement!
- Parfois, une dose de force brute au sein d'une attaque peut donner de bons résultats!

Attaque par Dictionnaire

- Une attaque par dictionnaire est une forme de force brute, où l'on va tester les clés dans une liste de clés plus probables d'abord.
- Utiles pour les recherche de mots de passe!

En 2009, une attaque sur les serveurs de l'entreprise RockYou a permis l'exposition d'une liste de plusieurs dizaines de millions de mots de passe en clair. Cette liste est toujours disponible facilement sur Internet sous le nom *rockyou.txt*, et est très utilisée pour tester la solidité de réseaux et systèmes.

Meet in the Middle

Compromis Temps-Mémoire

Parfois, pour améliorer les recherches par force brute, on va garder en mémoire un bout de la recherche pour accélérer les calculs.

- Pour un chiffrement sur $2r$ tours, au lieu de tester $C = E_{2r} \circ \cdots \circ E_1(P)$, on peut tester si $D_{2r} \circ \cdots \circ D_{r+1}(C) = E_r \circ \cdots \circ E_1(P)$.
- Attention aux faux positifs!
- Analyse en TD.

Paradigme utile pour monter des attaques parfois plus efficaces.