Notions de complexité informatique

Quentin Deschamps

Printemps 2021 *M1, ISFA*

Modèle cryptographique

Deux entités (Alice et Bob) souhaitent communiquer dans un environnement non sécurisé.

Hypothèse fondamentale

- Alice et Bob ne se sont à priori jamais rencontré de manière sécurisée, on ne peut pas supposer un quelconque accord secret entre eux.

Hypothèse fondamentale

- Alice et Bob ne se sont à priori jamais rencontré de manière sécurisée, on ne peut pas supposer un quelconque accord secret entre eux.
- En particulier la méthode utilisée pour communiquer est connue de l'attaquant.

Exemple : le masque jetable (One time-pad)

Definition

Soit $n \in \mathbb{N}$. On utilise les espaces $K = P = C = \{0,1\}^n$ pour définir le schéma d'encryption.

- $\mathsf{KeyGen}(1^\lambda) \xleftarrow{\$} k \in K$
- $\mathsf{Enc}(k, m) = k \otimes m$
- $\mathsf{Dec}(k, c) = k \otimes c$

Exemple : le masque jetable (One time-pad)

Definition

Soit $n \in \mathbb{N}$. On utilise les espaces $K = P = C = \{0,1\}^n$ pour définir le schéma d'encryption.

- $\mathsf{KeyGen}(1^\lambda) \xleftarrow{\$} k \in K$
- $\mathsf{Enc}(k, m) = k \otimes m$
- $\mathsf{Dec}(k, c) = k \otimes c$

Il s'agit bien d'un schéma d'encryption car il vérifie la propriété

$$\forall k \in K, \forall m \in P, \mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$$

Avantages et inconvénients

Avantages

- Permet une sécurité parfaite (ou sécurité sémantique) : impossible à casser.

Avantages et inconvénients

Avantages

- Permet une sécurité parfaite (ou sécurité sémantique) : impossible à casser.
- Enc et Dec sont très rapides à mettre en oeuvre.

Avantages et inconvénients

Avantages

- Permet une sécurité parfaite (ou sécurité sémantique) : impossible à casser.
- Enc et Dec sont très rapides à mettre en oeuvre.

Inconvénients

- La clef doit être aussi longue que le message : ralentit énormément le processus.

Avantages et inconvénients

Avantages

- Permet une sécurité parfaite (ou sécurité sémantique) : impossible à casser.
- Enc et Dec sont très rapides à mettre en oeuvre.

Inconvénients

- La clef doit être aussi longue que le message : ralentit énormément le processus.
- Chaque clef ne peut être utilisée qu'une seule fois.

Avantages et inconvénients

Avantages

- Permet une sécurité parfaite (ou sécurité sémantique) : impossible à casser.
- Enc et Dec sont très rapides à mettre en oeuvre.

Inconvénients

- La clef doit être aussi longue que le message : ralentit énormément le processus.
- Chaque clef ne peut être utilisée qu'une seule fois.
- Pour avoir une sécurité parfaite la clef doit être parfaitement aléatoire.

Nouveau paradigme

Le masque jetable est le seul système permettant d'avoir une sécurité parfaite, pour avoir des systèmes efficaces en pratique il faut relâcher la contrainte de sécurité. On va passer à une définition plus concrète de la sécurité.

Définition : Sécurité calculatoire

On veut qu'il soit impossible de casser le code en un temps raisonnable.

Nouveau paradigme

Le masque jetable est le seul système permettant d'avoir une sécurité parfaite, pour avoir des systèmes efficaces en pratique il faut relâcher la contrainte de sécurité. On va passer à une définition plus concrète de la sécurité.

Définition : Sécurité calculatoire

On veut qu'il soit impossible de casser le code en un temps raisonnable.

On va essayer de formaliser cette définition...

Notation de Landau

Définition : Grand O

Soient f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{N}$. On dit que $f = O(g)$ si il existe une constante $c \in \mathcal{R}$ et un entier $N \in \mathbb{N}$ tel que $\forall n \geq N, f(n) \leq cg(n)$.

On dit que la fonction g domine la fonction f où que f est un grand O de g.

Notation de Landau

Définition : Grand O

Soient f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{N}$. On dit que $f = O(g)$ si il existe une constante $c \in \mathcal{R}$ et un entier $N \in \mathbb{N}$ tel que $\forall n \geq N, f(n) \leq cg(n)$.

On dit que la fonction g domine la fonction f où que f est un grand O de g.

Définition : Petit o

Soient f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{N}$. On dit que $f = o(g)$ si f est négligeable devant g (au sens mathématique) : $\forall \epsilon > 0, \exists N \in \mathbb{N}, f(n) \leq \epsilon g(n)$

Complexité d'un algorithme

Définition : Complexité pire cas

Soit $\mathcal{A}$ un algorithme résolvant un problème $\mathcal{P}$. La complexité de $\mathcal{A}$ est une fonction $c : \mathbb{N} \to \mathbb{N}$ tel que $c(n)$ est le nombre maximal d'opération qu'effectue $\mathcal{A}$ sur une entrée de n bits.

Complexité d'un algorithme

Définition : Complexité pire cas

Soit $\mathcal{A}$ un algorithme résolvant un problème $\mathcal{P}$. La complexité de $\mathcal{A}$ est une fonction $c : \mathbb{N} \to \mathbb{N}$ tel que $c(n)$ est le nombre maximal d'opération qu'effectue $\mathcal{A}$ sur une entrée de n bits.

Remarque

On peut définir de manière similaire la complexité moyenne et la complexité meilleure cas, cette dernière étant rarement pertinente.

Opération élémentaire

- Affectation
- Addition et multiplication
- Comparaison
- Modulo
- Test
- ...

Exemple

Algorithme 1 : Moyenne des nombres impairs

Input : L une liste de n entiers

```
1 M ← 0
2 nb − impair = 0
3 for (i = 0 ; i < n; i++) do
4     if L[i] mod 2 = 1 then
5         M=M+L[i]
6         nb-impair++
```

Output : M/nb-impair

Exemple

Algorithme 2 : Moyenne des nombres impairs

Input : L une liste de n entiers

```
1 M ← 0
2 nb − impair = 0
3 for (i = 0 ; i < n; i++) do
4     if L[i] mod 2 = 1 then
5         M=M+L[i]
6         nb-impair++
```

Output : M/nb-impair

$3n + 3$ opérations élémentaires.

Des torchons et des serviettes

- Selon les machines les opérations élémentaires n'ont pas la même vitesse. De plus pour un algorithme complexe il est illusoire d'espérer calculer précisément le nombre d'opérations.

Des torchons et des serviettes

- Selon les machines les opérations élémentaires n'ont pas la même vitesse. De plus pour un algorithme complexe il est illusoire d'espérer calculer précisément le nombre d'opérations.
- On va seulement calculer un ordre de grandeur du nombre d'opérations.

Des torchons et des serviettes

- Selon les machines les opérations élémentaires n'ont pas la même vitesse. De plus pour un algorithme complexe il est illusoire d'espérer calculer précisément le nombre d'opérations.
- On va seulement calculer un ordre de grandeur du nombre d'opérations.
- Sur l'exemple complexité en $O(n)$.

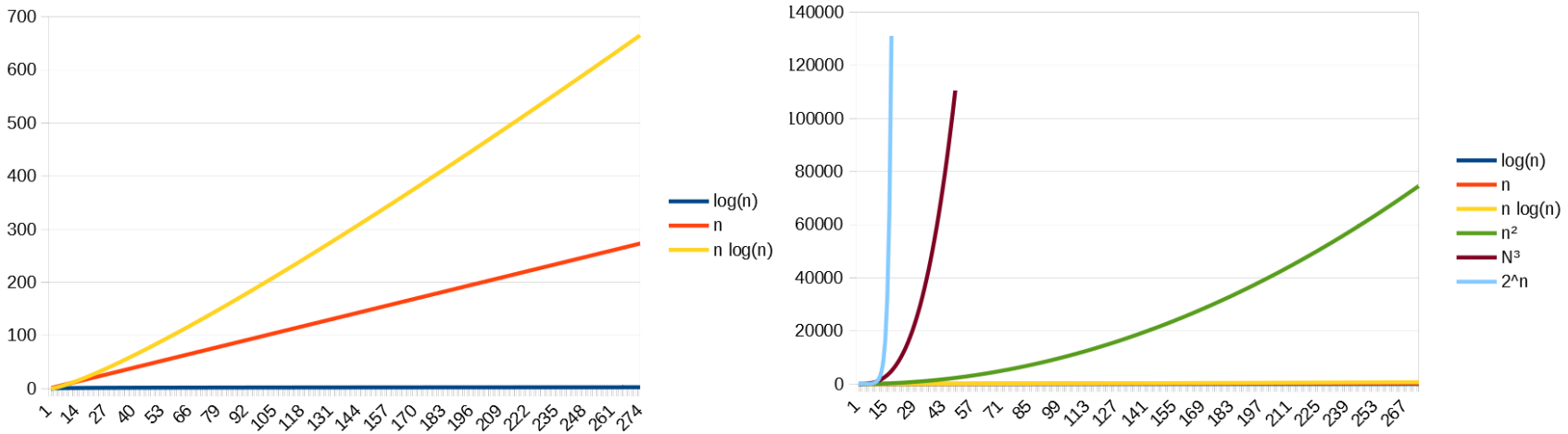
Signification

Remarque

La complexité représente à quel point l'information que l'on cherche est difficile à extraire des données brutes.

Signification

La complexité d'un algorithme est une donnée relative, elle mesure comment le temps de calcul évolue avec l'augmentation de la taille des données.



Ordres de grandeur

- Fréquence d'un ordinateur : quelques Ghz (10^9 Hz)
- Centre de calcul de l'université : 1500 processeurs.

n	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n^3)$	$O(2^n)$
10	$<$ 1ns	$<$ 1ns	$<$ 1ns	$<$ 1ns	$<$ 1ns
20	$<$ 1ns	$<$ 1ns	$<$ 1ns	$<$ 1ns	$<$ 1000ns
50	$<$ 1ns	$<$ 1ns	$<$ 1ns	$<$ 1ns	1 minute
100	$<$ 1ns	$<$ 1ns	$<$ 10ns	$<$ 1000ns	10^9 années
10^6	$<$ 1ms	$<$ 1ms	0,1s	18h	$+\infty$
10^7	$<$ 1ms	$<$ 1ms	6s	2 ans	$+\infty$
10^8	$<$ 1ms	$<$ 1ms	11min	2100 ans	$+\infty$
10^9	$<$ 1ms	1ms	18h	10^6 années	$+\infty$

Sécurité

Définition : Sécurité calculatoire

Un protocole est calculatoirement sécurisé si il est impossible de le casser en un temps de calcul polynomial.

Sécurité

Définition : Sécurité calculatoire

Un protocole est calculatoirement sécurisé si il est impossible de le casser en un temps de calcul polynomial.

Remarque

Comment faire pour montrer ce genre d'impossibilité ?

Problème NP-difficile

- On définit des problèmes de difficulté équivalente.

Problème NP-difficile

- On définit des problèmes de difficulté équivalente.

Définition : Problème NP-difficile

Un problème est NP-difficile si il peut être réduit polynomialement au problème SAT.

Problème NP-difficile

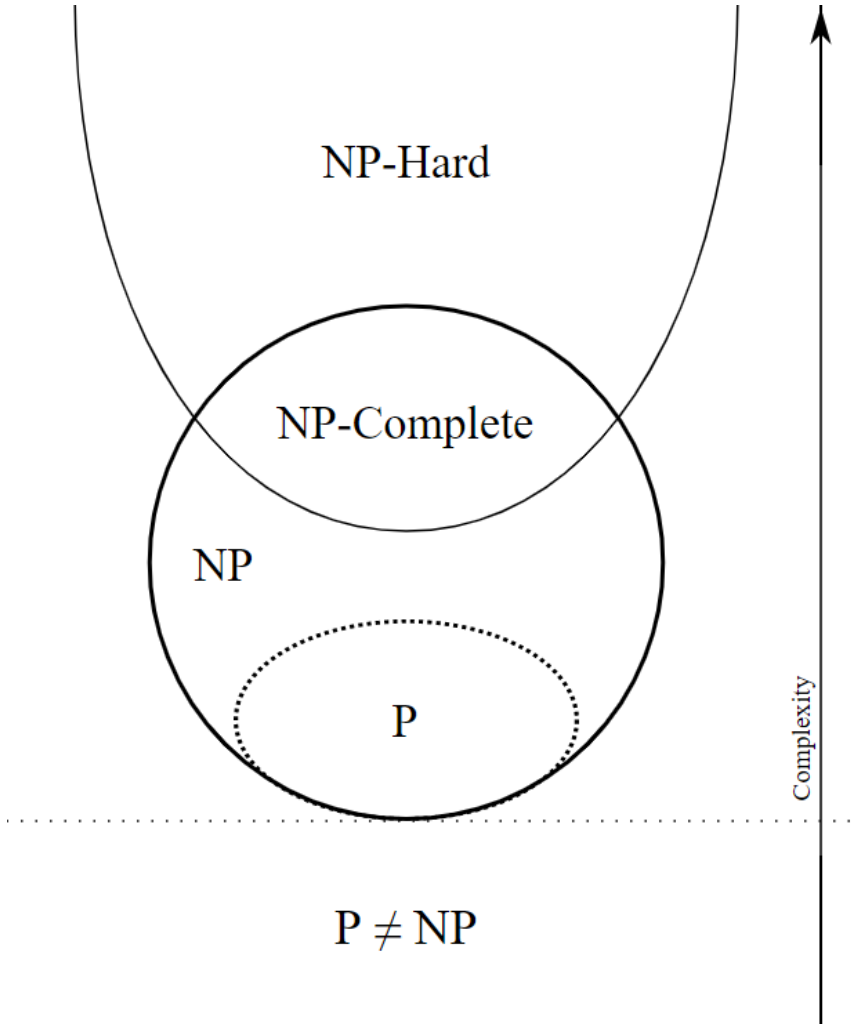
- On définit des problèmes de difficulté équivalente.

Définition : Problème NP-difficile

Un problème est NP-difficile si il peut être réduit polynomialement au problème SAT.

- L'idée est simplement d'avoir un ensemble de problème de difficulté équivalente.

En dessin



Postulat

On considère qu'il existe des problèmes dont la résolution ne peut se faire qu'en temps exponentiel.

Postulat

On considère qu'il existe des problèmes dont la résolution ne peut se faire qu'en temps exponentiel.

Exemple : RSA

Soient p et q deux "grands" nombres premiers.
Connaissant $n = p * q$ retrouver la factorisation de n.