# Sécurisation des communications

M. Tellene

# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

1. L'URL du site est décodé par le navigateur

# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

1. L'URL du site est décodé par le navigateur
2. Ce dernier isole le protocole (`HTTP`), le nom de domaine (`github.com`) et le chemin vers la ressource (`/mtellene`)

# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

1. L'URL du site est décodé par le navigateur
2. Ce dernier isole le protocole (`HTTP`), le nom de domaine (`github.com`) et le chemin vers la ressource (`/mtellene`)
3. Le navigateur effectue une résolution de nom pour déterminer l'adresse IP correspondant au nom de domaine (`140.82.114.4`)

# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

1. L'URL du site est décodé par le navigateur
2. Ce dernier isole le protocole (`HTTP`), le nom de domaine (`github.com`) et le chemin vers la ressource (`/mtellene`)
3. Le navigateur effectue une résolution de nom pour déterminer l'adresse IP correspondant au nom de domaine (`140.82.114.4`)
4. Le navigateur peut alors établir une connexion TCP vers l'adresse IP du serveur web, sur le port 80

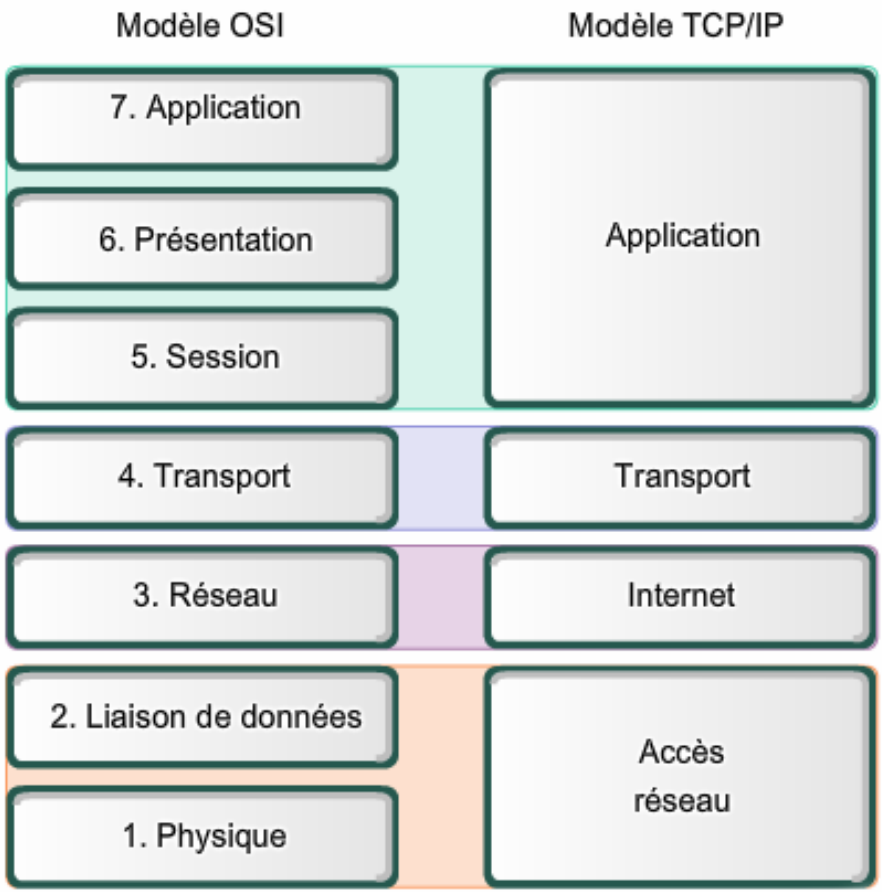
# Sécurisation des communications

Nous pouvons, depuis la classe de première, comprendre ce qu'il se passe lorsque l'on navigue vers un site web. Prenons par exemple : `https://github.com/mtellene`

1. L'URL du site est décodé par le navigateur
2. Ce dernier isole le protocole (`HTTP`), le nom de domaine (`github.com`) et le chemin vers la ressource (`/mtellene`)
3. Le navigateur effectue une résolution de nom pour déterminer l'adresse IP correspondant au nom de domaine (`140.82.114.4`)
4. Le navigateur peut alors établir une connexion TCP vers l'adresse IP du serveur web, sur le port 80
5. Une fois la connexion établie, client et serveur échangent des données en utilisant le protocole HTTP

# Sécurisation des communications

On se souvient que les communications sur Internet utilisent un ensemble de protocoles, organisés en couches



# Sécurisation des communications

1. Couche matérielle (ou réseau) avec des protocoles tels que Ethernet

# Sécurisation des communications

1. Couche matérielle (ou réseau) avec des protocoles tels que Ethernet
2. Couche Internet avec le protocole IP, permettant de définir des routes

# Sécurisation des communications

1. Couche matérielle (ou réseau) avec des protocoles tels que Ethernet
2. Couche Internet avec le protocole IP, permettant de définir des routes
3. Couche transport avec les protocoles UDP ou TCP, qui s'occupent en particulier de garantir l'intégrité des données transmises (garanties minimales pour UDP et très fortes pour TCP)

# Sécurisation des communications

1. Couche matérielle (ou réseau) avec des protocoles tels que Ethernet
2. Couche Internet avec le protocole IP, permettant de définir des routes
3. Couche transport avec les protocoles UDP ou TCP, qui s'occupent en particulier de garantir l'intégrité des données transmises (garanties minimales pour UDP et très fortes pour TCP)
4. Couche d'application dans laquelle se trouvent les protocoles de haut niveau comme HTTP

# Sécurisation des communications

Ce processus a été très peu modifié depuis la conception de TCP/IP à la fin des années 1970

Cependant, avec la démocratisation d'Internet, du Web et la diversification des usages, des problèmes sont apparus

# Sécurisation des communications

Chaque routeur peut donc **inspecter les paquets pour en connaître le contenu**.

Cette situation n'est clairement pas idéale. En effet, si l'on utilise le site web pour effectuer des transactions bancaires, renseigner des informations personnelles, ou simplement exprimer son opinion. On souhaite que le contenu des messages envoyés ne soit connu que de deux entités : **la source et la destination**

# Sécurisation des communications

Ce simple constat nous permet de mettre en avant trois aspects liés à la sécurisation des communications :

- Comment chiffrer le contenu des communications, afin qu'elles ne soient lisible que par la source et la destination ?

# Sécurisation des communications

Ce simple constat nous permet de mettre en avant trois aspects liés à la sécurisation des communications :

- Comment chiffrer le contenu des communications, afin qu'elles ne soient lisible que par la source et la destination ?
- Comment garantir que le serveur auquel on se connecte est bien celui de la personne et/ou de l'entité auquel on pense se connecter ?

# Sécurisation des communications

Ce simple constat nous permet de mettre en avant trois aspects liés à la sécurisation des communications :

- Comment chiffrer le contenu des communications, afin qu'elles ne soient lisible que par la source et la destination ?
- Comment garantir que le serveur auquel on se connecte est bien celui de la personne et/ou de l'entité auquel on pense se connecter ?
- Comment garantir les deux propriétés ci-dessus en réutilisant l'infrastructure d'Internet, à savoir les communications TCP/IP ?

# Sécurisation des communications

Avant de commencer, nous allons fixer quelques termes :

- **Coder (ou encoder)** : représenter de l'information par un ensemble de signes prédéfinis

# Sécurisation des communications

Avant de commencer, nous allons fixer quelques termes :

- **Coder (ou encoder)** : représenter de l'information par un ensemble de signes prédéfinis
- **Décoder** : interpréter un ensemble de signes pour extraire l'information qu'ils représentent

# Sécurisation des communications

Avant de commencer, nous allons fixer quelques termes :

- **Coder (ou encoder)** : représenter de l'information par un ensemble de signes prédéfinis
- **Décoder** : interpréter un ensemble de signes pour extraire l'information qu'ils représentent
- **Chiffrer** : rendre une suite de symboles incompréhensible au moyen d'une clé de chiffrement

# Sécurisation des communications

Avant de commencer, nous allons fixer quelques termes :

- **Coder (ou encoder)** : représenter de l'information par un ensemble de signes prédéfinis
- **Décoder** : interpréter un ensemble de signes pour extraire l'information qu'ils représentent
- **Chiffrer** : rendre une suite de symboles incompréhensible au moyen d'une clé de chiffrement
- **Déchiffrer (ou décrypter)** : retrouver la suite de symboles originale à partir du message chiffré. On utilise le terme déchiffrer lorsque l'on utilise une clé de déchiffrement pour récupérer le texte initial et le terme décrypter lorsque l'on arrive à déterminer le message original sans utiliser la clé

# Sécurisation des communications

Une première technique lorsque l'on souhaite chiffrer un message est d'utiliser une méthode de **chiffrement symétrique**.

Une telle méthode est donnée par deux fonctions :

- `c(m,k)` est la fonction de chiffrement.
  - Input : un message en clair `m` (`str`) et une clé de chiffrement `k` (`str`, `int`, ...)
  - Output : une chaîne de caractères chiffrée `s`

# Sécurisation des communications

Une première technique lorsque l'on souhaite chiffrer un message est d'utiliser une méthode de **chiffrement symétrique**.

Une telle méthode est donnée par deux fonctions :

- `c(m,k)` est la fonction de chiffrement.
  - Input : un message en clair `m` (`str`) et une clé de chiffrement `k` (`str`, `int`, ...)
  - Output : une chaîne de caractères chiffrée `s`
- `d(s,k)` est la fonction de déchiffrement.
  - Input : un message chiffré `s` (`str`) et une clé de chiffrement `k` (`str`, `int`, ...)
  - Output : le message en clair `m`

# Sécurisation des communications

Le terme symétrique vient du fait que la même clé est utilisée pour chiffrer et déchiffrer le message

# Sécurisation des communications

Un exemple simple de méthode de chiffrement symétrique est le chiffrement symétrique par décalage, appelé codage César. La méthode consiste à choisir un entier $n$ et à décaler chaque lettre du message initial de $n$ lettres dans l'alphabet

# Sécurisation des communications

Un exemple simple de méthode de chiffrement symétrique est le chiffrement symétrique par décalage, appelé codage César. La méthode consiste à choisir un entier $n$ et à décaler chaque lettre du message initial de $n$ lettres dans l'alphabet

Exemple pour $n = 5$

```
L INFORMATIQUE C EST SUPER
```

# Sécurisation des communications

Un exemple simple de méthode de chiffrement symétrique est le chiffrement symétrique par décalage, appelé codage César. La méthode consiste à choisir un entier $n$ et à décaler chaque lettre du message initial de $n$ lettres dans l'alphabet

Exemple pour $n = 5$

```
L INFORMATIQUE C EST SUPER
Q NSKTWRFYNVZJ H JXY XZUJW
```

# Sécurisation des communications

Un exemple simple de méthode de chiffrement symétrique est le chiffrement symétrique par décalage, appelé codage César. La méthode consiste à choisir un entier $n$ et à décaler chaque lettre du message initial de $n$ lettres dans l'alphabet

Exemple pour $n = 5$

```
L INFORMATIQUE C EST SUPER
```

```
Q NSKTWRFYNVZJ H JXY XZUJW
```

La fonction de déchiffrement consiste à prendre un message chiffré et à décaler chaque lettre de $n$ positives vers l'arrière

# Sécurisation des communications

Une méthode de chiffrement un peu moins naïve est le déchiffrement par `XOR`. Celle-ci repose sur l'utilisation de l'opérateur binaire $\oplus$

# Sécurisation des communications

Une méthode de chiffrement un peu moins naïve est le déchiffrement par `XOR`. Celle-ci repose sur l'utilisation de l'opérateur binaire $\oplus$

Étant donné un message `m` et une clé de chiffrement `k`, on recopie plusieurs fois la clé de façon à obtenir une chaîne de même longueur que le message

# Sécurisation des communications

Exemple pour `m = L'INFORMATIQUE C'EST SUPER` et `k = NSI`

```
L'INFORMATIQUE C'EST SUPER
NSINSINSINSINSINSINSINSINS
```

1. fonction `ord()` en Python

# Sécurisation des communications

Exemple pour `m = L'INFORMATIQUE C'EST SUPER` et `k = NSI`

```
L'INFORMATIQUE C'EST SUPER
NSINSINSINSINSINSINSINSINS
```

Chaque caractère du message et de la clé augmentée est ensuite converti en nombre, par exemple en son code Unicode [1]

```
76 39 73 78 70 79 82 77 65 84 73 81 85...
78 83 73 78 83 73 78 83 73 78 83 73 78...
```

1. fonction `ord()` en Python

# Sécurisation des communications

Exemple pour `m = L'INFORMATIQUE C'EST SUPER` et `k = NSI`

```
L'INFORMATIQUE C'EST SUPER
NSINSINSINSINSINSINSINSINS
```

Chaque caractère du message et de la clé augmentée est ensuite converti en nombre, par exemple en son code Unicode [1]

```
76 39 73 78 70 79 82 77 65 84 73 81 85...
78 83 73 78 83 73 78 83 73 78 83 73 78...
```

On applique ensuite un `xor` entre chaque nombre du message et de la clé et on fait les opérations dans l'ordre inverse

1. fonction `ord()` en Python

# Sécurisation des communications

Une propriété intéressante du `xor` : il est réversible

Si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$

# Sécurisation des communications

Une propriété intéressante du `xor` : il est réversible

Si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$

Quel problème apparaît ?

# Sécurisation des communications

Une propriété intéressante du `xor` : il est réversible

Si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$

Quel problème apparaît ?

Le message chiffré peut être déchiffré en réexécutant l'opérateur $\oplus$ avec la clé étendue

# Sécurisation des communications

Une propriété intéressante du `xor` : il est réversible

Si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$

Quel problème apparaît ?

Le message chiffré peut être déchiffré en réexécutant l'opérateur $\oplus$ avec la clé étendue

`xor` est l'une des opérations de base implémentée par le matériel dans l'ALU

`xor` est une brique de base couramment utilisée dans les algorithmes de chiffrement modernes

# Sécurisation des communications

Une propriété intéressante du `xor` : il est réversible

Si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$

Quel problème apparaît ?

Le message chiffré peut être déchiffré en réexécutant l'opérateur $\oplus$ avec la clé étendue

`xor` est l'une des opérations de base implémentée par le matériel dans l'ALU

`xor` est une brique de base couramment utilisée dans les algorithmes de chiffrement modernes

⚠si la méthode de chiffrement est connue, alors une clé trop courte peut compromettre la sécurité

# Sécurisation des communications

Parmi les algorithmes de chiffrement symétriques les plus utilisés, on peut citer :

- `AES` (*Advanced Encryption Standard*)
- `ChaCha20`

# Sécurisation des communications

Parmi les algorithmes de chiffrement symétriques les plus utilisés, on peut citer :

- `AES` (*Advanced Encryption Standard*)
- `ChaCha20`

Bien que ces algorithmes soient plus complexes, ils reposent sur des principes similaires au chiffrement `xor` :

# Sécurisation des communications

Parmi les algorithmes de chiffrement symétriques les plus utilisés, on peut citer :

- `AES` (*Advanced Encryption Standard*)
- `ChaCha20`

Bien que ces algorithmes soient plus complexes, ils reposent sur des principes similaires au chiffrement `xor` :

- une clé initiale est étendue
- la clé et le message sont mélangés, de façon réversible

# Sécurisation des communications

Malgré les nombreux avantages du chiffrement symétrique, ce dernier possède un défaut important : si deux personnes veulent établir un canal de communication sûr, les deux personnes doivent d'abord se mettre d'accord sur la clé à utiliser

# Sécurisation des communications

Malgré les nombreux avantages du chiffrement symétrique, ce dernier possède un défaut important : si deux personnes veulent établir un canal de communication sûr, les deux personnes doivent d'abord se mettre d'accord sur la clé à utiliser

Or, les deux participants sont dans la situation où ils ne peuvent pas communiquer de façon sûre, ils donc avoir un problème pour se mettre d'accord sur la clé

# Sécurisation des communications

- soit ils échangent la clé par un moyen de communication non sûr (mail par exemple) mais dans ce cas, un attaquant pourrait s'emparer de la clé

- soit ils échangent la clé par un moyen « non pratique », par exemple en stockant une clé générée sur un support de stockage et en se rencontrant physiquement pour se donner la clé

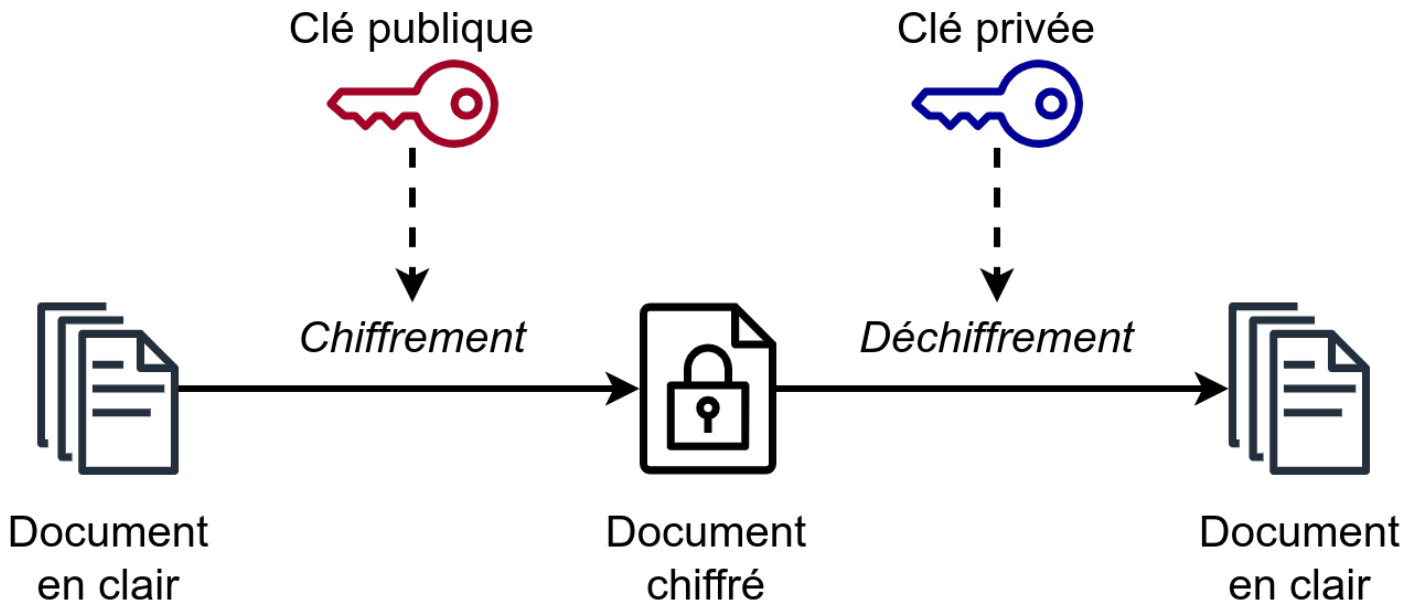
# Sécurisation des communications

Pour résoudre ce problème, diverses techniques ont été développées dès les années 1970, notamment par les services secrets puis par les académies publiques

Ces techniques reposent sur la **cryptographie asymétrique**

# Sécurisation des communications

La cryptographie asymétrique, ou cryptographie à clé publique peut être illustrée de manière suivante :




Un exemple de système de chiffrement asymétrique connu est RSA [2]

2. Pour Rivest, Shamir, Adleman, les inventeurs de RSA

# Sécurisation des communications

La sécurité d'un cryptosystème asymétrique repose sur :

- la difficulté de déduire un texte clair à partir du texte chiffré (et inversement)
- la difficulté de déduire la clé privée à partir de la clé publique

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

Principe :

- Choisir deux très grands nombres premiers ($p$ et $q$)

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

Principe :

- Choisir deux très grands nombres premiers ($p$ et $q$)
- Les multiplier

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

Principe :

- Choisir deux très grands nombres premiers ($p$ et $q$)
- Les multiplier
- En déduire trois nombres $e$, $d$, $n$

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

Principe :

- Choisir deux très grands nombres premiers ($p$ et $q$)
- Les multiplier
- En déduire trois nombres $e$, $d$, $n$
- La clé publique est le couple $Kp = (e, n)$
- La clé privée est le couple $Kpr = (d, n)$

# Sécurisation des communications

Dans cette partie, nous nous intéresserons à `RSA`

Cryptosystème à génération de clés : déduire la clé privée à partir de la clé publique doit être difficile

Principe :

- Choisir deux très grands nombres premiers ($p$ et $q$)
- Les multiplier
- En déduire trois nombres $e$, $d$, $n$
- La clé publique est le couple $Kp = (e, n)$
- La clé privée est le couple $Kpr = (d, n)$

Pour inverser le processus et déduire la clé privée, il faut factoriser le nombre $n$

# Sécurisation des communications

Chiffrement/déchiffrement : déduire le texte clair à partir du texte chiffré doit être très difficile

3. Ce problème, appelé Racine $n^{ime}$ modulo un entier composite est très difficile à résoudre

# Sécurisation des communications

Chiffrement/déchiffrement : déduire le texte clair à partir du texte chiffré doit être très difficile

Principe :

- chiffrer : $C = E_{Kp}(M) = M^e \mod n$
- déchiffrer : $M = D_{Kpr}(C) = C^d \mod n$

3. Ce problème, appelé Racine $n^{ime}$ modulo un entier composite est très difficile à résoudre

# Sécurisation des communications

Chiffrement/déchiffrement : déduire le texte clair à partir du texte chiffré doit être très difficile

Principe :

- chiffrer : $C = E_{Kp}(M) = M^e \mod n$
- déchiffrer : $M = D_{Kpr}(C) = C^d \mod n$

Pour inverser le processus et décrypter le texte chiffre, il faut, connaissant $C$, $e$, $n$, trouver $M$, qui est la racine $e^{ime}$ de $C$ modulo $n$ [3]

3. Ce problème, appelé Racine $n^{ime}$ modulo un entier composite est très difficile à résoudre

# Sécurisation des communications

Le cryptosystème RSA utilise les outils mathématiques suivants :

- Génération de très grands nombres premiers (pour que la clé soit difficile à trouver)
- Arithmétique modulaire
- Exponentiation modulaire
- Inversion modulaire
- Totient d'Euler et (un des) théorème

# Sécurisation des communications

Le cryptosystème RSA utilise les outils mathématiques suivants :

- Génération de très grands nombres premiers (pour que la clé soit difficile à trouver)
- Arithmétique modulaire
- Exponentiation modulaire
- Inversion modulaire
- Totient d'Euler et (un des) théorème

Nous allons grandement simplifier RSA

# Sécurisation des communications

Déroulé de `RSA` :

1. On génère deux nombre premiers aléatoire : $p$ et $q$

4. appelé *Indicateur d'Euler*, sert à calculer $e$ et $d$

# Sécurisation des communications

Déroulé de `RSA` :

1. On génère deux nombre premiers aléatoire : $p$ et $q$
2. On réalise $p \times q$ afin d'avoir $n$, mais également $(p-1) \times (q-1)$ pour avoir $\phi(n)$ [4]

4. appelé *Indicateur d'Euler*, sert à calculer $e$ et $d$

# Sécurisation des communications

Déroulé de `RSA` :

1. On génère deux nombre premiers aléatoire : $p$ et $q$
2. On réalise $p \times q$ afin d'avoir $n$, mais également $(p-1) \times (q-1)$ pour avoir $\phi(n)$ [4]
3. On choisit $e$ et $d$ tel que $(e \times d) \mod \phi(n) = 1$, tout en respectant `PGCD(`$e$`,` $\phi(n)$`) = 1`

4. appelé *Indicateur d'Euler*, sert à calculer $e$ et $d$

# Sécurisation des communications

Déroulé de `RSA` :

1. On génère deux nombre premiers aléatoire : $p$ et $q$
2. On réalise $p \times q$ afin d'avoir $n$, mais également $(p-1) \times (q-1)$ pour avoir $\phi(n)$ [4]
3. On choisit $e$ et $d$ tel que $(e \times d) \mod \phi(n) = 1$, tout en respectant `PGCD(`$e$`,` $\phi(n)$`) = 1`
4. La clé publique est $(e, n)$
5. La clé privée est $(d, n)$

4. appelé *Indicateur d'Euler*, sert à calculer $e$ et $d$

# Sécurisation des communications

Comment chiffrer un texte avec `RSA` ?

# Sécurisation des communications

Comment chiffrer un texte avec `RSA` ?

Il suffit simplement de chiffrer la représentation `ASCII` ou `Unicode` de chaque caractère du texte

# Sécurisation des communications

Lorsqu'un utilisateur accède à un site Web via `HTTPS`, les étapes suivantes se produisent :

1. L'utilisateur envoie une demande au serveur Web en utilisant `HTTPS`. Cette demande se fait à l'aide du protocole `SSL/TLS`

# Sécurisation des communications

Lorsqu'un utilisateur accède à un site Web via `HTTPS`, les étapes suivantes se produisent :

1. L'utilisateur envoie une demande au serveur Web en utilisant `HTTPS`. Cette demande se fait à l'aide du protocole `SSL/TLS`
2. Le serveur Web reçoit la demande et envoie une réponse contenant un certificat numérique, contenant une clé publique, signé par une autorité de certification

# Sécurisation des communications

Lorsqu'un utilisateur accède à un site Web via `HTTPS`, les étapes suivantes se produisent :

1. L'utilisateur envoie une demande au serveur Web en utilisant `HTTPS`. Cette demande se fait à l'aide du protocole `SSL/TLS`
2. Le serveur Web reçoit la demande et envoie une réponse contenant un certificat numérique, contenant une clé publique, signé par une autorité de certification
3. Le navigateur Web de l'utilisateur vérifie l'authenticité du certificat. S'il est considéré comme authentique, le navigateur utilise la clé publique pour chiffrer les données qui seront envoyées

# Sécurisation des communications

Lorsqu'un utilisateur accède à un site Web via `HTTPS`, les étapes suivantes se produisent :

1. L'utilisateur envoie une demande au serveur Web en utilisant `HTTPS`. Cette demande se fait à l'aide du protocole `SSL/TLS`
2. Le serveur Web reçoit la demande et envoie une réponse contenant un certificat numérique, contenant une clé publique, signé par une autorité de certification
3. Le navigateur Web de l'utilisateur vérifie l'authenticité du certificat. S'il est considéré comme authentique, le navigateur utilise la clé publique pour chiffrer les données qui seront envoyées
4. Le serveur Web utilise sa clé privée pour déchiffrer les données envoyées

# Sécurisation des communications

Lorsqu'un utilisateur accède à un site Web via `HTTPS`, les étapes suivantes se produisent :

1. L'utilisateur envoie une demande au serveur Web en utilisant `HTTPS`. Cette demande se fait à l'aide du protocole `SSL/TLS`
2. Le serveur Web reçoit la demande et envoie une réponse contenant un certificat numérique, contenant une clé publique, signé par une autorité de certification
3. Le navigateur Web de l'utilisateur vérifie l'authenticité du certificat. S'il est considéré comme authentique, le navigateur utilise la clé publique pour chiffrer les données qui seront envoyées
4. Le serveur Web utilise sa clé privée pour déchiffrer les données envoyées
5. Les données échangées entre le navigateur et le serveur sont ainsi sécurisées