Crypto101 Express

Public-key encryption

DaVinciCode

02/10/20

Description

- Système dans lequel vous avez une paire de clés
  - une clef publique, utilisée pour chiffrer des messages
  - une clef privée, utilisée pour déchiffrer des messages
- impossible?

Description

- Années 70: RSA
  - 3 cryptologues du MIT: Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman

Description

- 3 types d'algorithmes
  - algorithmes d'échange de clés (e.g. Diffie-Hellman)
  - algorithmes de chiffrement (e.g. RSA)
  - algorithmes de signature numérique

Pourquoi on ne l'utilise pas toujours?

- Très lent
- On utilise souvent ces algorithmes pour s'échanger une clef qu'on utilisera par la suite avec un algorithme de chiffrement symétrique (e.g. AES) pour nos communications

RSA

- arithmétique modulaire
- version simplifiée

RSA

- pour générer une clé, on choisit deux nombres premiers très grands ($\{p, q\}$)
- à partir de ces nombres, on génère le modulus $N = p \times q$ (publique)
- on choisit un exposant e, très souvent 3 ou 65537 (publique)

```
bin(3)
```

```
## '0b11'
```

```
bin(65537)
```

```
## '0b10000000000000001'
```

Chiffrement RSA

- clef publique: $\{N, e\}$
- pour chiffrer un message M en utilisant une clef publique:
  - $C \equiv M^e \pmod{N}$

Déchiffrement RSA

- il existe un nombre d (exposant de déchiffrement) qui nous permet de passer de C à M
- il est facile à calculer sachant $\{p, q\}$
- $M \equiv C^d \pmod{N}$
- la sécurité de cet algorithme repose sur
  - C indéchiffrable sans d
  - d très compliqué (presque impossible) à calculer à partir de la clef publique $\{N, e\}$

Futurama

- $C \equiv M^e \pmod{N}$
- pour casser RSA, on a besoin de factoriser $N = p \times q$ pour calculer d
- il n'existe pas d'algorithme qui permette de factoriser N de manière efficace
- il existe un algorithme (algorithme de Shor) qui permet de factoriser N sur un ordinateur quantique