# Chapitre 5 : Communication Anonyme
# Mixnets, Onion Routing, Tor

Cours de Cryptographie

12 janvier 2026

## Table des matières

# 1 Introduction

## 1.1 Motivation

Le chiffrement protège le **contenu** des messages mais pas les **métadonnées** :

— Qui communique avec qui ?

— Quand ? À quelle fréquence ?

— Depuis où ? (adresses IP, localisation)

— Patterns de trafic (tailles, timings)

**Problème** : Les métadonnées révèlent énormément d'informations !
**Objectif** : Communiquer de manière **anonyme** (cacher l'identité) et/ou **non-traçable** (impossible de lier deux communications du même utilisateur).

## 1.2 Applications

— Dissidents politiques, journalistes dans régimes autoritaires

— Whistleblowers (ex : Edward Snowden, WikiLeaks)

— Navigation web privée

— Contournement de censure

— Protection contre surveillance de masse

## 1.3 Définitions

**Anonymat de l'émetteur** : Cacher l'identité de celui qui envoie
**Anonymat du récepteur** : Cacher l'identité de celui qui reçoit
**Unlinkability** : Impossible de lier deux messages au même utilisateur
**Adversaire global** : Peut observer tout le réseau (NSA-level)

# 2 Chaum's Mixnet (1981)

## 2.1 Principe

**Idée** : Faire passer les messages par une cascade de serveurs (mixes) qui :

1. Déchiffrent une couche de chiffrement
2. Réordonnent les messages (batching)
3. Transmettent au nud suivant

## 2.2 Construction

**Setup** : $n$ mixes $M_1, \ldots, M_n$ avec paires de clés $(pk_i, sk_i)$
**Alice envoie message $m$ à Bob** :

1. Alice construit :
$$c = \mathsf{Enc}_{pk_1}(\mathsf{Enc}_{pk_2}(\cdots \mathsf{Enc}_{pk_n}(m, \text{Bob}) \cdots))$$
2. Envoie $c$ à $M_1$
3. $M_1$ déchiffre avec $sk_1$, obtient $c_2 = \mathsf{Enc}_{pk_2}(\cdots)$, transmet à $M_2$
4. $M_2$ déchiffre avec $sk_2$, obtient $c_3$, transmet à $M_3$
5. ...
6. $M_n$ déchiffre, obtient $(m, \text{Bob})$, délivre à Bob

### 2.3 Sécurité

**Théorème (informel)** : Si **au moins un** mix est honnête, l'anonymat est garanti (adversaire ne peut pas lier Alice → Bob).

**Limitations** :

— Nécessite batching ⇒ latence élevée

— Vulnérable aux attaques de trafic (si adversaire observe entrées/sorties d'un mix)

— Nécessite serveurs de confiance

## 3 Onion Routing

### 3.1 Principe

**Différence avec mixnet** : Pas de batching, communication en temps réel

**Métaphore** : Pelure d'oignon - chaque nud enlève une couche de chiffrement

### 3.2 Construction

**Circuit** : Alice choisit un chemin $R_1 \to R_2 \to R_3 \to$ Destination

**Encryption layers** :

— Layer 3 (extérieur) : Chiffré pour $R_1$

— Layer 2 : Chiffré pour $R_2$

— Layer 1 (intérieur) : Chiffré pour $R_3$

**Traversée** :

— $R_1$ déchiffre layer 3, voit "transmettre à $R_2$", envoie

— $R_2$ déchiffre layer 2, voit "transmettre à $R_3$", envoie

— $R_3$ déchiffre layer 1, voit destination finale, envoie

**Propriété** : Chaque nud connaît seulement le précédent et le suivant (pas la source ni la destination complète)

## 4 Tor (The Onion Router)

### 4.1 Architecture

**Composants** :

— **Relays** : ∼7000 nuds volontaires

— **Entry guards** : Premier nud du circuit

— **Exit nodes** : Dernier nud (celui qui sort vers Internet public)

— **Directory servers** : Maintiennent la liste des relays

— **Hidden services** (.onion) : Sites accessibles uniquement via Tor

### 4.2 Fonctionnement

**Établissement de circuit** :

1. Client choisit 3 relays aléatoires : Guard → Middle → Exit
2. Négocie clés symétriques avec chaque relay (Diffie-Hellman)
3. Construit circuit avec 3 couches de chiffrement

**Transmission** :

— Données chiffrées 3 fois (onion)
— Chaque relay enlève une couche
— Exit node envoie en clair vers destination (peut voir contenu si pas HTTPS !)

### 4.3 Sécurité et limitations

**Menaces atténuées** :
— Surveillance locale (FAI)
— Géolocalisation
— Censure par blocage IP

**Limitations** :
— **Exit node eavesdropping** : Exit voit trafic non-HTTPS
— **Traffic correlation** : Adversaire global peut corréler entrées/sorties
— **Timing attacks** : Patterns temporels peuvent révéler liens
— **Compromission de guards** : Si guard malveillant, peut tracer utilisateur
— **Performance** : 3-5x plus lent que connexion directe

### 4.4 Hidden Services (.onion)

**Principe** : Serveur accessible uniquement via Tor, sans révéler son IP
**Fonctionnement** :

1. Serveur choisit introduction points
2. Publie descripteur avec clé publique dans DHT
3. Client récupère descripteur
4. Client et serveur se rencontrent au rendezvous point
5. Communication via circuit Tor double (6 sauts !)

**Exemples** : Facebook .onion, ProtonMail .onion, sites de whistleblowing

## 5 Attaques sur Tor

### 5.1 Traffic Analysis

**Attack** : Adversaire observe à la fois l'entrée (utilisateur → Guard) et la sortie (Exit → Destination)
**Corrélation** : Patterns temporels, tailles de paquets ⇒ lien probable
**Défense** : Padding, dummy traffic (coût en bande passante)

### 5.2 Website Fingerprinting

**Attack** : Classifier le site visité à partir du pattern de trafic chiffré
**Résultats** : Accuracy > 90% pour top-100 sites (recherche académique)
**Défense** : WTF-PAD (adaptive padding), mais overhead important

### 5.3 Sybil Attacks

**Attack** : Adversaire déploie beaucoup de relays malveillants
**Impact** : Augmente probabilité que circuit contienne nud malveillant
**Défense** : Directory authorities vérifient relays, limitent influence nouveaux nuds

# 6 Alternatives et extensions

## 6.1 I2P (Invisible Internet Project)

**Différence avec Tor** :

— Réseau overlay complètement séparé (pas d'accès à Internet public par défaut)

— Circuits bidirectionnels (vs unidirectionnels Tor)

— Optimisé pour hidden services

## 6.2 Mixminion (email anonyme)

**Principe** : Mixnet pour emails avec réponses anonymes

## 6.3 Private Information Retrieval (PIR)

**Problème** : Récupérer un élément d'une base de données sans révéler lequel
**Solutions** :

— PIR computationnel (chiffrement homomorphe)

— PIR information-théorique (multiple serveurs non-colluding)

# 7 Notebooks pratiques

— $\texttt{05}_{d}emo_{o}nion_{r}outing.ipynb : Simulationonionroutingsimplifi$

- $\texttt{05}_{d}emo_{m}ixnet.ipynb : Implmentationmixnetavecbatching$

- $\texttt{05}_{e}xercices.ipynb : Exercicessuranonymatettraffica nalysis$

# 8 Considérations éthiques et légales

**Usage légitime** :

— Protection de dissidents, journalistes

— Contournement de censure

— Vie privée numérique

**Usage illégal** :

— Darknet markets (drogue, armes)

— Distribution de contenu illégal

— Cybercriminalité

**Position légale** :

— Tor est légal dans la plupart des pays (USA, Europe)

— Utilisé par militaires US, journalistes, ONG

— Financé en partie par US Government (Bureau of Democracy, Human Rights and Labor)

— Mais interdit/bloqué dans certains pays (Chine, Iran)

# 9 Conclusion

**Points clés** :

— Anonymat $\neq$ chiffrement (protège métadonnées, pas seulement contenu)

— Mixnets : Batching, haute latence, forte anonymat

— Onion Routing / Tor : Temps réel, latence acceptable, anonymat pratique

— Adversaire global reste une menace (traffic correlation)

— Trade-off fondamental : Anonymat vs Performance vs Usabilité

**Recherche active** :

— Résistance aux adversaires globaux

— Anonymat post-quantique

— Systèmes à faible latence avec anonymat fort

— Blockchain et anonymat (Zcash, Monero)

*"Privacy is necessary for an open society in the electronic age."*
— A Cypherpunk's Manifesto, Eric Hughes (1993)