Anonymat et vie privée sur internet

Mémoire

Guillaume Pillot

Maîtrise en informatique - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Anonymat et vie privée sur internet

Mémoire

Guillaume Pillot

Sous la direction de:

Mohamed Mejri, directeur de recherche

Résumé

L'explosion de la bulle internet au début des années 2000 a eu d'énormes impacts sociaux et économiques. Aujourd'hui, le nombre d'internautes approche les quatre milliards et internet s'est ancré dans notre vie quotidienne. De plus en plus d'informations circulent dans ce réseau et depuis les révélations d'Edward Snowden, le public a pris conscience du besoin de protéger sa vie privée.

Ce mémoire présente dans un premier temps les concepts généraux de l'anonymat et de la protection de la vie privée sur internet. Ensuite, les réseaux anonymes les plus populaires y sont étudiés : JAP, Mixmaster, TOR et I2P.

Nous verrons que la meilleure protection de ces réseaux est leur taille. [1] a élaboré un système de paiement pour rémunérer les relais de TOR dans le but d'encourager les internautes à participer sur le réseau anonyme. Nous verrons comment adapter ce système au réseau anonyme I2P.

Abstract

Since the beginning of this century, the explosion of the internet has had an important social and economic impact. Today, the number of internet users has approached four billion and it has become a part of our daily lives. More and more information circulates on the internet and since Edward Snowden's global surveillance disclosure in 2013, the public is now aware about the necessity to protect their private lives.

In a first time, this thesis introduces anonymity and privacy general concepts'. Then, the following popular anonymous networks are studied: JAP, Mixmaster, TOR and I2P.

We will see that the best protection for these network is their size. [1] has elaborates a payment system for remunerates the TOR relays in order to encourage Internet users to participate in the anonymous network. We will see how adapt this system on the I2P anonymous network.

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes et abréviations

API Application Programming Interface

DHT Distributed Hash Table

DNS Domain Name System

DoS Denial of Service

DDoS Distributed Denial of Service

EFF Electronic Frontier Foundation

FAI Fournisseur d'Accès à Internet

FTP File Transfer Protocol

HTTP HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

I2P Invisible Internet Project

IRC Internet Relay Chat

JAP Java Anon Proxy

IETF Internet Engineering Task Force

IID Interface IDentifier

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

MAC Media Access Control

MIT Massachusetts Institute of Technology

NAT Network Address Translation

P2P peer-to-peer

PGP Pretty Good Privacy

RFC Request For Comments

SMTP Simple Mail Transfer Protocol

TCP Transfer Control Protocol

TLD top-level domain

TLS Transport Layer Security

TOR The Onion Router

UDP User Datagram Protocol

URL Uniform Resource Locator

VPN Virtual Private Network

Prétendre que votre droit à une sphère privée n'est pas important parce que vous n'avez rien à cacher n'est rien d'autre que de dire que la liberté d'expression n'est pas essentielle, car vous n'avez rien à dire

---

E.Snowden

Remerciements

Je remercie en premier lieu mes parents qui m'ont toujours soutenu dans mes projets. Sans vous, arriver au bout de mes études aurait été bien plus difficile. Je remercie mon directeur de recherche Mohamed Mejri pour son encadrement et son suivi tout en long de ma maîtrise. Je remercie la faculté sciences et génie de l'université Laval pour avoir toujours répondu a mes besoins lorsque je m'occupai du club de hacking. Je remercie le hackfest pour toute la promotion apportée dans le domaine de la sécurité informatique dans la ville de Québec.

Introduction

0.1 Motivation

L'anonymat garantit la liberté d'expression et préserve la vie privée. Elle protège les individus ordinaires face aux possibles représailles de gouvernements, multinationales, groupes de pression ou organisations criminelles. Comme le mentionne [2], la vie privée est un droit décrit dans les articles 12 et 19 de la déclaration universelle des droits de l'homme. Dans la plupart des pays démocratiques, l'anonymat est garanti par la loi, entre autres pour le vote à bulletin secret.

Les données qui transitent sur Internet augmentent de manière exponentielle d'année en année. Il existe aujourd'hui une multitude d'applications collectant les informations des internautes, que ce soit avec leur accord ou non. On peut citer les réseaux sociaux comme Facebook ou le pistage des moteurs de recherches comme Google. Pour certains services, les informations sont confidentielles et doivent être protégées. Dans le premier chapitre de son livre "Anonymous Communication Networks"[3], Kun Peng mentionne l'e-finance, le commerce électronique, la télésanté ou le vote en ligne.

L'article [2] classifie en quatre catégories les principales menaces contre la vie privée et l'anonymat sur Internet.

Les deux premières sont pour des raisons sociales et politiques. La capacité d'observer toutes communications sur Internet peut permettre de prédire l'opinion publique et de l'influencer. Dans les pays autoritaires, les dissidents politiques seront plus repérables et il sera plus aisé d'empêcher la diffusion de leurs idées. Plusieurs pays ont mis ou prévoient de mettre en place des lois de surveillance sur Internet provocant plusieurs polémiques sur le respect de la vie privée. L'Affaire Snowden, la loi HADOPI ou le projet de loi C-30 en sont de bons exemples.

Les problèmes technologiques sont la troisième catégorie. Le manque de technologies appropriées peut permettre la fuite d'informations confidentielles intentionnelles ou non. On peut citer les bogues logiciels ou la mauvaise configuration des services Internet.

Les raisons économiques sont la dernière catégorie. En ayant accès aux informations privées et aux comportements des utilisateurs d'Internet, une compagnie peut cibler ses publicités et

améliorer ses produits. De plus, la vente de données personnelles, légales ou illégales, génère plusieurs milliards de dollars par année. Les géants du web ainsi que les gouvernements ne sont pas les seuls à collecter ces données. La RAND (Research ANd Development) Corporation a publié un rapport en 2014 intitulé "Markets for Cybercrime Tools and Stolen Data"[4] où il y est décrit le fonctionnement et la complexité du marché noir en ligne.

En conséquence, la population cherche de plus en plus a être anonyme sur Internet comme le montre les articles [5] et [6].

Trend Micro a publié un article en 2013 intitulé "Deepweb and Cybercrime"[7]. On peut y voir que l'anonymat sur Internet entraîne irrémédiablement toutes sortes d'activités clandestines tel que la vente de drogues, d'armes, de contrebandes, de pornographie pédophile ou d'autres services illégaux (contrat d'assassinat ou piratage informatique) ainsi que du cyberterrorisme. Plusieurs sites de commerce électronique mettant à disposition ces services ont vu le jour générant des bénéfices de plusieurs millions de dollars. Les transactions s'effectuent majoritairement avec des cryptomonnaies comme le bitcoin ou le litecoin. La nature décentralisée de ce nouveau type de monnaie rend la traçabilité des échanges plus difficile. L'exemple le plus connu est Silk Road dont le fondateur a été arrêté et le site fermé en 2013 par le FBI (Federal Bureau of Investigation).

Néanmoins, l'anonymat permet aux dissidents politiques de s'exprimer et aux journalistes de diffuser des informations sensibles comme le fait Wikileaks.

L'anonymat sur Internet permet à la fois des activités légitimes et illégitimes. La loi doit trouver un équilibre entre une prévention efficace des crimes et la protection de la vie privée et de la liberté d'expression.

0.2 Problématique et objectifs

Dans un premier temps, un état de l'art de l'anonymat et de la vie privée sur Internet sera effectué. Le fonctionnement des réseaux anonymes y sera étudié en profondeur.

Nous essaierons de répondre aux questions suivantes :

— Quelles sont les technologies permettant de surveiller les internautes ?

— Quels sont les moyens permettant de se protéger ?

— Qu'est-ce qu'un réseau anonyme ?

— Quelles sont les attaques existantes pour désanonymiser ces réseaux ?

Nous verrons que la principale défense des réseaux anonymes est leur taille. Sur TOR, les relais sont publics et maintenus par des bénévoles. Non seulement ils fournissent leur bande passante gratuitement pour que le réseau puisse fonctionner mais en plus, ils doivent payer les

frais pour maintenir leur serveur (électricité, abonnement du fournisseur d'accès à Internet, etc.). De ce fait, les questions suivantes s'imposent :

— Comment encourager les internautes à participer dans les réseaux anonymes ?

— Quel système de rémunération est-il possible de créer sans diminuer l'anonymat et la performance des réseaux anonymes ?

Pour ce faire, nous devons premièrement :

— Décrire ce qu'est l'anonymat et les termes qui lui sont associés.

— Comprendre le fonctionnement d'Internet et les problèmes que cela implique d'un point de vue de la protection de la vie privée.

— Étudier les différentes technologies mises en place pour se protéger.

— Étudier le fonctionnement des réseaux anonymes existants.

— Étudier les attaques de désanonymisation développées sur ces réseaux.

Pour ensuite pouvoir :

— Étudier les différentes approches pour encourager les internautes à participer sur les réseaux anonymes.

Par la suite, nous verrons qu'une des meilleures solutions est celle de [1]. La très grande majorité de la littérature porte sur TOR. Le système de paiement est donc conçu uniquement pour ce réseau anonyme. Nous verrons comment adapter ce système au deuxième réseau anonyme le plus populaire : I2P.

0.3 Structure du mémoire

Le mémoire est organisé en cinq chapitres :

— Le premier chapitre porte sur des notions préliminaires qu'il est essentiel de connaître tel que la suite des protocoles Internet TCP/IP ou le chiffrement asymétrique.

— Le deuxième chapitre porte sur la vie privée et l'anonymat sur Internet. Il débute sur une définition de la vie privée et des terminologies qui lui sont associées. Ensuite, les moyens de surveillance et de protection de la vie privée sont passés en revue.

— Le troisième chapitre porte sur le fonctionnement des réseaux anonymes. Le fonctionnement des relais de proxy et du chiffrement par couches y est expliqué. Les caractéristiques de ces réseaux ainsi que les attaques existantes y sont ensuite énumérées.

— Le quatrième chapitre rassemble une étude sur quatre réseaux anonymes populaires : JAP, Mixmaster, TOR et I2P.

— Le dernier chapitre porte sur le fonctionnement du système de citepow et son adaptation au réseau anonyme I2P.

Chapitre 1

Notions préalables

1.1 Introduction

Internet est le plus gros réseau de communications existant. D'après l'Internet World Stat[8], le nombre d'internautes atteindrait environ 3 675 millions. Ce réseau a été conçu à la base pour des besoins scientifiques et militaires via l'ARPANET. Internet tel que nous le connaissons aujourd'hui est venu avec l'apparition du Web au début des années 90. Le principe du Web consiste à naviguer entre des documents (page Web) reliés par des hyperliens. Ceux-ci pointant vers des URL (Uniform Resource Locator) qui sont des adresses Web permettant d'identifier de manière unique une ressource sur le Web et d'y accéder. À l'époque, il n'était pas question de protection de la vie privée et par conséquent Internet n'a pas été conçu dans ce sens. Ce chapitre énumère des notions essentielles à la compréhension d'Internet et de la suite de ce mémoire.

1.2 Domain Name System

Une adresse IP étant numérique, elle est facilement traitée par une machine, mais difficilement mémorisable pour un être humain. Le système de noms de domaine ou DNS, permet de traduire un nom de domaine plus facile à retenir, par exemple www.google.ca, en une adresse IP. Le DNS est une hiérarchie dont le sommet est nommé racine représenté par un point. Les domaines situés immédiatement en-dessous de la racine, sont nommés domaine de premier niveau (TLD). Ils correspondent aux domaines génériques comme .org ou .com ou au code de pays comme .ca ou .fr.

Pour obtenir l'adresse IP attachée à un nom de domaine, l'hôte effectue ses requêtes DNS à un resolver. Celui-ci ne connaît que les adresses IP des serveurs DNS de la racine. En questionnant l'un d'eux, celui-ci fournit les noms des serveurs DNS du TLD recherché. Le resolver questionne ensuite l'un d'eux pour recevoir les noms des serveurs DNS du sous-domaine recherché. L'opération continue jusqu'à l'obtention de l'adresse IP correspondant au nom de domaine

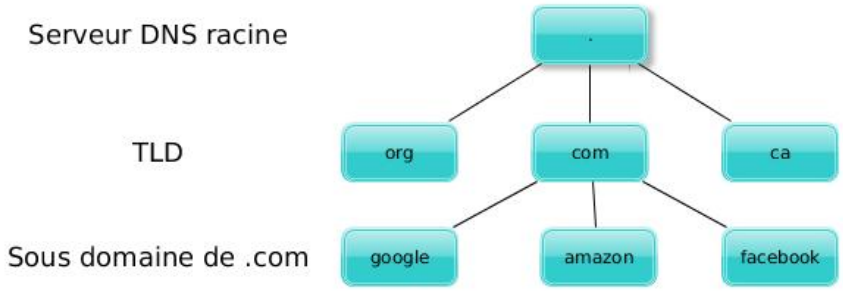

FIGURE 1.1 – Hiérarchie du DNS

demandé par l'hôte. Dans le but d'éviter de répéter toutes ces opérations, les correspondances de nom de domaine à une adresse IP sont conservées dans le cache du resolver. Les FAI fournissent habituellement un resolver à leur client, mais ils existent des résolvers ouverts comme Google Public DNS (8.8.8.8) ou OpenDNS.

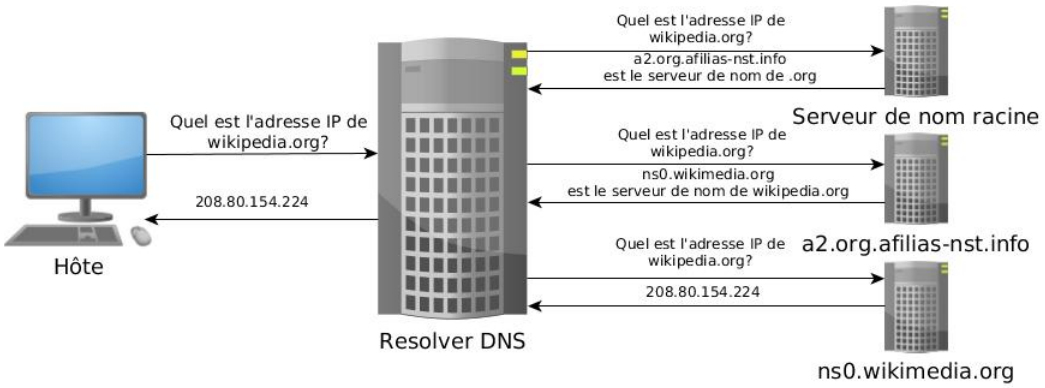

FIGURE 1.2 – Résolution DNS avec wikipedia.org

1.3 Protocole IP version 6

La version 4 du protocole IP permet d'attribuer, en théorie, 2^{32} adresses publiques, soit environ 2 294 millions d'IP. Avec la multiplication d'appareils mobiles connectés, l'avènement de l'IoT (Internet of Things) ainsi que le nombre d'internautes toujours croissant, les adresses IP publiques disponibles manquent.

La version 6 du protocole IP est destinée à remplacer l'IPv4. Une adresse IPv6 est composée de 128 bits au lieu de 32, il est donc possible d'avoir 2^{128} adresses publiques, soit un nombre quasi illimité d'adresses permettant à chaque machine d'avoir son adresse IP publique. La notation d'une adresse IPv6 est hexadécimale, séparée en 8 groupes de 2 octets par le caractère ":", par

exemple : 2002:8ac3:802d:1242:20d1:60ff:fe38:6d16. Une interface réseau peut avoir plusieurs adresses IPv6 de portée différente. L'adresse de lien local est utilisée pour contacter un hôte directement connecté (via un commutateur par exemple). L'adresse globale est, elle, accessible partout. C'est l'adresse publique sur Internet. Cette adresse est divisée en deux parties de 64 bits.

La première partie est le préfixe et est structurée comme suit :

— IANA : L'Internet Assigned Numbers Authority est l'organisme qui gère les adresses IP sur Internet. Les 3 premiers bits d'une adresse IPv6 globale doivent commencer par 2000.

— RIR : Le Regional Internet Registry correspond à la zone géographique où est située la machine.

— LIR : Le Local Internet Registry correspond au FAI, à l'entreprise ou à l'institution académique qu'utilise la machine.

— Client : Ce bloc identifie le client.

— Sous-réseau : Ce bloc permet au client de gérer des sous-réseaux.

FIGURE 1.3 – Structure du préfixe d'une adresse globale IPv6

La deuxième partie est l'IID pour Interface IDentifier qui identifie l'interface de la machine. Il existe différents mécanismes d'allocation d'IID :

— Manuel.

— Via SLAAC (Stateless Address Autoconfiguration) en générant l'IID à partir de l'adresse MAC, d'une clé cryptographique, d'un nombre temporaire ou d'une valeur stable.

— Via DHCP (Dynamic Host Configuration Protocol).

— À partir d'une adresse IPv4.

1.4 Chiffrement symétrique/asymétrique et fonctions de hachage

1.4.1 Chiffrement symétrique

Le chiffrement symétrique est la première forme de chiffrement à être apparu. Elle consiste à chiffrer un message à l'aide d'une clé. Pour déchiffrer ce message, il faut utiliser la même clé.

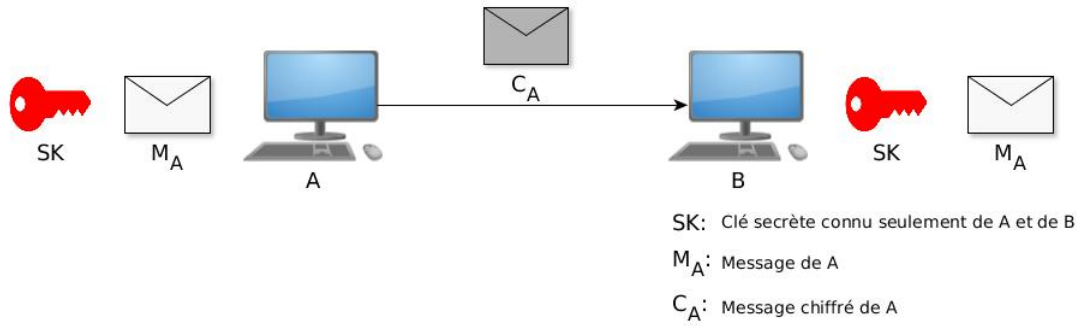

FIGURE 1.4 – Fonctionnement du chiffrement symétrique

Un des chiffrements symétriques les plus utilisés et les plus sûrs du monde est l'AES pour Advanced Encryption Standard. C'est le standard de chiffrement du gouvernement des États-Unis et a été approuvé par la NSA. L'algorithme prend en entrée un bloc de 16 octets. La taille d'une clé peut être de 128, 192 ou 256 bits. Chaque bloc subit une séquence de quatre transformations. Celles-ci, sont répétées un certain nombre de fois suivant la taille de la clé. 10 pour 128 bits, 12 pour 192 et 14 pour 256. Le but de ces transformations permet au chiffrement de respecter les deux propriétés de cryptologie suivantes :

— La confusion : Il est difficile de lier le message en clair avec le message chiffré.

— La diffusion : Une différence, même minime, entre deux messages clairs, doit entraîner deux messages chiffrés très différents.

De plus, le seul secret à connaître est la clé. Même les concepteurs du chiffrement ne peuvent pas déchiffrer un message sans la clé.

1.4.2 Chiffrement asymétrique

Le problème du chiffrement symétrique est qu'il est nécessaire d'échanger les clés via un canal secret. Pour résoudre ce problème, le chiffrement asymétrique fut créé. Son principe repose sur l'utilisation de deux clés. L'une est publique et peut-être distribuée à tous et l'autre est privée et ne doit pas être divulguée. En chiffrant un message avec l'une de ces clés, on peut le déchiffrer avec l'autre. L'échange d'un message entre deux entités s'effectue en trois étapes :

— Le nœud A envoie sa clé publique au nœud B.

— B chiffre son message avec cette clé.

— B envoie ensuite le message chiffré à A qui peut le déchiffrer avec sa clé privée qu'il a conservée.

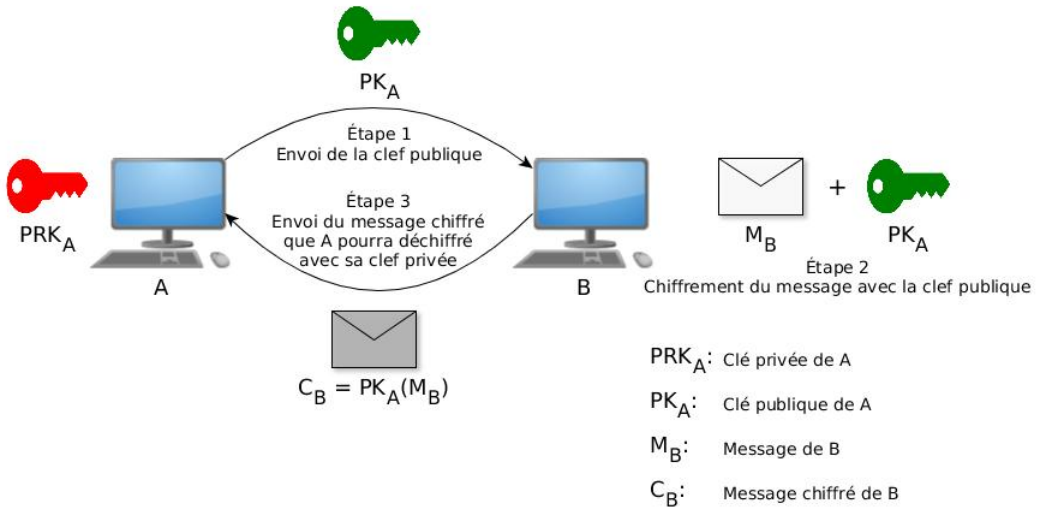

FIGURE 1.5 – Fonctionnement du chiffrement asymétrique

Ainsi, si un attaquant écoute la communication, il ne peut déchiffrer le message sans la clé privée de A.

L'inconvénient du chiffrement asymétrique est qu'il est beaucoup plus lent que le chiffrement symétrique. Pour pallier ce problème, le chiffrement asymétrique peut être utilisé seulement pour l'échange de la clé symétrique, le reste de la communication s'effectue par la suite avec le chiffrement symétrique. On appelle cette méthode le chiffrement hybride.

1.4.3 Fonction de hachage

Une fonction de hachage permet de calculer l'empreinte (ou hash) d'une donnée. Chaque empreinte permet d'identifier le contenu d'une donnée. L'empreinte est toujours de taille fixe, non réversible et unique. Les fonctions de hachage sont utilisées principalement comme somme de contrôle (checksum en anglais) ou pour stocker des mots de passe. Les fonctions de hachage les plus populaires sont MD5 (Message Digest 5), SHA-1 et SHA-256 (Secure Hash Algorithm).

1.4.4 Signature et certificat numérique

Pour assurer la non-répudiation d'un message, c'est-à-dire prouver que le message provient de son propriétaire, la signature numérique est employée. Une fonction de hachage est utilisée sur le message. Le résultat, nommé condensat, a les propriétés suivantes : Il est de taille fixe, est à sens unique et ne peut-être identique à un autre condensat issu d'un message différent.

Texte	Hash MD5	Hash SHA-1
Les pommes sont mûres	79249f9bfd9e34d94d02e18363767be5	73b714caf734e7f0587a4bd1fa63adafb5aeb3a0
Les pommes sont vertes	c0cb70217ed44129f2ee7c2b3603673c	45654529e21e59481da44db8295c74f4d45fea8b
pomme	ede0f9c3a1d2093e3f48fcafd3c70915	752c14ea195c460bac3c3b7896975ee9fd15eeb7

TABLE 1.1 – Exemple de fonction de hachage

Le condensat est ensuite chiffré à l'aide de la clé privée et joint au message. Le destinataire déchiffre le condensat avec la clé publique de l'expéditeur, utilise la même fonction de hachage sur le message et vérifie que les deux condensats sont identiques.

C'est avec ce mécanisme que fonctionnent les certificats numériques. Ceux-ci contiennent diverses informations d'identification sur leurs porteurs, au moins une clé publique et une signature numérique fournie par une autorité de certification. L'autorité de certification est un tiers de confiance qui délivre des certificats numériques et permet de valider l'authenticité des certificats qu'elle délivre. Les certificats numériques sont utilisés entre autres pour identifier et authentifier un serveur.

1.4.5 RSA

RSA est l'algorithme asymétrique le plus utilisé à ce jour. Il a été inventé par trois mathématiciens en 1977, Ron Rivest, Adi Shamir et Leonard Adleman. Pour générer une clé publique, l'algorithme fonctionne de la manière suivante :

1. Choix de deux nombres premiers distincts p et q.
2. Calcul de $n = p * q$
3. Calcul de l'indicatrice d'Euler $m = (p-1)(q-1)$
4. Choix d'un entier e qui soit premier avec m et inférieur à celui-ci.

Le couple *(n,e)* constitue la clé publique. À ce jour, il n'existe pas d'algorithme permettant de décomposer n, avec de grand nombre premier pour p et q, en un temps raisonnable. La clé privée peut se calculer avec l'algorithme d'Euclide étendu où elle correspond à un des coefficients de Bézout. Soit $c * u + m * v = 1$ et u est la clé privée.
Le chiffrement d'un message M s'effectue avec le calcul suivant :
$C = M^e (mod\, n)$
Et pour le déchiffrement :
$M = C^d (mod\, n)$
On peut ajouter une chaîne de caractère aléatoire à M pour éviter que C soit identique à chaque chiffrement de M.

1.4.6 PGP

PGP pour Pretty Good Privacy est un logiciel de chiffrement hybride développé en 1991 par Philip Zimmermann. Il est principalement utilisé pour la signature de donnée et la communication par courriel.

1.4.7 ElGamal

ElGamal est un autre algorithme de chiffrement asymétrique inventé par Taher ElGamal en 1984. La génération de la paire de clés s'effectue de la manière suivante :

1. Choix d'un grand nombre premier p.
2. Choix de g, un générateur du groupe cyclique Z_p.
3. Choix de la clé privée x, un nombre compris entre 1 et $p-1$.
4. Calcul de y selon l'équation suivante : $y = g^x mod\, p$

Le couple (p, g, y) constitue la clé publique.
Pour le chiffrement, le message a chiffré est représenté par plusieurs entiers. Chaque entier M doit être supérieur à 0 et doit être plus petit que p - 1. Pour chacun de ces entiers, un nombre k est généré aléatoirement et ensuite $C1$ et $C2$ sont générés avec les équations suivantes :
$C1 = g^k mod\, p$
$C2 = (M * y^k) mod\, p$
Le couple $(C1, C2)$ est le message crypté.
Pour le déchiffrement, le calcul suivant est réalisé avec la clé privée x :
$M = C2 * C1^{-x} mod\, p$
L'algorithme ElGamal est basé sur la difficulté du problème du logarithme discret. Il n'existe pas d'algorithme efficace permettant, avec un groupe cyclique G et de son générateur g, de retrouver x à partir de $y = g^x$.

Prenons l'exemple suivant, Alice veut envoyer l'entier M 89 à Bob. Bob génère le nombre premier p 1289 et le générateur g 141. Il choisit comme clé privée x 61.
Il peut maintenant calculer y :
$y = g^x mod\, p = 141^{61} mod\, 1289 = 1279$
La clé publique de Bob est (1289,141,1279).
Alice peut maintenant chiffrer son message. Elle prend pour k 99 et calcule $C1$ et $C2$:
$C1 = g^k mod\, p = 141^{99} mod\, 1289 = 732$
$C2 = (M * y^k) mod\, p = (1275^{99} * 89) mod\, 1289 = 763$
Alice transmet $C1$ et $C2$ à Bob.
Celui-ci peut déchiffrer le message avec sa clé privée :
$M = C2 * C1^{-x} mod\, p = 763 * 732^{-61} mod\, 1289$
$M = (763 mod\, 1289 * 732^{-61} mod\, 1289) mod\, 1289 = 89$
Si Alice veut envoyer un autre entier, elle choisira une autre valeur pour k.

1.5 Protocole HTTP

Le HTTP pour HyperText Transfer Protocol est un protocole de la couche application et permet le transfert de fichiers localisés par une URL (Uniform Resource Locator). Une URL est une adresse web permettant d'identifier de manière unique une ressource sur le web et d'y accéder. Le mode de communication est le client-serveur. Le client qui correspond au navigateur web, envoie une requête HTTP au serveur web. Celui-ci traite la requête et envoie une réponse HTTP.

1.5.1 Requête HTTP

Une requête HTTP est constituée d'un ou plusieurs entêtes séparés entre elles par une ligne, suivie obligatoirement par une ligne blanche. Une requête HTTP débute par une ligne constituée de la méthode HTTP utilisée, suivi de l'URL ciblé et finissant par la version HTTP utilisée. Ensuite, les autres lignes représentent les différents entêtes respectant le format suivant : **nom-header : value**.

Voici un exemple de requête HTTP :

```
GET / HTTP/1.1
Host: google.ca
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

1.5.2 HTTPS

Les communications passant par le HTTP ne sont pas cryptées. TLS (Transport Layer Security) est un des protocoles de cryptage le plus connu. Il est placé en-dessous de la couche d'application, les protocoles applicatifs ont seulement besoin de l'implémenter sans être profondément modifiés. C'est le cas pour le HTTPS (S pour Secure). TLS utilise des algorithmes de chiffrement sûr comme RSA ou AES (Advanced Encryption Standard) et il est massivement utilisé sur Internet par les banques et les sites de commerce électronique. Le HTTPS utilise des certificats numériques signés par une autorité de certification (CA) pour valider l'identité du serveur web.

1.6 Réseau P2P

1.6.1 Mode de communication Client-Serveur

C'est le mode de communication le plus populaire sur Internet. Les clients envoient des requêtes à un serveur pour obtenir un service. Cela peut être par exemple l'accès à un site web (HTTP), à des courriels (SMTP) ou à des fichiers (FTP). Pour supporter un nombre de requêtes important, les serveurs ont souvent une puissance de calcul plus grande que les clients. Ce mode de communication est donc centralisé. Le mode Client-Serveur permet une administration du réseau simplifiée et un meilleur contrôle de la distribution des données. Son principal inconvénient est son manque de robustesse, si le serveur ne fonctionne plus, les clients n'ont plus accès au service.

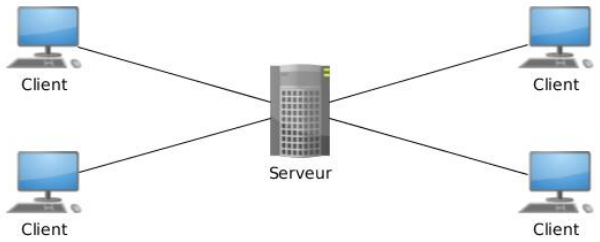

FIGURE 1.6 – Mode de communication Client-Serveur

1.6.2 Mode de communication peer-to-peer

P2P pour peer-to-peer est un mode de communication décentralisé. Tout nœud ou paire sur le réseau peut communiquer avec les autres et échanger des données. Les nœuds sont à la fois client et serveur. Il permet une meilleure résilience, si quelques nœuds dans le réseau ne fonctionnent plus, les données seront toujours accessibles. De plus la charge de travail est mieux repartie, la distribution des données n'est plus à la charge d'une seule machine. Son principal inconvénient est qu'il est plus difficile de contrôler le contenu des données.

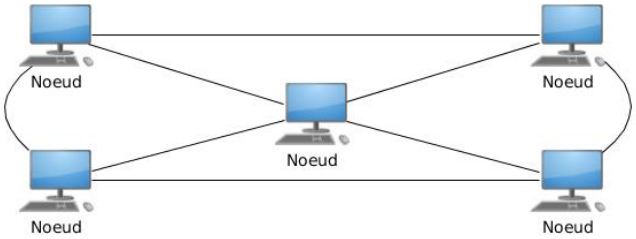

FIGURE 1.7 – Mode de communication P2P

Ces réseaux sont plus complexes à gérer que le mode client-serveur. Les plus gros réseaux P2P peuvent être constitués de millions de machines. Les différents algorithmes gérant les réseaux P2P doivent résoudre ces deux problèmes :

— Identifier chaque nœud. Le but étant de distinguer les nœuds les uns des autres. Les réseaux P2P assignent à chaque nœud un identifiant global unique.

— Localiser chaque nœud. Le réseau doit pouvoir retrouver un nœud avec son identifiant. Avec des réseaux constitués de millions de nœuds, il n'est pas possible que chaque nœud contienne une liste à jour de tous les nœuds du réseau. Des nœuds quittent et rejoignent le réseau continuellement. Au lieu de ça, un nœud conserve une liste d'un groupe de nœud. Le contenu de cette liste diffère d'un algorithme à l'autre.

1.6.3 Table de hachage distribuée

Une table de hachage est une structure de donnée de type clé-élément. La clé permet d'accéder à un élément. Celle-ci est hachée et c'est ce hash qui indexe l'élément dans la table.

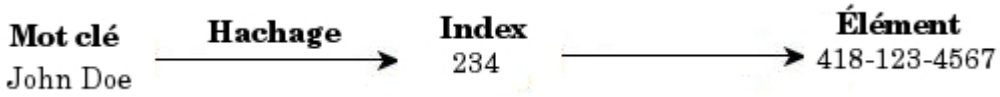

FIGURE 1.8 – Accès à un élement dans une table de hachage

La recherche d'un élément dans une table de hachage est d'une complexité de $O(1)$. Grâce au hachage, il n'est pas nécessaire de connaître l'index d'un élément pour y accéder, mais seulement son mot clé.

Une table de hachage est dite distribuée (DHT en anglais pour Distributed Hash Table) quand elle est diffusée dans un système réparti tels les réseaux P2P. Cette table contient l'annuaire des données disponibles sur le réseau. Elle est divisée et dupliquée sur plusieurs nœuds. Les premiers réseaux P2P utilisent un serveur d'annuaire centralisé pour indiquer quel nœud possède quelle donnée. Si ce serveur n'est plus accessible, les utilisateurs ne peuvent plus savoir qui possède quoi. La DHT décentralise et autonomise l'annuaire tout en rendant son accès plus résilient. De plus, le système fonctionne toujours efficacement malgré un nombre élevé de nœuds (dans l'ordre du million).

1.6.4 Kademlia

Kademlia est un protocole permettant la mise en place de DHT. Le fonctionnement du protocole est détaillé dans [9]. Kademlia est souvent utilisé dans les réseaux P2P. Chaque nœud est identifié par un nombre de 160 bits unique dans le réseau. L'ID est choisit au hasard. Il existe 2^{160} ID différents et les probabilités de choisir un ID identique à un autre nœud sont extrêmement faibles. La clé de chaque donnée est aussi un nombre de 160 bits.

Le réseau est un arbre binaire où chaque feuille correspond à l'ID d'un nœud. Chaque nœud possède une table de routage. Cette table est constituée de listes nommées k-buckets. Chacune de ses listes contient les informations de contacts des nœuds d'un des sous-arbres auquel il n'appartient pas. k étant le nombre maximum de nœuds que peut contenir chaque k-bucket. La valeur de k diffère suivant les implémentations de Kademlia. Habituellement, plus les nœuds du réseau sont fiables, plus la valeur de k baisse. Les nœuds ayant le plus d'ancienneté sont choisis. Les informations de contact sont constituées de l'ID du nœud, de son adresse IP et de son port UDP. Par exemple, prenons l'arbre ci-dessous :

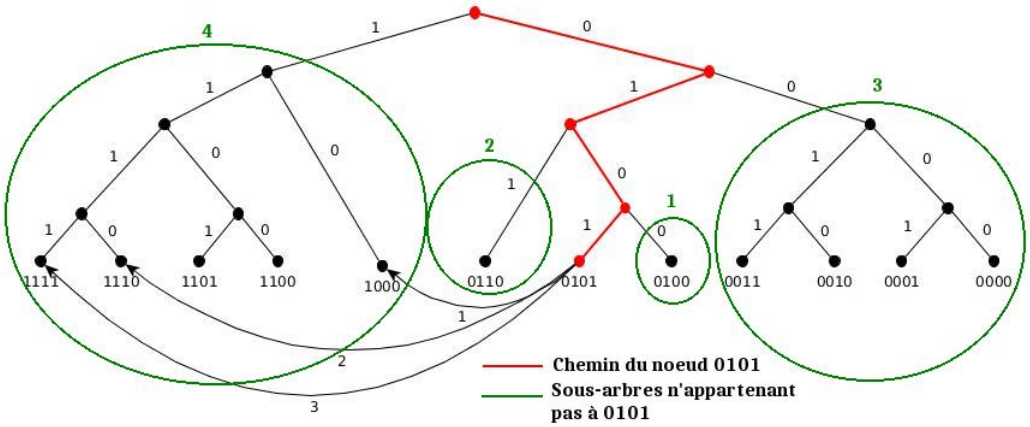

FIGURE 1.9 – Arbre binaire Kademlia

Les nœuds de ce réseau auront 4 k-buckets dans leur table de routage. Dans notre exemple, k vaut 2. Les k-buckets de 0101 seront constitué d'au plus deux nœuds de chaque sous-arbre auquel n'appartient pas le nœud. Dans la figure ci-dessus, les sous-arbres sont entouré en vert. Le premier k-bucket de 0101 ne peut être constitué que de 0100. Idem pour le deuxième k-bucket qui ne peut-être constitué que de 0110. Le troisième k-bucket est constitué des nœuds 0010 et 0000. Le quatrième k-bucket est constitué des nœuds 1000 et 1110. Ainsi chaque nœud connaît au moins un nœud de chaque sous-arbre auquel il n'appartient pas. Cela permet une recherche efficace des éléments du réseau. La complexité de la recherche d'un nœud est de $O(log(n))$ où n est le nombre de nœuds dans le réseau.

k-bucket	**k = 2**	
1	0101	-
2	0110	-
3	0010	0000
4	1000	1110

FIGURE 1.10 – Table de routage du nœud 0101

Dans la figure 1.9, le nœud 0101 recherche le nœud 1111. 0101 connaît 1000 qui appartient au

même sous-arbre que le nœud recherché, le nœud 0101 interroge ce nœud (flèche 1). 0101 ne connaît pas 1111. Mais ce nœud connaissant lui aussi au moins un nœud de chaque sous-arbre auquel il n'appartient pas, il va interroger à son tour le nœud appartenant au même sous arbre que 1111 qu'il connaît. Dans notre exemple, on considéra que c'est 1110. Le nœud 0101 interroge donc le nœud 1110 (flèche 2). Celui-ci connaît le nœud recherché 1111 et le nœud 0101 peut maintenant le contacter (flèche 3).

Pour décider quel nœud va stocker une paire clé-élément, la distance entre les nœuds et la clé est calculée grâce à l'opérateur XOR (OU-exclusif). Par exemple, la distance entre le nœud 0011 et la clé 1001 est égale à $0011 \oplus 1001 = 1010$, soit 10 en notation décimale. Le nœud ayant la distance la plus courte est choisi pour stocker la paire clé-élément. La recherche d'une clé sur le réseau s'effectue de la même manière que pour rechercher un nœud. Un nœud va chercher d'autre nœuds ayant un ID proche de la donnée stockée et il va répliquer cette donnée vers eux k fois.

Pour rejoindre le réseau, un nœud doit contacter un nœud déjà sur le réseau via son adresse IP et son numéro de port. On nomme ce nœud le nœud "bootstrap". Le nouveau nœud crée un ID unique aléatoirement. L'ID du nœud "bootstrap" est ajouté à l'un de ses k-buckets. Ensuite, le nœud effectue une recherche sur lui-même en interrogeant le nœud "bootstrap". Les nœuds interrogés pourront ajouter l'ID du nouveau nœud dans leur table de routage. Durant cette première recherche, le nouveau va ajouter les nœuds interrogés dans ses k-buckets. Pour finir, le nouveau nœud va chercher d'autre nœud pour remplir ses k-buckets.

Chaque nœud effectue une recherche sur un k-bucket auquel aucune recherche n'a été effectuée depuis plus d'une heure. Cela permet d'éviter des k-buckets non à jour. Il n'y pas d'action à effectuer pour un nœud quittant le réseau. Si un nœud ne répond pas, celui-ci est supprimé de la k-bucket.

1.7 Les Cryptomonnaies

1.7.1 Le Blockchain

Comme son nom l'indique, la blockchain est une structure de donnée de bloc chaîné. Chaque bloc contient une liste d'enregistrement. Pour les cryptomonnaies, les enregistrements correspondent à des transactions. Un blockchain est un registre de toutes les transactions qui ont été effectuées avec la cryptomonnaie qui lui ait rattaché. Les blocs sont ordonnés dans une seule et unique chaîne. Les transactions sont groupées dans des blocs suivants à quels moments elles ont eu lieu. Par exemple, une transaction située dans le bloc 10 a eu lieu avant une transaction située dans le bloc 12.

La blockchain est distribuée sur Internet, elle est complètement décentralisée, il n'y a pas d'autorité centrale. Ces transactions sont vérifiées par les nœuds du réseau relié au blockchain

et sont inscrites dans celui-ci. La blockchain ne peut donc pas être modifiée. Toute transaction est irréversible et est inscrite à jamais dans le blockchain. La blockchain résout le problème de la double dépense. Ce problème consiste à émettre deux transactions sur le même avoir. Pour éviter cela, une des meilleures solutions est d'ordonner les transactions. Une première transaction effectuée sur le réseau l'emporte sur toutes les autres effectuées sur le même avoir. L'ensemble du réseau doit se mettre d'accord pour reconnaître cette transaction et crée un consensus. Il existe plusieurs systèmes de validation permettant cela, l'un des plus utilisés pour les cryptomonnaies est la "preuve de travail".

Le système de validation par preuve de travail consiste à la résolution d'un problème mathématique difficile à résoudre que l'on nomme défi. Les transactions qui ne sont pas encore groupées dans un bloc sont dites en attente. Plusieurs nœuds créent de nouveaux blocs locaux avec des transactions en attentes. Ces nœuds sont en compétition entre eux pour savoir si leur bloc va être ajouté à la blockchain. Ils doivent résoudre le problème mathématique en premier. Le problème diffère suivant la cryptomonnaie associée à la blockchain. La difficulté est ajustée en temps réel selon la puissance totale du réseau. Les blocs sont ainsi toujours écrits à intervalles réguliers. On nomme ce processus du "minage" et les nœuds cherchant de nouveau blocs des "mineurs". Ceux-ci lorsqu'ils trouvent de nouveaux blocs sont récompensés. Les récompenses proviennent des commissions émises par les nœuds voulant ajouter de nouvelles transactions. Le système de validation par "preuve de travail" permet ainsi de protéger la blockchain de toute falsification. Il faudrait qu'un mineur malveillant contrôle plus de 50% de la puissance totale du réseau pour briser la blockchain.

1.7.2 Le Bitcoin

Le Bitcoin est la cryptomonnaie la plus utilisée aujourd'hui. La monnaie est complètement décentralisée, il n'y a pas d'autorité centrale comme une banque. Les échanges se font directement de pair à pair sur Internet. Dans la blockchain du Bitcoin, un bloc est constitué d'une liste de transaction, d'un nonce ainsi que du double hachage SHA-256 du bloc précédent. Dans le Bitcoin, le défi pour ajouter un nouveau bloc dans la blockchain consiste à calculer le double hachage SHA-256 constitué des transactions en attente, de l'identifiant du bloc précédent et du nonce. Le bloc est accepté si le hachage généré contient n zéros en entête. Plus n augmente, plus la difficulté de trouver le bon hachage augmente. Le nombre n est ajusté suivant la puissance de calcul des mineurs dans le but de maintenir la publication d'un nouveau nœud toutes les 10 minutes. Les mineurs créant de nouveaux blocs génèrent de nouveaux Bitcoin. Le nombre généré dépend du nombre de Bitcoin déjà en circulation. Le nombre diminue de façon logarithmique. On prévoit atteindre le maximum de Bitcoin fixé à 21 millions d'ici 2140, les mineurs seront ensuite rémunérés uniquement avec les commissions.

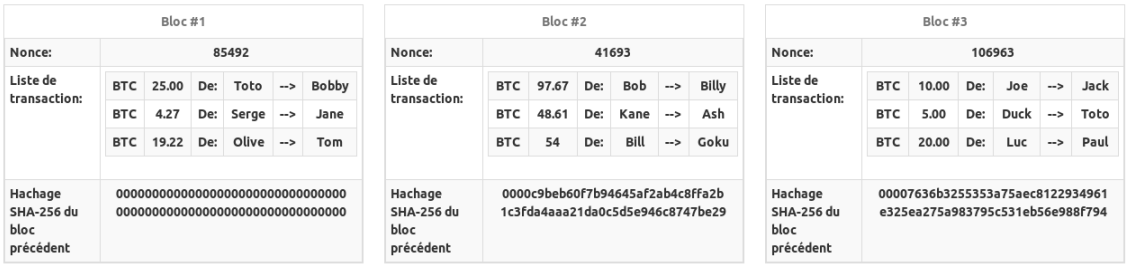

FIGURE 1.11 – Exemple de trois premiers blocs dans une blockchain

1.7.3 Les mining pool

Au début de la décennie 2010, la popularité des cryptomonnaies était plus faible qu'aujourd'hui, la concurrence pour miner l'était aussi. Un particulier avec une carte graphique (GPU) de bonne qualité pouvait miner avec un bon rendement. Aujourd'hui, il existe des entrepôts constitués de milliers de machines conçues spécialement pour le minage. Certains mineurs n'ayant pas toutes ses ressources s'associent entre eux et forment un groupe nommé "mining pool". En travaillant ensemble, ils augmentent leur chance de générer un bloc. Les gains sont ensuite partagés entre eux.

Chaque mineur d'un pool essaie de résoudre un bloc avec une difficulté plus faible que le bloc d'origine. Ce bloc se nomme une "part" ou un "share" en anglais. Suivant une certaine probabilité, la part peut résoudre également le bloc d'origine. Le mineur adapte la difficulté des shares à calculer selon sa puissance de calcul ainsi les mineurs résoudront leurs parts dans un intervalle de temps identique. Si la difficulté est trop faible, le mineur va dépenser trop de temps pour l'échange de donnée avec l'opérateur de la mining pool. Si la difficulté est trop grande, le mineur risque de ne pas soumettre autant de share qu'il le pourrait. L'opérateur fait en sorte qu'une part n'est jamais calculée une deuxième fois. En général, l'opérateur fournit une unité de travail composé d'une plage particulière de nonce. Dès que le mineur a terminé le calcul sur cette plage, il demande une nouvelle unité de travail. Ce système permet aux opérateurs de la mining pool de vérifier la quantité de travail de chaque mineur et ainsi partager les gains de manière équitable.

P2Pool

P2Pool est un modèle de mining pool décentralisé. Son fonctionnement est décrit dans [10]. Au lieu d'utiliser un ou plusieurs opérateurs, le réseau utilise une chaîne cryptographique similaire au blockchain nommé sharechain. Au lieu de contenir des transactions, la sharechain contient les récompenses des mineurs. Lorsqu'un mineur accepte un share, il est diffusé sur tout le réseau et dès qu'une majorité de participants l'accepte, la part est confirmé dans la sharechain.

L'anonymat et le bitcoin

L'historique des transactions du Bitcoin est complètement public. Toutes les transactions sont identifiées par une adresse qui correspond à la clé publique générée par l'utilisateur. La clé ne relie pas directement l'utilisateur. Néanmoins, celui-ci doit fournir des éléments de son identité pour recevoir un bien ou un service. De plus, étant donné que la blockchain est permanente, rien ne garantit qu'une transaction intraçable aujourd'hui le reste dans l'avenir. Il est possible par des techniques de data mining de relier plusieurs adresses publiques à une personne. De plus, l'utilisateur peut être identifié si celui-ci poste ses adresses publiques sur un site web ou par courriel. La communauté du Bitcoin recommande donc d'utiliser une adresse qu'une seule fois.

Pour mitiger le manque d'anonymat, il est possible de passer par "un service de blanchiment" (laundry service) nommé aussi "mixing services" ou "trumblers". L'utilisateur envoie sa monnaie à un service anonyme et celui-ci mélange la monnaie reçue avec celle d'autre utilisateur la rendant intraçable. Malheureusement, ce service comporte deux limitations. La première est qu'il faut faire confiance au service, rien n'empêche celui-ci de garder la monnaie. La deuxième est qu'il peut être illégal dans certains pays d'utiliser ce genre de service pour de gros montants.

Le site officiel du Bitcoin donne une liste des recommandations pour protéger son anonymat dans [11]. L'article [12] présente une vue d'ensemble de l'anonymat et de la vie privée sur les cryptomonnaies similaire aux Bitcoin.

1.8 Conclusion

Nous avons vu comment fonctionne les protocoles d'Internet TCP/IP ainsi que le HTTP. Ensuite, le chiffrement symétrique et asymétrique y a été expliqués. Nous avons continué avec les différents modes de communication (client-serveur et P2P) dont le protocole P2P Kademlia. Le chapitre se conclut avec le fonctionnement des cryptomonnaies , la compréhension de cette section est essentielle pour la lecture du dernier chapitre de ce mémoire.

Chapitre 2

Vie privée et anonymat sur Internet

2.1 Introduction

Ce chapitre débute avec une définition de l'anonymat et de la vie privée sur Internet. Ensuite, nous énumérerons les attaques basées sur les applications ainsi que les technologies d'écoute et les faiblesses de conception d'Internet d'un point de vue de la vie privée. Pour finir, nous verrons qu'un anonymat complet sur Internet n'est pas possible, mais que l'on peut néanmoins obtenir différents niveaux d'anonymat suivant les techniques employées.

2.2 Définition et terminologie

La vie privée est interprétée de différente façon par les juridictions. La RFC (Request for comments) 4949[13] la définit comme le droit d'un individu de contrôler la collection et la diffusion d'informations le concernant. Selon la RFC 6973[14], sur Internet, ce concept est associé aux données à caractère personnel d'un individu identifié ou identifiable.

Kun Peng mentionne dans le premier chapitre de son livre[3] ainsi que [15], la terminologie développée par Andreas Pfitzmann à propos de la vie privée. [2] se base sur les normes ISO 15408. Bien que les définitions diffèrent légèrement, les termes employés sont identiques. La suite de cette section sera basée sur ces documents.

Pour garantir la vie privée, plusieurs conditions sont requises :

— L'anonymat : C'est l'état qui garantit que l'identité d'un individu qui utilise un service ou une ressource ne puisse être divulguée.

— L'inobservabilité : C'est l'état qui garantit l'impossibilité de distinguer si un élément existe ou non.

— L'inassociabilité : C'est l'impossibilité d'établir un lien entre les activités réalisées par un utilisateur.

Il est nécessaire, pour garantir l'anonymat au sein d'un réseau de communication, de considérer l'existence d'attaquants. Un attaquant souhaite surveiller ou manipuler les communications du réseau. D'après [3], on peut classifier trois types d'attaquants :

— L'attaquant passif : Il ne fait qu'observer les communications.

— L'attaquant passif qui peut envoyer des messages : Celui-ci étant un utilisateur du réseau anonyme, il est indétectable.

— L'attaquant actif : Il peut manipuler et perturber les communications en effaçant, en modifiant, en envoyant ou encore en retardant le trafic des messages.

2.3 Surveillance sur Internet

2.3.1 Deep Packet Inspection

Plusieurs technologies ont été déployées pour observer les communications sur Internet. On nomme l'analyse du contenu de paquet réseau "Deep Packet Inspection" ou "Inspection des paquets en profondeur" en français abrégé DPI.

L'article [2] se base sur le modèle OSI pour analyser les possibilités d'écoute sur chaque couche du TCP/IP :

— La couche physique : Toute écoute sur cette couche demande un accès direct au matériel. Il est nécessaire d'utiliser des appareils nommés "TAP réseau". On les place entre un point A et un point B pour y capturer tout le trafic. Ils sont conçus pour ne pas perturber le réseau. Chaque type de réseau (RJ45, fibre optique, WIFI, etc.) a son propre TAP réseau.

— La couche liaison : La plupart des technologies ciblant cette couche analysent les adresses MAC leur permettant d'identifier les constructeurs et les modèles des interfaces présentes sur le réseau.

— La couche réseau : C'est la couche la plus importante pour l'analyse du trafic Internet. La capture de paquet IP permet d'analyser leurs entêtes et d'obtenir l'adresse IP source et l'adresse IP de destination, ainsi que le protocole de transport employé. L'obtention d'une d'adresse IP permet d'identifier directement l'utilisateur.

— La couche transport : Étant donné que le TCP est le protocole de cette couche le plus utilisé, c'est celui-ci qui est le plus ciblé pour les écoutes. L'entête TCP contient le port source et le port destination permettant de déterminer quel service est utilisé. De plus, toutes les données non cryptées au-dessus de cette couche permettent d'obtenir le contenu en clair de la transmission.

— La couche application : L'analyse du trafic sur cette couche diffère suivant le protocole ciblé. Le trafic Web et le trafic de courriel sont le plus souvent ciblés étant donné que les informations qu'elles contiennent peuvent être critiques.

2.3.2 Profilage DNS

Le DNS n'a pas été conçu pour protéger la vie privée et n'est pas chiffré. Nous avons vu dans la section DNS du chapitre des notions préalables que pour obtenir l'adresse IP attachée à un nom de domaine, l'hôte effectue ses requêtes DNS à un resolver. L'avantage de passer par un resolver masque l'adresse IP du client aux autres serveurs de noms, mais celui-ci a connaissance de tous les sites web consultés via leur nom de domaine. De plus, comme le montre la figure 1.2, tous les serveurs de noms interrogés voient le nom de domaine au complet. La révélation du programme de la NSA (National Security Agency) "MoreCowBell" montre que les vulnérabilités du DNS sont déjà exploitées, voir [16]. L'IETF (Internet Engineering Task Force) a publié le RFC 7626 "DNS Privacy Considerations" [17] qui décrit les problèmes de confidentialité du DNS.

Plusieurs solutions ont été proposées dans [16] et [18], comme le chiffrement du trafic DNS, la minimisation des requêtes DNS, qui consiste à questionner les serveurs DNS avec le strict nécessaire (par exemple pour wikipedia.org, le resolver questionnera le serveur de nom racine seulement avec .org), ou l'utilisation d'un système alternatif tels les namecoins ou le système de nom GNU. Ces solutions sont toujours en cours d'étude au sein de l'IETF.

2.3.3 Vie privée sur l'IPv6

Pour repousser l'épuisement d'adresse IPv4, les FAI utilisent le NAT (Network Address Translation) qui permet de partager une adresse IP publique entre plusieurs machines d'un même réseau local via le modem personnel du client ou celui fourni par le FAI.

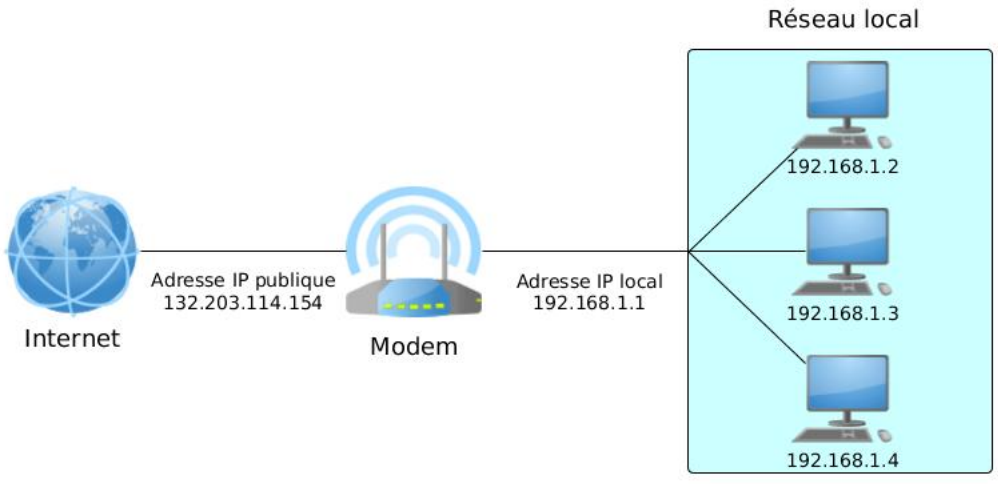

FIGURE 2.1 – Network Address Translation

Nous avons vu dans la section IPv6 du chapitre des notions préalables le fonctionnement de ce protocole. Le nombre quasi illimité d'adresses disponibles en IPv6 permet de se passer du

NAT. Un avantage du NAT est qu'étant donné que plusieurs machines sur un réseau local ont la même adresse IP publique, il est plus difficile d'identifier directement un utilisateur. Avec l'IPv6, un attaquant peut identifier directement un utilisateur avec son adresse.

La RFC 7721[19] décrit les faiblesses, d'un point vue de la protection de la vie privée, que provoquent les IID générés à partir d'une adresse MAC. L'adresse MAC d'une interface étant unique et ne pouvant être changée (sauf au niveau logiciel), il est possible de tracer une machine à chaque changement de réseau et donc de corréler les activités de son utilisateur en plus de fournir des renseignements sur le constructeur et le modèle de l'interface. L'allocation avec une valeur stable produit un IID pour un réseau donné, mais change à chaque changement de réseau. Par contre, lorsque le client revient sur un réseau, il conserve son IID. La corrélation des activités est donc toujours possible tant que l'utilisateur reste et revient dans un réseau. Les adresses temporaires ne permettent la corrélation que pendant la durée de vie de l'adresse. C'est donc la meilleure solution pour protéger la vie privée de l'utilisateur. L'inconvénient de cette méthode est que ce type d'adresse peut rendre difficile l'administration des réseaux et elles sont donc plus susceptibles d'être désactivées.

2.3.4 Attaques basées sur les applications

Un attaquant peut récupérer des informations sur l'utilisateur au niveau de l'application. Cela est dû principalement au manque de connaissance ou à la négligence de l'utilisateur, mais le développeur de l'application est aussi à prendre en compte.

Le but principal est l'obtention de l'adresse IP de l'utilisateur sans avoir besoin d'analyser le réseau, mais l'adresse IP n'est pas la seule information permettant d'identifier l'utilisateur.

Même si l'utilisateur passe par un réseau anonyme, l'attaquant peut contourner le réseau anonyme en ayant un accès direct à l'utilisateur par une attaque sur l'application. L'article [20] énumère ce genre d'attaques.

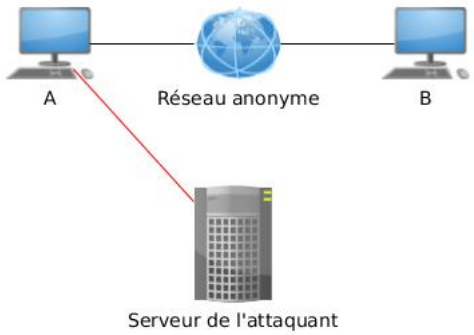

FIGURE 2.2 – Anonymat de A compromis par une connexion directe de l'attaquant

Le vecteur d'attaque le plus courant est le navigateur web. Il est le moyen principal pour accéder à Internet, il permet d'interpréter le HTML (Hypertext Markup Language). L'outil étant complexe, plusieurs fuites d'informations sont possibles.

Plug-ins

Les plug-ins ajoutent des fonctionnalités sur le navigateur web, mais peuvent permettre à un attaquant d'établir une connexion directe à l'utilisateur. Les plug-ins les plus connus sont Flash développé par Abobe Systems ainsi que QuickTime développé par Apple. L'API de Flash peut établir une connexion TCP sans passer par le réseau anonyme de l'utilisateur tandis qu'avec QuickTime, un paramètre dans la configuration permet d'établir une connexion directe pour voir une vidéo.

DNS

Une page web malicieuse peut contenir un lien vers un nom de domaine unique. Nous avons vu précédemment dans les notions préalables comment fonctionne le DNS. Étant donné que le nom de domaine est unique, il n'est donc pas stocké dans le cache du resolver et l'utilisateur doit interroger les serveurs de noms. Le navigateur de l'utilisateur effectue la recherche et il est possible que l'application d'anonymisation ne prévoie pas d'effectuer les recherches DNS en passant par le réseau anonyme. Nous avons vu dans la sous-section sur le profilage DNS comment il est possible de tracer un utilisateur à travers ses requêtes DNS. Le nom de domaine est donc unique pour chaque utilisateur permettant ainsi de les retracer.

Applet Java

Java est un langage de programmation ayant la particularité que ses applications soient aisément portables. Cette portabilité est assurée grâce à sa machine virtuelle qui permet l'exécution d'application Java sur n'importe quelle plate-forme.

Un applet Java est un logiciel écrit généralement en Java s'exécutant dans le navigateur web. Un attaquant peut exploiter un applet Java pour dévoiler l'identité d'un utilisateur en exploitant l'API Java. Il peut le faire de deux méthodes différentes.

La première consiste à utiliser la fonction de l'API qui permet de résoudre les noms de domaine. Si le nom de domaine ne correspond pas à celui du site web, l'applet lève une exception de sécurité, mais l'utilisateur interroge malgré tout les serveurs de nom.

La seconde est basée sur les paquets UDP. Lorsqu'un applet Java envoie un paquet UDP, il l'envoie directement au destinataire sans passer par le réseau anonyme. Il suffit pour l'attaquant d'ajouter un identifiant unique au paquet UDP pour révéler l'adresse IP de l'utilisateur.

En plus de ces deux méthodes permettant une connexion directe à l'utilisateur, l'API Java peut obtenir l'adresse IP, le nom d'hôte ainsi que d'autres configurations réseau de la machine de l'utilisateur. Ces informations peuvent être envoyées à travers le réseau anonyme pour compromettre l'identité de l'utilisateur.

Document actif

Un document actif permet une interaction avec l'utilisateur. Les documents actifs les plus populaires sont les fichiers PDF (Portable Document Format) développés par Adobe Systems ainsi que les documents Word et Excel développés par Microsoft. L'interaction avec les fichiers PDF est possible grâce à du code JavaScript intégré dans le fichier et à l'aide de macros pour les documents Word et Excel.

Un attaquant peut user d'ingénierie sociale et envoyer des documents actifs contenant du code malicieux à l'utilisateur. L'utilisateur non vigilant peut exécuter ce code en cliquant sur un bouton ou en visionnant une vidéo incluse dans le document par exemple. L'exécution de ce code peut connecter l'utilisateur à l'attaquant directement ou envoyer des informations personnelles.

Cookie

Un cookie est une suite d'informations générée par un serveur web pour reconnaître un client. Celui-ci est stocké sur l'ordinateur du client et laisse donc une trace de sa visite ce qui peut compromettre son anonymat.

JavaScript

JavaScript est le langage de script le plus populaire. Il permet de dynamiser les pages web. La grande majorité des navigateurs intègrent un moteur JavaScript permettant l'exécution de code JavaScript.

Un attaquant peut faire en sorte qu'un utilisateur exécute du code JavaScript malicieux à l'aide d'ingénierie sociale ou sur un site web vulnérable. L'attaquant pourra récupérer le cookie, l'historique, l'adresse IP réelle de l'utilisateur ainsi que d'autres informations sur la configuration de l'utilisateur (plug-ins utilisés, taille d'écran, etc.).

Applications BitTorrent

Le navigateur web n'est pas le seul vecteur d'attaque. Les applications P2P BitTorrent sont souvent utilisés dans les réseaux anonymes. BitTorrent est un protocole de transfert de données P2P. Lorsqu'un client demande un fichier, celui-ci est fragmenté sur différents pairs du réseau. Tous ces morceaux sont téléchargés et rassemblés sur la machine du client. Un tracker est un serveur qui permet d'amorcer un téléchargement. C'est lui qui relie les pairs possédant les

parties du fichier au client. Lorsque le client souhaite télécharger un fichier, il doit s'annoncer au tracker. Cela s'effectue avec une requête HTTP de type GET. Sur certaines applications BitTorrent, la requête peut contenir l'adresse IP du client.

Fingerprinting

Le fingerprinting permet de créer une empreinte digitale à partir des informations fuitées par le navigateur de l'utilisateur.

Plusieurs entêtes HTTP peuvent être utilisées pour traquer et profiler un utilisateur :

- Le User-agent : Il contient des informations sur la configuration du client comme la version de son navigateur, son système d'exploitation ou sa langue.
- Le Referer : Il contient l'URL qui a permis d'accéder à la page web.
- L'entête Langage : Il permet de savoir avec quelle langue est configurée le navigateur et donc de savoir d'où peut provenir géographiquement l'utilisateur.
- Le Content-Type : Il donne des indications sur le type de données que le navigateur peut interpréter.
- L'entête Encoding : Il indique avec quel encodage le navigateur reçoit les données.

L'Etag est un identifiant unique fourni par le serveur web. Il permet de valider le cache d'une ressource accessible par URL. L'Etag étant unique, il peut permettre au serveur web d'identifier l'utilisateur.

La session HTTP permet à un serveur web de garder en mémoire ses utilisateurs. Le réseau anonyme ou tout autre fournisseur d'anonymat peut générer une session HTTP. Plus cette session dure dans le temps, plus le risque que l'utilisateur soit identifié augmente.

Pour finir, certains navigateurs permettent au site web d'envoyer des données d'authentification à des sites tiers. Même la taille d'écran du navigateur peut être utilisée pour tracer l'utilisateur. De plus, l'ensemble des entêtes HTTP peut être utilisé pour générer une signature sous forme de hash permettant de distinguer les entêtes d'utilisateurs personnalisés.

Vulnérabilités logicielles

En utilisant un navigateur ou toute autre application non mise à jour, un attaquant peut exploiter les vulnérabilités de ces applications pour obtenir des informations personnelles sur l'utilisateur et ce même s'il passe par un réseau anonyme. Même si le navigateur est à jour, il est toujours possible à l'attaquant d'exploiter des failles non corrigées par une mise à jour.

2.4 Moyens de défense contre la surveillance

2.4.1 Défense au niveau applicatif

Le navigateur web étant le principal vecteur d'attaque au niveau applicatif, il est important de le configurer correctement pour protéger sa vie privée. Il existe plusieurs extensions conçues pour se protéger des attaques énumérées plus haut.

TLS

L'extension HTTPS Everywhere[21], développée par l'EFF (Electronic Frontier Foundation) et le Projet Tor, permet d'utiliser le protocole HTTPS, sur les sites le supportant, au lieu du HTTP souvent fourni par défaut.

Bloqueur de publicité, d'anti-mouchard et de script

Il est recommandé d'ajouter un bloqueur de publicité comme Adblock Plus[22] ainsi qu'un anti-mouchard comme Ghostery[23] sur le navigateur pour éviter d'être tracé par de tierces parties.

uBlock Origin est le bloqueur utilisé sur le système d'exploitation Tails que nous verrons dans le chapitre sur TOR. Ce bloqueur est plus léger qu'Adblock Plus. La documentation de cette extension est disponible sur le GitHub de l'auteur [24].

L'extension NoScript[25] permet de bloquer l'exécution de tout code JavaScript, Java et autres plug-ins.

Protection contre le fingerprinting

Plus la configuration de l'utilisateur est rare, plus celui-ci est identifiable. Panopticlick[26] est un site, développé par l'EFF, qui permet de calculer l'entropie de votre navigateur. Il y a aussi IP check[27] développé par JonDoNym.

Il existe des navigateurs configurés pour préserver l'anonymat de leur utilisateur. Un des plus connus est TOR Browser[28], le navigateur permettant d'accéder au réseau TOR.

Autres protections

Pour se protéger des dernières attaques énumérées plus tôt, il faut désactiver l'utilisation des cookies sur le navigateur, configurer le navigateur pour qu'il n'utilise pas le Referer. Pour éviter toute connexion directe à la machine, l'idéal est d'utiliser un système d'exploitation qui par défaut bloque toutes connexions entrantes ou sortantes extérieures au réseau anonyme. Par exemple, pour le réseau TOR, le système d'exploitation Tails bloque tout trafic qui ne passe pas par TOR.

2.4.2 Les proxys

Se protéger au niveau applicatif permet d'éviter de diffuser des informations d'identification, mais ne suffit pas à être anonyme. Il est toujours possible d'identifier un utilisateur avec son adresse IP ou son trafic. Une solution populaire consiste à utiliser un intermédiaire nommé proxy pour faire transiter notre trafic. Ainsi, c'est l'adresse IP du proxy qui sera visible par les serveurs visités au lieu de celle de l'utilisateur.

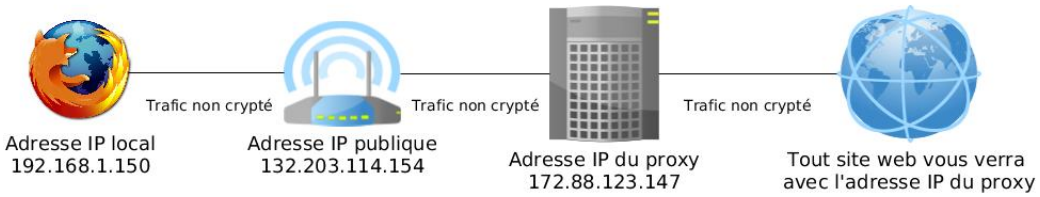

FIGURE 2.3 – Exemple de fonctionnement d'un proxy

Les deux protocoles les plus communs utilisés par les proxys sont :

1. Le HTTP : Ce type de proxy ne peut accepter uniquement le trafic web. La configuration se fait au niveau du navigateur web.
2. Le SOCKS : Les proxys supportant le SOCKS (Secured Over Credential-based Kerberos), contrairement au proxy HTTP, acceptent n'importe quel type de trafic (FTP ou bitorrent par exemple). Son seul inconvénient est qu'il a plus de traitement à effectuer que le proxy HTTP, il est donc plus lent.

Bien que relativement simple à mettre en œuvre, cette solution contient plusieurs inconvénients. La connexion entre l'ordinateur du client et du proxy n'est pas cryptée, il faut configurer l'accès au proxy sur chaque application accédant à Internet et rien n'empêche au proxy de conserver des traces de notre trafic dans ses journaux d'événements (logs). Les proxys sont donc principalement utilisés pour contourner les blocages basés sur la géolocalisation ou sur l'IP que pour assurer un réel anonymat.

Une autre solution pour masquer son adresse IP est le VPN (Virtual Private Network). Le principe est similaire à celui d'un proxy, notre trafic Internet est relayé par un serveur à la différence que la connexion entre l'ordinateur du client et le serveur VPN est cryptée et que tout le trafic entrant et sortant de l'ordinateur du client passe par le serveur.

L'utilisation d'un VPN assure un meilleur niveau d'anonymat et de protection des données, mais tout comme les proxys, rien ne garantit au client que le VPN ne conserve aucune trace de son trafic. Un des intérêts du VPN est qu'il est accessible partout sur Internet donc si le client est dans un réseau non sûr comme un aéroport ou un cybercafé, il peut accéder à Internet sans crainte d'être écouté sur ce réseau.

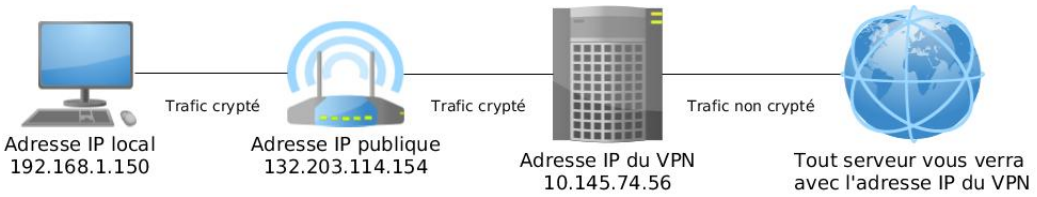

FIGURE 2.4 – Exemple de fonctionnement d'un VPN

2.4.3 Les réseaux anonymes

Les réseaux anonymes sont spécifiquement conçus pour assurer l'anonymat de leurs utilisateurs. Étant plus complexe que les autres solutions abordées, la suite de ce mémoire leur sera consacrée. Les réseaux anonymes les plus connus aujourd'hui sont TOR[29] (The Onion Router) et I2P[30] (Invisible Internet Project).

2.5 Conclusion

Dans ce chapitre, nous avons vu différentes techniques pour surveiller les internautes, soit le "Deep Packet Inspection", le profilage DNS et les attaques applicatives sur les navigateurs web. Ensuite, nous avons vu plusieurs technologies de défense, soit l'utilisation de plug-ins pour les navigateurs web, les proxys et les réseaux anonymes. Chacune permet d'obtenir un niveau d'anonymat différent. Les réseaux anonymes étant ceux permettant d'accéder au plus haut niveau d'anonymat. Ils sont le sujet du prochain chapitre.

Chapitre 3

Principes généraux des réseaux anonymes

3.1 Introduction

Ce chapitre vise à définir ce qu'est un réseau anonyme. La grande majorité des réseaux anonymes sont des mix networks. Ils sont constitués de relais de proxy et emploient le chiffrement en couche. Nous verrons leur caractéristique et le modèle de menaces qui leur sont associées. Ensuite, le mix network de Chaum sera étudié. Pour finir, nous verrons les différents algorithmes de sélections des proxys, nommées mixes ainsi que les stratégies d'éviction.

3.2 Définition et propriétés

3.2.1 Relais de proxy et couches de chiffrement

Le premier réseau anonyme qui a été créé est le mix network de Chaum[31] en 1981, celui-ci est étudié en détail plus loin dans ce chapitre. La grande majorité des réseaux anonymes actuels dans ce mémoire suivent un fonctionnement commun. Un relais de proxy est constitué dans le réseau et le trafic est chiffré avec la clé publique de chaque proxy de sorte à former plusieurs couches de chiffrement. Ainsi, seul le premier proxy pourra déchiffrer la première couche. Il transmettra ensuite le message au proxy suivant qui déchiffrera une couche à son tour pour le transmettre au proxy suivant jusqu'à ce que le message arrive à destination. Le but étant de rendre les communications difficiles à tracer.

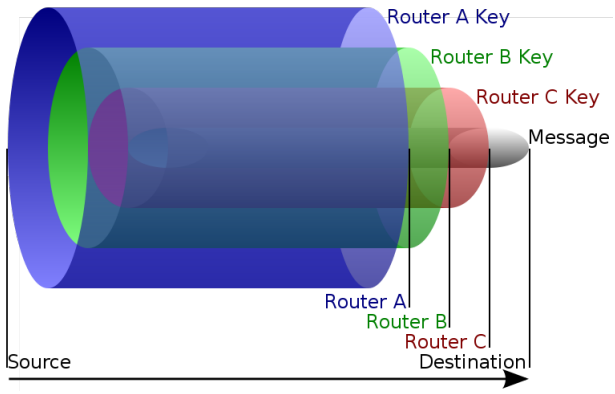

FIGURE 3.1 – Chiffrement en couche d'un message

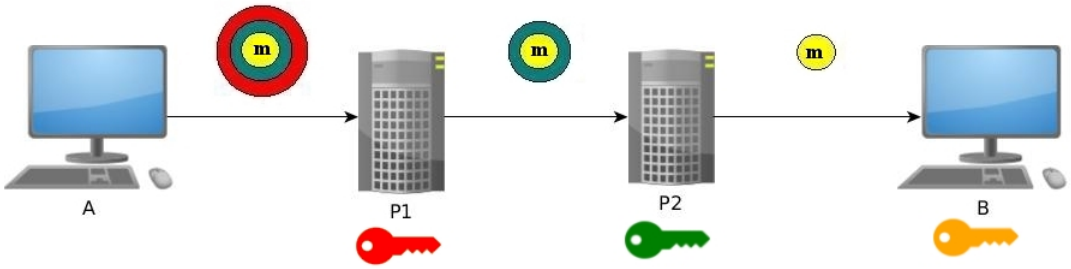

FIGURE 3.2 – Transport du message dans le relais

3.2.2 Propriétés

[32] énumère les propriétés que doivent satisfaire les mix networks et les divise en trois catégories : la sécurité, la performance et l'implémentation. Un mix network consiste en un relais de proxy. Chaque proxy est nommé "mixe". Chaque mixe transforme de manière cryptographique les messages qu'il reçoit et les transmet au prochain mixe, et ceci jusqu'aux destinataires finaux. Les mix networks sont employés principalement pour le vote électronique et l'envoi de courriel anonyme. Ses propriétés s'appliquent aussi pour les réseaux anonymes TOR et I2P étant eux aussi des mix networks.

Pour garantir la sécurité de ses utilisateurs, le mix network doit satisfaire les propriétés suivantes :

1. Anonymat : L'entrée d'un message et sa sortie dans un mixe ne doivent pas être corrélées. Le mix network doit assurer l'intraçabilité entre l'émetteur et le récepteur.
2. Intégrité : Les messages circulant dans les mix networks ne doivent pas être corrompus.
3. Exactitude : La sortie d'un mixe doit correspondre à la transformation du message reçu en entrée.

4. Vérifiabilité : Le mix network doit pouvoir vérifier l'exactitude des messages à la sortie des mixes.
5. Robustesse : Si un ou plusieurs mixes sont défaillants, le mix network doit continuer de fonctionner correctement. De plus, le mix network doit pouvoir se protéger contre des attaquants passifs ou actifs.

La performance d'un mix network est évaluée sur deux propriétés :

1. Latence : Les transformations des mixes, le transfert des messages entre chacun d'eux ainsi que les mécanismes de vérifiabilité et de robustesse augmentent la latence du réseau. Celle-ci doit satisfaire au besoin des applications. Par exemple, une application temps-réel doit avoir une faible latence.
2. Débit : Il correspond au nombre de messages réel qu'un mix network peut produire en un temps donné permettant d'avoir une estimation de la surcharge du mix network. La surcharge correspond au trafic utilisé pour le bourrage ou les faux messages pour renforcer l'intraçabilité du réseau.

Tout en assurant les contraintes de sécurité et de performance, le mix network doit rester implémentable. Cela prend en compte deux propriétés :

1. Évolutivité : Le niveau d'anonymat augmente avec l'accroissement du nombre d'utilisateurs et le nombre de mixes dans un relais, mais l'évolutivité du réseau doit respecter les contraintes de latence.
2. Efficacité : Le protocole du mix network doit réduire la complexité des calculs et des communications.

3.3 Caractéristiques des réseaux anonymes

La façon dont sont routées les données au sein d'un réseau anonyme détermine leur niveau de sécurité et de performance. [33] classifie les caractéristiques des protocoles de routage des communications anonymes en trois groupes de catégories :

— La structure du réseau
— Les informations de routage
— Le modèle de communication

3.3.1 Structure du réseau

La structure d'un réseau anonyme consiste en trois caractéristiques : la topologie, le type de connexion et la symétrie.

Topologie

La topologie correspond à l'architecture du réseau définissant les liaisons de ses nœuds. Dans le cas des communications anonymes, seule la topologie logique est prise en compte, c'est-à-dire

la façon dont les données transitent au sein du réseau. Les réseaux anonymes peuvent être classifiés selon leur connectivité :

— Entièrement connecté : Les nœuds du réseau peuvent se connecter à au moins 95% des autres nœuds.

— Principalement connecté : Les nœuds du réseau peuvent se connecter à au moins la moitié des autres nœuds.

— Partiellement connecté : Les nœuds du réseau ne peuvent se connecter qu'à un nombre restreint d'autres nœuds.

Une connectivité élevée permet une meilleure résilience du réseau, mais éliminer les connexions posant des problèmes de sécurité ou ayant une forte latence, permet d'améliorer l'anonymat et la performance de la communication.

Type de connexion

Un type de connexion implique sa direction et sa synchronisation.

Une connexion est soit unidirectionnelle, soit bidirectionnelle. Les connexions unidirectionnelles ne permettent à un attaquant d'observer les données que dans un seul sens. Néanmoins, les connexions bidirectionnelles sont moins coûteuses lors de la construction du circuit.

La synchronisation d'une connexion est soit asynchrone, soit synchrone, c'est-à-dire si la communication est décalée dans le temps ou pas. Les connexions asynchrones imposent le moins de contraintes, autant pour la conception que pour l'utilisation, mais les connexions synchrones offrent un meilleur niveau d'anonymat en protégeant les communications de certaines attaques temporelles telles que la corrélation basée sur les temps de début et de fin de communications.

Symétrie

Un protocole de communication anonyme est symétrique quand tous les nœuds du réseau ont des rôles et des responsabilités similaires. Il existe trois dimensions de la symétrie : le mode de communication, la structure hiérarchique et le degré de centralité.

Il peut y avoir trois modes de communication différents :

— Le mode P2P (peer-to-peer) : Un réseau est qualifié de P2P lorsque ses nœuds ont un rôle équivalent. Les utilisateurs finaux doivent opérer en tant que nœud relais.

— Le mode client-serveur : Les nœuds clients exploitent les services fournis par les nœuds serveurs. Les utilisateurs finaux ne peuvent pas opérer en tant que nœud relais.

— Le mode hybride : Les utilisateurs finaux peuvent être ou ne pas être des nœuds relais.

Plus le nombre de nœuds dans un réseau P2P est grand, plus sa capacité augmente. De plus, ce type de réseau est plus robuste et disponible. Tandis que dans une architecture client-serveur,

les serveurs sont souvent plus fiables et les clients ont besoin de moins de ressources pour utiliser le réseau.

La structure d'un réseau est dite plate lorsque tous les nœuds ont le même niveau d'importance dans la prise de décision de routage et est hiérarchique dans le cas contraire. Les structures hiérarchiques sont plus performantes, mais moins robustes que les structures plates. Si un nœud placé en haut de la hiérarchie ne fonctionne plus, la performance du réseau peut être grandement affectée.

Le degré de décentralisation correspond au nombre de nœuds offrant un service de routage essentiel au fonctionnement du réseau. Les réseaux semi-centralisés ont un nombre restreint de nœuds fournissant ce genre de service. Un haut niveau de confiance est placé sur ces nœuds, mais rend le réseau moins robuste. Les réseaux complètement décentralisés n'ont pas ce type de nœud. Le niveau de confiance est le même pour tous les éléments du réseau. Les réseaux centralisés ne sont pas pris en compte dans le cas des communications anonymes étant donné qu'un seul point garantit leur fonctionnement. Cela faciliterait la surveillance des communications et rendrait le réseau non sécuritaire.

3.3.2 Informations de routage

Les informations de routage permettent les prises de décision de routage. Elles consistent en deux points : La vue du réseau et la fréquence des mises à jour de ces informations.

Vue du réseau

La vue du réseau détermine la portée des informations disponibles. Elle est complète lorsque les informations de routage de tous les nœuds sont disponibles. Elle est partielle dans le cas où seule l'information d'un groupe de nœud l'est. Une vue complète permet de prendre une décision sur l'ensemble des nœuds du réseau mais en contrepartie la consommation de la bande-passante et des ressources est beaucoup plus grande.

Fréquence des mises à jour

La fréquence des mises à jour des informations de routage peut être périodique ou basée sur des événements. L'absence de mise à jour est aussi possible.

3.3.3 Modèle de communication

Le modèle de communication regroupe les caractéristiques concernant la création des routes dans le réseau. Ils sont au nombre de trois : le type de routage, la planification et la sélection de nœuds.

Type de routage

Le type de routage est la façon dont sont sélectionnés les nœuds pour constituer la route sur le réseau. Dans une communication anonyme, deux types de routage sont utilisés : le "source-routing" où l'utilisateur choisit les nœuds pour atteindre son destinataire et le "hop-by-hop" où l'utilisateur ne choisit que le premier nœud du relais et c'est ensuite celui-ci qui choisit le deuxième nœud et ainsi de suite jusqu'au destinataire. Le "source-routing" permet à l'expéditeur de choisir des nœuds de confiance tandis qu'avec le "hop-by-hop", l'utilisateur ne connaît pas tous les nœuds du relais, il est donc plus difficile pour un attaquant de reconnaître un utilisateur via son choix de route.

Planification

La planification consiste au traitement des connexions entrantes d'un nœud. Elle peut être équitable, tous les types de nœuds sont traités de la même manière ou priorisés, certaines connexions ont une plus grande priorité que d'autres. La planification priorisée prévient les congestions et augmente la performance du réseau, mais rend le trafic plus distinguable.

Sélection de nœuds

Le nombre de nœuds qui peut être sélectionné dans une route peut être fixé ou déterminé par probabilité. La sélection peut se faire à partir de tous les nœuds ou selon certaines restrictions basées sur la sécurité, le réseau ou les préférences de l'utilisateur. La distribution de probabilité pour la sélection de nœuds peut être uniforme ou pondérée par certains paramètres comme la bande passante ou le temps de réponse.

La sélection de nœuds est une caractéristique primordiale des réseaux anonymes et une sous-section lui est consacrée plus loin dans ce chapitre.

3.3.4 Évaluation de la performance

La latence et le mode de communication du protocole sont deux critères qui permettent d'évaluer la performance d'un réseau de communications anonymes. Il peut y avoir deux modes, le mode connecté et le mode non connecté.

Ci-dessus, voici la table basée sur [33] regroupant la liste de toutes les caractéristiques :

	Nom		Valeurs
Structure réseau	Topologie		Entièrement, Principalement ou Partiellement
	Type de connexion	Direction	Unidirectionnel ou Bidirectionnel
		Synchronisation	Synchrone ou Asynchrone
	Symétrie	Rôles	P2P, Client-serveur ou Hybride
		Hiérarchie	Plate ou Hiérarchique
		Décentralisation	Semi-décentralisé ou Complètement décentralisé
Info. de routage	Vue du réseau		Complète ou Partielle
	Mise à jour		Périodique ou basée sur les Événements
Mode de communication	Type de routage		Source-routing ou Hop-by-hop
	Planification		Équitable ou Priorisée
	Sélection de nœud	Déterminisme	Fixé ou par Probabilité
		Ensemble	Tous ou Restreint
		Distribution	Uniforme ou Pondérée

TABLE 3.1 – Caractéristiques des protocoles de routage des communications anonymes

3.4 Modèle de menace

Aucun réseau ne peut fournir un anonymat complètement sûr. Les réseaux anonymes mettent en place plusieurs techniques pour rendre les attaques d'un adversaire plus coûteuses. [20] et [34] énumèrent différents types d'attaques sur ces réseaux.

3.4.1 Attaques par intersection

Une attaque par intersection consiste à analyser le trafic d'un utilisateur de manière passive. À partir de cette analyse, l'attaquant peut dans le temps réduire l'ensemble des utilisateurs qui sont suspectés de communiquer avec l'utilisateur pour possiblement en isoler un.

Plusieurs attaques peuvent être catégorisées comme des attaques par intersection :

Attaque du prédécesseur

L'attaque consiste à deviner l'initiateur d'une communication. Un ensemble de nœuds travaillent ensemble pour surveiller tout initiateur potentiel d'un flux de communications. [35] prouve qu'avec suffisamment de reconstruction de relais, le nœud compromis tend à se connecter plus souvent avec l'initiateur du flux de communication qu'avec les autres nœuds, augmentant les chances d'une analyse de trafic réussi.

Attaque par comptage de paquets

L'attaquant va surveiller les communications entre deux nœuds dans le but de compter les paquets entrants et sortants de chaque nœud. Ainsi, l'attaquant pourrait identifier un lien de communication entre les deux nœuds.

Contre-mesures

L'inconvénient majeur des attaques par intersection est qu'elles demandent une longue écoute du trafic pour réussir. Par conséquent, ce type d'attaque ne peut fonctionner avec des sessions courtes de l'utilisateur.

De plus, plus le réseau est grand, plus le profilage d'un utilisateur est difficile. Un attaquant contrôlant k nœuds sur les n nœuds d'un réseau anonyme a une probabilité de k/n d'observer le nœud d'entrée d'un relais et $(k-1)/(n-1)$ d'observer le nœud de sortie du relais, soit une probabilité de $(k^2-k)/(n^2-n)$ d'observer à la fois le nœud d'entrée et le nœud de sortie d'un même relais.

La plupart des réseaux anonymes reconstruisent les relais au bout d'un certain temps limitant le succès des attaques par intersection. Contre les attaques de comptage de paquets, les réseaux peuvent ajouter des données aléatoires dans la session de communication. Et pour contrer les attaques du prédécesseur, le réseau anonyme doit choisir les nœuds du relais non aléatoirement.

3.4.2 Attaques par étiquetage

L'attaque par étiquetage est une attaque active. Elle consiste à modifier ou ajouter des paquets dans le relais (des cellules pour TOR par exemple) d'un utilisateur dans le but de reconnaître le trafic plus loin dans le relais. Pour réaliser cela, l'attaquant doit maintenir une présence sur plus d'un nœud du circuit d'un utilisateur. L'anonymat d'un utilisateur pourrait être compromis si l'utilisateur sélectionne un nœud d'entrée et un nœud de sortie sous contrôle de l'attaquant. Ainsi, celui-ci pourrait envoyer ou modifier les paquets d'un nœud et observer les changements sur l'autre nœud pour reconnaître le lien de communication entre l'utilisateur et son destinataire.

Il existe quatre méthodes pour effectuer cette attaque :

— La méthode par renvoi : Elle consiste à dupliquer un paquet qui a déjà été envoyé. Ce paquet provoquera une erreur de décryptage qui va se propager jusqu'au nœud de sortie et permettra de faire le lien vers la source.

— La méthode par suppression : Elle consiste à supprimer le premier d'un flux de communication tandis que les autres seront relayés. L'erreur provoquée par l'absence du premier paquet sera transmise jusqu'au relais de sortie.

— La méthode par modification : L'attaquant modifie un paquet, par exemple en mettant à zéro le premier octet du contenu crypté. Comme pour les méthodes précédentes, cela provoquera une erreur qui sera relayée jusqu'au nœud de sortie.

— La méthode par injection : Comparable à la méthode par modification, elle consiste à ajouter des données aléatoires dans le paquet.

L'attaquant doit avoir une présence suffisamment importante pour pouvoir contrôler à la fois le nœud d'entrée et le nœud de sortie d'un même circuit. La probabilité que cela se produise est la même que pour une attaque par intersection soit $(k^2 - k)/(n^2 - n)$ avec k le nombre de nœuds contrôlés par l'attaquant et n l'ensemble des nœuds sur le réseau anonyme.

FIGURE 3.3 – Attaque par étiquetage

3.4.3 Attaques par fingerprinting

L'attaquant collecte le trafic de sa cible et le compare avec sa base de données de signatures de trafic. Si la cible visite un site web, cela va requérir le même nombre de paquets pour visiter ce site. Si l'attaquant possède la signature contenant ces paquets, il pourra déterminer que sa cible visite ce site web. Bien sûr, le contenu du site web ne doit pas changer entre-temps et l'attaquant doit contrôler le nœud d'entrée du relais de sa cible.

Contrairement aux autres attaques décrites plus haut, l'attaquant a une probabilité de k/n de contrôler le nœud d'entrée de sa cible avec k le nombre de nœud contrôlé par l'attaquant et n l'ensemble des nœuds sur le réseau anonyme. Néanmoins, avec la multiplication des pages web dynamiques, l'efficacité de l'attaque reste limitée sauf si l'attaquant fait en sorte que l'utilisateur visite son propre site web. Il existe plusieurs mécanismes de protection comme l'ajout de données aléatoires dans les paquets.

3.4.4 Attaques par congestion

Cette attaque consiste à surcharger certains nœuds dans le but d'identifier le chemin complet d'un relais. Les nœuds étant affectés par cette surcharge verront leur bande passante et leur

capacité de transfert diminuée par rapport aux autres nœuds. Un utilisateur passant par ces nœuds verra son trafic ralentir et l'attaquant pourra identifier son chemin de proxy.

L'attaquant cible les nœuds qu'il soupçonne d'appartenir au relais d'un utilisateur et il a besoin d'observer le trafic des autres nœuds pour pouvoir comparer leur débit. Les réseaux anonymes modernes changent de relais au bout d'un certain temps et avec un nombre de nœuds importants sur le réseau, le succès de ce genre d'attaque est limité.

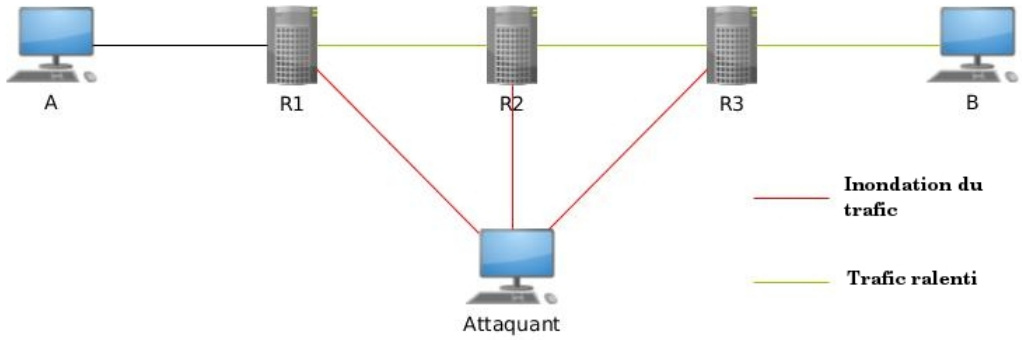

FIGURE 3.4 – Attaque par congestion

3.4.5 Attaques par les ressources

Cette attaque fait en sorte que l'utilisateur sélectionne des nœuds contrôlés par l'attaquant pour construire son relais. L'attaquant peut exploiter les critères de sélection du réseau anonyme pour que les nœuds qu'il contrôle aient une plus forte probabilité d'être sélectionnés que les autres. Par exemple, il peut le faire en fournissant de fausses informations sur les performances de ses nœuds. L'attaquant pourrait aussi augmenter le nombre de nœuds qu'il contrôle bien que cela augmenterait les chances d'être détecté par le réseau anonyme si celui-ci dispose de mécanisme de détection. Si les nœuds d'entrée et de sortie sélectionnés par l'utilisateur sont contrôlés par l'attaquant, la mise en œuvre des autres attaques vue précédemment sera grandement facilitée.

3.4.6 Attaques par déni de service

L'attaque par déni de service ou DoS (Denial of Service) en anglais n'est pas propre au réseau anonyme. Son but est d'empêcher l'accès à un service. Les utilisateurs n'ayant plus accès au réseau anonyme seront poussés à employer d'autres moyens moins sécuritaires pour assurer les communications. L'attaque DoS peut être divisée en deux catégories.

Blocage total

Tout le réseau est bloqué. Cela peut arriver si un gouvernement bloque l'accès au réseau anonyme ciblé.

Attaque ciblée

L'attaque cible un ou plusieurs nœuds essentiels au bon fonctionnement du réseau anonyme. Elle peut s'effectuer grâce à une attaque DDoS à partir d'un botnet de l'attaquant. Un botnet est un réseau de machines contrôlées par l'attaquant. Celui-ci peut faire en sorte que toutes ces machines envoient des paquets sur un nœud du réseau anonyme dans le but de le surcharger et de le rendre indisponible. Ce genre d'attaque est en général très visible et donc facilement détectable.

3.4.7 Longueur du relais

TOR et I2P dans [36] et [37] recommande d'utiliser des relais de trois nœuds pour assurer un anonymat optimal. La plupart des attaques se concentrent sur le contrôle des nœuds d'entrée et de sortie du relais, il est donc inutile d'avoir plus d'un nœud intermédiaire.

3.5 Mix Network de Chaum

Le mix network de Chaum[31] est le premier réseau de communication anonyme qui a été créé. L'utilisateur chiffre son message avec chacune des clefs publiques des nœuds du relais. Ensuite, chaque mixe décrypte une couche de chiffrement et relaie le message jusqu'au destinataire.

Prenons l'exemple suivant, A veut communiquer avec B anonymement. Il passe par les mixes M1 et M2 et a les adresses de chaque nœud du relais (@B, @M1 et @M2) ainsi que leurs clefs publiques (PK_B, PK_{M1} et PK_{M2}). A chiffre son message avec PK_B. Supposons que le contenu du message soit "Bonjour" et qu'un attaquant écoute les communications de B et capture le message chiffré. Si cet attaquant chiffre un message avec PK_B ayant le même contenu que le message de A, soit dans notre exemple "Bonjour", alors le message chiffré sera identique au message capturé. Pour éviter qu'un attaquant puisse deviner le contenu des messages, on rajoute une chaîne de caractères aléatoire R0 au message qui va agir comme un sel.

La première couche de chiffrement ressemble donc à ceci :

$$\mathbf{C0 = PK_B(R0, "Bonjour")}$$

Chaque couche de chiffrement possédera une chaîne de caractère aléatoire, on a donc pour les prochaines couches :

C1 = $\mathbf{PK_{M1}}$(R1, C0, @B)
C2 = $\mathbf{PK_{M2}}$(R2, C1, @M1)

A envoie C2 à M2 qui le déchiffre et obtient C1 qu'il transmet à M1 qui le déchiffre à son tour avec sa clef privée et obtient C0 qu'il transmet à B qui déchiffre la dernière couche de chiffrement pour obtenir le message. À chaque étape de décryptage, les chaînes de caractères aléatoires sont rejetées.

Pour que B puisse répondre à A sans connaître son adresse, A va ajouter à son message une clef publique PK_X à utilisation unique qu'il aura générée ainsi qu'une adresse de retour constituée comme ceci :

D2 = $\mathbf{PK_{M2}}$(S2, @A)
D1 = $\mathbf{PK_{M1}}$(S1, @M2, D2)
où S1 et S2 sont des clés qui font aussi office de chaînes de caractères aléatoires.

Pour répondre à A, B chiffre sa réponse à l'aide de PK_X et l'envoie à M1 avec D1 :

D1, $\mathbf{PK_X}$(S0, réponse)

M1 chiffre la réponse avec S1 et envoie le tout à M2 avec D2 :

D2, S1($\mathbf{PK_X}$(S0, réponse))

M2 fait de même avec S2 et transfère à A :

D2, S2(S1($\mathbf{PK_X}$(S0, réponse)))

A ayant généré S1, S2 et PK_X, il est le seul à pouvoir déchiffrer la réponse.

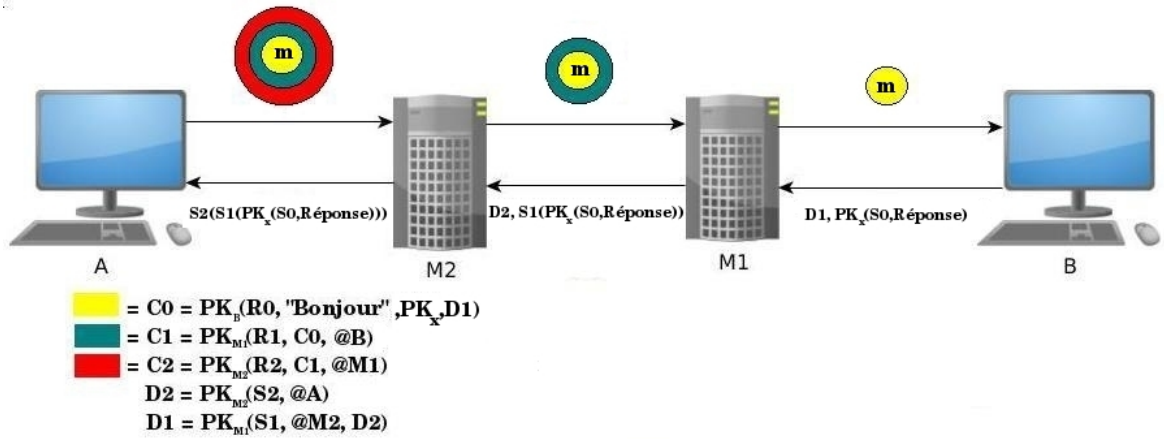

FIGURE 3.5 – Fonctionnement du mix network de Chaum

3.5.1 Mix Network hybride

Le mix network hybride décrit dans [32] utilise à la fois le chiffrement asymétrique et le chiffrement symétrique. Le message est chiffré, pour chaque couche, avec une clé symétrique au lieu d'une clé publique. Seule la clé symétrique est chiffrée par la clé publique. En reprenant notre exemple précédent, A enverrait :

$\mathbf{C0 = PK_B(P_B),\ P_B(R0,\ message)}$
$\mathbf{C1 = PK_{M1}(P_{M1}),\ P_{M1}(R1,\ C0,\ @B)}$
$\mathbf{C2 = PK_{M2}(P_{M2}),\ P_{M2}(R2,\ C1,\ @M1)}$

Étant donné que la taille des données à chiffrer augmente à chaque couche et que le chiffrement asymétrique est plus lent que le chiffrement symétrique, le chiffrement asymétrique n'est utilisé que pour chiffrer les clés symétriques. Ainsi, la taille des données à chiffrer avec les clés publiques pour chaque couche reste identique réduisant le temps de calcul du chiffrement et augmentant l'efficacité du mix network.

3.6 Sélection des mixes

L'échange de message via un relais de mixes prédéterminés que propose le mix network de Chaum est nommé "mix cascade". Une des faiblesses de cette topologie est son manque de robustesse. Si un mixe du relais ne fonctionne pas, la procédure doit être entièrement recommencée. Pour combler cette lacune, une autre topologie a été créée, le "free-route". La chaîne de mixes n'est pas fixée et les nœuds sont interconnectés offrant plusieurs chemins différents à l'émetteur pour joindre son destinataire.

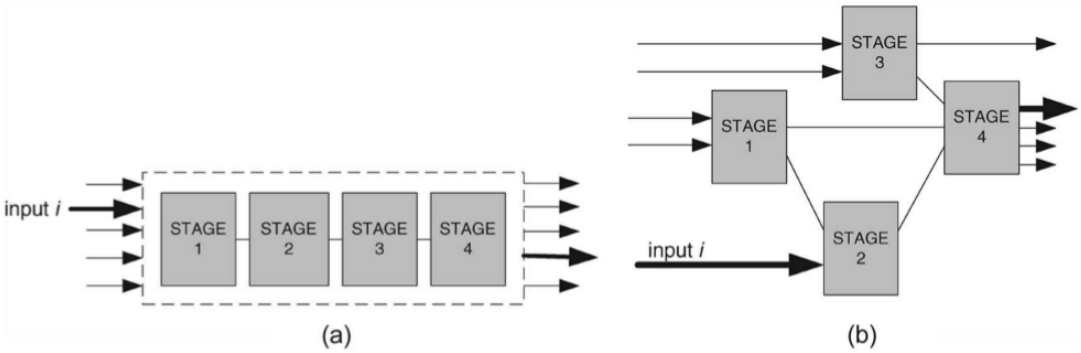

FIGURE 3.6 – Topologies des mix networks : a) "mix cascade" b) "free-route"

La figure 3.6 est tirée de [32]. a) représente la topologie en "mix cascade". Les cinq entrées passent par les mêmes mixes et dans l'ordre préétabli. b) représente la topologie en "free-route". Les cinq entrées peuvent passer par n'importe quel mixe et peuvent emprunter n'importe quel chemin pour se rendre à destination.

Il existe différentes stratégies de sélection de nœuds. Elle peut-être déterministe, aléatoire ou basée sur d'autres critères tels que l'état du réseau ou la réputation des nœuds. Par exemple, le mix network de [38] utilise un système de réputation basé sur un groupe de témoins qui permet d'évaluer la fiabilité des nœuds. Un mixe M qui transfère un message attend de recevoir un reçu du prochain mixe M+1. Si celui-ci ne répond pas à un temps donné, M envoie au groupe de témoins (ou à une partie d'entre eux) le message et chacun d'eux teste M+1. Si un témoin reçoit le reçu, il est retransmis à M sinon le témoin conclut que M+1 n'est pas fiable et informe M. La topologie du mix network est le "free-route", le choix de la route s'effectue selon le niveau de réputation des mixes, ceux n'ayant aucun point sont ignorés. La faiblesse de ce mix network est qu'il est nécessaire de faire confiance aux témoins, ceux-ci ne doivent pas transmettre de fausses informations.

3.7 Stratégie d'éviction

Les mixes peuvent employer différentes stratégies d'éviction (flushing en anglais) pour contrer certaines attaques énumérées dans la section Modèle de menace, principalement les attaques par intersection. Une stratégie d'éviction définit à quel moment les messages reçus par un mixe devront être transférés.

[33] décrit trois types d'algorithmes :

1. Le premier consiste à atteindre un seuil de messages reçus avant de les transférer. Le désavantage de cette méthode est qu'elle augmente la latence du mix network.
2. La seconde randomise le délai de transfert de chaque message. Cette stratégie ne protège pas d'une attaque par corrélation dans le cas où le trafic du réseau est faible. Il n'est d'aucune utilité si un mixe ne reçoit qu'un message après que le délai de transfert soit passé.
3. La dernière se nomme "pool mixe". Les messages reçus sont transférés par lot. Le mixe garde en mémoire le reste des messages pour les tours d'éviction suivants. Le nombre de messages à envoyer peut être déterminé de façon déterministe ou non déterministe et la sélection des messages peut être aléatoire ou pondérée en fonction des conditions du réseau. Le pool mixe peut combiner les autres algorithmes vus précédemment et ainsi adapter sa stratégie d'éviction suivant l'état du mix network.

3.8 Conclusion

Nous avons vu comment sont construits la plupart des réseaux anonymes ainsi que leurs caractéristiques. Le modèle de mix network consiste à l'utilisation de relais de proxy et du chiffrement en couches. À partir de là, un modèle de menaces peut être établi. Six catégories ont été énumérées. Nous avons vu différentes topologies de mix networks, les deux principales sont

le "mix cascade" et le "free-route". Ensuite, différentes stratégies d'éviction ont été énumérées. Ces stratégies permettent de contrer les attaques par intersection.

Chapitre 4

Études de réseaux anonymes

4.1 Introduction

Ce chapitre va présenter en détails les réseaux anonymes les plus populaires. Nous allons commencer avec le webmixe JAP. Ensuite, nous allons voir les différents types de remailer ainsi que les deux réseaux anonymes les plus utilisés : TOR et I2P. Pour finir, une section sera consacrée à d'autres réseaux anonymes moins connus.

4.2 JAP

JAP (Java Anon Proxy), aussi nommé JonDonym, est un webmixe. Il est conçu pour la navigation en temps réel sur internet. Il est issu du projet AN.ON[39] et commercialisé par JonDos GmbH[40]. Son fonctionnement est décrit en détail dans [41]. Sa topologie est en cascade et il n'a pas de stratégie d'éviction. C'est un mix network hybride. La connexion est bidirectionnelle. L'architecture de ce réseau peut-être divisée en quatre parties.

4.2.1 Programme client

Le programme client est codé en Java. Il agit en tant que proxy local sur la machine du client. Son interface permet de configurer et d'obtenir des informations au sujet de la connexion.

Sur la figure 4.1, 1. permet à l'utilisateur de choisir son relais de mixes, 2. fournit des informations sur les relais sélectionnés, 3. est l'Anonym-O-Meter qui évalue le niveau d'anonymat de la connexion. Celui-ci se base sur le nombre d'utilisateurs actifs et la répartition géographique des mixes. Plus le nombre d'utilisateurs est important et plus les mixes sont répartis sur différent endroit du globe, plus le niveau d'anonymat est élevé. En cliquant sur le bouton "Configuration", on peut accéder à plus de détail sur les relais de mixes disponibles, tels que le nombre de mixes dans le relais, leur localisation ainsi que leur opérateur comme illustré dans la figure 4.2.

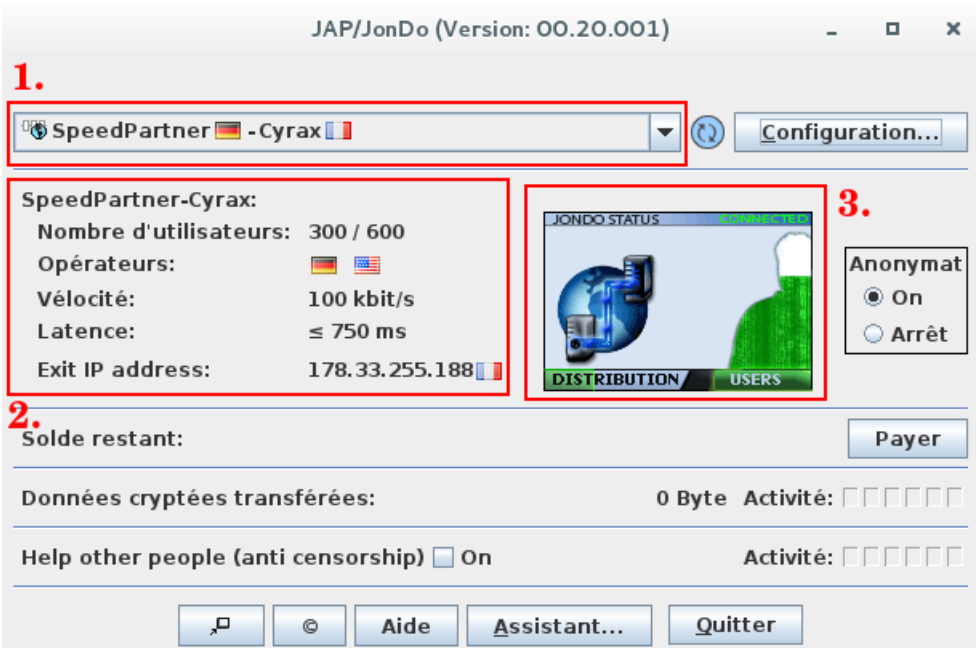

FIGURE 4.1 – Interface utilisateur de JAP

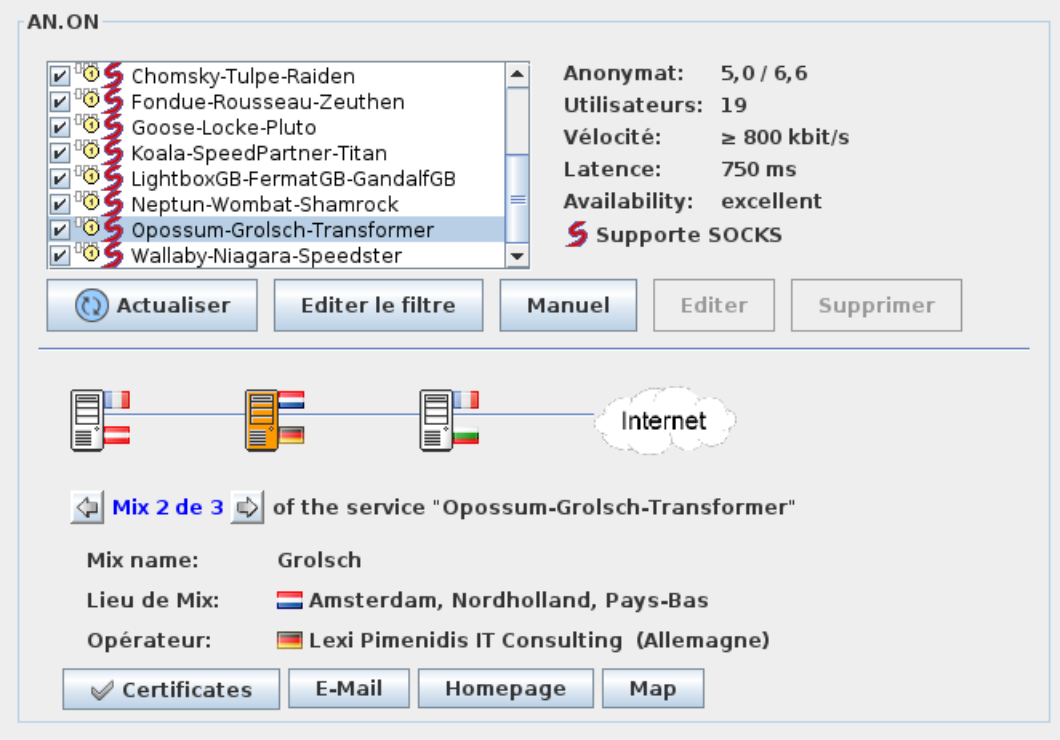

FIGURE 4.2 – Configuration des relais de mixes dans JAP

La version premium payante donne l'accès à tous les relais du réseau qui en général offre un meilleur niveau d'anonymat et de meilleures performances. Ces relais supportent le protocole SOCKS. Pour se protéger des attaques applicatives, JonDos GmbH fournit JonDoFox. C'est un navigateur basé sur Firefox, il est préconfiguré pour laisser le minimum de traces. Il est possible de tester l'anonymat de son navigateur grâce à l'outil IP check[27].

4.2.2 Relais de mixes

Le nombre de mixes varie d'un à trois sur tous les relais disponibles. Chaque mixe possède un opérateur. Ils sont à charge de maintenir le bon fonctionnement du mixe. Leur identité est vérifiée via un certificat SSL, l'autorité de certification étant JonDos GmbH ou AN.ON.

4.2.3 Serveur d'information

Les informations disponibles sur le programme client proviennent du serveur d'information. Il communique avec tous les nœuds du réseau (clients, mixes et opérateurs). Il fournit les clés publiques de chaque mixe et des informations sur le trafic (nombre d'utilisateurs, latence, localisation des mixes, etc.).

4.2.4 Proxy-caches

Le dernier mixe du relais envoie les données déchiffrées à un proxy-cache. Celui-ci communique directement avec les serveurs sur internet et renvoie leurs réponses à l'utilisateur dans le relais en sens inverse. Il conserve dans son cache les pages et le contenu demandé dans le but de réduire le nombre de requêtes et d'améliorer la performance du réseau.

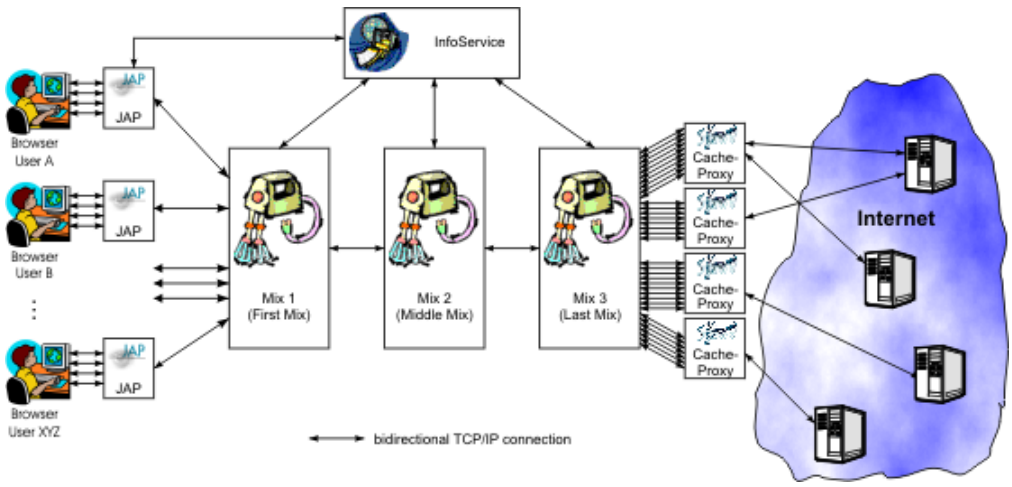

FIGURE 4.3 – Architecture de JAP [39]

4.2.5 Mécanisme de défense

Attaques par intersection

JAP utilise un algorithme "chop-and-slice". Il consiste à découper en morceaux de même taille les grands messages. Ces morceaux sont nommés "slice". Chaque slice est envoyé via un canal du mixe. Tout utilisateur actif n'émettant pas de message en envoie des fictifs. Ainsi, un trafic est constamment maintenu dans le réseau et avec le chiffrement, il n'est pas possible pour un attaquant de différencier les messages réels des faux. De plus, les mixes et les cache-proxy envoient eux aussi des messages fictifs aux utilisateurs ne recevant pas de données. Cela permet d'éviter qu'un attaquant puisse reconnaître un utilisateur qui envoie uniquement des messages fictifs. Cet algorithme rend le trafic plus uniforme et il est donc plus difficile pour un attaquant de distinguer le trafic d'un seul utilisateur. Les slices de chaque utilisateur actifs sont envoyés et se terminent en même temps empêchant toute corrélation basée sur le temps. En pratique, il est difficile d'assurer que chaque utilisateur envoie exactement autant de données que les autres, étant donné la différence de connexion que peuvent avoir les utilisateurs (vitesse, qualité, bande passante, etc.). Si un utilisateur a une mauvaise connexion, les autres utilisateurs devraient attendre que celui-ci ait envoyé autant de données que tout le monde. Cela nuirait à la qualité de service du réseau.

Les mixes des relais sont fixes et ne sont donc pas choisis aléatoirement. JAP est donc protégé contre les attaques du prédécesseur et avec l'envoi de messages fictifs dans le réseau, une attaque par comptage de paquets est plus difficile.

Les mixes d'entrée et de sortie de chaque relais sont publics. Un attaquant connaît donc d'avance quel nœud écouter. Ensuite, un attaquant peut simuler plusieurs utilisateurs en exécutant plusieurs programmes clients permettant de tromper les autres utilisateurs sur leur niveau d'anonymat. Néanmoins, plus le nombre d'utilisateurs est grand, plus il sera difficile et coûteux pour un attaquant de briser l'anonymat d'un utilisateur.

Attaques par étiquetage

Pour réaliser une attaque par étiquetage, il faut que l'attaquant contrôle les mixes du relais. Étant donné que chaque mixe est géré par un opérateur de confiance, ce type d'attaque est difficile à mettre en place.

L'activité des utilisateurs peut être révélée si tout les mixes d'un relais sont contrôlés par un attaquant et que ceux-ci conservent leurs fichiers de logs. Néanmoins, tous les opérateurs de mixes doivent signer un engagement promettant de ne pas conserver les fichiers de logs. De plus, l'internationalisation des mixes limite les risques qu'un attaquant puisse contrôler tout le relais.

Attaques par fingerprinting

L'attaquant connaît tous les nœuds du relais étant donné que ceux-ci sont fixes et peut être un utilisateur du réseau. Néanmoins, JAP utilise un système d'authentification par ticket. Chaque slice contient un ticket qui permet de vérifier que l'utilisateur peut utiliser le système avec ce slice durant un certain temps. Le ticket est inclus et est propre à l'utilisateur. Ainsi, l'empreinte du trafic est toujours différente même si l'utilisateur visite le même site web.

Attaques par congestion

Cette attaque n'est pas applicable pour JAP étant donné que les mixes des relais sont fixes et prédéterminés. Une attaque par congestion s'apparenterait plus à une attaque par déni de service.

Attaques par les ressources

Comme pour les attaques par congestion, cette attaque n'est pas applicable pour JAP étant donné que les mixes des relais sont fixes et prédéterminés.

Attaques par déni de service

La topologie des relais est le "mix-cascade", il suffit donc qu'un seul des mixes ne fonctionne pas pour que tout le relais ne soit plus utilisable. Le nombre de mixes est limité et leur localisation est publique. Les autorités peuvent aisément bloquer le réseau en saisissant les mixes. Pour ce qui concerne des attaques DDoS, il n'existe pas de protection pour assurer la résilience du réseau.

Un des pires scénarios est que l'attaquant puisse bloquer tous les utilisateurs réels sauf un, anéantissant l'anonymat de celui-ci sur le réseau.

4.2.6 Récapitulatif

Nom et valeur	Explication
Topologie : Partiellement	Le relais de mixes est fixe et préétabli.
Direction : Bidirectionnelle	Le cache-proxy se charge de renvoyer la réponse.
Synchronisation : Synchrone	JAP utilise le protocole TCP
Rôles : Client-serveur	Un client ne peut pas devenir un mixe.
Hiérarchie : Plate	Les mixes ont la même importance.
Décentralisation : Semi	Le serveur d'information est un élément crucial du réseau.
Vue du réseau : Complète	Voir serveur d'information.
Mise à jour : Périodique	Voir serveur d'information.
Type de routage : Source-routing	L'utilisateur choisit son relais.
Planification : Équitable	Les connexions sont uniformes et ont la même vitesse.
Déterminisme : Fixé	Les mixes des relais sont prédéterminés.
Ensemble : Choix de l'utilisateur	L'utilisateur choisit son relais.
Distribution : Pondérée	L'utilisateur choisit son relais.

TABLE 4.1 – Caractéristiques de JAP

4.3 Remailers

Un remailer permet l'envoi de courriels anonymes. Contrairement au webmixe, il n'a pas besoin d'une faible latence. Les remailers peuvent être classés en trois types.

4.3.1 Type I : Cypherpunk

Cypherpunk est le premier remailer. Il a été créé en 1992. Il utilise le logiciel PGP ou son équivalent libre GnuPG pour chiffrer les courriels. Seul le logiciel de chiffrement a besoin d'être installé sur la machine. Il suffit de chiffrer le courriel avec la clé publique d'un remailer et de le lui envoyer pour qu'il transmette le courriel. L'envoi d'un courriel via Cypherpunk s'effectue en suivant les étapes suivantes :

1. Obtenir les clés privées des remailers que l'on souhaite utiliser. Pour cela, il faut envoyer à chaque remailer un courriel vide avec comme sujet "remailer-key".
2. Importer les clés dans PGP ou GnuPG avec cette commande :
   ```
   # gpg -import key.txt
   ```
3. Rédiger le message dans le format suivant :
   ```
   ::
   Anon-To: <Courriel du destinataire>

   ##
   Subject: <Sujet>
   ```

```
<Contenu du courriel>
```

4. Chiffrer le message avec la clé publique du remailer. Si l'on souhaite passer par plusieurs remailers, on répète l'étape 3 et 4 avec comme contenu du courriel, le message chiffré par PGP.
5. Le message est prêt à être envoyé au dernier remailer qui a chiffré le message en respectant le format suivant :

```
::
Encrypted: PGP

---BEGIN PGP MESSAGE---
<message chiffré>
---END PGP MESSAGE---
```

Bien que son utilisation reste simple et demande peu de moyens, Cypherpunk ne possède aucune protection contre l'analyse du trafic et ne permet pas au destinataire de répondre.

4.3.2 Type II : Mixmaster

Mixmaster a été créé en 1994. Son protocole est décrit dans [42]. Contrairement au remailer de type I, Mixmaster est protégé contre l'analyse du trafic. C'est un mix network hybride. Tous les messages sont découpés en morceaux de même taille. Des données aléatoires sont ajoutées à la fin des plus petits paquets. Les morceaux sont envoyés indépendamment les uns des autres sans forcément utiliser le même relais de mixes. Seul le dernier mixe a besoin d'être identique pour reconstituer le message. Sa topologie est donc "free-route", mais la sélection des mixes n'est pas indiquée dans le protocole. L'algorithme d'éviction utilisé est le "pool mixe". Il est basé sur trois paramètres, l'intervalle de temps entre chaque éviction, le nombre minimum de messages à placer dans un pool et le pourcentage maximal de messages à envoyer dans un tour d'éviction. Les messages à envoyer sont sélectionnés aléatoirement dans le pool. Chaque remailer génère des messages fictifs aléatoirement. Un remailer devrait générer en moyenne un message fictif sur trente-deux messages réels entrants et un autre tous les neuf tours d'éviction. Ces messages fictifs circulent ensuite dans quatre remailers choisis au hasard et de sorte qu'un remailer ne soit pas choisi deux fois. Comme Cypherpunk, Mixmaster ne permet pas au destinataire de répondre.

Un logiciel doit être installé sur la machine du client. Un des plus populaires est Quicksilver Lite[43], son utilisation reste simple, mais il ne fonctionne que sur Windows et nécessite que l'utilisateur fournisse un courriel valide avec son hôte SMTP. D'après le logiciel, ses informations seront utilisées pour créer un message par défaut et un article sur USENET. La figure 4.4 illustre son interface.

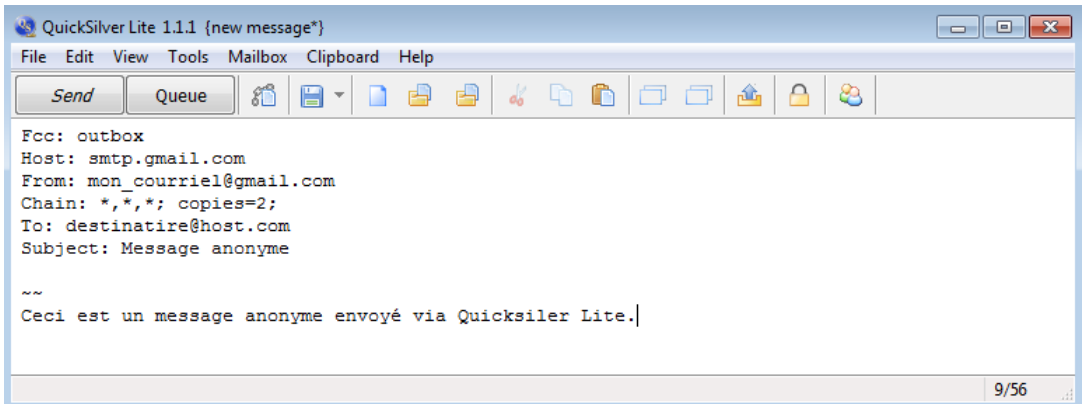

FIGURE 4.4 – Interface utilisateur de Quicksilver Lite

4.3.3 Type III : Mixminion

Mixminon a été créé en 2003. Son fonctionnement est décrit dans [44]. C'est un mix network hybride avec une topologie "free-route". La stratégie d'éviction est identique à celle de Mixmaster. Son principal apport est la possibilité au destinataire de répondre. Pour prévenir les attaques par rejeu, c'est à dire où l'attaquant envoie plusieurs fois le même message dans le relais dans le but d'effectuer les correspondances entre les entrées et les sorties des mixes, Mixminion utilise les SURB (Single Use Reply Blocks). Une réponse est attachée à un SURB, ainsi la répétition de celui-ci dans le réseau n'est pas possible.

Une autre différence importante est l'utilisation de serveurs d'annuaire. Leur fonction est de fournir aux utilisateurs les clés, les capacités et l'état des mixes dans le réseau. Ces serveurs sont synchronisés et redondants permettant à tous les utilisateurs d'obtenir des informations de routage identiques.

Sur le site officiel [45], une note de l'auteur datant de 2013 indique que le projet n'est plus en cours de développement et n'est pas recommandé pour une utilisation en production. La dernière version du projet date de 2007.

4.3.4 Mécanisme de défense

Attaques par intersection

Mixmaster est le seul réseau anonyme à haute latence étudié dans ce mémoire et le seul à utiliser une stratégie d'éviction. De plus, il ne permet l'envoi que de courriel unidirectionnel. Il n'y a donc pas de communication continue avec le destinataire. Tout cela permet à Mixmaster de se prémunir de tout type d'attaque par intersection.

Attaques par étiquetage

Chaque paquet contient un identifiant et permet de contrer les attaques par renvoi. D'après [42], l'intégrité des paquets ne peut pas être vérifiée dans Mixmaster étant donné que l'entête de chaque paquet contient des données aléatoires. Le réseau peut donc être vulnérable si l'attaquant modifie ou ajoute des données dans le paquet. Si l'attaquant supprime tous les messages provenant d'un seul utilisateur, l'attaquant peut voir que certains types de messages anonymes ne sont plus véhiculés dans le réseau et donc il pourra compromettre l'anonymat de l'utilisateur.

Attaques par fingerprinting

Chaque entête de paquets contient des données aléatoires. De plus, le contenu d'un courriel est plus difficile à prévoir qu'une page web statique. Par conséquent, les attaques par fingerprinting sont donc impraticables.

Attaques par congestion

Étant donné que Mixmaster utilise une stratégie d'éviction pour véhiculer ces messages, une attaque par congestion ne permettra pas de ralentir le trafic d'un remailer.

Attaques par les ressources

Le protocole de Mixmaster [42] ne spécifie pas comment les remailers sont sélectionnés. Par conséquent, il est difficile de savoir si le réseau est vulnérable aux attaques par les ressources.

Attaques par déni de service

La topologie en free-route de Mixmaster le rend plus robuste face à une attaque par déni de service. Il faudrait que les autorités saisissent un bon nombre de remailers pour rendre celui-ci inopérable. Néanmoins, il n'y a pas de protection particulière face à une attaque de type DDoS.

4.3.5 Récapitulatif

Le projet Mixminion est au point mort depuis 2007 et Cypherpunk ne peut pas être considéré comme un réseau anonyme à part entière étant donné qu'il ne protège pas des analyses réseau. Il n'y aura donc uniquement que le récapitulatif de Mixmaster.

Nom et valeur	Explication
Topologie : Entièrement	Un client peut se connecter à tous les remailers disponibles.
Direction : Unidirectionnelle	Le destinataire ne peut pas répondre au courriel.
Synchronisation : Asynchrone	Un courriel met plusieurs jours pour arriver à destination.
Rôles : Client-serveur	Un client ne peut pas devenir un remailer.
Hiérarchie : Plate	Les remailers ont la même importance.
Décentralisation : Semi	Seuls les remailers assurent le routage du trafic.
Vue du réseau : Complète	Le client peut voir tous les remailers disponibles.
Mise à jour : Périodique	Les informations sont véhiculées par un réseau ad hoc.
Type de routage : Source-routing	Le client peut choisir tous les nœuds du relais.
Planification : Équitable	Toutes les connexions ont la même priorité.
Déterminisme : Probabiliste	Sélection en free-route.
Ensemble : Tous	Tous les remailers peuvent être sélectionnés.
Distribution : Uniforme	Aucune spécification dans le protocole.

TABLE 4.2 – Caractéristiques de Mixmaster

4.4 TOR : The Onion Router

4.4.1 Onion Routing

Le routage en oignon (onion routing en anglais) a été conçu à la fin des 1990. Son but est de fournir une communication privée à faible latence à travers un réseau public (soit internet). Le réseau est constitué de nœuds nommés "Routeur Oignon". Le fonctionnement est similaire au mix network, un utilisateur établit un circuit de routeur oignon, la communication est cryptée en couche avec chaque clé publique des nœuds du circuit.

Chaque nœud du circuit connaît uniquement son prédécesseur et son successeur. Le dernier nœud du circuit transmet le message au destinataire, il est le seul à le connaître, tout comme le premier nœud du circuit est le seul à connaître l'utilisateur. La communication est bidirectionnelle, le destinataire répond sur le même circuit en sens inverse. Comme pour les webmixes, pour assurer une faible latence sur le réseau, il n'y a pas de stratégie d'éviction, les messages entrant dans un nœud sont immédiatement expédiés au nœud suivant. L'inconvénient est que cela rend la communication vulnérable aux attaques par corrélation basée sur le temps. La figure 4.5 illustre un circuit de routeur oignon.

4.4.2 Tor : La deuxième génération du routage en oignon

TOR pour The Oignon Router est une amélioration de la conception originale du routage par oignon. Le projet est gratuit et son code ouvert. Sa conception est décrite dans [29]. Sa première version date de 2001. TOR est le réseau de communication anonyme le plus populaire. D'après Tor metrics [46], le nombre d'utilisateurs stagne autour de deux millions depuis 2014. Le réseau a connu un pic de six millions d'utilisateurs en 2013 après les révélations d'Edward

Snowden. Début 2017, le trafic atteint 100 Go/s et le nombre de routeurs passe au-dessus de 7000.

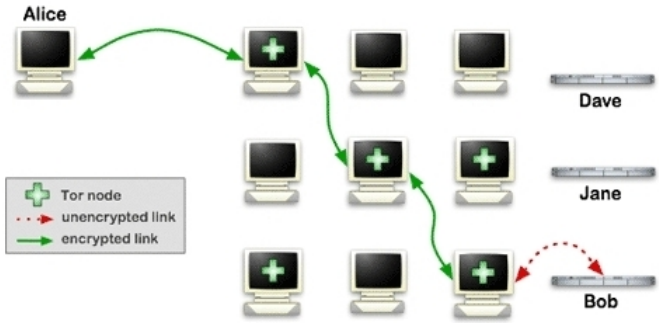

FIGURE 4.5 – Exemple d'un circuit de routeur oignon [47]

L'utilisateur doit installer un proxy local qui va se charger de construire les circuits et de gérer les connexions sur le réseau. Ce proxy accepte les flux TCP et supporte le SOCKS permettant d'éviter d'installer un proxy spécifique à chaque application. Chaque routeur oignon possède une clé d'identification qui est utilisée pour signer son certificat numérique et sa description. La description d'un routeur est constituée de ses clés, son adresse, de sa bande passante, de sa politique de sortie et d'autres informations.

La communication entre les nœuds s'effectue via TLS avec des clés éphémères. Les données sont véhiculées dans des cellules de 512 octets. Les entêtes de chaque cellule contiennent un identifiant du circuit auxquelles elles appartiennent ainsi qu'une commande sur ce qu'il faut faire avec le contenu. Il existe deux types de cellules, la cellule de contrôle qui contient les commandes de création ou de destruction de circuit et la cellule de relais qui assure le transport de flux de données.

Un circuit peut partager plusieurs flux TCP. Pour prévenir la liaison entre les flux, le proxy de l'utilisateur change de circuit chaque minute et expire ceux qui ne véhiculent plus de flux. La construction d'un circuit s'effectue un nœud à la fois. Le proxy échange les clés symétriques avec chaque nœud du circuit. Un circuit se construit selon les étapes suivantes :

1. Le proxy envoie au premier nœud une cellule de contrôle avec la commande *create*.
2. Le nœud répond avec une cellule de contrôle avec la commande *created*.
3. La clé symétrique ayant été échangée, le circuit est établi et le proxy peut envoyer des cellules de relais chiffré avec cette clé.
4. Pour étendre le circuit à un autre nœud, le proxy envoie une cellule de relais avec la commande *extend* qui contient l'adresse du deuxième nœud.
5. Le premier nœud effectue les étapes 1 et 2 avec le deuxième nœud.

6. Lorsque le premier nœud reçoit la cellule de contrôle *created* du deuxième nœud, il transfère son contenu au proxy par une cellule de relais avec la commande. *extended*.
7. Ses étapes sont répétées pour chaque nœud à ajouter au proxy.

Pour détruire le circuit, le proxy envoie une cellule de contrôle avec la commande *destroy*. Chaque nœud du circuit ferme les flux liés au circuit et transfère la cellule au prochain nœud.

Pour ouvrir un nouveau flux, le proxy envoie une cellule de relais avec la commande *relay begin* au nœud de sortie. Chaque flux possède un identifiant. Une fois que le noeud de sortie a établi la connexion avec le serveur distant, il envoie une cellule de relais avec la commande *relay connected* au proxy. Ensuite, les données sont échangées par des cellules de relais avec la commande *relay data*.

4.4.3 Garde d'entrée

Étant donné que ce nœud est le seul à connaître l'identité de l'utilisateur, il est la cible privilégiée des attaquants. Si un attaquant surveille le nœud d'entrée ainsi que le nœud de sortie, l'anonymat de TOR est brisé. Si un attaquant surveille C nœud sur leur ensemble N, il peut corréler tout le trafic envoyé par un utilisateur avec une probabilité de $(C/N)^2$. Pour contrer ce genre d'attaque, il n'est pas possible de choisir aléatoirement son nœud d'entrée et de sortie. Les "gardes d'entrées" sont un ensemble de nœuds choisis aléatoirement par chaque client TOR. Ces nœuds seront utilisés uniquement comme nœud d'entrée. Tant que l'attaquant ne surveille pas tous les nœuds, l'attaque par corrélation ne peut fonctionner. S'ils les surveillent, la probabilité que l'attaque par corrélation réussisse est plus faible qu'auparavant. L'utilisateur à $(N - C)/N$ chances d'éviter l'attaque.

Un nœud qui rejoint le réseau ne peut pas devenir immédiatement un garde d'entrée, il doit répondre à certains critères et passer par plusieurs phases décrites dans la sous-section du cycle de vie des nouveaux nœuds.

4.4.4 Nœud de sortie

Le nœud de sortie peut voir les requêtes en clair des clients sans pouvoir remonter jusqu'à eux. Étant donné que le nœud sort de TOR pour accéder à la page demandée, un attaquant peut facilement obtenir l'adresse IP du nœud de sortie. Si ce nœud est utilisé pour véhiculer du trafic illégal, le propriétaire peut être tenu comme responsable au niveau de la loi. Tor project fournit une liste de recommandation [48] pour éviter ce genre de problème.

Pour qu'un nœud puisse devenir un nœud de sortie celui-ci doit ouvrir au moins deux de ces trois ports : 80, 443 ou 6667 et doit permettre de sortir sur au moins 16777214 adresses IP sur internet (\8).

4.4.5 Service caché

Darknet

En référence à [49] et [50], le web peut être divisé en trois couches :

- Le web surfacique (ou indexable) est la partie accessible en ligne, celle connue par le public. Elle contient toutes les ressources indexées par les moteurs de recherche classiques comme Google ou Bing.
- Le web profond (ou DeepWeb) est la partie qui n'est pas indexée par les moteurs de recherches classiques. Elle représenterait plus de 90% du contenu d'internet. Elle est principalement constituée de données gouvernementales, médicales, financières, scientifiques, etc. Leur format de données est incompréhensible pour le moteur de recherche, c'est la raison pour laquelle il ne peut pas les indexer.
- Le darknet est une partie du web profond dont les données sont volontairement cachées. Y accéder demande l'utilisation d'outils spéciaux assurant l'anonymat de leurs utilisateurs.

FIGURE 4.6 – Les trois couches du web [50]

Point de Rendez-vous

Les services cachés de TOR font partie du darknet. Pour y accéder, il est obligatoire de passer par le réseau TOR. L'intérêt d'un service caché est qu'il permet d'assurer un anonymat mutuel entre le serveur et le client. Son architecture est illustrée dans la figure 4.7.

L'établissement d'un service caché sur le réseau TOR s'effectue en six étapes :

1. Le serveur génère une paire de clés RSA. Les services cachés utilisent des noms d'hôtes `.onion`. Une adresse `.onion` correspond à la clé publique du serveur encodée en base

32. Par exemple, `https://76qugh5bey5gum7l.onion` est l'adresse qui permet d'accéder au service caché "Deep Web Radio". Ces adresses sont publiques, mais il n'est possible d'accéder au service caché uniquement en passant par le réseau TOR.

2. Le serveur choisit plusieurs routeurs oignon qui vont servir de "point d'introduction". Ces points vont servir à transmettre le message initial entre le client et le serveur. Un circuit est construit pour chaque point.
3. Le serveur génère un descripteur contenant sa clé publique et ses points d'introduction. Le serveur choisit un ensemble de nœuds nommés "HSDir" (Hidden Service Directory). Il envoie son descripteur à l'un de ces nœuds. Le HSDir choisi change toutes les 24 heures. Un nœud doit être actif depuis au moins 25 heures pour pouvoir être un HSDir. Le descripteur est signé avec la clé privée du serveur.
4. Le client prend connaissance de l'existence du service caché grâce à l'adresse `.onion` et télécharge le descripteur dans le DHT. Il construit un circuit avec un routeur oignon nommé "point de rendez-vous" qui va permettre de communiquer avec le serveur. Le client fournit un cookie aléatoire au point de rendez-vous qui servira à authentifier le serveur plus tard.
5. Le client crée un circuit vers un des points d'introduction du serveur et envoie un message chiffré avec la clé publique du serveur contenant les informations du point de rendez-vous et le cookie. Le point d'introduction transfère ensuite le message vers le serveur.
6. Le serveur peut maintenant construire un circuit vers le point de rendez-vous et envoie le cookie qui permet de l'identifier. La connexion entre le client et le serveur est établie.

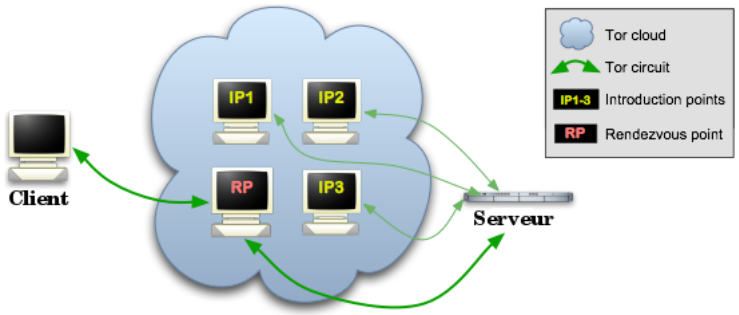

FIGURE 4.7 – Architecture d'une communication à un service caché [51]

4.4.6 Serveurs d'annuaire

Pour fournir la liste de nœuds disponibles sur le réseau ainsi que leurs caractéristiques (bande passante, adresse IP, clés de chiffrement, etc.) TOR utilise des serveurs d'annuaire similaire à ceux de Mixminion. Ces serveurs sont au nombre de 9 actuellement. Ceux-ci sont synchronisés et redondants pour fournir en tout temps le même annuaire. On peut obtenir la liste de ces serveurs sur le site Atlas [52]. Ce site est une application web permettant d'explorer les routeurs

oignon. Chaque routeur publie périodiquement son état à tous les serveurs d'annuaire. Les informations envoyées sont signées dans le but d'authentifier les routeurs oignon. Les nouveaux nœuds doivent être approuvés par un administrateur d'un des serveurs d'annuaire. Sans cela, un attaquant pourrait créer de nombreux routeurs compromettant l'anonymat du réseau. Un serveur d'annuaire doit être signé par les autres serveurs d'annuaire, ainsi un attaquant ne pourra pas se faire passer pour un de ces serveurs.

4.4.7 Cycle de vie des nouveaux nœuds

Lorsqu'un nœud rejoint le réseau, il passe par quatre phases.

Première phase dite "Non mesurée"

D'une durée de trois jours, le nœud n'est d'aucune utilité dans le réseau durant cette phase. Le nœud teste sa bande passante en construisant quatre circuits qui reviennent vers lui. Il envoie 125 Ko sur chaque circuit pour mesurer sa bande passante. Cela lance le système de mesure de bande passante passive de Tor. Celui-ci estime la bande passante sur le débit envoyé pendant 10 secondes donc la première bande passante calculée devrait être 4*125/10 = 50 Ko. Le résultat est publié dans la description du nœud.

Un groupe d'ordinateurs rapide nommé "bwauths" vérifie la bande passante du nœud sur les trois jours. Le serveur d'annuaire ajuste la valeur de la bande passante vers le haut ou vers le bas en comparant les résultats à d'autres nœuds qui indiquent une vitesse similaire. Dans cette phase, la bande passante est limitée à 20 Ko. Cette faible bande passante fait en sorte que le nœud ne puisse pas être sélectionné pour la construction de circuits et donc n'est pas utilisé.

Deuxième phase dite de "Mesure à distance"

Cette phase dure cinq jours. Au début de cette phase, seulement quelques clients construisent leurs circuits avec le nouveau nœud. Le trafic généré permet d'augmenter l'estimation de la bande passante calculée par le système de mesure passif. Les "bwauths" vont comparer le nouveau résultat avec des nœuds plus rapides ce qui va augmenter la valeur de la bande passante réajustée. Avec cette nouvelle valeur, plus de nœuds choisiront le nouveau nœud pour leur circuit ce qui générera encore plus de trafic et ainsi la valeur de la bande passante du nouveau nœud augmentera progressivement.

Seuls les nœuds considérés comme stables et fiables peuvent être utilisés comme "garde d'entrée". De plus, le nouveau nœud est configuré de telle sorte qu'il ne peut pas être un nœud de sortie. À cette phase, le nœud ne peut être qu'un nœud intermédiaire dans un circuit.

Troisième phase dite de "passage en garde d'entrée"

Cette phase commence au huitième jour depuis que le nœud a rejoint le réseau jusqu'au soixante-huitième. Les serveurs d'annuaires attribuent le "flag de garde" à un nœud lorsque sa bande passante et son temps de disponibilité sont hauts. Même si le nœud satisfait ces deux caractéristiques, celui-ci ne peut obtenir le flag qu'à partir du huitième jour où il a rejoint le réseau.

Dès que le nœud obtient le flag de garde, les clients ne l'utiliseront plus que comme un nœud intermédiaire. Les nœuds d'entrée et de sortie sont plus rares que les nœuds intermédiaires, les clients considèrent donc que ces nœuds ont potentiellement plus de charges que les autres et évitent de les utiliser comme nœud intermédiaire. À cette phase-là, le nœud étant garde d'entrée que depuis peu, il n'a pas beaucoup de charges et son trafic va diminuer fortement. Pour éviter que les nouveaux gardes d'entrée soient sous-exploités, les clients changent de nœud d'entrée parmi ceux qu'ils ont sélectionnés tous les quatre à huit semaines. Ainsi les nouveaux gardes d'entrée ne seront pas isolés et les anciens ne verront pas leur charge s'alourdir avec le temps.

Quatrième phase dite de "Garde d'entrée stabilisée"

Dernière phase du cycle, elle commence au bout du soixante-huitième jour. Une fois que le nœud a été garde d'entrée durant une période de changement complète (de huit à douze semaines suivant la version de TOR utilisée), le nombre de clients qui quittent le nœud pour changer de garde d'entrée et ceux qui l'ajoutent à leur liste, est sensé se stabiliser.

4.4.8 Tails

Tails pour The Amnesic Incognito Live System est un système d'exploitation basé sur Debian. Celui-ci s'utilise en live sur la plupart des supports tels que l'USB, les cartes SD ou le DVD. Il ne laisse aucune trace sur l'ordinateur. Le navigateur web TorBrowser y est installé. Il est conçu pour naviguer sur internet et accéder aux services cachés sur le réseau TOR . Ce navigateur contient plusieurs extensions protégeant l'utilisateur des attaques applicatives. Au démarrage, Tails construit des circuits et bloque toutes les connexions qui ne transitent pas sur le réseau TOR. Par conséquent, Tails est l'une des solutions fournissant le meilleur niveau d'anonymat. Le site officiel [53] fournit des instructions ainsi qu'un logiciel `tails-installer` facilitant l'installation du système d'exploitation sur un support amovible.

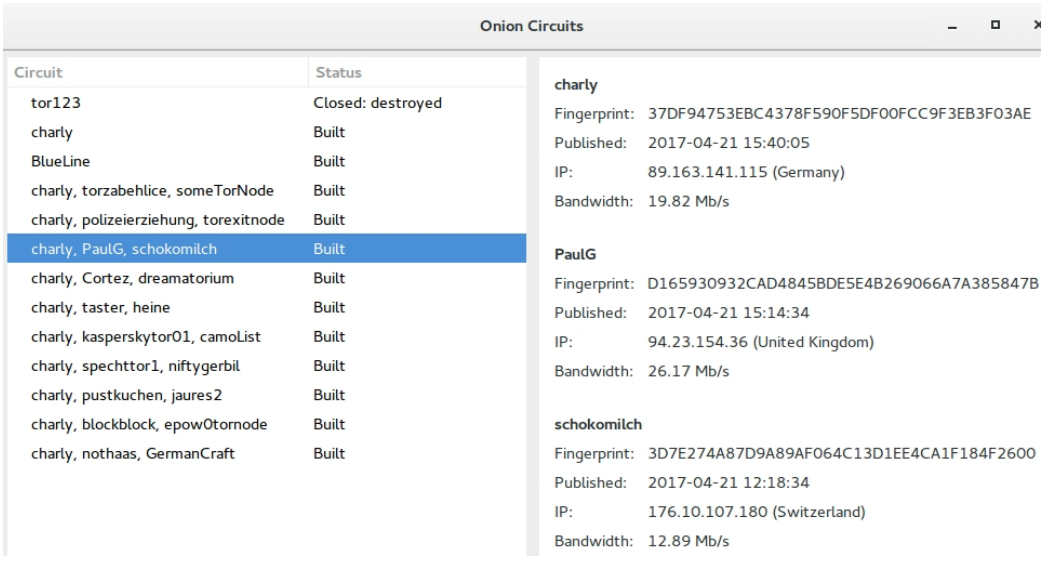

FIGURE 4.8 – Liste de circuit construit dans Tails

4.4.9 Mécanisme de défense

Attaques par intersection

TOR change de circuit toutes les dix minutes réduisant les chances de succès d'une attaque par intersection. De plus, l'utilisateur choisit ses gardes d'entrée uniquement parmi un ensemble de nœuds constitué au début de la connexion au réseau permettant de se protéger des attaques du prédécesseur. Pour finir, TOR ajoute des données aléatoires à ses paquets pour se prémunir des attaques par comptage de paquets.

Attaques par étiquetage

D'après [20], TOR ne serait pas protégé des attaques par étiquetage. Néanmoins, elles seraient difficiles à mettre en place contre un utilisateur particulier étant donné qu'elles requièrent le contrôle du nœud d'entrée et du nœud de sortie.

Attaques par fingerprinting

[20] n'a trouvé aucune information à propos de la protection contre ce genre d'attaque, mais comme pour les autres attaques, plus la taille du réseau est grande plus l'attaque est difficilement réalisable.

Attaques par congestion

Les circuits sont changés toutes les dix minutes et la taille du réseau rend cette attaque pratiquement impraticable.

Attaques par ressources

Nous avons vu dans la section sur le cycle de vie des nouveaux nœuds que TOR se base sur deux critères pour sélectionner les nœuds d'un relais. La première est la durée de vie du relais, plus un relais est vieux, plus il a de chance d'être sélectionné. Ensuite, le deuxième critère est la bande passante, les nœuds qui ont la plus grande bande passante sont plus susceptibles d'être sélectionnés.

Toutes ces informations sont fournies par les serveurs d'annuaire, mais celles-ci ne sont pas toujours vérifiées et un attaquant peut fournir de fausses informations sur des nœuds qu'ils contrôlent dans le but qu'ils soient sélectionnés dans le relais d'un utilisateur.

D'après [20], ce type d'attaque serait facile à détecter, mais TOR ne possède aucun mécanisme de détection.

Attaques par déni de service

Étant donné que la liste des relais du réseau est publique, un FAI ou un gouvernement peut bloquer leur accès. Pour contourner ce blocage, TOR a mis en place les "bridges" (ponts en français). Un bridge est un utilisateur volontaire ayant accès à TOR et propose aux utilisateurs qui sont bloqués de passer par lui pour accéder au réseau. L'adresse IP du bridge n'étant pas publique doit être distribuée en privé. Néanmoins, certains gouvernements se font passer pour des utilisateurs dans le but d'obtenir les adresses IP des bridges et ainsi les bloquer. De plus, l'obtention de l'adresse IP ne passant pas par un réseau anonyme, l'attaquant peut surveiller les communications de l'utilisateur et bloquer l'IP quand celle-ci est fournie à l'utilisateur.

Précédemment, nous avons vu comment les circuits sont construits. Un attaquant pourrait provoquer un déni de service en le submergeant de cellules CREATE étant donné que le processus demande beaucoup de ressources pour la machine. De plus, cette attaque pourrait être utilisée pour faciliter une attaque par les ressources. La performance d'un relais attaqué va décroître et il sera moins susceptible d'être choisi dans le circuit d'un utilisateur au profit des relais contrôlés par l'attaquant.

4.4.10 Récapitulatif

Nom et valeur	Explication
Topologie : Principalement	Les nœuds ayant le même espace de sous-réseau IP/16 ne peuvent pas se connecter entre eux.
Direction : Bidirectionnelle	La réponse est renvoyée dans le circuit en sens inverse.
Synchronisation : Asynchrone	Il n'y a aucun délai pour l'envoi de données.
Rôles : Hybride	Un nœud peut devenir un relais.
Hiérarchie : Plate	Les relais ont la même importance.
Décentralisation : Semi	Les serveurs d'annuaire sont des éléments cruciaux du réseau.
Vue du réseau : Complète	Tous les nœuds disponibles sur le réseau peuvent être choisis.
Mise à jour : Périodique	Voir serveurs d'annuaires.
Type de routage : Source-routing	Le client choisit tous les nœuds de son circuit.
Planification : Équitable	Toutes les connexions ont la même priorité.
Déterminisme : Probabiliste	Voir Cycle de vie des nouveaux nœuds.
Ensemble : Restreint	Voir ce paragraphe et ce paragraphe
Distribution : Pondéré	Basé sur la bande passante des nœuds.

TABLE 4.3 – Caractéristiques de TOR

4.5 I2P : Invisible Internet Project

I2P pour Invisible Internet Project est le deuxième réseau anonyme le plus utilisé derrière TOR. Il a été conçu en 2003. Son utilisation est gratuite et son code ouvert. Sa conception est détaillée sur le site officiel[30]. D'après [54], en 2017, le nombre total de nœuds sur le réseau est estimé autour de 50000.

Le fonctionnement est similaire au mix network, le trafic transite par plusieurs nœuds. Sa topologie est le free-route. La grande différence avec les autres réseaux anonymes est qu'I2P est cloisonné, il n'est pas conçu pour accéder à l'internet standard à moins de passer par un nœud qui ferait office de proxy (outproxies). Ainsi, tout nœud dans le réseau est anonyme. L'émetteur ne peut pas connaître l'identité du destinataire et inversement. I2P supporte la plupart des applications disponibles sur l'internet standard tel que la navigation et l'hébergement web, IRC, le courriel ou le partage de fichiers.

Le logiciel à installer sur la machine de l'utilisateur est codé en Java, il est donc nécessaire d'installer l'environnement d'exécution Java (JRE). Le logiciel agit en tant que routeur sur le réseau. Tout routeur connecté au réseau peut participer à la transition des données. À partir du moment où l'utilisateur est connecté à I2P, il peut faire partie d'un relais d'un ou plusieurs autres utilisateurs. I2P est "orienté message", les communications ne sont pas bidirectionnelles.

4.5.1 Tunnels

La communication à travers I2P s'effectue à l'aide de tunnel. Un tunnel est une chaîne de routeurs par laquelle sont véhiculés les messages. Les messages ne vont que dans un seul sens. Deux types de tunnels existent, les tunnels sortants permettant d'envoyer des messages et les tunnels entrants permettant d'en recevoir. Le premier nœud d'un tunnel est nommé la passerelle, le dernier, le nœud de sortie et les autres, les nœuds intermédiaires. Chaque routeur peut avoir plusieurs de ces tunnels. L'utilisation de multiples tunnels pour un routeur augmente la résilience et équilibre la charge du trafic. Un routeur envoie son message dans un de ses tunnels sortants, son nœud de sortie transmet le message à la passerelle d'un des tunnels entrants du destinataire. Pour que le nœud de sortie puisse savoir à quel tunnel entrant envoyer le message, le routeur ajoute les instructions dans le message chiffré. Les nœuds intermédiaires ne peuvent pas savoir s'ils transmettent des données d'un tunnel entrant ou sortant. Chaque tunnel a un identifiant unique choisi aléatoirement par le routeur qui l'a créé. La durée de vie d'un tunnel est de dix minutes.

I2P utilise son propre protocole I2NP (I2P Network Protocol) pour gérer le routage des messages. La passerelle d'un tunnel accumule un nombre de messages I2NP, les fragmente en morceaux de taille fixe (1024 octets), les chiffre et les envoie au prochain nœud. Le nœud de sortie est chargé de reconstruire le message. La passerelle ajoute à chaque fragment chiffré l'identifiant du tunnel ainsi qu'un vecteur d'initialisation généré aléatoirement. Les fragments sont nommés "message du tunnel". Lorsqu'un nœud reçoit un message, il vérifie que ce message provienne du même nœud qui a envoyé les messages précédents. Si le message provient d'un autre nœud ou s'il est identique à un message déjà reçu, le message est rejeté.

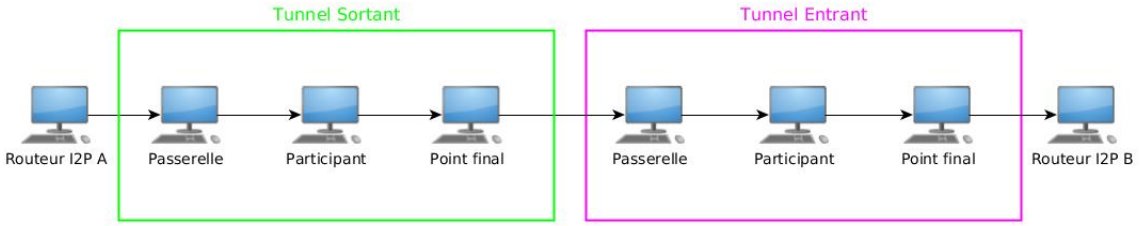

FIGURE 4.9 – Tunnels entrant et sortant

4.5.2 netDb

netDb est une base de données distribuées sur le réseau. Chaque donnée est signée et vérifiée par l'entité qui l'utilise ou la stocke. Les données sont distribuées par la méthode "floodfill". Une partie des routeurs du réseau nommée les "routeurs floodfill" maintient la base de données. netDb est décentralisé, il n'y a pas d'autorité centrale pour gérer la base de données, elle fonctionne comme une DHT. Les routeurs floodfill n'ont pas besoin d'être vérifiés

et peuvent changer dans le temps. Un routeur doit passer quelques tests de performance pour devenir floodfill. Environ 6% des routeurs du réseau sont floodfill. Les routeurs avec une bande passante élevée deviennent automatiquement floodfill lorsque le nombre de routeurs floodfill descend en dessous d'un seuil et inversement, ces routeurs ne le sont plus lorsqu'il y en a trop. L'algorithme de Kademlia est utilisé pour déterminer la proximité entre une clé et un routeur. Pour augmenter le coût des attaques Sybil, la date sous le format yyyyMMdd est ajoutée à la clé pour former le hash. Ce hash est nommé clé de routage.

Il existe deux types de données :

— RouterInfo : Ce type de données contient des informations à propos d'un routeur I2P permettant de le contacter. Elle est composée de l'identité d'un routeur qui consiste en une clé de chiffrement, une clé de signature et un certificat ainsi que ses adresses de contact et d'autres informations secondaires.

— LeaseSets : Ce type de donnée contient un ensemble de points d'entrées de tunnels nommées "lease" pour une destination particulière. Chaque "lease" contient l'identité du routeur qui fait office de passerelle, l'identifiant du tunnel et sa date d'expiration.

Les nœuds envoient leur "RouterInfo" directement dans netDb tandis que les "leaSets" sont envoyés à travers les tunnels.

4.5.3 Construction des tunnels

I2P collecte sur chaque nœud des données de profilage tel que le temps de réponse ou le taux d'échec de ses tunnels. Ces données permettent de connaître les performances de chaque nœud. Ceux-ci sont ensuite classés en trois groupes suivant leur vitesse de transfert par tunnel et leur capacité à construire et maintenir leurs tunnels :

— "Haute capacité" : Regroupe les nœuds ayant une capacité au-dessus de la médiane de l'ensemble des nœuds. Le nombre de nœuds dans ce groupe est limité à 75.

— "Rapide" : Les nœuds doivent être de haute capacité et leur vitesse de transfert doit être supérieure à la médiane de l'ensemble des nœuds. Ce groupe est inclus dans le premier et est limité à 30.

— "Standard" : Tous les nœuds n'étant pas de "haute capacité".

Pour construire ses tunnels, un nœud collecte des données "RouterInfo" dans netDb. Le but étant de constituer une liste de nœuds qui pourrait être utilisée dans ses tunnels. Ensuite, le nœud envoie un message de construction du tunnel au premier nœud choisi. Ce nœud retransmettra la demande de construction au nœud suivant qui fera de même pour le dernier nœud.

En plus des deux types de tunnels existants, ceux-ci peuvent avoir deux rôles différents, celui d'explorateur et celui de client. Un tunnel explorateur sert pour la maintenance de la base de données du réseau et des tunnels, tandis qu'un tunnel client sert pour les messages de bout

en bout. La sélection des nœuds lors de la construction du tunnel diffère suivant le rôle que le tunnel doit jouer. La sélection des nœuds pour un tunnel explorateur est aléatoire parmi les nœuds du groupe standard tandis que pour les tunnels clients, la sélection des nœuds s'effectue aléatoirement dans le groupe rapide.

Une fois ses tunnels construits, le nœud, pour contacter un autre nœud, va récupérer son "leaseSet" dans netDb et donc la liste des passerelles des tunnels entrants du nœud distant. Ensuite, le nœud envoie dans un de ses tunnels sortants un message contenant les informations d'un des tunnels entrants du nœud distant permettant au nœud de sortie du tunnel sortant de savoir à qui retransmettre le message.

Par exemple, le nœud A vient de rejoindre le réseau. Il veut contacter le nœud B. A doit construire un tunnel sortant avant de pouvoir contacter B. Il va chercher dans NetDb un routeur du groupe rapide et construit le début du tunnel avec lui. A va ensuite chercher un deuxième routeur, les messages de construction du tunnel passeront par le premier routeur. A effectue les mêmes étapes pour le dernier routeur, les messages de construction du tunnel passeront par les deux premiers routeurs. Une fois le tunnel construit, A va récupérer le "leaseSet" de B dans netDb. A peut maintenant se connecter à un des tunnels entrants de B pour le contacter.

4.5.4 Chiffrement Garlic

I2P utilise le chiffrement Garlic qui peut être vu comme une extension du chiffrement en couche onion. Au lieu de chiffrer les messages un par un, ceux-ci le sont ensemble. On nomme ce tout un "garlic" et les messages le contenant des "cloves". Le chiffrement Garlic est une combinaison du chiffrement asymétrique ElGamal, du chiffrement symétrique AES avec un tag de session. Il est utilisé pour la construction des tunnels et le transfert des messages à travers ceux-ci, pour l'acquittement et pour la publication d'entrées dans la base de données du réseau.

Pour le transfert des messages de bout en bout, un message Garlic contient le plus souvent un seul clove mais régulièrement, deux cloves supplémentaires peuvent être ajoutés. Le premier est utilisé pour l'acquittement et l'autre contient le leaseset de l'émetteur permettant de garantir le maintien de la communication. La figure ci-dessous illustre son fonctionnement :

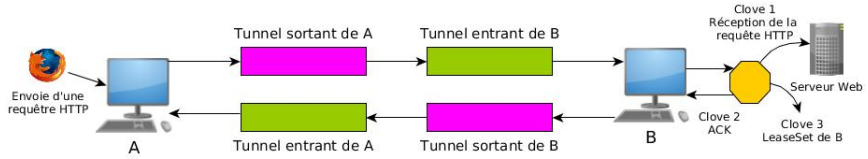

FIGURE 4.10 – Chiffrement Garlic

4.5.5 Mécanisme de défense

Attaques par intersection

Comme pour tout réseau anonyme à faible latence, I2P ne peut pas se protéger complètement contre ce type d'attaque. Néanmoins, quelques mécanismes de protection ont été implémentés :

1. L'ordre des nœuds dans les tunnels est toujours maintenu.
2. Les groupes de profilage changent lentement.
3. La durée de vie des tunnels est de 10 minutes.

Attaques par étiquetage

Un attaquant externe au réseau ne pourra pas modifier les messages, car ceux-ci sont signés. Néanmoins, si l'attaquant contrôle la passerelle d'un tunnel entrant ainsi qu'un autre nœud plus loin dans le tunnel, en modifiant les paquets il pourra vérifier qu'ils sont sur le même tunnel. Par contre, aucune collusion n'est possible si l'attaquant contrôle un nœud dans un tunnel sortant ou ailleurs dans un tunnel entrant.

Attaques par fingerprinting

Comme pour TOR, [20] n'a trouvé aucune information au sujet de protection contre ce genre d'attaque.

Attaques par congestion

Comme pour TOR, I2P change les tunnels toutes les 10 minutes. Néanmoins, la taille du réseau est plus petite, le réseau est donc moins robuste contre ce type d'attaque.

Attaques par les ressources

L'attaquant pourrait contrôler une grande partie de netDB et faire en sorte que ces nœuds soient sélectionnés dans les groupes de profilage rapide ou haute capacité et ainsi favoriser leur sélection dans les tunnels des utilisateurs. La prise de contrôle de netDB peut s'effectuer avec une attaque Sybil qui consiste à fournir de fausses informations pour qu'un maximum des nœuds de l'attaquant devienne "floodfill". Comme pour la plupart des autres attaques, plus le réseau est grand plus l'attaque deviendra coûteuse.

Attaques par déni de service

Un attaquant peut créer un grand nombre de nœuds dans le réseau et faire en sorte que ceux-ci ne fournissent aucune ressource. La base de données deviendra plus grande inutilement et les nœuds du réseau devront demander plus de tunnels. Pour se protéger de cette attaque, I2P

tente d'identifier les nœuds défaillants avec son système de profilage. Ces nœuds seront par la suite soit ignorés, soit sous-utilisés.

I2P ne possède pas de protection contre les attaques DDoS mais la décentralisation de son réseau rend l'attaque plus difficilement réalisable.

4.5.6 Récapitulatif

Nom et valeur	Explication
Topologie : Principalement	L'architecture d'I2P est basée sur Kademlia.
Direction : Unidirectionnelle	Les flux de communication ne vont que dans un sens.
Synchronisation : Asynchrone	Il n'y a aucun délai pour l'envoi de données.
Rôles : P2P	Un client peut être utilisé dans un tunnel.
Hiérarchie : Plate	Les nœuds ont la même importance.
Décentralisation : Complètement	netDb est décentralisé.
Vue du réseau : Complète	Tous les nœuds disponibles sur le réseau peuvent être choisis.
Mise à jour : Périodique	Voir netDb
Type de routage : Source-routing	Le client choisit tous les nœuds de ses tunnels.
Planification : Priorisée	Les messages n'ont pas la même priorité suivant leur type[55].
Déterminisme : Probabiliste	Basé sur les trois groupes de nœuds.
Ensemble : Restreint	Voir ce paragraphe.
Distribution : Pondéré	Voir ce paragraphe.

TABLE 4.4 – Caractéristiques de I2P

4.6 Autres réseaux anonymes

4.6.1 Crowds

Crowds[56] est un réseau anonyme P2P créé en 1998, c'est un des premiers permettant de naviguer sur le web anonymement. Les nœuds sont groupés dans des "Crowds". Seuls les nœuds d'un même Crowd peuvent se connecter entre eux pour relayer le trafic. Chaque nœud d'un crowd est nommé "jondo". Tout utilisateur qui rejoint le réseau devient un jondo. Un serveur nommé "blender" administre les nœuds du réseau. Un utilisateur rejoint un Crowd en s'enregistrant auprès du blender. Celui-ci avertit tous les nœuds du Crowd de l'arrivée du nouveau nœud. Lorsque l'utilisateur veut accéder à un site web, il va sélectionner un jondo au hasard et lui transfère la requête. Celui-ci va choisir aléatoirement s'il transfère le paquet à un autre jondo (qui sera toujours choisi au hasard) ou s'il l'envoie directement au serveur web. Le processus est répété jusqu'à ce que la requête arrive à destination. La réponse du serveur est renvoyée en sens inverse. Un jondo ne peut pas savoir si les paquets qu'il reçoit proviennent de la source ou d'un jondo intermédiaire. Le fait que Crowds construise ses chemins aléatoirement le rend vulnérable aux attaques du prédécesseur[35].

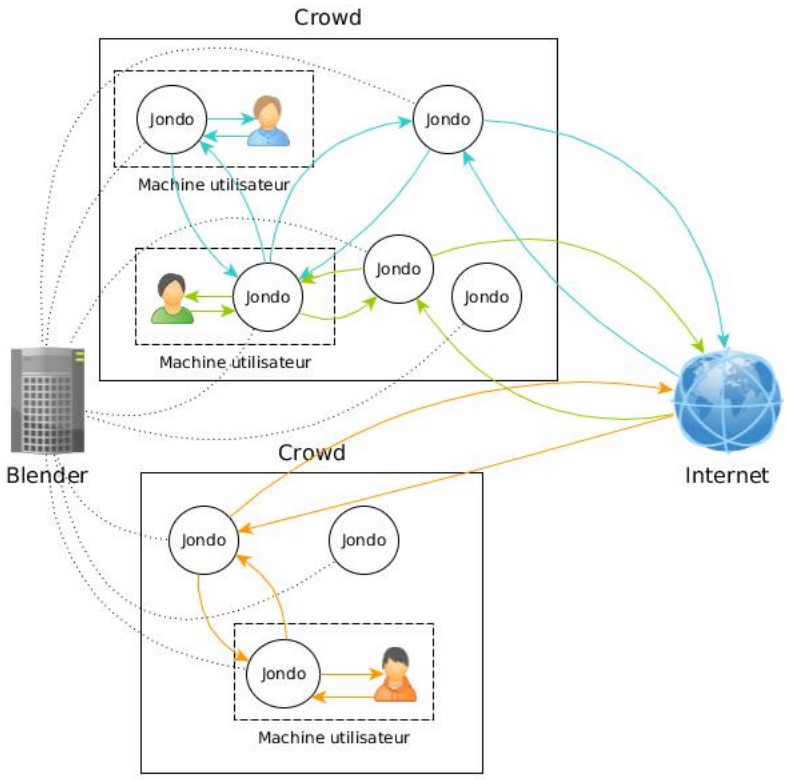

FIGURE 4.11 – Architecture de Crowds

4.6.2 Tarzan

Tarzan est un réseau anonyme P2P créé en 2002. Son fonctionnement est décrit dans [57] et dans [58]. Le réseau est complètement décentralisé et ouvert à tous. Les nœuds ont tous la même importance. Son fonctionnement est similaire aux mix networks. L'utilisateur doit installer un logiciel client. Ce logiciel transfère l'adresse IP que l'utilisateur souhaite accéder anonymement. Le message chiffré circule ensuite dans le circuit jusqu'au nœud de sortie. La sortie agit comme un NAT pour accéder à l'adresse IP que l'utilisateur a fournie. La réponse du serveur est ensuite renvoyée jusqu'à l'utilisateur en sens inverse.

Tarzan utilise l'algorithme de Chord pour stocker les informations sur ses nœuds. Chord est une DHT où chaque nœud est identifié par un hash de 160 bits généré à partir de l'adresse IP du nœud. Une recherche dans cette DHT a une complexité d'au plus $\mathcal{O}(log(n))$. Un utilisateur rejoignant le réseau peut contacter des nœuds seulement dans son espace d'adressage par exemple, les nœuds appartenant à son sous-réseau. Ensuite, l'utilisateur peut construire ses circuits en choisissant aléatoirement les nœuds de cet ensemble.

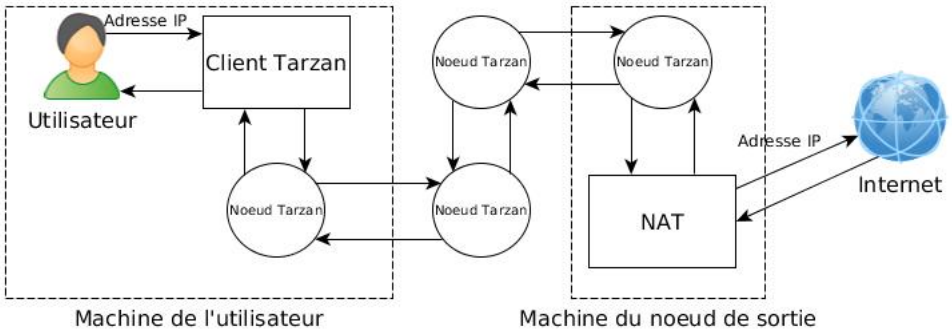

FIGURE 4.12 – Architecture de Tarzan

4.6.3 DC-nets

Les DCnets pour Dining cryptographers networks sont une alternative aux mix-networks. Ils permettent l'échange de messages entre tous les participants et garantissent leur anonymat. Chaum est le premier à avoir conçu un DCnets dans son article "The Dining Cryptographers Problem : Unconditional Sender and Recipient Untraceability"[59].

Chaum illustre le fonctionnement des DCnets en posant le problème suivant : Trois cryptographes A,B et C sont à table. Ils savent que celui qui va payer le repas est soit l'un d'entre eux soit la National Security Agency (NSA). Ils veulent savoir si c'est l'un d'entre eux qui a payé ou la NSA sans dévoiler l'identité du payeur si celui-ci est un cryptographe. Pour ce faire, ils développent un protocole en deux étapes :

1. Tous les deux cryptographes partagent un bit secret. Par exemple, A et B partage le bit 0, A et C partage le bit 1 et B et C partage le bit 1.
2. Chaque cryptographe annonce publiquement la valeur d'un bit en suivant les règles suivantes :
   - Si le cryptographe n'est pas payé, il publie le OU-EXCLUSIF de ses bits. Par exemple A publiera $1 \oplus 0 = 1$.
   - Si le cryptographe a payé, il publie l'inverse du résultat de l'opération OU EXCLUSIF. Par exemple A publiera $\neg(1 \oplus 0) = 0$.

   Ensuite, on effectue un OU EXCLUSIF de chacun des bits publiés. Si le résultat est égal à 1, un des cryptographes a payé sans que l'on sache son identité sinon c'est que c'est la NSA. Dans notre exemple, dans le premier cas on aurait $1 \oplus 1 \oplus 0 = 0$ et dans le deuxième où A paie, on a $0 \oplus 1 \oplus 0 = 0$.

L'exemple est illustré dans la figure 4.13. Si un utilisateur veut envoyer plus d'un bit sur le réseau, il faut que le protocole soit répété autant de fois qu'il y a de bits à envoyer. Dans le cas où le nombre de participants est supérieur à trois, la topologie du réseau n'est pas forcément

un cercle. Un participant peut avoir plus de deux bits secrets, mais le nombre de bits partagés doit toujours être pair pour s'assurer que le résultat soit toujours 0 si personne ne transmet. De plus, le graphe du réseau n'est pas nécessairement complet, mais celui-ci doit être connecté.

Bien que simple, le protocole contient plusieurs défauts :

— Si deux cryptographes paient le repas, le résultat final sera 0. Un seul participant peut transférer des données à la fois.

— Le protocole n'est pas protégé contre les participants malveillants. Rien n'empêche celui-ci de fausser le résultat.

— La connexion entre les participants est déjà établie. En pratique, cette connexion n'est pas facile à réaliser.

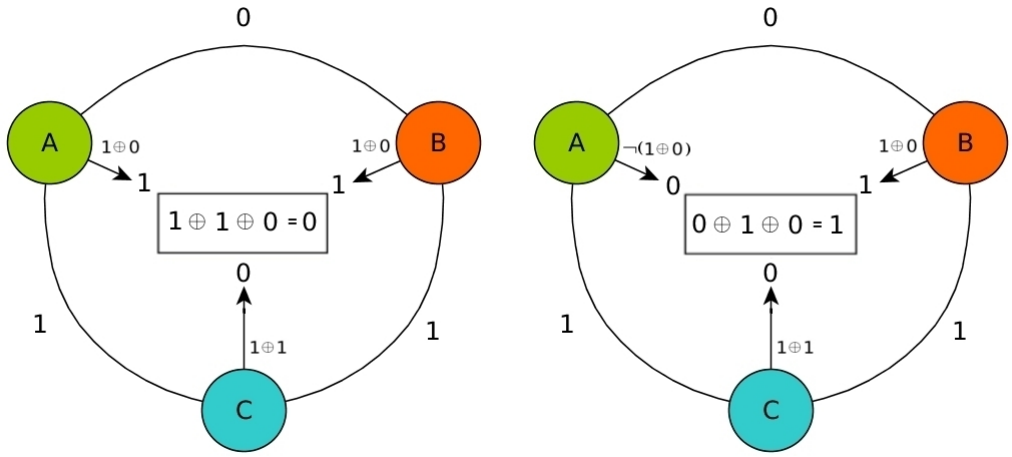

FIGURE 4.13 – Fonctionnement des DCnets : À gauche la NSA paie. À droite A paie.

Pour désanonymiser un participant, tous les participants partageant un bit secret avec lui doivent le fournir à l'attaquant.

Plusieurs solutions ont été proposées pour combler les défauts du protocole et en particulier pour se protéger des participants malicieux. Les projets les plus connus étant Herbivore[60] et Dissent[61].

L'apport important du DCnet est sa non-interactivité. Il n'est pas nécessaire aux participants d'échanger un message directement. Les mix-networks ne peuvent pas faire cela. Néanmoins, l'architecture complexe des DCnets et surtout la facilité pour n'importe quel utilisateur malveillant de perturber la communication font que ce type de réseau est peu utilisé aujourd'hui.

4.7 Comparaison

La particularité de JAP est que les relais sont préétablis et les mixes connus de tous. Tous les utilisateurs passent par les mêmes relais. Cette centralisation en fait à la fois son point fort et son point faible. Les mixes sont authentifiés et sont donc plus sûrs. Étant donné qu'il n'y a pas d'algorithme de sélection de nœuds, un attaquant ne peut pas corrompre le réseau, mais celui-ci est plus vulnérable au déni de service. En effet, il suffit qu'un seul des mixes ne fonctionne plus pour rendre le relais inutilisable. Il existe assez peu de recherche récente sur ce réseau anonyme. Néanmoins, le projet est toujours actif et maintenu par JonDos.

Mixmaster est le seul réseau anonyme à faible latence étudié dans ce mémoire. De ce fait son application se limite à l'envoi de courriel. Il est aussi le seul à mettre en place une stratégie d'éviction et par conséquent de contrer un bon nombre d'attaques. Néanmoins, la dernière version du protocole date de 2005 et sa dernière implémentation date de 2008. Comme pour JAP, il y a peu de recherches récentes à propos de ce réseau anonyme et bien qu'il soit toujours utilisable, il ne peut pas être considéré comme sûr.

L'essentiel des recherches récentes portent sur TOR et I2P, en particulier TOR. [62] et [63] comparent ses deux réseaux anonymes sur plusieurs points :

— Les circuits de TOR sont construits à partir de nœuds volontaires tandis que I2P se base sur les performances de ses nœuds pour construire ses tunnels. TOR utilise l'interface SOCKS permettant le support d'un bon nombre d'applications sans que celles-ci aient besoin d'être modifiées. I2P fait office de middleware et fournit sa propre API pour communiquer dans le réseau nécessitant une modification des applications pour que celle-ci puisse communiquer dans le réseau. Néanmoins, SOCKS ne permet de transférer les messages qu'avec TCP tandis que I2P permet l'utilisation d'UDP ce qui permet de fournir de meilleures performances pour certaines applications.

— I2P étant un réseau cloisonné, la grande majorité de ses applications ne peuvent être accessibles qu'à travers son réseau. TOR quant à lui permet d'accéder à la majorité des applications disponibles sur internet.

— Dans TOR, le nœud d'entrée est le seul à connaître l'identité de l'utilisateur et le nœud de sortie est le seul à connaître le destinataire. Dans I2P, un nœud ne peut pas savoir si les messages qu'il transmet, proviennent directement d'un utilisateur ou d'un autre nœud dans le tunnel.

— L'architecture de TOR le rend plus vulnérable à la congestion. Un faible nombre de nœuds d'entrée et de sortie disponibles peut causer un ralentissement du trafic si ceux-ci sont surchargés. La décentralisation complète d'I2P rend le réseau plus robuste de ce point de vue là.

— Les serveurs d'annuaire de TOR sont authentifiés et au nombre de neuf. Une collusion de tous les serveurs mettrait en danger l'anonymat des utilisateurs. netDb d'I2P est

décentralisé, une collusion n'est donc pas possible.

Pour conclure, TOR est conçu pour accéder anonymement à l'internet public, en particulier les pages web et offre de meilleures performances qu'I2P en général. Tandis qu'I2P offre un anonymat plus fort. Un autre point important, il n'existe plus de système d'exploitation conçu pour I2P contrairement à Tails de TOR.

4.8 Conclusion

Nous avons vu les différences entre plusieurs réseaux anonymes. Chacun offre un service et un niveau d'anonymat différents. Beaucoup sont restés à la phase d'expérimentation ou ne sont plus maintenus depuis plusieurs années. Les solutions les plus sûres actuellement sont TOR et I2P. Bien qu'ils ne protègent pas des attaques par intersection, les mécanismes de défense mis en place demanderont à l'attaquant des moyens importants pour désanonymiser ses utilisateurs. La recherche se concentre principalement sur ces deux réseaux anonymes et régulièrement de nouvelles attaques aparaissent. On peut citer les articles [64] et [65] qui se base sur des machines d'apprentissage pour analyser le trafic de ses réseaux anonymes. Il existe un grand nombre de réseaux anonymes qui n'ont pas été étudiés dans ce mémoire. Par exemple, les réseaux anonymes conçus pour partager les fichiers anonymement. Les plus connus sont Freenet[66] et GNUnet[67]. On peut aussi mentionner le projet Riffle[68], développé par le MIT censé être dix fois plus rapide que le réseau TOR et où son trafic ne pourrait être analysé.

Chapitre 5

Rémunérer les utilisateurs des réseaux anonymes

5.1 Introduction

La principale défense d'un réseau anonyme est sa taille. Plus le nombre de nœuds sur le réseau est grand, plus il est difficile pour un attaquant de contrôler et de surveiller le réseau. Les relais TOR sont constitués de volontaires. Ceux-ci ne gagnent rien en mettant à disposition leur machine. Au contraire, leur bande passante diminue et leur consommation électrique augmente. L'article [29] a brièvement mentionné l'idée d'inciter les internautes à participer au réseau.

Une approche développée dans [69] consiste à distinguer les relais du réseau ayant un bon comportement avec un flag particulier nommé "Gold Star". Le trafic de ces relais est ensuite privilégié. Le problème de cette approche est que l'ensemble du réseau serait divisé en deux et cela diminuerait le niveau d'anonymat du réseau.

[70] propose d'utiliser une monnaie électronique où les clients paient pour que leur trafic soit priorisé. Étant donné que tout le monde peut acheter de la monnaie électronique, il est difficile de discerner ceux qui le dépensent dans le réseau. Ce genre de système doit faire en sorte de limiter les abus et de conserver l'anonymat des utilisateurs. Dans le modèle de [70], l'argent provient d'une seule banque centralisant davantage TOR. Les utilisateurs doivent acheter de la monnaie virtuelle avec leur argent réel, ce qui peut permettre des corrélations. Par exemple, si un utilisateur achète de la monnaie virtuelle et dépense cet argent peu de temps après dans le réseau, un attaquant pourrait remonter jusqu'à l'utilisateur.

Une autre approche consiste à payer les relais avec des tickets. [71] et [72] utilisent cette approche, mais leur défaut est de les générés à partir d'une banque centrale. [73] se base sur ces deux modèles, mais décentralise la distribution des tickets. Au lieu d'utiliser une seule

banque, [73] répartit la distribution des tickets sur plusieurs serveurs.

[74] utilise une monnaie virtuelle conçue pour TOR nommée TORcoin. La génération de monnaie est basée sur la bande passante des nœuds du réseau. Un mécanisme de mesure de bande passante nommé "TorPath" permet d'éviter que des utilisateurs puissent générer de la monnaie sans fournir de bande passante au réseau. TorPath ne permet pas aux clients de choisir leur circuit eux-mêmes, ceux-ci sont construits à partir de serveur d'assignation centralisant TOR. De plus, la bande passante n'est plus prise en considération pour construire les circuits ce qui faciliterait les attaques par les ressources.

Dans ce chapitre, nous allons voir le système de paiement que propose [1]. Un relais TOR rejoint un mining pool et transmet le travail à effectuer à ses clients. En échange, le relais leur fournit des tickets qui leur permettront un accès prioritaire sur ce relais. Ce système de paiement ne fonctionne que sur TOR et nous verrons dans la seconde partie de ce chapitre comment l'adapter à I2P.

5.2 Paiement par preuve de travail sur TOR

5.2.1 Objectifs

[1] propose un système de paiement qui répond aux objectifs suivant :

1. Le système de paiement ne dégrade pas l'anonymat des utilisateurs.
2. Les utilisateurs ne font aucun paiement direct.
3. Le système de paiement n'est pas basé sur des mesures de la bande passante.
4. Il n'y a pas de banque centrale.
5. L'utilisateur n'a pas besoin de devenir un relais pour obtenir un meilleur service.

5.2.2 Fonctionnement

Les utilisateurs fournissent une preuve de travail à un relais TOR. L'utilisateur reçoit par la suite un ticket de priorité propre au relais qui lui permet de bénéficier d'une meilleure bande passante ou d'une meilleure latence. Les tickets sont émis par des relais utilisant des signatures aveugles partielles et envoyés aux utilisateurs à travers des circuits. La signature aveugle [75] consiste à signer un document masqué. Ainsi le signataire ne peut en voir le contenu. Mais le signataire ne peut pas ajouter d'attributs supplémentaires tel un "timestamp". [76] a créé la signature aveugle partielle qui permet de faire cela.

Le système fonctionne sur plusieurs phases :

La phase d'initialisation

Le relais choisit un "mining pool", la cryptomonnaie à utiliser et un algorithme de preuve de travail. Ensuite, le relais génère une paire de clés privées/publiques pour la génération des tickets. Les informations sont ajoutées à son descripteur.

La création de ticket

Le protocole d'achat fonctionne sur plusieurs étapes :

1. Le client établit un circuit TOR vers le relais. Tout le reste de la communication passera par ce circuit.
2. Le client s'enregistre pour un nouveau travail de minage. Le relais lui répond avec l'algorithme de preuve de travail à utiliser, la difficulté de la "part" et suffisamment de données pour construire une part.
3. Le client peut commencer à miner.
4. Si la "part" est résolue, le client génère un nombre aléatoire x et son hash H(x). Il envoie la part résolue w avec H(x) au relais.
5. Le relais vérifie w et l'envoi dans la "mining pool".
6. Le relais génère une signature aveugle partielle S de H(x) avec un timestamp d du jour actuel. Ensuite, le relais envoie S et d au client.
7. Le client peut maintenant utiliser le ticket constitué des éléments S,d,x et H(x).

Une fois le ticket soumis au relais, le client peut bénéficier de la priorité pendant un certain temps. Pour éviter la réutilisation des tickets, le relais conserve un historique des tickets dépensés et pour limiter la taille de cet historique les tickets expirent au bout de 48 heures.

Groupe de priorité

La bande passante d'un relais est répartie suivant une hiérarchie de groupes. D'un côté les "payeurs" (ceux qui ont fourni un ticket) et de l'autre les "non-payeurs". Il peut exister plusieurs sous-groupes de payeurs avec différents niveaux de priorité. Chaque groupe a un ordre de priorité et doit respecter un niveau minimum et maximum de bande passante. Par exemple, le relais peut fournir jusqu'à 10 Mo/s de bande passante. Le groupe de "non-payeur" a une bande passante minimum de 8 Mo/s et un maximum de 10 Mo/s réparti dans deux sous-groupes, le groupe 1 avec un minimum 5 Mo/s et un maximum de 10 Mo/s et le groupe 2 avec 3 Mo/s minimum et 10 Mo/s maximum. Les utilisateurs sont affectés à différents groupes suivant

Relais de secours

Au cas où un relais ne serait plus disponible, le client peut dépenser son ticket dans un "relais de secours". Pour se protéger d'une collusion entre le relais et un client qui produirait des tickets gratuits, le relais de secours demande un paiement à chaque N tickets servis. Pour éviter les doubles dépenses, les bases de données des deux relais sont régulièrement synchronisées.

Implémentation dans TOR

Pour implémenter ce système de paiement, le protocole de TOR a besoin d'être modifié. Trois nouvelles cellules sont à ajouter :

— RELAY MINING REGISTER : permet au client de demander de miner pour un relais.

— RELAY MINING JOB : permet à un autre relais d'envoyer le travail de minage à un client.

— RELAY TICKET : permet au relais d'envoyer les données nécessaires pour que le client puisse construire un ticket et pour avertir les relais de secours des tickets dépenser.

Bien sûr, les logiciels gérant le minage des cryptomonnaies devront être installés autant chez le client que chez le relais.

Analyses

L'article [1] a évalué les performances de son système de paiement sur différentes cryptomonnaies avec des algorithmes de preuves de travail compatibles avec le système. Le montant du profit d'un relais dépend du consensus établi par le réseau et de la bande passante que peut fournir le relais. Les relais fournissant la meilleure bande passante (200 Mo/s) peuvent gagner autour de 500 $US par mois.

L'anonymat des clients utilisant ce système n'est pas diminué. Les échanges des parts ainsi que des tickets entre un relais et un client s'effectuent toujours à travers un circuit de relais. La signature en aveugle permet d'empêcher que le relais puisse distinguer un client des autres. Le client n'ayant besoin que de la part à calculer n'a pas besoin d'avoir un compte dans une quelconque cryptomonnaie évitant ainsi tout problème d'anonymat énoncé dans le chapitre 1. Pour éviter qu'un relais puisse connaître le taux de hachage d'un client, [1] recommande de randomisé ce taux.

5.3 Paiement par preuve de travail sur I2P

Pour mettre en place un système similaire sur I2P, il faut prendre en compte les différences suivantes :

— Par défaut, les utilisateurs d'I2P relaient du trafic.

— Le réseau est fermé et n'est pas conçu pour accéder à internet contrairement à TOR.

5.3.1 Banque et ticket universel

La première étape du protocole est le choix du mining pool. Sur TOR, étant donné que les relais sont publics, ils peuvent rejoindre n'importe quel mining pool sur Internet. Les nœuds d'I2P doivent conserver leur anonymat et rejoindre un mining pool à l'extérieur du réseau peut réduire leur anonymat.

Une solution serait d'utiliser certains nœuds d'I2P comme relais vers un mining pool extérieur à I2P. Seuls les opérateurs risqueraient une perte d'anonymat.

Nous avons dit plus haut que dans le cas où un relais n'est pas disponible et où un utilisateur souhaite dépenser ses tickets immédiatement, [1] propose d'employer un relais de secours. Sur I2P, les utilisateurs sont plus volatils que les relais de TOR. Il serait idéal que tout ticket généré puisse être dépensé par n'importe quel routeur. Pour ce faire, il faut prendre en compte les contraintes suivantes :

1. Un routeur qui loue sa bande passante doit obtenir les gains issus du ticket généré.
2. Les doubles dépenses doivent être évitées.

Pour décrire la suite du fonctionnement du système de paiement, nous utiliserons la terminologie suivante :

— Client : Routeur I2P souhaitant bénéficier d'une meilleure bande passante.

— Routeur : Routeur I2P fournissant de la bande passante.

— Banque : Routeur I2P connecté à une mining pool à l'extérieur d'I2P.

Le protocole d'achat de ticket de [1] n'a pas besoin d'être modifié pour I2P, à la différence près que l'achat du ticket se fait auprès d'une banque au lieu du routeur. Le protocole peut être décrit comme suit :

1. Un client établit une connexion avec des tunnels à une banque.
2. La banque fournit le travail au client de la même façon que le protocole 1 de [1].
3. Une fois que le client a résolu la part, il l'envoie à la banque un nombre aléatoire avec son hachage comme pour le protocole 1 de [1].
4. La banque vérifie la part et s'il est correct, elle le transmet à la mining pool.
5. Une fois les gains reçus, la banque conserve les gains et génère un ticket de la même façon que le protocole 1 de [1] et le transfère au client. La banque conserve dans sa base de données une copie du ticket. Les bases de données des différentes banques sont constamment synchronisées.

La figure 5.1 illustre ces étapes.

Le protocole d'échange de tickets s'effectue avec les étapes suivantes :

1. Le client cherche dans netDB un routeur disponible et lui envoie le ticket.

2. Le routeur transfère le ticket à une banque. Celle-ci vérifie dans sa base de données si le ticket existe et s'il n'est pas expiré. Si c'est le cas, elle transfère les gains associés au ticket au routeur.
3. Le routeur peut ensuite donner une priorité d'accès au client.

Dans ce modèle, la banque agit en tant que tiers de confiance. Celle-ci prend une commission sur chaque gain transféré aux routeurs. La figure 5.2 illustre ces étapes.

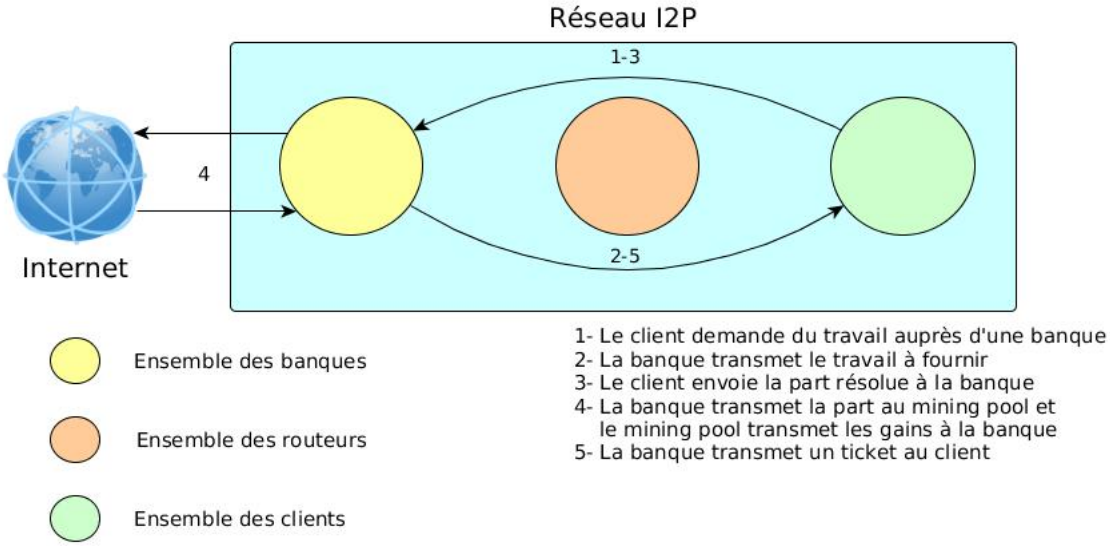

FIGURE 5.1 – Création d'un ticket universel

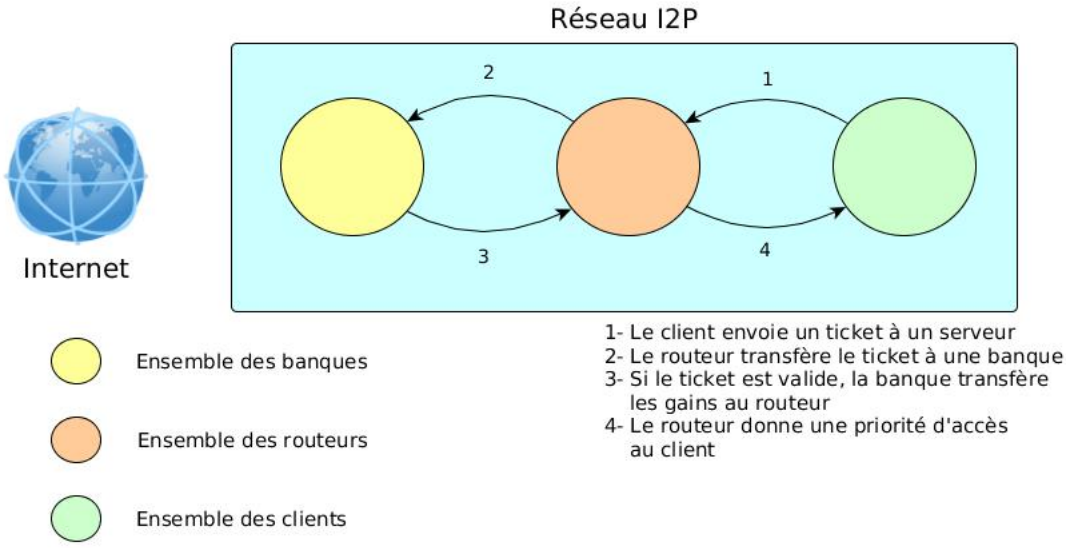

FIGURE 5.2 – Échange de ticket

Pour que la bande passante reste la même du client jusqu'au destinataire, le tunnel entrant du destinataire doit avoir la même bande passante que le tunnel sortant du client. Le client a le contrôle sur la construction de ses tunnels, mais pas sur celui du destinataire. Une solution serait que le client fournisse des tickets au destinataire pour qu'il puisse construire son tunnel entrant avec une bande passante équivalent au tunnel sortant du client.

5.3.2 Défaut et inconvénient

Cette approche présente plusieurs défauts :

1. Elle réduit le niveau d'anonymat des banques.
2. Le système de paiement est centralisé sur les banques. Sans les banques ou avec des banques malhonnêtes, le système ne peut pas fonctionner. Il faut faire confiance aux banques. Tous les gains passent par eux.
3. Elle réduit l'anonymat des routeurs. Les gains étant fournis par les banques, ceux-ci pourront plus facilement lier les dépenses des routeurs dans la blockchain. Une solution pour éviter cela serait que les banques convertissent les cryptomonnaies en une cryptomonnaie anonymes comme Zerocoin[77]. Cette conversion n'est pas rendue possible sur toutes les cryptomonnaies et de plus elle engendrera des frais de transaction.

Le principal avantage de cette approche est qu'elle permet le choix d'un grand nombre de mining pool disponibles sur Internet et les tickets sont utilisables pour n'importe quel routeur.

5.3.3 Anoncoin

Une solution pour conserver le niveau d'anonymat des banques et des relais serait d'utiliser des cryptomonnaies supportées par I2P. Pour le moment, seul Anoncoin[78] est supporté. Cette cryptomonnaie est un fork du Litecoin[79]. Son fonctionnement est similaire au Bitcoin à la différence qu'elle permet les transactions à travers I2P et TOR. Comme pour le Bitcoin, les transactions sont publiées dans la blockchain. Pour régler tout risque de traçage sur la blockchain, les développeurs de l'Anoncoin prévoient d'implémenter Zerocoin[77].

L'avantage de cette approche est qu'étant donné que tout le trafic ne sort pas d'I2P, le niveau d'anonymat est conservé pour tous les nœuds du réseau. En contrepartie, les utilisateurs n'ont pas d'autre choix que d'utiliser Anoncoin.

5.4 Conclusion

Le système de paiement de [1] basé sur la preuve de travail des cryptomonnaies permet de rémunérer les relais du réseau TOR tout en conservant l'anonymat de leur client. Dans ce domaine, la recherche se concentre essentiellement sur le réseau TOR. Aucun système équivalent n'existe sur I2P. Dans ce chapitre, le système de paiement de [1] a été adapté pour

I2P. Contrairement à sa version sur TOR, les tickets peuvent être utilisés par n'importe quel routeur offrant le service. L'utilisation de banques rend cela possible. La banque est le cœur du système de paiement. Il agit en tant qu'intermédiaire entre une mining pool à l'extérieur d'I2P et les clients. Il conserve les gains récupérés dans la mining pool et génère les tickets pour les clients. Les clients peuvent dépenser leurs tickets sur n'importe quel routeur disponible. Ces routeurs récupéreront ensuite les gains auprès d'une banque en fournissant le ticket donné par le client. Bien qu'il permet la dépense des tickets sur n'importe quel routeur, l'inconvénient majeur de ce système de paiement est sa centralisation sur les banques. Ils agissent comme tiers de confiance et sont indispensables au fonctionnement du système. Un autre inconvénient du système est qu'il réduit le niveau d'anonymat des banques étant donné que celles-ci doivent communiquer à l'extérieur du réseau d'I2P. Néanmoins, il est possible d'utiliser la cryptomonnaie Anoncoin. Celle-ci fonctionne sur I2P mais c'est la seule disponible sur ce réseau.

Conclusion

Dans ce mémoire, nous avons énuméré les techniques et technologies permettant de surveiller et protéger les internautes. Nous avons vu que la solution la plus sûre est l'utilisation des réseaux anonymes. Ceux-ci sont conçus pour protéger l'anonymat de leur utilisateur. Les réseaux anonymes sont souvent construits sur le même modèle, celui du relais de proxy et du chiffrement en couche. Les attaques pour désanonymiser ces réseaux peuvent être divisées en deux catégories :

— Les attaques applicatives : Ce genre d'attaque vise à se connecter directement à l'utilisateur et ainsi contourner le réseau anonyme. La plupart des attaques applicatives s'effectuent sur le navigateur web de l'utilisateur. L'attaquant y tente d'exploiter les mauvaises configurations de l'utilisateur. En général, elles sont peu coûteuses pour l'attaquant. Néanmoins, il ne tient essentiellement qu'à l'utilisateur d'employer les bons outils et les bonnes méthodes pour s'en protéger.

— Les attaques au niveau réseau : Contrairement à la première catégorie, l'utilisateur ne peut pas se protéger lui-même contre ces attaques. Ce sont les concepteurs du réseau qui sont responsables de cela. Les attaques de cette catégorie sont souvent difficiles à mettre en œuvre et demande beaucoup de moyens.

Nous avons étudié quatre réseaux anonymes en profondeur dans ce mémoire : JAP, Mixmaster, TOR et I2P. Mixmaster est le seul réseau anonyme à faible latence étudié dans ce mémoire. Il se limite à l'envoi de courriels anonymes. Le projet est à l'abandon depuis de nombreuses années, il est donc aujourd'hui peu recommandé de l'utiliser. JAP contrairement à TOR ou I2P, utilise des routes fixes et préétablies dans ses relais. Ses mixes sont authentifiés et publics. TOR et I2P sont les deux réseaux anonymes les plus utilisés. La plus grande différence entre les deux est que TOR permet d'accéder à internet tandis que le réseau d'I2P est fermé. I2P est complètement décentralisé, ce qui n'est pas le cas de TOR et ses serveurs d'annuaire. Pour finir, tout nœud d'I2P peut transférer du trafic d'un autre nœud alors que sur TOR, seuls les volontaires peuvent devenir des relais et ceux-ci sont publics.

Nous avons vu que la principale défense des réseaux anonymes est leur nombre d'utilisateurs. Plus celui-ci est grand, plus il est difficile pour un attaquant d'isoler le trafic d'un utilisateur. [1] a développé un système de rémunération sur TOR dans le but d'encourager les internautes

à devenir des relais. Le concept repose sur l'utilisation de cryptomonnaie pour rémunérer les relais. Un utilisateur souhaitant un trafic priorisé sur le réseau va miner pour le relais. Une fois que le relais a empoché les gains de ce minage, celui-ci donne un ticket à l'utilisateur qui peut le dépenser pour bénéficier d'une meilleure bande passante sur ce relais. La contribution de ce mémoire est l'adaptation de ce protocole sur le réseau anonyme I2P.

Bibliographie

[1] A. Biryukov and I. Pustogarov. Proof-of-work as anonymous micropayment : Rewarding a tor relay. *Financial Cryptography and Data Security - 19th International Conference*, 2015.

[2] A. Yanes. Privacy and anonymity. *CoRR*, abs/1407.0423, Juillet 2014.

[3] K. Peng. *Anonymous Communication Networks*, chapter Anonymity in Network Communication. CRC Press, 2014.

[4] M. C. Libicki L. Ablon and A. A. Golay. *Markets for Cybercrime Tools and Stolen Data : Hackers' Bazaar*. RAND Corporation, 2014.

[5] R. Kang L. Rainie, S. Kiesler and M. Madden. Anonymity, privacy, and security online. *Pew Research Center*, Septembre 2013.

[6] S. Brown R. Kang and S. Kiesler. Why do people seek anonymity on the internet ? informing policy and design. *SIGCHI Conference on Human Factors in Computing Systems*, pages 2657–2666, Avril 2013.

[7] M. Goncharov V. Ciancaglini, M. Balduzzi and R. McArdle. Deepweb and cybercrime : It's not all about tor. *Trend Micro*, 2013.

[8] Internet World Stats. World internet users statistics and 2016 world population stats. http://www.internetworldstats.com/stats.htm, 2016.

[9] P. Maymounkov and D. Mazières. Kademlia : A peer-to-peer information system based on the xor metric. *IPTPS : International Workshop on Peer-to-Peer Systems*, 2002.

[10] Welcome to p2pool ! http://p2pool.org/learn/index.php.

[11] Protect your privacy. https://bitcoin.org/en/protect-your-privacy.

[12] M. C. K. Khalilov and A. Lev. A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys and Tutorials*, pages 2543–2585, 2018.

[13] Internet Engineering Task Force. Rfc 4949 : Internet security glossary, version 2, 2007.

[14] Internet Engineering Task Force. Rfc 6973 : Privacy considerations for internet protocols, 2014.

[15] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization : Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[16] M. Ermert C. Grothoff, M. Wachs, H. Verlag, and J. Appelbaum. Nsa's morecowbell : Knell for dns. 2015.

[17] Internet Engineering Task Force. Rfc 7626 : Dns privacy considerations, 2015.

[18] J. Heidemann L. Zhu, Z. Hu, D. Wessels, A. Mankin, and N. Somaiya. T-dns : connection-oriented dns to improve privacy and security. *SIGCOMM '14 Proceedings of the 2014 ACM conference on SIGCOMM*, pages 379–380, Août 2014.

[19] Internet Engineering Task Force. Rfc 7721 : Security and privacy considerations for ipv6 address generation mechanisms, 2016.

[20] C. Zachor E. Erdin and M. H. Gunes. How to find hidden users : A survey of attacks on anonymity networks. *IEEE Communication surveys & tutorials, vol. 17, no. 4*, 2015.

[21] Electronic Frontier Foundation and Tor project. Https everywhere. https://www.eff.org/https-everywhere.

[22] Eyeo GmbH. Adblock plus. https://adblockplus.org/.

[23] Evidon. Ghostery. https://www.ghostery.com/.

[24] ublock origin. https://github.com/gorhill/uBlock.

[25] Noscript. https://noscript.net/.

[26] Electronic Frontier Foundation. Panopticlick. https://panopticlick.eff.org/.

[27] Ip check. http://ip-check.info/?lang=en.

[28] Tor browser. https://www.torproject.org/projects/torbrowser.html.en.

[29] N. Mathewson R. Dingledine and P. F. Syverson. Tor : The second-generation onion router. *USENIX SecuritySymposium*, pages 303–320, 2004.

[30] The invisible internet project. https://geti2p.net.

[31] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communication of the ACM*, 24 :84–88, 1981.

[32] K. Sampigethaya and R. Poovendran. A survey on mix networks and their secure applications. *Proceedings of the IEEE*, 94 :2142–2181, 2016.

[33] F. Shirazi, M. Simeonovski, M. Rizwan Asghar, M. Backes, and C. Díaz. A survey on routing in anonymous communication protocols. *CoRR*, abs/1608.05538, Août 2016.

[34] Modèle de menace d'i2p. https://geti2p.net/fr/docs/how/threat-model.

[35] M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack : An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Security, vol. 4, no. 7*, pages 489–522, 2012.

[36] You should let people choose their path length. https://www.torproject.org/docs/faq.html.en#ChoosePathLength.

[37] Tunnel length. https://geti2p.net/fr/docs/how/tunnel-routing#length.

[38] D. Hopwood R. Dingledine, M. Freedman and D. Molnar. A reputation system to increase mix-net reliability. *In I. S. Moskowitz, editor, Information Hiding (IH 2001)*, pages 126–141, 2001.

[39] Project : An.on - anonymity.online. https://anon.inf.tu-dresden.de/index_en.html.

[40] Jondonym. https://anonymous-proxy-servers.net/index.html.

[41] H. Federrath O. Berthold and S. Köpsell. Web mixes : A system for anonymous and unobservable internet access. *In Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA*, page 115–129, 2000.

[42] P. Palfrader U. Moller, L. Cottrell and L. Sassaman. Mixmaster protocol version 2. https://tools.ietf.org/html/draft-sassaman-mixmaster-03#section-1.

[43] R. Christman. Quicksilver lite. https://www.quicksilvermail.net/.

[44] R. Dingledine G. Danezis and N. Mathewson. Mixminion : Design of a type iii anonymous remailer protocol. *ISecurity and Privacy, 2003. Proceedings. 2003 Symposium on*, 2003.

[45] N. Mathewson. Mixminion : A type iii anonymous remailer. https://www.mixminion.net/.

[46] Tormetrics. https://metrics.torproject.org/.

[47] Tor : Overview. https://www.torproject.org/about/overview.html.en.

[48] Tips for running an exit node. https://blog.torproject.org/running-exit-node.

[49] J. Caverlee D. S. Rudesill and D. Sui. The deep web and the darknet : A look inside the internet's massive black box. *Woodrow Wilson International Center for Scholars, STIP 03, Ohio State Public Law Working Paper No. 314*, Octobre 2015.

[50] Deep Web Technologies. The "deep web" is not all dark. http://www.deepwebtech.com/deepweb-not-darkweb/, 2016.

[51] Tor : Hidden service protocol. https://www.torproject.org/docs/hidden-services.html.en.

[52] Atlas. https://atlas.torproject.org/#search/flag:authority.

[53] Tails - confidentialité et anonymat, pour tout le monde et partout. https://tails.boum.org/index.fr.html.

[54] stats.i2p. http://stats.i2p.

[55] i2np. https://geti2p.net/sv/docs/protocol/i2np.

[56] M. K. Reiter and A. D. Rubin. Crowds : Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur., vol. 1*, pages 66–92, 1998.

[57] M. J. Freedman, E. Sit, J. Cates, and R. Morris. Introducing tarzan, a peer-to-peer anonymizing network layer. *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge*, page 121–129, 2002.

[58] M. J. Freedman and R. Morris. Tarzan : A peer-to-peer anonymizing network layer. *Proceedings of the 9th ACM Conference on Computer and Communications Security*, page 193–206, 2002.

[59] D. Chaum. The dining cryptographers problem : Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1988.

[60] S. Goel, M. Robson, M. Polte, , and E. Sirer. Herbivore : A scalable and efficient protocol for anonymous communication. *Cornell University*, 2003.

[61] H. Corrigan-Gibbs and B. Ford. Dissent : Accountable anonymous group messaging. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 340–350, 2010.

[62] B. Conrad and F. Shirazi. A survey on tor and i2p. *ICIMP 2014 : The Ninth International Conference on Internet Monitoring and Protection*, 2014.

[63] N. Negi. Comparison of anonymous communication networks-tor, i2p, freenet. *International Research Journal of Engineering and Technology Volume : 04 Issue : 07*, Juillet 2017.

[64] K. Shahbar and A. N. Zincir-Heywood. How far can we push flow analysis to identify encrypted anonymity network traffic ? *2018 IEEE/IFIP Network Operations and Management Symposium*, 2018.

[65] K. Shahbar and A. N. Zincir-Heywood. Effects of shared bandwidth on anonymity of the i2p network users. *2017 IEEE Security and Privacy Workshops (SPW)*, 2017.

[66] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet : A distributed anonymous information storage and retrieval system. *Workshop on Design Issues in Anonymity and Unobservability*, page 311–320, 2000.

[67] K. Bennett, T. Stef, C. Grothoff, T. Horozov, and I. Patrascu. Gnet. *Purdue University*, 2002.

[68] A. Kwon, D. Lazar, S. Devadas, and B. Ford. Riffle : An efficient communication system with strong anonymity. *Proceedings on Privacy Enhancing Technologies 2016*, 2016.

[69] T. J. Ngan, R. Dingledine, and D. S. Wallach. Building incentives into tor. *International Conference on Financial Cryptography and Data Security FC 2010 : Financial Cryptography and Data Security*, pages 238–256, 2010.

[70] E. Androulaki, M. Raykova, S. Srivatsan, A. Stavrou, and S. M. Bellovin. Par : Payment for anonymous routing. *Proceedings of the 8th international symposium on Privacy Enhancing Technologies*, pages 219–236, 2008.

[71] R. Jansen, N. Hopper, and Y. Kim. Recruiting new tor relays with braids. *Proceedings of the 17th ACM conference on Computer and communications security*, pages 319–328, 2010.

[72] J. Rob, J. Aaron, and S. Paul. Lira : Lightweight incentivized routing for anonymity. *Proceedings of the Network and Distributed System Security Symposium - NDSS'13*, 2013.

[73] R. Jansen, A. Miller, P. Syverson, and B. Ford. From onions to shallots : Rewarding tor relays with tears. 2014.

[74] M. Ghosh, M. Richardson, B. Ford, and Rob Jansen. A torpath to torcoin : Proof-of-bandwidth altcoins for compensating relays. *7th Workshop on Hot Topics in Privacy Enhancing Technologies*, 2014.

[75] D. Chaum. Blind signatures for untraceable payments. *Advances in Cryptology Proceedings of Crypto, vol. 82, no 3*, 1983.

[76] M.Abe and T. Okamoto. Blind signatures for untraceable payments. *CRYPTO 2000 : Advances in Cryptology — CRYPTO 2000*, pages 271–286, 2000.

[77] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin : Anonymous distributed e-cash from bitcoin. *IEEE Symposium on Security and Privacy (Oakland) 2013*, 2013.

[78] Anoncoin. https://anoncoin.net/.

[79] Litecoin. https://litecoin.org/.