Faculté des Sciences

UMONS CHARLEROI

Université de Mons

Faculté des Sciences

US-MC-INFO60-014-C — Lecture et rédaction scientifique

TOR:
Le Routage Onion de Seconde Génération

Auteur:
Baptiste Grosjean

Directeur:
Alain Buys

Numéro d'étudiant:
232732

Résumé

The Onion Router (TOR) est un système de communication sécurisé et fiable. Ce travail de rédaction scientifique vise à fournir une compréhension détaillée du protocole TOR, de son fonctionnement et de son importance dans les réseaux modernes. Nous utiliserons LaTeX pour produire une documentation claire et précise du protocole ainsi que de son implémentation pratique.

Juin 2024

Sommaire

1 Introduction **3**

2 Description **4**

3 Conception **6**

3.1 Cellules . . . 6

3.2 Circuits . . . 7

4 Sécurité **8**

4.1 Données . . . 9

4.1.1 Cryptographie symétrique . . . 9

4.1.2 Cryptographie asymétrique . . . 9

4.1.3 Comparaison cryptographie symétrique et asymétrique . . . 9

4.2 Communications . . . 11

4.2.1 Diffie-Helman . . . 11

4.2.2 TLS . . . 11

5 Fonctionnement **12**

5.1 Initialisation . . . 12

5.2 Construction . . . 12

5.3 Utilisation . . . 12

6 V3 Onion Services **13**

7 Conclusion **14**

A Latence et anonymat **15**

B Menaces **16**

B.1 Passive . . . 16

B.2 Active . . . 17

B.3 Annuaire . . . 18

B.4 Points de rendez-vous . . . 18

Figures **19**

Tableaux **20**

Algorithmes **21**

Glossaire **22**

1 Introduction

À l'aube du XXIe siècle, jamais les aspects liés à la sécurité de nos interactions en ligne n'auront été aussi importants. Jamais nos moindres faits et gestes n'auront été autant traqués et analysés. Jamais nos données personnelles n'auront été aussi précieuses. Jamais nos vies privées n'auront été aussi publiques. Jamais notre sécurité en ligne n'aura autant impacté notre sécurité hors ligne.

Que ce soit lors d'un virement en ligne, lors de téléchargements de contenus ou encore lors de recherches, tous nos usages se voient traqués et analysés. Tous sans exception voient nos données personnelles s'envoler à jamais dans les méandres du réseau internet mondial. Tous sans exception voient nos vies privées rendues publiques à l'instant où nous nous connectons. Si ces inconforts mettent en lumière les failles des services dont on ne sait plus se passer, ils ne mettent cependant pas nos vie en danger, ni ne restreignent notre accès à l'information. Cela est dû au fait que nous nous connectons depuis les lieux les plus sûrs au monde, hors ligne.

Qu'en est-il des citoyens de régimes répressifs ? Des journalistes opérant depuis des zones à risque ? Des personnes devant transmettre des informations sensibles via internet ? C'est bien ici, pour ces usages particuliers, que l'existence même de TOR se justifie. Pour le citoyen Russe souhaitant accéder à des sources d'informations censurées, pour le journaliste opérant depuis une zone sous conflit armé, pour le lanceur d'alerte ne souhaitant pas être inquiété, pour les besoins militaires nécessitant des transferts de données fiables et intraçables. C'est bien dans ces contextes si particuliers que le routage onion de seconde génération ainsi que les services qu'il propose se révèlent cruciaux.

Le protocole The Onion Router (TOR) est un réseau informatique décentralisé permettant l'anonymat, l'intégrité et la non vulnérabilité des données transmises et rendant le traçage sur le réseau internet mondial complexe. En permettant une navigation anonyme et en protégeant les données contre l'analyse de trafic, TOR joue un rôle crucial dans la préservation tant de la vie privée en ligne que de la non-vulnérabilité des données transmises. Et ce, tant pour des applications civiles que pour des usages plus sensibles liés à la sécurité nationale et à la protection des sources journalistiques.

Ce travail se propose d'explorer en profondeur le protocole TOR, en mettant en lumière non seulement son architecture de routage en oignon de deuxième génération mais aussi son évolution depuis sa création. À travers une analyse rigoureuse de l'article fondateur "TOR: The Second-Generation Onion Router" [1, Tor] de Dingledine, Mathewson, et Syverson, ce rapport aspire à offrir une compréhension nuancée des mécanismes qui sous-tendent le fonctionnement de TOR ainsi que les défis sécuritaires qu'il cherche à surmonter, contribuant ainsi à une meilleure appréciation de son impact sur les paradigmes de sécurité et d'anonymat sur Internet.

Outre cette introduction, la section 2 "Description" présentera une vue d'ensemble de TOR, posant le contexte. La section 5 "Fonctionnement" décrira le fonctionnement du protocole de manière détaillée. La section 4 "Sécurité" examinera les aspects sécuritaires liés à TOR, tandis que la section 6 "V3 Onion Services" envisagera les perspectives d'évolution du protocole. Enfin, la section 7 "Conclusion" conclura en résumant les principales trouvailles et recommandations découlant de cette analyse.

2 Description

L'anonymat en ligne, objet de ce rapport au travers de l'analyse de TOR, va de pair avec une latence accrue en raison des différentes techniques et méthodes mises en place pour y parvenir. Si TOR et le routage oignon sur lequel il repose font partie des systèmes dits de faible latence, les premiers à offrir l'anonymat appartenaient à une seconde catégorie dite à forte latence.

Prémices de l'anonymat (0G)

En 1981, Chaum introduisit la notion de mixnets (réseaux mélangés) [2] afin de permettre aux utilisateurs d'utiliser un réseau sans compromettre leur anonymat. Il était alors question de redéfinir l'ordre dans lequel les messages étaient transmis à travers le réseau: un message pouvait arriver en premier dans un noeud, mais être transmis après les 3 messages suivants. Ce système permettait d'empêcher le suivi des données entre noeuds. Si l'anonymat était garanti, des taux de latence particulièrement élevés venaient restreindre son utilité pour des usages nécessitant des temps de réponses réduits. En effet, les mixnets étants basés uniquement sur la Asymetric Cryptography (AC), ils ne sont pas adaptés à des usages interactifs.

Routage en Oignon (1G)

Initialement développé dans les années 1990, le routage en oignon est venu résoudre ces problèmes en anonymisant des applications basées sur TCP, telles que la navigation web et la messagerie instantanée et les connexions SSH avec un taux de latence réduit. Ce système, bien que novateur, présentait plusieurs lacunes significatives, notamment l'absence de sécurité parfaite vers l'avant, qui signifie que la sécurité des données n'était pas garantie si une partie du chemin était compromise après la transmission des données. De plus, il nécessitait des proxies d'application distincts pour chaque protocole de la 7e couche du modèle OSI supporté (*e.g.,* HTTPS, FTP,...). Ce qui a limité sa polyvalence ainsi que son efficacité face à divers types de traffics en ajoutant des couches de traitements supplémentaires.

Evolution vers TOR (2G)

En réponse à ces défis, The Onion Router (TOR) a été introduit en 2002, marquant une évolution majeure du concept original.

> *Onion Routing is a distributed overlay network designed to anonymize TCP-based applications like web browsing, secure shell, and instant messaging.* [1, p. 1, sec. 1]

Il est important de noter que le protocole UDP n'est actuellement pas pris en charge par TOR. En effet, ce dernier ne nécessitant pas de connexion et donc d'accusé de réception, cela le rend intrinsèquement plus complexe à anonymiser sans induire une latence importante qui serait contraire aux objectifs de conception du réseau.

TOR a introduit plusieurs améliorations significatives :

- Sécurité parfaite vers l'avant : Cette fonctionnalité assure que, même si un nœud intermédiaire est compromis, les données transmises antérieurement restent protégées.
- Simplification des proxies d'application : Grâce à l'adoption de l'interface proxy standard SOCKS, TOR supporte plusieurs types de trafic TCP sans nécessiter de modifications logicielles spécifiques, augmentant ainsi sa flexibilité et réduisant la complexité technique.
- Contrôle de congestion décentralisé : TOR améliore la réactivité du système et la gestion de la charge réseau par un mécanisme de contrôle de congestion qui ne nécessite pas de communication inter-nœuds, facilitant ainsi une meilleure scalabilité et performance du réseau.

Les utilisateurs établissent des circuits sécurisés à travers le réseau, où chaque nœud, ou "routeur oignon", ne connaît que le nœud précédent et le suivant. Les données transitent en cellules chiffrées, chaque nœud dévoilant progressivement le chemin jusqu'à la destination finale, ce qui préserve l'anonymat de la source et de la destination.

Défis et Perspectives (3G)

Malgré ses avancées, TOR fait face à des défis continus, notamment en matière de latence réseau et de résistance aux attaques d'analyse de trafic, où des adversaires sophistiqués pourraient théoriquement corréler les modèles de trafic entrant et sortant pour compromettre l'anonymat. Les efforts continus de la communauté pour mettre à jour et améliorer TOR sont cruciaux pour répondre aux défis de l'Internet moderne et maintenir la robustesse du système face aux menaces émergentes.

En résumé, TOR représente une solution robuste et flexible pour l'anonymat en ligne, mais comme tout système, il n'est pas exempt de limitations qui doivent être adressées pour assurer son efficacité à long terme. La Section 6 "V3 Onion Services" concerne la 3e génération du routage oignon.

3 Conception

La communication dans TOR se fait au travers de flux de données composés de cellules et empruntant des circuits prédéfinis.

1. 3.1 Cellules de types contrôle ou relais, leurs fonctions sont spécifiques.
2. 3.2 Circuits construits via des cellules de contrôles, ils transmettent les données via des cellules de relais.

3.1 Cellules

D'une taille fixe de 512 octets, elles se divisent en deux parties principales : un en-tête (header) et une charge utile (payload). L'en-tête contient un identifiant de circuit (circID) et une commande (CMD), comme illustré ci-dessous dans le schéma de la Figure 1.

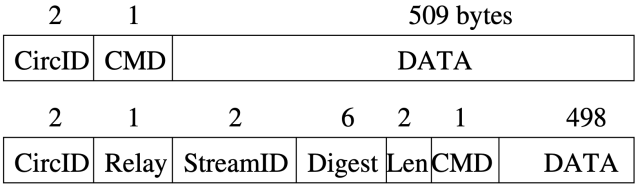

Figure 1: Structure de cellule vs structure d'une cellule de relais. [1, Tor: The Second-Generation Onion Router]

Il existe deux types de cellules distincts:

1. Contrôle : les cellules de contôle sont traitées directement par le noeud qui les reçoit; leurs fonctions sont administratives: initialisation, maintenance et clôture du circuit.
2. Relais : les cellules de relais permettent le transfert de bout en bout des données utilisateur, leurs fonctions concernent le flux de données

Le tableau 1 "Comparaison des Types de Cellules et de leurs commandes dans le Réseau Tor" ci-dessous reprend l'ensemble des différents types de cellules utilisées dans TOR. Les commandes y sont spécifiques aux type dont il est question.

Type	**Commandes**	**Fonctions**
Contrôle	PADDING	Garder la connexion active
	CREATE	Établir un nouveau circuit
	CREATED	Confirmer la création du circuit
	DESTROY	Fermer un circuit
Relais	RELAY_BEGIN	démarrer une connexion TCP
	RELAY_DATA	transfert de données
	RELAY_END	fermer une connexion TCP
	RELAY_CONNECTED	confirmer l'établissement de la connexion TCP
	RELAY_EXTEND	étendre un circuit à un autre nœud
	RELAY_TRUNCATED	signaler une coupure de circuit partielle
	RELAY_SENDME	contrôle de congestion (demande de données)

Table 1 Comparaison des Types de Cellules et de leurs commandes dans le Réseau Tor

3.2 Circuits

Contrairement à la première génération de routage onion, chaque circuit peut multiplexer différents flux TCP. Ces circuits sont reconstruits chaque minute par les OP afin de limiter les liens entre flux (rotation périodique).

Les circuits sont composés de trois types de nœuds:

- Gardien d'entrée
- Noeud(s) intermédiaire(s)
- Noeud de sortie

Sur la Figure 2 "Circuit à deux Routeurs Onions permettant à Alice de joindre un site web.[1, Tor: The Second-Generation Onion Router]" ci-dessous, "OR 1" représente le "Gardien d'entrée" tandis que "OR 2" représente le "Noeud de sortie". Le processus est initié par l'expéditeur, qui utilise des cellules de contrôle afin de construire le circuit. Lorsque la connection avec le premier OR est établie via une cellule de création, le processus continue jusqu'à atteindre la destination finale. Une fois le circuit construit, des cellules de relais le parcourent jusqu'à atteindre la destination et retourner le résultat à l'expéditeur.

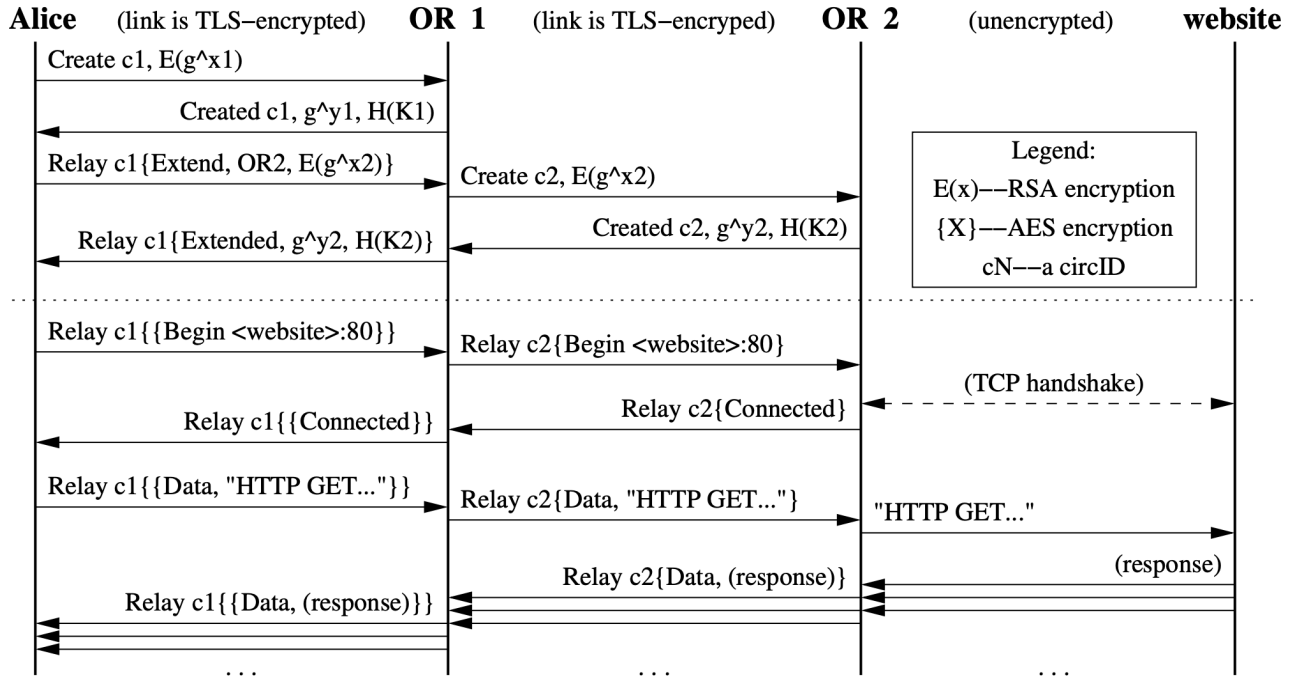

Figure 2: Circuit à deux Routeurs Onions permettant à Alice de joindre un site web.[1, Tor: The Second-Generation Onion Router]

4 Sécurité

La sécurité au sein de TOR s'articule autour d'un réseau de noeuds relais comprenant un noeud d'entrée, des noeuds intermédiaires ainsi qu'un noeud de sortie. Lors de l'initialisation de la connexion entre l'expéditeur et le destinataire, des méthodes cryptographiques avancées (*cfr.*, sec. 4.1 "Données") sont utilisées afin de sécuriser non seulement les données transmises mais également leurs canaux de communications. Le routage en oignon offre l'anonymat en encapsulant les données dans plusieurs couches de chiffrement. Chaque nœud dans le circuit ne peut déchiffrer que la couche lui étant assignée, ne révélant jamais l'identité ni de l'expéditeur ni du destinataire.

Les aspects techniques de la sécurité dans TOR, incluant le routage en oignon, la sélection des relais, les protocoles de chiffrement, les mesures défensives et les politiques de confidentialité, contribuent à sa robustesse comme outil d'anonymat en ligne.

La sécurité dans TOR passe par le chiffrement des données ainsi que par la sécurisation des communications.

1. 4.1 Données chiffrées via cryptographie symétrique et asymétrique
2. 4.2 Communications sécurisées via poignées de mains Diffie-Hellman (DH) et Transport Layer Security (TLS)

La Figure 3 "Modélisation des différentes couches de sécurité dans TOR" ci-dessous illustre les différentes couches de sécurité qui se supperposent dans TOR. Les données étant transmises via des cellules chiffrées via cryptographie symétrique. La cryptographie symétrique (SC) découle d'un processus impliquant la cryptographie asymétrique (AC) elle-même impliquant les poignées de mains Diffie-Hellman (DH). Il est important de noter que les poignées de main DH ainsi que la AC n'interviennent qu'au moment de l'établissement du circuit. Ces dernières utilisent des canaux de communications sécurisés via le protocole TLS.

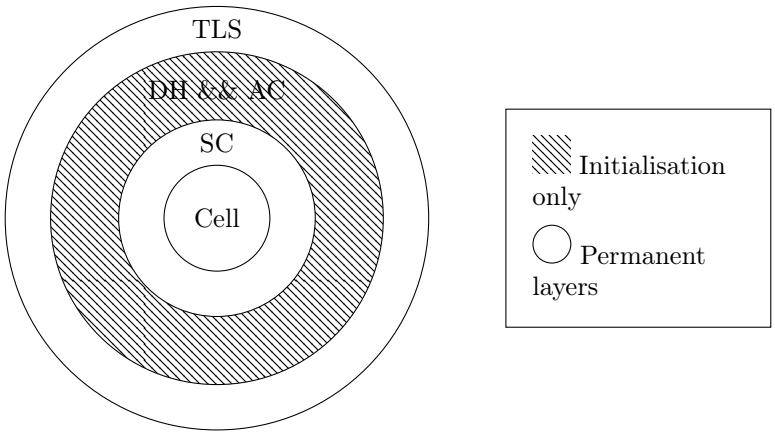

Figure 3: Modélisation des différentes couches de sécurité dans TOR

4.1 Données

La sécurité au niveau des données est assurée par une combinaison de cryptographie asymétrique et symétrique pour sécuriser le contenu.

1. 4.1.1 Cryptographie symétrique Utilisée pour déchiffrer les cellules de données transmises.
2. 4.1.2 Cryptographie asymétrique Utilisée pour l'échange de clés privées.

4.1.1 Cryptographie symétrique

La cryptographie symétrique repose sur l'utilisation de clés privées partagées. Le protocole d'échange de clés Diffie-Hellman permet d'établir la connexion et de générer les clés privées via l'algorithme AES (Advanced Encryption Standard). Ces clés peuvent être comparées à un cadenas: un code protège le cadenas, quiconque possède le code peut ouvrir ce cadenas, cependant si le code venait à être partagé, cela compromettrait la sécurité apportée par ce cadenas.

La Figure 4 "Structure du chiffrement en couches. [3, Layers-of-the-Onion-source]" ci-dessous illustre la structure de chiffrement d'une cellule au sein du réseau TOR. La première couche de chiffrement sera déchiffrée par le premier noeud du circuit. L'oignon sera ainsi épluché jusqu'à atteindre le dernier noeud et retirer sa dernière couche. Ce processus permettra ainsi au message d'être transmis au destinataire de manière sécurisée.

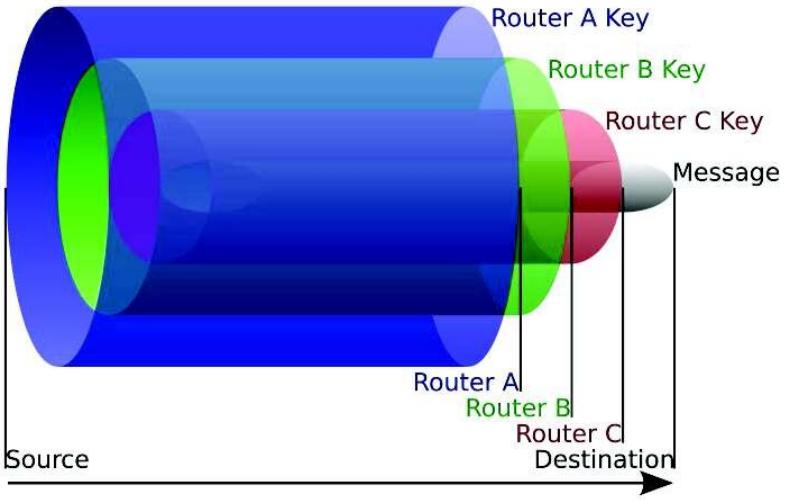

Figure 4: Structure du chiffrement en couches. [3, Layers-of-the-Onion-source].

4.1.2 Cryptographie asymétrique

La cryptographie asymétrique repose sur l'utilisation d'une paire de clés publique et privée associées. Ces clés sont mathématiquement liées: dans RSA il s'agit de la factorisation de grands nombres premiers tandis que dans ECC il s'agit de logarithmes discrets.

Un message chiffré avec une clé publique ne peut être déchiffré qu'avec la clé privée correspondante: si une personne reçoit un message lui demandant de crier dans un auditoire afin de pouvoir la situer, lorsque celle-ci criera, seule la personne qui le lui aura demandé sera en mesure de déchiffrer le message "Je suis ici", les autres ne disposant tout simplement pas des clés permettant de décoder la situation. Dans cet exemple, la clé publique est le système de messagerie permettant de cacher le contenu du message aux autres personnes présentes. La clé privée est le contenu du message en lui-même.

4.1.3 Comparaison cryptographie symétrique et asymétrique

Le Tableau 2 "Comparaison de la cryptographie symétrique et asymétrique" ci-dessous permet de mettre en évidence les différences d'usages entre cryptographie symétrique et asymétrique. D'un côté une cryptographie

plus simple à mettre en place et plus adaptée pour le transfert d'importants volumes de données. De l'autre une cryptographie plus exigeante mais offrant d'avantages de garanties lors du partage de données particulièrement sensibles.

Caractéristic	**Cryptographie Symétrique**	**Cryptographie Asymétrique**
Clés	Unique, partagée	Paire de clés (publique, privée)
Algorithmes	AES, DES	RSA, ECC
Complexité	Moindre	Élevée
Gestion des clés	Difficile pour de nombreux utilisateurs	Plus simple grâce à la clé publique
Utilisation	Chiffrement de masse, sécurisé si la clé est sûre	Transactions sécurisées, signatures numériques
Vulnérabilités	Analyse des clés, gestion compromise	Attaques cryptanalytiques, paramètres faibles

Table 2 Comparaison de la cryptographie symétrique et asymétrique

4.2 Communications

Il s'agit, ici, d'une combinaison de poignées de mains Diffie-Helman et de connections TLS pour sécuriser les communications.

1. 4.2.1 Diffie-Helman Permet de générer des clés privées sans les transmettre directemetn sur le circuit.
2. 4.2.2 TLS Permet de sécuriser les communications entre relais.

4.2.1 Diffie-Helman

Les poignées de mains Diffie-Helman permettent aux parties prenantes d'une communication sécurisée par cryptographie asymétrique de générer les clés privées nécessaires à la cryptographie symétrique.

Sur base de paramètres communs, chaque partie génère une clé privée qui lui permettra ensuite de calculer sa clé publique.

Une fois les clés publiques partagées, chaque partie utilise sa propre clé privée ainsi que la clé publique de l'autre partie afin de générer la clé symétrique partagée.

Algorithm 1 Échange de clés Diffie-Hellman entre Alice et Bob

1: Alice et Bob s'accordent sur un nombre premier p et une base g.
2: Alice choisit un nombre secret a et envoie $g^a \mod p$ à Bob.
3: Bob choisit un nombre secret b et envoie $g^b \mod p$ à Alice.
4: Alice calcule $(g^b \mod p)^a \mod p$.
5: Bob calcule $(g^a \mod p)^b \mod p$.
6: Alice et Bob utilisent ce nombre comme leur clé partagée.

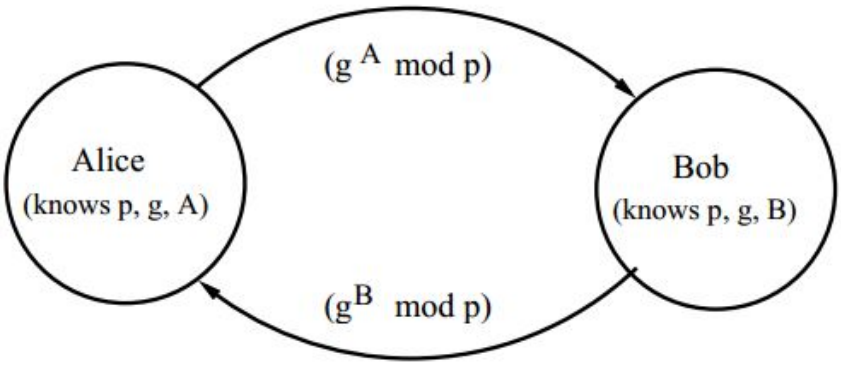

Figure 5: Illustration de l'algorithme Diffie-Hellman. [4, Diffie-Hellman Algorithm.].

La motivation principale de l'utilisation de DH réside dans sa capacité à fournir un secret partagé qui peut ensuite être utilisé pour un chiffrement symétrique robuste, sans que la clé elle-même ne doive être transmise sur le réseau, offrant ainsi une sécurité contre les interceptions. DH offre également le secret parfait vers l'avant: la compromission des clés privées n'entraîne pas celle des clés de sessions déjà établies.

4.2.2 TLS

Transport Layer Security (TLS), permet d'établir et de sécuriser les communications entre les relais en se basant sur une poignée de mains Diffie-Helman et une méthode de chiffrement (symétrique).

Le protocole TLS utilise donc une combinaison de cryptographie asymétrique pour l'échange de clés et l'authentification ainsi que de cryptographie symétrique pour le chiffrement des données.

5 Fonctionnement

TOR est un réseau de serveurs, appelés nœuds ou relais, permettant aux utilisateurs d'accéder à Internet de manière anonyme. Son fonctionnement est basé sur le principe de la transmission de données en couches chiffrées, d'où son nom qui fait référence à l'oignon ainsi qu'à ses couches. Cette section détaillera les étapes clés du processus permettant l'utilisation de TOR de l'"Initialisation" à l' "Utilisation" en passant par la "Construction".

5.1 Initialisation

L'utilisateur initie la connexion avec le premier noeud.

1. **Démarrage du Proxy Oignon (OP):** Il exécute d'un proxy oignon (OP *Onion Proxy*) qui sert d'interface entre l'application utilisateur et le réseau TOR.
2. **Récupération des répertoires:** L'OP récupère la liste des OR actifs depuis les serveurs d'annuaires ainsi que les adresses, clés publiques et politiques de sorties. Ces informations signées permettent de vérifier leur authenticité.
3. **Sélection des noeuds:** Les nœuds du circuit sont sélectionnés sur base de critères géographiques et liés aux politiques de sortie. De cette manière, l'anonymat ainsi que la compatibilité avec les besoins utilisateurs sont garantis.
4. **Initialisation TLS:** L'OP établit une connexion TLS avec le gardien d'entrée (*entry node*) afin de garantir la confidentialité et l'intégrité des communications initiales. Ce nœud connaît l'origine de la connexion mais ne connaît pas la destination finale des données.

5.2 Construction

TOR crée un circuit chiffré à travers le réseau.

1. **Négociation Clé Diffie-Hellman:** Pour chaque OR sur le circuit, l'OP envoie une cellule "CREATE" contenant la première moitié d'un échange de clés DH, chiffrée avec la clé publique de l' OR (clé d'oignon). Ce processus établit une clé de session symétrique partagée qui garantit que seul l' OR peut déchiffrer et ainsi répondre à la requête.
2. **Extension du circuit:** L'OP envoie des cellules "RELAY EXTEND" à travers le circuit existant vers les autres OR pour négocier les nouvelles clés de session symétriques via DH. Chaque OR ajoute sa propre couche de chiffrement afin de s'assurer que seuls les noeuds suivants puisse lire les instructions pour l'extension du circuit.
3. **Vérification du circuit:** Le chiffrement itératif à chaque étape du circuit assure que chaque OR ne puisse accéder qu'aux informations de son prédécesseur et successeur direct.

5.3 Utilisation

Les données passent ensuite par plusieurs nœuds intermédiaires (*middle nodes*), où chaque nœud ne connaît que le nœud précédent et le suivant, mais jamais l'ensemble du circuit, comme expliqué dans.

1. **Chiffrement en Couches :** Les données sont encapsulées dans des couches de chiffrement, une pour chaque nœud que le paquet de données traversera.
2. **Déchiffrement Progressif :** À chaque étape du circuit, un nœud enlève une couche de chiffrement pour découvrir à quel nœud envoyer le paquet suivant.
3. **Destination Finale :** Lorsque les données atteignent le nœud de sortie (*exit node*), la dernière couche de chiffrement est retirée et les données sont envoyées à leur destination finale.

Grâce à ce mécanisme, l'adresse IP de l'utilisateur est masquée, et les données transmises sont rendues indéchiffrables pour les observateurs extérieurs. Cela assure l'anonymat de l'utilisateur et la confidentialité des informations échangées.

6 V3 Onion Services

Depuis 2021, TOR a abandonné la V2 des services oignons pour migrer vers la V3 [5, A Comprehensive and Long-term Evaluation of TOR V3 Onion Services] [6, On the state of V3 onion services].

La principale différence entre les deux générations réside dans la complexité des clés: là où la v2 utilisait des adresses de 16 caractères dérivées d'une clé RSA de 1024 bits, la v3 utilise quant à elle des adresses de 56 caractères dérivées d'une clé ED25519. Ce qui contribue à renforcer leur sécurité.

De plus, les communications se voient également renforcées par l'utilisation de clés éphémères pour chaque session.

7 Conclusion

Tout au long du présent rapport, nous avons pu nous rendre compte des avancées considérables réalisées dans The Onion Router (TOR) en comparaison à l'onion routing de première génération.

De par son secret parfait vers l'avant, ses services cachés et autres options configurables, TOR se positionne aujourd'hui comme une référence avec une place de choix parmi les systèmes de communications permettant l'anonymat.

Cependant, cette seconde génération n'était pas exempte de défauts et a donné suite à une 3e version offrant un niveau de sécurité encore accru.

Pour palier aux défauts de TOR, d'autres solutions émergent dont TomiNet [7], produit de Tomi, une compagnie Web3. TomiNet entend résoudre 2 défauts de fabrication de TOR:

> « When the internet was created, I am sure they wanted to allow freedom of information and speech, but because the architecture of the technology includes [internet protocols] IPs and centralized entities like [Internet Corporation for Assigned Names and Numbers] ICANN control domain names, the result is that governments can, through IPs, know who says what, go after them and block websites. », — Techno Prince, a pseudonymous member of Tomi's founding team [7]

Si TomiNet repose sur une architecture similaire à celle de TOR, le réseau y ajoute une couche de gouvernance décentralisée via Decentralized Autonomous Organisation (DAO) permettant ainsi de réduire les usages liés aux activités illicites.

A Latence et anonymat

Dans les sytèmes de communication en réseau, le niveau d'anonymat est toujours corrélé avec le niveau de latence de ce système.

L'anonymat d'un client sur un réseau informatique lui confère l'intraçabilité de son activité en ligne. Dans le cadre de TOR, cela est réalisé via le routage du trafic à travers plusieurs nœuds ou Onion Router (OR): chaque nœud n'ayant connaissance que de son prédécesseur ainsi que de son successeur.

La latence - somme des délais - dans un réseau informatique désigne le temps écoulé entre l'envoi d'un paquet de données depuis une source et sa réception par la destination. Dans le cadre de TOR, son accroissement s'explique d'une part par la répartition géographique globale des OR (*cfr,* Propagation Delay (PD)) et d'autre part par les délais de chiffrement, déchiffrement qui surviennent à chaque OR (*cfr,* Transmission Delay (TD)).

Cette latence est donc un compromis nécessaire pour obtenir un niveau élevé d'anonymat et de sécurité, car elle rend plus difficile pour un observateur de corréler le trafic entrant et sortant.

B Menaces

Comme décrit tout au long de ce rapport, TOR assure l'anonymat de ses client, la non-vulnérabilité des données qu'ils transmettent ainsi que l'intraçabilité de leurs communications de bout en bout. La présente section s'attardera sur les différents types d'attaques recensées dans l'article ainsi que les axes de défenses mis en place dans TOR. L'anonymat sera abordé au point B.1 "Passive", les aspects cryptographiques au point B.2 "Active". Les servrus de répertoires au point B.3 "Annuaire" et les point de rendez-vous au B.4 "Points de rendez-vous".

B.1 Passive

Les attaques passives concernent uniquement l'établissement de profils sur base d'analyses permettant ainsi de compromettre l'anonymat. Le tableau 3 "Menaces et défenses dans TOR : attaques passives" ci-dessous présente les différents types d'attaques passives recensées dans l'article et y associe un axe de défense.

Menace	**Solution**
Observation des profils de trafic	Le multiplexage de traffics sur un même circuit empêche l'analyse précise de profils de trafics.
Observation du contenu utilisateur	Si la destination est un noeud hostile, l'usage de Privoxy[1] permet de conserver l'anonymat du client.
Distinguabilité des options	En permettant aux utilisateurs de confugurer leurs profils, cela compromet leur anonymat.
Corrélation temporelle de bout en bout	Masquer la communication entre le OP et le premier OR via un OP sur le OR. Cela contraint à séparer le traffic en provenance du trafic traversant.
Corrélation de taille de bout en bout	Rembourage[2] ou topologie du tuyau fuyant[3].
Empreintes digitales de site Web	Multiplexage des flux, modification de la taille des cellules et rembourages.

Table 3 Menaces et défenses dans TOR : attaques passives

B.2 Active

Les attaqques actives concernent uniquement la compromission des techniques de chiffrement via divers procédés. Le tableau 4 "Menaces et défenses dans TOR : attaques actives" ci-dessous présente les différents types d'attaques actives recencées dans l'article et y associe un axe de défense.

Menace	Solution
Compromission des clés	La rotation periodique permet de contrer la compromission des clés de sessions TLS car même si l'attaquant peut observer les cellules relayées sur chacun des circuits de cette connexion, sans la clé oignon, il ne peut déchiffrer le contenu.
Compromission itérative	Le secret parfait vers l'avant (*i.e.* Perfect Forward Secrecy) empêche de remonter un circuit même si un attaque a lieu d'un OR intermédiaire jusqu'au dernier.
Exploitation d'un serveur web	TOR dépend de Privoxy pour empêcher un serveur web d'identifier des profils temporels des utilisateurs qui s'y connectent d'adapter ses réponses.
Exploitation d'un proxy onion	À un client ne peut correspondre qu'un OP local. Cependant, compromettre un proxy onion compromet toutes ses connexions.
Attaque DoS sur des nœuds non observables	La mise en place de stratégies permettant au réseau de garder le contrôle sur les ressources matérielles utilisables permet de contrer les attaques de type Déni de service (*i.e.* Denial of Service (DoS)).
Exploitation d'un OR hostile	Un nœud hostile doit etre immediatement adjacent aux deux extremites pour compromettre l'anonymat d'un circuit. Si un adversaire contrôle plus d'un OR, il ne peut compromettre plus qu'une partie du trafic.
Introduction de timing dans les messages	Cela représente une version plus forte des attaques de timing passives déjà discutées.
Attaques par marquage	Les contrôles d'intégrité[4] sur les cellules empêchent un nœud hostile de marquer une cellule en la modifiant.
Remplacement de contenus de protocoles non authentifiés	La préférence des clients pour des protocoles authentifiés de bout en bout permet d'empêcher un nœud de sortie hostile de se faire passer pour le serveur cible.
Attaques par rejeu	Le protocole d'échnage de clés DH permet de renégocier de nouvelles clés de session et ainsi contrer les attaques par replay.
Attaques de diffamation	Les politiques de sorties réduisent l'impact des attaques par diffamation.
Distribution de code hostile	Des clés publiques de version officielles du code de TOR permettent de vérifier son authenticité avant de l'exécuter.

Table 4 Menaces et défenses dans TOR : attaques actives

B.3 Annuaire

Les serveurs d'annuaires sont un coposante essentielle de tout réseau TOR.

Les attaques contre les serveurs d'annuaires ciblent celles dont l'objectif est la compromission d'une partie des circuits créés. Le tableau 5 "Menaces et défenses dans TOR : points de rendez-vous" ci-dessous présente les différents types d'attaques contre les serveurs d'annuaires et y associe un axe de défense.

Menace	Solution
Destruction des serveurs d'annuaires	Tant que la moitié des serveurs d'annuaires sont en activité, ils continuent de fournir un annuaire valide.
Subversion d'un serveur d'annuaire	N'a qu'une influence partielle et minime sur la composition de l'annuaire final.
Subversion de la majorité des serveurs d'annuaire	Les opérateurs de serveurs d'annuaires doivent être indépendants et résistants.
Encouragement à la dissension entre serveurs d'annuaire	Aucune solution n'est proposée pour empecher de diviser les opérateurs et ainsi leurs utilisateurs.
Tromper les serveurs d'annuaire pour lister un OR hostile	Les opérateurs de serveurs de annuaire sont capables de filtrer la plupart des ORs hostiles.
Convaincre les annuaires qu'un OR défaillant fonctionne	Les serveurs d'annuaires doivent impérativement tester les OR de manière appropriée pour empêcher qu'un OR ne puisse accepter une connexion TLS en ignorant les cellules et ainsi passer les barrières de sécurité de TOR.

Table 5 Menaces et défenses dans TOR : serveurs d'annuaires

B.4 Points de rendez-vous

Les attaques contre les points de rendez-vous. Le tableau 6 "Menaces et défenses dans TOR : points de rendez-vous" ci-dessous recence les différent types d'attaques contre les points de rendez-vous et y associe un axe de défense.

Menace	Solution
Multiples demandes d'introduction	Les points de rendez-vous peuvent bloquer les requêtes qui ne contiennent pas de jetons d'autorisation, de restreindre le nombre de requêtes recevables ou d'exiger une certaine quantité de calcul pour chaque requête reçue.
Attaquer un point d'introduction	Désactiver les points de rendez-vous mais ils sont liés à des clés publiques.
Compromission d'un point d'introduction	Un point de rendez-vous compromis peut inonder le trafic ou empêcher de nouvelles demandes.
Compromission d'un point de rendez-vous	Trafic chiffré par une clé de session.

Table 6 Menaces et défenses dans TOR : points de rendez-vous

Figures

1 Structure de cellule vs structure d'une cellule de relais. [1, Tor: The Second-Generation Onion Router] 6

2 Circuit à deux Routeurs Onions permettant à Alice de joindre un site web.[1, Tor: The Second-Generation Onion Router] 7

3 Modélisation des différentes couches de sécurité dans TOR 8

4 Structure du chiffrement en couches. [3, Layers-of-the-Onion-source]. 9

5 Illustration de l'algorithme Diffie-Hellman. [4, Diffie-Hellman Algorithm.]. 11

Tableaux

1 Comparaison des Types de Cellules et de leurs commandes dans le Réseau Tor 6
2 Comparaison de la cryptographie symétrique et asymétrique 10
3 Menaces et défenses dans TOR : attaques passives 16
4 Menaces et défenses dans TOR : attaques actives 17
5 Menaces et défenses dans TOR : serveurs d'annuaires 18
6 Menaces et défenses dans TOR : points de rendez-vous 18

Algorithmes

1 Échange de clés Diffie-Hellman entre Alice et Bob 11

Glossaire

AC Asymetric Cryptography. 4, 8

DAO Decentralized Autonomous Organisation. 14

DH Diffie-Hellman. 8, 9, 11, 12

ECC Elliptic Curve Cryptography. 9, 10

OP Onion Proxy. 7, 12, 16

OR Onion Router. 7, 12, 15, 17, 18

PD Propagation Delay. 15

Privoxy Privoxy est un proxy non-caching conçu pour améliorer la confidentialité des utilisateurs en filtrant le contenu web et en gérant les cookies. Il offre plusieurs fonctionnalités, telles que le blocage de publicités, l'anonymisation des requêtes et la suppression des éléments de suivi. Privoxy est souvent utilisé en complément de réseaux d'anonymat comme Tor pour offrir une couche supplémentaire de protection en ligne en nettoyant les protocoles d'application sans compromettre les fonctionnalités de Tor. . 16

RSA Rivest–Shamir–Adleman. 9, 13

SC Symetric Cryptography. 8

TCP Transmission Control Protocol. 4, 6

TD Transmission Delay. 15

TLS Transport Layer Security. 8, 11, 12, 17, 18

TOR The Onion Router. 1, 3–6, 8, 9, 12–18, 20

UDP User Datagram Protocol. 4

Références

[1] Roger Dingledine, Nick Mathewson, and Paul Syverson. "Tor: The Second-Generation Onion Router:" in: Fort Belvoir, VA: Defense Technical Information Center, Jan. 2004. DOI: 10.21236/ADA465464. URL: http://www.dtic.mil/docs/citations/ADA465464 (visited on 07/07/2023).

[2] David Chaum et al. "cMix: Anonymization by High-Performance Scalable Mixing". en. In: ().

[3] *Layers-of-the-Onion-source.png 802 × 528 pixels*. URL: https://www.researchgate.net/profile/Nguyen-Phong-Hoang/publication/275098019/figure/fig31/AS:391887249788946@1470444609470/Layers-of-the-Onion-source.png (visited on 04/08/2024).

[4] Tech Spider. *Diffie-Hellman Key Exchange*. Apr. 2024. URL: https://googler700.blogspot.com/2016/01/diffie-hellman-key-exchange.html (visited on 05/12/2024).

[5] Chunmian Wang et al. "A Comprehensive and Long-term Evaluation of Tor V3 Onion Services". en. In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. New York City, NY, USA: IEEE, May 2023, pp. 1–10. ISBN: 9798350334142. DOI: 10.1109/INFOCOM53939.2023.10229057. URL: https://ieeexplore.ieee.org/document/10229057/ (visited on 11/13/2023).

[6] Tobias Hoeller, Michael Roland, and René Mayrhofer. "On the state of V3 onion services". In: *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*. FOCI '21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 50–56. ISBN: 978-1-4503-8640-1. DOI: 10.1145/3473604.3474565. URL: https://dl.acm.org/doi/10.1145/3473604.3474565 (visited on 04/08/2024).

[7] Oluwaseun Adeyanju. *How Blockchain Can Fix Tor Project's Biggest Flaws And Create A Truly Free Internet*. en. Section: Forbes Digital Assets. URL: https://www.forbes.com/sites/oluwaseunadeyanju/2022/12/23/how-blockchain-can-fix-tor-projects-biggest-flaws-and-create-a-truly-free-internet/ (visited on 05/15/2024).