

Approche technique de l'espace numérique

Bonus
Le réseau Tor



Pablo Rauzy <pr@up8.edu>
`pablo.rauzy.name/teaching/aten`

Le réseau Tor

- ▶ Réseau décentralisé superposé à Internet.
- ▶ Sert à anonymiser la source d'une connexion TCP.
- ▶ Logiciel libre sous licence BSD, développé en C.
- ▶ Développement assuré par la fondation "The Tor Project".

- ▶ Tout type de communication TCP :
 - Navigation web.
 - Email.
 - Messagerie instantanée.
 - etc.

- ▶ Anonymisation :
 - Cache l'adresse IP réelle. *C'est tout !*
 - Attention aux empreintes des navigateurs, et aux autres traces laissées en ligne.
 - Le projet Tor développe aussi le *Tor Browser*, qui minimise les traces identifiantes.

- ▶ Contournement de la censure :
 - Le réseau Tor permet d'accéder à des parties d'Internet qui peuvent être bloqués à l'endroit où l'on se trouve.
 - Il permet aussi de masquer localement son trafic (par le chiffrement, mais on peut voir que vous utilisez Tor).

- ▶ Petit rappel (ou petite introduction superficielle) à la *cryptographie asymétrique*.
- ▶ Le but est de pouvoir communiquer de manière sécurisée sur un canal non sécurisé.
- ▶ Le principe est d'avoir deux clefs par participant·es, de telle sorte que
 - un message chiffré avec la première clef ne peut être déchiffré qu'avec la seconde,
 - un message chiffré avec la seconde clef ne peut être déchiffré qu'avec la première.
- ▶ Par convention, l'une de ces clefs est appelée la *clef privée* et l'autre la *clef publique*.

Utilisation des clefs

- ▶ Assurer la confidentialité du message :
 - L'émetteur chiffre un message avec la clef publique du destinataire.
 - Seul le destinataire peut le déchiffrer avec sa clef privée.

- ▶ Assurer l'authenticité de l'émetteur :
 - L'émetteur chiffre un message avec sa clef privée.
 - Le destinataire peut déchiffrer ce message avec la clef publique de l'émetteur.

- ▶ Le but est de permettre à une source S de se connecter à une destination D tout en gardant son anonymat.
- ▶ Personne à part S ne doit connaître à la fois S et D .

Création d'un circuit

- ▶ D'abord, S récupère depuis un annuaire public une liste de *nœuds Tor* (communication chiffrée).
- ▶ Ensuite, trois nœuds sont choisis au hasard :
 - un *nœud d'entrée* NE ,
 - un *nœud relaie* NR ,
 - un *nœud de sortie* NS ,
- ▶ Ils formeront un *circuit* jusqu'à D .

Utilisation d'un circuit

► Pour envoyer un message m à D :

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = Enc(m, pk_{NS})$.

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = \text{Enc}(m, pk_{NS})$.
 - S chiffre m_1 avec la clef publique de NR , $m_2 = \text{Enc}(m_1, pk_{NR})$.

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = Enc(m, pk_{NS})$.
 - S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1, pk_{NR})$.
 - S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2, pk_{NE})$.

Utilisation d'un circuit

► Pour envoyer un message m à D :

- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
- S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
- S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
 - S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
 - S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.
 - S envoie m_3 à NE .

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
 - S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
 - S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.
 - S envoie m_3 à NE .
 - NE déchiffre $m_2 + NR = Dec(m_3, sk_{NE})$, et envoie m_2 à NR .

Utilisation d'un circuit

► Pour envoyer un message m à D :

- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
- S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
- S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.
- S envoie m_3 à NE .
- NE déchiffre $m_2 + NR = Dec(m_3, sk_{NE})$, et envoie m_2 à NR .
- NR déchiffre $m_1 + NS = Dec(m_2, sk_{NR})$, et envoie m_1 à NS .

Utilisation d'un circuit

- Pour envoyer un message m à D :
- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
 - S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
 - S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.
 - S envoie m_3 à NE .
 - NE déchiffre $m_2 + NR = Dec(m_3, sk_{NE})$, et envoie m_2 à NR .
 - NR déchiffre $m_1 + NS = Dec(m_2, sk_{NR})$, et envoie m_1 à NS .
 - NS déchiffre $m + D = Dec(m_1, sk_{NS})$, et envoie m à D .

Utilisation d'un circuit

► Pour envoyer un message m à D :

- S chiffre m avec la clef publique de NS , $m_1 = Enc(m + D, pk_{NS})$.
- S chiffre m_1 avec la clef publique de NR , $m_2 = Enc(m_1 + NS, pk_{NR})$.
- S chiffre m_2 avec la clef publique de NE , $m_3 = Enc(m_2 + NR, pk_{NE})$.
- S envoie m_3 à NE .
- NE déchiffre $m_2 + NR = Dec(m_3, sk_{NE})$, et envoie m_2 à NR .
- NR déchiffre $m_1 + NS = Dec(m_2, sk_{NR})$, et envoie m_1 à NS .
- NS déchiffre $m + D = Dec(m_1, sk_{NS})$, et envoie m à D .

► De cette façon :

- D ne connaît que NS ,
- NS ne connaît que D et NR ,
- NR ne connaît que NS et NE ,
- NE ne connaît que NR et S .

Question

- ▶ Comme on l'a dit, Tor permet de transporter du trafic TCP.
- ▶ Par rapport aux connaissances que vous avez déjà du fonctionnement d'Internet, est-ce que vous voyez un soucis ?

DNS

- ▶ Pour utiliser Tor on passe par un proxy SOCKS (qu'on a déjà vu avec SSH).
- ▶ Ce type de proxy permet de faire passer les requêtes DNS comme un cas particulier, même si il n'est pas capable de faire transiter du trafic UDP.

- ▶ En plus de permettre d'anonymiser la source d'une connexion TCP, Tor permet de créer des *services cachés*.
- ▶ Un service caché vit entièrement dans le réseau Tor.
- ▶ Il est identifié par un hash de 16 chiffres en base 32 (le fameux **.onion**). Depuis les "onion v3", il s'agit de 56 chiffres.

Initialisation

- ▶ À son lancement, un service caché crée des circuits jusqu'à des nœuds aléatoires. Ces nœuds sont ses *points d'introduction*.
- ▶ Ensuite il crée son *descripteur de service* qui contient sa clef publique + la liste de ses points d'introduction.
- ▶ Ce descripteur est stocké dans une table de hachage distribuée (DHT).
- ▶ Le nom **.onion** est une empreinte de la clef publique et permet de retrouver le descripteur du service dans la DHT.

- ▶ Quand un client C veut se connecter à un service caché S :

- ▶ Quand un client C veut se connecter à un service caché S :
 - C crée un circuit vers un nœud aléatoire, le *point de rendez-vous*.

- Quand un client C veut se connecter à un service caché S :
- C crée un circuit vers un nœud aléatoire, le *point de rendez-vous*.
 - C demande à la DHT le descripteur du service caché.

- Quand un client C veut se connecter à un service caché S :
- C crée un circuit vers un nœud aléatoire, le *point de rendez-vous*.
 - C demande à la DHT le descripteur du service caché.
 - C demande à l'un des points de d'introduction de S de lui envoyer un message chiffré avec la clef publique de S qui contient le point de rendez-vous et un secret.

- Quand un client C veut se connecter à un service caché S :
- C crée un circuit vers un nœud aléatoire, le *point de rendez-vous*.
 - C demande à la DHT le descripteur du service caché.
 - C demande à l'un des points de d'introduction de S de lui envoyer un message chiffré avec la clef publique de S qui contient le point de rendez-vous et un secret.
 - Le point d'introduction délivre le message à S qui se connecte alors au point de rendez-vous, et lui envoie le secret.

- Quand un client C veut se connecter à un service caché S :
- C crée un circuit vers un nœud aléatoire, le *point de rendez-vous*.
 - C demande à la DHT le descripteur du service caché.
 - C demande à l'un des points de d'introduction de S de lui envoyer un message chiffré avec la clef publique de S qui contient le point de rendez-vous et un secret.
 - Le point d'introduction délivre le message à S qui se connecte alors au point de rendez-vous, et lui envoie le secret.
 - Le point de rendez-vous notifie C que S est bien là, et ne sert alors plus que de relaie dans le circuit qui va de C à S .

- ▶ Un mot sur la sécurité.
- ▶ Un mot sur les polémiques.
- ▶ Un mot sur les blocages.
- ▶ Liens :
 - <https://www.torproject.org/>
 - <https://tails.boum.org/>
 - <https://nos-oignons.net/>
 - <https://guide.boum.org/>