Systèmes distribués en milieu antagoniste

Vincent Botbol, Marc Beunardeau,
Mathias Bourgoin, Julien Tesson

22 octobre 2019

Plan du cours

7 semaines + projet

1. Aperçu général
2. Cryptographie
3. Algorithmes de consensus
4. Privacy
5. Systèmes distribués en milieu antagoniste
6. Protocoles économiques
7. Soutenance de projet

Rappels : Blockchain

Blockchain

Algorithme générique :

1. De nouvelles transactions sont diffusées à *tous* les **nœuds**"participants" qui les agrègent dans des **blocs**.
2. Le prochain bloc est diffusé par un (ou plusieurs) nœud(s)
3. Les nœuds expriment leur acceptation d'un bloc en incluant son hash dans le prochain bloc qu'ils créent.

Deux composants principaux

Le protocole économique

- Définit les règles de validation (des transactions, des blocs)
- Permet l'exécution de smart contracts (VM, interprète, etc.)
- Organise le consensus

Le nœud / Le "shell"

- Organise les échanges de données
- Optimise les communications, le stockage, etc.
- Indépendant du protocole économique ?

Deux composants attaquables

Le protocole économique

- Selfish-mining
- 51% attack
- Smart contracts : l'attaque DAO

Le nœud / Le "shell"

- Attaque Éclipse
- Attaque Sybil
- Spam

Attaques consensus

Selfish Mining

Principe

- Un (groupe) de mineur(s) ne publie pas ses blocs trouvés
- Ceci produit un *fork* de la chaîne
- Les autres mineurs continuent sur la chaîne publique
- Quand le (groupe) de mineur(s) malhonnête(s) obtient une chaîne plus longue que la chaîne publique, il la publie.
- Ce qui entraine une réorganisation globale de la chaîne et annule les transactions uniquement présentes sur la chaîne publique

Majority is not Enough :Bitcoin Mining is Vulnerable - Ittay Eyal and Emin Gun Sirer - 2013

Selfish Mining - 2

Pour quoi faire ?

- Double dépense
- Censure
- Augmenter ses revenus
- Diminuer la difficulté de la chaîne

- Difficile a mettre en oeuvre ...
- ... mais difficile à détecter
- Si tout le monde fait du selfish mining, tout s'arrête

Attaque 51%

Principe

- Une entité possède 51% de la puissance de minage (PoW) du Stake (PoS)
- Elle peut alors décider de l'état global de la chaîne

Pour quoi faire ?

- Double dépense
- Censure

- Difficile à mettre en oeuvre
- Casse la confiance dans la chaîne (et donc la valeur des

Smart Contracts : l'attaque DAO - 1

The DAO

- Decentralized Autonomous Organization (DAO)
- Outis de financement participatif de projets sur la blockchain Ethereum
- Financement participatif via un vote, le tout contrôlé par des smart contracts
- TheDAO collecte plus 150M de dollars
- Pour protéger la minorité, il était possible de récupérer ses fonds lors qu'un vote passe dont on refuse le résultat
- Cette solution était implantée sous la forme d'un `split` en un second DAO

Smart Contracts : l'attaque DAO - 2

L'attaque

- Un programmeur a trouvé deux failles dans ce mécanisme :
  1. Le split récupérait les fonds dans le DAO avant de vérifier la balance du compte
  2. Il n'y avait pas de vérification d'appels récursifs de la fonction `split`
- Il utilisa un appel récursif de la fonction split pour récupérer plusieurs fois ses fonds et obtenir ainsi environ 3.6M Ether

Smart Contracts : l'attaque DAO - 3

Conséquences

- Après la découverte de l'attaque, un grand débat a eu lieu dans la communauté ethereum
- Trois propositions :
  - Accepter l'attaque et ne rien faire
  - Avec l'aide des mineurs, détruire le DAO créé et les fonds qu'il contenait
  - Annuler l'attaque en effectuant un hard fork de la chaîne

Smart Contracts : l'attaque DAO - 4

Conséquences

- Après la découverte de l'attaque, un grand débat a eu lieu dans la communauté ethereum
- Trois propositions :
  1. Accepter l'attaque et ne rien faire
  2. Avec l'aide des mineurs, détruire le DAO créé et les fonds qu'il contenait
  3. Annuler l'attaque en effectuant un hard fork de la chaîne

Toute la communauté n'a pas accepté le fork, donnant naissance à une chaîne alternative : *Ethereum Classic*.

Présentation du shell

Le shell

Plusieurs composants

- Une couche P2P
- Un système de stockage
- Un mempool
- Un serveur RPC
- et plus encore

Shell : La couche P2P

Propriétés

- Décentralisée
- Chargée de la communication des informations à tous les nœuds
- La plupart du temps implanté via un protocole épidémique (*gossip*)
- Maintient les connexions avec les pairs
- Permet l'arrivée de nouveau pairs
- Prend en compte l'existence de potentiels pairs malhonnêtes.

Shell : La couche P2P

Propriétés

- Décentralisée
- Chargée de la communication des informations à tous les nœuds
- La plupart du temps implanté via un protocole épidémique (*gossip*)

Gossip network

- Lorsqu'un nœud reçoit une **nouvelle** information, il sélectionne quelques noeuds auxquels il est connecté et leur transmet l'information.

Shell : La couche P2P

Propriétés

- Maintient les connexions avec les pairs
- Permet l'arrivée de nouveau pairs
- Prend en compte l'existence de potentiels pairs malhonnêtes.

Situation contradictoire

- Noeud connecté à son maximum de pairs autorisés
- 1 nouveau pair demande une connexion
- Comment maintenir les connexions tout en permettant au nouveau pair d'entrer dans le réseau

Shell : La couche P2P

Situation contradictoire

- Noeud connecté à son maximum de pairs autorisés
- 1 nouveau pair demande une connexion
- Comment maintenir les connexions tout en permettant au nouveau pair d'entrer dans le réseau

Solution : ajouter de l'aléa

- Autorise de nouvelles connexions aléatoirement (en en fermant d'anciennes)
- Echange des pairs avec ses voisins aléatoirement

Shell : La couche P2P

Exemples de problèmes

- Performance ou décentralisation
  - Favoriser des nœuds de zones géographiques différentes ?
  - Favoriser des nœuds proches de moi ?
- Nouveaux ou anciens
  - Favoriser les nouvelles connexions ?
  - Faire confiance aux nœuds anciens ?
- Performance locale ou santé globale du réseau
  - Favoriser un pair qui répond vite ?
  - Envoyer des infos a un pair qui répond lentement ?
  - Favoriser les pairs qui ont des informations nouvelles ?
  - Informer les pairs qui manquent d'informations ?

Shell : Système de stockage

Stockage

- Base de données répliquée
- Ne fait que grandir
- Souvent développé à l'aide d'outils généralistes (LevelDB, LMDB, Irmin)
- Parfois en utilisant des outils dédiés

Shell : Mempool

Objectifs

- Stocker les opérations en attente d'intégration dans un bloc
- Sélectionner quelles opérations diffuser ou non aux nœuds voisins

Comment ?

- Maintient un ensemble d'opérations à partager
- S'appuie directement sur les deux autres composants (P2P et Stockage)

Attaques shell

Pourquoi attaquer le shell ?

Le protocole économique

- Fixe les règles
- Valide les opérations
- Valide les blocs
- Permet le consensus

Pourquoi attaquer le shell ?

Le protocole économique

- Fixe les règles
- Valide les opérations
- Valide les blocs
- Permet le consensus

Mais

- Difficile (cher) à attaquer

Pourquoi attaquer le shell ?

Idée ?

Si le protocole économique est résistant, tant que la majorité des producteurs de blocs est honnête tout devrait bien se passer.

Pourquoi attaquer le shell ?

Idée ?

Si le protocole économique est résistant, tant que la majorité des producteurs de blocs est honnête tout devrait bien se passer.

Mais...

C'est la couche P2P et l'ensemble du Shell qui diffusent les informations, si on arrive à prendre le contrôle d'une partie du réseau P2P, on peut prendre le contrôle de la chaîne !

Attaque Éclipse

Principe

- Empêcher un (ou plusieurs) nœud(s) d'accéder à toute l'information

Pour quoi faire ?

- Double dépense
- Censure

Mais aussi

- Attaque 51% avec beaucoup moins de 51%
- Premier pas d'une attaque selfish mining

Attaque Éclipse

Principe

- Empêcher un (ou plusieurs) nœud(s) d'accéder à toute l'information

Défense

- Pas de solution miracle
- Conserver des connexions avec des nœuds connus
- Limiter les connexions multiples d'un même sous-réseau
- Associer plusieurs nœuds
- ...

Attaque Sybil

Principe

Créer de nombreuses identités sur un réseau P2P pour en prendre le contrôle.

Pour quoi faire ?

- Double dépense
- Censure

Défense

Les algorithmes de consensus des blockchains se protègent en adossant une ressource rare au droit de produire des blocs (PoW) ou en utilisant une sélection pseudo-aléatoire des prochains producteurs de blocs (PoS)

Spam

Principe
Saturer un/plusieurs nœuds ou l'ensemble du réseau de messages

Pour quoi faire ?

- DoS (Denial of Service)
- Ralentir le réseau
- Et plus...

Spam

Principe

Saturer un/plusieurs nœuds ou l'ensemble du réseau de messages

Défense

- Ne pas accepter/conserver/propager tous les messages reçus
  - Prévalidation des opérations
  - Augmenter les frais
  - Détecter le Spam ? (*dust operations*, . . .)

Conclusion

Blockchain publique

- Permissionless
- Antagoniste

Contradictions

- Accepter les nouveaux arrivants
- Se protéger des utilisateurs malhonnêtes

Problème complexe : tout doit être pensé pour limiter les risques d'attaques sans entraver le fonctionnement de la chaîne et la liberté des utilisateurs

Conclusion

Contradictions

- Accepter les nouveaux arrivants
- Se protéger des utilisateurs malhonnêtes

Problème complexe : tout doit être pensé pour limiter les risques d'attaques sans entraver le fonctionnement de la chaîne et la liberté des utilisateurs

Beaucoup de solutions reposent sur

- Incitation économique (surtout pour les attaques du consensus)
- Ajouter de l'aléa