Merkle Trees et intégrité

Pierre Bourdon

LSE 2013

26 octobre 2011

But

- Vérification de l'intégrité inline
- Cryptographiquement sûr
- Léger

But

- Vérification de l'intégrité inline
- Cryptographiquement sûr
- Léger

But

- Vérification de l'intégrité inline
- Cryptographiquement sûr
- Léger

Structure

- Arbre binaire
- Une feuille pour chaque partie des données à vérifier
- Chaque noeud a comme valeur un hash

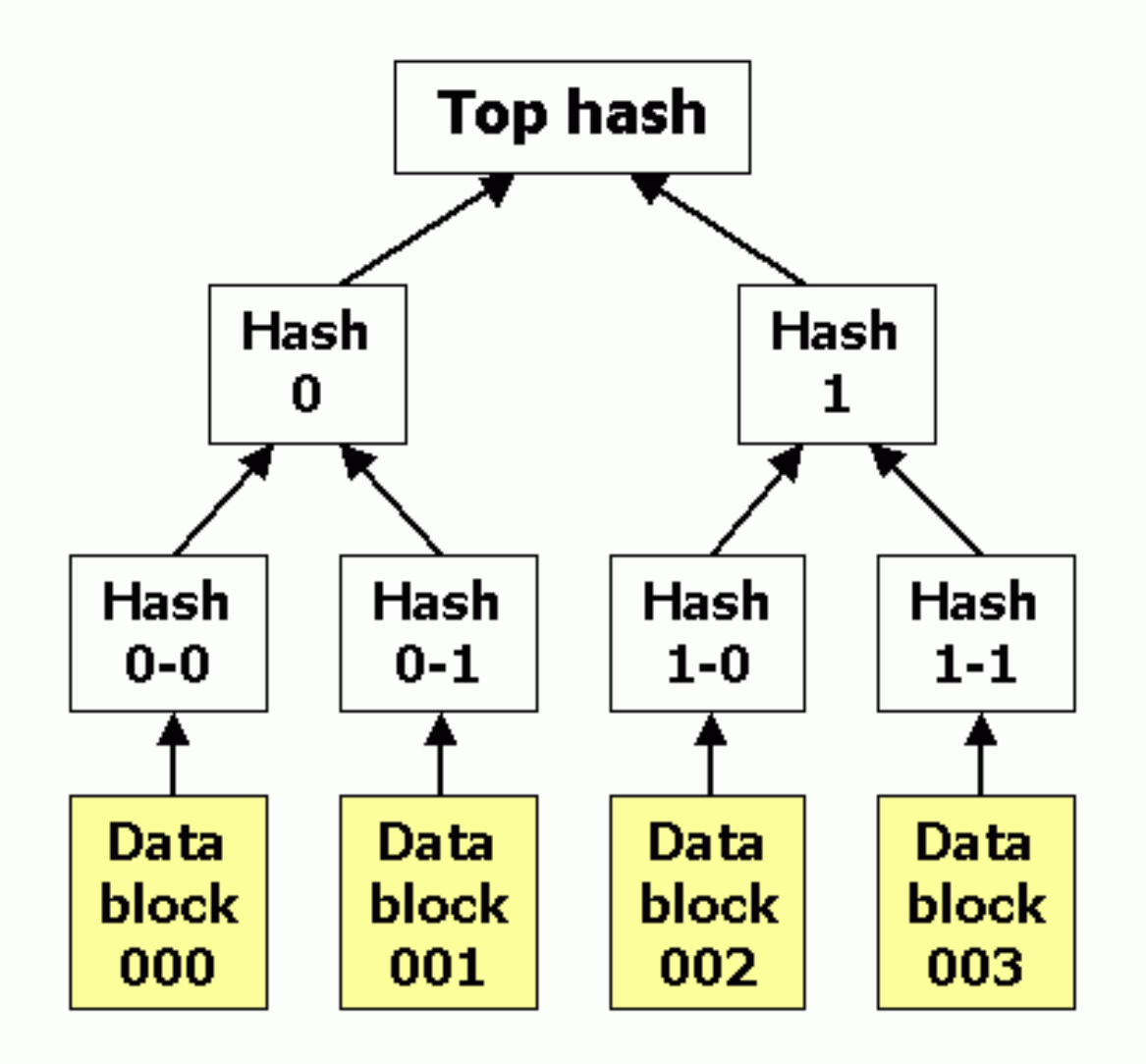
Structure

- Arbre binaire
- Une feuille pour chaque partie des données à vérifier
- Chaque noeud a comme valeur un hash

Structure

- Arbre binaire
- Une feuille pour chaque partie des données à vérifier
- Chaque noeud a comme valeur un hash

Schéma


Top hash
Hash 0
Hash 1
Hash 0-0
Hash 0-1
Hash 1-0
Hash 1-1
Data block 000
Data block 001
Data block 002
Data block 003

Magie

- Soit N le nombre de parties
- $log_2(N)$ hashes pour vérifier une partie
- Surcoût total de $2N$ hashes
- Seul le hash racine a besoin d'être sûr

Magie

- Soit N le nombre de parties
- $log_2(N)$ hashes pour vérifier une partie
- Surcoût total de $2N$ hashes
- Seul le hash racine a besoin d'être sûr

Magie

- Soit N le nombre de parties
- $log_2(N)$ hashes pour vérifier une partie
- Surcoût total de $2N$ hashes
- Seul le hash racine a besoin d'être sûr

Magie

- Soit N le nombre de parties
- $log_2(N)$ hashes pour vérifier une partie
- Surcoût total de $2N$ hashes
- Seul le hash racine a besoin d'être sûr

Exemple 1

- Bittorrent
- Hash racine dans le .torrent
- Chaque partie est téléchargée avec $log_2(N)$ hashes
- Pas de risque de mauvaises données reçues

Exemple 1

- Bittorrent
- Hash racine dans le .torrent
- Chaque partie est téléchargée avec $log_2(N)$ hashes
- Pas de risque de mauvaises données reçues

Exemple 1

- Bittorrent
- Hash racine dans le .torrent
- Chaque partie est téléchargée avec $log_2(N)$ hashes
- Pas de risque de mauvaises données reçues

Exemple 1

- Bittorrent
- Hash racine dans le .torrent
- Chaque partie est téléchargée avec $log_2(N)$ hashes
- Pas de risque de mauvaises données reçues

Exemple 2

- ZFS
- Chaque objet du filesystem est un noeud de l'hash tree
- Empêche la corruption
- Empêche l'altération malicieuse lors d'un export

Exemple 2

- ZFS
- Chaque objet du filesystem est un noeud de l'hash tree
- Empêche la corruption
- Empêche l'altération malicieuse lors d'un export

Exemple 2

- ZFS
- Chaque objet du filesystem est un noeud de l'hash tree
- Empêche la corruption
- Empêche l'altération malicieuse lors d'un export

Exemple 2

- ZFS
- Chaque objet du filesystem est un noeud de l'hash tree
- Empêche la corruption
- Empêche l'altération malicieuse lors d'un export

Exemple 3

- DVD de jeu de Wii
- Chaque cluster du disque contient $log_2(N)$ hashes
- Le hash racine est signé cryptographiquement
- Aucune modification non approuvée possible

Exemple 3

- DVD de jeu de Wii
- Chaque cluster du disque contient $log_2(N)$ hashes
- Le hash racine est signé cryptographiquement
- Aucune modification non approuvée possible

Exemple 3

- DVD de jeu de Wii
- Chaque cluster du disque contient $log_2(N)$ hashes
- Le hash racine est signé cryptographiquement
- Aucune modification non approuvée possible

Exemple 3

- DVD de jeu de Wii
- Chaque cluster du disque contient $log_2(N)$ hashes
- Le hash racine est signé cryptographiquement
- Aucune modification non approuvée possible

Conclusion

- Questions ?
- @delroth_
- http ://blog.delroth.net/