Blockchain CheatSheet - Hashing

🕓 **Temps de lecture : 5 min**

Table des matières

§ Fondamentaux

Le hashing n'est pas le chiffrement car on ne peut pas reconstruire les données originales à partir du hash comme on le fait avec les fichiers chiffrés.

On peut considérer le hashing comme une empreinte digitale ; il fournit une référence génétique sécurisée des données mais n'est pas les données "en personne".

Caractéristiques clés d'un bon hachage cryptographique

1. **Vitesse** : Il doit être facile à calculer dans une certaine mesure, car nous ne voulons pas que l'algorithme soit facilement soumis à des attaques par force brute en raison de sa vitesse.
2. **Déterminisme** : La même entrée doit toujours produire la même sortie.
3. **Unidirectionnel** : Il doit être infaisable de recréer les données originales à partir du hash. En particulier, c'est difficile car lors du hashing, nous pouvons perdre des données.
4. **Sécurisé** : Si vous modifiez les données à hacher, vous obtiendrez un hash complètement différent, mais si vous modifiez à nouveau pour revenir en arrière, vous obtiendrez le hash original.
5. **Collision** : Il est impossible que deux ensembles de données différents aient la même valeur de hash, donc le hashing est sécurisé contre les collisions*.
6. **Taille** : Peu importe la taille des données à hacher. La pratique du hashing a généralement une grande capacité.

> - **Problème de collision** : La préoccupation principale n'est pas seulement la probabilité que deux hash collident, mais plutôt la probabilité qu'au sein d'un ensemble de données, il y ait au moins deux points de données identiques avec le même hash. Cette probabilité augmente considérablement avec la taille de l'ensemble de données, similaire au paradoxe de l'anniversaire.

Salage

Salage est la pratique d'ajouter une valeur aléatoire au mot de passe haché stocké. C'est le seul moyen de hacher les mots de passe de manière sécurisée.

Mineurs

La tâche des mineurs est de prendre les transactions ou les données du tampon de la blockchain et de les regrouper en blocs. Chaque en-tête de bloc fait 80 octets.

Avant d'ajouter ces blocs à la blockchain, les mineurs doivent inclure un hash de 32 octets et un nonce qui répond aux exigences de difficulté actuelles.

Ils le font en parcourant différentes valeurs de nonce jusqu'à ce qu'ils trouvent une valeur qui produit un hash satisfaisant les conditions de preuve de travail.

---

§ Mathématiques du Hashing

Aperçu

- ***SHA (Secure Hash Algorithm)*** : markdown Copia codice `1. SHA-1 : 160 bits 2. SHA-2 : - SHA-224 : 224 bits - SHA-256 : 256 bits - SHA-384 : 384 bits - SHA-512 : 512 bits 3. SHA-3 : - SHA3-224 : 224 bits - SHA3-256 : 256 bits - SHA3-384 : 384 bits - SHA3-512 : 512 bits`
- ***Termes techniques*** :
  - **Padding** : Ajout de bits pour indiquer la fin du message.
  - **Padding avec des zéros** : Ajout de bits '0' pour atteindre une longueur spécifique.
  - **Ajouter la longueur** : Ajout de la longueur originale du message en bits.
  - **Fonction de compression** : Le processus de mélange des bits qui inclut des opérations cryptographiques.
  - **Valeur de hash** : Le code secret unique résultant.

Procédure

1. Préparer le message

Notre cas : Imaginons que vous avez une phrase, par exemple : "Bonjour le monde". C'est notre entrée. Calculez la longueur de l'entrée en bits (88 bits dans ce cas).

2. Ajouter un signal de fin (Padding)

Pour indiquer à l'algorithme que la phrase est terminée, nous ajoutons un symbole spécial à la fin.

Notre cas : Nous ajoutons un bit '1'. Ce signal est le bit de padding. Nous avons donc maintenant "Bonjour le monde1".

3. Structures de blocs

L'algorithme préfère travailler avec des blocs d'une certaine taille, pour optimiser la puissance de calcul. Pour SHA-256, la taille du bloc est de 512 bits (64 octets) à la fois.

Données petites - Ajouter des pièces manquantes (Padding avec des zéros)

Notre cas : Si la phrase n'est pas assez longue comme "Bonjour le monde1", nous ajoutons des zéros pour la remplir. Donc, si "Bonjour le monde1" fait 88 bits, nous ajoutons 424 zéros supplémentaires pour atteindre 512 bits.

Données grandes - Portionnement

Si les données à hacher sont plus longues que 512 bits, l'algorithme effectue plusieurs passes sur les morceaux de données.

4. Ajouter la longueur (Ajouter la longueur)

À la fin, nous ajoutons la longueur du message original en bits, comme l'exigent les règles de padding de SHA-256.

Notre cas : "Bonjour le monde" faisait 88 bits, donc nous ajoutons une représentation de 64 bits de "88". Nous avons maintenant un total de 512 bits : 448 bits de données et padding + 64 bits de longueur.

5. Mélanger les caractères (Fonction de compression)

Maintenant, l'algorithme commence à mélanger les caractères. Il prend chaque bloc de 512 bits et effectue de nombreuses opérations complexes sur eux, modifiant les bits de manière très compliquée que seul l'algorithme connaît. Cette étape comprend des opérations telles que XOR, décalages de bits et additions modulaires.

Données grandes - Racine Merkle

L'algorithme prend tous les morceaux de 512 bits et les concatène par paires, en effectuant le hashing encore et encore jusqu'à obtenir un hash de 256 bits.

Techniquement, en exécutant un processus de réduction sur de grands groupes de données hachées en un seul hash, appelé **Racine Merkle**.

6. Obtenir le code secret (Valeur de hash)

Après que l'algorithme a terminé le mélange, nous obtenons un code secret unique appelé hash ou digest, comme "a7b9c3d2". Ce code est spécial car même si vous changez juste une lettre du message original, le hash sera complètement différent.

---

§ Applications

Le hashing est utile pour vérifier si des données ont été corrompues ou modifiées dans une période définie depuis leur création ou pour certifier l'origine des données. Cela est possible en vérifiant le Hash de T0 avec T1.

Suivi suggéré

Blockchain CheatSheet - Cryptographie & Signatures

Auteur : Kenneth Boldrini

4o mini