Blockchain CheatSheet - Hashing

🕓 **Tempo di lettura: 5 min**

Indice

§ Fondamenti

L'hashing non è cifratura perché non è possibile ricostruire i dati originali a partire dall'hash come si farebbe con i file cifrati.

Dovremmo considerare l'hashing come un'impronta digitale; fornisce un riferimento genetico sicuro ai dati ma non è i dati "in persona".

Caratteristiche Chiave di un Buon Hashing Criptografico

1. **Velocità**: Deve essere facile da calcolare fino a un certo punto, poiché non vogliamo che l'algoritmo sia facilmente attaccabile tramite forza bruta a causa della sua velocità.
2. **Deterministico**: La stessa input dovrebbe sempre produrre lo stesso output.
3. **Unidirezionale**: Deve essere difficile ricreare i dati originali a partire dall'hash. In particolare, è difficile perché durante l'hashing potremmo perdere dati.
4. **Sicuro**: Se modifichi i dati da hashare, otterrai un hash completamente diverso, ma se modifichi di nuovo per tornare indietro, otterrai l'hash originale.
5. **Collisione**: È impossibile che due set di dati diversi abbiano lo stesso valore di hash, quindi l'hashing è sicuro contro le collisioni*.
6. **Dimensione**: Non importa quanto siano grandi i dati da hashare. La pratica dell'hashing ha generalmente una grande capacità.

> - **Problema di Collisione**: La preoccupazione principale non è solo la probabilità che due hash collidano, ma piuttosto la probabilità che, all'interno di un set di dati, ci siano almeno due punti di dati identici con lo stesso hash. Questa probabilità aumenta significativamente con la dimensione del set di dati, simile al paradosso dell'anniversario.

Salting

Salting è la pratica di aggiungere un valore casuale alla password hashata memorizzata. Questo è l'unico modo per hashare le password in modo sicuro.

Minatori

Il compito dei minatori è di prendere le transazioni o i dati dal buffer della blockchain e raggrupparli in blocchi. Ogni intestazione di blocco è di 80 byte.

Prima di aggiungere questi blocchi alla blockchain, i minatori devono includere un hash di 32 byte e un nonce che soddisfi i requisiti di difficoltà attuali.

Lo fanno passando attraverso diverse valori di nonce finché non trovano uno che produce un hash che soddisfi le condizioni di proof-of-work.

---

§ Matematica dell'Hashing

Panoramica

- ***SHA (Secure Hash Algorithm)*** : markdown Copia codice `1. SHA-1 : 160 bit 2. SHA-2 : - SHA-224 : 224 bit - SHA-256 : 256 bit - SHA-384 : 384 bit - SHA-512 : 512 bit 3. SHA-3 : - SHA3-224 : 224 bit - SHA3-256 : 256 bit - SHA3-384 : 384 bit - SHA3-512 : 512 bit`
- ***Termini Tecnici*** :
  - **Padding** : Aggiunta di bit per indicare la fine del messaggio.
  - **Padding con Zeri** : Aggiunta di bit '0' per raggiungere una lunghezza specifica.
  - **Aggiungere la Lunghezza** : Aggiunta della lunghezza originale del messaggio in bit.
  - **Funzione di Compressione** : Il processo di miscelazione dei bit che include operazioni criptografiche.
  - **Valore dell'Hash** : Il codice segreto unico risultante.

Procedura

1. Preparare il Messaggio

Il nostro caso: Immagina di avere una frase, per esempio: "Ciao Mondo". Questa è la nostra input. Calcola la lunghezza dell'input in bit (88 bit in questo caso).

2. Aggiungere un Segnale di Fine (Padding)

Per indicare all'algoritmo che la frase è finita, aggiungiamo un simbolo speciale alla fine.

Il nostro caso: Aggiungiamo un bit '1'. Questo segnale è il bit di padding. Ora abbiamo "Ciao Mondo1".

3. Strutture dei Blocchi

L'algoritmo preferisce lavorare con blocchi di una certa dimensione, per ottimizzare la potenza di calcolo. Per SHA-256, la dimensione del blocco è di 512 bit (64 byte) alla volta.

Dati Piccoli - Aggiungere Pezzi Mancanti (Padding con Zeri)

Il nostro caso: Se la frase non è abbastanza lunga come "Ciao Mondo1", aggiungiamo zeri per riempirla. Quindi, se "Ciao Mondo1" è lunga 88 bit, aggiungiamo altri 424 zeri per arrivare a 512 bit.

Dati Grandi - Porzionamento

Se i dati da hashare sono più lunghi di 512 bit, l'algoritmo esegue più passaggi sui pezzi di dati.

4. Aggiungere la Lunghezza (Aggiungere la Lunghezza)

Alla fine, aggiungiamo la lunghezza del messaggio originale in bit, come richiesto dalle regole di padding di SHA-256.

Il nostro caso: "Ciao Mondo" era lunga 88 bit, quindi aggiungiamo una rappresentazione di 64 bit di "88". Ora abbiamo un totale di 512 bit: 448 bit di dati e padding + 64 bit di lunghezza.

5. Mescolare i Caratteri (Funzione di Compressione)

Ora l'algoritmo inizia a mescolare i caratteri. Prende ogni blocco di 512 bit e esegue molte operazioni complesse su di essi, modificando i bit in un modo molto complicato che solo l'algoritmo conosce. Questo passaggio include operazioni come XOR, spostamenti di bit e aggiizioni modulari.

Dati Grandi - Radice Merkle

L'algoritmo prende tutti i pezzi di 512 bit e li concatena a coppie, eseguendo l'hashing ancora e ancora fino a ottenere un hash di 256 bit.

Tecnologicamente, eseguendo un processo di riduzione su grandi gruppi di dati hashati in un unico hash, chiamato **Radice Merkle**.

6. Ottenere il Codice Segreto (Valore dell'Hash)

Dopo che l'algoritmo ha terminato il mescolamento, otteniamo un codice segreto unico chiamato hash o digest, come "a7b9c3d2". Questo codice è speciale perché anche se cambi solo una lettera del messaggio originale, l'hash sarà completamente diverso.

---

§ Applicazioni

L'hashing è utile per verificare se alcuni dati sono stati corrotti o modificati in un periodo definito dalla loro creazione o per certificare l'origine dei dati. Questo è possibile confrontando l'Hash da T0 con T1.

Seguito Suggerito

Blockchain CheatSheet - Criptografia e Firme

Autore: Kenneth Boldrini