Blockchain CheatSheet - Technische Nutzung

🕓 **Lesezeit: 5 m**

Inhaltsverzeichnis

§ Adressen

Anwendungsfälle

- Transaktionen mit einem öffentlichen Schlüssel signieren zur Identifizierung und Validierung von Daten.
- Jeder, der den öffentlichen Schlüssel besitzt, kann die Daten identifizieren und validieren.

Die Schritte

1. **Erzeugen von Schlüsselpaaren**:
   - Erstellen eines ***Privaten Schlüssels***: `256 Bit oder 64 hexadezimale Zeichen`
     - Zufällig erzeugt.
   - Ableiten des ***Öffentlichen Schlüssel-Basis***: `512 Bit oder 128 hexadezimale Zeichen`
     - Den ***Privaten Schlüssel*** mit dem *Elliptic Curve Digital Signature Algorithm* verwenden (Algorithmus => x*coordinate-256bit + y_coordinate-256bit = 512 Bit* **Öffentlicher Schlüssel-Basis**_).
2. **Hashing (Ethereum)**:
   - Hash der ***Öffentlichen Schlüssel***: `Von 512 Bit auf 256 Bit oder 64 hexadezimale Zeichen`
     - Den ***Öffentlichen Schlüssel-Basis*** mit *Kekkak-256* oder *Sha-3* hashen.
3. **Generierung der Öffentlichen Adresse (Ethereum)**:
   - Erstellen einer ***Öffentlichen Adresse***: `Von 64 hexadezimalen Zeichen auf 42 hexadezimale Zeichen`
     - Die letzten 40 hexadezimalen Zeichen (20 Byte) nehmen und mit 0x voranstellen, um 42 hexadezimale Zeichen zu erhalten.

---

§ Krypto-Transaktionen

Analogie

Stellen Sie sich vor, dass die Parteien **A**, **B** und **C** jeweils eine *Sicherheitstruhe* haben, die Inhalte enthält, die durch das Blockchain-Protokolls-System reisen, das die Regeln festlegt, wie alles funktioniert. Diese *Sicherheitstruhen* haben einen Schlitz, der nur eingehende

Inhalte akzeptiert, und der einzige Weg, den Inhalt zu erhalten, ist mit dem privaten Schlüssel des Eigentümers.

Mechanik der Transaktionen

A Sendet -> an B Daten oder Kryptowährung

1. **B** erstellt ***Öffentliche Adresse*** und ***Öffentlichen Schlüssel*** aus ***Privatem Schlüssel***:
   - ***Private Schlüssel*** von **B** :: ***Öffentliche Adresse*** und ***Öffentlicher Schlüssel*** von **B**.
2. **B** sendet die ***Öffentliche Adresse*** -> an **A** (*Die öffentliche Adresse kann sich bei jeder Transaktion ändern*).
   - ***Öffentliche Adresse*** von **B** -> an **A**.
3. **A** fügt die ***Öffentliche Adresse*** von **B** und die Daten oder den Betrag zu einer "Transaktionsnachricht" hinzu:
   - **A** Initialisiert Transaktion :: ***Öffentliche Adresse*** von **B** und Inhalt.
4. **A** signiert die Transaktion mit der ***Digitalen Signatur***:
   - ***Digitale Signatur*** :: Abgeleitet aus **A**'s eigenem Privaten Schlüssel mit dem *Elliptic Curve Digital Signature Algorithm* (x_coordinate-256bit + y_coordinate-256bit).
5. Die Transaktion von **A** wird vom Blockchain-Protokoll im *Memory Pool* *Vorgeschlagen*:
   - ***Validierung*** :: Miner versuchen, die Transaktion zu validieren, indem sie sie in einen Block des Memory Pools aufnehmen.

Validierung des Vorschlags

B sendet dann -> an C

- Vor der Aufnahme der Transaktion in die Blockchain muss überprüft werden, ob **B** tatsächlich den erforderlichen Inhalt hat, um ihn erneut zu senden: **Transaktion von B** wird -> zum **Memory Pool** der Blockchain gesendet und dann vom Protokoll -> an **C** weitergeleitet.

Krypto-Transaktionen Bitcoin im Detail

Bitcoin vs Ethereum

- **Bitcoin**: Jede Transaktion muss als Behälter einer einzigartigen, nicht gemischten Kryptowährung betrachtet werden.
- **Ethereum**: Verwendet ein Buchhaltungssystem, das den Gesamtbetrag verfolgt, im Gegensatz zu Bitcoin.

Transaktionsmanagement

Kryptowährung, da sie an Transaktionsbehälter gebunden ist, die wir *X*-Trsct-C*n* nennen (*X* = ID, Trsct = Transaktion, C*n* = Behälternummer), muss durch Manipulation des Behälters verwaltet werden.

A sendet 10 Bitcoin -> an B aus einem Transaktionsbehälter, der 20 Bitcoin enthält

1. **B** erstellt ***Öffentliche Adresse*** und ***Öffentlichen Schlüssel*** aus ***Privatem Schlüssel***.
2. **B** sendet die ***Öffentliche Adresse*** -> an **A** (*Die öffentliche Adresse kann sich bei jeder Transaktion ändern*).
3. **A** fügt die ***Öffentliche Adresse*** von **B** und den Betrag zu einer "Transaktionsnachricht" hinzu.
4. Der *Neue leere Transaktionsbehälter* (**A**-Trsct-C4) nimmt ein Eingangs- und ein oder zwei Ausgangswerte, den Betrag und ggf. das Wechselgeld:
    - Der Eingang basiert auf den Transaktionsbehältern, die die nicht ausgegebene Kryptowährung oder UTXO (Unspent Transaction Output) enthalten, die den Betrag der neuen Transaktion abdeckt.
        - A-Trsct-C1 = 10 Bitcoin
        - A-Trsct-C2 = 30 Bitcoin -> Eingang
        - A-Trsct-C3 = 5 Bitcoin
    - Der erste Ausgang wird der Betrag der neuen Transaktion sein.
        - A-Trsct-C4 = 20 Bitcoin -> Ausgang an B-Trsct-C1
    - Der optionale Ausgang wird das Wechselgeld sein, das an den Absender A zurückgesendet wird.
        - A-Trsct-C4 = 10 Bitcoin -> Ausgang an A-Trsct-C4
        - !!!
        - A-Trsct-C2 = 30 Bitcoin wird dann zerstört
5. **A** signiert die Transaktion mit der ***Digitalen Signatur***.
6. Die Transaktion von **A** wird vom Blockchain-Protokoll im *Memory Pool Vorgeschlagen*.
7. Validierung des *Vorschlags*.

Miner Proof of Work Validierung

Diagramm

mathematica

Copia codice

```
`Transaktionen/Daten -> 80 Byte Transaktionsgruppe
v                                                  Block-Header
```

```
|                                                                    v
* Nonce-Suche
|                                                                    v
Hash-Algorithmus-Berechnung
|                                                  ----------------------------
-----                                              |
|                                  Gültiger Hash                  Ungültiger
Hash                                               |
|                  # Block validiert und hinzugefügt   * Nonce-Inkrement
zur Blockchain                 Wiederholung des                          &
Miner wird belohnt              Prozesses`
```

Struktur des Blockchain-Blocks

Im Kontext der Blockchain erstellen Miner Blöcke mit einer spezifischen Struktur. Ein typischer Bitcoin-Block-Header ist 80 Byte groß und enthält Folgendes:

- 4 Byte: Versionsnummer
- 32 Byte: Hash des vorherigen Blocks
- 32 Byte: Merkle-Wurzel (Hash der Transaktionen im Block)
- 4 Byte: Zeitstempel
- 4 Byte: Schwierigkeitsziel
- 4 Byte: Nonce

> In der Regel sind die einzigen Unterschiede zwischen den Hash-Versuchen der Miner:
>
> - Der Hash der Daten (der erste Teil davon ist die Belohnung für den Miner).
> - Der Zeitstempel (der je nach Standort und Anzahl der Versuche, den Nonce zu finden, variieren kann).
> - Der Nonce selbst.
> - Außerdem kann die Reihenfolge, in der die Daten gruppiert sind, zwischen den Minern variieren.

---

§ Skalierbarkeit

Schichten

- **Layer 0**: Internet, wie wir es kennen.
- **Layer 1**: Blockchain Layer 1 Transaktionen sind langsamer als traditionelle Methoden, bis zu 10 Minuten für eine Abwicklung.

- **Layer 2**: Wallets für kleinere Transaktionen schneller.

Layer 2 Lightning Network

Beschreibung: Das Lightning Network ist eine Off-Chain-Lösung, die als Zahlungskanal fungiert, aufgebaut auf einer Netzwerkstruktur, die Benutzer verbindet. Es ermöglicht die Verarbeitung von Transaktionen, ohne jede Transaktion in der Bitcoin-Blockchain aufzuzeichnen.

Was es löst: Es erhöht die Transaktionsgeschwindigkeit erheblich, indem es ein Doppel-Signatursystem als Vereinbarung für den Austausch verwendet.

Wie es funktioniert: Wenn Kunden eine Zahlung tätigen müssen, senden beide Parteien eine Transaktion wie in der Transaktionsverwaltung beschrieben. Der Absender legt den fälligen Betrag fest und der Empfänger stellt eine Transaktion mit einem nahezu null Wert ein. Die Zahlungsanforderung reist durch das Netzwerk und sucht den kürzesten Pfad von verbundenen Kanälen, um den Empfänger zu erreichen. Jeder Kanal hält ein Guthaben, das zwischen den beteiligten Parteien übertragen werden kann, und nur die Eröffnung und Schließung der Kanäle werden in der Blockchain aufgezeichnet.

Empfohlene Fortsetzung

Blockchain CheatSheet - Konsens

Autor: Kenneth Boldrini