Transaktionen

Ture Claußen, 1531067, ture.claussen@stud.hs-hannover.de und Jannes Neemann, 1530893, jannes.neemann@stud.hs-hannover.de

Fakultät IV, Abteilung Informatik, Hochschule Hannover, Ricklinger Stadtweg 120, 30459 Hannover

Zusammenfassung. Schlüsselwörter:

Inhaltsverzeichnis

1 Einführung

2 Struktur und technische Umsetzung einer Transaktion

Die Komponenten welche eine Transaktion in Ethereum ausmachen sind vergleichbar mit denen eines Briefes. Sie besitzen einen Empfänger sowie eine Frankierung, welche die Transportkosten zum Empfänger bezahlen. Außerdem besitzt eine Transaktion Ether oder Nutzdaten zu verschicken, wie es bei einem Brief ebenfalls möglich ist. Ether wären dabei ein Geldschein und Nutzdaten ein Text oder eine Karte. Im folgenden sollen die einzelnen Felder die eine Transaktion ausmachen vorgestellt werden.

2.1 Komponenten von Transaktionen

Allgemein enthält eine Transaktion, wie sie im offiziell [3, S. 4] definiert ist, folgende Felder:

nonce: Ein Skalar welcher gleich der Anzahl der vom EOA versendeten Transaktionen ist. Der Nutzen wird in 3.1 erläutert.

gasPrice: Ein Skalar der angibt, wie viel Wei man pro Einheit *Gas* bezahlt, die bei der Gesamtheit aller Berechnungen die während der Ausführung der Transaktion anfallen (S. 3.2)

gasLimit: Ein Skalar der die maximal Anzahl an *Gas* angibt, die während der Ausführung der Transaktion verbraucht werden darf. Dieser Betrag muss im Voraus bezahlt werden.
to: Die 160-Bit Adresse des Empfängers.
value: Skalar der die Menge Wei angibt, die der Empfänger erhält.
v,r,s: Komponenten der ECDSA-Signatur (S. 3.4), um den Sender der Transaktion zu bestimmen
init: Contract Creation
data: Ein Byte-Array unbegrenzter Länge, welches die Nutzdaten des Kontrakts enthält

2.2 Typen von Transaktionen

2.3 Serialisierung

3 Aufbau einer Transaktion

3.1 Nonce

3.2 Gas

Gas ist ein zentraler konzeptioneller Lösungsansatz im Rahmen von Ethereum. Da Ethereum turing-vollständig ist [3, S. 1], ergibt sich unter anderem das sogenannte "Halteproblem". Dieses besagt, dass im Voraus nicht vorhergesagt werden kann, ob das Programm einer Turing-Maschine jemals zu einem Ende kommt. [1, S.70] Um die Funktionalität des Netzwerks zu gewährleisten, wird die Laufzeit einer jeden Zustandsveränderung der Blockchain, sprich Transaktion, durch Gas begrenzt.

Gas ist eine eigenständige Währung innerhalb von Ethereum, dessen Einheit einen Rechenschritt in der EVM bemisst [2, S. 9:3], wobei für jeden Opcode die Kosten in Gas spezifiziert werden. [3, S. 25 ff.] Gas ist also eine Gebühr für Rechnenaufwand. Vor jeder Transaktion muss der externe Akteur festlegen, welchen Rechenaufwand er zu zahlen bereit ist.

Das Datenfeld *gasPrice* definiert bei Transaktionen welcher Preis pro Einheit Gas gezahlt werden soll. Dem entsprechend bezeichnet das Feld einen skalaren Wert in Wei. Gas kann bewusst nur mit Ether erworben werden, da die Gas-Preise möglichst unabhändig von den Preisschwankungen (von Ether) sein sollen. [3, S. 7]

Das *gasLimit* wiederrum gibt an, wie viel Gas für diese Transaktion maximal aufgewendet werden darf. Somit gilt es im Voraus abzuschätzen wie hoch der Rechenaufwand sein wird. Gerade wegen des Halteproblems kann dies aber nur grob vorgenommen werden. Eine Größe sind zunächst die intrinsischen Kosten einer Transaktion. Das ist der Overhead der allein durch die Transaktion und deren Inhalt betsteht. Diese intrinsischen Kosten lassen sich schon vor Ausführung mit folgendem Algorithmus validieren.

Preis und Latenz

Anreiz und Spieletheorie

3.3 Value und Data

3.4 Signatur

ECDSA

Multisignaturen

4 Transaktionsabwicklung

4.1 Propagation

4.2 Speicherung

5 Ausblick

Literatur

1. Davis, M.: Computability and Unsolvability. Courier Corporation (Apr 2013)
2. M.Spain, M.Foley: OASIcs-Tokeneconomics. Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany (2019)
3. Wood, G.: Ethereum/yellowpaper. 2019-10-20 (Oct 2019)