Vorwissenschaftliche Arbeit

Blockchain-Technologie in Bezug auf Bitcoin

Elias Kounakas
8B

BG|BRG Mössinger
9020 Klagenfurt am Wörthersee

Betreuer: Mag. Franz Furtschegger
Abgabedatum: 21.02.2023

Abstract

Immer öfter hört man in den Medien von Kryptowährung, vor allem aber Bitcoin. Viele Menschen machen sich Gedanken darüber, ob sie in Bitcoin investieren sollten und ob es denn das Zahlungsmittel der Zukunft sei, ohne aber zu wissen: Wie funktioniert Bitcoin überhaupt?

Das Ziel dieser Arbeit ist es, herauszufinden, wie die Einzelteile des Bitcoin-Netzwerkes zusammenarbeiten, um eine dezentrale Währung zu kreieren. Genutzt werden vor allem literarische Werke, aber auch ein selbstinstallierter Knotenpunkt, welcher Daten im Bitcoin-Netzwerk einsehen kann. Als Hauptquelle dient hierzu das Buch "Mastering Bitcoin: Programming the Open Blockchain", da es die meisten Aspekte von Bitcoin detailliert behandelt.

Inhaltsverzeichnis

1 Einleitung

Bitcoin basiert auf einem dezentralen Netzwerk von Computern, die die Transaktionen validieren und auf einer öffentlich zugänglichen Blockchain aufzeichnen. Eine Blockchain ist im Grunde eine dezentrale Datenbank, die alle Transaktionen enthält, die jemals in einem Netzwerk durchgeführt wurden. Jede Transaktion wird von den Benutzern des Netzwerks validiert und durch ein komplexes kryptografisches Verfahren gesichert, um sicherzustellen, dass sie nicht manipuliert werden kann.

Das Ziel dieser Arbeit besteht darin, die innere Funktionsweise von Bitcoin und allen dazugehörigen Technologien, besonders die Blockchain, zu untersuchen. Es wird untersucht, wie das Bitcoin-Netzwerk aufgebaut ist, wie es zustande kam und welche Individuen daran teilnehmen können. Des Weiteren wird die Eröffnung eines Kontos im Bitcoin-Netzwerk erläutert und wieso diese Konten praktisch nicht kompromittierbar sind. Außerdem wird gezeigt, wie Transaktionen getätigt werden können und wie diese ablaufen.

Hierbei handelt es sich um eine reine Literaturarbeit. Das Hauptwerk, auf welches sich bezogen wird, ist "Mastering Bitcoin" von Andreas Antonopoulos. (Antonopoulos 2017) Es beschreibt vor allem Aufbau und Funktion des Bitcoin-Netwerkes und wie alle Einzelteile zusammen arbeiten. Für unterstützende Technologiem, welche Bitcoin benötigt, wird vor allem Mario Stipčevićs Arbeit "True Random Number Generators" zur Entdeckung von natürlichen, zufälligen Zahlen genutzt. (Stipčević 2014) Außerdem wurde eine Bitcoin-Full-Node von zu Hause aus zur Erforschung der Eigenschaften des Bitcoin-Netzwerkes betrieben.

1.1 Was ist Bitcoin?

Häufig wird Bitcoin einzig und allein mit der Kryptowährung selbst assoziiert. Der Begriff Bitcoin umfasst jedoch alle Konzepte und Technologien, die das digitale Zahlungssystem ermöglichen. (vgl. Antonopoulos 2017, S. 1) Eines dieser Konzepte ist das Bitcoin-Netzwerk, wodurch Nutzer Zahlungen propagieren und verifizieren können. Mit der Kryptowährung Bitcoin selbst kann man Käufe tätigen, Geld an Verwandte oder Organisationen versenden und Waren verkaufen, genauso wie es bei einer herkömmlichen Währung der Fall ist. Der große Unterschied zwischen Bitcoin und herkömmlicher Währung ist, dass es kein handgreifliches Bitcoin gibt. Die Menge an Bitcoin, die man besitzt, wird durch die Zusammenfassung aller erhaltenen Transaktionen auf die eigene Bitcoin-Addresse erlangt.

Als Grundlage für Transaktionen in Bitcoin dient die asymmetrische Kryp-

tographie, ein kryptographisches System, welches auf einem Schlüsselpaar basiert. Das Schlüsselpaar setzt sich aus dem privaten Schlüssel und öffentlichen Schlüssel zusammen. Mit dem öffentlichen Schlüssel wird die Bitcoin-Addresse des Nutzers abgeleitet, welche benötigt wird, um Zahlungen zu empfangen. Ähnlich wie der PIN oder das Passwort bei einem herkömmlichen Bankkonto lassen sich Zahlungen mit dem privaten Schlüssel im Bitcoin-Netzwerk tätigen.

In Bitcoin gibt es keine zentrale Macht oder Bank, die neues Geld produziert. Stattdessen werden neue Bitcoins mit einem Prozess namens Mining gewonnen, welcher die Lösung zu einem mathematischen Problem sucht und gleichzeitig neue Transaktionen verifiziert. Alle 10 Minuten wird eine neue Lösung gefunden und alle Miner starten wieder von vorne. Das bedeutet, dass jeder Nutzer auch ein Miner sein kann und Transaktionen maximal 10 Minuten brauchen, um verifiziert zu werden.

Die kleinste Einheit von Bitcoin ist der Satoshi, welcher nach dem Erfinder von Bitcoin, Satoshi Nakamoto benannt wurde. Ein Bitcoin entspricht dem Wert von 100 Millionen Satoshis. Da der Wert von Bitcoin in den letzten Jahren exponentiell gestiegen ist, gewann der Satoshi immer mehr an Bedeutung. Transaktionen haben meist einen 8-stelligen Dezimalwert unter 1 (Bsp.: 0,00140209 BTC), was konventionell schlecht darstellbar ist. Hier wäre es sinnvoll, den Wert in Satoshis anzugeben, also 140 209 Satoshis.

1.2 Geschichte von Bitcoin

2008 veröffentlichte jemand unter dem Alias Satoshi Nakamoto "Bitcoin: A Peer-to-Peer Electronic Cash System". (vgl. Nakamoto 2009, S. 1) Mit dieser Arbeit erfand Nakamoto die erste Kryptowährung Bitcoin. Er kombinierte frühere Erfindungen mit seinen Ideen und erfand somit das erste komplett dezentralisierte, elektronische Zahlungssystem. Die Schlüsselrolle spielte die Implementierung von einem verteilten Aufwand der Prozessorkraft, wodurch alle 10 Minuten demokratisch für die korrekte Folge von Aktionen abgestimmt werden kann.

In dem Jahr 2011, als Nakamoto sich von seinem Projekt abwandte und es nicht länger optimierte, stieg Bitcoin mit einem Aufschwung von 3000% in nur drei Monaten von 1USD auf 32USD. Explodiert ist der Wert jedoch erst 2018, als Bitcoin einen Höchstwert von 19,000USD erreichte. Im vierten Quartal des Jahres 2021 hat Bitcoin seinen bislang höchsten Wert (Stand: 24/08/2022) von 67,000USD erreicht.

2 Struktur des Bitcoin-Netzwerkes

2.1 Peer-to-Peer-Netzwerk

Im Bitcoin-Netzwerk gibt es keine zentrale Macht. Alle verbundenen Computer arbeiten miteinander als Peers. Peer-to-Peer bzw. P2P heißt, dass alle teilnehmenden Computer die gleichen Rechte und Hindernisse haben. Die einzelnen verbundenen Computer nennt man Nodes. Jede Node kommuniziert rund um die Uhr mit ihren benachbarten Nodes. Nodes können Informationen abfragen, weiterleiten und verifizieren. Das Bitcoin-Netzwerk ist demnach die Gesamtheit aller Nodes.

Was passiert aber, wenn eine Node für bösartige Zwecke falsche Informationen an andere Nodes verschickt? Dafür hat man ein Vertrauens-System im Bitcoin-Netzwerk implementiert. Bevor eine Node empfangene Informationen abspeichert, prüft sie diese auf Korrektheit in Bezug auf den derzeitigen Standard. Dieser Standard wird im BIP (Bitcoin Improvement Proposal) festgelegt, welcher periodisch immer neue Richtlinien festlegt. (vgl. Taaki 2011, S. 1) Natürlich hat das BIP keine Macht über die Nodes, diese können nämlich selber entscheiden, ob sie nun nach den neuen Richtlinien arbeiten wollen oder nicht. Jedoch übernehmen die meisten Nodes mit einem kurzen Update den neuen Standard, da dieser lediglich für das Interesse der Teilnehmer entwickelt wird.

Bitcoin Nodes sind nicht leistungshungrig und benötigen wenig Energie, solange sie nicht für Mining genutzt werden. Deswegen ist es für so gut wie jede Person möglich, selber zu Hause eine Bitcoin Node zum Laufen zu bringen. Die Zwecke für diese Node können aber von Person zu Person variieren. Zum einen gibt es Leute, die mit ihrer Node nur mithören wollen, zum anderen aber auch diejenigen, die durch Mining Geld verdienen wollen. Aus diesem Grunde gibt es verschiedene Implementierungen und Rollen von Nodes.

2.1.1 Full Node

Eine Full Node benötigt von allen Nodetypen am meisten Speicherplatz, ist dafür aber die mächtigste und fasst alle Funktionen in einer Node zusammen. Der distinkte Unterschied von der Full Node ist, dass diese die komplette Blockchain herunterlädt und verifiziert. Der benötigte Speicherplatz der Blockchain liegt bei 423GB (Stand: 25/8/2022).

Damit die Blockchain erweitert wird und alle Transaktionen verifiziert werden können, müssen sogenannte Mining Nodes alle 10 Minuten eine Lösung zu einem mathematischen Problem finden, indem sie mit der Hashfunktion SHA256 einen bestimmten Wert suchen. Das muss vor allem schnell und prä-

zise geschehen, damit die Chance, die Lösung als Erstes zu finden, maximiert wird. Mining Nodes profitieren vor allem von modernen GPUs, welche Hunderte Millionen Hashes pro Sekunde berechnen.

Die wahrscheinlich beste Node fÍür den Otto Normalverbraucher ist der SPV Client (Simplified Payment Verification Client). Dieser installiert lediglich eine Wallet, eine Kopie der Block Header und stellt eine Verbindung zum Bitcoin-Netzwerk her. Er ist einzig und allein dafür ausgestattet, Transaktionen auf das eigene Konto zu empfangen und Geld zu versenden. (vgl. Antonopoulos 2017, 172f)

2.2 Die Blockchain

Die Blockchain ist eine Datenstruktur, welche Transaktionen in miteinander verbundenen Blöcken speichert. Visualisieren kann man diese wie einen Turm von Blöcken. Der unterste Block dient als Fundament und die Anzahl der Blöcke bestimmt die Höhe. Ein neuer Block wird hinzugefügt, wenn ein Miner eine neue Lösung mit der SHA256 Hashfunktion gefunden hat. Der Miner kann dann aussuchen, welche Transaktionen er in seinem Block inkludieren möchte. Bei einer Transaktion kann man eine beliebig hohe Spende an den Miner abgeben. Transaktionen mit einer hohen Spende haben eine größere Wahrscheinlichkeit, im nächsten Block inkludiert zu werden.

Nun kann es jedoch passieren, dass zwei Miner gleichzeitig eine Lösung finden und im Bitcoin-Netzwerk zwei verschiedene Versionen der Blockchain im Umlauf sind. Diese Diskrepanz wird aufgelöst, sobald einer der Ketten einen nächsten Block bekommt. Der Grund dafür ist, dass das Bitcoin-Netzwerk nach einem Konzept namens "Proof of workärbeitet. Wenn ein neuer Block hinzugefügt wird, kann man sich sicher sein, dass ein Miner Prozesskraft und Energie dafür aufopfern musste. Die Kette, welche im Endeffekt mehr Blöcke besitzt, wird als gültig anerkannt.

Einer der bekanntesten Wege, um das Bitcoin-Netzwerk anzugreifen, ist der "51% Attack". Wenn man mit bösartigen Absichten in Besitz von mehr als 50% der Mining-Kraft kommt, kann man zwei schwerwiegende Dinge tun. Zum einen kann das Bitcoin-Netzwerk komplett lahmgelegt werden, indem keine neuen Transaktionen in den Blöcken inkludiert werden. Zum anderen können Zahlungen rückgängig gemacht werden, indem man in der einen Kette die Zahlung akzeptiert hat, dann jedoch eine andere Kette ohne diese Zahlung weiterführt und verlängert. Ein Angriff dieser Art wird jedoch immer unwahrscheinlicher, da die Warscheinlichkeit für einen solchen Angriff exponentiell abfällt. (vgl. Antonopoulos 2017, S. 8)

2.2.1 Struktur eines Blocks

Ein Block ist ein Behälter für eine Datenstruktur, welche für die Einschließung von Transaktionen in der Blockchain sorgt. Der Block besteht aus einem Header, welcher für die Identifizierung des Blocks notwendig ist, gefolgt von einer Liste aller Transaktionen. Im Block Header befinden sich Informationen wie die derzeitige Schwierigkeit zur Lösung des Problems der Miner, einen Zeitstempel, welcher die Entstehung des Blocks angibt und einer Zusatzzahl, welche nützlich für Miner ist.

Zur Identifizierung von Blöcken gibt es zwei Wege: Die Berechnung des Block Header Hashes und die Höhe des Blocks. Einen Block durch seine Höhe zu identifizieren scheint simpler, jedoch gibt es hierbei ein Problem. Bei dem vorher angesprochenen Fall, dass zwei Miner gleichzeitig einen Block finden, scheitert die Identifizierung eines Blocks durch dessen Höhe. Um den Block Header Hash zu berechnen, wird der Block Header (Schwierigkeit, Zeitstempel, Zusatzzahl) zwei Mal mit dem SHA256 Algorithmus gehashed. Hierbei kommt ein 32-byte Hash heraus, welcher mit vielen Nullstellen beginnt. Jeder Block hat einen distinkten Block Header Hash, wodurch es keine doppelten Blöcke geben kann. Der erste Block (Genesis-Block) der Blockchain hat einen Hashwert von 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f. (vgl. Antonopoulos 2017, S. 199)

2.2.2 Genesis-Block

Der Genesis-Block ist der erste Block der Blockchain mit einer Höhe von 0. Anders als die darauffolgenden Blöcke gilt dieser als unveränderbar. Kreiert wurde dieser 2009, und wenn man jeden einzelnen Block auf der Blockchain zurückverfolgt, gelangt man schlussendlich an den Genesis-Block. Standardmäßig fügt jede Implementierung von Bitcoin den Genesis-Block schon bei der Installation hinzu.

3 Adressen

3.1 Bitcoin-Adressen generieren

In Bitcoin werden Adressen verwendet, um Transaktionen zu empfangen und zu senden. Eine Bitcoin-Adresse ist eine Zeichenfolge, die aus einer Reihe von Zahlen und Buchstaben besteht und einem öffentlichen Schlüssel entspricht.

Um eine Bitcoin-Adresse zu generieren, wird der öffentliche Schlüssel mithilfe einer kryptographischen Hashfunktion verarbeitet. Die Hashfunktion nimmt

den öffentlichen Schlüssel als Eingabe und gibt einen Hash als Ausgabe. Der öffentliche Schlüssel selbst wird nicht als Adresse genutzt, weil der Hash weniger Speicherplatz benötigt. Der Hash wird dann mithilfe einer weiteren Mathematikfunktion, der Base58Check-Codierung, in eine lesbare Adresse umgewandelt.

Um große Zahlen mit wenigen Zeichen darzustellen, nutzen Computersysteme häufig alphanumerische Zahlensysteme mit einer Basis, die größer als 10 ist. Das Dezimalsystem nutzt die 10 Ziffern 0-9, während das Hexadezimalsystem zusätzlich die Buchstaben A-F nutzt, wodurch es 16 Ziffern hat. Deshalb sind Zahlen, welche hexadezimal dargestellt werden, kürzer. Base64 ist ein weiteres Zahlensystem, das noch viel kompakter ist. Es nutzt die 26 Buchstaben des Alphabets doppelt, nämlich in Klein- und Großschreibung. Zusätzlich nutzt Base64 die 10 Ziffern 0-9 und die Symbole + sowie /.

Base58 hingegen ist eine Ableitung von Base64, welche ähnlich aussehende Symbole vermeidet, da sie in manchen Schriftarten identisch aussehen können. Diese Symbole sind die Zahl null, der Buchstabe O (großes o), l (kleines L), I (großes i), das Zeichen + und /. Das vollständige Alphabet von Base58 ist:

$$123456789ABCDEFGHJKLMNPQRSTUVWXYZ$$

$$abcdefghijkmnopqrstuvwxyz$$

Base58Check wiederum ist ein Codierungsformat von Base58, welches oft in Bitcoin genutzt wird und einen eingebauten Fehlerüberprüfungscode besitzt. Die Prüfsumme wird vom Hash der codierten Daten abgeleitet und kann deswegen genutzt werden, um Tippfehler und andere Fehler zu erkennen. Wenn die Prüfsumme also nicht mit den codierten Daten übereinstimmt, gilt die eingegebene Bitcoin-Adresse als ungültig. Dies verhindert den Versand an falsche Bitcoin-Adressen und dadurch den Verlust von Geld. (vgl. Antonopoulos 2017, S. 67)

Die meisten Daten in Bitcoin werden Base58Check-codiert, was es einfacher macht, mit den Daten zu arbeiten. Zudem wird ein Präfix bei jeder Base58Check-Codierung genutzt, welche den Datentyp angibt, um welchen es sich handelt. Das ermöglicht dem Nutzer, schnell zu erkennen, welche Art von Daten dargestellt werden. Beispielsweise haben Bitcoin-Adressen in Base58Check den Präfix 0x00, welcher umgewandelt auf Base58 1 ergibt. Private Schlüssel wiederum haben den Präfix 0x80 in Base58Check bzw. 5 in Base58.

3.2 Private Schlüssel

Im Bitcoin-Netzwerk ist der private Schlüssel eine geheime Zahl, die genutzt wird, um Bitcoin-Ausgaben ausgehend von einer bestimmten Bitcoin-Adresse

zu autorisieren. Der private Schlüssel ist eine kritische Komponente der kryptographischen Sicherheit, auf welche das Bitcoin-Netzwertk aufbaut. Genutzt wird der private Schlüssel, um Authentizität bei Transaktionen zu beweisen und versichert dabei, dass die Transaktionen nicht verändert oder gefälscht sind.

Meist sind private Schlüssel in einer sogenannten Wallet, eine Applikation, welche dem Nutzer ermöglicht, Transaktionen zu tätigen, gespeichert. Jede Wallet hat einen einzigartigen privaten Schlüssel, welche für Unterschriften bei Transaktionen genutzt wird und den Besitz der jeweiligen Bitcoins beweist. Es ist essenziell, dass der private Schlüssel geschützt ist und mit niemanden geteilt wird, da jeder mit Zugriff auf den privaten Schlüssel Bitcoins ausgeben kann.

Private Schlüssel werden als Abfolge von Zahlen und Buchstaben dargestellt. Genauer gesagt werden private Schlüssel durch ein bestimmtes Format wie das Base64 Zahlensystem codiert. Folgende Hexadezimalzahl zeigt, wie ein privater Schlüssel im Normalfall aussieht:

$$8a708d03d461a5c5839b53da4c40912ca094cc0edee8574a62d6895d033db7ea$$

Diese Zahl entspricht einem Dezimalwert von ungefähr $6.2 * 10^{70}$. Generiert werden private Schlüssel mit verschiedenen Methoden wie beispielsweise ein Zufallszahlengenerator. Gespeichert werden private Schlüssel im Normalfall auf einem Computer oder Smartphone in einer sicheren Datei.

3.2.1 Private Schlüssel generieren

Prinzipiell sind private Schlüssel eine zufällige Zahl zwischen 1 und 2^{256} bzw. $1,15 * 10^{77}$. Um diese enorme Zahl zu relativisieren: die geschätzte Anzahl der Atome im sichtbaren Universum beträgt 10^{80}. (vgl. Antonopoulos 2017, S. 59) Die Chance, zweimal denselben privaten Schlüssel zu generieren, ist also extrem gering. Die einfachste Methode, um einen privaten Schlüssel zu generieren, ist der Münzwurf: Kopf und Zahl wird jeweils 0 und 1 zugewiesen. Danach muss die Münze 256-mal geworfen werden und man erhält eine 256-Bit lange Zahl. Prinzipiell kann man mit allem, was zufällig geschieht, einen privaten Schlüssel generieren. Natürlich möchte man nicht für jeden privaten Schlüssel 256 Münzwürfe machen, deswegen werden hierfür Rechner genutzt. Ein großes Problem in der Computerwelt ist es jedoch, zufällige Zahlen erzeugen zu können. Computer sind schlicht und ergreifend nicht dazu geeignet, Zufälle generieren zu können.

3.2.2 Natürliche Zufallsquellen

Aufgrund des Kerchoff'schen Prinzips muss die Definition eines Zufallszahlengenerators, der für Kryptografie geeignet ist, beinhalten, dass selbst wenn jedes Detail über den Generator bekannt ist (Schaltplan, Algorithmen usw.), er immer noch vollständig unvorhersehbare Bits erzeugen muss. Im Gegensatz zu PRNGs(Pseudorandom number generator) extrahieren physische (echte Hardware) Zufallszahlengeneratoren Zufälligkeit aus physischen Prozessen, die auf fundamental undeterministischer Weise verhalten, was sie zu besseren Kandidaten für die Erzeugung von wahren Zufallszahlen macht. Physische Prozesse sind (vgl. Stipčević und Koç 2014, 5f):

- Geräusche (Radiowellen, Atmosphärische)
- harmonische Oszillatoren
- Quantenfluktuation

Diese VWA beschäftigt sich mit den geräusch-basierten Zufallsgeneratoren.

Wärmerauschen ist eine Art von Rauschen, das in elektrischen Schaltungen auftritt und durch die thermische Bewegung von Ladungsträgern (normalerweise Elektronen) in elektrischen Leitern verursacht wird. Die thermische Bewegung der Ladungsträger in einem elektrischen Leiter führt zu Schwankungen im Stromfluss, die als Rauschen interpretiert werden können. Dieses Rauschen ist proportional zur Temperatur des Leiters und der elektrischen Leitfähigkeit des Materials. Es ist auch unabhängig von der Art des Leiters und tritt in allen elektrischen Leitern auf, einschließlich Metallen, Halbleitern und sogar Isolatoren. Das Johnson-Nyquist-Rauschen ist ein weißes Rauschen, das bedeutet, dass die Frequenzverteilung des Rauschens gleichmäßig über einen breiten Frequenzbereich verteilt ist. Es hat auch eine bestimmte Spectral Density (Leistung pro Frequenzband) die proportional zur Temperatur und der elektrischen Leitfähigkeit des Leiters ist.

Eine weitere Art des Rauschens ist das chaotische Rauschen, das durch chaotische Systeme erzeugt wird. Chaotische Systeme sind Systeme, die sehr empfindlich auf kleine Änderungen in den Anfangsbedingungen reagieren und dazu neigen, unvorhersehbare und zufällige Muster zu erzeugen. Ein Beispiel für ein chaotisches System, das zur Erzeugung von Chaos-Noise verwendet werden kann, ist der Lorenz-Attraktor, die auf drei gekoppelten Differenzialgleichungen basiert und eine Art von chaotischem Verhalten erzeugt, das als Butterfly-Effekt bekannt ist. (vgl. Jönck und Prill 2003, 3ff)

Manche Arten von Rauschen (insbesondere Johnson-Rauschen) erzeugen sehr kleine Spannungen, die vor der Umwandlung in digitale Form stark ver-

stärkt werden müssen. Die starke Verstärkung führt zu weiteren Abweichungen von der Zufälligkeit aufgrund der begrenzten Bandbreite und nicht-linearen Verstärkung des Verstärkers. Außerdem kann schnelles elektrisches Schalten von binärer Logik, die in der RNG-Schaltung verwendet wird, starke elektromagnetische Störungen verursachen, sodass mehrere RNGs in der Nähe (insbesondere auf einem Chip) tendenziell zur gegenseitigen Synchronisierung neigen, was zu einem starken Rückgang der Gesamtentropie führt. Darüber hinaus können empfindliche Verstärker durch externe elektromagnetische Felder leicht manipuliert werden, was für Kryptoangriffe ausgenutzt werden kann. (vgl. Stipčević und Koç 2014, S. 7)

3.3 Öffentliche Schlüssel

Um Bitcoin-Adressen ihre irreversiblen Eigenschaften zu verleihen, wird das Public-Key-Kryptosystem bzw. das asymmetrische Kryptosystem genutzt. Dieses System wurde 1970 basierend auf mathematischen Funktionen erfunden und ist Grundlage für die heutige Cybersecurity.

Im Laufe der Geschichte wurden mehrere mathematische Funktionen entdeckt, die für Public-Key-Kryptosysteme verwendet werden können wie beispielsweise das Potenzieren von Primzahlen und die Multiplikation eines bestimmten Punktes auf einer elliptischen Kurve (diese Funktion wird auch von Bitcoin genutzt). Das besondere an diesen Funktionen ist, dass man sie nur in eine Richtung berechnen kann bzw. ist es sehr viel schwerer, sie in die andere Richtung zu berechnen. Basierend auf diesen mathematischen Funktionen erlaubt die Kryptographie es, digitale Geheimnisse und nicht fälschbare Unterschriften zu kreieren. (vgl. Antonopoulos 2017, S. 62)

Zwischen dem privaten und öffentlichen Schlüssel gibt es eine mathematische Beziehung. Der öffentliche Schlüssel wird nämlich aus dem privaten Schlüssel mit der Multiplikation eines Punktes auf einer elliptischen Kurve generiert. Aus dieser mathematischen Beziehung können digitale Unterschriften generiert werden. Die digitale Unterschrift kann auf Korrektheit einzig und allein mit dem öffentlichen Schlüssel geprüft werden. Um jedoch eine Unterschrift generieren zu können, benötigt man den privaten Schlüssel. Das bedeutet, dass jeder prüfen kann, ob eine Transaktion valide ist, aber nur derjenige beziehungsweise diejenige mit Zugriff auf den privaten Schlüssel kann Transaktionen tätigen.

Bei einer vollständigen Transaktion muss der Besitzer beziehungsweise die Besitzerin sowohl den öffentlichen Schlüssel als auch die dazugehörige Unterschrift, welche bei jeder Transaktion anders ist, vorzeigen, um Bitcoins ausgeben zu können. Daraufhin kann jeder Knotenpunkt im Bitcoin-Netzwerk für sich selbst entscheiden, ob die Transaktion korrekt ist und somit bestätigen,

dass der Versender beziehungsweise die Versenderin tatsächlich zum Zeitpunkt der Transaktion die Bitcoins besitzt.

Einfach dargestellt ist die Berechnung des öffentlichen Schlüssels durch die Multiplikation elliptischer Kurven unter Verwendung des privaten Schlüssels wie folgt:

$$K = k * G$$

k ist der private Schlüssel, G ist eine Konstante mit dem Namen "Generator Point"und K ist der resultierende öffentliche Schlüssel. In diese Richtung ist die Berechnung relativ simpel, in die andere jedoch fast unmöglich. Um den Wert von k herauszufinden, wenn man K kennt, bleibt nichts anderes übrig, als alle möglichen Werte für k einzusetzen, bis man auf den Wert von K stößt. (vgl. Antonopoulos 2017, S. 65) Demnach nennt man Funktionen wie diese Einwegfunktionen.

4 Transaktionen

Bitcoin-Transaktionen sind Aufzeichnungen über Übertragungen von Bitcoins von einem Benutzer an einen anderen. Jede Bitcoin-Transaktion besteht aus mindestens einer Eingabe- und einer Ausgabeadresse. Die Eingabe-Adresse stellt die Bitcoins bereit, die übertragen werden sollen, und die Ausgabe-Adressen empfangen die übertragenen Bitcoins. Eine Bitcoin-Transaktion kann auch eine Gebühr beinhalten, die an die Miner gezahlt wird, die die Transaktion bestätigen und in die Blockchain aufnehmen. Es ist wichtig zu beachten, dass Bitcoin-Transaktionen irreversibel sind, d.h. sie können nicht rückgängig gemacht werden, nachdem sie einmal bestätigt wurden. Daher sollte man sorgfältig darauf achten, an die richtige Adresse zu senden, bevor man eine Bitcoin-Transaktion durchführt.

4.1 Wallets

Eine Wallet besteht aus Software, die private und öffentliche Schlüssel enthält und die Blockchain nutzt, um Währung zu senden und zu empfangen. Die Währung in diesen Wallets wird in Form von Coins, wie beispielsweise Bitcoin, hinzugefügt. Um Coins oder Währung empfangen oder senden zu können, muss eine Wallet erstellt werden. Die Währung selbst ist nicht an einem Ort in der Wallet gespeichert, sondern befindet sich in der Form von Transaktionsaufzeichnungen in der Blockchain. (vgl. Yadav, Goar und Kuri 2020,

S. 4) Demnach enthält die Wallet lediglich die Informationen (Schlüssel), um Währung auszugeben, nicht aber die Währung selbst.

Es gibt zwei primäre Arten von Wallets, welche sich darin unterscheiden, ob die enthaltenen Schlüssel zusammenhängend sind oder nicht. Die erste Art einer Wallet ist die nichtdeterministische Wallet, bei welcher jeder Schlüssel unabhängig von einer zufälligen Zahl generiert wird. Die Schlüssel in dieser Wallet sind NICHT zusammenhängend. Die zweite Art einer Wallet ist die deterministische Wallet, bei welcher alle Schlüssel Derivate von einem Masterschlüssel, welcher auch als Samen bezeichnet wird. Alle Schlüssel in dieser Art von Wallet sind zusammenhängend und können neu generiert werden, solange man den originalen Samen beziehungsweise den Masterschlüssel hat. Es gibt mehrere Methoden zur Derivation von Schlüsseln, welche in deterministischen Wallets genutzt werden. Die am häufigsten genutzte Derivationsmethode nutzt eine baumähnliche Struktur und wird als hierarchische deterministische (HD) Wallet bezeichnet. (vgl. Antonopoulos 2017, 93f)

4.1.1 Deterministische Wallets

Deterministische Wallets beinhalten private Schlüssel, welche alle von einem gemeinsamen Samen abstammen. Die privaten Schlüssel werden vom Samen durch Einwegfunktionen beziehungsweise Hashfunktionken berechnet. Der Samen ist eine zufällig generierte Zahl, welche mit anderen Daten wie zum Beispiel einer Indexnummer kombiniert wird, um dann mittels einer Hashfunktion einen privaten Schlüssel generiert. Im Gegensatz zu der nichtdeterministischen Wallet benötigt man bei der deterministischen Wallet einzig und allein den Samen, um auf alle Schlüssel zugreifen zu können, da diese alle vom Samen abstammen. Aus diesem Grunde benötigt man nur ein einziges Backup, welches bei der Erstellung der Wallet gespeichert wird. Vorteilhaft ist dabei auch die Mobilität der Wallet: Man muss einzig und allein den Samen auf die neue Wallet-Umgebung transferieren, um wieder Zugriff auf alle Schlüssel zu haben. (vgl. Antonopoulos 2017, S. 95)

Im BIP-32 Standard wurde die häufigste Art der deterministischen Wallets implementiert: die hierarchische deterministische Wallet. HD Wallets enthalten Schlüssel, die in einer Baumstruktur angeordnet sind. Ein Schlüssel kann Töchterschlüssel generieren, von welchen wiederum Enkelschlüssel generiert werden können, und so weiter bis in die Unendlichkeit. Abgesehen von der Mobilität hat die HD Wallet noch zwei weitere Vorteile gegenüber nichtdeterministischen Wallets. Erstens können bestimmte Äste von Schlüsseln genutzt werden, um ihnen organisatorische Bedeutung zu verleihen. Beispielsweise kann ein Ast von Schlüsseln genutzt werden, um Zahlungen zu senden und ein an-

dere Ast von Schlüsseln erhält dann das Wechselgeld. Zweitens können Nutzer von HD Wallets eine Reihe von öffentlichen Schlüsseln generieren, ohne Zugriff auf die zugehörigen privaten Schlüssel zu haben. Dadurch können die Schlüssel auf unsicheren Servern genutzt. (vgl. Antonopoulos 2017, S. 96)

4.1.2 Nichtdeterministische Wallets

Anfangs bestanden Bitcoin Wallets aus einer Sammlung von zufällig generierten privaten Schlüsseln. Der erste Bitcoin Client hatte 100 Schlüssel vorprogrammiert und generierte bei Bedarf mehr, wobei jeder Schlüssel nur einmal genutzt wird. Dies mag für die Sicherheit vor Hackern gut klingen, jedoch ist es wegen der fehlenden Verwandtschaft der Schlüssel sehr schwierig zu verwalten. Keine der Schlüssel haben eine Verbindungen miteinander und es muss jeweils für jeden Schlüssel ein Backup erstellt werden. Desto mehr Schlüssel generiert werden, desto aufwändiger ist es, alle Schlüssel zu verwalten und ein Backup für diese zu erstellen.

4.2 Ausgaben und Eingaben

Die Blockchain ist ein global vereinbartes Register aller jemals durchgeführten Transaktionen und wird ständig als eine einzige verknüpfte Liste erweitert. Jede Transaktion in der Blockchain, zum Beispiel Alice, die Bob 10 Bitcoins zahlt, hat einen oder mehrere Transaktionausgänge (TXO), die als Summen aus verwendbarer Bitcoins dienen. Hierbei muss beachtet werden, dass Bitcoin Summen nicht in kleinere Summen aufgeteilt werden können, d. h., dass es im Normalfall einen Überschuss an Bitcoins gibt, der an den Sender zurückgezahlt wird, ähnlich wie beim Wechselgeld im Falle normaler Währung. Sobald Bob die TXO beziehungsweise die Summen seiner verwendbaren Bitcoins erhält, bezeichnet man sie als unausgegebene Transaktionausgänge (UTXO). Sie bleiben UTXOs, bis der Besitzer (in diesem Beispiel Bob) sie einlöst, um jemand anderen zu bezahlen (in diesem Fall werden sie wieder als ausgegebene TXOs bezeichnet). (vgl. Konrad und Pinto 2015, S. 1) Vor jeder Transaktion befindet sich eine 4-byte lange Transaktionsversionnummer, welche anderen Peers und Minern sagt, mit welchen Regeln diese Transaktion bestätigt werden soll. UTXOs haben eine Indexnummer basierend auf ihrer Lage in der Transaktion. Wenn ein beliebiger Bitcoin (oder nur ein Anteil) zum ersten Mal versendet wird, nachdem dieser gemined wurde, hat die zugehörige Transaktion eine Indexnummer von 0. Im Gegensatz zu Ausgaben (UTXOs) haben Eingaben (TXOs) eine Ausgabenindexnummer, um den UTXO zu spezifisieren, der ausgegeben werden soll. Zudem hat es einen Skript mit bestimmten Parametern,

welcher erfüllt werden muss, um die Bitcoins ausgeben zu können.

Meist enthält eine Transaktion auch eine Transaktionsgebühr, welche an Miner als Ausgleich zur Versicherung des Bitcoin-Netzwerkes ausgezahlt wird. Zudem schützen Transaktionsgebühren auch vor Hackerangriffen, indem sie es finanziell unmöglich machen, das Bitcoin-Netzwerk mit Transaktionen zu überschütten. Zudem können Miner entscheiden, welche Transaktion sie in ihrem Block inkludieren möchten, d.h. eine Transaktion mit einer höheren Transaktionsgebühr wird eher inkludiert als eine Transaktion mit niedrigen Transaktionsgebühren. Berechnet werden Transaktionsgebühren anhand der Größe einer Transaktion in Kilobyte und nicht anhand der Menge von Bitcoins, die versendet werden. Im Laufe der Zeit änderte sich die Berechnung von Transaktionsgebühren und die Bedeutung von Transaktionsgebühren bei der Prozessierungspriorität. Zuerst hatten Transaktionsgebühren einen festen Wert und waren konstant. Die Gebührenregeln lockerten sich jedoch immer mehr und erlaubten es, dass Marktkräfte den Gebührenpreis durch Kapazität des Bitcoin-Netzwerkes und Transaktionsvolumen bestimmten. 2016 explodierte Bitcoin an Popularität, wodurch das Kapazitätslimit erreicht wurde und die Transaktionsgebührenpreise in die Höhe schossen; Transaktionen ohne Transaktionsgebühren sind dadurch Geschichte gewesen. (vgl. Antonopoulos 2017, S. 127)

4.3 Transaktionenskripts

Bitcoin nutzt eine eigene Stapel-basierte Skriptsprache mit einer Reihe von Opcodes. Diese Skriptsprache, welche auch ganz einfach "Script" genannt wird, wird genutzt, um festzustellen, ob ein unausgegebener Transaktionsausgang (UTXO) ausgegeben werden kann oder nicht. Wenn Bitcoin versendet wird, dann wird lediglich ein Bitcoinwert zu einem Skript zugewiesen, welcher nur ausgegeben kann, wenn der Skript erfolgreich ausgeführt wird und ein einziger "true" Wert auf dem Stapel entsteht. Der Skript, welcher dem UTXO zugewiesen ist, nennt man "scriptPubKey". Um die Bedingungen des Skriptes zu erfüllen, muss man eine korrekte "scriptSig" zur Verfügung stellen, welche auch ein Skript ist und ausgeführt wird, um zu testen, ob der Versand von Bitcoin funktioniert hat. (vgl. Baczuk 2021, 1ff) Skript ist eine sehr einfach Sprache, welche dazu gemacht wurde, nicht viele Möglichkeiten zu bieten und auf fast jedem Gerät ausführbar ist, wie zum Beispiel einem eingebetteten System. Die Sprache benötigt minimale Prozessorkraft und kann nicht viele der schicken Dinge tun, die moderne Programmiersprachen können. Zweck ist, dass eine Sprache, die nicht viele Möglichkeiten bietet, sich einfach überprüfen lässt und sehr robust gegen Bugs o.Ä. ist, was sie zu einem perfekten Kandidaten für

die Versendung von Währung in Bezug auf Sicherheit macht.

Bitcoins Skriptsprache wird Stapel-basiert genannt, weil sie eine Datenstruktur namens "Stapel" nutzt. Ein Stapel ist ein sehr einfaches Datenkonstrukt, welches man sich wie einen Stapel von Karten vorstellen kann. Ein Stapel hat zwei mögliche Operationen: Push (drauflegen) und Pop (herunternehmen). Push legt ein Objekt an die Spitze des Stapels. Pop entfernt ein Objekt von der Spitze des Stapels. Operationen auf einem Stapel können nur mit dem obersten Objekt des Stapels interagieren. Deshalb nennt man dieses Konzept auch Last-In-First-Out-Prinzip (LIFO). Die Skriptsprache führt den Skript aus, indem sie jedes Objekt von link nach rechts ausliest. Dabei werden Zahlen (Datenkonstanten) auf den Stapel hinaufgelegt. Operatoren können mehrere Elemente auf einmal von dem Stapel herunternehmen und drauflegen, mit diesen interagieren und möglicherweise ein daraus resultierendes Objekt zurück auf den Stapel legen. Beispielsweise kann OP_ADD zwei Objekte von dem Stapel herunternehmen, diese addieren, und das resultierende Ergebnis wieder auf den Stapel legen. Bedingte Operatoren prüfen eine Bedingung und geben Boolean Wert von TRUE oder FALSE zurück. Beispielsweise kann OP_EQUAL zwei Objekte vom Stapel herunternehmen und legt TRUE (wird in Script als 1 dargestellt) zurück hinauf, wenn die Elemente denselben Wert haben und FALSE (wird in Script als 0 dargestellt), wenn die Elemente nicht denselben Wert haben. Im Normalfall haben Bitcoin-Transaktions-Skripts einen bedingten Operator, um die Gültigkeit der Transaktion als gültig beziehungsweise TRUE oder ungültig beziehungsweise FALSE zu bestimmen. (vgl. Antonopoulos 2017, S. 133)

Im ursprünglichen Bitcoin Client (heute Bitcoin Core) waren der "scriptPubKey" beziehungsweise der Ausgabeskript und der "scriptSig" beziehungsweise der Eingabeskript miteinander verkettet und in einer Sequenz ausgeführt worden. Aus Sicherheitsgründen wurde dies 2010 geändert, da es eine Verwundbarkeit gab, wodurch missgebildete Ausgabeskripts in der Lage waren, Daten auf den Stapel des Eingabeskripts zu legen und diesen dadurch zu korrumpieren. Heutzutage werden die Skripte deswegen separat ausgeführt und die Stapel werden zwischen den Ausführungen transferiert, wie folgend beschrieben: Zuerst wird der Ausgabeskript mit dem Script Engine ausgeführt. Falls der Ausgabeskript ohne Fehler (d. h. keine verbleibenden Operatorn) ausgeführt wird, wird der Hauptstapel kopiert und der Eingabeskript wird ausgeführt. Wenn das Ergebnis der Ausführung des Eingabeskriptes mit den Stapeldaten des kopierten Ausgabeskriptes übereinstimmt (d. h. das Ergebnis ist TRUE oder 1), hat der Ausgabeskript es geschafft, die Bedingungen des Eingabeskriptes zu erfüllen und ist daher erlaubt, die enthaltenen UTXOs auszugeben. Resultiert irgendein anderes Ergebnis außer TRUE beziehungsweise 1, nachdem der kombinierte Skript ausgeführt wurde, ist die Eingabe ungül-

tig, weil er es nicht geschafft hat, die Bedingungen zu erfüllen, welche auf dem UTXO gesetzt sind.

5 Probleme

5.1 Skalierbarkeit

Da Bitcoin ein selbstregulierendes System ist, welches durch das "Mining" von Blöcken alle 10 Minuten funktioniert, ist die höchste Anzahl von Transaktionen prinzipiell die maximale Größe eines Blocks dividiert durch das Mining-Intervall. Bei einem Intervall von 10 Minuten, um Transaktionen zu bestätigen, schafft es Bitcoin, auf einem Transaktionsdurchsatz von 7 Transaktionen pro Sekunde zu kommen. Um dies in Vergleich zu stellen, bei einer herkömmlichen Zahlungsmethode wie zum Beispiel VISA werden Transaktion innerhalb von Sekunden bestätigt. Zudem liegt der Transaktionsdurchsatz von VISA bei über 2000 Transaktionen pro Sekunde. Offensichtlicherweise existiert eine riesige Lücke zwischen dem Stand, auf welchem Bitcoin sich befinden und herkömmlichen Zahlungsmethoden. (vgl. Croman u. a. 2016, S. 1) Des Weiteren ist die Zeit, die eine neue Full Node benötigt, um die gesamte Blockchain herunterzuladen, enorm. Die derzeitige Größe der Bitcoin-Blockchain beträgt 425 Gigabyte, während die durchschnittliche Internetgeschwindigkeit bei etwa 30Mbps liegt. Folgend ist die benötigte Zeit $\frac{425000}{3.75} = 113333$ Sekunden beziehungsweise 31.5 Stunden. Für die meisten Nutzer ist es daher undenkbar, eine Full Node zu betreiben, obwohl Bitcoin nur dadurch sicher bleiben kann.

Um solche Probleme zu lösen, müssen sogenannte "Forks" das System von Grund auf ändern.

Eine Art von Fork wird als Hard Fork bezeichnet, da das Netzwerk nach dem Fork nicht wieder auf eine einzige Blockchain konvergiert. Stattdessen entwickeln sich die beiden Blockchains unabhängig voneinander weiter. Hard Forks treten auf, wenn ein Teil des Netzwerkes unter einem anderen Satz von Konsensregeln arbeitet als der Rest des Netzwerkes. Dies kann aufgrund eines Fehlers oder aufgrund einer bewussten Änderung in der Implementierung der Konsensregeln auftreten. Hard Forks können verwendet werden, um die Konsensregeln zu ändern, erfordern jedoch eine Koordination zwischen allen Teilnehmern im System. Alle Nodes, die nicht auf die neuen Konsensregeln aktualisieren, können nicht am Konsensmechanismus teilnehmen und werden zum Zeitpunkt des Hard Forks auf eine separate Kette gezwungen. Eine Änderung, die durch einen Hard Fork eingeführt wird, kann also als nicht "vorwärtskompatibel" betrachtet werden, da nicht aktualisierte Systeme die neuen

Konsensregeln nicht mehr verarbeiten können. (vgl. Antonopoulos 2017, S. 257)

Nicht alle Änderungen der Konsensregeln führen zu einem Hard Fork. Nur Konsensänderung, die nicht vorwärtskompatibel sind, verursachen einen Fork. Wenn die Änderung so implementiert wird, dass ein unveränderter Client die Transaktion oder den Block weiterhin als gültig gemäß den vorherigen Regeln betrachtet, kann die Änderung ohne Fork erfolgen. Der Begriff "Soft Fork" wurde eingeführt, um diese Upgrade-Methode von einem "Hard Fork" zu unterscheiden. In der Praxis ist ein Soft Fork überhaupt kein Fork. Ein Soft Fork ist eine vorwärtskompatible Änderung der Konsensregeln, die es nicht aktualisieren Clients weiterhin ermöglicht, im Konsens mit den neuen Regeln zu arbeiten. Soft Forks können auf verschiedene Arten implementiert werden - der Begriff definiert keine spezifische Methode, sondern eine Reihe von Methoden, die alle eines gemeinsam haben: Sie erfordern nicht, dass alle Nodes aktualisiert werden oder zwingen nicht aktualisierte Nodes aus dem Konsens. (vgl. Antonopoulos 2017, S. 261)

5.2 Verluste

Kritiker betonen oft die fehlende physikalische Form von Bitcoin und das zurecht. Bitcoins können nur in der digitalen Welt existieren und man ist dadurch auf Wallets angewiesen. Wenn man in irgendeiner Art und Weise Zugriff auf diese Wallet verliert, sind die darauf enthaltenen Schlüssel (und im Zuge dessen auch die Bitcoins) für immer verloren. Das kann bei digitalen Wallets sehr schnell gehen, wenn beispielsweise die Festplatte, auf welcher die Wallet betrieben wird, kaputt geht. Oder wenn man bei einem Online-Anbieter sein Passwort vergisst. Anders als bei Banken, bei welchen man trotz verlorener Kreditkarte Zugriff auf sein Konto bekommen kann, gibt es bei Bitcoin keine zentrale Einheit, die solche Problematiken löst.

Bitcoin ist derzeit so konzipiert, dass es in Einheiten namens Satoshis unterteilbar ist, die 0.00000001 eines Bitcoins entsprechen. Die Tatsache, dass Bitcoins so teilbar sind, soll bedeuten, dass aufgrund der asymptotischen Grenze von 21 Millionen Bitcoins * 100.000.000 Satoshis pro Bitcoin insgesamt 2.100.000.000.000.000 (2,1 Billiarden) Satoshis möglich sind. Dadurch kann Bitcoin eine lebensfähige globale Währung sein, die praktisch jede Art von Expansion im Maßstab von Nationen unterstützen kann. (vgl. Hanley 2013, S. 7) Hier tritt das Problem der Verluste in einer neuen Form auf: Wenn Wallets verloren gehen, dann sind auch die darauf enthaltenen Bitcoins für immer aus dem Bitcoin-Netzwerk verloren. Die einzige Methode, mit welcher man die Bitcoins zurückholen könnte, ist, alle Private-Schlüssel-Kombinationen auszuprobieren, bis man auf eine Wallet stoßt, auf welcher Bitcoins enthalten sind.

Diese Methode ist jedoch extrem rechenintensiv und praktisch unmöglich. Da im Laufe der Zeit immer mehr Wallets verloren gehen, wird die Anzahl der Bitcoins auch immer geringer.

6 Fazit

Nach dem Schreiben dieser Arbeit ergibt sich die Erkenntnis, dass Bitcoin ein unglaublich vielseitiges Thema ist, wenn man über den rein finanziellen Aspekt hinwegblickt. Abgesehen von den etlichen Nutzern der Währung gibt es auch wahnsinnig viele, aber notwendige Kleinigkeiten, welche Bitcoin überhaupt erst ermöglichen. Diese Arbeit kratzt jedoch nur an der Oberfläche, besonders im Bereich der Kryptographie. Die Mathematik hinter Bitcoin würde nämlich eine eigene wissenschaftliche Arbeit an sich benötigen, wenn nicht sogar mehrere.

Des Weiteren hat sich ergeben, dass das Teilhaben an Bitcoin gar nicht so kompliziert sein muss. Die zu Hause aufgestellte Full Node lies sich überraschend einfach bedienen. Wenn man Bitcoin nur zum Kauf oder Verkauf nutzen möchte, ist die Sache nochmal um einiges einfacher. Mehr als eine Wallet und etwas Guthaben benötigt man hierfür nicht.

Arbeitstechnisch habe ich Fehler gemacht, die mir sehr viel Zeit kosteten. Bevor ich überhaupt anfing zu schreiben, habe ich versucht, mir die ganze Literatur einzuprägen. Das hat logischerweise nicht gut funktioniert und ich musste mir die Literatur beim Schreiben nochmals durchlesen. Zudem wollte ich beim Schreiben stetig mehr Themen behandeln, welche ich ursprünglich gar nicht vorgeplant hatte. Bitcoin ist möglicherweise auch einfach ein etwas zu breites Thema für eine VWA. Zukünftig werde ich versuchen, mein Thema gezielter einzugrenzen und bewusster zu schreiben.

Literaturverzeichnis

Antonopoulos, Andreas M. (2017). *Mastering bitcoin: Programming The open blockchain*. O'Reilly Media, Inc.

Baczuk, Jordan (Dez. 2021). *Bitcoin Script Tutorial - Jordan Baczuk*. URL: https://medium.com/jordan.baczuk/bitcoin-script-tutorial-34752ec0bbb.

Croman, Kyle u. a. (Feb. 2016). „On Scaling Decentralized Blockchains (A Position Paper)". In.

Hanley, Brian (Dez. 2013). „The False Premises and Promises of Bitcoin". In.

Jönck, Uwe und Florian Prill (Feb. 2003). *Das Lorenz-System*. URL: https://www.math.uni-hamburg.de/home/lauterbach/scripts/seminar03/prill.pdf.

Konrad, Robert und Stephen Pinto (Dez. 2015). *Bitcoin UTXO lifespan prediction*. URL: http://cs229.stanford.edu/proj2015/225_report.pdf.

Nakamoto, Satoshi (März 2009). „Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com*.

Stipčević, Mario und Çetin Koç (Nov. 2014). „True Random Number Generators". In: S. 275–315. ISBN: 978-3-319-10682-3. DOI: 10.1007/978-3-319-10683-0_12.

Taaki, Amir (2011). *Bitcoin/bips: Bitcoin improvement proposals*. URL: https://github.com/bitcoin/bips/.

Yadav, Nagendra Singh, Vishal Goar und Manoj Kuri (Feb. 2020). „Crypto Wallet: A Perfect Combination with Blockchain and Security Solution for Banking". In: *International Journal of Psychosocial Rehabilitation* 24, S. 6056–6066. DOI: 10.37200/IJPR/V24I2/PR2021078.