Abstract

Satoshi Nakamoto beschreibt Bitcoin als ein Peer-to-Peer Zahlungssystem **?**. Dabei wurde eine Vielzahl verschiedener, möglicher Einsatzzwecke der zugrundeliegenden Blockchain-Technologie vernachlässigt. In dieser Arbeit wird gezeigt, dass ein dezentrales Zahlungsmittel nur ein möglicher Einsatzzweck für die eigentliche Innovation der dezentralen, Vertrauenslosen Konsens-Erzielung ist. Während klassischerweise für die Übertragung von Werten zwischen zwei Parteien ein vertrauenswürdiger Mittelsmann benötigt wurde - beispielsweise eine Bank zur Kontenführung oder eine Abwicklungsgesellschaft zur Abwicklung von Wertpapiergeschäften - kann eine solche Instanz durch Blockchain-Technologie eliminiert werden. Die Vorteile eines dezentrales Konsens bestehen aus einer anonymen, transparenten, nachvollziehbaren, effizienteren und kostengünstigeren Abwicklung von Wertübertragungen.
Anwendungen, die auf Basis der Blockchain-Technologie basieren, werden als smarte Verträge bezeichnet, mit der Definition einer Crypto-Währung als einer der grundlegenden Verträge. In dieser Arbeit wird gezeigt, dass smarte Verträge einen disruptiven Charakter haben, der neben klassischen Anwendungen zur Wertübertragung ohne Bedarf eines Mittelmanns, völlig neue Anwendungszwecke bis hin zu innovativen Unternehmensorganisationen erlauben.
Dazu wird einerseits die Blockchain Technologie sowie ihre historische Entwicklung bis hin zu einer turing-vollständigen Technologie erläutert und auf technische Herausforderungen geprüft. Andererseits wird eine Auswahl möglicher Anwendungsszenarien vorgestellt, gewertet und auf ihre fachliche Umsetzbarkeit geprüft.
Die Ergebnisse zeigen, ...

Inhaltsverzeichnis

wohl nicht mehr aktuell

Abkürzungsverzeichnis

UTXO	Unspent Transaction Outputs
BTC	Bitcoin
DoS	Denial of Service
SPV	Simplified Payment Verification
BGB	Bundesgesetzbuch

Glossar

Bitcoin Überbegriff für die Technologie sowie das Netzwerk, die hinter der Crypto-Währung bitcoin steht. 1

Blockchain Das technologische Grundkonzept von Cryptowährungen, das auf dem Prinzip basiert, alle Transaktionen miteinander zu verketten. 1, III

1. Einleitung

1.1. Relevanz der Blockchain-Technologie

Seitdem 2008 von dem nur unter seinem Psudonym Satoshi Nakamoto bekannten Autor erstmals die Bitcoin-Architektur beschrieben worden ist, wurde in der Presse vornehmlich der bitcoin als Alternativwährung zu etablierten Währungen diskutiert. Mit einer Marktkapitalisierung von knapp 5,7 Mrd USD (Stand 01.02.2016) (Statista) erfüllt der bitcoin vornehmlich eine Funktion als alternative Anlagemöglichkeit zu etablierten Geldanlagen wie Devisen oder Gold. Verschiedene Betrachtungen dieser Crypto-Währungen zeigen , dass zu einer hohen Wahrscheinlichkeit ein großer Teil der Marktkapitalisierung durch Anleger, die den bitcoin als Wertanlage statt als Zahlungsmittel interpretieren, entstanden ist.

Zitat

Zitat

Zumindest für die zugrundeliegende Technologie könnte sich dies jedoch bald ändern - auch eine Vielzahl anderer Einsatzmöglichkeiten ist möglich . In der Fachpresse wird mittlerweile die zugrundeliegende Technologie der Blockchain - oder Distributed Ledger - als eigentlich disruptive Innovation bezeichnet. Kern der Blockchain ist, dass - im Gegensatz zu bisherigen Lösungen - das Hauptbuch nicht in einer zentralen Datenbank, wie z.B. bei einer Bank gespeichert wird, sondern über alle Nutzer verteilt ist. Dadurch wird eine Überweisung von Werten zwischen zwei Parteien nicht von einer zentralen Instanz kontrolliert, sondern von allen Nutzern des Systems. Die wirtschaftlichen Vorteile liegen hierbei auf der Hand - eine zentrale Stelle agiert immer als Dienstleister und verlangt für die Dienstleistung als vertrauenswürdiger Mittelsmann entsprechende Gebühren. Weiterhin kontrolliert sie das System komplett, d.h. bei einem Angriff agiert das System als Single-Point-of-failure. Zusätzlich kann es sich als nicht-vertrauenswürdig herausstellen und selbst die Werte aller Nutzer klauen. Bei verteilten Hauptbüchern entfällt der Bedarf nach einer vertrauenswürdigen dritten Partei, was oben beschriebene Nachteile zumindest reduziert.

Unbedingt Einleitung ändern

Die Blockchain-Technologie eignet sich hierbei jedoch nicht nur als Überweisungssystem für Crypto-Währungen. Über sogenannte Smart Contracts, kann eine digitale Münze eine Stellvertreterfunktion für eine Vielzahl verschiedener Einsatzzwecke haben. So kann sie z.B. für ein Wertpapier, eine Domain, eine Identität oder einen reellen Gegenstand stehen. Hierbei wurden bestehende Blockchains - wie z.B. die Bitcoin-Blockchain

um verschiedene Funktionalitäten erweitert, als auch komplett neue Entwickelt.

Durch die Flexibilität der Einsatzmöglichkeiten kann die Blockchain-Technologie theoretisch überall dort eingesetzt werden, wo bisher ein Vertrag definiert wird, bzw. eine vertrauenswürdige dritte Partei benötigt wird um möglichen Betrug bei einer Transaktion zu verhindern, bzw. als Exekutivkraft eine Umsetzung des Vertrags erzwingt.

1.2. Zielsetzung

Die Blockchain, die Technologie auf der beispielsweise die Krypto-Währung Bitcoin basiert – verspricht in seiner Urform, durch seine Funktion als dezentrales Hauptbuch – eine sichere, vollautomatisierte Möglichkeit, Werte zwischen verschiedenen Parteien zu übertragen.

Seitdem die Technologie das erste Mal beschrieben wurde, haben einige Entwicklungen stattgefunden, die eine Vielzahl weiterer Anwendungszwecke erlauben. Seit einiger Zeit wird der Begriff Blockchain 2.0 für eine teilweise, bzw. vollständig programmierbare Blockchain verwendet. Hierfür wurden entweder bestehende Blockchains (z.B. Bitcoin) erweitert, oder wie im Falle von Ethereum, komplett neue entwickelt.
In dieser Arbeit wird untersucht, welche Möglichkeiten die Blockchain-Technologie bietet, aber auch welche Herausforderungen mit ihr verbunden sind. Dazu wird die grundsätzliche Blockchain-Technologie, nach ihrem unter dem Pseudonym Satoshi Nakamoto bekannten Erfinder, beschrieben, sowie die konzeptuellen Änderungen, die eine Turing-vollständige Blockchain – als Einzelentwicklung oder durch Verwendung von Sidechains – ermöglichen, erläutert.

Auf Basis dieser Erkenntnisse werden technische Fragestellungen, wie beispielsweise Skalierbarkeit, Kompatibilität zwischen Blockchains, Möglichkeiten zur Anbindung verschiedener Schnittstellen, Sicherheit – insbesondere die Gefahr einer Systemübernahme durch Mining Pools, sowie Möglichkeiten zur Erkennung und zum Auflösen von Manipulationen – diskutiert.
Ferner werden die wichtigsten, möglichen Anwendungsbereiche smarter Verträge beschrieben. Deren derzeitige Umsetzbarkeit soll durch eigene Verträge auf Basis einer Turing-vollständigen Blockchain überprüft und bewertet werden.

Wichtig sind hier selbstverständlich die bereits beschriebenen technische Fragestellungen, sowie andere nicht-technische: Beispielsweise Schutzmechanismen gegen eine hohe Volatilität der zugrundeliegenden Crypto-Währung, der mögliche Umgang mit fehlerhaften oder betrügerischen Überweisungen, oder die Rolle eines programmierten Vertrags gegenüber eines juristischen.

Der Mehrwert der Arbeit stellt eine solide Übersicht über die Praxistauglichkeit einer Technologie dar, die 2008 erstmals beschrieben wurde und deren erste Implementierung als Turing-vollständige Blockchain – die nicht nur das Bitcoin-Protokoll um einen bestimmten Anwendungszweck erweitert – im Juli 2015 veröffentlicht wurde.

Leser sollen einen Einblick in die Architektur der Blockchain-Technologie bekommen und das Konzept smarter Verträge und seiner Anwendungsbereiche verstehen. Weiter sollen sie bewerten können, in welchen Situationen der Einsatz der Blockchain-Technologie vorteilhaft gegenüber traditionellen Lösungen ist.

Zusätzlich sollen sie beurteilen können, welche Möglichkeiten die Technologie derzeit bietet, welche Nachteile sie mit sich bringt und in welche Richtung sich die weitere Entwicklung bewegen wird.

1.3. Aufbau der Arbeit

2. Aufbau und Architektur von Blockchains

2.1. Begriffsdefinition

Distributed ledger Technologie: Allgemeine Technologien, die ein verteiltes Hauptbuch beschreiben (Kerntechnologie der Blockchain) *Vgl. Geiling (2016)* Gelegentlich kommt in der Literatur der Begriff Distributed ledger vor. Dieser eignet sich sehr gut um das Kernprinzip von Crypto-Währungen zu beschreiben: Die dezentrale Speicherung sämtlicher im Netzwerk stattgefundener Transaktionen bei allen Teilnehmern im Netzwerk. Trotzdem ist der Begriff irreführend, da auf einer Blockchain kein klassisches Hauptbuch gespeichert wird, sondern lediglich eine Transaktionshistorie, aus der ein Hauptbuch mittels spezialisierter Software (i.d.R. „Wallet"-Programme) gebildet werden kann. Weiterhin gibt es Konzepte wie MaidSafe, die keine wie in Bitcoin verwendete Blockchain benutzen[1]. In dieser Arbeit werden nur Ansätze für smarte Verträge auf der Basis einer Blockchain betrachtet, d.h. ein Header, unter welchem alle Transaktionen gespeichert werden.

2.2. Crypto-Währungen als digitales Zahlungsmittel

entscheiden, ob notwendig

2.3. Smarte Verträge

2.3.1. Begriffsdefinition

Smarte Verträge werden derzeit sehr unterschiedlich definiert. Dies liegt wohl zum Großteil an einem unterschiedlichen Verständnis von Vertragsrecht über verschiedene Rechtssysteme hinweg - insbesondere zwischen dem in vielen Teilen der Welt praktizierten „Common Law" gegenüber dem in Kontinentaleuropa ausgeübten „Civil Law", bzw. Zivilrecht. Doch auch verschiedene wissenschaftliche Disziplinen haben

[1] http://blog.maidsafe.net/2016/01/10/evolving-terminology-pt-2/

verschiedene Betrachtungsweisen - insbesondere Informatiker und Juristen haben hier unterschiedliche Einordnungen. Zusätzlich wirkt der Begriff smarter Vertrag irreführend. Er suggeriert, dass es sich bei einem smarten Vertrag um einen rechtlichen Vertrag handelt, der zusätzlich noch „smart“ ist. Diese Arbeit zeigt, dass die Definition eines smarten Vertrags für sich selbst steht und dabei nicht der Definition eines rechtlichen Vertrags entsprechen muss.
Es wird sich an dem deutschen Vertragsbegriff orientiert, welcher insbesondere den Grundsatz der Privatautonomie beinhaltet. Eine allgemeingültige, globale Abgrenzung smarter Verträge zu rechtlichen Verträgen ist nicht Kern dieser Arbeit und wird an dieser Stelle nicht gesucht. Stattdessen wird eine pragmatische Definition im Kontext zu deutschem Vertragsrecht aufgezeigt, die es erlaubt, die in Kapitel 3 aufgeführten Anwendungsmöglichkeiten smarter Verträge zu anderen Anwendungsmöglichkeiten, die nicht dieser Definition entsprechen, abzugrenzen.

Als Wortschöpfer gilt gemeinhin der Jurist und Informatiker Nick Szabo, welcher einen smarten Vertrag als *„eine Menge an Versprechungen, spezifiziert in digitaler Form, Protokolle beinhaltend, innerhalb welcher die Vertragsparteien ihre Versprechungen ausführen“*, bezeichnet. *Szabo (1996)*

Nees fragen ob Zitat übersetzt werden muss oder nicht

Interessant ist hier, dass Szabo einen smarten Vertrag so definiert, dass er komplett innerhalb der spezifizierten Protokolle ausgeführt wird. Er ist also selbst-erfüllend. Damit grenzt er sich deutlich von den gängigen Vertragsdefinitionen ab, deren Erfüllung durch ein zugrundeliegendes Rechtssystem erzwungen werden kann. Diese Definitionen werden im folgenden betrachtet.

Überarbeiten

Nach dem Bundesgesetzbuch (BGB) definiert sich ein Vertrag wie folgt:
„Der Vertrag ist ein Rechtsgeschäft. Es besteht aus inhaltlich übereinstimmenden, mit Bezug aufeinander abgegebenen Willenserklärungen (Angebot und Annahme) von mindestens zwei Personen.
Durch den Grundsatz der Vertragsfreiheit (Privatautonomie) wird sichergestellt, dass jeder Mensch das Recht hat, im Rahmen der Gesetze seine Verhältnisse durch Verträge eigenverantwortlich zu gestalten.“

Zitat

Einen rechtlichen Vertrag definiert Szabo als: *„eine Menge an Versprechungen welche in einer „Übereinstimmung gegenseitiger Vorstellungen“ getroffen werden.“*

Ein großer Unterschied beider Definitionen besteht sowohl in der Vertragsausgestaltung als auch in der Definition der Vertragspartner. Laut BGB ist ein Vertrag ein Rechtsgeschäft. Dies setzt unter anderem voraus, dass beide Vertragspartner geschäftsfähig sind.[2] Geschäftsfähig können entweder juristische oder natürliche Personen sein. [3] Ein Vertrag

Quelle

[2] https://de.wikipedia.org/wiki/Rechtsgesch%C3%A4ft

zwischen Entitäten, die nicht dieser Definition entsprechen, kann somit aus rechtlicher Sicht nicht Zustandekommen.

Bezüglich der Vertragsgestaltung setzt das BGB die Einschränkung, dass ein Vertrag nur im Rahmen der Gesetze gestaltet werden kann. Im Rahmen des BGB gilt entsprechend deutsches Recht. Dies umfasst beispielsweise die Essentialia negotii[4], die minimalen Inhalte, welche ein Vertrag benötigt um rechtlich bindend zu sein, oder vom Gesetzgeber geforderte Vertragsformen. Eine exakte Aufstellung wird an dieser Stelle nicht vorgenommen.

Durch den nicht vorhandenen Rechtsraum in welchem smarte Verträge auf der Blockchain ausgeführt werden,[5] kann auch kein lokales Recht aufgeführt werden. Die Vertragsdefinition soll laut Szabo in digitaler Form, d.h. in maschinenlesbarem Code vorgenommen werden. Auch die Vertragsparteien unterliegen nach der Definition Szabos keiner Einschränkung.
Als Besonderheit smarter Verträge auf der Blockchain gilt, dass die beiden Parteien sich durch die Pseudonymisierung ihrer Accounts nicht kennen, wodurch der smarte Vertrag mit dem auf der Blockchain installiertem Computerprogramm getroffen wird, welcher selbst den Vertrag darstellt. Als Beispiel dient das in Abschnitt beschriebene Forward-Rate-Agreement.

Zusammenfassend wird für smarte Verträge auf der Blockchain folgende Definition getroffen:
„Smarte Verträge sind auf der Blockchain installierte Programme, welche von den Vertragspartnern, oder anderen smarten Verträgen aufgerufen werden. Die Vertragsbedingungen bestehen aus in maschinenlesbarem Code definierten wenn-dann Bedingungen. Ihre Erfüllung wird ex-ante durch den Code erzwungen.“

2.3.2. Einführung

Der Begriff der smarten Verträge wurde bereits 1995 von Nick Szabo definiert und damit einige Zeit vor der ersten Beschreibung der Blockchain-Technologie. Sein Ziel ist es, verschiedene Aspekte von Verträgen so in Hardware und Software einzubinden, dass ein Vertragsbruch entweder teuer oder unmöglich für den Vertragsbrecher wird.
Als Beispiel für einen „Urahnen“ eines smarten Vertrags führt er den Verkaufsautomaten auf. Juristisch gesehen wird mit den angegeben Produktpreisen ein Angebot abgegeben,

[3] http://www.steuerazubi.com/geschaeftsfaehig
[4] https://en.wikipedia.org/wiki/Essentialia_negotii
[5] Durch pseudonyme Vertragspartner sowie die globale Verteilung der Nodes ist eine Zurückverfolgung der Vertragsparteien auf einen spezifischen Rechtsraum nicht möglich.

quelle

Abschnitt angeben

macht vermutlich am Ende mehr Sinn

Auch ein Vertragsbruch im Sinne von: Ich muss aus dem Vertrag aussteigen, deshalb bezahle ich vorab eine Kaution, die ich dafür bezahlen muss, wäre eine erzwungene Vertragserfüllung.

dem ein Käufer durch Einwerfen einer passenden Summe zustimmt. Der Verkaufsautomat kommt darauf durch Ausgabe des Produkts seiner juristischen Bringschuld nach. Für den Käufer ist ein Vertragsbruch prinzipiell nicht möglich, da er verpflichtet ist, den passenden Geldbetrag zu zahlen, bevor das Produkt ausgegeben wird. Für den Automaten ist er unmöglich, da aufgrund des einprogrammierten Vertrags das passende Produkt nach Bezahlung ausgegeben wird. *Szabo (1997)*

Verträge müssen nach Szabo folgende Design-Prinzipien erfüllen:

Beobachtbarkeit	Es muss für beide Parteien der Fortschritt der Vertragserfüllung der anderen Partei beobachtbar sein, oder es muss Möglich sein, den Fortschritt gegenüber anderen Klienten zu beweisen.
Überprüfbarkeit	Es muss für einen Klienten möglich sein, einem Schlichter gegenüber zu beweisen, ob ein Vertrag gebrochen oder eingehalten wurde.
vertrauliches Mitwissen	Wissen und Kontrolle über die Vertragsinhalte und die Vertragserfüllung sollten an die verschiedenen Parteien nur so weit verteilt werden, wie es für die Vertragserfüllung notwendig ist.
Durchsetzbarkeit	Durchsetzbarkeit beschreibt sowohl die Möglichkeit, dass ein smarter Vertrag so aufgebaut ist, dass er seine Einhaltung selbst erzwingt, als auch dass durch die Überprüfbarkeit eine dritte Partei eine Ausführung erzwingen kann.

Szabo (1996)
Ähnliche Design-Prinzipien sollen auch für die Definition smarter Verträge gelten.

2.3.3. Anwendung auf der Blockchain

Typischerweise erfüllt ein smarter Vertrag auf der Blockchain folgende Funktionen:

1. Es wird der smarte Vertrag als Code definiert.
2. Dieser Code wird durch Implementierung in ein verteiltes Netzwerk in Produktion gesetzt.
3. Partei eins steigt in den Vertrag ein, indem sie einen gewissen Token zu einer von dem Code kontrollierten Adresse sendet.
4. Zusätzliche Parteien steigen in den Vertrag mit dem selben Mechanismus ein.
5. Sobald ein Ereignis eintritt, welches den Code auslöst, werden verschiedene Funktionen ausgeführt, die voraussichtlich die Tokens neu zwischen den teilnehmenden Parteien verteilt.

van Valkenburgh et al. (2015)
Um eine Übereinstimmung der Willenserklärungen zu gewährleisten, muss der Code für die Partei, die den smarten Vertrag ausführt ersichtlich sein. Dies kann bei einem an alle Nutzer gerichteten smarten Vertrags durch eine Veröffentlichung des Codes geschehen. Dies widerspricht jedoch dem Design-Prinzip des vertraulichen Mitwissens. Überprüfbar sind die Vertragsbedingungen beispielsweise über den kompilierten Code des smarten Vertrags und/oder mittels eines Hashwerts. Tokens können

Problem, bzw. Lösung erläutern

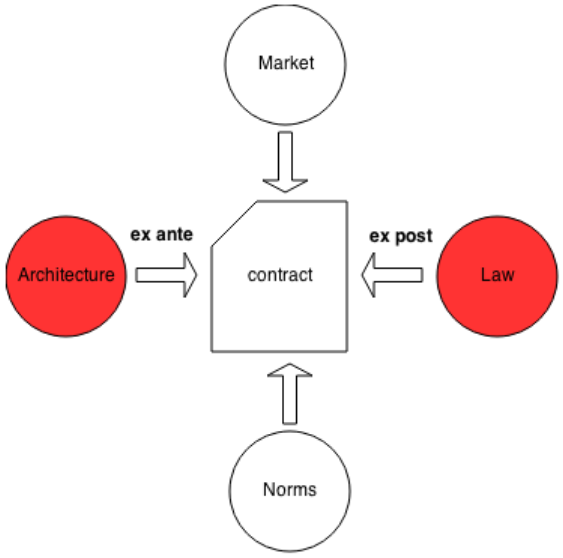


Abbildung 2.1.: *Smart Contracts im zeitlichen Kontext. Vgl. Glatz (2014)*

Ein smarter Vertrag muss dabei nicht zwingend einem juristischen Vertrag entsprechen.

ausführen

Dabei haben sie einen entscheidenden Unterschied gegenüber juristischen Verträgen: die zeitliche Komponente. Während

2.4. Entwurf der Blockchain nach Satoshi Nakamoto

2.4.1. Der Header

Für die Entwicklung der Technologie hinter Bitcoin wurden verschiedene bekannte Techniken aus den Bereichen Kryptographie, verteilte Systeme, Datenstrukturen miteinander verknüpft. Den zentralen Teil stellt eine verkettete Liste dar, die in jedem Node - jedem Teilnehmer in dem Netzwerk - gespeichert wird. Jeder Listeneintrag beinhaltet

einen Header mit Metadaten, der im folgenden beschrieben wird. Weiterhin enthält er eine genaue Auflistung aller in diesem Eintrag zusammengefassten Transaktionen. Ein header mit allen enthaltenen Transaktionen wird Block genannt. Da ein Block immer auf seinen Vorgänger verweist, entsteht eine Verkettung aller Blöcke untereinander, wodurch der Name Blockchain entstanden ist.

Listing 2.1: *Ein beispielhafter Block-Header Webbtc (2015)*

```
{
"hash": "000000000000000001f942eb4bfa0aeccb6a1 ...",
"ver": 2,
"prev_block": "00000000000000000a3ed9a4e2540751 ...",
"mrkl_root": "9b7d5896398581a7ff26be4b3684ddd95a ...",
"time": 1432723472
"bits": 404129525,
"nonce": 226994584,
"n_tx": 1031,
"size": 749157,
"tx": [
{...
}]
}
```

Der Block-Header, wie beispielhaft in Liste 2.1 angegeben, beinhaltet folgende Elemente:

hash	ein Hash des kompletten Headers
ver	Eine Versionsnummer, die Software/Protokoll-Upgrades anzeigt
prev_block	Der previous_block ist die Referenz auf den vorherigen Block. Dadurch kann jede getätigte Transaktion bis zum ersten Block, dem sogenannten Genesis-Block zurückgeführt werden
mrkl_root	Die Merkle Root ist ein Hash sämtlicher Transaktionshashes, wodurch sich alle Transaktionen effizient zusammenfassen lassen. Das genaue Vorgehen sowie der Einsatzzweck sind in Abschnitt beschrieben
time	Ein Zeitstempel des Blocks
bits	Die angestrebte Schwierigkeit der Berechnung des Blocks
nonce	Eine Zufallszahl. Sie ist die Zahl, mit welcher der Hash des Blocks errechnet wird. Ein Miner probiert dazu verschiedene Zufallszahlen aus, bis er auf die gesuchte oder eine höherwertige gestoßen ist.

Referenz

n_tx	Die Anzahl der im Block enthaltenen Transaktionen
size	Die Größe des Blocks in Bytes
tx	Die durchgeführten Transaktionen in Listenform

Vgl. Antonopoulos (2015), S.161
Es ist anzumerken, dass im eigentlichen JSON-Format eines Blocks kein Eintrag gegeben ist, an welcher Position nach dem Genesis-Block ein spezifischer Block steht. Dieser Wert wird oft angegeben zur einfachen Identifizierung eines Blocks, ist jedoch nicht immer eindeutig, da im Fall einer Gabelung der Chain zwei Blöcke die gleiche Positionsnummer haben können.

2.4.2. Transaktionen

Eventuell Schlüsse-laustausch beschrei-ben?

2.4.2.1. Vorgehen

Um Transaktionen durchzuführen, wird ein gültiger privater sowie öffentlicher Schlüssel benötigt. Dieser kann selbst erzeugt werden, z.B. durch die Installation eines Wallets, oder von einem externen Dienstleister wie einer Börse, welche die Schlüssel sowie das Guthaben verwaltet. Das Schlüsselpaar wird per ECDSA-Verfahren erzeugt, wobei der öffentliche Schlüssel danach weitere Hashfunktionen durchläuft, so dass eine Berechnung des privaten Schlüssels aus dem öffentlichen mit extrem hohen Aufwänden verbunden ist.
Umgekehrt lässt sich der öffentliche Schlüssel jedoch relativ einfach aus dem privaten Schlüssel generieren, wodurch es für einen Angreifer genügt, in Besitz des privaten Schlüssels zu kommen um ein Wallet komplett in seinen Besitz zu bringen. Theoretisch ist ein zufälliges Erraten eines privaten Schlüssels sehr unwahrscheinlich, da es für den 32-Byte langen Schlüssel 10^{77} Kombinationsmöglichkeiten gibt. *Vgl. Antonopoulos (2015)* Zum Vergleich: Es wird geschätzt, dass es 10^{78} Atome im Universum gibt. Zufällig einen privaten Schlüssel zu finden, der auf ein gültiges Konto mit Guthaben referenziert, ist also extrem unwahrscheinlich. *Vgl. Universität Frankfurt (2016)*

Seitenzahl angeben

Ein bis zur Entwicklung des Bitcoin-Protokolls nicht gelöstes Problem, war die Möglichkeit des „double-spending“ von Crypto-Währungen. Bei diesem Verfahren wird eine Geldeinheit and 2-n Empfänger überwiesen. Erhalten diese die Transaktionsnachricht gleichzeitig wird festgestellt, dass der Wert dem Sender noch nicht abgebucht wurde und Werten die Transaktion somit als gültig. Bisher wurde eine zentrale Instanz benötigt, die validiert hat, dass ein Wert nicht bereits an einen anderen Empfänger überwiesen wurde. Abbildung 2.2 zeigt, wie sich dieses Problem dezentral lösen lässt. Will B einen Betrag an C überweisen , muss C sicherstellen können, dass

Satoshis Zeichnung anpassen

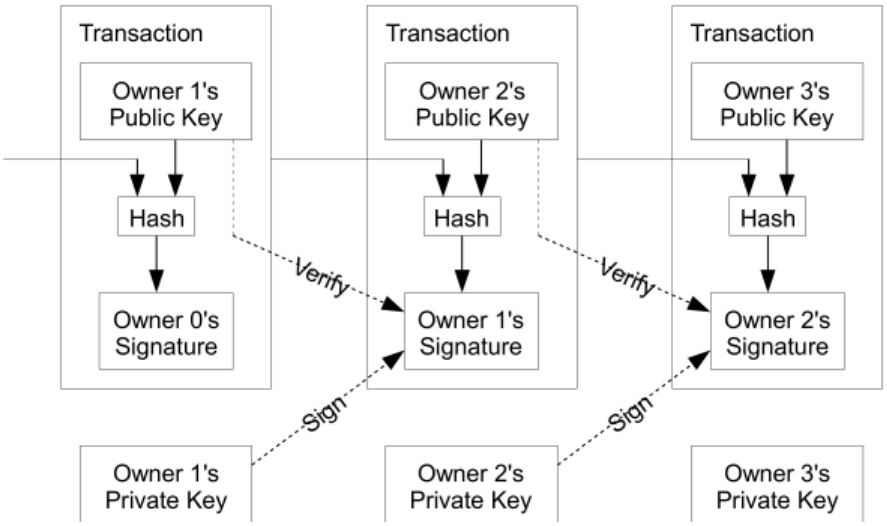


Abbildung 2.2.: *Die Transaktionskette Vgl. Nakamoto (2008), S.2*

1. B der Betrag gehört
2. B den Betrag nicht schon an jemand anderen überwiesen hat

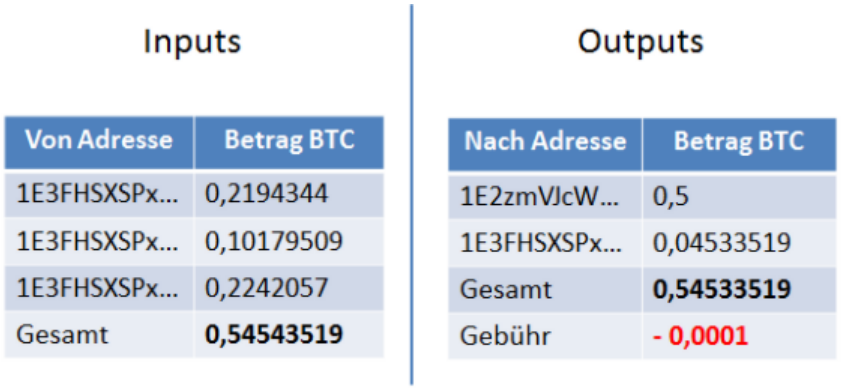
Besseres Wort

Dazu wird die vorherige Transaktion A –> B zusammen mit dem Public Key von C gehasht und schließlich mit B's privatem Schlüssel verschlüsselt . C kann nun die Transaktion mit dem öffentlichen Schlüssel von B entschlüsseln und erhält den von B zuvor generierten Hashwert. C kann diesen nun selbst generieren und weiß bei Übereinstimmung, dass kein Betrugsfall vorliegt. Dadurch ergibt sich eine Transaktionskette bis zum ersten „prägen" des Betrags, der sogenannten Coinbase-Transaktion. Diese wird automatisch als Belohnung für das Lösen neuer Blöcke - dem sogenannten Minen - ausgezahlt, wodurch neue Münzen in den Bitcoin-Kreislauf gelangen.

Guthaben werden nicht klassisch in einer Datenbank gespeichert, sondern sind in der Blockchain als Unspent Transaction Outputs (UTXO) einer Adresse zugeordnet. Wenn eine Wallet-Software also die Bilanz eines Kontos errechnet, werden alle Transaktionen die dem Besitzer des Wallets zugeordnet sind, ermittelt und diejenigen, die noch nicht übertragen worden sind, aufsummiert. eine UTXO ist unteilbar, d.h. Überweisungen werden aus UTXO's zusammengesetzt. Ist dieser Input höher als der sich daraus ergebende Output an die neue Adresse, wird die Differenz (abzüglich einer Transaktionsgebühr) als neuer UTXO an den Sender als Wechselgeld zurücküberwiesen. Antonopoulos (2015)[S.112ff]

eigene Zeichnung

Abbildung 2.2 stellt eine Transaktion dar. Um eine Überweisung von 0,5 Bitcoin (BTC) werden UTXO im Wert von mindestens 0,5 BTC der Adresse 1E3FHSXSPx... gesammelt. Sind diese gefunden, wird der Betrag von 0,5 BTC an die Zieladresse



Inputs

Von Adresse	Betrag BTC
1E3FHSXSPx...	0,2194344
1E3FHSXSPx...	0,10179509
1E3FHSXSPx...	0,2242057
Gesamt	**0,54543519**

Outputs

Nach Adresse	Betrag BTC
1E2zmVJcW...	0,5
1E3FHSXSPx...	0,04533519
Gesamt	**0,54533519**
Gebühr	- 0,0001

Abbildung 2.3.: *Input und Output einer Transaktion*

1E2zmVJcW... gesendet. Die Differenz - abzüglich einer Gebühr von 0,0001 BTC - wird als neue UTXO an den Empfänger zurücküberwiesen.

2.4.2.2. Transaktionsarten

Im Bitcoin-Protokoll sind neben der standardmäßigen Überweisung an eine öffentliche Adresse noch weitere Transaktionsarten für andere Einsatzzwecke möglich:

P2PKH — Der Pay-to-Public-Key-Hash ist das, was bei Bitcoin als die normale Überweisung an einen Empfänger gilt. Der Hash des öffentlichen Schlüssel des Empfängers stellt dabei dessen Adresse dar.

PTPK — Die Pay-to-Public-Key Variante ist eine vereinfachte Form des PTPKH, bei dem direkt an den öffentlichen Schlüssel gezahlt wird. Dies wurde zu Beginn von Bitcoin für Coinbase-Transaktionen genutzt, ist jedoch mittlerweile veraltet.

Multi Signature — Das Multi Signature Verfahren speichert N öffentliche Schlüssel im Skript, von denen eine Teilmenge M die Transaktion bewilligen muss, um diese auszulösen.

Data Output — Das Data Output Feld, bzw. op_return Feld erlaubt es zusätzliche Informationen an eine Transaktion einzufügen, z.B. den digitalen Fingerabdruck einer Datei, die folgenden Vertrag darstellt: „Ich erkläre hiermit, die Wertanlage A an XYZ zu Zeitpunkt ZEITSTEMPEL zu übergeben!“ Eine solche Transaktion kann nicht mehr weitergeleitet werden - sie kann nur einmal genutzt werden. Werden mit ihr Bitcoins übertragen, so sind diese Verloren, da sie nicht mehr ausgegeben werden können. In der Bitcoin-Community werden solche Übertragungen zwie-

Vermutlich wird dies für den Proof-of-Burn ansatz genutzt

spältig angesehen, da sie nicht die grundlegenden Funktionen der Übertragung von Bitcoins nutzen, andererseits aber zeigen, welche Anwendungsmöglichkeiten eine sichere und schwer manipulierbare Blockchain bietet. Aus diesen Gründen ist das op_return Feld auf 40 byte limitiert.

P2SH — Bitcoin verwendet zur Validierung von Transaktionen eine Skript-Sprache. Diese ist in ihrer Funktionalität sehr eingeschränkt, beispielsweise können keine Schleifen programmiert werden. Diese Einschränkung soll ein zuspammen des Netzwerks durch Funktionen mit Endlosschleifen oder anderen unlösbaren Rechnungen verhindern. Dennoch lassen sich Skripte mit einer gewissen Komplexität erzeugen, wie beispielsweise das Multi Signature Verfahren. Bei dem Pay-to-Script-Hash Verfahren wird der Hash des Skripts als Adresse verwendet und erst nach Validierung der Hashwerts das Skript ausgeführt. Dadurch werden Transaktionen insbesondere für Sender einfacher, da sie nur den Hashwert des Skripts brauchen und das eigentliche Skript verborgen werden kann.

Antonopoulos (2015), S.127ff

2.4.3. Peer-to-peer Netzwerke

Das Bitcoin Netzwerk ist als sogenanntes peer-to-peer Netzwerk aufgebaut. Es folgt dabei einer Architektur, die nicht dem zentralen Client/Server Aufbau entspricht, in welchem ein, bzw. mehrere zentrale Server die Anfragen von Clients bearbeiten. Stattdessen ist jeder Peer gleichzeitig Client und Server, wodurch es keinen zentralen Single-Point-of-failure gibt. Ein Beispiel für ein bekanntes peer-to-peer Netzwerk ist Bittorrent, als Netzwerk zum Filesharing. Dennoch können Nodes verschiedene Anwendungszwecke haben und somit unterschiedliche Aufgaben erfüllen:

Referenz Client	enthält Wallet, Miner, die komplette Blockchain und einen Network Routing Node
Kompletter Blockchain Node	Enthält die komplette Blockchain und einen Network Routing Node
Solo Miner	Enthält einen Miner, die komplette Blockchain und einen Network Routing Node
SPV-Wallet	Ein Simplified Payment Verification (SPV)-Wallet Enthält ein Wallet und einen Network Node

Pool mining Node	Enthält einen Miner und einen Network Routing Node der auf einen Mining Pool zugreift

Antonopoulos (2015), S.140
Ein Wallet stellt die digitale Geldbörse dar, in der Schlüsselpaare sowie Adressen der Nutzer gespeichert sind. Oft haben Wallets zusätzliche Funktionalitäten wie die Berechnung der Bilanz oder Verschlüsselung des Wallets zur Erhöhung der Sicherheit.
Ein Miner ist ein Programm, dass genutzt wird um die mathematischen Rätsel der Blockchain zu lösen. Ein Solo Miner benötigt dazu eine komplette Kopie der Blockchain, während einem Miner in einem Mining Pool i.d.R. ein Teil zur Berechnung zugewiesen wird. Die komplette Blockchain beinhaltet alle Blockheader sowie deren Transaktionen. Bei anderen Nodes werden nur die Header gespeichert.
Der Network Routing Node beinhaltet alle Funktionalitäten, die zur Kommunikation mit den anderen Nodes im Netzwerk benötigt werden.

2.4.4. Aufbau und Lösen von Blöcken

2.4.4.1. Mining

Als Mining wird das Lösen von Blöcken bezeichnet. Jeder Block ist mit einem mathematischen Rätsel versehen, das gelöst werden muss, damit ein Block in die Blockchain aufgenommen wird. dabei wird ein Rätsel benötigt, welches folgende Attribute aufweist:

1. Schwer zu errechnen
2. Leicht zu validieren
3. Einfach zu skalieren

Die Skalierbarkeit wird benötigt um die Berechnungszeit konstant zu halten. Bei zu geringen Berechnungszeiten entstehen viele Side-Chains und die Wahrscheinlichkeit, dass eine Instanz für einen gewissen Zeitraum genügend Blöcke löst um das System zu manipulieren, steigt stark an. Bei zu hohen Berechnungszeiten dauern Transaktionen sehr lange und das System wird ineffektiv. Die angestrebte Blockbildungszeit beträgt bei Bitcoin 10 Minuten, bei den meisten anderen Währungen liegt sie zwischen 1-10 Minuten. Pfnür (2014)[S.36]

Bitcoin verwendet als Rätsel den SHA256 Algorithmus. Der „hash“ Eintrag eines Blockheaders hasht den kompletten Header. In Liste 2.1 sieht man, dass dieser Hash mit einer gewissen Anzahl nullen beginnt. Diese stellen die Schwierigkeit der Berechnung dar und werden über den Wert „nonce“ gelöst. Wird ein neuer Block gebildet, so ist der Wert der Nonce auf 0. Die über das Feld „bits“ angegebene Schwierigkeit sagt aus, mit wie vielen Nullen der Hash beginnen muss, damit der Block als gelöst gilt. Ein Miner

hat zur Berechnung keine andere Möglichkeit als so lange Werte für die Nonce auszuprobieren, bis er auf das korrekte Ergebnis stößt. Dieses Ergebnis wird von dem Miner über das Netzwerk publiziert. Andere Nodes können den Hashwert des neuen Blocks durch die Hashfunktion mit der berechneten Nonce einfach validieren, wodurch sich ein gemeinsamer Konsens auf den neuen Block bildet. Bei ausreichendem Konsens wird der Miner belohnt und die Transaktionen werden ausgeführt. Da für dieses Verfahren Rechenleistung und Strom der Miner aufgewendet werden, die Rechner also für die Lösung „arbeiten“, wird dieses Verfahren Proof-of-Work genannt. **?**[S.3]

2.4.4.2. Belohnungen für das Mining

Die Belohnung für den Miner besteht in der Auszahlung einer gewissen Anzahl Bitcoins, die als erste Transaktion des neuen Blocks ausgeführt wird. Diese wird als Coinbase Transaktion bezeichnet, da sie als einzige keine einem Sender zugehörige Adresse hat, sondern die Bitcoins neu erzeugt werden. Zusätzlich erhält der Miner alle anfallenden Transaktionsgebühren. Transaktionsgebühren sind vom Sender festgelegte Gebühren, die ihm eine höhere Priorität bei der Aufnahme in Blöcke verschaffen. Blocks haben eine maximale Größe.[6] Transaktionen werden nach absteigender Priorität Blöcken nach folgender Formel hinzugefügt:

$$priority = \sum_{n=0}^{N} \frac{Wert\ des\ Inputs * Alter\ des\ Inputs}{Groesse\ in\ Bytes}$$

Mit zunehmendem Alter steigt also die Priorität, eine Transaktion dem Block hinzuzufügen, selbst wenn keine Transaktionsgebühren bezahlt wurden. Da die Anzahl über die Coinbase-Transaktion ausgezahlter Bitcoins degressiv verläuft und auf 21. Mio Bitcoins limitiert ist, soll Zukünftig die Belohnung der Miner ausschließlich über Transaktionsgebühren erfolgen. **?**[S.4] Ein solches Vorgehen hat einen zusätzlichen positiven Nebeneffekt: Es verhindert das „zuspammen“ des Netzwerks mit wertlosen bzw. sehr geringwertigen Transaktionen, was einem Denial of Service (DoS) Angriff gleichkäme.

2.4.4.3. Duplizierte Chains

Sollten zwei Miner zeitgleich einen Block lösen und über das Netzwerk propagieren, so werden beide zunächst als gültig erklärt und Miner arbeiten an jeweils dem Block, der sie zuerst erreicht hat. Sobald nun einer der beiden Blöcke gelöst wurde, arbeiten Miner an der längeren Chain, wodurch die kürzere nicht mehr bearbeitet wird und damit ungültig wird. In diesem ungültig gewordenen Block enthaltene Transaktionen werden wieder dem Pool an Transaktionen hinzugefügt und in den kommenden Blöcken verarbeitet.

[6] https://en.bitcoin.it/wiki/Block_size_limit_controversy

2.4.5. Manipulationssicherheit

2.4.6. Der Einsatz von Merkle-Trees zur effizienten Validierung von Blöcken

2.4.6.1. Berechnung der Merkle-Root

Mit Hilfe von Merkle-Trees lässt sich ein Block sehr effizient validieren. Die im Block-Header angegebene Merkle Root ist eine Zusammenfassung der Hashwerte aller Transaktionen.

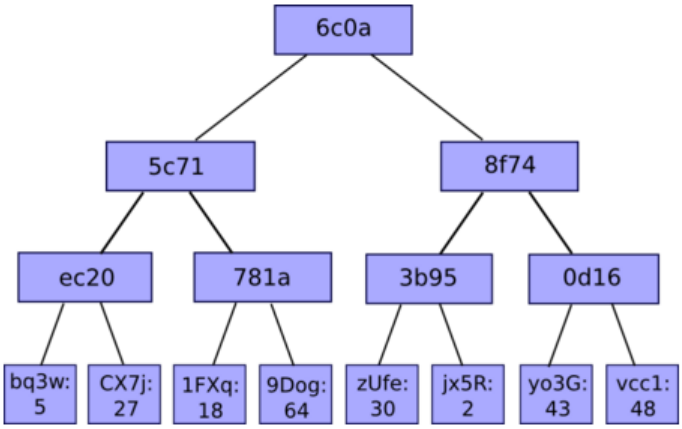


Abbildung 2.4.: *Beispiel für einen Merkle Tree Buterin (2014a)*

Vernünftige Grafik einbauen

Abbildung 2.4 stellt Grafisch den Algorithmus dar. Die Blätter des Baums sind die Hashwerte der einzelnen Transaktionen. In diesem Fall werden acht Transaktionen angenommen. Der Merkle Tree fasst diese Transaktionen immer paarweise zusammen [0,1][2,3][4,5][6,7] und erstellt einen Hash aus diesen. Aufgrund seiner Struktur benötigt der Merkle Tree immer eine gerade Zahl an Transaktionen. Ist diese nicht gegeben, wird der letzte Hashwert verdoppelt um den Baum auszubalancieren. Im Falle des Bitcoin Protokolls wird zweimal mit dem SHA256 Algorithmus gehasht. Dieser Vorgang wird solange wiederholt, bis ein Hashwert übrigbleibt: Die Merkle Root.
Versucht nun ein Angreifer eine Transaktion „einzuschmuggeln“, ändert sich der Hashwert der Merkle-Root. Es kann also die Validität eines Blocks effizient überprüft werden, ohne jede Transaktion abzugleichen.
Dies ist eine Voraussetzung für das Betreiben von unvollständigen Nodes, die nur die Block-Header speichern. **?**[S.5]

Tabelle 2.1.: *Arten von Blockchains*

Art der Chain	Art der Cryptowährung	Beispiele
Bitcoin Blockchain	Bitcoin	Bitcoin
Bitcoin Blockchain	Altcoin	Counterparty
Sidechain	Bitcoin	Rootstock, Testversionen von Bitcoin
Sidechain	Altcoin	Dogecoin, Ethereum, Litecoin, Dash

2.4.6.2. Simplified Payment Verification durch Merkle-Path

Die SPV speichert nur die Block-Header. Dadurch ist diese Form von Node sehr leichtgewichtig und kann beispielsweise auch auf einem Smartphone betrieben werden.

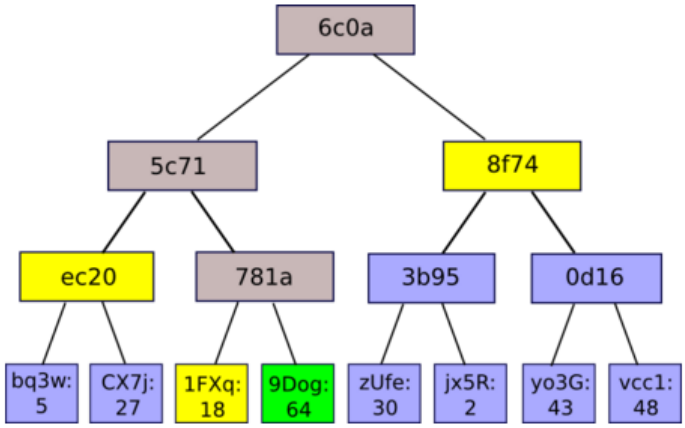


Abbildung 2.5.: *Beispiel für einen Merkle Path Buterin (2014a)*

Vernünftige Grafik einbauen

Um zu ermitteln ob ein Input einer Zahlung gültig ist, kann ein SPV nicht alle Transaktionen durchgehen. Zur Autorisierung müssen daher Full Nodes angefragt werden. Dazu werden die benötigten Merkle Hashes zur Bildung eines Pfads zur eigentlichen Transaktion abgefragt. In Abbildung 2.5 wird Block 9Dog abgefragt. Zum verifizieren der merkle_root werden nur noch Block ec20 und 8f74 benötigt. Buterin (2016)

2.4.7. Mining Pools

2.5. Erweiterungen des Bitcoin-Protokolls

2.5.1. Arten von Chains

Tabelle 2.1 zeigt die verschiedenen existierenden Chain-Arten. Die meisten existierenden Coins basieren auf Klonen der Bitcoin Blockchain in denen einige Paramater geändert

worden sind, wie z.B. der PoW-Algorithmus, die Blocklösungszeit, etc. Beispiele sind Dogecoin, Dash oder Litecoin.

2.5.2. Übertragung von Werten zwischen verschiedenen Chains

Durch verschiedene Modelle lassen sich Werte zwischen verschiedenen Chains übertragen und auf diesen halten.

Keine Erweiterung des Bitcoin-Protokolls sondern anwendbar auf alle chains

2.5.2.1. Gemeinsames Mining

In den unten beschriebenen Konzepten wird die Bitcoin-Blockchain als Parent-Chain und mögliche andere Chains als Side-Chains bezeichnet. Sidechains können dabei entweder

1. Auf der Bitcoin-Blockchain aufsetzten und keine eigene Währung besitzen
2. Auf der Bitcoin-Blockchain aufsetzten und 1-n eigene Währungen besitzen (Counterparty)
3. Nicht auf der Bitcoin-Blockchain aufsetzen

Dabei liegt die Annahme zugrunde, dass eine Sidechain die Möglichkeit bieten möchte, Transaktionen innerhalb der Chain mit Bitcoin ausführbar zu machen, da Bitcoin als Währung mit der höchsten Marktkapitalisierung am bekanntesten ist und am häufigsten benutzt, kann so die Einstiegshürde gesenkt werden.
Eine weitere Annahme ist, dass die Sidechain bereit ist, ihr Protokoll so anzupassen, dass auf ihr entweder nur, oder zusätlich noch Bitcoin gehandelt werden kann. Zur Realisierung im Bitcoin-Protokoll soll jedoch maximal ein Soft-Fork möglich sein, d.h. eine abwärtskompatible Anpassung, in der es für Miner, die ihr Protokoll nicht updaten, keine Einschränkungen gibt.

2.5.2.2. One-way peg

Der One-way peg erlaubt den Transfer von Einheiten in eine Richtung.[7] Dazu wird eine Geldeinheit an eine nicht weiterverwendbare Adresse geschickt. Durch den Nachweis, dass die Transaktion stattgefunden hat, sowie einem Nachweis, dass man der Sender der Transaktion war (i.d.R. über den Private-Key), werden einem auf einer anderen Chain eine entsprechende Menge an Geldeinheiten über eine Coinbase-Transaktion gutgeschrieben. Diese Methode ist vergleichsweise einfach implementierbar und relativ sicher. Jedoch kann sie nur in eine Richtung verwendet werden, da die gesendete Geldeinheit nachweisbar zerstört werden muss. Der Ansatz entspricht dem Proof-of-burn Ansatz.
Erstmals wurde er von der Firma Counterparty verwendet.Sie bietet erweiterte Funk-

Zitat

[7] https://www.mail-archive.com/bitcoin-development@lists.sourceforge.net/msg02346.html

tionalitäten auf der Bitcoin-Blockchain wie z.B. dem Erstellen eigener Tokens. Dafür besitzt sie eine eigene Währung (XCP), welche durch „verbrennen“ von Bitcoins erstellt worden ist. Transaktionen in XCP werden über eine Gebühr in Bitcoins bezahlt, wodurch Bitcoin-Miner einen Anreiz haben, XCP-Transaktionen in Bitcoin-Blöcken aufzunehmen.

2.5.2.3. Symmetrischer two-way peg

Beim two-way peg wird statt einer nicht mehr verwendbaren Adresse eine wiederverwendbare Benutzt. Dieser Ansatz bringt zwei Schwierigkeiten mit sich

1. Wie kann sichergestellt werden, dass Bitcoins nicht verdoppelt werden?
2. Wie werden nach Transaktionen auf der Sidechain die Besitzansprüche auf der Parent-Chain korrekt zugeordnet?
3. Wie wird nachgewiesen, dass bei „Wiederherstellung“ der Bitcoins auf der Parent-Chain, diese nicht auf der Sidechain noch übertragen werden?

Um diesen Anforderungen gerecht zu werden, muss eine Transaktion an eine Sidechain und zurück wie folgt aussehen:

1. Man führt eine Transaktion an eine bestimmte Bitcoin-Adresse aus, die so aufgebaut ist, dass sie nur dann wiederverwendet werden können, wenn nachgewiesen werden kann, dass sie nicht mehr auf einer Sidechain verwendet werden und dass man der rechtmäßige Besitzer ist.
2. Nach einer bestimmten Wartezeit, welche die Wahrscheinlichkeit senkt, durch eine DoS-Attacke das Netzwerk lahmzulegen und so eine double-spending Attacke in der nächsten Warteperiode ermöglicht, wird eine Nachricht an die Sidechain geschickt, die dann in der Parent-Chain per SPV prüft, ob die Überweisung korrekt durchgeführt wurde und durch eine ausreichende Anzahl Blöcke gesichert ist. Der Vorschlag des Unternehmens Blockstream beinhaltet eine Wartezeit von ein bis zwei Tagen. *Back et al. (2014), S.9*
3. Danach müssen die Bitcoins eine gewisse Zeit auf der Sidechain liegen, welches die Wahrscheinlichkeit senkt, dass kein double-spending vorliegt durch Überweisen von Transaktionen in einer geteilten chain. Ohne diese Wartezeit wird wie in Abbildung 2.6 dargestellt, auch eine Transaktion in einer geteilten Chain als valide anerkannt, obwohl es möglich ist, dass sie beim Lösen weiterer Blöcke als invalide eingestuft wird. Ist dies der Fall, so müssen die auf der Sidechain generierten, aber noch nicht freigeschalteten Bitcoins zerstört werden. Auch hier empfehlen Blockstream eine Wartezeit von ein bis zwei Tagen. *Back et al. (2014), S.9*
4. Diese Sidechain-Bitcoins können nun auf der Sidechain frei gehandelt werden.

5. Sollen sie wieder auf der Parent-Chain eingesetzt werden, läuft der Prozess genauso ab. Sie werden auf der Sidechain „eingefroren“ und der Besitz sowie die Überweisung auf das geblockte Konto per SPV nachgeprüft.

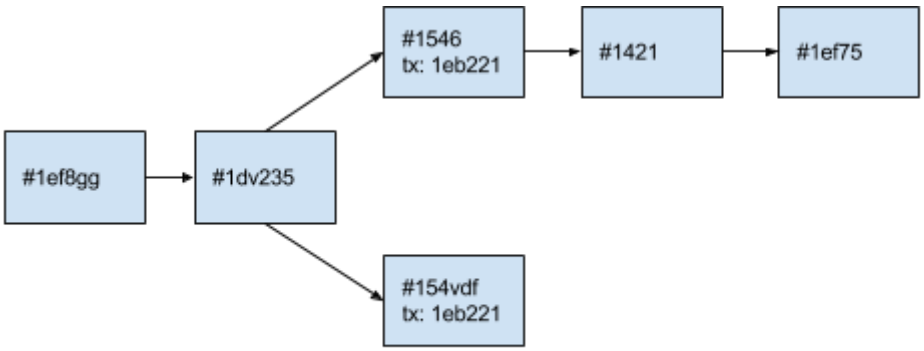


Abbildung 2.6.: *Double-spending über eine Sidechain (eigene Abbildung)*

Grafik überarbeiten

Bitcoin benötigt dafür jedoch einen Soft-Work um fähig zu sein, eine SPV Prüfung auf der Sidechain durchzuführen.

2.5.2.4. Asymmetrischer two-way peg

Sidechain Nutzer validieren die Parent-Chain komplett. Ein Litecoin-Nutzer hat also auch die Bitcoin-Blockchain gespeichert und kann alle Transaktionen validieren. Umgekehrt wird SPV-Proof genutzt.

2.5.2.5. Atomic Swaps

Atomic Swaps sind ein weiterer Ansatz, um Werte zwischen zwei Parteien auf verschiedenen Chains auszutauschen. Dabei wird die gleiche Menge an Münzen, bzw. der gleiche Wert (bei unterschiedlichen Währungen) zw. zwei Parteien A und B ausgetauscht. Dazu benötigt A auf Chain 1 eine öffentliche Adresse pkA sowie ein Geheimnis a. B benötigt lediglich eine öffentliche Adresse. Der Austausch findet wie folgt statt:

1. A initiiert Transaktion 1 (tx1) über eine Münze von pkA an einen Output O1, welcher entweder mit a und B’s Signatur oder A’s und B’s Signatur geöffnet werden kann. Diese Transaktion wird noch nicht im Netzwerk veröffentlicht.

2. A initiiert Transaktion 2 (tx2) in der nach einer locktime[8] von 48 Stunden die Münze zurück an pkA überwiesen wird.

[8] Eine locktime ist eine bestimmte Zeit, die eine Transaktion nicht weiter ausgeführt werden kann

3. B signiert tx2
4. A kann nun tx1 im Netzwerk öffentlich machen
5. B initiiert Transaktion 3 (tx3) über eine Münze von pkB an einen Output O2, welcher entweder mit a und B's Signatur oder A's und B's Signatur geöffnet werden kann. Diese Transaktion wird noch nicht im Netzwerk veröffentlicht.
6. B initiiert Transaktion 4 (tx4) in der nach einer locktime von 24 die Münze zurück an pkB überwiesen wird.
7. A signiert tx4
8. B kann nun tx3 im Netzwerk öffentlich machen
9. Nun kann A mit a auf die Münze in O2 zugreifen. Dies muss innerhalb von 24 Stunden geschehen, da sonst B automatisch Zugriff auf O2 erlangt.
10. Durch den Zugriff auf O2 wird a veröffentlicht. Dadurch erlangt B Zugriff auf O1 und muss die Münze innerhalb von 48 Stunden überweisen, da sonst A automatisch Zugriff auf O1 erlangt.

TierNolan (2013)
Dadurch, dass - im Gegensatz zu two-way-pegged Sidechains - bei Atomic Swaps keine Münzen in einem Output „eingefroren", sowie keine neuen Münzen auf der Sidechain erzeugt werden müssen, sondern sich lediglich der Besitzer ändert, können diese Swaps in etwa so schnell wie zwei normale Transaktionen durchgeführt werden. Er ist jedoch nur möglich, wenn eine Partei Interesse an Werten auf jeweils der anderen Chain hat und stellt daher keine Überweisung dar.
Im Gegensatz zu two-way-pegged Sidechains ist es jedoch möglich, einen Swap zw. zwei Währungen auszuführen, wenn sich beide Parteien auf einen bestimmten Wechselkurs einigen.

2.5.2.6. Bewertung der Ansätze

Das Koppeln einer Blockchain an die Bitcoin-Blockchain bringt verschiedene Vor- und Nachteile sowohl für die Parent-Chain als auch die Sidechain.

1. + Es entstehen Synergieeffekte. Eine erfolgreiche Sidechain profitiert von der Bitcoin Community und muss keine eigene Infrastruktur aufbauen. Dafür wächst die Bitcoin Community um die Anzahl der Sidechain-Nutzer.
2. - Es ist nicht mehr über den Kurs der eigenen Crypto-Währung ersichtlich, wie erfolgreich die Sidechain ist. Die Marktkapitalisierung ist ein guter Indikator, wie

stark ein neuer Ansatz akzeptiert wird. Dieser ist nur noch indirekt über die Entwicklung der Bitcoin Marktkapitalisierung sowie der auf die Sidechain übertragenen Münzen ableitbar.

3. - Unausgereifte Sidechains stellen eine Gefahr für die Nutzer dar. Eine Übertragung von Werten zwischen verschiedenen Sidechains sollte einen Nutzer nicht interessieren müssen . Ist die Sidechain jedoch unsicher implementiert, oder wird sie nur von wenigen Minern geschützt, so ist sie anfällig für Angriffe. Durch den Diebstahl der Parent-Chain Währung würde dies stark den Ruf der Parent-Chain schädigen.

ordentlicher Satz

4. + Nutzer müssen zur Verwendung der Zusatzfunktionen einer spezifischen Sidechain ihre Währung nicht umtauschen. Dies senkt die Eintrittshürde zur Verwendung.

Sie sind jedoch ausgezeichnet dazu geeignet, um neue Features, Updates und Weiterentwicklungen zu testen, die bei Erfolg in die Parent-Chain übernommen werden. Dazu können auch ökonomische Experimente zählen.
Ein weiterer, vielversprechender Ansatz ist die Möglichkeit, mehrere Parallelwährungen auf einer Chain zu führen. Back et al. (2014)[S.15f]

2.5.3. Übertragung von Verträgen auf verschiedenen Chains

Nicht nur die Übertragung von Werten zwischen verschiedenen Blockchains ist eine wichtige Funktionalität. Auch eine Kompatibilität und Ausführbarkeit von Vertragsbedingungen ist notwendig. Dies ist nur möglich, wenn entweder ein gemeinsamer Standard genutzt wird, oder verschiedene Blockchains jeweils den Interpreter einer Vertragssprache implementieren. Weiterhin müssen diese Blockchains per Definition Turing-Vollständig sein, da sie sonst keinen komplexen Programmcode interpretieren können
Rootstock behauptet, dass es Fähig ist, den Vertragscode von Ethereum zu kompilieren. Rootstock (2016)

2.6. Architektur Turing-Vollständiger Blockchains am Beispiel Ethereum

2.7. Das Hyperledger Projekt/R3CEV

wohl nicht mehr aktuell

2.8. Coin-Arten

Gedeckeltes inflationäres Modell –> Bitcoin ungedeckeltes inflationäres Modell –> Ethereum, Dogecoin Freicoin –> Bestimmte Geldmenge aber Geld wird regelmäßig von allen Kontenhaltern abgezogen und wieder dem Pool hinzugefügt

2.9. Blockchain-as-a-Service

3. Anwendungsmöglichkeiten der Blockchain/smarter Verträge

3.1. Anwendungsmöglichkeiten

3.1.1. Dezentrale, autonome Organisationen (DAOs)

3.1.1.1. Begriffsbeschreibung

Einer der wohl spannendsten Anwendungsfelder besteht in der Verwendung von Blockchain-Technologie um eine dezentrale, autonome Organisation (DAO) aufzubauen. Vitalik Buterin bemüht sich hierbei um eine Begriffserklärung, was eine DAO ist.

Decentralized Autonomous Corporation - Daniel Larimer

DAO - Vitalik Buterin

Distributed Collaborative Organization - Joel Dietz

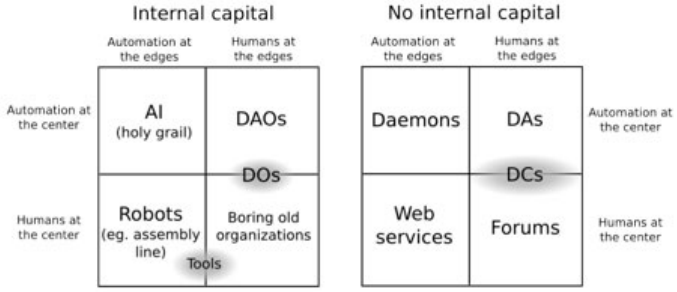


Abbildung 3.1.: *Verschiedene Unternehmensformen nach Art von Mensch und Maschinen-interaktion Buterin (2014*b*)*

Wie in Abbildung 3.1 ersichtlich wird, unterteilt er verschiedene Organisationsarten zunächst danach, ob sie eigenes Kapital verwalten und dann danach, in welcher weise Mensch und Maschine miteinander interagieren. Automatisierung in der Mitte drückt aus, dass im Zentrum der Organisation Maschinen stehen mit denen interagiert wird. Automatisierung am Rand drückt aus, dass Maschinen mit dem Zentrum der Organisation interagieren. Gleiches gilt für Menschen im Zentrum und am Rand. Entsprechend ergeben sich acht Organisationsformen:

1. Decentralized Applications (DAs) - Dezentralisierte Anwendungen, wie z.B. Peer-to-Peer Netzwerke wie Skype oder Bittorrent, in denen Menschen über ein dezentralisiertes Netzwerk direkt miteinander interagieren. Zu solchen Anwendungen ließe sich auch das frühe Internet zählen, bevor es hierarchisiert wurde.

2. Foren - Menschen interagieren mit Menschen und bauen dadurch eine Organisation auf.

3. Web Services -

4. Daemons - Beispielsweise Systemprogramme, die automatisch ausgeführt werden

5. gewöhnliche Organisationen - Systeme werden zwar unterstützend genutzt, jedoch wird sie komplett von Menschen geführt

6. Roboter -

7. DAOs - in der Mitte steht eine Anwendung, die autonom läuft, d.h. Grundfunktionen, wie z.B. Gehaltsströme würden auch ohne menschliche Interaktion funktionieren. Dezentral ist sie derart, dass ihr Programmcode nicht von einer zentralen Stelle verändert oder gelöscht werden kann. Menschliche Interaktion besteht in der Verwaltung der DAO, beispielweise ihrer Mitglieder oder dritter Parteien, die durch eine DAO bezahlt werden und dafür einen Profit erwirtschaften sollen.

8. AI - Eine Organisation, die sich komplett selbst verwaltet

Genauer beschreiben was eine DAO ist, durch zusammensetzen der Wörter dezentralisiert, autonom und organisation - Bitcoin als erste DAO oder DAC (Begriffflichkeiten klären - siehe https://www.youtube.com/watch?v=QG-CcbtwKKU&list=PLjgfpSQFJTLpKmTGCG8FjvDFbfst6F-x5&index=2 und https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-

[1] Einen starken medialen Anklang findet das Konzept von **DAO!** (**DAO!**)s. Sie stellen eine Art von Organisation auf der Blockchain dar, die ähnlich dem Crowdfounding von einer breiten Masse finanziert werden. Eine DAO verknüpft dabei jedoch die Vorteile des Crowdfounding und einer Aktiengesellschaft. Anstatt das Geld zu spenden, wird eine von der DAO verwaltete Währung - ein sogenannter Token - gekauft. Dieser Token repräsentiert einerseits einen gewissen Anteil an Stimmrechten, die Entscheidungen der DAO betreffen, sowie ein gewisser Anteil an Gewinnauszahlungen. Jeder Käufer von Tokens ist damit quasi ein Anteilseigner dieser Organisation.

3.1.1.2. Rechtliche Grundlagen einer DAO

Der rechtliche Status einer DAO ist bisher völlig ungeklärt. Da sie als Entität auf der Blockchain nicht in irgendeinem bestimmten Land ansässig ist, wird sie wohl für unbestimmte Zeit keinen Status als juristische Person erlangen. Dies führt zu zum Teil

[1] http://cointelegraph.com/news/decentralized-autonomous-organizations-ethereum-sparks-up-googles-of-tomorrow

recht abenteuerlichen rechtlichen Konstrukten. Stephen Palley beschreibt in einem Blog-Beitrag den Status einer DAO aus der Perspektive Britischen Rechts. Laut ihm würde eine DAO einem „General Partnership“ entsprechen. In dieser Form sind die Teilnehmer in einer DAO voll haftbar. Zusätzlich würde es vor Gericht ausreichen, einen Teilnehmer einer DAO zu identifizieren, dieser wäre dann verantwortlich dafür, eine Strafe gleichmäßig auf die Teilnehmer zu verteilen.
- Weltweit verteilt –> wem und wie werden steuern bezahlt?
- Wie kann sie angeklagt werden?
- Wer klagt sie an?

3.2. Praxisbeispiele

3.2.1. Verknüpfung von IOT und Blockchain: slock.it

3.2.2. Hedging über digitales Gold: digix

Gold kaufen über Blockchain

3.2.3. Dezentralisierte Prognosemärkte: Augur

3.2.4. Coins mit gekoppelter Wertentwicklung: Bitshares

3.2.5. smart Property

- Tokens/eigene Währungen
- Namecoins
- Colored coins
- Smart-Contract
- Decentralized autonomous organizations DAOs
- Smart properties
- Token Systeme (Generierung einer eigenen Währung, die z.B. den Anspruch auf ein Wertpapier definiert (smart property)
- Financial derivatives and Stable-Value Currencies (Generierung einer eigenen Währung, die einen fixen Wert, z.B. 1000USD hat. Wenn sich beide Parteien darauf einigen,

- Identity and Reputation Systems (Named coins, z.B. registrierung von Domain-Namen auf der blockchain
- Decentralized File Storage (Benutzung von speicherplatz auf Basis von mikro-payments
- Decentralized Autonomous Organizations
- Savings wallets
- Crop insurance
- A decentralized data feed
- Smart multisignature escrow
- Cloud computing
- Peer-to-peer gambling
- Prediction markets Augur
- On-chain decentralized marketplace

4. Überprüfung der Anwendungsmöglichkeiten

4.1. Ethereum als BaaS-Plattform

4.1.1. Bewertungskriterien

5. Technische Herausforderungen der Blockchain/smarter Verträge

5.1. Herausforderungen auf Basis von Wallets

5.1.1. Diebstahl von Wallets

Wie auch bei zentralen Diensten muss ein Nutzer sein Passwort schützen. Bei Bitcoin-Wallets besteht das Problem, dass es ausreicht an den privaten Schlüssel zu kommen, um als Angreifer Macht über das gesamte Konto zu erlangen. Theoretisch ist die Wahrscheinlichkeit einen privaten Schlüssel zu erraten sehr gering, praktisch hängt es von der gewählten Passphrase ab, auf die der ECDSA-Algorithmus angewandt wird. Korantin Auguste stellt in einem Blogbeitrag dar, wie durch die Verwendung kleiner Zahlen oder von Beispielsätzen (sogenannten Brainwallets) bereits erfolgreich Attacken auf Wallets verübt worden sind.
Außerdem ist es möglich durch den Einsatz von Malware, Phishing Tools, Social Engineering, etc. direkt an den Besitz des privaten Schlüssels zu gelangen.
Einen besseren Schutz bietet Ethereum. Hier muss bei der Erstellung eines Wallets zusätzlich noch ein Passwort angegeben werden. Mit diesem wird der private Schlüssel verschlüsselt. Dadurch muss ein Angreifer, wenn er in den Besitz des verschlüsselten Schlüssels gelangt zusätzlich noch das Passwort erraten.

5.1.2. Verlust von Wallets

Verliert ein Nutzer seinen privaten Schlüssel, so verliert er auch jede Zugriffsmöglichkeit auf das Wallet. Da es keine zentrale Instanz gibt, die den Schlüssel wiederherstellen kann, ist das Wallet unwiderruflich Verloren. Werden Coins an ein solches Wallet überwiesen, sind auch diese verloren.

5.2. Skalierbarkeit

5.3. Fehlende Belohnungen für die Nutzung von Full Nodes

Full Nodes sind bei der Blockchain-Technologie ein wichtiger Baustein um Qualität und Sicherheit des Netzwerks sicherzustellen. Sie Überprüfen unter anderem die Gültigkeit einer propagierten Blockchain, versorgen andere Nodes mit Informationen über die Blöcke und stehen für Light Nodes für den Merkle-Proof zur Verfügung.
Das Betreiben eines Full Nodes sorgt daher bei dem Nutzer für eine höhere Sicherheit und Anonymität als die Verwendung eines Light Nodes. Für die meisten Nutzer ist es jedoch vollkommen ausreichend einen Light Node zu betreiben, weshalb bei Bitcoin die Anzahl an Full Nodes stark abnehmend ist. Ohne zusätzliche Belohnungen, die Strom und Hardwarekosten kompensieren lohnt sich das Betreiben eines Full Nodes für den durchschnittlichen Nutzer nicht.

5.4. Kompatibilität von Blockchains

5.5. Anbindung von Schnittstellen

5.6. Light Nodes

durch SPV kann nur festgestellt werden, ob eine Transaktion tatsächlich durchgeführt worden ist. Um sicherzustellen, dass diese Transaktion nicht doppelt ausgeführt wurde (double spending), muss ein SPV bei mehreren Nodes nachfragen, um die Wahrscheinlichkeit zu erhöhen, dass einer dieser Nodes ein ehrlicher ist und nicht der eines Angreifers.
Dadurch ist ein SPV angreifbar durch partitoning attacks oder sybil attacks. Antonopoulos (2015)[S.149]

5.7. Sicherheit

5.7.1. Sybil-Attacke

5.7.2. Selfish-Mining

5.7.3. 51% Attacke

Die 51% Attacke umschreibt einen Angriff, in dem ein Angreifer 51% der Rechenleistung des Netzwerks aufbringt. Dadurch hat der Angreifer eine höhere Wahrscheinlichkeit, neue Blöcke zu bilden als der Rest des Netzwerks. Durch nicht-Veröffentlichung der gefundenen Blöcke kann er eine parallele Blockchain bilden, die länger als die von den ehrlichen Minern erarbeitete ist. Sobald er diese veröffentlicht, wird nach der "longest chain rule", welche besagt, dass immer der längsten Kette gefolgt wird, sofort von allen Minern von der bekannten Chain auf die des Angreifers gewechselt. **?**[S.3] Doch welche Möglichkeiten ergeben sich nun für den Angreifer?

1. Der Angreifer kann alle oder eine Teilmenge aller ab der Teilung der Kette getätigten Transaktionen rückgängig machen

2. Der Angreifer kann ein „double spending“ durchführen, indem er in der „ehrlichen“ Kette eine Transaktion ausführt bis sie vom Empfänger als gültig akzeptiert wird und diese dann in seiner „neuen“ Kette rückgängig macht. Dadurch geht die UTXO auf sein Konto zurück und er kann die Geldeinheiten wiederholt ausgeben.

Nach dem Bitcoin-Design von Satoshi Nakamoto soll durch das Mining sichergestellt werden, dass viele Miner, die jeweils einen kleinen Anteil an der gesamten Rechenleistung haben, es für einen Angreifer zu kostenintensiv ist eine solche Attacke auszuüben. Zusätzlich wird von ihm angenommen, dass die Belohnungen für ehrliches Mining höher sind, als die die durch Angriffe erzielt werden können. Insbesondere da bei einem erfolgreichen Angriff der Wert der Währung massiv sinken würde. Durch das Einführen von sogenannten ASIC-Minern [1] mit Hashraten von mehreren Tera-Hashes pro Sekunde, wurde das normale Mining auf klassischen Rechnern nicht mehr profitabel. Für diejenigen, die trotzdem noch Minen oder in einen ASIC-Miner investiert haben besteht zusätzlich noch das Problem eines sehr unregelmäßigen Kapitalrückflusses. Mit ca 1.200.000 Tera Hashes pro Sekunde [2] ist es relativ unwahrscheinlich selbst einen Block zu lösen. Daher haben sich viele Miner in Mining-Pools zusammengeschlossen (siehe Mining Pools) Abbildung 5.1 zeigt, das bereits 2 der größten Mining Pools mehr als 50% der Mining Power besitzen. Ein möglicher Angriff wäre beispielsweise die Server der Pool Manager

Begründung oder Quelle

Referenz

Quelle

[1] http://asicminer-shop.de/Bitmain-Antminer-S7-486-TH-s-Gebrauchtgeraet
[2] https://blockchain.info/de/charts/hash-rate

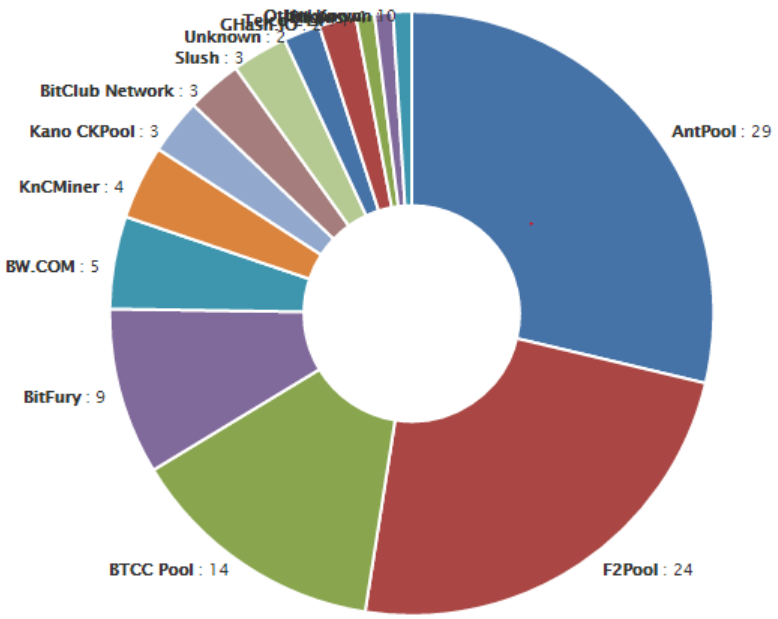


Abbildung 5.1.: *Verteilung der Bitcoin Mining pools*

zu hacken. Doch auch ein Angriff mit eigener Hardware wäre für einige Organisationen möglich. James D'Angelo beschreibt in einer 2-teiligen Youtube Serie mögliche Organisationen, die für bereits ab 20 Millionen Euro kosten eine mögliche Gefahr für das Bitcoin Netzwerk darstellen können. Dazu zählen hauptsächlich die Hersteller von Mining Computern. Angelo (2014*a*) Angelo (2014*b*)

Ein Angriff aus finanziellen Gründen ist jedoch sehr unwahrscheinlich. Die einzige Motivation wäre das Durchführen von „double spending" Angriffen. Diese wären selbst für einen Chip-Hersteller vergleichsweise teuer und auch wenn ein solcher Angriff auf der Chain nicht ersichtlich ist, würden die geschädigten Parteien wohl sehr schnell den Betrug propagieren. Ein vorstellbares Szenario wäre jedoch der Versuch, die Währung als ganzes zu zerstören. Insbesondere Staaten wie Russland und China, die eine sehr restriktive Geldpolitik haben und befürchten, dass die Bürger ihre Einlagen in Crypto-Währungen sichern, hätten hierzu eine deutliche Motivation. Sie könnten für eine längere Zeit mit 51% der Rechenpower eine Chain mit keinen oder nur sehr wenigen Transaktionen mitführen. Da zum Zeitpunkt der Veröffentlichung in die neue Chain gewechselt würde, würden wie beschrieben alle bis zum Zeitpunkt des Forks getätigten Transaktionen zurück gerollt werden, was Crypto-Währungen als Zahlungssystem wahrscheinlich zumindest für einige Jahre obsolet machen würde.

5.7.4. Notizen

- Wucherungen des Hauptbuchs. Jedes Wallet muss eine Kopie des Hauptbuchs herunterladen. Diese wächst mit Bekanntheit der Anwendung. Insbesondere problematisch für mobile Anwendungen
- Tendenz zur Monopolisierung des Netzwerks in einem einzigen Mining Pool
- Tendenz zur Zentralisierung der Geldmenge
- Wie viel des Netzwerkes benötige ich um die Blockchain zu „Kapern“? Was sind die Auswirkungen?
- Code als Single-Point-of-Failure. Wie kann – insbesondere bei programmierbaren Coins Sicherheit gewährleistet werden?
- Bei Smart contracts kann jeder die Vertragsbedingungen (bzw. den gecodeten Vetrag) einsehen. Dies kann z.B. zu einem Interessenkonflikt mit anderen Vertragspartnern führen. Welche Möglichkeiten, dass zu unterbinden bestehen?
- Schutz durch Proof-of-work. Annahme: Bitcoin-miner haben ausschließlich finanzielle Interessen. Ihr Verdienst ist die Belohnung durch das Lösen von Blöcken. d.h. Gewinn = Belohnung – Mining-kosten Will jemand (Eve) das Netzwerk übernehmen, besticht er die Mining-Pools. Diese haben dadurch keine extra-kosten, Eve hat folgende Gewinn-Funktion: Gewinn = Belohnung durch übernehmen des Netzwerks – Kosten für Bestechung der Mining-Pools. diese Kosten sind bei der Annahme rein finanzieller Interessen >= 0
- Sybil attack

5.8. Resümee

6. Fachliche Herausforderungen der Blockchain/smarter Verträge

6.1. Dezentralisierte Problemlösung/Updates der Blockchain

Hearn (2016)

7. Lösungsansätze für derzeitige Hindernisse

7.1. Skalierbarkeit

7.2. Bedarf einer regulierenden Instanz

7.3. Volatilität

7.4. Fehlerhafte Überweisungen

7.5. Anonymität vs Dezentralisierung

Satoshis Ansatz, die Entitäten zu anonymsieren und den Zahlungsverkehr öffentlich zu machen, kann ein großer Fehler sein, da niemand prüfen kann, ob das System nicht von einer einzigen zentralen Entität kontrolliert wird. [1] Lösungsvorschlag: Identity Mining. Durch hinterlegen einer Identität kann überprüft werden, dass verschiedene Personen minen (Proof-of-Reputation) + PoW

7.6. Gültigkeit programmatischer Verträge

7.7. Mining

7.7.1. Proof-of-Stake

7.7.2. Proof-of-Burn

7.7.3. Proof-of-Reputation

[1] https://www.youtube.com/watch?v=J52AM5SrOHw

8. Abschliessende Betrachtung

8.1. Resümee

8.2. Ausblick

Literaturverzeichnis

Angelo, J. D. (2014*a*), 'Bitcoin 101 - the nightmare of a 51% attack - part 1 - calculating the costs'.
https://www.youtube.com/watch?v=bi2thGzzNSs

Angelo, J. D. (2014*b*), 'Bitcoin 101 - the nightmare of a 51% attack - part 2 - how to destroy bitcoin'.
https://www.youtube.com/watch?v=Kjtgp5h-jEY

Antonopoulos, A. M. (2015), *Mastering bitcoin: Unlocking digital cryptocurrencies*, first edition edn, O'Reilly, Sebastopol, CA.

Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J. & Wuille, P. (2014), 'Enabling blockchain innovations with pegged sidechains'.
http://www.blockstream.com/sidechains.pdf

Buterin, V. (2014*a*), 'Daos, dacs, das and more: An incomplete terminology guide'.
https://blog.ethereum.org/2014/05/06/
daos-dacs-das-and-more-an-incomplete-terminology-guide/

Buterin, V. (2014*b*), 'What is a merkle tree'.
https://www.weusecoins.com/what-is-a-merkle-tree/

Buterin, V. (2016), 'A next-generation smart contract and decentralized application platform'.
https://github.com/ethereum/wiki/wiki/White-Paper

Geiling, L. (2016), 'Distributed ledger: Die technologie hinter den virtuellen währungen am beispiel der blockchain', *BaFin Journal* (2/2016), 28–32.
http://www.bafin.de/SharedDocs/Downloads/DE/BaFinJournal/
2016/bj_1602.pdf?__blob=publicationFile&v=3

Glatz, F. (2014), 'What are smart contracts? in search of a consensus.'.
https://medium.com/@heckerhut/whats-a-smart-contract-in-search-of-a-co
.n75czj8ro

Hearn, M. (2016), 'The resolution of the bitcoin experiment'.
https://medium.com/@octskyward/the-resolution-of-the-bitcoin-experim
.owgtvb9ry

Nakamoto, S. (2008), 'Bitcoin: A peer-to-peer electronic cash system'.
https://bitcoin.org/bitcoin.pdf

Pfnür, C. (2014), Vergleich dezentraler elektronischer Währungen, PhD thesis, Hochschule für Technik und Wirtschaft Dresden, Dresden.
http://www2.htw-dresden.de/~jvogt/abschlussarbeiten/Pfn%
C3%BCr.pdf

Rootstock (2016), 'Following the yellow brick road'.
http://www.rootstock.io/blog/following-the-yellow-brick-road

Szabo, N. (1996), 'Smart contracts: Building blocks for digital markets'.
http://szabo.best.vwh.net/smart_contracts_2.html

Szabo, N. (1997), 'Formalizing and securing relationships on public networks', *First Monday* **2**(9).

TierNolan (2013), 'Re: Alt chains and atomic transfers'.
https://bitcointalk.org/index.php?topic=193281.
msg2224949#msg2224949

Universität Frankfurt (2016), 'Anzahl der atome'.
https://elearning.physik.uni-frankfurt.de/data/
FB13-PhysikOnline/lm_data/lm_282/auto/kap09/cd236b.htm

van Valkenburgh, P., Dietz, J., de Filippie, P., Shadab, H., Xethalis, G. & Bollier, D. (2015), 'Distributed collaborative organisations: Distributed networks & regulatory frameworks'.
http://bollier.org/sites/default/files/misc-file-upload/
files/DistributedNetworksandtheLaw%20report%2C%
20Swarm-Coin%20Center-Berkman.pdf

Webbtc (2015), 'Beispielhafter block'.
https://webbtc.com/block/000000000000000001f942eb4bfa0aeccb6a14c268f4
json

A. Anhang