Leibniz Universität Hannover
Institut für Theoretische Informatik

Bitcoin

Der Stärkere gewinnt

Gerion Entrup

6. September 2013

Betreut von Dr. Arne Meier

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

_____

Inhaltsverzeichnis

1. Einleitung

Geld wurde vor etwa 2700 Jahren zu dem wichtigsten Tauschmittel der Menschheit [60]. Anstatt die Waren selbst untereinander zu tauschen, wurde gegen ein Metallstück getauscht, das an anderen Orten denselben Wert hatte. Mit der Einführung des Papiergeldes war der materielle Wert erstmals erheblich kleiner als der repräsentierte Wert. Banken und Staaten standen dabei für den Wert ein. In der heutigen digitalen Welt wird das System noch einmal mehr auf bloße Zahlen reduziert, die elektronisch transferiert werden. Dabei ist jedoch immer eine dritte Partei vonnöten, die für den Wert bürgt. Bitcoin ist ein System, das genau diesen Missstand beseitigen will.

In dieser Arbeit werde ich mich mit der Technik des Systems auseinandersetzen und dabei den Fokus auf die Netzwerkstruktur richten.

1.1. Bob will mit Bitcoin bezahlen

Alice betreibt einen Versand für Strohhüte und Bob braucht einen davon. Anstatt mit der Währung eines beliebigen Landes zu zahlen, beschließt Bob, Bitcoin zu verwenden. Alice hat dafür ihren öffentliche Adresse `1PGT34HT4TwsN8VJrCrisVcJjDpc7Qxepp`[1] über einen sicheren Kanal Bob zukommen lassen, beispielsweise über ihre SSL-verschlüsselte Website. Bob öffnet dann seinen Bitcoin-Client und überweist von den 2,53483 BTC[2], die er hat, 0,00012 BTC an Alice, da das der Preis für einen Hut ist. Diese Überweisung signiert er mit seinem Schlüssel, damit jeder weiß, dass der rechtmäßige Besitzer der Bitcoins diese überwiesen hat und Alice sie jetzt besitzt. Der Client macht daraufhin die Transaktion bekannt, indem er seinen gesamten Nachbarn im Netzwerk davon berichtet und diese wiederum ihren gesamten Nachbarn usw.. Alle anderen Knoten im Netzwerk überprüfen dann, ob Bob wirklich Besitzer des Geldes war und ob er dieses Geld nicht schon einmal überwiesen hat. Wenn alles korrekt verlaufen ist, verrechnen die anderen Knoten in einer rechenintensiven Aufgabe die Transaktion, geben das Ergebnis bekannt und bestätigen so die Transaktion. Bob hat damit erfolgreich einen Strohhut per Bitcoin bezahlt.

1.2. Ziel

Vor der Existenz des Bitcoin-Systems war eine Transaktion im digitalen Raum immer an eine dritte Partei gebunden, die verifiziert hat, dass

1. dasselbe Geld nicht doppelt überwiesen wird und
2. das Geld, das überwiesen wird, auch im Besitz des Überweisenden ist.

Dieser dritten Partei muss vertraut werden. Zudem erhebt sie üblicherweise Transaktionsgebühren. Das Bitcoin-System wurde von Satoshi Nakamoto entworfen, um diesen Missstand zu beseitigen:

[1] Ausgedachte Adresse
[2] Kurzschreibweise für Bitcoins

> „What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers.“ [61]

Die Transaktionen müssen aber dennoch verifiziert werden. Bei Bitcoins passiert dieser Vorgang durch die gesammelte Rechenleistung des gesamten Netzwerkes.

1.3. Geschichte

Mit dem Aufkommen der Computer wurde auch der Handel mit Geld digitalisiert. Dabei gab es jedoch immer eine dritte Partei – eine Bank – die den eigentlichen Überweisungsvorgang durchführte. 2008 [3] veröffentlichte Satoshi Nakamoto ein Paper [61], das einen Geldhandel ohne dritte Partei beschrieb. Kurze Zeit später folgte eine erste Implementierung. Das Netzwerk als solches besteht seit 2009 [3].

Inzwischen hat sich Satoshi Nakamoto vollständig aus dem Projekt zurückgezogen und die Weiterentwicklung einer Vielzahl von Entwicklern überlassen. 2012 wurde die Bitcoin Foundation gegründet um Bitcoin zu standardisieren, zu schützen und zu vermarkten [64].

Das System erlangte in den Fachmedien und z. T. sogar in der normalen Presse bereits zweimal erhöhte Aufmerksamkeit (im Mai 2011 und April 2013), beide Male einhergehend mit rasanten Kurszu- und abnahmen[3] (siehe Abschnitt 4.1.2).

Satoshi Nakamoto ist dabei ein Pseudonym für eine Einzelperson oder eine Gruppe von Personen. Gute Englischkenntnisse und keinerlei sonstige Veröffentlichungen in Japanisch lassen eventuell auf einen US-Amerikaner oder Briten schließen [29].

1.4. Aufgabe

Ziel dieser Bachelorarbeit ist es, den technischen Aufbau des Bitcoin-Netzwerkes zu untersuchen. Dabei soll besonders auf die Kommunikation zwischen den Teilnehmern eingegangen werden. Dabei soll auch mit anderen Netzwerken verglichen werden. Außerdem sollen die Aspekte Sicherheit und Anonymität näher beleuchtet werden. Zu guter Letzt soll ein Überblick über die Einbettung von Bitcoin in bisherige Zahlungspraktiken gegeben werden.

[3] Der Wechselkurs von Bitcoin zu Dollar

2. Technik

Das grundlegende Element des Bitcoin-Netzwerkes ist die *Transaktion* (2.1). Eine Repräsentation für einen Bitcoin selbst oder Bruchteile davon existiert nicht. Transaktionen werden mit Schlüsseln signiert, die in der *Wallet* (2.2) gespeichert sind. Mehrere Transaktionen werden durch *Mining* (2.4) zu einer *Block Chain* (2.3) zusammengefasst.

2.1. Transaktion

Eine Transaktion besteht aus 6 Teilen [33]:

1. Versionsnummer
2. Anzahl der Eingänge
3. Liste aller Eingänge
4. Anzahl der Ausgänge
5. Liste aller Ausgänge
6. Zeitstempel

Interessant davon sind die Listen der Ein- und Ausgänge. Da es keine digitale Entsprechung für Bitcoins gibt, werden die zu überweisenden Bitcoins als Summe von (früheren) Transaktionen angegeben. Diese bildet dann die Liste der Eingänge. Weiterhin können in einer Transaktion mehrere „Ziele“ angegeben werden, welche die Liste der Ausgänge bilden. Das ist notwendig, da immer der gesamte Geldbetrag der Eingangstransaktionen weitertransferiert werden muss. Will man nun einen kleineren Geldbetrag transferieren, als durch die Eingänge gegeben ist, transferiert man den gewünschten Geldbetrag an das gewünschte Ziel und den Rest an sich selbst.

Klarer wird das durch ein Beispiel (siehe auch Abbildung 1):

> S will 3 BTC an E transferieren. Von A und B wurden S jeweils 2 BTC transferiert. Die Transaktion von S an E besäße nun als Liste der Eingänge die Transaktion von A nach S und die von B nach S. Die Liste der Ausgänge bestünde aus 3 BTC an E und 1 BTC zurück zu S.

Der Geldbetrag, der an sich selbst zurücktransferiert wird, wird auch *Wechselgeld* genannt. Es kann vorkommen, dass die Summe der Bitcoins der Eingänge höher ist, als die der Ausgänge. Die Differenz wird als *Transaktionsgebühr* demjenigen gutgeschrieben, der den Block verifiziert (siehe Abschnitt 2.4).

Die Ausgänge bestehen aus einer Liste von Bitcoin-Adressen. Diese Adressen sind Hashwerte von Public Keys, im Genauen *ripemd160(sha256(public_key))* [32]. Der Public Key ist dabei der öffentliche Schlüssel einer asymmetrischen Verschlüsselung [61]. Bitcoin benutzt dafür den *Elliptic Curve Digital Signature Algorithm (ECDSA)* mit der *secp256k1*-Kurve [28] (siehe Anhang B).

Die gesamte Transaktion wird mit den Private Keys signiert, die zu den Eingängen der Transaktion gehören (die Eingänge bestehen aus Transaktionen mit den Hashwerten der zu den Private Keys gehörenden Public Keys als Adressen). Diese Signaturen können mit den Public Keys, die

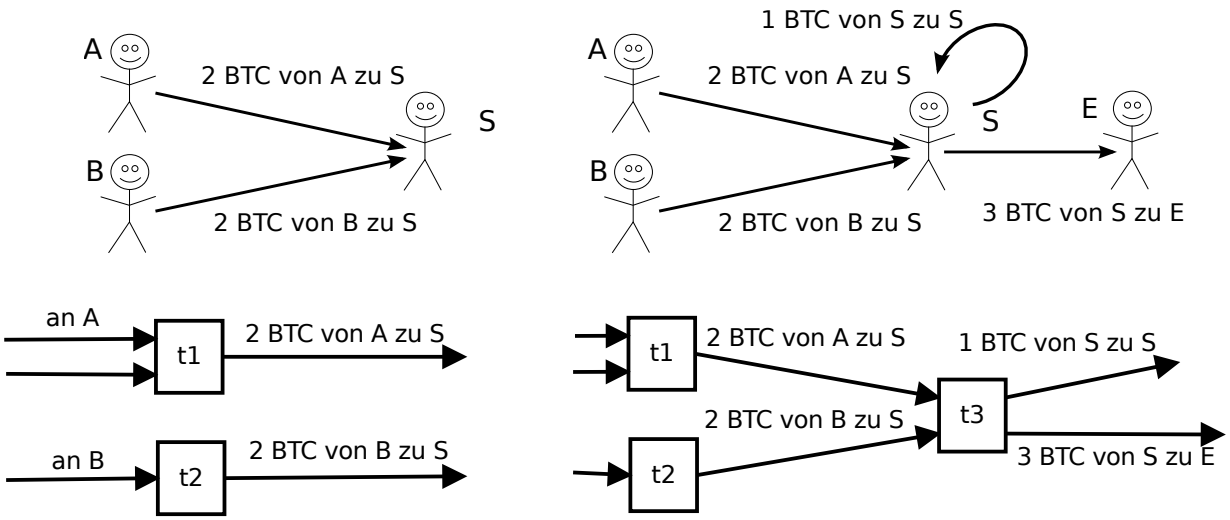

Abbildung 1: Graphische Verdeutlichung des Beispiels, links: momentane Situation, rechts: Situation, wenn S 3 BTC an E überweisen will, die unteren Abbildungen entsprechen den Transaktionen, die abgespeichert werden; die Personen sind dort nur noch in Form von Ein- und Ausgängen vorhanden. Strichmännchen von [42]

ebenfalls in die Transaktion eingebettet werden, überprüft werden. Die Überprüfung der Korrektheit der Public Keys ist wiederum durch Hashen (siehe Anhang A) und Vergleichen mit den Adressen möglich (siehe auch Abbildung 2).

Die Eingänge werden als Hashwerte der vorherigen Transaktionen angegeben. Bitcoin verwendet dazu einen doppelten SHA-256 Hash als Funktion: *sha256(sha256(transaction))* [28].

Eine Transaktion kann man nicht rückgängig machen. Zurücktransferieren würde eine neue Transaktion nach sich ziehen. Die Menge der Transaktionen kann man also als einen gerichteten azyklischen Graphen interpretieren [40] (siehe Abbildung 1 und Abbildung 3).

2.2. Wallet

Die Wallet oder Brieftasche speichert σ Public/Private-Key-Paare. Insbesondere werden dort *keine* Bitcoins gespeichert, da es keine digitale Entsprechung dafür gibt. Der Private Key wird benötigt, um eine neue Transaktion zu signieren und somit zu beweisen, dass man im Besitz der Bitcoins ist, die an die Adresse überwiesen wurden, die dem Public Key entspricht.

Verliert man ein oder mehrere Key-Paare, sind die an die zugehörige Adresse überwiesenen Bitcoins unwiderruflich verloren. Werden die Key-Paare gestohlen, ist der Dieb damit im Besitz der an diese Schlüssel übertragenen Bitcoins. Der Standard-Client bietet deswegen die Möglichkeit, die Brieftasche mit einem Passwort zu verschlüsseln und ein Backup davon anzulegen.

2.3. Block Chain

Es muss sichergestellt werden, dass dieselben Bitcoins nicht doppelt überwiesen werden. Das Bitcoin-Netzwerk verifiziert die Überweisungen durch die kumulierte Rechenleistung des gesamten Netzwerkes. Tätigt man eine Transaktion, wird diese im Netzwerk bekannt gemacht. Diese

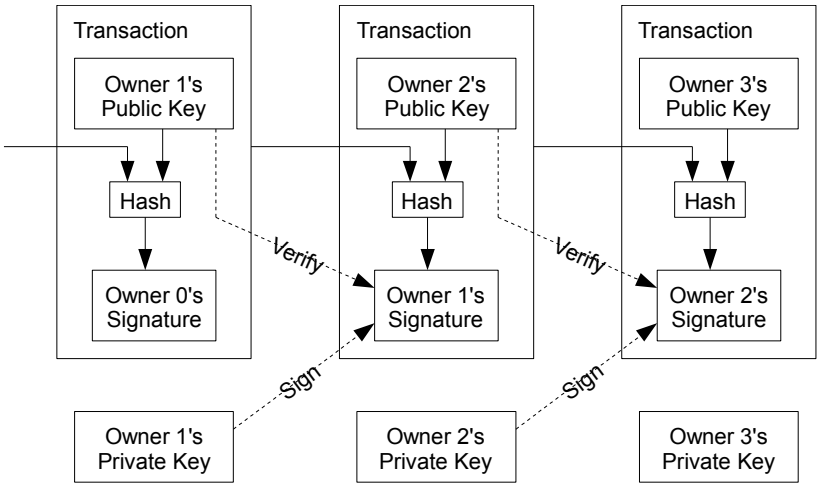

Abbildung 2: Darstellung der Verbindung von Transaktionen mit Verdeutlichung der Schlüssel [61]

Transaktionen werden nach Überprüfung ihrer Korrektheit zusammengefasst und mit einer Referenz auf einen Vorgängerblock versehen. Diese Datensammlung wird mit einer rechenintensiven kryptographischen Aufgabe verrechnet und bildet ab da zusammen mit der Lösung einen Block (siehe Abschnitt 2.4). Dieser wird im Netzwerk bekannt gemacht. Die Lösung der Aufgabe bedeutet die Verifizierung der Transaktionen, die im Block vorhanden sind. Mehrere Blöcke bilden eine Kette, die *Block Chain*. Es wird sichergestellt, dass etwa alle zehn Minuten ein Block generiert wird. Ein Block fasst somit die Transaktionen von zehn Minuten zusammen (siehe Abschnitt 3.5).

Es kann passieren, dass gleichzeitig zwei Knoten im Netzwerk die kryptographische Aufgabe für verschiedene Transaktionen lösen, sodass zwei Überweisungen akzeptiert werden, die vielleicht dasselbe Geld an andere Ziele transferieren. Die Lösung des Problems besteht darin, dass jeder Knoten (nur) die längste Block Chain speichert. Durch die Latenz des Netzwerkes verbreiten sich die verifizierten Blöcke verschieden schnell, sodass sich nach einiger Zeit nur eine einzige Transaktionskette durchsetzen wird [61].

Theoretisch ist es möglich, dass zwei oder mehr Ketten sich immer die Waage halten und somit die Transaktionen nicht eindeutig sind. Da Bitcoin aber als Underlay-Netzwerk das Internet verwendet, für das mehr oder weniger zufällige Latenzschwankungen charakteristisch sind, kommt eine solche Situation mit großer Wahrscheinlichkeit nicht vor. Dennoch gilt eine Transaktion erst als sicher, wenn sie mit einem Block verifiziert worden ist, der bereits vier Nachfolgerblöcke hat [21].

2.4. Mining

Ein Block wird mit einer kryptographischen Aufgabe verifiziert. Diese Aufgabe besteht darin, nach einem definierten Schema dem Block Bits hinzuzufügen und dann zu hashen. Entsteht dabei eine gewisse Anzahl Nullen am Anfang, ist die Aufgabe gelöst. Die Anzahl der Nullen ist variabel und steigt mit der Rechenleistung im Netzwerk. Die Anpassung ist derart, dass etwa alle zehn

Minuten eine Aufgabe gelöst wird. Eine spezielle Hashwertberechnung als Aufgabe zu verwenden, hat den Vorteil, dass das Ergebnis sehr mühsam zu finden ist, aber sehr leicht nachgeprüft werden kann. Die Art der Aufgabe wird in Abschnitt 3.5 genauer beschrieben [61].

Damit die Clients einen Anreiz haben, Rechenleistung in das Netzwerk zu investieren, werden sie für jede erfolgreich gelöste Aufgabe mit einer gewissen Summe neu geschöpfter Bitcoins belohnt. Dies passiert in Form einer Transaktion ohne Quelle mit der Adresse des Finders des Hashwertes als Ausgang. Den Versuch, Lösungen für die kryptographische Aufgabe zu finden, und dabei Bitcoins zu verdienen, nennt man Mining [61].

Die Höhe der Belohnung wird dabei über die Zeit immer geringer. Genauer wird das in Abschnitt 3.3 beschrieben.

Durch die immer geringere Belohnung wird der Anreiz, Blöcke zu erzeugen, zurückgehen. Aus diesem Grund können Miner[4] für die Blockerzeugung bezahlt werden. Das geschieht mit Transaktionsgebühren. Die Höhe dieser Gebühr kann dabei vom Ersteller der Transaktion festgelegt werden (aus der GUI[5] entnehmbar). Da die Transaktionen, die in einen Block aufgenommen werden, frei wählbar sind, kann mit einer höheren Transaktionsgebühr eine schnellere Verifizierung erreicht werden[6].

[4] Das sind die Personen, deren Clients das Mining ausführen.
[5] **G**raphical **U**ser **I**nterface, die Benutzeroberfläche des Programms, mit dem man Bitcoins überweist.
[6] Das setzt eine entsprechende Software des Miners voraus, die diese Gebühren auch prüft.

3. Merkmale

3.1. Sicherheit

Sicherheit im Bitcoin-Netzwerk hat einen sehr hohen Stellenwert, da es einerseits direkt um Geld geht, ein erfolgreicher Angriff also zumeist unmittelbar belohnt wird, anderseits aber auch ein Peer-to-Peer-Netzwerk ohne zentrale Instanz verwendet wird, die die Möglichkeit hätte, Transaktionen zu signieren o.ä.. Neben Angriffen, die Peer-to-Peer-Netzwerke in ihrer Struktur im Allgemeinen betreffen (auf die in Abschnitt 3.4.5 eingegangen wird), gibt es auch Angriffe auf das System als solches. Die Sicherheit von Bitcoin beruht dabei im Wesentlichen auf der Integrität der längsten Block Chain. Um zu verstehen, wie *sicher* das Netzwerk ist, kann man sich in die Perspektive eines Angreifers versetzen und versuchen, den Überweisungsmechanismus zu kompromittieren. Folgende Angriffsszenarien liegen nahe.

Bitcoin-Diebstahl: Es ist nicht ohne weiteres realisierbar, Bitcoins zu stehlen, da keine Bitcoins existieren. Bitcoins sind ausschließlich in Form von Transaktionen vorhanden, die wiederum mit den privaten Schlüsseln der Benutzer signiert sind. Also hat ein Dieb vorerst nur die Möglichkeit, die Signatur zu brechen (also den privaten Schlüssel zu extrahieren) oder den privaten Schlüssel zu stehlen. Nach aktuellem Stand der Forschung ist es nur mit exorbitantem Aufwand möglich, eine Signatur zu brechen [67]. Somit wird ein Angreifer eher versuchen, den Schlüssel zu stehlen. Damit liegt die Sicherheit aber in der Hand des Nutzers, der seine Brieftasche schützen muss. Der Standard-Client bietet dazu z. B. die Möglichkeit, die Brieftasche zu verschlüsseln.

Der Mechanismus bringt es mit sich, dass an Hashwerte der Public Keys überwiesen wird. Wenn wir also annehmen, dass ein Angreifer die Möglichkeit hätte, gezielt Kollisionen zu erzeugen, kann er zu einer bekannten Adresse ein neues Schlüsselpaar erzeugen und damit die zu dieser Adresse gehörenden Bitcoins überweisen. Die Schlüssel werden mit SHA-256 und RIMEMD-160 erzeugt – beides kryptographische Hashverfahren. Bis heute ist es nicht gelungen, mit vertretbarem Aufwand Kollisionen zu erzeugen [67]. Die Bitcoin-Entwickler gehen außerdem davon aus, dass der Rechenaufwand für Mining, mit dem man „legal“ Geld verdienen kann, immer sehr viel geringer sein wird als die Kollisionserzeugung [32]. Zusammengefasst kann man also sagen, dass das Protokoll selbst so sicher wie die verwendete asymmetrische Verschlüsselung ist – in diesem Fall *ECDSA* – zusammen mit den verwendeten Hashfunktionen.

Doppelte Überweisung von Bitcoins: Es ist ebenfalls nicht ohne Weiteres realisierbar, zweimal dieselben Bitcoins zu überweisen, da das gesamte Netzwerk sämtliche Überweisungen in Form der Block Chain verifiziert. Ein Angreifer hat hier prinzipiell zwei Möglichkeiten:

1. Schneller sein als der Blockmechanismus
2. Stärker sein als das Netzwerk

Die erste Möglichkeit bedeutet, zwei Überweisungen direkt hintereinander abzusetzen und zu versuchen, so schnell wie möglich reale Ware oder anderes Geld dafür zu bekommen. Die Bitcoin-Entwickler empfehlen, fünf Blöcke in der Kette abzuwarten, bevor wirklich sicher ist, dass das Geld überwiesen wurde. Die Generierung eines neuen Blocks dauert etwa zehn Minuten, damit wäre eine Überweisung erst nach 50 Minuten „sicher“. Kauft der potentielle Dieb z. B. ein Eis mit Bitcoins, wird er keine 50 Minuten darauf warten. Es wird eher so laufen, dass der Eisverkäufer

wartet, bis die Transaktion zu seinem Client propagiert wurde. Sie ist dann noch nicht verifiziert. Der Dieb kann somit dasselbe Geld eine Minute später für ein zweites Eis bei einer anderen Eisdiele ausgeben. Der Betrug wird genau dann öffentlich, wenn der nächste Block generiert wurde, in dem nur eine der Transaktion verarbeitet worden ist. Die andere Transaktion wird dann abgelehnt.

Diese Art des Angriffs ist keine Schwäche des Bitcoin-Systems, da doppelte Transaktion aufgedeckt werden. Das System ist also immer noch „sicher“, wenn man die entsprechenden fünf Blöcke abwartet, was bei den meisten Kaufvorgängen möglich sein sollte. Ghassan Karame *et. al.* [52] beschäftigen sich weiter mit dem Thema.

Die zweite Möglichkeit ist eine reale Gefahr für das Netzwerk. Der Verifizierungsmechanismus findet in Form der Blockgenerierung statt. Diese Generierung erfordert sehr viel Rechenleistung. Der Grundgedanke von Bitcoin ist dieser:

Das Netzwerk zusammen ist stärker als ein Einzelner.

In anderen Worten ist die Wahrscheinlichkeit, das ein Rechner in der Menge der gutartigen Peers die Lösung findet, höher, als das dies einem einzelnen Rechner – selbst mit mehr Rechenleistung als jeder einzelne der Peers – gelingt.

Stimmt diese Aussage nicht mehr und hat ein Angreifer mehr Rechenleistung als das gesamte sonstige Netzwerk, ist er in der Lage beliebige Blöcke zu generieren. „Beliebig“ heißt in diesem Fall, dass dort beliebige Überweisungen enthalten sein können. Außerdem ist er in der Lage, eine Überweisung x rückgängig zu machen, indem er sämtliche Blöcke ab dem Block, in dem x vorkommt, neu generiert (ohne x) bis die neu entstandene Kette wieder am längsten ist [61].

Diese Art des Angriffs in nicht vermeidbar. Die kumulierte Rechenleistung des jetziges Netzwerkes ist allerdings extrem hoch, sodass ein derartiger Angriff (momentan) unwahrscheinlich ist [35].

3.2. Anonymität

Anonymität bedeutet, dass ein Akteur in einer Struktur nicht identifiziert werden kann [16].

Bei Bitcoin bedeutet dies im Wesentlichen, dass die Währung anonym ist, wenn nicht nachgewiesen werden kann, welche (reale) Identität Überweiser und/oder Empfänger haben. Diese Zuordnung kann dabei auf zwei Weisen geschehen:

1. Zuordnung anhand der IP-Adresse
2. Zuordnung anhand des öffentlichen Schlüssels

Die Zuordnung von Transaktionen zu realen Identitäten anhand der IP-Adresse wird im Abschnitt 3.4.6 diskutiert.

Die Zuordnung des öffentlichen Schlüssels zu einer realen Identität kann mit einem gewissen Aufwand möglich sein, wie von Fergal Reid *et. al.* [40] nachgewiesen wurde.

Strukturbedingt sind alle Transaktionen öffentlich einsehbar, sodass man sie in einem Graphen Γ darstellen kann (Abbildung 3). Der Graph ist dabei so zu interpretieren, dass ein Knoten eine Transaktion darstellt. Die Eingänge des Knotens sind die Eingänge der Transaktion und die Ausgänge des Knotens die Ausgänge der Transaktion.

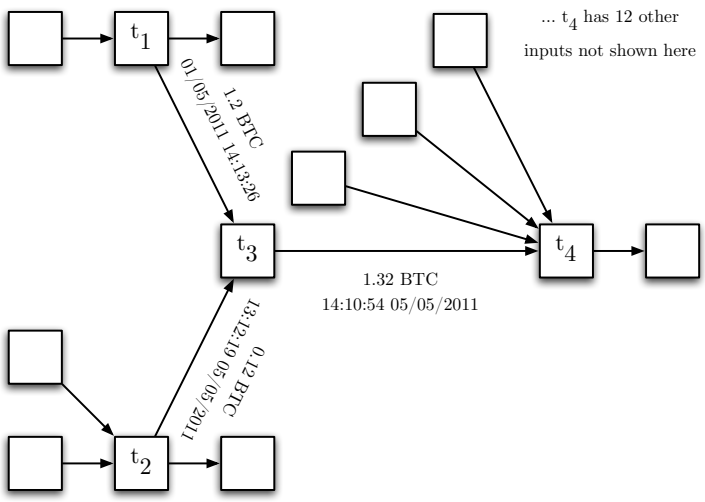

Abbildung 3: Graphische Darstellung aufeinanderfolgender Transaktion [40]

Die Schwierigkeit liegt nun darin, die Transaktionen und die dort enthaltenen Schlüssel Benutzern zuzuordnen. Der Aufbau der Transaktion hilft aber hierbei.

Eine Transaktion hat mehrere Eingänge, die mit verschiedenen Schlüsseln signiert sein können [15]. Graphisch ist das durch eine Überlagerung des Transaktionsgraphen darstellbar. Da der Besitzer aller dieser Schlüssel dieselbe Person sein muss, kann man mehrere Schlüsselpaare und damit ggf. mehrere Transaktionen einer Person (oder Instanz) zuordnen. Die Zuordnung von Schlüsselpaaren zu einer Person resultiert ebenfalls in der Zuordnung der öffentlichen Adressen zu dieser Person. Die Informationen kann man in einem Benutzergraphen zusammentragen (Abbildung 5). Fergal Reid und Martin Harrigan [40] haben weiterhin versucht, die Benutzer IP-Adressen oder realen Personen zuzuordnen. Dabei halfen unter anderem öffentlich einsehbare Informationen. So konnten über eine einfache Internetsuche öffentliche Schlüssel Forenbenutzern zugeordnet werden oder über eine Seite, die Bitcoins spendet und dabei IP-Adressen der Empfänger veröffentlicht, Zuordnungen von IP-Adressen zu Schlüsseln getroffen werden. Je nachdem, wie unvorsichtig ein Benutzer mit seinen Schlüsseln umgeht, können so schnell Daten, die nichts direkt mit Bitcoin zu tun haben, Schlüsseln zugeordnet werden, gerade größere Organisationen werden in diesem Bereich mehr Möglichkeiten haben.

3.2.1. Alternativen/Erweiterungen

Mehrfache Überweisungen: Um die Anonymität zu wahren, wurde vorgeschlagen, Geld mehrfach an sich selbst zu überweisen [75]. Das kann aber dann zum Problem werden, sobald man seine falschen Überweisungen zusätzlich zu einer Richtigen als Eingänge einer Transaktion an einen anderen nehmen muss. Es wird also komplizierter, aber nicht unmöglich den User-Graphen zu konstruieren.

Mixer: Als weitere Lösung wurde vorgeschlagen, *Mixer* einzuführen, die Überweisungen von verschiedenen Personen entgegennehmen und vermischen. Bestenfalls würde das Geld sogar durch

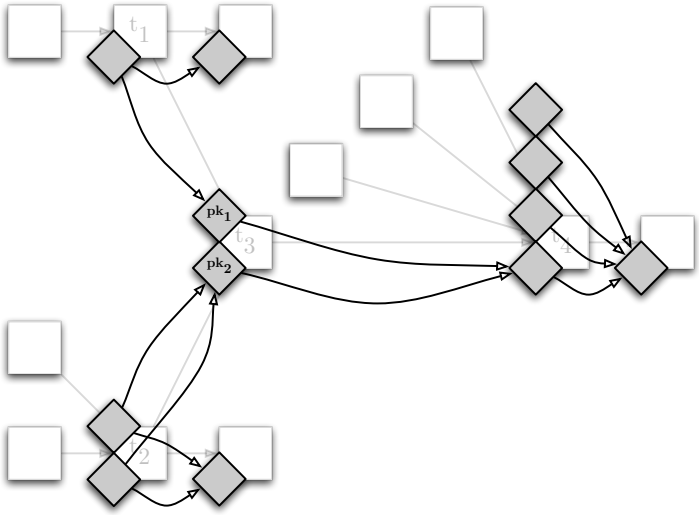

Abbildung 4: Graphische Darstellung der Schlüsselpaare, die zum Signieren genutzt wurden. Transaktion t_3 hat als Eingänge z. B. die Transaktion die mit pk_1 und pk_2 signiert wurden [40].

mehrere dieser Mixer laufen und erst danach wieder zurück an den Eigentümer fließen. Problematisch daran ist die fehlende Kontrolle über die Mixer, da diese keine Datenbank o. ä. führen können. Der Anonymitätsgedanke wäre ansonsten wieder hinfällig. Ein Benutzer müsste also darauf vertrauen, sein Geld irgendwann wieder zurückzuerhalten [75].

Zerocoins: Im Gegensatz zu den beiden vorherigen Erweiterungen, existiert eine Protokollerweiterung (die bislang aber nicht zum Einsatz kommt) mit dem Namen Zerocoin, die sehr vielversprechend ist. Der Grundgedanke hierbei ist die Einführung von zusätzlichen Zerocoins, die zwar nicht zum Zahlen genutzt werden, deren Weg aber dafür im Netzwerk vollständig anonym ist. Zerocoin benötigt hierzu einen anonymen Pool und basiert auf einem kenntnisfreien Beweis[7] [59].

Ein kenntnisfreier Beweis zeichnet sich dadurch aus, dass ein Beweiser gegenüber einem Verifizierer mit einer hohen Wahrscheinlichkeit beweisen kann, ein bestimmtes Wissen zu besitzen, ohne das Wissen selbst preiszugeben. Außerdem darf eine dritte Partei von außen nicht feststellen können, ob der Beweiser das Wissen besitzt. Sei folgender Fall gegeben: Der Beweiser B will gegenüber einem Verifizierer V nachweisen, dass $x \in L$. L sei dabei eine formale Sprache. B und V kommunizieren dazu, während M die beiden von außen beobachtet. Dann gilt für einen kenntnisfreien Beweis [36]:

- Vollständigkeit: Wenn $x \in L$, akzeptiert V (fast) immer.
- Zuverlässigkeit: Wenn $x \notin L$, akzeptiert V (fast) immer nicht.
- Zero-Knowledge-Eigenschaft: M kann zu keinem Zeitpunkt feststellen, ob einerseits $x \in L$ und anderseits, ob B weiß, dass $x \in L \vee x \notin L$.

[7] Auch Zero-Knowledge-Beweis

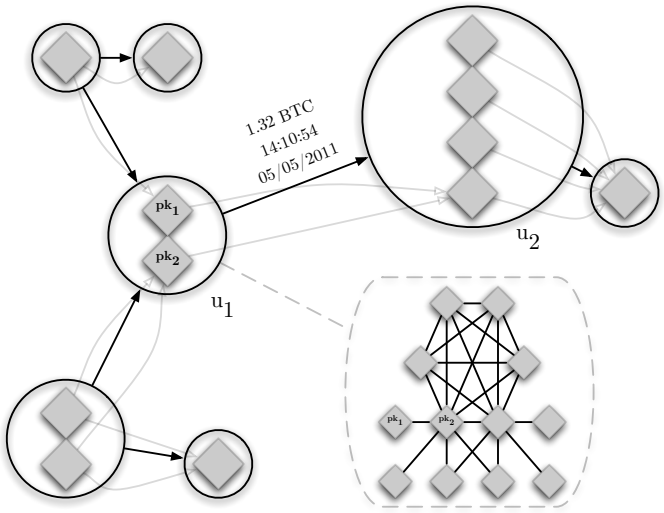

Abbildung 5: Graphische Darstellung der Benutzer, die Transaktionen getätigt haben. Im Hintergrund ist der Graph mit den einzelnen Schlüsseln als Sender und Empfänger zu sehen [40].

Das Prinzip von Zerocoins kann unabhängig von Bitcoins formuliert werden. Man stelle sich einen Pool[8] vor, auf den alle Nutzer Zugriff haben. Alice will jetzt einen Euro[9] durch Zerocoins anonymisieren. Sie erzeugt dazu zuerst eine eindeutige zufällige Zahl S, die fortan als Name für den Euro fungiert. Dann erzeugt sie daraus über ein kryptographisch sicheres Verfahren den Zerocoin C und zwar so, dass S nur mithilfe einer Zahl r aus C errechnet werden kann und vice versa. Der Wert von r hängt dabei vom gewählten Verfahren ab. C wird dann zusammen mit der Information, dass C einen Euro wert ist, in den Pool gelegt. Sämtliche Nutzer prüfen daraufhin, ob C valide ist und verrechnen seinen Wert.

Um ihren Euro wiederzuerhalten, sammelt Alice zuerst die Menge an Zerocoins $\{C_1, \ldots, C_N\}$, die bis dahin von allen Nutzer in den Pool gelegt wurde und erzeugt dann einen kenntnisfreien Beweis π, der beweist, dass sie einerseits ein $C \in \{C_1, \ldots, C_N\}$ kennt und andererseits den geheimen Wert r kennt, sodass daraus S entsteht. Alice anonymisiert sich daraufhin und gibt das Paar (S, π) den anderen Nutzern bekannt. Diese verifizieren π (also die Korrektheit des Beweises), dass S nicht bereits abgeholt wurde und erlauben der anonymisierten Alice dann, einen Euro irgendwo aus dem Pool zu nehmen.

Dieses System kann mit Bitcoin verbunden werden. Die Block Chain dient dabei als Pool, da diese nicht kompromittiert werden kann. Die Einbettung erfolgt dabei folgendermaßen: Um einen Zerocoin in den Pool zu legen, erzeugt Alice eine Transaktion mit dem Zerocoin C als Ausgang. Das zugehörige S[10] und r hält sie geheim. Um im Folgenden getarnt den Wert dieses Zerocoins wiederzuerhalten, erzeugt sie eine neue Bitcoin-Adresse, erzeugt dann mithilfe von r und S den Beweis π und eine neue Transaktion mit dem Paar (S, π) als Eingang und ihrer neuen Bitcoin-Adresse als Ausgang. Da alle Transaktionen zuerst in einem Block integriert werden müssen, kann sowohl die Transaktion, die C einbettet, als auch die, die (S, π) als Eingang hat, auf ihre

[8] Also etwas, in das man Daten hereinlegen und herausholen kann.
[9] Die Währung ist willkürlich gewählt.
[10] Das S ist die ID für die in der Transaktion als Eingang vorkommenden Bitcoins.

Korrektheit geprüft werden. Insbesondere darf S nicht schon einmal verwendet worden sein, muss aber in Form von C im Pool enthalten sein. Nur wenn die Transaktionen korrekt sind, werden sie in einen Block und somit in die Block Chain integriert und somit gültig. Das Besondere dieser Vorgehensweise ist, dass die Korrektheit zwar geprüft werden kann, die Verbindung von S zu C aber unbekannt bleibt. Die Integration in Bitcoin erfordert eine Protokollerweiterung [59].

Paysafecard: Zu guter Letzt sei hier noch auf Paysafecard als alternatives anonymes Zahlungsmittel hingewiesen, das aber ansonsten keine Verbindung zu Bitcoin hat. Paysafecard ist ein Angebot der Prepaid Services Company Ltd. [57] und bietet Prepaidkarten mit einer PIN in diversen Geschäften zum Verkauf an. Man kann diese gegen Bargeld erwerben und mit der PIN dann bei einem Händler zahlen, der diese Zahlungsmethode akzeptiert. Das Unternehmen verdient an dem Angebot durch Gebühren für den Händler und Gebühren bei Karten die älter als 12 Monate sind [27,57]. Durch die zentrale Kontrolle unterscheidet es sich ansonsten grundlegend von Bitcoin. So ist das Unternehmen z. B. leicht in der Lage, Zahlungen an bestimmte Nutzergruppen und Unternehmenstypen zu verweigern, wie es unlängst mit VPN Providern[11] passierte [54].

3.3. Festgelegte Geldmenge

Der Geldbetrag, der beim Mining erzeugt wird, wird alle 210000 Blöcke halbiert [23]. Da alle zehn Minuten ein Block generiert wird, entspricht das $\frac{210000 \cdot 10}{60 \cdot 24 \cdot 356} \approx 4{,}1$ Jahren. Im Moment liegt der Betrag bei 25 BTC, anfangs lag er bei 50 BTC [18]. Ein Bitcoin kann in max. 100 Millionen Einheiten[12] zerlegt werden [29]. Die Anzahl der Bitcoins wird 21 Millionen nie übersteigen. Man kann die Anzahl folgendermaßen berechnen:

$$\lim_{n\to\infty} \sum_{i=0}^{n} \left(210000 \cdot \frac{50}{2^i} \right) = 210000 \cdot 50 \cdot \lim_{n\to\infty} \sum_{i=0}^{n} \frac{1}{2^i} = 210000 \cdot 50 \cdot 2 = 21000000$$

In der Formel strebt n gegen unendlich und somit wird $\frac{50}{2^i}$ immer kleiner. Durch die Diskretisierung der Bitcoins ist dies allerdings nicht möglich, somit wird die Anzahl von 21 Millionen nie genau existieren. Da der Bruch nicht unendlich klein werden kann und somit die Summe auch nicht unendlich lang ist, wird die Gesamtanzahl in endlicher Zeit erreicht.

3.4. Netzstruktur

Bitcoin ist ein Peer-to-Peer-Netzwerk (auch P2P-Netzwerk). Das bedeutet, dass im Gegensatz zum klassischen Client-Server-Modell nur eine gleichberechtigte Menge von Peers existiert.

Beim Client-Server-Modell ist der Server ein Computer oder Programm, das Daten oder Ergebnisse von Berechnungen an den Client ausliefert, die dieser angefragt hat. In einem P2P-Netzwerk stellen alle Peers von sich aus Anfragen zu Daten, liefern aber auch Antworten auf solche, falls ein anderer diese stellt. Das resultiert in einer verteilten Struktur ohne zentrale Instanz.

P2P-Netzwerke haben für gewöhnlich diese Anforderungen [66]:

[11] VPN (Virtual Private Network) bezeichnet ein virtuelles Netzwerk, das über einen Tunnel durch ein fremdes Netzwerk zwei andere verbinden kann, wird oft zur Anonymisierung verwendet.

[12] Diese Einheit hat man zu Ehren des Erfinders „Satoshi“ getauft [29].

- Dynamik: Clients können ohne Informationsverlust oder sonstige Folgen kommen und gehen.
- Gleichberechtigte Peers: Es gibt keine zentrale Instanz.
- Vollständiges Routing: jeder Peer ist in der Lage, jedes Datum im Netzwerk zu finden.
- Schnelligkeit: Die Bandbreite sollte voll ausgenutzt werden.

Außerdem gibt es unter anderem folgende „Stellschrauben“, über die optimiert werden kann:

- Grad: Mächtigkeit der Nachbarschaftsmenge, anders gesagt die Anzahl der bekannten Peers pro Peer
- Durchmesser: Längster Weg durch das Netzwerk, abhängig vom Grad
- Methode des Routings: Welche Peers kennt ein Peer? Sind diese latenzarm im Vergleich zum unterliegenden Netzwerk?

Zu guter Letzt gibt es noch Anforderungen hinsichtlich der Anonymität:

- Der Absender/Ersteller eines Datums ist anonym.
- Die Route ist nicht nachvollziehbar.
- Der Empfänger/Anfrager ist anonym.

und Sicherheit:

- Große Teile des Netzwerkes können wegbrechen.
- Angriffe werden erschwert oder abgewehrt, zu nennen sind hierbei insbesondere (siehe auch Abschnitt 3.4.5):
  - Der Sybil-Angriff
  - Das Problem der byzantinischen Generäle

Das Bitcoin-Netzwerk hat im Speziellen jedoch vor allem diese Anforderungen und Eigenschaften:

- Es werden im Vergleich zu anderen Netzwerken marginale Datenmengen übertragen (zumindest hatte es den Anschein).
- Die Integrität der Block Chain muss gewährleistet werden.

Um diesen Anforderungen zu genügen, wählt Bitcoin eine sehr einfache Methode: Jeder Peer speichert alles und kennt jeden. Nachrichten werden darüber hinaus nicht intelligent geroutet, sondern per *Flooding* verteilt – in anderen Worten, an jeden bekannten Peer weitergegeben.

Die von Bitcoin benutzten Methoden werden im weiteren Verlauf genauer vorgestellt und danach mit anderen Netzwerken verglichen.

Die nachfolgenden Ausführungen beziehen sich auf den Standard-Client *Bitcoin-QT*. Andere Programme können in der Art der IP-Adressbeschaffung eventuell abweichen.

3.4.1. Verbindungsaufbau

Startet man den Client, so nimmt dieser automatisch die Verbindung mit dem Netzwerk auf. Die Adressbeschaffung läuft dabei standardmäßig über DNS-Pools[13] ab. Namentlich sind in der jetzigen Version folgende Pools eingetragen [13]:

```
static const char *strMainNetDNSSeed[][2] = {
    {"bitcoin.sipa.be", "seed.bitcoin.sipa.be"},
    {"bluematt.me", "dnsseed.bluematt.me"},
    {"dashjr.org", "dnsseed.bitcoin.dashjr.org"},
    {"xf2.org", "bitseed.xf2.org"},
    {NULL, NULL}
};
```

Des Weiteren hat der Client die Möglichkeit Adressen über IRC[14] zu beschaffen oder fest eingebaute Adressen zu benutzen. Sobald ein anderer Peer gefunden wurde, kann auch dieser nach anderen Adressen gefragt werden [26].

3.4.2. Kommunikationsprotokoll

Bitcoin-Clients kommunizieren über 16 Nachrichten miteinander. Die Nachrichten werden alle in ein Paket mit einem 24 Byte Header gekapselt. Der Header besteht dabei aus [28]:

Feldgröße	Beschreibung	Datentyp	Kommentar
4	magic	uint32_t	Kodiert das Netzwerk, in dem die Nachricht verschickt wird
12	command	char[12]	Der Nachrichtentyp, der mit einem ASCII-String kodiert ist
4	length	uint32_t	Länge des Paketinhalts (payload) in Bytes
4	checksum	uint32_t	Die ersten vier Bytes von $sha265(sha256(payload))$
variabel	payload	uchar[]	die eigentlichen Daten

Tabelle 1: Aufbau des Headers einer Bitcoinnachricht

magic dient dazu, das Netzwerk zu spezifizieren. Neben dem eigentlich Hauptnetzwerk *main* gibt es noch zwei Testnetze *testnet* und *testnet3*, in denen von Entwicklern auf einer alternativen Block Chain Protokolländerungen getestet werden können, und das „Bitcoinderivat“ *Namecoin*, das einen DNS-Service mit derselben Technik wie Bitcoin zur Verfügung stellt [25].

[13] Das Domain Name System löst menschenlesbare URLs in IP-Adressen auf.
[14] Internet Relay Chat, ein Chatprotokoll

command beschreibt die 16 Nachrichten. Diese sind im einzelnen [26]:

version	Informationen über die Programmversion und die heruntergeladenen Blöcke. Diese Nachricht wird gesendet, wenn sich der Client mit einem anderen verbinden will.
verack	Die Bestätigung auf eine *version*-Nachricht eines anderen Clients. Mit der Bestätigung gibt der Client zu verstehen, dass er die Verbindung zulässt.
addr	Liste von IP-Adressen und Ports. Antwort auf eine *getaddr*-Anfrage.
inv	Propagieren von Wissen über Objekte (Inventar). Kann benutzt werden, um neue Transaktion bekanntzugeben.
getdata	Anforderung für eine Transaktion. Antwort auf eine *inv*-Nachricht.
notfound	Angeforderte Daten sind nicht vorhanden. Mögliche Antwort auf eine *getdata*-Nachricht.
getblocks	Fordert eine *inv*-Nachricht für alle Blöcke in einer gewissen Spannweite an.
tx	Sendet eine Transaktion. Antwort auf eine *getdata*-Anfrage.
block	Sendet einen Block. Antwort auf eine *getdata*-Anfrage.
header	Sendet bis zu 2000 Block-Header, falls nur diese statt den kompletten Blöcken gespeichert werden sollen.
getaddr	Fordert eine Liste von Adressen von Peers an.
submitorder, *checkorder*, *reply*	Funktionen zum Übermitteln von IP Transaktionen. Das sind Transaktionen, die statt einer Bitcoin-Adresse als Ausgang eine IP-Adresse haben.
alert	Sendet eine Nachricht durch das Netzwerk. Enthält eine Signatur, mit der Nachrichten vom Bitcoin-Entwicklerteam erkannt werden können und dem Nutzer angezeigt werden.
ping	Überprüft, ob die Verbindung zu einem Peer noch valide ist.

Tabelle 2: Mögliche Werte für *command*

Die Kommunikation lässt sich in 3 Teile unterteilen:

1. Nachbarschaftsmenge beschaffen und aktuell halten
2. Block Chain aktualisieren (beinhaltet das Beschaffen der neuen Blöcke)
3. Transaktion tätigen

Nachbarschaftsmenge beschaffen und aktuell halten. In diesem Schritt wird per oben genanntem Bootstrapping (Abschnitt 3.4.1) außerhalb des eigentlichen Protokolls eine Nachbarschaftsmenge aufgebaut. Jeder Nachbar besitzt einen Zeitstempel, der den Zeitpunkt des letzten Lebenszeichens angibt (*nLastSend*) [12]. Jedesmal wenn eine Nachricht des Peers empfangen wird,

wird der Zeitstempel entsprechend angepasst. Falls länger als 30 Minuten keine Kommunikation mehr stattfindet, sendet der Client eine *ping*-Nachricht an den Nachbarn [12]:

```
// Keep-alive ping. We send a nonce of zero because we don't use it anywhere
// right now.
if (pto->nLastSend && GetTime() - pto->nLastSend > 30 * 60 && pto->vSendMsg.empty()) {
    uint64 nonce = 0;
    ...
    pto->PushMessage("ping", nonce);
    ...
}
```

Eine *ping*-Nachricht wird mit *pong* beantwortet [12]:

```
else if (strCommand == "ping")
{
    ...
    uint64 nonce = 0;
    vRecv >> nonce;
    ...
    pfrom->PushMessage("pong", nonce);
    ...
}
```

Nach 90 Minuten, in denen keine Kommunikation stattgefunden hat, wird der Peer aus der Nachbarschaftsmenge entfernt [26].

Block Chain aktualisieren. Jeder Teilnehmer des Netzwerkes besitzt die komplette Block Chain oder zumindest die Header dieser. Der Client kümmert sich selbst darum, dass seine Block Chain aktuell bleibt. Dazu sendet er eine *getblocks*-Nachricht an seine Nachbarn [12]:

```
// Start block sync
if (pto->fStartSync && !fImporting && !fReindex) {
    pto->fStartSync = false;
    PushGetBlocks(pto, pindexBest, uint256(0));
}
```

Die Arbeit wird dabei an die Funktion PushGetBlocks delegiert, die nach einiger Vorarbeit die Nachricht sendet [14]:

```
void PushGetBlocks(CNode* pnode, CBlockIndex* pindexBegin, uint256 hashEnd)
{
    // Filter out duplicate requests
    if (pindexBegin == pnode->pindexLastGetBlocksBegin && hashEnd == pnode->
     hashLastGetBlocksEnd)
        return;
    pnode->pindexLastGetBlocksBegin = pindexBegin;
    pnode->hashLastGetBlocksEnd = hashEnd;

    pnode->PushMessage("getblocks", CBlockLocator(pindexBegin), hashEnd);
}
```

Eine *getblocks*-Nachricht wird mit einer *inv*-Nachricht beantwortet, in der die neuen Blöcke gesendet werden [12]:

```
else if (strCommand == "getblocks")
{
    CBlockLocator locator;
    ...
    // Find the last block the caller has in the main chain
    CBlockIndex* pindex = locator.GetBlockIndex();

    // Send the rest of the chain
    if (pindex)
        pindex = pindex->GetNextInMainChain();
    ...
    for (; pindex; pindex = pindex->GetNextInMainChain())
    {
        ...
        pfrom->PushInventory(CInv(MSG_BLOCK, pindex->GetBlockHash()));
        ...
    }
}
```

`PushInventory` sendet dabei die *inv*-Nachricht.

Transaktion tätigen. Eine neue Transaktion wird zunächst in die Wallet eingetragen und in unregelmäßigen Abständen ins Netzwerk geschickt. Das geschieht, um die Anonymität zu wahren (siehe Abschnitt 3.4.6). Das Senden für sich geschieht wieder mit der *inv*-Nachricht. Die Wallet ruft dazu die Methode `RelayTransaction` auf, die wiederum `PushInventory` für jeden Node ausführt [13]:

```
void RelayTransaction(const CTransaction& tx, const uint256& hash, const CDataStream& ss)
{
    ...
    BOOST_FOREACH(CNode* pnode, vNodes)
    {
        ...
        pnode->PushInventory(inv);
        ...
    }
}
```

Diese Nachricht enthält dabei keine Transaktionen, sondern bringt die anderen Peers dazu, eine *getdata*-Nachricht zu senden, die dann mit einer *tx*-Nachricht beantwortet wird [28].

Nach dem Empfang wird die Transaktion in das bestehende Inventar eingefügt, sofern sie valide ist [12]:

```
else if (strCommand == "tx")
{
    ...
    CTransaction tx;
    vRecv >> tx;

    CInv inv(MSG_TX, tx.GetHash());
    pfrom->AddInventoryKnown(inv);
    ...
}
```

Das Einfügen in das bestehende Inventar bedeutet insbesondere, dass die Transaktion zusammen mit den anderen weiterpropagiert wird.

3.4.3. Alternativen

Das Bitcoin-Netzwerk ist nur ein Peer-To-Peer-Netzwerk von vielen. Viele dieser Netzwerke sind zu einem speziellen Zweck oder mehreren entwickelt worden. Die dabei verwendeten Routingalgorithmen sind oft um ein Vielfaches komplexer.

In diesem Abschnitt werde ich einige dieser Netzwerke nennen und im weiteren Verlauf diskutieren, ob ein ähnliches Vorgehen bei Bitcoin sinnvoll ist.

Pastry ist ein P2P-Netzwerk, das auf der Idee des Routing-Verfahrens von Plaxton, Rajaraman und Richa aufbaut. Dieses Routing-Verfahren beschreibt ein effizientes verteiltes Verfahren um in einem statischen Netzwerk auf Datenkopien zuzugreifen. Eng verwandt mit Pastry ist Tapestry, das auf demselben Verfahren beruht, allerdings weniger heuristische Methoden benutzt [66].

Pastry benutzt verteilte Hashtabellen (DHT) zur Zuordnung zwischen Daten und Peers. Sei $P = \{p_1, \ldots, p_n\}$ die Menge an Peers und $X = \{x_1, \ldots, x_m\}$ die Menge der Daten im Netzwerk, dann existieren zwei kryptographische Hashfunktionen mit folgender Eigenschaft:

$$h_P : P \to \mathbb{Z}_{2^m}$$

$$h_X : X \to \mathbb{Z}_{2^m}$$

Sowohl jeder Peer als auch jedes Datum erhält damit eine Zahl im gleichen Zahlenbereich. m wird groß genug gewählt um Kollisionen zu vermeiden. Man kann die Peers graphisch als Ring darstellen (siehe auch Abbildung 6). Jeder Peer ist für alle Daten zuständig, die ihm numerisch am nächsten sind, bei dem also der Differenz zwischen $h_P(p_i)$ und $h_X(x_j)$ am kleinsten ist.

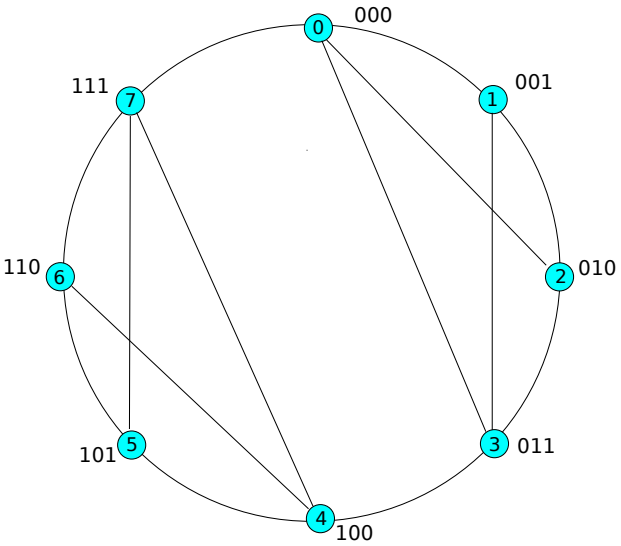

Abbildung 6: graphische Darstellung von acht Knoten auf einer eindimensionalen Ringstruktur, hier gilt: $m = 3$. Außerdem sind Knoten mit demselben Präfix verbunden.

Jeder Peer verwendet zum Routen von Nachrichten drei Tabellen, die Routingtabelle R, das Leaf-Set L und eine Nachbarschaftsmenge M.

R wird dabei so aufgebaut, dass Einträge gesucht werden, deren Präfix möglichst weit mit dem eigenen übereinstimmt. Das Besondere an der Tabelle ist, dass im Falle von mehreren Möglichkeiten diejenige gewählt wird, deren Latenz im unterliegenden Netzwerk am geringsten ist.

Für $|L| = l$ werden in L die nächsten $\frac{l}{2}$ höheren IDs und die nächsten $\frac{l}{2}$ tieferen IDs gespeichert. Der nächstkleinere Knoten des Knotens mit der kleinsten ID ist der Knoten mit der höchsten ID. Durch L wird die Ringstruktur von Pastry aufgebaut.

M besteht aus $|M| = 2^b$ Verweisen auf Peers, deren Latenz gegenüber dem Peer selbst am geringsten ist. M wird dabei normalerweise nicht zum Routen verwendet, sondern eher im Problemfall (z. B. Reparaturarbeiten).

Der Routingalgorithmus selbst ist wie folgt aufgebaut (ich gebe hier nur eine vereinfachte Version an, insbesondere treten Ausnahmefälle nicht auf). Gesucht ist ein Zieldatum und die Suche beginnt an Peer p:

```
1  id_Z = h_X(Zieldatum)
2  Suche(id_Z) {
3      if (L_{-l/2} ≤ id_Z ≤ L_{l/2}) {
4          Leite an den Peer p' ∈ L weiter, für den |id_Z − p'| minimal.
5      } else {
6          p' = np(p, id_Z);
7          Leite weiter an Peer p'.
8      }
   }
10
   np(p_old, id){
12     return den Peer p_new, bei dem das gemeinsame Präfix zu id um eins größer ist als zu p_old.
13 }
```

$L_{\pm l/2}$ bezeichnen die äußersten Peers in L. Um den nächsten Peer zu finden, wird somit zuerst in L geschaut, ob der Peer zur direkten Nachbarschaft gehört und ansonsten an den Peer geleitet, dessen Präfix um eins besser mit dem Zieldatum zusammenpasst. Um $\mathrm{np}(p_{\mathrm{old}}, id)$ zu berechnen, muss ausschließlich die gemeinsame Länge des Präfixes zwischen p_{old} und id ausgerechnet werden. Um danach p_{new} zu finden, reicht ein einfaches Nachschauen in R.

Dieser Routingalgorithmus braucht höchstens $O(n/l)$ Schritte und im Erwartungswert $O\left(\frac{\log n}{b}\right)$ Nachrichten (für einen Beweis sei auf [66] verwiesen). n bezeichnet dabei die Menge an Peers und b die Basis der IDs anhand der R aufgebaut wird. l und b sind willkürliche Parameter und werden heuristisch berechnet.

Außerdem existieren Algorithmen zum Einfügen neuer Peers und für Reparaturen im Netzwerk, auf die hier aber nicht weiter eingegangen wird.

Das besondere Merkmal von Pastry ist die Weiterleitung der Nachrichten an möglichst latenznahe Peers, wobei sich die Latenz auf das unterliegende Netzwerk bezieht. Viele andere Peer-to-Peer-Netzwerke setzen dieses Netzwerk nur als gegeben voraus, beachten es aber nicht weiter.

Das Bitcoin-Netzwerk geht ebenfalls davon aus, dass ein Netzwerk vorhanden ist, optimiert aber seine Routingtabellen in keiner Weise in Bezug auf Latenz [12]. Ein Routing-Mechanismus wie bei Pastry wäre für Bitcoin zu kompliziert, da jeder Peer alles speichert, die einzigen Anfragen also die für neue Blöcke sind. Eine latenzbasierte Optimierung der Nachbarschaftsmenge wäre aber ein Ansatz, der in der Praxis wahrscheinlich einen Geschwindigkeitszuwachs bringen würde.

P-Grid wurde 2001 von Karl Aberer entwickelt. Es benutzt statt einer Zuordnung über eine Hashfunktion einen binären Suchbaum, sodass auf Daten über einen Index zugegriffen werden kann [66].

P-Grid verwendet die Huffman-Kodierung, um den Index auf den Baum abzubilden. Die Schwierigkeit besteht nun darin, den Baum auf die Peers abzubilden. P-Grid trifft dazu diese Annahmen:

1. Ein Peer liegt niemals auf dem Pfad eines anderen Peers.
2. Der Pfad eines Datums im Index des Baumes ist immer länger als der Pfad zum Peer, der dieses Datum speichert, anders gesagt: Peer und Datum haben dasselbe Präfix, das Datum hat allerdings den längeren Index. Das kann durch Anhängen von Kontrollinformationen an den Index erreicht werden, z. B. Erzeugungsdatum usw..
3. Jeder Pfad im Baum besitzt mindestens einen Peer.

Abbildung 7 verdeutlicht die Zuordnung von Peers und Daten.

Die Suche nach einem Datum passiert dann – ähnlich wie in Pastry – über den längsten Präfix, wobei dieser Algorithmus bei einem binären Suchbaum besonders einfach ist. Wenn n die Anzahl der Peers ist und Peers und Daten proportional in jedem Teilbaum verteilt sind, beträgt die Anzahl der Schritte $O(\log n)$.

Sehr viel Mühe wird bei P-Grid darauf verwendet, den Baum auf die Peers zu balancieren und umzustrukturieren. Die Algorithmen werden hier aber nicht näher beschrieben. In P-Grid ist außerdem noch ein Vertrauensmodell implementiert, das bösartigen Peers vorbeugen soll.

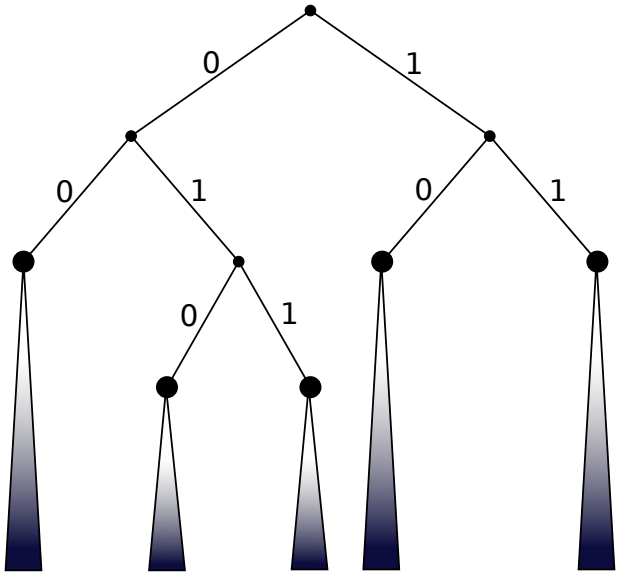

Abbildung 7: graphische Darstellung von P-Grid. Die dickeren Knoten stehen dabei für Peers, die die Daten in den Dreiecken verwalten. Die Dreiecke stehen für eine Menge von Pfaden.

P-Grid zeichnet sich durch die sehr schnelle Suche ähnlicher Daten (also deren Index sich nur um wenige Zeichen unterscheidet) in einem Netzwerk aus.

Eine Struktur wie die von P-Grid wäre in Bitcoin sinnvoll, wenn Daten zum einen nur verteilt gespeichert wären und zum anderen auch noch durchsucht werden müssten. Bei Bitcoin trifft allerdings zu, dass jeder Peer bereits alle Daten besitzt, also lokal mit eigenen Strukturen durchsuchen kann. Dadurch ist eine Struktur wie die von P-Grid für Bitcoin nicht sinnvoll.

Der Fall wäre anders, wenn nicht mehr alle Daten von jeden Peer gespeichert würden. Das wäre zum einen der Fall, wenn nur noch die Header der Blöcke gespeichert würden und zum anderen, wenn das gesamte Netzwerk umstrukturiert würde. Bereits Nakamoto [61] beschreibt aber eine Schwächung des Netzwerkes, wenn nur die Header gespeichert würden, da ein Großteil der Sicherheit auf der gemeinsamen Block Chain beruht. Eine entsprechende Umstrukturierung würde noch weiterreichen, das Problem also verstärken.

Tor wird nicht als Peer-to-Peer-Netzwerk wahrgenommen, besitzt aber ähnliche Eigenschaften. Es wurde ursprünglich vom U.S. Naval Research Laboratory entwickelt und setzt das Prinzip des Onion-Routings um [77]. Diese Art Routing soll die oben erwähnten Anonymitätsaspekte abdecken.

Onion-Routing verschlüsselt eine Nachricht vor dem Routen in mehreren Schichten[15]. Das Netzwerk besteht dabei auch aus Peers, die jeweils einen Public Key bereitstellen. Der Peer, der die Nachricht versenden will, sucht sich eine Route durch das Netzwerk aus und verschlüsselt daraufhin die Nachricht mit dem Key des letzten Knotens, dann fügt er die Adresse des letzten Knotens hinzu und verschlüsselt mit dem Key des vorletzten Knotens usw..

[15] Ähnlich wie bei einer Zwiebel, daher der Name.

Jeder Knoten der eine Nachricht bekommt, kann diese entschlüsseln und findet darin eine wiederum verschlüsselte Nachricht vor und eine Adresse. Die verschlüsselte Nachricht sendet er dann an den Knoten mit der Adresse. Das führt dazu, dass jeder Knoten nur den Vorgänger und Nachfolger kennt. Einzig der letzte Peer hat erweitertes Wissen, da dieser auch die Nachricht im Klartext lesen kann. Bei Tor heißen diese Peers *Exit-Nodes*.

Tor leitet eine Nachricht über mehrere Knoten weiter, bevor diese das Netzwerk verlässt. Das soll bei einem hinreichend großen Netzwerk sicherstellen, dass selbst bei kompromittierten Peers, immer noch über so viele Peers geleitet wird, dass die Route für einen Netzwerkteilnehmer nicht nachvollziehbar ist [10].

Eine Integration von Tor in Bitcoin oder die Verwendung von Tor in Bitcoin ist sinnvoll und wird diskutiert [2]. Die momentanen Schwachstellen, denen Tor entgegenwirken kann, werden in Abschnitt 3.4.6 diskutiert.

3.4.4. Laufzeitkomplexität und Geschwindigkeit

Bei Bitcoin ist die Menge der Nachbarn standardmäßig auf 125 festgelegt [13]. Man kann dem Client jedoch einen (Integer-)Parameter mitgeben, der diese Anzahl verkleinert oder vergrößert. Ich vernachlässige hier diese Fälle, sodass man sagen kann: Das Netzwerk hat den Grad $d = 125$. Daraus resultiert ein Durchmesser h von minimal:

$$h \geq \frac{\log n}{\log d} = k \cdot \log n \qquad \text{mit } k = \frac{1}{\log 125}$$

n ist dabei die Anzahl der Netzwerkknoten [66].

Die maximale Anzahl Hops beträgt $n-1$ (z. B. die Propagierung einer Transaktion). Jeder Knoten kann mit maximal 125 anderen verbunden sein, es sind aber auch nur zwei Verbindungen möglich. Sei also die Nachbarschaftsmenge gerade so, dass eine Kette aus n Knoten besteht. Dann ist der Weg vom ersten zum letzten Knoten gerade $n - 1$.

Die Block Chain hat momentan eine Größe von 12 GB[16]. Beim ersten Starten eines Clients dauert der Download durch das unoptimierte Netz unverhältnismäßig lange. Die Bitcoin-Entwickler reagieren darauf, indem das Protokoll die Möglichkeit bietet, nur die Header der Transaktionen herunterzuladen, damit kann die Block Chain auf Korrektheit überprüft werden, es werden aber nicht die gesamten Daten benötigt. Dieser Ansatz macht das Netzwerk aber von Peers abhängig, die die gesamte Block Chain speichern.

3.4.5. Bösartiges Verhalten

Mangels zentraler Instanz ist es für einen Angreifer ein Leichtes, einen modifizierten Client in das Netzwerk einzuschleusen. Im Folgenden soll untersucht werden, welcher Schaden damit angerichtet werden kann.

[16] Stand: August 2013

Die Sybil-Attacke hat ihren Namen von dem amerikanischen Buch *Sybil*, in dem es um eine Frau mit 16 separaten Persönlichkeiten geht. Auf P2P-Netzwerke übertragen bedeutet das, dass sich ein Angreifer als eine Vielzahl von (verschiedenen) Clients ausgibt und so verschiedene Teile des Netzwerks kompromittieren kann. Besonders brisant wird dies, wenn komplexe Routingstrukturen vorhanden sind, die von Teilnehmer zu Teilnehmer verschieden sind.

Der Angriff kann ein Auseinanderfallen des Netzwerks bewirken. Außerdem können die gutartigen Peers auf diese Weise sehr leicht mit einem Denial-of-Service Angriff belastet werden. Zu guter Letzt bietet sich die Sybil-Attacke zum Observieren der Peers an [66].

Generell kann jedes Einschleusen von falschen Peers als Sybil-Attacke benannt werden. Somit bildet eine Sybil-Attacke oft nur die Basis für eine andere Attacke.

Bei Bitcoin kann man die Sybil-Attacke z. B. anwenden, um den Netzwerkverkehr eines Peers zu belauschen. Dazu muss man allerdings erreichen, sich vollständig als dessen Nachbarn auszugeben.

Moshe Babaioff *et. al.* [3] kritisieren den Belohnungsmechanismus von Bitcoin. Momentan erhält der Client, der die kryptographische Aufgabe löst, die gesamten neu geschöpften Bitcoins. Moshe Babaioff et. al. schlagen vor, allen Clients auf dem Weg ebenfalls einen verminderten Betrag auszuzahlen, da diese durch die Weiterverbreitung der Nachricht einen Beitrag zur Lösung geleistet haben. Inspiriert wurde diese Idee durch die DARPA Network Challenge, bei der das Ziel war, einen wahllos treibenden roten Ballon zu finden. Erfolgreich war eine Universität, die nicht nur dem Finder eine Belohnung zugesprochen hatte, sondern auch denjenigen Personen, die den Finder auf den Wettbewerb aufmerksam gemacht haben.

Die Verbindung zu Bitcoin wird dann deutlich, wenn man den Clients bösartiges Verhalten zuschreibt. Das Netzwerk basiert auf dem Flooding-Algorithmus, der wiederum voraussetzt, dass die Clients die Nachrichten weiterverbreiten. Durch einen Belohnungsmechanismus, der auch denen eine (geringere) Belohnung zuspricht, die die Transaktion nur an den erfolgreichen Miner propagiert haben, erhöht sich die Wahrscheinlichkeit, dass Clients Transaktionen weiterleiten.

Bei diesem Ansatz ist problematisch, dass die simple Belohnung aller Clients in der Kette einen Angreifer gerade bei Bitcoin dazu einlädt über eine Sybil-Attacke, also durch falsche Clients, die Kette künstlich zu verlängern und sich so eine höhere Belohnung zu verschaffen. Dieses Problem bestand bereits beim DARPA Network Challenge, wurde dort aber nicht weiter behandelt. Moshe Babaioff et. al. wirken dem unter anderem dadurch entgegen, dass Ketten ab einer bestimmten Länge keine Belohnung mehr erhalten.

Das Problem der byzantinischen Generäle befasst sich mit der Aufgabe, in einem Netzwerk gemeinsam eine eindeutige Entscheidung zu treffen. Klassischerweise betrachtet man bei dem Problem drei Generäle, die eine Stadt belagern und von denen einer auf Seiten des Gegners steht. Die Stadt kann erobert werden, wenn mindestens zwei Generäle das Kommando zum Angriff geben (da die Stadt den Angriff nicht abwehren kann) oder aber, wenn die Generäle nichts weiter tun und die Belagerung fortsetzen (da die Stadtbevölkerung dann verhungert). Der Verräter versucht nun, dem einem General das Kommando zum Angriff zu geben und dem anderen General das Kommando zum Abwarten. Die gutartigen Generäle haben auch bei einem Informationsaustausch untereinander keinerlei Möglichkeit, die Korrektheit des Kommandos zu prüfen, sondern können auch weiterhin den Verräter nicht herausfinden [66].

Beachte hierzu auch Abbildung 8. General C ist der Verräter. Er gibt A den Befehl zum Angreifen und B den Befehl zum Abwarten. A gibt die Nachricht zum Angreifen dann an B weiter und B die Nachricht zum Abwarten an A. A weiß jetzt, dass irgendjemand falsche Befehle gegeben hat. Er kann aber selbst durch Nachfragen nicht verifizieren, was der richtige Befehl war, und wer die falschen Befehle gegeben hat. Übertragen auf ein Netzwerk mit drei Knoten, sind diese also ohne weiteres nicht in der Lage eine gemeinsame Entscheidung zu treffen, wenn einer absichtlich falschspielt.

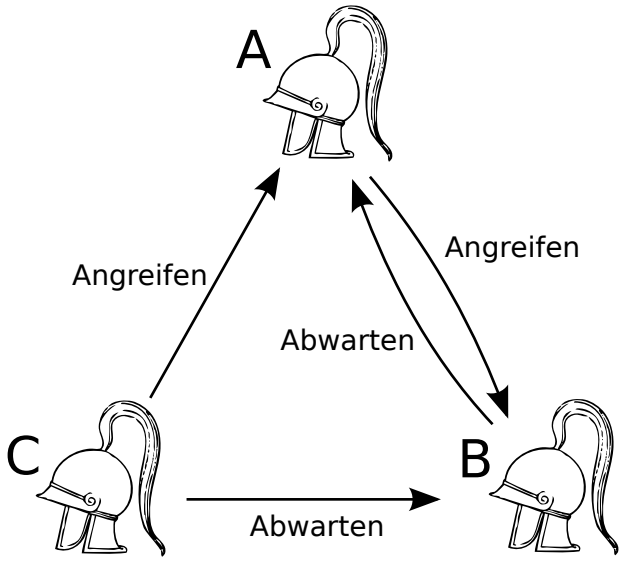

Abbildung 8: Problem der byzantinischen Generäle, Helm von [43]

Ab vier Generälen kann der falsche General überführt werden. Wenn zwei Generäle falsch spielen, ist das Problem aber wieder vorhanden. Grundsätzlich lautet das Theorem:

> Sei n die Anzahl der Knoten und t die Anzahl der bösartigen Knoten. Dann gilt: Es kann eine eindeutige Lösung erzielt werden, wenn $n \geq 3t + 1$ [66].

Eine Möglichkeit, das Problem zu lösen, ist die Einführung von eindeutigen Signaturen, sodass der Absender einer Nachricht identifiziert werden kann.

Das Bitcoin-Netzwerk ist ebenfalls von dem Problem betroffen, löst es aber. Transaktionen müssen eindeutig verbreitet werden, insbesondere müssen falsche und widersprüchliche Transaktionen abgefangen werden. Die Lösung bei Bitcoin besteht in der Mathematik der Block Chain. Etwa alle 10 Minuten lösen alle Generäle eine mathematische Aufgabe, in der die Nachrichten verarbeitet werden. Wenn diese gelöst wurde, wird die Lösung an alle Generäle weitergeleitet, die sie überprüfen und auf Grundlage dieser Lösung eine neue Lösung suchen usw.. Daraus entsteht eine Kette. Jeder General rechnet nur mit der längsten Kette. Auf das Angriffsszenario übertragen bedeutet dieser Ansatz, dass entweder das Kommando zum Angriff oder das zum Abwarten zuerst in der Lösung landet, die in die Kette einfließt, je nachdem, welcher General zuerst die Aufgabe löst. Nach einigen Durchläufen kann jeder General sicher sein, dass ein Konsens besteht, es sei denn, der falsche General hat die Mehrheit an Rechenkraft. Dann könnte er z. B. fortlaufend zwei Versionen der Kette produzieren [8].

3.4.6. Anonymität

Hier soll die am Anfang dieses Kapitels genannte Anonymität des Netzwerkes selbst diskutiert werden. Da Bitcoin IP[17] benutzt, kennen die Knoten ihre jeweiligen IP-Adressen. Mit der IP-Adresse kann über eine Anfrage beim Provider der Anschlussinhaber gefunden werden, sodass wirkliche Anonymität nicht gegeben ist.

Bitcoin versucht eigene Transaktionen zu verschleiern. Neue Transaktionen werden periodisch an die Nachbarn bekanntgegeben, die dann ihrerseits die Transaktionen an alle Nachbarn weitergeben usw.. Bitcoin geht nun davon aus, dass die eigenen Transaktionen in der Menge untergehen. Dazu versucht der Client, das Aufstellen von Statistiken zu erschweren [15]:

```
1  void CWallet::ResendWalletTransactions()
2  {
       // Do this infrequently and randomly to avoid giving away
       // that these are our transactions.
       static int64 nNextTime;
       if (GetTime() < nNextTime)
7          return;
       bool fFirst = (nNextTime == 0);
       nNextTime = GetTime() + GetRand(30 * 60);
       if (fFirst)
           return;
12     ...
```

Hier wird über nNextTime eine Zeit festgelegt, die mindestens verstrichen sein muss, bevor die Methode erneut durchläuft. nNextTime ist dabei ein zufälliger Wert, der bis zu 30 Minuten in die Zukunft reicht.

Es bleibt zu diskutieren, ob dieser Mechanismus eine wirksame Technik darstellt. Es sei angenommen, es würden ausschließlich periodisch sämtliche Transaktionen gesendet. Sei t_0 der Zeitpunkt, an dem Peer A seine Transaktionen das erste Mal sendet. Sei Peer B ein Nachbar von A und Peer $C \neq A$ ein Nachbar von B. Dann erhält B zum Zeitpunkt t_0 die Transaktionen von A und sendet diese (da auch er periodisch alles sendet) zum Zeitpunkt $t_0 + t_B$ an seine Nachbarn weiter – unter anderem an C. C hat über den Zeitpunkt des Empfangs keine Möglichkeit, aus der Menge der Transaktionen diejenigen herauszufiltern, die A gesendet hat. B hingegen kennt sowieso die von A gesendete Menge an Transaktionen (in denen er aber die von A selbst erzeugten nicht identifizieren kann), da er direkter Nachbar ist. Der Argumentation würde nicht greifen, wenn in den Transaktionen Zeitstempel eingebettet wären, die den Zeitpunkt des Empfangs speicherten. Es sind zwar Zeitstempel in Transaktionen vorhanden, diese werden aber erst in der Blockerzeugung gesetzt (dem Quellcode entnehmbar).

Diese Technik wird außerdem hinfällig, wenn ein Angreifer in der Lage ist, eingehenden und ausgehenden Datenverkehr eines Knotens zu kontrollieren. Ist dies der Fall, dann kann er leicht nachprüfen, welche Transaktion nur ausgegangen ist und kennt so die eigenen Transaktionen des Peers. Die Kontrolle über den Datenverkehr kann ein Angreifer durch eine erfolgreiche Sybil-Attacke erlangen. Eine weitere Möglichkeit ist das direkte Abgreifen des Datenverkehr, wie es z. B. für den Provider ein Leichtes wäre. Anonymisierungsdienste wie Tor helfen in diesem Punkt.

[17] „Internet Protocol“, das Standardprotokoll, in das Internetdatagramme gekapselt werden

3.5. Kryptographische Aufgabe

In einem Block werden die Transaktionen über einen Merkle Tree[18] gespeichert [17]. Somit reicht ein Hashwert um beliebig viele Transaktionen zu referenzieren. Dadurch ist auch die Rechenzeit (abgesehen von der einmaligen Hashwertbildung) für eine beliebige Anzahl Transaktionen dieselbe. Der Header eines Blocks besteht u. a. aus dem Hashwert des vorherigen Blocks, der Wurzel des Merkle Trees und einer 32-Bit Nummer, die *Nonce* genannt wird. Der Algorithmus für die kryptographische Aufgabe lautet damit wie folgt:

```
solution = ziel;
nonce = 0;
while (solution >= ziel) {
    solution = sha256(sha256(blockheader));
    nonce += 1;
}
```

Da Nonce nur ein 32 Bit Wert ist, ist die Wahrscheinlichkeit für einen Überlauf[19] relativ groß. Die Referenzimplementierung [30] verzichtet in einem solchen Fall darauf, weiterzurechnen [39]. Das Bitcoin-Wiki schlägt vor, in diesem Fall in der Generation-Transaktion (mit der der Miner seine neu geschöpften Coins erhält) den Input zu inkrementieren, der bei dieser Transaktion ansonsten unbenutzt ist. Damit ändert sich die Wurzel des Merkle Trees und somit auch der gesamte Hashwert des Headers [17, 33].

Mit „ziel“ ist die Grenze gemeint, die die Schwierigkeit der kryptographischen Aufgabe steuert. Im folgenden Abschnitt werde ich das genauer beschreiben.

3.5.1. Berechnung der Schwierigkeit

Die Blockerzeugung ist so implementiert, dass etwa alle zehn Minuten ein neuer Block erzeugt wird. Da die Rechenleistung des Netzwerkes nicht konstant bleibt, muss dazu in periodischen Abständen die Schwierigkeit der kryptographischen Aufgabe angepasst werden.

Die Schwierigkeit wird alle 2016 Blöcke neu definiert. Mit einer Zeit von zehn Minuten pro Block entspricht das $\frac{2016 \cdot 10}{60 \cdot 24 \cdot 7} = 2$ Wochen [19].

Die Berechnung erfolgt über das sogenannte *target* (das dem obengenannten „ziel“ entspricht), das von allen Clients individuell ausgerechnet und zudem in den Blöcken verkürzt gespeichert wird. Alle 2016 Blöcke wird diese Rechnung ausgeführt [12](Zeile 1102):

$$\text{target}_{\text{neu}} = \text{target}_{\text{alt}} \cdot \frac{\text{Zeit des aktuellen Blocks} - \text{Zeit des alten Blocks}}{2 \text{ Wochen}}$$

Mit „Zeit des alten Blocks“ ist der Block 2016 Stellen vor dem aktuellen gemeint. Dabei muss gelten:

$$\frac{1}{4} \leq \frac{\text{Zeit des aktuellen Blocks} - \text{Zeit des alten Blocks}}{2 \text{ Wochen}} \leq 4$$

Ansonsten wird mit $\frac{1}{4}$ bzw. 4 multipliziert.

[18] Ein Merkle Tree ist ein Binärbaum aus Hashwerten, in dem zwei konkatenierte Hashwerte (Kinderknoten) mit einem dritten gehasht werden (Elternknoten) [67].

[19] Ein Überlauf passiert dann, wenn die Zahl, die die Variable darstellt, für den Speicherplatz dieser zu groß wird.

3.5.2. Alternativen

Zu Bitcoin haben sich in letzter Zeit mehrere Alternativen entwickelt, die das eine oder andere verbessern sollen. Die bekanntesten sind Litecoin und PPCoin.

Litecoins verifizieren im Gegensatz zu Bitcoins etwa alle 2,5 Minuten einen neuen Block anstatt alle 10 Minuten. Das soll den Handel beschleunigen bzw. manche Transaktion erst möglich machen.

Außerdem verwendet Litecoin einen anderen Hashalgorithmus, der sehr speicherintensiv ist und so die Ausführung der Miningprozedur auf Spezialhardware einschränken soll.

PPCoin basiert auf Bitcoins, verteilt aber neue PPCoins nicht an die stärksten Miner, sondern lost aus, welcher PPCoin-Inhaber neues Geld bekommt. Die Wahrscheinlichkeit, ausgelost zu werden, hängt dabei von der Menge der PPCoins ab, die sich bereits im Besitz des Inhabers befinden.

Dieses Vorgehen soll den Energieverbrauch senken, der beim Bitcoin-Mining enorm ist. Außerdem ist die Gesamtmenge der PPCoins nicht begrenzt. [6, 71].

4. Bedeutung für Gesellschaft und Wirtschaft

Bitcoins waren bei ihrem Aufkommen mit der Eigenschaft ohne Bank auszukommen einmalig. Inzwischen gibt es alternative Ansätze, die aber einerseits auf Bitcoin basieren und anderseits bislang keine große Verbreitung gefunden haben. Bitcoins werfen mangels Kontrollmöglichkeit außerdem die Frage für Regierungen auf, wie mit ihnen umzugehen ist. So sind sie in Thailand z. B. bereits verboten worden.

Im Vergleich zu anderen Währungen haben Bitcoins bislang zwei Hypes durchlaufen, die den Kurs enorm ansteigen und gleich darauf stark fallen lassen haben.

Zu guter Letzt führt die Pseudoanonymität auch dazu, dass illegale oder politisch gefährliche Transaktionen über Bitcoin abgehandelt werden, wie z. B. Drogen- oder Waffenhandel oder Spenden an Wikileaks, einer Enthüllungsplattform. Nichtsdestotrotz kann man auch „normale“ Waren über Bitcoin kaufen.

4.1. Bitcoin und andere Währungen

Bitcoin kann an sogenannten Bitcoin-Börsen gegen andere Währungen getauscht werden. Im Gegensatz zu diesen fehlt eine zentrale Instanz, sodass der Kurs schlagartig steigen oder fallen kann. Der aktuelle Wert eines Bitcoins wird somit eher darüber definiert, wieviel er gerade an Bitcoin-Börsen in Form von Dollar einbringt. Silk Road, ein Online-Marktplatz, wo hauptsächlich Drogen gehandelt werden, unterstützt dementsprechend auch eine Liveanpassung des Bitcoin-Preises an den Dollar [9].

4.1.1. Bitcoin-Börsen

Anlaufstellen, bei denen man mit Bitcoin handeln oder Bitcoin gegen andere Währungen tauschen kann, heißen Bitcoin-Börsen. Die größte dieser Börsen ist Mt.Gox. Eine andere Börse, die für Schlagzeilen gesorgt hat, ist Bitcoin24.

Mt.Gox wird von der Mt. Gox Co. Ltd., die in Tokyo ansässig ist, bereitgestellt [73]. Die Börse sagt über sich selbst, die größte ihrer Art zu sein [74].

Mt.Gox hatte mit einigen rechtlichen und technischen Problemen zu kämpfen. So stand die Börse im April 2013 unter schwerem DDoS[20]-Beschuss [45], nachdem sie bereits im Juni 2011 erfolgreich gehackt wurde. Die Angreifer sollen dabei die zentrale Datenbank entwendet haben [1].

Im Mai 2013 wurde Mt.Gox die Möglichkeit genommen, über Dwolla, einem Zahlungsdienstleister ähnlich wie PayPal, zu handeln. Dwolla war eines der wenigen Finanzunternehmen, das es erlaubte, Bitcoins in gesetzliche Zahlungsmittel umzutauschen [37, 69].

Einige Wochen später, Ende Mai 2013, führte Mt.Gox eine Verifikation der Identität der Nutzer ein. Zumindest sofern sie Bitcoins in reales Geld tauschen wollten. Es wird vermutet, dass dies eine Reaktion auf den Schlag gegen „Liberty Reserve“ war, eine ebenfalls digitale Währung,

[20] Distributed Denial of Service, bezeichnet das massenhafte Stellen von Anfragen an den Server um ihn zum Absturz zu bringen.

die für massive Geldwäsche verwendet wurde. [7, 68]. Es bestand die Möglichkeit, dass Bitcoins Liberty Reserve ablösen könnten [65].

Im Juni 2013 setzte Mt.Gox dann temporär die Möglichkeit aus, Abhebungen in USD vorzunehmen. Als Begründung wurden technische Optimierungen genannt, da die zunehmende Anzahl Abhebungen das bisherige System zu langsam gemacht hatte [46].

Im August 2013 wurde bei Mt.Gox ein verbessertes Handelssystem eingeführt, das 500 000 Transaktionen pro Sekunde abwickeln können soll [44].

Bitcoin24 ist die größte europäische Bitcoin-Börse. Sie erregte im April 2013 Aufmerksamkeit, da sie kurzzeitig ausfiel, weil Konten in Polen gesperrt wurden. Dies geschah wegen des Verdachts auf Geldwäsche [47]. Die Aktion soll von deutschen Behörden ausgegangen sein [49]. Im Juni 2013 wurde die Pfändung des Kontos von Simon Hausdorf, dem Betreiber der Börse, aufgehoben, da sich der Verdacht nicht erhärten konnte. Die polnischen Konten sind weiterhin eingefroren [78].

4.1.2. Kursverlauf

In Abbildung 9 ist der Wechselkurs an der größten Bitcoin-Börse Mt.Gox seit 2010 – der Zeit ihrer Gründung [24] – dargestellt. Wie man erkennen kann, hat Bitcoin um den Mai 2011 und um

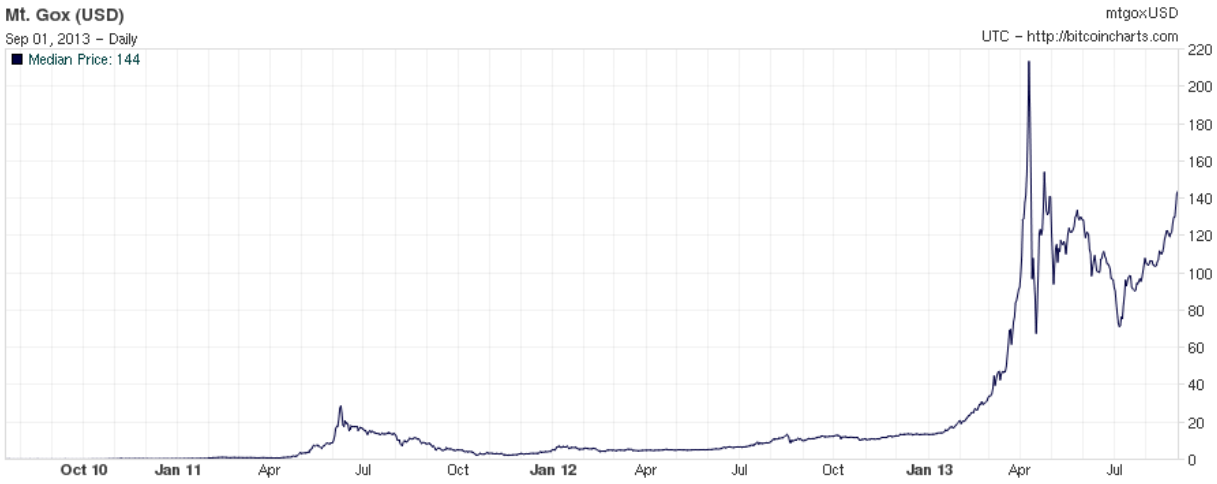

Abbildung 9: Wechselkurs von Bitcoin zu USD bei Mt. Gox, dargestellt wird der Wert von 1BTC [5]

den April 2013 zwei extreme Wertzuwächse zu verzeichnen. Die mediale Aufmerksamkeit stieg in dieser Zeit enorm. Ebenso ist zu erkennen, dass es sich um eine „Blase“ gehandelt hat und der Kurs genauso schnell, wie er gestiegen, auch wieder gefallen ist. Insgesamt ist aber dennoch ein starker Wertzuwachs erkennbar.

4.2. Politische Bedeutung

Politisch betrachtet ist die fehlende zentrale Instanz ein Problem. So wurde im Juli 2013 von der Bank of Thailand, der dortigen Zentralbank, ein Verbot für den Handel mit Bitcoins ausgesprochen. Als Begründung wurde der Mangel an einer gesetzlichen Grundlage und an Kontrollmechanismen genannt [4].

Ein US-Bundesrichter hingegen hat im August 2013 Bitcoin als Währung anerkannt. Dies geschah im Zusammenhang mit einem anderen Prozess, in dem der Angeklagte damit argumentierte, Bitcoins seien kein nach US-Gesetz reguliertes Geld [51].

Außerdem wurden Bitcoins bereits im März 2013 von einer dem US-Finanzministerium unterstellten Behörde unter das Geldwäschegesetz gestellt. [56].

Derzeit bemühen sich außerdem größere Bitcoin-Börsen um eine Anerkennung staatlicher Regulierer. So streben nach dem Londoner Finanzblatt *City A.M.* acht Bitcoin-Firmen in Großbritannien einen Status als Finanzdienstleister an, der ihnen durch die Finanzaufsicht FCA gegeben werden kann. Diese sei aber noch mit der Frage beschäftigt, wie überhaupt mit Bitcoins umzugehen sei.

In Deutschland strebt Bitcoin.de eine Zulassung seitens der Bankenaufsicht BaFin an, der Prozess sei aber noch nicht abgeschlossen [48].

Die Bundesregierung Deutschland hat im August 2013 Bitcoin als „privates Geld“ anerkannt, das in „multiliteralen Verrechnungskreisen“ eingesetzt werden kann [63].

Außerdem plädierte Bart Chilton in einem Interview für eine staatliche Regulierung von Bitcoin, damit sie nicht „wie ein Kartenhaus zusammenbrächen“. Wie das praktisch auszusehen hätte, wurde nicht gesagt [70].

4.3. Wirtschaftliche Lage

Nach inzwischen fünf Jahren Existenz werden immer mehr Zahlungen über Bitcoin abgewickelt. Infolgedessen will z. B. ein Paar aus dem US-Bundesstaat Utah drei Monate ausschließlich mit Bitcoin zahlen und die Erfahrungen dabei in einem Dokumentarfilm festhalten [72].

Abgesehen davon gibt es größere Internetfirmen und Organisationen, die Zahlungen mit Bitcoin ermöglichen.

Reddit bietet seit Februar 2013 die Möglichkeit, die angebotenen Premiumdienste mit Bitcoin zu bezahlen. Reddit ist eine hauptsächlich im englischsprachigen Raum bekannte Social News Site. Im August 2013 hatte die Seite über 73 Millionen Besucher, die 4.8 Milliarden Seiten angesehen haben [58].

Wikileaks gab im Juni 2012 bekannt, dass sie von diesem Zeitpunkt an anonyme Spenden in Form von Bitcoin akzeptieren. Die Anonymität des Bitcoin-Systems wurde dabei besonders betont [40].

Silk Road ist ein Online-Marktplatz, an dem hauptsächlich Drogen gehandelt werden. Die Internetseite ist ausschließlich über Tor Hidden Services[21] zu erreichen und akzeptiert nur Zahlungen über Bitcoin [9].

Berlin Kreuzberg: In Berlin Kreuzberg existieren einige Läden, in denen man mit Bitcoin bezahlen kann – darunter ein Laden für Langspielplatten, ein Hostel und ein Restaurant [53, 79].

Free Software Foundation: Die Organisation, die sich für freie Software einsetzt, akzeptiert Spenden in Bitcoin [20].

Weiterhin existieren zahllose kleinere Shops, die Zahlungen in Bitcoin akzeptieren, darunter viele Server- oder Domainanbieter wie „namecheap.com“, aber auch Schmuck-, Musik- , Lebensmittel- und Buchhändler sowie Bekleidungsgeschäfte. Außerdem nehmen viele Organisationen Spenden per Bitcoin entgegen wie netzpolitik.org[22], diaspora[23] und Torservers.net[24]. Eine Liste, die derartige Unternehmen und Organisationen sammelt ist z. B. im Bitcoin-Wiki [22, 31] zu finden.

Insgesamt hat die Marktkapitalisierung von Bitcoin (Menge an Coins mal Wechselkurs) bereits im März 2013 eine Milliarde Dollar überschritten und zeigt damit den Erfolg der Währung [50].

[21] Das sind Server, die nur über das Tor-Netzwerk zu erreichen sind.
[22] Ein Blog über netzpolitische Themen
[23] Ein alternatives verteiltes soziales Netzwerk
[24] Ein Projekt, das Tor-Server betreibt

5. Zusammenfassung und Ausblick

5.1. Bestehende Probleme

Die Praxis hat gezeigt, dass Bitcoin einige Mängel hat oder zumindest Eigenschaften, die als solche erscheinen.

Anonymität: Bitcoin ist nicht vollständig anonym. Durch die zugrunde liegende Technik ist es weiterhin nicht möglich, vollständige Anonymität zu erreichen. Der Ausweg sind Protokollerweiterungen wie Zerocoin.

Dead Coins: Verliert eine Person ihre privaten Schlüssel, sind die darauf transferierten Coins für immer verloren. Man nennt diese Coins *Dead Coins* [2]. Das Problem wird dadurch verschärft, dass nicht zwischen Dead Coins und Langzeitspeicherung (also die Verwendung von Bitcoin als Erspartes) unterschieden werden kann. Die Menge der Dead Coins ist unbekannt.

Gefahr der Zentralisierung: Die Block Chain wird immer größer, sodass nicht mehr jeder Anwender gewillt ist, diese komplett zu speichern. Satoshi Nakamoto hat dieses Problem bereits vor der ersten Implementierung erkannt und schlägt als Lösung vor, nur die Blockheader und die benötigten Transaktionen zu speichern. Dieses Vorgehen ist durch die Verwendung von Merkle Trees problemlos möglich. Ein solches Verhalten führt aber dazu, dass ein Client sich zur Verifikation bestimmter Transaktionen auf andere Clients verlassen muss, die aber nicht zwangsläufig durch das Protokoll dezentral verteilt sind. Nakamoto schlägt als Ausweg vor, bei einem Vertrauensbruch eines Peers sofort die betreffenden Transaktion herunterzuladen, erkennt aber auch, dass das Netzwerk durch ein ausschließliches Speichern der Header anfälliger wird [61]. Es besteht die Gefahr in der Entstehung von wenigen Peers, die alles speichern und vielen, die nur die Header speichern, was dem Dezentralisierungsgedanken zuwiderläuft.

Ein weiteres Problem ist der fortwährende Anstieg der Rechenleistung des Netzwerkes. Mining auf CPUs ist schon seit langem ineffizient (die Stromkosten sind höher, als der Wert der durch das Mining verdienten Coins), das Aufkommen von spezieller Hardware wie FPGAs oder extra dafür hergestellte ASICs macht das Mining auf GPUs inzwischen ebenfalls immer ineffizienter [41]. Diese Spezialhardware wird aber voraussichtlich von der Mehrheit der Benutzer nicht extra gekauft werden. Folglich ist die Gefahr der Entstehung von sogenannten Supernodes gegeben, die das Mining übernehmen, dementsprechend aber auch die Kontrolle über das Netzwerk haben.

Eine weitere (existierende) Entwicklung ist die Bildung von Pools aus Minern, die sich zusammenschließen, um eine größere Wahrscheinlichkeit zu besitzen einen Block zu erzeugen. Der Ertrag, der bei der Lösung der kryptographischen Aufgabe entsteht, wird gleichmäßig im Pool verteilt [76]. Der Gedanke ist dabei ähnlich wie der von Moshe Babaioff *et. al.* [3]. Ein solcher Pool stellt aber wieder eine Zentralisierung dar.

5.2. Bitcoin 2

Die Probleme sind allgemein bekannt und es wurde bereits nach Auswegen gesucht. Im Juni 2013 erschien ein Paper, das Bitcoin 2 vorgestellt hat, eine Erweiterung oder Fortführung des originalen Bitcoin-Systems, das sich zum Ziel gesetzt hat, mit möglichst wenig Aufwand die Probleme zu tilgen [2].

Diese Ideen werden dabei eingearbeitet:

Sicherheit: Setze das KISS-Prinzip[25] um, genauer gesagt, ändere möglichst wenig am Protokoll.

Halte die Block Chain klein: Mit zwei Vorgaben wird versucht, die Block Chain klein zu halten und so dem Problem entgegenzuwirken, dass immer mehr Speicherplatz und Bandbreite zur Übertragung benötigt wird.

- Limitiere die Größe der Blöcke auf 150 KB. Begrenze also die Anzahl an Transaktionen pro Block. Sortiere dazu unscharf nach der Höhe der Transaktionsgebühr, um die Transaktionen zu erhalten, die in den Block aufgenommen werden.
- Alte Blöcke verfallen. Ab einem gewissen Alter werden die Transaktionen in Blöcken ungültig und nur noch die Header gespeichert, um die kryptographische Korrektheit der Block Chain zu gewährleisten. Die entstehenden Coins[26] werden in einer Lotterie neu verteilt. Dieses Verhalten wirkt Dead Coins entgegen.

Mache Bitcoin technisch unverwüstlich: Damit ist die Unempfindlichkeit gegenüber Kontrollversuchen gemeint. Folgende zwei Ideen sollen helfen.

- Unterstütze mehrere Übertragungswege, im Genauen Tor, I2P[27] und Sneakernet[28].
- Erzwungenes Mischen über Zerocoin z. B. alle 12 Stunden.

Dezentralisierung von Minern und Code: Sowohl die Miningvorgänge als auch die Entwicklung des Codes soll möglichst dezentral ablaufen.

- Füge für die Miner eine Motivation hinzu sich zu dezentralisieren. Der vorgeschlagene Mechanismus dazu ist die eindeutige Identifizierung eines Miners zu einem von ihm erzeugten Block (über eine eindeutige Nummer), sodass Miner unterschieden werden können. Zusätzlich muss es eine Möglichkeit geben, Clients bestimmten Minern das Verrechnen ihrer Transaktionen zu verbieten. Zu mächtige oder „böse“ Miner können daraufhin von den Clients ausgeschlossen werden. Die Identifikationsnummer ist nur für eine bestimmte Anzahl von Blöcken gültig, sodass Miner nach einer gewissen Anzahl von Blöcken eine neue Chance bekommen. Die Erzeugung der Nummer ist dabei so, dass sie beim ersten Mal sehr

[25] Wird oft übersetzt mit „**K**eep **I**t **S**imple and **S**tupid“, anders gesagt, wähle den einfachsten Weg.
[26] Bitcoins, die an Adressen transferiert wurden, die in keiner neuen Transaktion mehr verwendet werden.
[27] Ebenfalls ein Anonymisierungsnetzwerk
[28] Ironischer Name für ein Netzwerk aus Turnschuhen, damit ist das manuelle Hin- und Hertragen der Daten von PC zu PC gemeint [34].

teuer (vom Rechenaufwand her) ist, sodass verhindert wird, dass Miner fortlaufend neue Identitäten erzeugen.

- Standardisiere das Protokoll und veröffentliche es.
- Führe eine Versionierung in Blöcken und Transaktionen ein. Sobald 80 % des Netzwerkes softwareseitig die neue Version verwenden können, verwende diese.

Die in Bitcoin 2 genannten Ansätze sind teilweise ziemlich radikal, insbesondere der Verfall von alten Transaktionen, lösen aber viele bestehende Probleme ohne das System grundlegend zu verändern.

5.3. Zusammenfassung

Der existierende wirtschaftliche Erfolg der Währung zeigt, dass Bitcoin mit seinem neuen Konzept Bestand haben kann. Der Mechanismus, dezentral Geld zu verteilen und zu überweisen, ist zum Zeitpunkt der Erfindung einmalig gewesen und stellt auch heute noch Wirtschaft und Politik vor Probleme, da sämtliche Kontrolle kryptographisch mathematisch passiert.

Es wurden bis heute keine schwerwiegenden Fehler in der technischen Implementierung entdeckt, sodass Bitcoin vorerst als sicher gelten kann. Der Umstand, dass ausschließlich standardisierte Verfahren zum Verschlüsseln und Hashen verwendet werden, unterstützt diese Annahme. Die einzige Gefahr für das Netzwerk ist ein übermächtiger Angreifer, der es schafft, die Rechenleistung der anderen zu überbieten. Es gilt also:

Der Stärkere gewinnt.

Nakamoto stellt aber bereits im originalen Paper über Bitcoin [61] die Frage, ob ein solcher Angreifer nicht eher zum Miner werden würde und so auf legale Weise Geld verdiente (auch beim Mining gilt, dass der Stärkere gewinnt).

Bitcoin hat aber außer diesem einige weitere Mängel. Denen wird aber z. B. mit Bitcoin 2 aktiv entgegengewirkt.

Alles in allem kann man sagen, dass Bitcoin es geschafft hat, einen neuartigen Ansatz für Geldtransfers zu etablieren.

Bitcoin hat außerdem in der Geschichte der P2P-Netzwerke Fortschritte gebracht. So löst es das Problem der byzantinischen Generäle auf eine neuartige Weise, schafft es also, eindeutige Nachrichten in einem Netzwerk aus gleichberechtigten Teilnehmern zu verbreiten.

Anhang

A. SHA-256 und RIPEMD-160

Hashfunktionen bilden eine beliebig lange Bitfolge, die Urbild genannt wird, auf eine begrenzt lange Bitfolge ab. Bei SHA-256 sind dies 256 Bit, wodurch auch der Name zustande kommt. Wenn zwei Urbilder denselben Hashwert hervorrufen – ein Fall, der aufgrund der Zuordnung einer unendlichen Menge zu einer endlichen Menge eintritt – nennt man dies Kollision.

A.1. Kryptographische Hashfunktion

SHA-256 und RIPEMD-160 gehören zu den kryptographischen Hashfunktionen. Kryptographische Hashfunktionen zeichnen sich dadurch aus, dass ein potentieller Angreifer keine Möglichkeit hat Kollisionen herbeizuführen. Es gibt dabei zwei Arten von Funktionen:

Stark kollisionssichere Hashfunktionen haben die Eigenschaft, dass man keine zwei Urbilder (mit vertretbarem zeitlichen Aufwand) finden kann, die zu Kollisionen führen.

Schwach kollisionssichere Hashfunktionen haben die Eigenschaft, dass man zu einem gegebenen Urbild kein zweites Urbild (mit vertretbarem zeitlichen Aufwand) finden kann, das zu einer Kollision führt.

SHA-256 ist eine stark kollisionssichere Hashfunktion.

Der Zusammenhang zwischen Kryptographie und Kollisionssicherheit besteht in der Verwendung der Hashfunktionen. Oft wird nur ein Hash der zu unterschreibenden Nachricht signiert. Ein potentieller Angreifer hätte also bei einer nicht kollisionssicheren Hashfunktion die Möglichkeit, bei gleicher Signatur ein anderes Urbild unterzuschieben und so den Inhalt zu verändern [67].

A.2. Technik von SHA-256

SHA-256 ist eine Weiterentwicklung von SHA-1, das wiederum eine Weiterentwicklung der NSA[29] von MD4 ist, einer Hashfunktion von Ron Rivest. Sowohl SHA-1 als auch MD4 sind für heutige Großrechner mit einer Hashwertlänge von 160 Bit bzw. 128 Bit zu kurz. Außerdem wurden Angriffe entwickelt, die Fehler in den Funktionen ausnutzen. Angriffe auf SHA-256 sind zum einen wegen der Hashwertlänge von 256 Bit deutlich zeitaufwendiger und zu anderen sind nur Brute-Force-Angriffe bekannt [67].

SHA-256 verarbeitet das Urbild in Blöcken der Länge von 512 Bit. Das Urbild muss dabei in den meisten Fällen auf eine Länge gebracht werden, die durch 512 teilbar ist. Dies geschieht nach folgender Vorschrift:

[29] National Security Agency, der Auslandsgeheimdienst der USA

```
rest = urbildlaenge mod 512;
if (urbild mod 512 < 64) {
    hänge das Bit „1“ an das Urbild an
    hänge 512 − 64 − 1 + rest „0“-Bits an das Urbild an
} else {
    hänge das Bit „1“ an das Urbild an
    hänge 512 − 64 − 1 − rest „0“-Bits an das Urbild an
}
schreibe in die 64 verbleibenden Bits: originale_urbildlaenge mod 2^64
```

Anschließend wird eine Kompressionsfunktion k auf das Urbild angewendet. Die zentrale Rolle spielen dabei acht Kettenvariablen H_i, $(i = 0, \ldots, 7)$. k generiert aus einem Block des Urbilds und den acht alten Kettenvariablen acht neue Kettenvariablen $(H_{i,n})$:

$$(H_{0,n}, H_{1,n}, H_{2,n}, H_{3,n}, H_{4,n}, H_{5,n}, H_{6,n}, H_{7,n}) = k(U, H_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7)$$

U ist dabei der Urbildblock.

H_0 bis H_7 werden für die Berechnung mit Konstanten vorinitialisiert, dann wird mehrfach k darauf angewendet. Am Ende bildet die Konkatenation dann den Hashwert:

```
K_0 = 6a09e667, K_1 = bb67ae85, K_2 = 3c63f372, K_3 = a54ff53a;
K_4 = 510e527f, K_5 = 9b05688c, K_6 = 1f83d9ab, K_7 = 5be0cd19;

for (int i = 0, i < urbildbloecke.length−1, i++)){
    K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7 = k(urbildbloecke[i], K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7)
}

hashwert = concatenate(K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7)
```

k arbeitet in 4 Runden mit jeweils 16 Teilrunden. Der generelle Ablauf ist in Abbildung 10 zu sehen. K_t ist dabei eine von 64[30] Konstanten [62] und W_t ist ein 32 Bit großer Teil des Urbildes. Da nur 512 Bit des Urbildes pro Anwendung von k verwendet werden, die Teilrunden aber $64 \cdot 32 = 2084$ Bit benötigen, existiert eine Expansionsformel, die die 512 Bit nach einem bestimmten Muster auf 2048 erweitert. Σ_0, Σ_1, Ch, Maj, σ_0 und σ_1 sind boolesche Funktionen[31] auf die Bits, z. B. ist:

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

A.3. Technik von RIPEMD-160

RIPEMD-160 ist eine Weiterentwicklung von RIPEMD, da dies einige Schwachstellen aufwies, seinerseits aber eine Weiterentwicklung von MD4 ist. Somit sind SHA-256 und RIPEMD-160 verwandt, was zu einem ähnlichen Verfahren führt.

Wie SHA-256 verwendet RIPEMD-160 512-Bit-Blöcke, generiert aber einen 160 Bit Hashwert. Der Auffüllmechanismus ist ebenfalls derselbe wie bei SHA-256. RIPEMD-160 verwendet aber

[30] Für jede Teilrunde eine
[31] Für eine genaue Beschreibung sei auf [62] verwiesen.

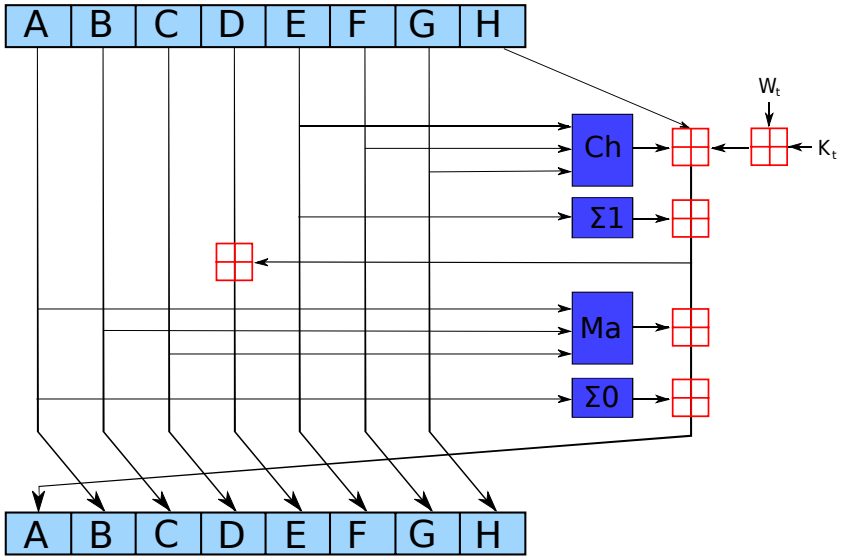

Abbildung 10: Eine Teilrunde in SHA-256, A-H sind dabei H_0-H_1, W_t ist ein Teil des Urbildes, K_t eine Konstante und Ch, $\Sigma 1$, Ma, $\Sigma 0$ Funktionen, die roten Kästen stehen für eine XOR-Verknüpfung [55].

nur fünf Kettenvariablen, andere Konstanten und Funktionen und eine andere Funktion k. k besteht aus zwei Ablaufrunden, die jeweils aus fünf Runden mit 16 Teilrunden bestehen. Die Ablaufrunden unterscheiden sich wiederum in Konstanten und Funktionen. Wie bei SHA-256 muss das Urbild expandiert werden, dies geschieht aber mithilfe einer Tabelle statt mit einer Formel.

RIPEMD-160 ist ein Verfahren, bei dem bisher keine Schwächen gefunden wurden, insbesondere greifen die Angriffe auf SHA-1 dort nicht. Die Hashwertlänge von 160 Bit ist aber wie bei SHA-1 zu kurz, um wirkliche Sicherheit zu bieten [67].

A.4. Doublehashing

Bitcoin wendet Hashfunktionen ausschließlich doppelt an. Zum Einsatz kommen dabei entweder *ripemd160(sha256(value))* oder *sha256(sha256(value))*. Diese Vorgehensweise könnte auf die Verhinderung der *Length-extension-Attacke* abzielen, bei der $H(x|y)$[32] konstruiert werden kann, wenn $H(x)$ bekannt ist, ohne x selbst zu kennen. H ist hierbei eine für diesen Angriff anfällige Hashfunktion wie SHA-256. Die doppelte Anwendung der Hashfunktion verhindert einen solchen Angriff. Ob ein solcher Angriff bei Bitcoin sinnvoll ist, ist nicht bekannt [11].

B. ECDSA

Bei ECDSA handelt es sich um ein kryptographisches Signaturverfahren. ECDSA steht für „Elliptic Curve Digital Signature Algorithm“. Es ist die Anwendung von elliptischen Kurven auf das

[32] Konkatenation von x und y

Standardverfahren DSA [67].

B.1. DSA

DSA wurde von der NSA entwickelt, 1991 veröffentlicht und 1994 standardisiert. Es basiert auf dem diskreten Logarithmus und somit auf dem ElGamal-Verfahren, welches das erste Signaturverfahren auf Basis des diskreten Logarithmus war. DSA verwendet einige Sätze der Gruppentheorie, die hier im Detail nicht erläutert werden, die ich aber kurz vorstellen werde.

Für eine Primzahl p kann man eine Gruppe $Z(p, \cdot)$ definieren. Die Elemente der Gruppe sind dabei die Zahlen $0, \ldots, p-1$. Die Operation „$*$“ bezeichnet die über die natürlichen Zahlen definierte Multiplikation. Die Operation „$\cdot$“ der Gruppe ist definiert durch

$$a \cdot b = a * b \mod p$$

Sie wird auch *Modulo-Multiplikation* genannt.

Eine Untergruppe ist eine Teilmenge der Gruppe, die mit derselben Verknüpfung und demselben neutralen Element[33] wieder eine Gruppe bildet. Die Anzahl der Elemente in einer Untergruppe ist dabei ein Teiler der Elemente der zugehörigen Gruppe. Die eben definierte Gruppe besitzt $p-1$ Elemente. Die möglichen Untergruppen bestehen folglich aus einer Anzahl von Elementen, die $p-1$ teilt. Wenn p eine Primzahl ist, gibt es für jeden Teiler von $p-1$ genau eine Untergruppe.

Man kann nachweisen, dass die Menge $\{a, a^2, a^3, \ldots, a^{p-1} \pmod p\}$ für ein beliebiges Element a der Gruppe eine Untergruppe bildet. a heißt dann Generator dieser Gruppe.

Um eine DSA-Signatur zu erstellen, braucht man zuerst ein Schlüsselpaar. Dazu wählt man eine große Primzahl p. Außerdem wählt man eine Primzahl q, die $p-1$ teilt. q hat nach der Spezifikation dabei eine Länge von 160 Bit. Nun wählt man einen Generator g der (einen) Untergruppe mit q Elementen und eine beliebige Zahl $x < q$. x bildet den privaten Schlüssel. Den öffentlichen Schlüssel kann man über $g^x \pmod p$ berechnen und veröffentlichen.

Der eigentliche Signaturprozess benutzt das Schlüsselpaar folgendermaßen: Sei die Nachricht, die unterschrieben werden soll, m. Außerdem wähle man eine Zufallszahl $y < q$, daraus wird die Zahl $r = (g^y \pmod p) \pmod q$ berechnet. Mit diesen Zahlen kann man die Gleichung:

$$m = y \cdot s - x \cdot r \mod q$$

aufstellen. Alles außer s ist davon bekannt. Durch algebraische Umformung kann s berechnet werden. (r, s) bilden zusammen die Signatur von m.

Die Verifikation geschieht dann per

$$g^m = \frac{r^s}{g^{x \cdot r}} \mod p$$

x wird hierfür nicht benötigt, da g^x bereits als öffentlicher Schlüssel vorliegt. Für die Verifikation der Signatur reicht also der öffentliche Schlüssel. Die Signatur selber, insbesondere s, kann aber

[33] Das Element n, für das für ein beliebiges a gilt: $a \cdot n = a$.

nur mithilfe von x berechnet werden. Die Sicherheit des Verfahrens beruht auf der nicht möglichen Umkehrung der Modulo-Exponentiation:

$$o = g^x \mod p$$

Bei gegebenem g, x und p ist es sehr leicht o zu berechnen. Bei gegebenem g, o, und p hingegen ist es in zumutbarer Zeit nicht möglich x zu berechnen.

DSA setzt einen guten Zufallsgenerator voraus, um y zu generieren, da diese Zahl nicht erraten werden darf [67].

B.2. Erweiterung auf elliptische Kurven

Eine elliptische Kurve ist als eine Kurve auf einem Körper definiert, die diese Gleichung erfüllt:

$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5$$

Außerdem gehört zu der Kurve ein Punkt, der im Unendlichen liegt und als 0 bezeichnet wird.

Ein Körper ist dabei eine Menge, auf der zwei Verknüpfungen definiert sind, wie z. B. die Modulo-Addition und Modulo-Multiplikation. Zu jeder Primzahl und jeder natürlichen Zahl gibt es nun einen Körper mit p^n Elementen, der $GF(p^n)$ genannt wird. Kryptographische Systeme auf Basis elliptischer Kurven lassen sich am besten auf den Körpern realisieren, bei denen $n = 1$ oder $p = 2$ ist, also $GF(p)$ oder $GF(2^n)$, im Folgenden $GF(c)$ genannt.

Elliptische Kurven haben die Eigenschaft, dass bei zwei Schnitten mit einer beliebigen Geraden immer noch ein dritter Schnittpunkt entsteht. Folgende Fälle treten dabei auf (vergleiche Abbildung 11):

1. Bei einer parallel zur y-Achse verlaufenden Geraden ist der dritte Schnittpunkt 0.
2. Bei einer Tangente wird der Berührpunkt doppelt gezählt.
3. Es gibt drei „normale“ Schnittpunkte.

Die elliptische Kurve e über dem Körper K bildet mit einer geeigneten Operation eine Gruppe $E(K)$. Die Punkte von e bilden die Elemente der Gruppe. Die Multiplikation kann man wie folgt definieren. Seien a und b die Punkte, die multipliziert werden sollen. Dann ist das Ergebnis c der Multiplikation der dritte Schnittpunkt, den die Gerade, die durch a und b verläuft, mit e bildet.

Man kann die Potenzfunktion für $E(GF(c))$ als mehrfache Ausführung der Multiplikation auffassen und dementsprechend den Logarithmus als Umkehrfunktion, mit dem die Potenz herausgefunden wird. Bei elliptischen Kurven ist die Potenz sehr einfach und der Logarithmus sehr schwer zu berechnen. Deswegen lassen sich kryptographische Verfahren, die auf dem diskreten Logarithmus beruhen, auf elliptische Kurven übertragen.

Der große Vorteil bei der Verwendung von elliptischen Kurven besteht darin, dass für die Berechnung des Logarithmus nur wenig effiziente Verfahren existieren. Sei f der Logarithmus, der auf $E(GF(c))$ definiert ist, und g der Logarithmus, der auf $GF(c)$ definiert ist, dann gilt (c, die Anzahl der Elemente des Körpers, ist variabel):

$$O(f) = tc \qquad O(g) = log(c) \quad \text{mit } t = \text{konst.}$$

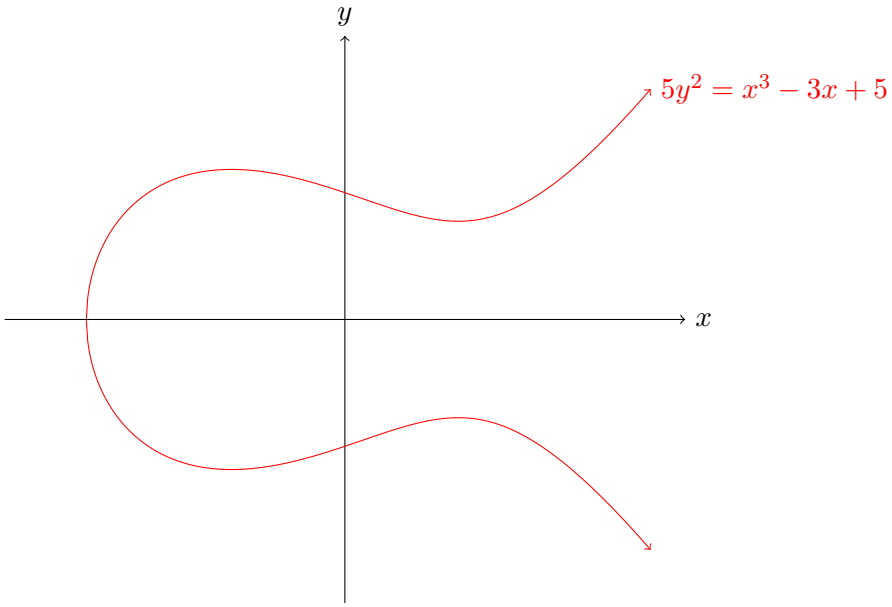

Abbildung 11: Eine elliptische Kurve über dem Körper der reellen Zahlen Formel von [38]

Da also der Berechnungsaufwand für den Logarithmus in $E(GF(c))$ schneller zunimmt als bei $GF(c)$, sind bei ECDSA kürzere Schlüssellängen und deswegen kürzere Rechenzeiten als bei DSA bei gleicher Sicherheit möglich [67].

Abbildungsverzeichnis

Literatur

[1] Detlef Borchers Andreas Wilkens. Bitcoin-Tauschbörse nach Angriff geschlossen. http://www.heise.de/newsticker/meldung/Bitcoin-Tauschboerse-nach-Angriff-geschlossen-1263327.html. Abgerufen am 01.09.2013.

[2] Eric Angeles, Sid und Gonzales. Bitcoin 2: Freedom of Transaction. July 2013.

[3] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 56–73, New York, NY, USA, 2012. ACM.

[4] Sylvia Beckers. Bitcoin-Verbot in Thailand. http://www.heise.de/newsticker/meldung/Bitcoin-Verbot-in-Thailand-1926521.html. Abgerufen am 01.09.2013.

[5] bitcoincharts.com. Pricechart. http://www.bitcoincharts.com/charts/mtgoxUSD#tgMzm1g10zm2g25. Abgerufen am 01.09.2013, linzensiert unter CC-BY-SA 3.0.

[6] Hanno Böck. Alternativen zu Bitcoin. http://www.golem.de/news/virtuelle-waehrungen-alternativen-zu-bitcoin-1304-98879.html. Abgerufen am 01.09.2013.

[7] Hanno Böck. Bitcoin-Tauschdienst führt Verifikation ein. http://www.golem.de/news/mt-gox-bitcoin-tauschdienst-fuehrt-verifikation-ein-1305-99530.html. Abgerufen am 01.09.2013.

[8] Paul Bohm. Bitcoin's Value is Decentralisation. http://paulbohm.com/articles/bitcoins-value-is-decentralization/. Abgerufen am 01.09.2013.

[9] Nicolas Christin. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace, November 2012.

[10] Carlisle Clark, Jeremy und P.C. van Oorschot und Adams. Usability of anonymous web browsing: An examination of tor interfaces and deployablility, 2007. Symposium on Usable Privacy and Security (SOUPS) 2007, July 18-20, 2007, Pittsburgh, PA, USA.

[11] CodesInChaos. Where is Double hashing performed in Bitcoin? http://bitcoin.stackexchange.com/questions/8443/where-is-double-hashing-performed-in-bitcoin. Abgerufen am 01.09.2013.

[12] Satoshi Nakamoto und The Bitcoin developers. main.cpp. Quellcodedatei. Git-Commit: 122e9f8ba4b403fbefe20740d7d030a1eec6795f.

[13] Satoshi Nakamoto und The Bitcoin developers. net.cpp. Quellcodedatei. Git-Commit: 122e9f8ba4b403fbefe20740d7d030a1eec6795f.

[14] Satoshi Nakamoto und The Bitcoin developers. net.h. Quellcodedatei. Git-Commit: 122e9f8ba4b403fbefe20740d7d030a1eec6795f.

[15] Satoshi Nakamoto und The Bitcoin developers. wallet.cpp. Quellcodedatei. Git-Commit: 122e9f8ba4b403fbefe20740d7d030a1eec6795f.

[16] diverse. Anonymität. http://de.wikipedia.org/wiki/Anonymit%C3%A4t. Abgerufen am 01.09.2013, lizensiert unter CC-BY-SA 3.0.

[17] diverse. Block hashing algorithm. https://en.bitcoin.it/wiki/Block_hashing_algorithm. Abgerufen am 22.06.2013.

[18] diverse. Controlled supply. https://en.bitcoin.it/wiki/Controlled_Supply. Abgerufen am 22.06.2013.

[19] diverse. Difficulty. https://en.bitcoin.it/wiki/Difficulty. Abgerufen am 22.06.2013.

[20] diverse. Donate to the FSF. https://my.fsf.org/donate. Abgerufen am 01.09.2013.

[21] diverse. FAQ. https://en.bitcoin.it/wiki/FAQ. Abgerufen am 22.06.2013.

[22] diverse. Handel. https://de.bitcoin.it/wiki/Handel. Abgerufen am 01.09.2013.

[23] diverse. Mining. https://en.bitcoin.it/wiki/Mining. Abgerufen am 22.06.2013.

[24] diverse. Mt.Gox. http://en.wikipedia.org/wiki/Mt.Gox. Abgerufen am 01.09.2013, lizensiert unter CC-BY-SA 3.0.

[25] diverse. Namecoin. https://en.bitcoin.it/wiki/Namecoin. Abgerufen am 05.09.2013.

[26] diverse. Network. https://en.bitcoin.it/wiki/Network. Abgerufen am 22.06.2013.

[27] diverse. Paysafecard. https://de.wikipedia.org/wiki/Paysafecard. Abgerufen am 05.09.2013, lizensiert unter CC-BY-SA 3.0.

[28] diverse. Protocol specification. https://en.bitcoin.it/wiki/Protocol_specification. Abgerufen am 22.06.2013.

[29] diverse. Satoshi Nakamoto. https://en.bitcoin.it/wiki/Satoshi_Nakamoto. Abgerufen am 05.09.2013.

[30] diverse. Software. https://en.bitcoin.it/wiki/Software. Abgerufen am 22.06.2013.

[31] diverse. Spenden. https://de.bitcoin.it/wiki/Spenden. Abgerufen am 01.09.2013.

[32] diverse. Technical background of version 1 Bitcoin addresses. https://en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses. Abgerufen am 01.09.2013.

[33] diverse. Transactions. https://en.bitcoin.it/wiki/Transactions. Abgerufen am 22.06.2013.

[34] diverse. Trunschuhnetzwerk. https://de.wikipedia.org/wiki/Sneakernet. Abgerufen am 05.09.2013, lizensiert unter CC-BY-SA 3.0.

[35] diverse. Währungs Statistik. http://blockchain.info/stats. Abgerufen am 05.09.2013.

[36] diverse. Zero-Knowledge-Beweis. http://de.wikipedia.org/wiki/Zero-Knowledge-Beweis. Abgerufen am 01.09.2013, lizensiert unter CC-BY-SA 3.0.

[37] Nico Ernst. US-Behörde beschlagnahmt Konto von MtGox. http://www.golem.de/news/bitcoins-us-behoerde-beschlagnahmt-konto-von-mtgox-1305-99263.html. Abgerufen am 01.09.2013.

[38] Fredstober. Difficulty. http://de.wikipedia.org/w/index.php?title=Datei:Ell_kurve.png&filetimestamp=20041112233133&. Abgerufen am 01.09.2013, mit Maple erzeugt, das Werk unterliegt der Gemeinfreiheit.

[39] Jeff Garzik. pyminer.py. https://github.com/jgarzik/pyminer/blob/8a5bf07b81bc0de095a9a05689c29c358baad6ca/pyminer.py. Lizenz: GPL, Git-Commit: 8a5bf07b81bc0de095a9a05689c29c358baad6ca.

[40] Fergal Reid und Martin Harrigan. *An Analysis of Anonymity in the Bitcoin System*. Springer, 2013. Die in dieser Arbeit enthaltenen Abbildungen wurden mit freundlicher Genehmigung der Autoren verwendet.

[41] Adrianne Jeffries. Miner problem: big changes are coming for Bitcoin's working class. http://www.theverge.com/2012/11/16/3649784/bitcoin-mining-asics-block-reward-change. Abgerufen am 05.09.2013.

[42] Jleedev. File:Stick Figure.svg. http://commons.wikimedia.org/wiki/File:Stick_Figure.svg. Abgerufen am 05.09.2013, unter Public Domain.

[43] johnny_automatic. Greek helmet 12. https://openclipart.org/detail/20907/greek-helmet-12-by-johnny_automatic. Abgerufen am 01.09.2013, unter Public Domain.

[44] Axel Kannenberg. Bitcoin-Börse Mt. Gox führt verbessertes Handelssystem ein. http://www.heise.de/newsticker/meldung/Bitcoin-Boerse-Mt-Gox-fuehrt-verbessertes-Handelssystem-ein-1930505.html. Abgerufen am 01.09.2013.

[45] Axel Kannenberg. Bitcoin-Börse Mt. Gox unter DDoS-Feuer. http://www.heise.de/newsticker/meldung/Bitcoin-Boerse-Mt-Gox-unter-DDoS-Feuer-1835148.html. Abgerufen am 01.09.2013.

[46] Axel Kannenberg. Bitcoin-Börsen: Abhebungssperre bei Mt. Gox. http://www.heise.de/newsticker/meldung/Bitcoin-Boersen-Abhebungssperre-bei-Mt-Gox-1894323.html. Abgerufen am 01.09.2013.

[47] Axel Kannenberg. Bitcoin-Börsen: Bitfloor und Bitcoin24 geschlossen. http://www.heise.de/newsticker/meldung/Bitcoin-Boersen-Bitfloor-und-Bitcoin24-geschlossen-1844920.html. Abgerufen am 01.09.2013.

[48] Axel Kannenberg. Bitcoin-Börsen: Gelöste Sperren und das Ringen um Zulassungen. http://www.heise.de/newsticker/meldung/Bitcoin-Boersen-Geloeste-Sperren-und-das-Ringen-um-Zulassungen-1911136.html. Abgerufen am 01.09.2013.

[49] Axel Kannenberg. Bitcoin24: Staatsanwälte ermitteln wegen Betrug. http://www.heise.de/newsticker/meldung/Bitcoin24-Staatsanwaelte-ermitteln-wegen-Betrug-1850884.html. Abgerufen am 01.09.2013.

[50] Axel Kannenberg. Marktkapitalisierung: Bitcoin knackt Milliarden-Grenze. http://www.heise.de/newsticker/meldung/Marktkapitalisierung-Bitcoin-knackt-Milliarden-Grenze-1832819.html. Abgerufen am 01.09.2013.

[51] Axel Kannenberg. US-Bundesrichter erkennt Bitcoin als Währung an. http://www.heise.de/newsticker/meldung/US-Bundesrichter-erkennt-Bitcoin-als-Waehrung-an-1931729.html. Abgerufen am 01.09.2013.

[52] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. *IACR Cryptology ePrint Archive*, 2012:248, 2012. informal publication.

[53] Guy Grandjean Kate Conelly. Bitcoin: the Berlin streets where you can shop with virtual money. http://www.theguardian.com/technology/2013/apr/26/bitcoins-gain-currency-in-berlin. Abgerufen am 01.09.2013.

[54] Jan-Peter Kleinhans. Auch PaysafeCard bannt VPN Provider. https://netzpolitik.org/2013/auch-paysafecard-bannt-vpn-provider/. Abgerufen am 01.09.2013.

[55] knockmeyer. SHA-2.svg. http://de.wikipedia.org/wiki/Datei:SHA-2.svg. Abgerufen am 01.09.2013, Lizenz CC-BY-SA 3.0.

[56] Gilles Lopez. US-Bundesbehörde stellt Bitcoin unter Geldwäschegesetz. http://www.heise.de/newsticker/meldung/US-Bundesbehoerde-stellt-Bitcoin-unter-Geldwaeschegesetz-1826290.html. Abgerufen am 01.09.2013.

[57] Prepaid Services Company Ltd. sichere Prepaidkarte zum Bezahlen im Internet. https://www.paysafecard.com/de-de/produkte/paysafecard/. Abgerufen am 05.09.2013.

[58] Erik Martin. Independence. http://blog.reddit.com/2011/09/independence.html. Abgerufen am 01.09.2013.

[59] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 397–411, Washington, DC, USA, 2013. IEEE Computer Society.

[60] Dirk Mölleken. *Bitcoin. Geld ohne Banken - ist das möglich?* Diplomica Verlag GmbH, 2012. Diplomarbeit, Originaltitel: Bitcoin, eine neue Art von Geld.

[61] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Originales Bitcoin-Paper, 2008.

[62] National Institute of Standards and Technology. Secure Hash Standard (SHS). FIPS Publication 180-4, March 2002.

[63] Franz Nestler. Deutschland erkennt Bitcoins als privates Geld an. http://www.faz.net/aktuell/finanzen/devisen-rohstoffe/digitale-waehrung-deutschland-erkennt-bitcoins-als-privates-geld-an-12535059.html. Abgerufen am 01.09.2013.

[64] Bitcoin Project. About Bitcoin. http://bitcoin.org/en/about. Abgerufen am 05.09.2013.

[65] Ole Reißmann. Online-Währung: Warum Bitcoin Liberty Reserve ersetzen könnte. http://www.spiegel.de/netzwelt/netzpolitik/online-waehrung-warum-bitcoin-liberty-reserve-ersetzen-koennte-a-902496.html. Abgerufen am 01.09.2013.

[66] Peter Mahlmann und Christian Schindelhauer. *Peer-to-Peer-Netzwerke*. Springer Verlag, 2007.

[67] Klaus Schmeh. *Kryptografie - Verfahren, Protokolle, Infrastrukturen (5. Aufl.)*. dpunkt.verlag, 2013.

[68] Jan Schüßler. Bitcoin-Börse Mt. Gox führt Identitätsprüfungen ein. http://www.heise.de/newsticker/meldung/Bitcoin-Boerse-Mt-Gox-fuehrt-Identitaetspruefungen-ein-1874193.html. Abgerufen am 01.09.2013.

[69] Jan Schüßler. US-Behörde sägt Bitcoin-Handel zwischen Dwolla und Mt. Gox ab. http://www.heise.de/newsticker/meldung/US-Behoerde-saegt-Bitcoin-Handel-zwischen-Dwolla-und-Mt-Gox-ab-1863301.html. Abgerufen am 01.09.2013.

[70] Jan Schüßler. US-Regulierer plädiert für Bitcoin-Kontrolle. http://www.heise.de/newsticker/meldung/US-Regulierer-plaediert-fuer-Bitcoin-Kontrolle-1858753.html. Abgerufen am 01.09.2013.

[71] Ben Schwan. Litecoin und PPCoin statt Bitcoin. http://www.heise.de/newsticker/meldung/Litecoin-und-PPCoin-statt-Bitcoin-1842503.html. Abgerufen am 01.09.2013.

[72] Volker Briegleb Sylvia Beckers. Leben nur mit Bitcoins. http://www.heise.de/newsticker/meldung/Leben-nur-mit-Bitcoins-1922195.html. Abgerufen am 01.09.2013.

[73] Mt.Gox Co. Ltd Team. August 2013 Mt. Gox Status Update. https://www.mtgox.com/press_release_20130805.html. Abgerufen am 01.09.2013.

[74] Mt.Gox Co. Ltd Team. Trade with confidence on the world's largest Bitcoin exchange! https://www.mtgox.com/. Abgerufen am 01.09.2013.

[75] theymos. Anonymity. https://bitcointalk.org/index.php?topic=241.0. Abgerufen am 22.06.2013.

[76] unbekannt. BITCOINCZ Mining aka Slush's pool. http://mining.bitcoin.cz/. Abgerufen am 01.09.2013.

[77] unbekannt. Tor: Overview. https://www.torproject.org/about/overview.html.en. Abgerufen am 05.09.2013.

[78] Bert Ungerer. Bitcoin24-Geld teilweise wieder verfügbar. http://www.heise.de/newsticker/meldung/Bitcoin24-Geld-teilweise-wieder-verfuegbar-1874705.html. Abgerufen am 01.09.2013.

[79] Lucia Weiß. Berliner Läden testen digitale Währung. http://www.tagesspiegel.de/wirtschaft/internet-geld-bitcoin-berliner-laeden-testen-digitale-waehrung/8276274.html. Abgerufen am 01.09.2013.