# Wie funktioniert eigentlich Bitcoin?

 Dennis Schulmeister-Zimolong, dhbw@windows3.de

## 1 Hinweis

Dies ist eine stark vereinfachte Darstellung, wie die Kryptowährung Bitcoin funktioniert. Es werden nur die grundlegenden Konzepte erklärt, um zu zeigen, dass Bitcoin auf einer Peer-To-Peer-Netzwerkstruktur sowie der kryptographischen Sicherheit von Hashalgorithmen aufbaut. Für genauere Informationen siehe die Links am Ende des Dokuments.

## 2 Grundbegriffe der Kryptographie

Verschlüsselung: Ein umkehrbarer Algorithmus, um einen Klartext mithilfe eines *Schlüssels* so abzuändern, dass er für unberechtigte Dritte keinen Sinn ergibt. Im Idealfall verhalten sich die verschlüsselten Daten wie zufälliges Rauschen, damit keine Rückschlüsse auf den Ursprungstext gezogen werden können. Wird zum Beispiel ein Ursprungstext aus dem Alphabet A-Z auf ein Chiffrat mit demselben Alphabet A-Z abgebildet, sollten im Ergebnis alle Buchstaben mit der gleichen Wahrscheinlichkeit auftreten und es darf nicht passieren, dass derselbe Eingangsbuchstabe häufiger zum selben Ausgangsbuchstaben führt. Denn sonst könnte ein Angreifer durch Häufigkeitsanalysen große Teile des Ursprungstextes rekonstruieren, da bestimmte Buchstabenkombinationen in der natürlichen Sprache häufiger vorkommen als andere.

Symmetrische Verschlüsselung: Die Entschlüsselung erfolgt mit demselben Schlüssel wie die Verschlüsselung. Die Sicherheit basiert allein auf der Geheimhaltung des Schlüssels, der daher auch *Shared Secret* genannt wird. Die Kommunikationspartner sollten den Schlüssel *Out-Off-Band* austauschen, sprich nicht über denselben Kanal wie die zu verschlüsselnden Daten.

Asymmetrische Verschlüsselung: Zum Verschlüsseln wird ein anderer Schlüssel benötigt, als zum Entschlüsseln. Welchen der beiden Schlüssel man beim Verschlüsseln verwendet spielt keine Rolle, solange man das Chiffrat stets mit dem anderen Schlüssel entschlüsselt. Die Schlüssel müssen so gewählt werden, dass es in akzeptabler Zeit nicht möglich ist, den einen Schlüssel aus dem anderen abzuleiten.

Public Key/Private Key: Bei der Verwendung eines asymmetrischen Verschlüsselungsalgorithmus besitzt jeder Kommunikationsteilnehmer ein Schlüsselpaar bestehend aus *Public Key* und *Private Key*. Der Private Key ist streng geheim zu halten und darf niemals aus der Hand gegeben werden. Der Public Key hingegen wird mit der gesamten Welt öffentlich geteilt. Will man jemandem eine sichere Nachricht schicken, verschlüsselt man sie mit dem Public Key des Empfängers, da somit nur er die Nachricht mit seinem Private Key wieder entschlüsseln kann.

Hash: Eine mathematische Funktion, die einen Klartext auf einen viel kürzeren Wert, den *Hashwert*, abbildet. Hashfunktionen werden auch Einwegfunktion oder Falltürfunktion genannt: In die eine Richtung sind sie einfach zu berechnen, man kommt aber nur extrem schwer wieder auf den Ursprungswert zurück. Da durch die Hashfunktion Informationen verloren gehen, gibt es immer mehrere zulässige Eingangswerte, die zum selben Hashwert führen. Man spricht dann von Kollisionen. Gute Hashfunktionen sind so gewählt, dass Kollisionen möglichst selten auftreten und dass bereits eine kleine Änderung am Eingangswert zu einer drastischen Abweichung im Hashwert führt.

Digitale Signatur: Ein mit dem Private Key des Absenders verschlüsselter Hashwert einer Nachricht, mit dem unerlaubte Modifikationen an der Nachricht erkannt werden können, wodurch die *Integrität* der Nachricht sichergestellt wird. Da die Signatur mit dem Private Key des Absenders gebildet wird, kann nur der Originalsender eine zum Inhalte passende Signatur bilden. Der Empfänger kann die Nachricht prüfen, indem er selbst den Hashwert ausrechnet und ihn mit der entschlüsselten Signatur vergleicht.

## 3 Struktur der Bitcoin P2P-Netzwerks

Ein Bitcoin wird in 100 Millionen *Satoshi* als kleinste Geldeinheit unterteilt, zu Ehren seines fiktiven Erfinders Satoshi Nakamato. Tatsächlich ist nicht bekannt, wer sich hinter diesem Pseudonym verbirgt. Die Bitcoin-Software ist komplett Open-Source.

Die Idee hinter Bitcoin ist, dass es keine zentrale Instanz gibt, durch welche die Währung kontrolliert wird. Es gibt also keine zentrale „Notenbank“, der man Kraft ihres Gewichts gezwungen ist zu vertrauen. Stattdessen existiert ein P2P-Netzwerk, in dem sämtliche Geldflüsse (*Transaktionen* genannt) allen Teilnehmern bekannt sind. Jede Geldbewegung ist daher öffentlich und kann von jedem Teilnehmer des Netzwerks verifiziert werden. Dies funktioniert, da alle Transaktionen vom Anbeginn der Zeit in der sog. *Blockchain* gespeichert werden, die prinzipiell für alle Teilnehmer verfügbar ist. Die Blockchain ist somit das globale Buchungsjournal der Bitcoin-Währung.

Jeder Teilnehmer des Bitcoin-Netzwerks besitzt mindestens ein Schlüsselpaar bestehend aus *Public Key* und *Private Key*. Der Public Key wird auch *Account Key* genannt, da er sozusagen die „Kontonummer“ des Teilnehmers ist. Über eine *Wallet* (engl. Geldbeutel) genannte Software werden die Kontostände verwaltet, wobei sich der Kontostand stets aus der Summe der ein- und ausgehenden Zahlungen ermittelt.

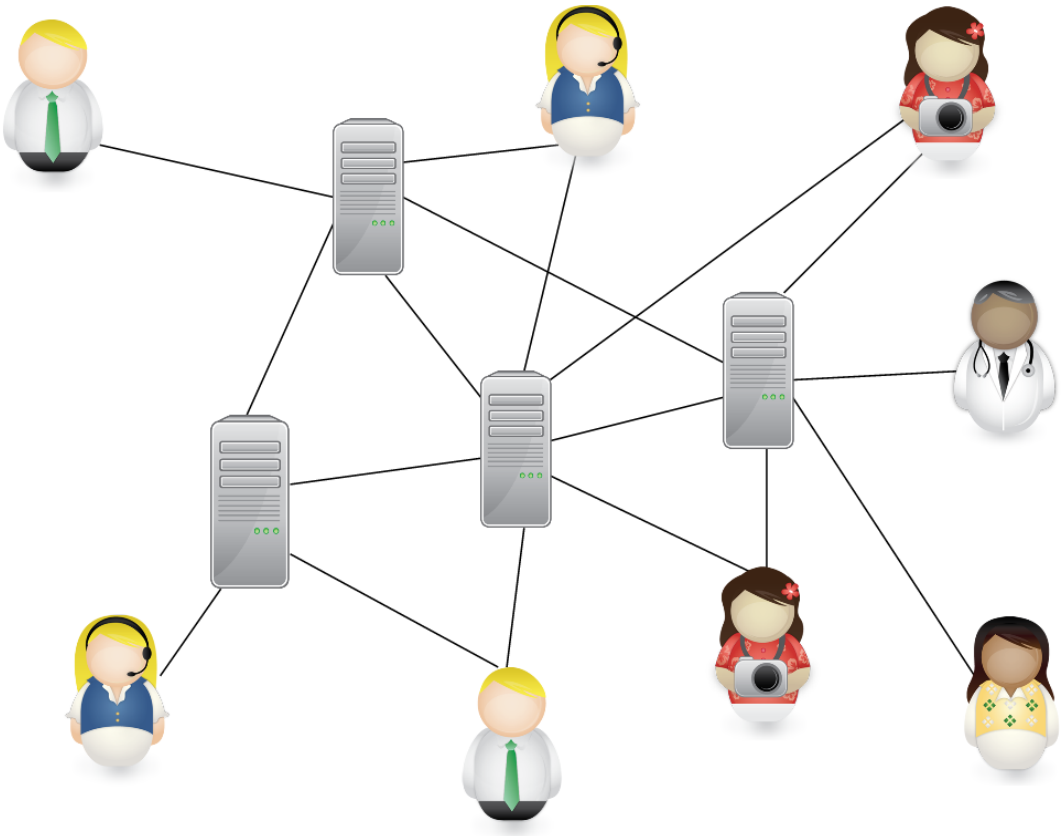

Abb. 1: Peer-To-Peer-Struktur des Bitcoin-Netzwerks

Im Prinzip werden zwei Arten von Teilnehmern unterscheiden: Die normalen Anwender, die untereinander Geldbeträge austauschen und die „Hauptknoten“, die eine Verifikation der Transaktionen vornehmen. Technisch gesehen macht das keinen Unterschied, jeder Teilnehmer kann zum „Hauptknoten“ aufsteigen, indem er beginnt die Transaktionen der anderen Teilnehmer zu überprüfen. Aufgrund der hierfür benötigten Rechenleistung ist dies mit normalen Desktopcomputern jedoch nicht machbar. Jeder Hauptknoten besitzt hierfür eine vollständige Kopie der Blockchain und somit die gesamte Buchungshistorie (stand 02/2017 rund 100 GB). Um zum Hauptknoten aufzusteigen besteht der erste Schritt daher darin, sich die Blockchain zu besorgen und tatsächlich alle darin enthaltenen Transaktionen zu validieren, bevor man damit anfangen kann, neue Transaktionen zu prüfen.

Will ein Anwender nun einen Geldbetrag „überweisen“, erstellt er hierfür eine Transaktion, in der Absender, Empfänger und der überwiesene Geldbetrag enthalten sind[1], und schickt diese an das P2P-Netzwerk. Dabei wird über einen *Floodfill*-Algorithmus sichergestellt, dass möglichst alle Hauptknoten die Transaktion empfangen und diese unabhängig voneinander prüfen. Die Transaktion gilt als ausgeführt, wenn die Mehrheit sie bestätigt, wobei eine Bestätigung innerhalb von 10 Minuten angestrebt wird. Dieses *Konsensverfahren* hat den Vorteil, dass man keiner zentralen „Bank“ vertrauen muss, ob die Transaktion ist und es somit auch keinen zentralen Angriffspunkt gibt. Stattdessen baut man darauf, dass „eine Millionen Fliegen sich nicht irren können“. Dies macht das System allerdings gegen die sog. *51%-Attacke* verwundbar, da ein Angreifer, der mindestens 51% der Rechenleistung auf sich vereint, immer eine Mehrheit erzielen kann. Warum das Vertrauen in die breite Masse trotzdem besser als das klassische Bankenmodell ist, sehen wir bald.

## 4 Authentizität und Integrität beim Geldtransfer

Entgegen des Namens gibt es bei Bitcoin eigentlich gar keine virtuellen Münzen. Es gibt also keinen Datensatz, der eine Münze darstellt und der wie im Falle des physischen Geldes von einem Sender zu einem Empfänger übergehen kann. Stattdessen gibt es einfach nur *Transaktionen*, die dokumentieren, dass ein Sender einen bestimmten Geldbetrag an einen Empfänger übertragen will. In diesem Sinne gibt es auch keinen Kontostand, sondern nur eine Summe aller jemals empfangenen Geldbeträge abzüglich der selbst getätigten Zahlungen. Man kann sich das ein wenig so vorstellen, als wäre das Bargeld abgeschafft und es gäbe nur noch Onlinebanking[2].

Wie wir später noch sehen werden, gibt es spezielle Transaktionen, durch die neues Geld entsteht. Sie entstehen beim sogenannten *Mining*, wenn einer der Hauptknoten die Blockchain um einen weiteren Block fortschreiben kann, dadurch die in diesem Block aufgezeichneten Transaktionen bestätigt und sich dafür einen festen Geldbetrag gutschreiben darf. Die maximale Geldmenge ist jedoch auf 21 Millionen Bitcoin und da ein Bitcoin 100 Millionen Satoshi entspricht, somit auf 2,1 Billiarden diskrete Geldeinheiten begrenzt. Zumindest theoretisch kann es daher auch keine unbegrenzte Inflation geben.

Um sicherzustellen, dass ein Anwender nicht mehr Geld ausgibt, als er besitzt (Bitcoin kennt keinen Überziehungskredit), muss jede ausgehende Transaktion durch zuvor eingegangene Transaktionen gedeckt werden. Wann immer also ein Anwender eine Zahlung empfängt, wird diese Transaktion als *Unspent Transaction* (engl. noch nicht ausgegebene Transaktion) im Wallet abgelegt und kann zu einem späteren Zeitpunkt verwendet werden, um das Geld wieder auszugeben.

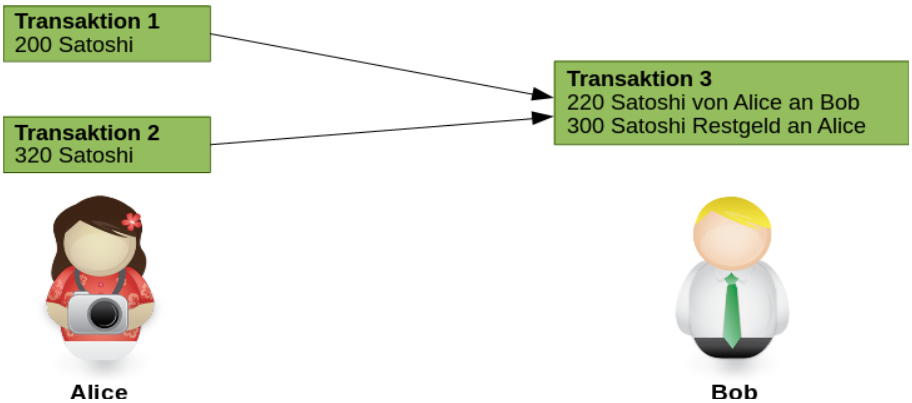

Abb. 2: Bezahlung von Alice an Bob

In der obigen Abbildung besitzt Alice zwei *Unspent Transactions* im Gesamtwert von 520 Satoshi, wobei sie nun 220 Satoshi an Bob bezahlen will. Da keine der beiden Transaktionen ausreicht, um diesen Betrag zu de-

1 Der Einfachheit halber betrachten wir nur Transaktionen mit einem Empfänger. Bitcoin erlaubt allerdings auch innerhalb einer Transaktion unterschiedliche Geldbeträge an mehrere Empfänger zu senden.

2 Was tatsächlich in der Politik schon öfters diskutiert wurde. Erste Länder haben auch bereits einen Teil ihres Bargelds abgeschafft.

cken, werden sie beide für die Bezahlung herangezogen. Technisch gesehen werden die Transaktionen dabei verknüpft, wobei man sagt, dass der Ausgang von Transaktion 1 und 2 mit dem Eingang von Transaktion 3 verbunden ist. Da der Wert der beiden Ausgangstransaktionen den zu zahlenden Betrag aber übersteigt, wird das Restgeld automatisch an Alice zurückgebucht. Empfänger der Transaktion 3 sind daher sowohl Alice als auch Bob, obwohl Alice auch der Sender der Transaktion ist.

Nach Abschluss der Transaktionen ergibt sich dann das folgende, neue Bild. Die beiden Ursprungstransaktionen werden nun als *Spent Transactions* (engl. bereits verbrauchte Transaktionen) gekennzeichnet und können nicht mehr verwendet werden. Stattdessen beinhalten beide Konten nun die neue Transaktion 3, die als *Unspent Transaction* für weitere Zahlungen zur Verfügung steht.

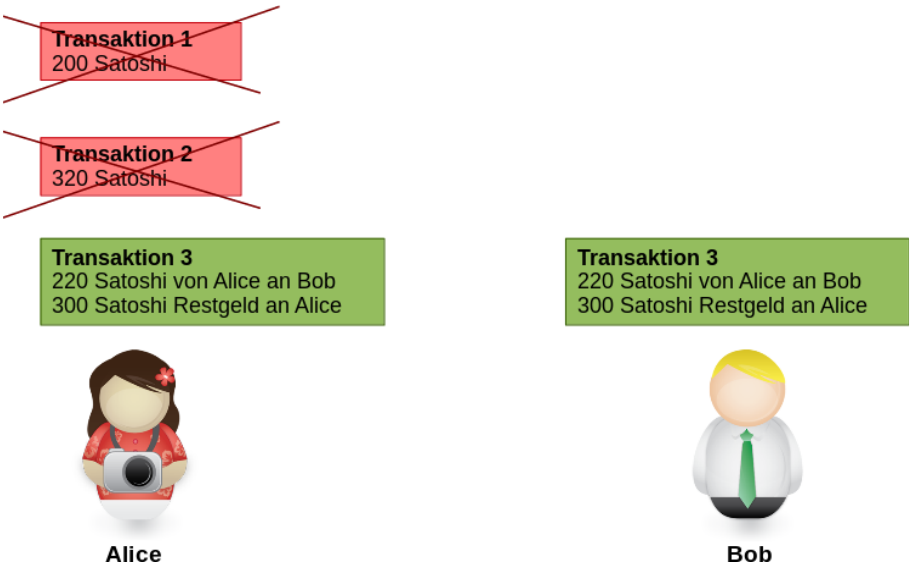


Abb. 3: Transaktionen nach Abschluss der Zahlung

Im Laufe der Zeit ergibt sich dadurch eine immer länger werdende Kette von zusammenhängenden Transaktionen, die sozusagen einen Geldfluss über mehrere Teilnehmer hinweg aufzeichnet:

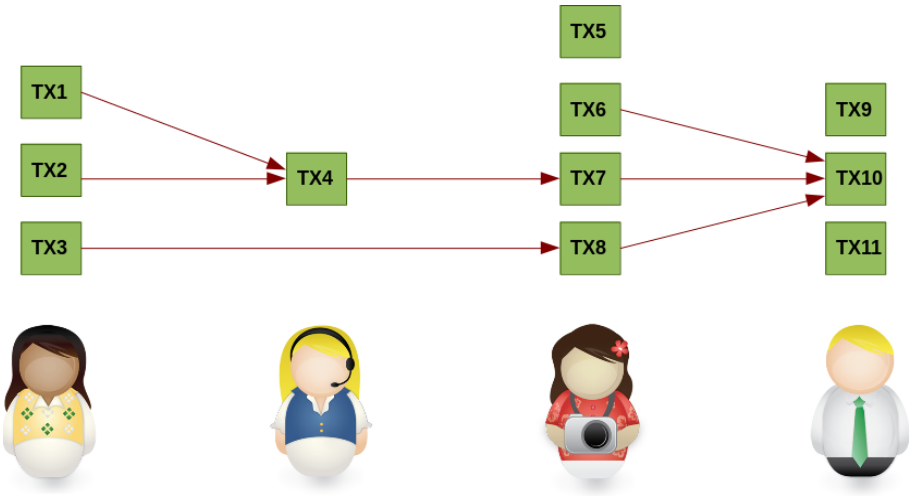


Abb. 4: Zusammenhängende Kette von Transaktionen

Die untenstehende Abbildung zeigt in drei Schritten, wie die Transaktionen nun in folgender Hinsicht kryptographisch abgesichert werden:

1. Um das System vor Manipulationen zu schützen, muss jede Transaktion vom Absender signiert werden. Da eine Signatur einfach nur ein mit dem eigenen Private Key verschlüsselter Hashwert ist,

kann dadurch sowohl der Sender einer Transaktion (*Authentizität*) sowie ihr Inhalt (*Integrität*) überprüft werden. Alles was man hierfür tun muss, ist die Signatur mit dem Public Key des Senders zu entschlüsseln und mit einem selbst gerechneten Hashwert zu vergleichen.

2. Des weiteren will man nicht nur den Sender sondern auch den Empfänger einer Transaktion validieren. Aus diesem Grund wird in der Transaktion auch der Public Key des Empfängers gespeichert und somit in die Signatur miteinbezogen.

3. Zusätzlich will man auch den zeitlichen Verlauf der Transaktionen fälschungssicher belegen. Dies wird erreicht, indem in der Transaktion die Hashes der Eingangstransaktionen enthalten sind, so dass diese ebenfalls in die Signatur einfließen.

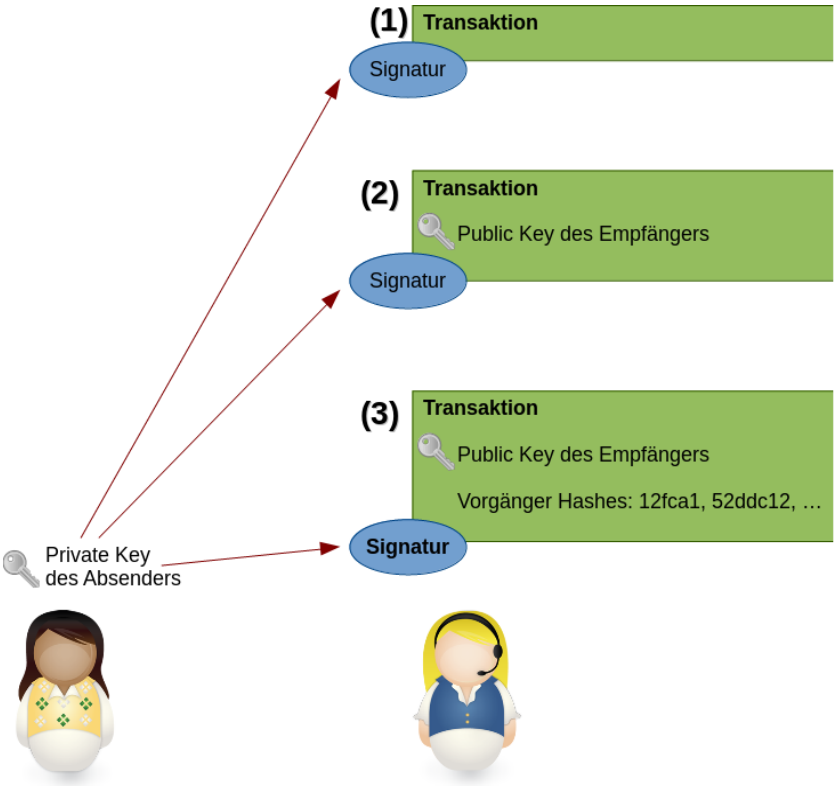


Abb. 5: Signierung einer Transaktion (vereinfacht)

Als Hashfunktion kommt dabei SHA256 zum Einsatz. Sie stellt sicher, dass jede nachträgliche Änderung zu einer abweichenden Signatur führt. Selbst wenn die Transaktionsdaten selbst nicht böswillig verändert, sondern nur im zeitlichen Verlauf anders eingereiht wird, ändert sich dadurch ihre Signatur, da sie nun gegen andere Vorgängerhashes gerechnet werden muss und sich der neue, veränderte, eigene Hashwert auf alle Folgetransaktionen auswirkt. Das Verfahren stellt also tatsächlich auch die zeitliche Abfolge der aufgezeichneten Transaktionen sicher.

## 5 Aufzeichnung und Verifikation aller Transaktionen in der Blockchain

Durch das eben gezeigte Verfahren können Geldflüsse über beliebig viele Zwischenschritte fälschungssicher dokumentiert werden. Denn jede nachträgliche Änderung einer Transaktion führt dazu, dass sich ihre Signatur verändert. Und da die Signatur (Hashes) aller Eingangstransaktionen in die Signaturbildung einbezogen werden, ändern sich dadurch automatisch auch die Signaturen aller nachfolgenden Transaktionen:

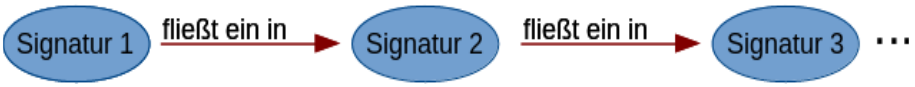


Abb. 6: Miteinander verkettete Signaturen

Allerdings hat das Verfahren auch einen entscheidenden Nachteil: Da die Transaktionsdaten nur den beteiligten Teilnehmern bekannt sind, ist keine echte Kontrolle der Vorgänge möglich. Insbesondere kann das *Double Spending*-Problem nicht verhindert werden, also dass ein böswilliger Teilnehmer mehr Geld ausgibt, als er eigentlich hat. Denn niemand kann kontrollieren, ob es sich bei einer Transaktion, die als Eingang für eine Zahlung verwendet wird, auch tatsächlich um eine *Unspent Transaction* handelt, oder ob ein Betrüger nicht einfach dieselbe Transaktionen immer wieder in unendlich vielen Zahlungsvorgängen nutzt.

In klassischen Finanzwesen hat die Bank bzw. das Zahlungsinstitut die Kontrollfunktion inne und als Kunde der Bank bleibt einem nichts anderes übrig, als ihr zu vertrauen. Dabei passieren der Bank durchaus auch mal Fehler. Nicht umsonst gibt es bei Monopoly die Ereigniskarte „Bankirrtum zu Ihren Gunsten“.

Bitcoin löst das Problem hingegen durch sein P2P-Netzwerk, in dem jeder Zahlungsvorgang veröffentlicht und in der Blockchain als zentralem „Kontenbuch“ verewigt wird. Dabei wird eine Zahlung bereits bei der Aufnahme in die Blockchain auf Richtigkeit geprüft:

- Beinhaltet die Transaktion eine gültige Signatur?
- Ist die ∑ Eingehende Transaktionen ≥ dem Zahlbetrag?
- Wurden die Eingangstransaktionen wirklich noch nicht verbraucht?

Jeder Hauptknoten nimmt seine Prüfungen unabhängig von den Knoten vor und nimmt die Transaktion nach erfolgreicher Prüfung in die Blockchain auf. Absender und Empfänger warten derweil, bis sie von einem oder mehrerer Knoten ein positives Prüfergebnis erhalten. Die Prüfungen erfolgen daher automatisch nach einem Viele-Augen-Prinzip. Da die Blockchain aber auch so öffentlich zugänglich ist, kann der Zahlungsempfänger jede Zahlung eigenständig nochmal überprüfen.

Im Prinzip gelten für die Blockchain ähnliche Anforderungen wie für die einzelnen Transaktionen. Das heißt, es muss sichergestellt werden, dass die in der Blockchain befindlichen und daher bereits geprüften Transaktionen nicht unbemerkt verändert werden können und dass ihre zeitliche Reihenfolge nicht verändert werden kann. Zusätzlich muss aber noch bedacht werden, dass die Prüfknoten völlig unabhängig voneinander arbeiten und kein Knoten Rücksicht darauf nimmt, was ein anderer macht. Um zu verstehen, warum alle Knoten dennoch ein einheitliches Verständnis von der Blockchain haben und warum es nicht zu Inkonsistenzen kommt, muss man sich anschauen, wie die Blockchain aufgebaut ist und wie das Einfügen neuer Einträge funktioniert: Der Name lässt ja schon vermuten, dass es sich um eine „Kette von Blöcken“ handelt, dass also mehrere Einträge irgendwie aneinandergereiht werden:


Abb. 7: Stark vereinfachte Darstellung der Blockchain

Bei den Blöcken handelt es sich dabei einfach um „Sammelcontainer“, in denen alle einem Knoten bekannten Transaktionen gesammelt werden, die innerhalb eines bestimmten Zeitraums angefallen sind. Das System justiert sich dabei selbst so, dass im Schnitt alle zehn Minuten ein neuer Block entsteht und in die Blockchain aufgenommen wird. Daher dauert auch die Bestätigung einer Zahlung in der Regel nicht länger als zehn Minuten, da die Zahlungstransaktion durch Aufnahme in die Blockchain ja als bestätigt gilt.

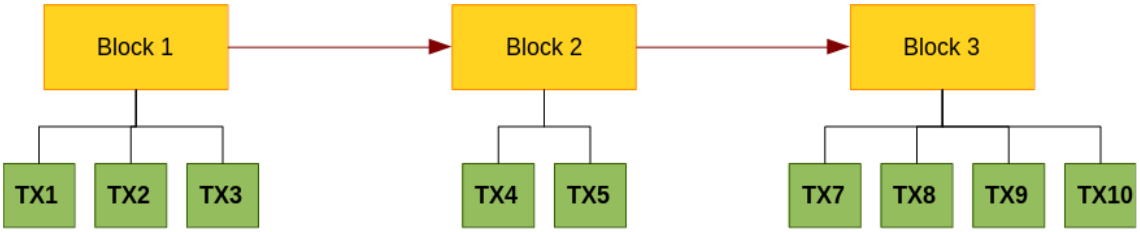


Abb. 8: Blockchain mit Zahlungstransaktionen

Die Integrität eines Blocks wird ähnlich wie bei den Transaktionen durch einen Hashwert sichergestellt, in den alle enthaltenen Transaktionen einfließen. Hierfür wird jede Transaktion einzeln gehasht und die daraus entstehenden Hashwerte werden über einen *Merkle-Hashbaum* zu einem einzigen Hashwert verdichtet. Damit ist gemeint, dass immer die Hashwerte zweier Transaktionen auf einen neuen Hashwert reduziert werden, so lange, bis am Ende nur noch ein Hashwert (*Merkle Root* genannt) übrigbleibt:

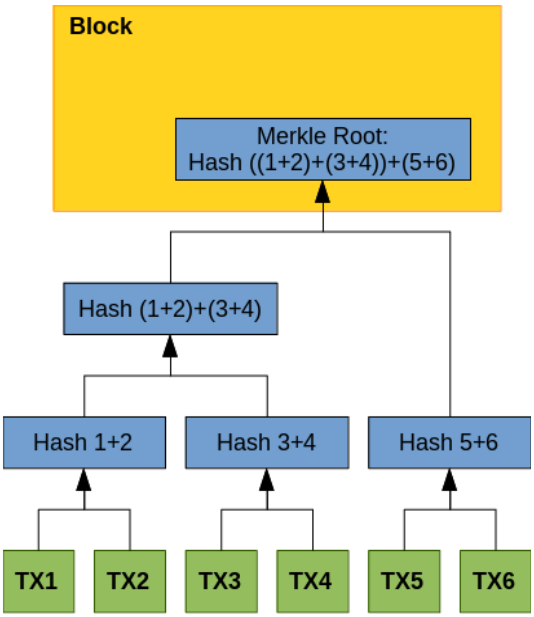


Abb. 9: Hashbaum aller Transaktionen eines Blocks

Das Berechnen des Hashbaums ist eine einfache Aufgabe, die nicht viel Rechenzeit benötigt. Aber nicht jeder Block ist auch ein gültiger Block. Den zusätzlich zu den oben gezeigten Hashes, besitzt ein Block noch folgende weitere Felder:

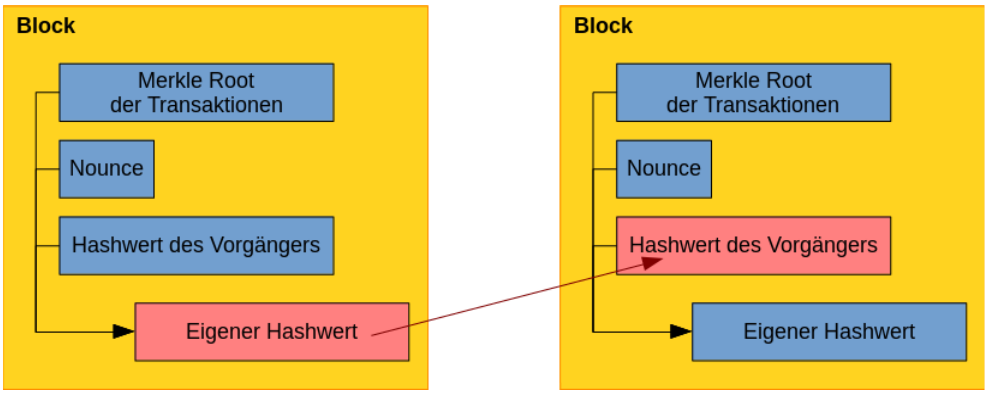


Abb. 10: Vollständiger Aufbau eines Blocks (ohne die Transaktionsdaten)

Wie man sieht, beinhaltet jeder Block die Felder *Merkle Root*, *Nounce* und *Hashwert des Vorgängers*, aus denen sich der eigentliche Hashwert des Blocks berechnet:

- **Merkle Root:** Auf einen Hash reduzierte Transaktionen. Die Transaktionsdaten selbst sind auch im Block gespeichert, für die weitere Berechnung aber nicht mehr erforderlich.
- **Nounce:** Variabler Wert, der solange angepasst wird, bis der Hashwert unterhalb der Challenge liegt
- **Hashwert des Vorgängers:** Verknüpfung zum vorherigen Block, um die zeitliche Konsistenz zu gewährleisten

Die Herausforderung liegt für die Knoten nun darin, solange die *Nounce* zu verändern, bis der Hashwert unterhalb einer gegebenen *Challenge* liegt. Schaffen Sie dies, hängen sie den Block an die Blocklist an und schreiben sich dafür einen gewissen Geldbetrag gut, der als erste Transaktion in den Transaktionsdaten enthalten sein muss. Da dies, je niedriger die Challenge ist, umso unwahrscheinlicher ist, erfordert sich eine enorme Rechenzeit und es ist nicht garantiert, dass ein Knoten überhaupt eine passende Nounce findet. Dieser Vorgang wird auch *Mining* (engl. Schürfen) genannt, da dies der einzige Weg ist, wie „frisches Geld“ in Umlauf kommen kann[3].

Wurde ein gültiger Block gefunden, wird er über das P2P-Netzwerk an alle anderen Knoten kommuniziert. Diese verifizieren den Block (was sehr schnell geht, da die Eingangswerte für die Hashes nun alle bekannt sind) und hängen ihn ans Ende ihrer Blockchain. Dadurch ergibt sich dann auch das oben genannte Konsensverfahren, da jeder Teilnehmer nochmal selbst entscheiden kann, ob er den neuen Block akzeptiert oder nicht. Ein Block kann sich daher nur dann durchsetzen, wenn er von der Mehrheit aller Teilnehmer tatsächlich in die Blockchain aufgenommen wird.

Doch auch hier gibt es ein Problem: Was passiert, wenn zwei Knoten nahezu zeitgleich einen gültigen Block generieren und an die anderen Teilnehmer verteilen? In diesem Fall entsteht ein sog. *Fork*, da es nun zwei oder mehr gültige Endstücke der Blockchain gibt:

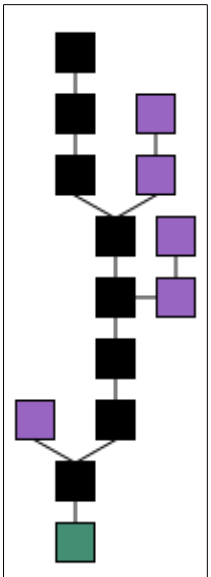

Abb. 11: Gabelungen in der Blockchain

Die Lösung des Problems ist allerdings einfacher als gedacht: Jeder Teilnehmer entscheidet selbst, mit welchem Endstück er weiterarbeiten will, um nach neuen Nouncen zu schürfen. Allgemein anerkannt ist aber, dass der längste zusammenhängende Pfad von Anfang bis Ende (in der Abbildung schwarz) als die allgemein vertrauenswürdige Blockchain gilt, da sie von mehr Teilnehmern als alle anderen Abzweige geprüft wurde.

3 Da die Rechenleistung laut Moore‘s Law ständig steigt, wird die Challange alle zwei Wochen angepasst und die gutgeschriebene Betrag für einen Erfolg halbiert sich alle 2016 Blöcke.

Ein kürzerer Pfad wird in der Regel daher einfach verworfen, da es unwahrscheinlich ist, dass er seinen Rückstand wieder aufholen kann und dadurch vertrauenswürdig wird. Daraus folgt, dass die schürfenden Teilnehmer in Konkurrenz zueinander stehen, weil jeder ein Interesse hat, selbst derjenige zu sein, der den nächsten gültigen Block liefert und die daraus resultierende Belohnung verbuchen darf.

Zusammengefasst ergibt sich daraus auch, warum ein Angreifer mindestens 51% der Gesamtrechenleistung braucht, um das System erfolgreich zu kompromittieren. Die rückwirkende Änderung einer Transaktion ist für den Angreifer wirtschaftlich unrentabel, da dadurch nicht nur der Block, in dem sie enthalten ist, sondern alle darauffolgenden Blöcke ungültig werden (da der Hashwert des einen Blocks in den Hashwert des nächsten Blocks einfließt). Der Angreifer müsste also die komplette Kette aber ab der manipulierten Transaktion neurechnen. Da die Kette aber spätestens alle zehn Minuten länger wird, sinkt die Wahrscheinlichkeit dies zu schaffen, exponentiell mit der Zeit. Die einzige Chance, die er hat, ist es mehr als 50% der Rechenknoten zu kontrollieren, um dadurch immer den nächsten Block der Blockchain zu kompromittieren und diesen vor den ehrlichen Teilnehmern zu berechnen. Andernfalls handelt es sich um eine Aufholjagd, die er praktisch nicht gewinnen kann.

## 6 Zusammenfassung

Ausgehend von den einfachen Definitionen ein paar grundlegender Begriffe der Kryptographie haben wir gezeigt, wie Bitcoin sicherere Zahlungen ermöglichen kann, die im Gegensatz zum herkömmlichen Geldwesen folgende Sicherheitsmerkmale aufweisen:

- Geldflüsse können fälschungssicher nachvollzogen werden
- Sender und Empfänger einer Zahlung können verifiziert werden, so dass ein Angreifer sich nicht für einen anderen Empfänger ausgeben kann
- Alle Geldflüsse sind öffentlich sichtbar und werden von vielen Augen kontrolliert
- Nur die von der Mehrheit der Teilnehmer validierten Transaktionen werden als „Single Truth“ anerkannt
- Angreifer haben keinen zentralen Ansatzpunkt zur Kompromittierung des Systems

Im Gegensatz dazu hat das System natürlich seine eigenen Schwachstellen, wie zum Beispiel der enorme Rechenbedarf und der damit zusammenhängende Energieverbrauch. Auch die Netzwerkauslastung und der Speicherbedarf sind ein zunehmend wichtiger werdendes Thema. Und letztlich können Fehler in der Software auch als eine Art Single Point Of Failure gewertet werden, über die das System verwundbar ist.

## 7 Weiterführende Links

https://bitcoin.org/bitcoin.pdf

https://bitcoin.org/en/developer-documentation

https://de.wikipedia.org/wiki/Bitcoin

https://de.wikipedia.org/wiki/Kryptow%C3%A4hrung

https://de.wikipedia.org/wiki/Blockchain