Passwords

A brief summary of solutions to the never dying problems of pins.

Basierend auf dem Buch *Bitcoin ohne Vorkenntnisse* von Benjamin Spahic, sowie diversen Online-Artikeln.

Version 0.1
Febuary 2023
Team HTL Anichstraße

no © by Lilo Zobl

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Abstract

2 Introduction

Password. Everyone has one or rather multiple. But it's a faulty construction. Nowadays passwords are everything but a good invention.
These are the reasons: i) Passwords are not user-friendly. In order to make passwords securer providers make them complexer. User have to write them down or use the same one over and over so they don't forget it. ii) Passwords are not secure because they can be shared and guessed. Of course a lot of passwords are also stolen, which is a lot easier than anyone would think. iii) Passwords are weak and get reused. Here are some examples to the most frequently used ones of 2022: "password", "123456", "123456789", "guest", "qwerty", ect.
Luckily there is a websites that lets users check if their account information was breached (that's when your email connected to a password is leaked; always a huge amount of accounts get breached from a branch such as Facebook). The website is called "haveibeenpwned.com". HIBP is open source and secure. If someone searches for a breach it doesn't store the data that was typed in (an email address or phone number). It only ever retrieves the data from storage then returns it in the response.

3 Haupteil?

Passwords have always been a security threat ever since they where introduced. Providers have tried to find solutions such as requiring users to make the pin at least eight characters long or use upper case and lower case letters. Sometimes they require you to include digits or symbols. However those ßolutions"didn't improve the situation overall and sometimes made it worse. To eliminate this security threat a few solutions created.

3.1 Password Save

3.1.1 Explanation

A password save allows you to save a list of your usenames linked to the password. It has one master password that looks the save. Instead of remembering a ton of password at once users only have to recall one single pin. The rest is done by the app or rather the software.

3.1.2 Providers

- **KeepassXC**
  This software is developed for users with extremely high demands of secure personal data management.
  KeePassXC uses Advanced Encryption Standard (AES) encryption algorithm with a 256-bit key to secure the password database.
  The biggest difference to other password saves is that the data (password, account information and additional data such as URLs, attachments ans notes) is stored in an offline, encrypted file that can be stored locally. This prevents your data from getting leaked when the could gets hacked or breached.
  The program is customizable. It allows its user to customize literally everything to their needs.

- **1Passwort**
  This password save is known for its easy to use interface and high security encryption.
  1Password uses an uncommon encryption known as dual-key encryption. If the server gets breached it's impossible for the hacker to decrypt the users sensitive information because of its two keys. The first key is the users master-key. The second key is a secret key, which is a 128-bit, machine-generated code. The secret key is generated on every device you log into. It will only be saved on your devices and never saved with your other pins.
  Depending on what account type you choose your data is stored differently. However in only one version users have the option to save their data locally. In every other option the user's data gets saved in a cloud-based vault.
  The interface is very user-friendly. It even creates strong passwords for accounts that are newly created.

- **Password Safe (MATESO)**

- **LastPass**

- **Dashlane**

3.2 2-factor authentication

3.3 The future of passwords

4 conclusion

5 Grundlagen der Kryptographie

Durch bekannte, erprobte Verschlüsselungsalgorithmen werden Transaktionen verschlüsselt, durchgeführt und bestätigt.

„Die Verschlüsselung funktioniert. Richtig implementierte starke Kryptosysteme sind eines der wenigen Dinge, auf die wir uns verlassen können". - Edward Snowden

5.1 Die drei Ziele der Kryptographie

- **Integrity (Datenintegrität)**
  Wie können wir sicher sein, dass Daten unverfälscht sind?
  Mit sogenannten Hashing-Funktionen (für mehr Details siehe Kapitel 5.2 auf Seite 3)
  Aktueller Standard: Secure Hash Algorithm 2 (SHA2); Beispiel: Wird auch oft bei Downloads angeboten um zu überprüfen, ob das heruntergeladene File mit dem am Server übereinstimmt.

- **Authentication (Authentifizierung)**
  Wie können wir sicher sein, dass die Daten auch tatsächlich vom vermutlichen Absender stammen?
  Mit keyed Hashes (siehe Kapitel 5.4 auf Seite 4) oder asymmetrischer Verschlüsselung (siehe Kapitel 5.5.2 auf Seite 4). Beispiel Algorithmen: HMAC-MD5, HMAC-SHA-1, RSA und DSA;

- **Confidentiality (Vertraulichkeit)**
  Wie können wir sicher sein, dass die Daten von niemand anderem gelesen werden können?
  Mit klassischer Verschlüsselung (symmetrisch oder asymmetrisch). Beispiel Algorithmen: DES, 3DES, AES, SEAL, Elliptic-Curve;

5.2 Datenintegrität - Hashing

Hashes werden verwendet um die **Integrität** einer Nachricht oder eines Files sicherzustellen. Hashing basiert auf einer **mathematischen Einweg-Funktion** - einfach durchzuführen aber schwer umzudrehen, wie zum Beispiel das Kaffee mahlen: Es ist einfach aus Kaffeebohnen Kaffeepulver zu mahlen, aber aus dem Pulver wieder die Bohnen zusammensetzen ist eher schwierig ;-)

Prinzip:

Aus einer beliebig langen Nachricht oder einem File-Inhalt wird durch hashing ein Hash-Value mit einer fixen Länge erzeugt. Beim Hashing wird der ursprüngliche Inhalt in mehreren Stufen verschoben, vertauscht, komprimiert oder ergänzt. Wie genau ist immer vom verwendeten Hashing-Algorithmus abhängig.
Am Ende dieser Prozedur entsteht ein immer gleich langer (bei SHA256 z.B: 256 bit) Hash-Wert.
Jede noch so kleine Änderung in der Nachricht oder dem File erzeugt ein unterschiedliches Hash-Value. Hashes werden deshalb auch oft als digitale Fingerabdrücke bezeichnet.

Fig. 1: Hashing Function

5.3 SHA

Hashing Algorithmen werden laufend weiterentwickelt. Der aktuell am weitesten verbreitete Algorithmus ist **SHA**.
SHA steht für Secure Hash Algorithm, und wurde vom National Institute of Standards and Technology (NIST) entwickelt; Erste Version stammt von 1994, aktuelle Versionen wurden im August 2002 (SHA-2-Familie) vorgestellt und sind SHA-256, SHA-384, SHA-512. Die nächste Generation (SHA-3-Familie) wurde 2015 vorgestellt.

SHA-256 hat auch für die Funktionsweise von Bitcoin eine große Bedeutung. Mehr dazu sie Kapitel 7.4 auf Seite 7.

File-Hash berechnen unter Linux - Darstellung in hexadezimaler Schreibweise!

```
sha256sum ./main.jpg
a7f837e28843fba098c0929037743d28edf6a97ed74ff17127aa329423acf52c  ./main.jpg
```

File-Hash berechnen unter Windows in der Power-Shell

```
Get-FileHash C:\Users\user1\Downloads\file.txt -Algorithm SHA256 | Format-List

Algorithm : SHA256
Hash      : 3CBCFDDEC145E3382D592266BE193E5BE53443138EE6AB6CA09FF20DF609E268
Path      : C:\Users\user1\Downloads\file.txt
```

Für mehr Informationen zur genauen Funktionsweise von SHA-256 siehe:
https://en.wikipedia.org/wiki/SHA-2

5.4 HMAC

Hash Message Authentication Code (HMAC) oder auch manchmal Keyed-hash message authentication code (KHMAC) sind Hashes die noch einen zusätzlichen, geheimen Schlüssel verwenden.

Fig. 2: HMAC - hash bases message authentication

Nur wer die Daten und den geheimen Schlüssel hat kann den richtigen Hash-Wert berechnen.

5.5 Verschlüsselung - Vertraulichkeit

Es gibt verschiedene Verschlüsselungsmethoden. Prinzipiell lässt sich eine Einteilung in symmetrische und asymmetrische Verfahren treffen.

5.5.1 Symmetrische Verschlüsselung

Symmetrisch bedeutet, dass sowohl für die Verschlüsselung als auch für die Entschlüsselung **der gleiche Schlüssel** verwendet wird.

Fig. 3: Symmetrische Verschlüsselung

Symmetrische Verfahren sind meist **schnell** (schneller als asymmetrische Verfahren) und verwenden im Vergleich zu asymmetrischen Verfahren eher kurz Schlüssel.
Die eigentliche Übersetzung kann blockweise erfolgen (z.B. es werden jeweils 128 Bit der Nachricht übersetzt) - dann spricht man von einem **Block Cipher**, oder Bit by Bit - dann spricht man von **Stream Chiper**.

AES (Advanced Encryption Standard) ist zur Zeit (seit dem Jahr 2000) der etablierte Standard für symmetrische Verschlüsselung. Es ist ein Blockcipher mit 128 Bit Länge und die Schlüssellänge kann 128, 192 oder 256 Bit betragen.
Für mehr Info zu AES siehe:
https://de.wikipedia.org/wiki/Advanced_Encryption_Standard

Das eigentliche **Problem** bei symmetrischen Verfahren liegt im **Schlüsselaustausch**! Wie schafft man es über ein unsicheres Medium einen geheimen Schlüssel auszutauschen?
Aber auch dafür gibt es Lösungen, z.B. den Diffie-Hellmann Key Exchange, aber dies sprengt den Rahmen dieses Skripts. Mehr dazu gibt es im Fach Netzwerktechnik!

5.5.2 Asymmetrische Verschlüsselung

Wird auch **Public Key Encryption** genannt.
Die asymmetrische Verschlüsselung umgeht das Problem des Key-Exchanges in dem es pro Teilnehmer **zwei Schlüssel** gibt: einen Public Key und einen Private Key. Die Besonderheit an den Keys ist, dass eine Message, die mit einem der Keys verschlüsselt wurde, nur mit dem anderen Key entschlüsselt werden kann! Der öffentliche Schlüssel und der private Schlüssel hängen über mathematische Funktionen zusammen. Es ist jedoch nicht möglich, von dem öffentlichen Schlüssel auf den privaten Schlüssel zurück zu schließen.

Fig. 4: Asymmetrische Verschlüsselung

Es gibt mehrere asymmetrische Verschlüsselungsverfahren z.B. RSA oder ECC. Sie sind in der Regel deutlich langsamer als symmetrische Verfahren, umgehen aber das Problem des Schlüsselaustauschs und erlauben zusätzlich zur Confidentiality auch noch Integrity und Authentication!

- **Public Key (Encrypt) + Private Key (Decrypt) = Confidentiality**
  Schickst Du mir etwas mit meinem Public Key, kann nur ich es mit meinem Private Key lesen!
- **Private Key (Encrypt) + Public Key (Decrypt) = Authentication**
  Schicke ich dir etwas mit meinem Private Key, kann es nur von mir gekommen sein, weil nur ich den Private Key habe - lesen kann es jeder, der meinen Public Key hat. Dies nennt man auch eine **signierte** Nachricht oder **digitale Signatur**!

- **Confidentiality, Authentication und Integrity?** - geht auch:
  Ich schicke dir eine Nachricht mit deinem Public Key, aber zusätzlich bilde ich noch einen Hash der Nachricht mit meinem Private Key! Du entschlüsselst die Nachricht mit deinem Private Key und überprüfst den Hash mit meinem Public Key - damit ist sicher gestellt, dass die Nachricht von mir ist und nicht verändert wurde (wenn der Hash übereinstimmt).

5.5.2.1 ECC - Eliptic Curve Cryptography

Bitcoin verwendet ECC um aus einem Private Key (256 Bit Länge - meist als Zufallszahl erzeugt) einen Public Key zu erzeugen.
Achtung: Es gibt eine technische Einschränkung: Der Private Key muss zwischen `0` und `n-1` liegen wobei `n=1,1578*10^77` ist und mit der elliptischen Kurve zu tun hat (Order der Kurve) die Bitcoin verwendet.
Der Public Key wird durch **Elliptische Kurvenmultiplikation** mit der Formel $K = k * G$ erzeugt. Wobei k der Private Key ist. G der *generator point* - ein Punkt auf der elliptischen Kurve und K der resultierende Public Key. Achtung: Der Operator $*$ ist keine normale Multiplikation, sondern eine elliptische Kurvenmultiplikation!
Die Besonderheit dabei ist, das K aus k und G relativ einfach berechnet werden kann, aber die Umkehrfunktion, nämlich k aus K und G zu berechnen, nahezu unmöglich ist.
Bitcoin verwendet eine elliptische Kurve aus dem `secp256k1` Standard, der vom amerikanischen NIST (National Institute of Standards and Technology) vorgestellt wurde.

Elliptische Kurven Multiplikation Im folgenden wird eine vereinfachte Kurve verwendet, um das Prinzip der elliptischen Kurven Multiplikation zu erklären - sie entspricht nicht der von Bitcoin verwendeten Kurve!
ECC-Multiplikation bedeutet, dass man die Tangente (differenzieren - erste Ableitung) zum Ausgangspunkt G nimmt und schaut, wo sich diese mit der Kurve kreuzt ($-2G$). Dann springt man auf die andere Seite der Kurve und erhält $2G$. Weitere Multiplikation ergibt $4G$, $8G$ usw.

Fig. 5: ECC-Multiplikation veranschaulicht

Der Private Key gibt also an, wie oft ECC-Multiplikationen durchgeführt werden müssen, um den Public Key zu erhalten. z.B.: Private Key = 2 führt zu Public Key = 4G, Private Key = 3 führt zu Public Key 8G, usw.

6 Die Entstehung von Bitcoin

Nach der Finanzkrise 2008 hat eine Person oder ein Kollektiv, unter dem Pseudonym **Satoshi Nakamoto**, eine digitale, dezentrale und begrenzte Währung erschaffen: den Bitcoin. Es ist bis heute ungeklärt, wer Satoshi Nakamoto wirklich ist.

Link zum ursprünglichen White-Paper:
https://bitcoin.org/bitcoin.pdf

Link zum Bitcoin-Sourcecode auf Github:
https://github.com/bitcoin/

Die Grundlage der Digitalen Währung sind:

- **Transparenz**
  alle Transaktionen werden in einer Blockchain aufgezeichnet und sind seit der ersten Transaktion einsehbar.
- **Sicherheit**
  die Blockchain ist im Nachhinein nicht mehr (nur sehr schwer) änderbar.
- **Dezentralisierung**
  die Kopien der Blockchain liegen verteilt im Netz (Full-Nodes) und werden regelmäßig aktualisiert.
- **Inflationssicherheit**
  es gibt nur eine begrenzte Anzahl an Bitcoins (21 Millionen)

6.1 Bitcoin, mBTC und Satoshi - die Währungseinheiten

Ähnlich wie es beim Euro noch Euro-Cents gibt, gibt es auch bei Bitcoin (**BTC**) kleinere Einheiten: Den Milli-Bitcoin (**mBTC**) und den **Satoshii**. Es gilt:

```
1 BTC = 1000 mBTC = 100.000.000 Satoshii
```

7 Blockchain

Im folgenden wird der Aufbau einer Blockchain am Beispiel von Bitcoin erklärt, das Prinzip kann aber auf beliebige Daten (nicht nur Bitcon-Transaktionen) angewandt werden.

7.1 Aufbau der Blockchain

Die Blockchain besteht aus Blöcken. Der **erste Eintrag** eines Blocks ist immer ein **Verweis** auf den **vorherigen Block**. Dadurch ist die Verbindung zum vorherigen Block festgelegt und die Blöcke verknüpfen sich zu einer Kette.
Bei Bitcoin enthält ein Block eine Reihe von Transaktionen die in einem bestimmten Zeitraum angefallen sind.
Die Blockgröße ist bei Bitcoin ursprünglich auf 1 MByte limitiert (ist inzwischen auf 4MByte angehoben, aber die durchschnittliche Blockgröße ist bei ca. 1,13 MByte). Dadurch können pro Block ca. 2000 Transaktionen durchgeführt werden. Über die gesamten Daten eines Blocks wird mittels SHA-256 ein Hash gebildet.

Fig. 6: Hash-Generierung aus den Daten eines Blocks

Ändern wir nur ein Zeichen des gesamten Blocks, ändert sich der Hash vollständig und unvorhersehbar. Eine einzige Änderung in nur einer Transaktion würde den kompletten Hash des Blocks und damit alle weiteren Blöcke verändern. Somit ist der Hash des aktuellen Blocks gleichzeitig ein eindeutiger Verweis auf diesen Block. Der Hash des aktuellen Blocks wird als erster Eintrag des Folgeblocks eingesetzt. So entsteht eine Reihenfolge von Blöcken - die Blockchain.

Fig. 7: Aneinanderreihung von Blöcken

Der erste Block in der Blockchain ist vorgegeben und wird **Genesisblock** genannt.

7.2 Dezentralisierung, Full-Nodes und Größe

Dezentralität bedeutet, dass es keine zentrale Institution wie z.B. eine Bank oder einen Staat gibt, der über ein System (die Blockchain) wacht. Stattdessen liegen viele Kopien der Blockchain verteilt in einem großen Netzwerk. Jeder Teilnehmer des Netzwerks, welcher die **volle Transaktionshistorie** (also die gesamte Blockchain) hostet, wird **Full-Node** genannt.
Damit jeder Knotenpunkt des Netzwerks (Full-Nodes) die aktuelle Blockchain besitzt, wird die Blockchain in regelmäßigen Abschnitten aktualisiert und an jeden Teilnehmer übermittelt. Nach einem gewissen zeitlichen Abschnitt, bei Bitcoin alle zehn Minuten, wird ein Paket mit neuen Informationen, ein sogenannter Block, an die bisherige Transaktionshistorie angehängt. Das hat den Vorteil, dass nicht jede einzelne Transaktion aktualisiert werden muss, sondern ein Bündel von Transaktionen.
Die zehn Minuten werden durch den Bitcoin Algorithmus gesteuert, indem die Mining-Difficulty angepasst wird. Mehr dazu siehe im Kaptiel 7.4 auf Seite 7. Die **momentane Größe** (Oktober 2022) der Bitcoin-Blockchain liegt bei ca. 432 GByte. Zum Startzeitpunkt von Bitcoin (03.01.09) war sie 285 Byte groß.

Fig. 8: Wachstum der Bitcoin Blockchain seit 2009

7.3 Konsensbildung in einem dezentralisiertem System

Wenn jeder Full-Node eine Kopie der Blockchain hat, und sie theoretisch verändern kann, wer hat dann die *richtige* Blockchain?
Gültige Blöcke werden nur durch das rechenintensive **Mining** (siehe Kapitel 7.4 auf Seite 7) erschaffen. So vertraut jeder Bitcoin-Node der **längsten gültigen Blockchain**, da hinter dieser die meiste Rechenleistung steht und deswegen auch die Mehrheit der Teilnehmer vermutet wird. Man kann also vereinfacht sagen: *Die Mehrheit bestimmt was wahr ist.*
Dies kann unter bestimmten Umständen problematisch werden - siehe *51% Angriff* im Kapitel 10.1.3 auf Seite 11.

7.3.1 Konsensmechanismen

Es gibt mehrere Konsensverfahren. Die zwei bekanntesten sind:

7.3.1.1 Proof-of-Work

Der Zweck eines **Arbeitsnachweises** (Proof-of-work) ist es sicherzustellen, dass das Erzeugen gültiger Blöcke mit einem gewissen Aufwand verbunden ist, so dass eine nachträgliche Modifikation der Blockkette praktisch ausgeschlossen werden kann (weil es für einen einzelnen Angreifer zu rechenintensiv wäre).
Der Proof-of-Work besteht bei Bitcoin darin, einen SHA256 Hashwert für einen Block zu finden, der unterhalb eines bestimmten Schwellwerts (muss mit einer bestimmten Anzahl von Nullen beginnen) liegt. Der Schwellwert ist variabel um bei steigender Hash-Leitung im Netzwerk trotzdem den gleichen Zeitaufwand (ca. 10 Minuten) für die Validierung zu benötigen. Mehr Details dazu im Kapitel 7.4 auf Seite 7.

7.3.1.2 Proof-of-Stake

Beim Proof-of-Stake wird eine **gewichtete Zufallsauswahl** eingesetzt. Die Wahrscheinlichkeit für die Validierung des nächsten Blocks ausgewählt zu werden, hängt vom Coin-Vermögen (dem „Stake“) und/oder der Teilnahmedauer der einzelnen Teilnehmer ab. Es ist ein wenig mit einer Aktiengesellschaft vergleichbar - wer mehr Aktien besitzt hat mehr Stimmrecht bei Entscheidungen.
Im Gegensatz zum, bei Bitcoin eingesetzten Proof of Work, kommt Proof of Stake ohne zeit- und energieintensives Mining aus. Zudem ist nicht möglich, das Netzwerk allein durch Besitz von Rechenleistung zu übernehmen.

7.3.2 Fork

Was passiert, wenn es zu Unstimmigkeiten oder Regeländerung kommt, und manche Nodes einen unterschiedlichen Block an ihre lokale Chain anhängen? In diesem Fall spricht man von einem **Fork**. Es bedeutet, dass die Blockchain im verteilten Netz nicht mehr einheitlich ist!

Fig. 9: Ein Fork - die Blockchain ist nicht mehr einheitlich

Dies kann zwei Ursachen haben:

1. **Zufällig:** Wenn zwei Miner fast zeitgleich einen passend Hash für einen Block finden und diesen im Netzwerk verteilen. Aufgrund der endlichen Verteilungsgeschwindigkeit bekommen Full-Nodes unterschiedliche Blöcke, die sie aber erfolgreich verifizieren und an ihre lokale Blockchain anhängen.
   Es gibt also gleich lange, aber unterschiedliche Chains im Netzwerk.

   Diese Forks lösen sich aber meist innerhalb von wenigen (meist 5-6) weiteren Blöcken wieder auf - weil bis dahin wieder eine längste Chain existiert, von der man annimmt, dass sie die richtige Chain ist. Sie wird wieder auf alle Full-Nodes verteilt.

   Fig. 10: Ein Fork löst sich wieder auf - die Blockchain ist wieder einheitlich

2. **Absichtlich:** Wenn es zu Regeländerungen im Sourecode kommt, z.B. wenn es Unstimmigkeit in der Community gibt. In diesem Fall spricht man von einem **Hard-Fork**. Ab diesem Zeitpunkt trennt sich das Netzwerk auf, manche Teilnehmer übernehmen die Regeländerung und manche nicht. Es entsteht dadurch eine neue Blockchain-Community wie z.B. beim Split von Ethereum und Ethereum-Classic im Juli 2016.

7.4 Mining

Das Mining hat zwei wichtige Aufgaben. Zum einen die Verifizierung eines Blocks und zum anderen erzeugt es neue Coins.

7.4.1 Verifizierung - Hashing

Die Miner berechnen den Hash des aktuellen Blocks. Allerdings gibt der Bitcoin Algorithmus vor, dass der Hash mit einer bestimmten Anzahl von Nullen beginnen muss. Da der Hash nur geändert werden kann, wenn die Daten des Blocks geändert werden, wird dem Block ein weiteres Element hinzugefügt, das als **Nonce** bezeichnet wird.
Der Miner startet nun z.B: mit einer Nonce von 1 und berechnet die SHA256 Summe des Blocks. Wenn der resultierende Hash nicht die geforderte Anzahl von Anfangs-Nullen aufweist, erhöht er die Nonce auf 2 und berechnet wieder den Hash, usw. Es kann viele Billionen Berechnungen (Proof-of-work) brauchen, bis eine akzeptable Hashsumme gefunden wird.

Fig. 11: Hashgenerierung aus den Daten eines Blocks inkl. der zusätzliche Variable **Nonce**

Seit dem Bekanntwerden der Kryptowährungen gibt es Firmen die spezielle Mining (Hashing-) Geräte herstellen. Die wahrscheinlich bekannteste Firma ist www.bitmain.com. Selbst kleine Geräte (Antminer) um wenige Tausend Dollar haben schon dreistellige **TH/s** (Terahashes pro Secunde) Werte.

Wenn ein Miner nun den passenden Hash gefunden hat, schickt er den gelösten Block, inklusive seiner gefundenen **Nonce** an alle **Full-Nodes**. Diese kontrollieren, ob ein Hash mit der geforderten Anzahl an Nullen entsteht (eine einmalige SHA256 Berechnung). Ist das der Fall, wird der Block an die Blockchain angehängt und der Miner erhält seine **Belohnung**.

7.4.2 Miner Belohnung und Transaction Fees

Was ist die Motivation, das sehr rechen- und damit leider sehr energieintensive Mining durchzuführen? Warum macht das ein Miner, was hat er oder sie davon?
Es gibt zwei Verdienstmöglichkeiten:

7.4.2.1 Transaction Fees

Im Bitcoin Netzwerk können nur ca. 3,3 Transaktionen pro Sekunde durchgeführt werden (Blocksize = 1 MByte -> ca. 2000 Transaktionen pro Block -> alle 10 Minuten ein neuer Block -> 2000/600 = 3,3). Wenn man nun eine dringende Transaktion machen möchte, kann man eine Transaktionsgebühr anbieten, damit die Miner die Transaktion in den aktuellen Block aufnehmen.
Ein Miner der einen Block löst (den richtigen Hash findet), bekommt alle Transaktionsgebühren der im Block enthaltenen Transaktionen.
Die anderen Miner gehen leer aus. Danach stürzen sich alle Miner sofort auf den nächsten Block und machen sich wieder auf die Suche nach der richtigen Nonce.
Die Transaktionsgebühren waren in der Vergangenheit starken Schwankungen unterworfen und lagen zeitweise über 60 USD pro Transaktion.

Fig. 12: Transaktionsgebühren (Blau) und Bitcoin Marktwert (Schwarz)

7.4.2.2 Belohnung - Block Subsidy

Zusätzlich zu den Transaktionsgebühren erhält der Miner, der den Block löst, eine Belohnung (neue Coins) die im Programm code verankert ist und je nach Kryptowährung unterschiedlich ist. Diese Belohnung nennt man Block Subsidy (Block-Subvention).
Bei Bitcoin war der Anfangswert der Block Subsidy 50 Bitcoins pro gefundenem Block. Da das Lösen eines Blocks im Mittel 10 Minuten dauert, wurden also alle 10 Minuten 50 neue Bitcoins in Umlauf gebracht.
Der Subsidy Wert halbiert sich allerdings alle 210 000 Blöcke (**Halving**), was einem Zeitraum von ca. 4 Jahren entspricht. Momentan (November 2022) gibt es *"nur"* noch 6,25 Bitcoins pro gelöstem Block.

Zeitraum	Block Subsidy	Neue Coins im Zeitraum	Alle Coins im Umlauf
2008-2012	50	10.500.000 BTC	10.500.000 BTC
2012-2016	25	5.250.000 BTC	15.750.000 BTC
2016-2020	12,5	2.625.000 BTC	18.375.000 BTC
2020-2024	6,25	1.312.500 BTC	19.687.500 BTC
2024-2028	3,125	656.250 BTC	20.343.750 BTC
2028-2032	1.5625	328.125 BTC	20.671.875 BTC
.	.	.	.

Tab. 1: **Halving** - die Block Subsidy halbiert sich alle 4 Jahre

Wie man aus Tabelle 1 auf Seite 8 erkennen kann, nimmt die Anzahl der neu ausgeschütteten Coins ab und wird sich einen Wert von 21 Millionen BTC annähern.

8 Teil der Blockchain werden

Welche Schritte sind nun notwendig um selbst Teil der Blockchain zu werden?

8.1 Meine Adresse auf der Blockchain

Eine **Adresse** auf der Blockchain ist vergleichbar mit einer Konto Nummer, mit dem Unterschied, dass die Adresse öffentlich und für jeden Teilnehmer einsehbar ist! Jeder Teilnehmer kann durch Rückschau auf vergangene Transaktionen berechnen welche Adresse wie viele Bitcoins enthält!
Man kann z.B. auf https://www.blockchain.com/btc/address/ nachsehen wie der "Kontostand"von Satoshi Nakamotos erster Adresse (hat mehrere) zur Zeit (November 2022) aussieht:

Fig. 13: "Kontostand"der Genesis Adresse - erste Adresse von Satoshi Nakamoto

Die **Adresse** auf der Bitcoin Blockchain ist in Wirklichkeit der leicht abgewandelte **öffentliche Schlüssel (public key)** eines asymmetrischen Schlüsselpaares. Siehe Kapitel 5.5.2 auf Seite 4 zur Erinnerung wie Puplic Key Encryption funktioniert.
Der public key wird aus einem zufällig gewählten private key (bei Bitcoin 256 Bit Länge) generiert. Aus diesem wird dann (mittels Hashfunktionen - zuerst SHA256 und dann RIPEMD160) die öffentliche Adresse (160 Bit = 20 Byte Länge) erstellt.

8.2 Wallets

Wallets sind kleine Programme die bei der Key-Erstellung und Verwaltung (eine Wallet kann mehrere Keys für mehreren Crypto-Währungen verwalten) helfen.

> ACHTUNG: Die Wallets sind nicht Vergleichbar mit einer Geldtasche oder einem Konto, weil sie selber keine Coins verwalten, sondern nur die Adressen! Die Blockchain selbst ist mit ihren transparenten Transaktionen das "Konto".

8.2.1 Arten von Wallets

8.2.1.1 Hot Wallet

Von einer Hot Wallet spricht man, wenn die Wallet online bei einer Handelsplattform (Exchange) liegt. Zum Beispiel bei www.bitpanda.com.
Dies ist zwar bequem (Online Zugriff via Browser) aber sehr unsicher, da schon einige Exchanges gehackt worden sind oder zahlungsunfähig wurden oder der Betreiber alle Keys gestohlen hat und sich auf die Bahamas abgesetzt hat :(

8.2.1.2 Cold Wallet

Die Wallet ist offline. Man unterscheidet noch zwischen:

- **Software Wallet** z.B. auf dem privaten PC Zuhause oder auf einem verschlüsselten USB-Stick. Ist sicher, aber Achtung wenn z.B. der USB Stick verloren geht oder der PC versehentlich entsorgt wird - siehe James Howell, der im Jahr 2013 eine Festplatte mit Keys zu ca. 7500 BTC auf den Müll geworfen hat ;-)
- **Hardware Wallets** sind spezielle Hardware und Software zur Key-Sicherung. Sie z.B. Firma www.ledger.com
- **Paper Wallet** die Keys werden mit open source Software einmalig erzeugt und nur analog (z.B: auf einem Blatt Papier) gespeichert. Sehr sicher gegen online Angriffe aber sorgfältige Verwahrung ist ein Muss!

8.2.2 Key Erstellung

Wie schon erwähnt hilft die Wallet bei der Key-Erstellung.
Als erstes wird der **private Key** (256 Bit) erstellt, entweder durch einen halb automatischen Algorithmus mit User Interaktion (z.B: Mouse Bewegungen) oder durch einen Zufallszahlengenerator.
Aus diesem wird mit Hilfe einer **elliptischen Kurven-Funktion** (siehe Kapitel 5.5.2.1 auf Seite 5) der zugehörige (unkomprimierte) **public Key** generiert.
Aus diesem wird dann (mittels Hashfunktionen - zuerst SHA256 und dann RIPEMD160) die öffentliche Adresse (160 Bit = 20 Byte Länge) erstellt.

Fig. 14: Generierung der Bitcoin-Adresse

8.2.3 Eine Transaktion senden

Eine weitere Aufgabe die von den Wallets erledigt wird, ist das Senden von Transaktionen.
Nehmen wir an, Alice möchte Bob einen Bitcoin (1 BTC) senden.
Die gewünschte Transaktion lautet dann: Von der öffentlichen Bitcoin-Adresse von Alice soll ein Bitcoin an die öffentliche Adresse von Bob geschickt werden.
Damit sichergestellt wird, dass diese Transaktion wirklich von Alice ausgeht, **signiert** sie sie mit ihrem **Private Key**. Siehe dazu Kapitel 5.5.2 auf Seite 4.
Diese signierte Transaktion (plus eine Transaktionsgebühr - siehe Kaptiel 7.4 Mining auf Seite 7) wird von der Wallet an alle umliegenden Miner und Nodes gesendet, welche sie wiederum weitersenden, bis jeder im Netzwerk diese Transaktion sehen kann.
Die Nodes überprüfen, ob der Absender genug Coins im Besitz hat (History der Blockchain).
Wenn die Transaktionsgebühr hoch genug ist, wird die Transaktion in den nächsten Block mit aufgenommen und wenn er voll ist (es passen ca. 4200 Transaktionen in einen Block) wird mit dem Mining begonnen.

Fig. 15: Abfolge beim Senden einer Transaktion

9 Stablecoins - Coins zu Dollar

Stablecoins werden verwendet um Crypto-Coins in eine reale Fiat-Währung wie den US-Dollar umzuwandeln.
Die Krypto-Tauschbörse Bitfinex hat eine Kryptowährung erschaffen, die den Wert von einem US-Dollar abbildet. Dafür wurden eine Milliarde US-Dollar auf einem Bankkonto hinterlegt und eine Milliarde sogenannte **Tether** (USDT) zum Kauf angeboten.
Die Firma hinter Tether garantiert, dass wir für den Tether-Coin bei Bedarf auch einen US-Dollar ausbezahlt bekommen.
Bitfinex verdient nun, wie eine Bank, beim Tausch zwischen verschiedenen Coins und der Auszahlung (Rückkauf von Tether).
Kursschwankungen werden versucht auszugleichen:
Wenn die Coin-Nachfrage zu niedrig ist, gleicht Tether die Kursschwankungen durch Kaufen des Coins aus. Ist die Coin-Nachfrage hingegen zu hoch, wirkt Te ther dem entgegen, indem neue Coins ausgegeben werden, sodass der Nominalwert von 1 USD eingehalten werden kann.

Wichtig bei Stablecoins ist, wie auch im Papiergeldsystem, dass die Währung gedeckt ist. Gedeckt bedeutet, dass zu jedem Tether-Coin auch ein entsprechender US-Dollar auf einem Bankkonto eingefroren ist. Nur so kann der Wert von Tether sichergestellt werden. Stellen wir uns vor, es gäbe eine Milliarde Tether-Coins, aber es sind nur eine Million Dollar auf einem Bankkonto hinterlegt.
Aktuelles Beispiel (November 2022) wo dies gerade nicht mehr der Fall ist: FTX. Siehe FTX Zahlungsunfähigkeit

10 Probleme

Bitcoin und die Crypto-Währungen werden häufig als das Zahlungsmittel der Zukunft verkauft. Es soll anonym, frei von Mittelsmännern (Banken) sein und keine Inflation aufweisen.
Anstelle einer stabilen Währung haben sie sich aber bis jetzt (November 2022) eher als hochspekulativ und instabil erwiesen.

10.1 Hohe Volatilität

Zu den grundlegenden hohen Kurschwankungen aufgrund des völlig unregulierten 24 Stunden Handels kommen noch Euphorie oder Panik hinzu. Zum Teil sind Tweets von Promis ausreichend, um massive Kursschwankungen auszulösen.

10.1.1 Mittelsmänner

Es braucht nach wie vor Mittelsmänner, statt Banken sind es nun **Crypto-Exchanges**. Auf Grund von fehlenden Regulierungen sind sie viel anfälliger gegen Missbrauch, gezielte Angriffe und Zahlungsausfall als traditionelle Banken.
Beispiel vom November 2022: FTX - Zahlungsausfall und Diebstahl

10.1.2 Fairness

Auch von Fairness kann keine Rede sein, weil in beiden Konsensmechanismen (Proof-of-Work und Proof-of-Stake) immer die *'starken'* oder *'die schon viel besitzen'* belohnt werden.

10.1.3 51% Angriff

Bei Coins, die Proof-of-Work als Konsensmechanismus nutzen, kann eine Gruppe die über 51% der gesamten Hashing-Power des Netzwerks besitzt, das Netzwerk komplett übernehmen. Sie sind schneller im Minen als alle anderen, und erzeugen so die längste Blockchain, welche von allen anderen Nodes als wahre Blockchain wahrgenommen wird.
Dies ist bei großen Coins wie Bitcoin zwar schwierig, aber nicht unmöglich. So erzielte der Mining-Pool GHash.IO im Juni 2014 über einen Zeitraum von 24 Stunden einen Anteil von etwa 55% der Bitcoin-Hashrate.
Auf kleinere Coins werden aber immer wieder mit 51% Angriffe durchgeführt.

10.1.4 Transaktionslimit

ca. 3,3 Transaktionen pro Sekunde weil 1 MByte Blockgröße mit ca. 2000 Transaktionen und 10 min. Zeit für einen Block!
Führt im August 2017 zum Hard-Fork und der Abspaltung Bitcoin-Cash (hat nun 8 MB Blockgröße).
Zum Vergleich: Visa wickelt ca 65000 Transaktionen pro Sekunde ab, das sind fast 150 Millionen Transaktionen pro Tag!

10.1.5 Finality

Finality beschreibt, ab wann ein Teilnehmer der Blockchain sicher sein kann, dass seine Transaktion auch Teil der Blockchain - und somit wahr - ist. Da es ja immer wieder zu Forks kommen kann und sich die längste Blockchain nach meist 5-6 Blöcken (jeweils 10 Minuten pro Block) durchsetzt. Kann ich als Teilnehmer erst nach 6x10 Minuten sicher sein, ob meine Transaktion auch Teil der Blockchain ist!

10.1.6 Energieaufwand bei Proof-of-Work

Der Cambridge Bitcoin Electricity Consumption Index hat das Mining des Bitcoin auf seinen Energieverbrauch hin untersucht. Zur Zeit (November 2022) liegt der Energieverbrauch bei ca.100 Terawattstunden. Im Vergleich dazu verbraucht Österreich pro Jahr ca. 70 Terawattstunden Energie.