Vom Bankcoin zum Bitcoin

# Grundlagen

Das Ziel ist es hier die Sicherheitsmechanismen im Bitcoin Protokoll leicht verständlich zu erklären. Daher werde ich hier zuerst auf ein paar Grundlagen eingehen. Kryptografische Grundbegriffe wie Hashfunktionen und digitale Signaturen sind Voraussetzung, um zu verstehen wie eine Blockchain funktioniert. Wir auf bekannte Probleme wie die Double-Spending-Attacke stoßen wenn wir versuchen Digitales Bargeld mithilfe von Digitalen Signaturen zu erschaffen.

Wir werden uns zu beginn auch Digitale Timestamps anschauen, weil man mithilfe dieser gut die Sicherheitskonzepte von Hashfunktionen und digitalen Signaturen illustrieren kann. Eine Blokchain ist ja praktisch nichts anderes als eine digitale Timestamp Methode.

Zum Schluss werden wir bei den Grundlagen nochmal auf das Konzept „Proof-of-Work“ eingehen da dieser ein wichtiger Bestandteil der Blockchain ist.

## Hashfunktionen

Mit einer Hashfunktion kann man die Echtheit eines Datensatzes prüfen. Hier ein Beispiel:

Print(hashlib.sha256(b“Diese Dokumentation ist nur eine drei“).hexdigest())

* a709067a6150468297588c67b72202aae237db898f3ea6ba24837bd24fbe1edf

Herr Lehmann hat die Benotung für die Dokumentationen gehasht und sich den Hash schnell per mail geschickt. Danach hat er seinen Rechner an gelassen und Bernhard ist schnell an den Rechner gegangen und hat die Bewertung in „Diese Dokumentation ist eine glatte eins“ geändert. Zuhause will Herr Lehmann das Zeugnis schreiben und hasht sicherheitshalber nochmal seine Bewertungen:

Print(hashlib.sha256(b“Diese Dokumentation ist eine glatte eins“).hexdigest())

* b45b570ac288afb313198be752b74fe0c4eed60cc11d13e69cb00ac8fe95e9ed

Man sieht das sich der Hash geändert hat. Herr Lehmann kann nicht sagen was genau geändert wurde, aber es ist nicht sein ursprünglicher Text.

Bei einer Hashfunktion hat die Menge des Inputs keinen Einfluss auf die länge des Outputs. Man kann die ganze Bibel hashen und hat trotzdem den 256-Bit ausgabewert (meist wird dieser Wert in Hexadezimalwerte umgewandelt).

An eine Hashfunktion werden zwei Sicherheitseigenschaften gestellt. Sie muss eine Einwegfunktion sein und Kollisionsresistent.

Eine Einwegfunktion bedeutet das es quasi unmöglich ist aus dem Hash den ursprünglichen Text zu rekonstruieren.

Kollisionsresistent bedeutet das es quasi unmöglich ist einen Text zu finden welcher den gleichen Hash liefert.

## Digitale Signaturen

Wir werden hier nur kurz auf asynchrone Signaturen eingehen da diese ausführlich in einer anderen Dokumentation erklärt werden.

Herr Lehmann will seine Zeugnisnoten zu Frau Becker schicken. Um sicherzugehen das die Noten wirklich von Herrn Lehmann kommen will er seine Nachricht digital Signieren. Dafür erstellt er sich zuerst ein Schlüsselpaar. Einen PrivateKey den Signierschlüssel, diesen hält er geheim und mit diesem kann er seine Signatur „Das hier ist wirklich von Herrn Lehmann“ verschlüsseln. Den Publickey oder Verifikationsschlüssel lässt Herr Lehmann Frau Becker so zukommen das Sie sich sicher sein kann das dieser von Herr Lehmann kommt. Nur mit Hilfe des Verifikationsschlüssel kann Frau Becker die Nachricht entschlüsseln und wenn ein lesbarer Text rauskommt weiß Sie das die Noten tatsächlich von Herr Lehmann kommen.

## Digitales Bargeld

Regeln für Digitales Bargeld

Geld wird nur direkt vom Zahlenden zum Zahlungsempfänger gegeben ist die wichtigste Regel. Außerdem muss gewährleistet sein:

1. Jeder kann die Echtheit des Geldes bestätigen
2. Nur der Herausgeber kann Geld erschaffen

Mit einer vom Herausgeber Signierten Nachricht wie:

*Diese Geldnote hat Seriennummer 8124 und ist 100€ wert*

Sind beide Kriterien erfüllt. Im Gegensatz zu echtem Bargeld kann digitales Bargeld aber einfach vervielfältigt werden. So kann Frau Becker diese Geldnote einfach an Herr Lehmann und Herr Trutz schicken. Das ist die Double-Spending-Attacke der wir uns noch widmen werden.

## Digitale Zeitstempel

Digitale Zeitstempel dienen dazu die Existenz einer bestimmten Datei zu einem bestimmten Zeitpunkt zu bestätigen.