Bitcoin and Beyond

Frederik Gierschner
Fachbereich Informatik
Hochschule Bonn-Rhein-Sieg
Sankt Augustin, Deutschland
Email: frederik.gierschner@smail.inf.h-brs.de

***Zusammenfassung*—Dezentrale digitale Zahlungssysteme befinden sich spätestens seit der Veröffentlichung von Bitcoin auf dem Vormarsch. Dabei sind sowohl die technischen Grundlagen als auch möglich Angriffe von Interesse. Diese Ausarbeitung geht zunächst auf die Technologien, die solchen Systemen zugrunde liegen ein. Dabei wird Bitcoin als Beispiel genommen, um zu beschreiben, wie Transaktionen gesichert und Besitzansprüche bewiesen werden. Außerdem wird die *Blockchain* anhand ihrer Aufgaben in Bitcoin erklärt. Im Anschluss werden verbreitete Angriffe auf Bitcoin und dazugehörigen Gegenmaßnahmen erläutert. Abschließend werden Weiterentwicklungen und Alternativen zu Bitcoin exemplarisch vorgestellt, wobei auch gezeigt wird, dass Technologien, wie die Blockchain nicht nur für Kryptowährungen eingesetzt werden können.**

I. Einleitung

KONZEPTE für digitale Währungen gibt es schon seit über 30 Jahren. Häufig blieb es allerdings bei einem Konzept, da für bestimmte Teilaspekte keine Lösungen gefunden wurden oder die Währungen keine Akzeptanz fanden. Anders verhält es sich mit Bitcoin, welches im November 2008 durch Satoshi Nakamoto vorgestellt wurde [1]. Aktuell besitzt Bitcoin eine Marktkapitalisierung von ca. 14,4 Milliarden US-Dollar und ist damit die stärkste Kryptowährung vor Ethereum mit ca. 933 Millionen US-Dollar Marktkapitalisierung (Stand 09.01.2017) [2]. Ab dem 26.10.2016 um 2 Uhr wurden innerhalb von 24 Stunden insgesamt 326.388 Transaktionen akzeptiert [3]. Dies verdeutlicht die aktive Nutzung von Bitcoin als alternative Währung für das etablierten Fiatgeld.

Doch auch bei Bitcoin kommt es immer wieder zu Meldungen, bei denen große Geldbeträge durch Kriminelle entwendet werden. Ein aktuelles Beispiel ist ein Hackerangriff auf die Bitcoin-Tauschbörse Bitfinex, der im August 2016 bekannt wurde. Insgesamt wurden 119.756 Bitcoins entwendet, was zu dem Zeitpunkt einem Gegenwert von ca. 58 Millionen Euro entsprach. Hierbei wurde allerdings keine Sicherheitslücke der Währung ausgenutzt, sondern ein Dienst angegriffen, mit dem Bitcoins gehandelt werden können [4]. Allerdings gab es neben Angriffen auf Systeme, die Bitcoins nutzten auch Angriffe auf das eigentliche Bitcoin-System. So wurde im August 2010 eine Transaktion durchgeführt, die zu einer Gutschrift von 184 Milliarden Bitcoins führte. Allerdings ist die Gesamtanzahl von Bitcoins auf 21 Millionen Stück beschränkt. Die Sicherheitslücke wurde geschlossen und die Transaktion konnte durch das Bitcoin-System für ungültig erklärt werden [5]. Wie die Angriffe möglich waren und wodurch die Transaktion für ungültig erklärt werden konnte, wird im Detail in dieser Arbeit erklärt.

Diese und weitere Meldungen zeigen, dass die Sicherheit einer digitalen Währung essentiell ist. Sicherheitslücken und fehlerhafte Konzepte können zu starken Kursschwankungen führen, wodurch alle Nutzer gefährdet wären. Eine Nutzung für den Handel oder als Kapitalanlage würde erschwert und die Akzeptanz würde gemindert.

Ein Ziel dieser Arbeit ist die Funktionsweise digitaler Zahlungsmittel zu beschreiben. Dazu werden sowohl Bitcoin untersucht als auch alternative Verfahren vorgestellt. Da insbesondere die Sicherheit einen großen Stellenwert bei einem digitalen Zahlungsmittel einnimmt, werden die Sicherheitsmechanismen und mögliche Angriffe ausführlich diskutiert. Die genutzten Mechanismen für Kryptowährungen beschränken sich nicht auf diesen Anwendungsfall. Ein Beispiel dafür ist die Blockchain, deren Funktionsweise später näher beschrieben wird.

Zunächst werden anhand von Bitcoin die Konzepte und technischen Grundlagen dezentraler digitaler Zahlungsmittel erklärt. Des Weiteren werden mögliche Angriffe auf Bitcoin und jeweils passende Gegenmaßnahmen beschrieben. Anschließend werden Weiterentwicklungen von Bitcoin und Alternativen dazu aufgezeigt. Den Abschluss bildet neben einem Ausblick die Zusammenfassung der Ergebnisse.

II. Technische Grundlagen

Anders, als traditionelle Bezahlsysteme, setzten digitale Zahlungssysteme wie Bitcoin auf eine dezentrale Verwaltung und Kontrolle. Aufgaben, die bei einer Fiatwährung eine zentrale Bank übernehmen würde, werden hier auf die Nutzer des Systems aufgeteilt. Zugrunde liegen dabei etablierte und weit verbreitete kryptografische Verfahren. In diesem Abschnitt sollen die technischen Grundlagen von digitalen Zahlungssystemen am Beispiel von Bitcoin dargestellt werden.

A. Peer-to-Peer

Um Dezentralität zu erreichen setzt Bitcoin auf ein *Peer-to-Peer*-Netzwerk, bei dem jeder Bitcoin-*Client* mehrere Verbindungen zu beliebigen anderen Bitcoin-Knoten aufbaut. In der Regel werden mit Hilfe des *Domain Name Systems* oder durch vorkonfigurierte IP-Adressen, Verbindungen zu anderen Knoten aufgebaut. Die verbundenen Knoten tauschen nun weitere, ihnen bekannte, IP-Adressen aus [6].

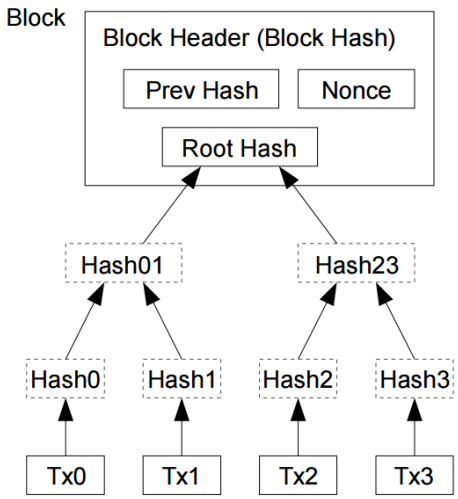

Abbildung 1. Ein Block mit Transaktionen, die mit einem Merkle-Baum gehasht wurden [1]

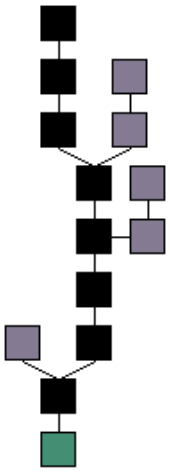

Abbildung 2. Blockchain mit Forks[8]

B. Blockchain

Das größte Problem eines digitalen Zahlungssystems ist die Doppelausgabe des Zahlungsmittels. Im Falle eines konventionellen Zahlungssystems würde eine Bank kontrollieren, dass das Zahlungsmittel nicht mehrfach ausgegeben wird. Um dies auch bei digitalen Zahlungsmitteln, wie Bitcoin zu gewährleisten, wird das oben beschrieben Peer-to-Peer-Netzwerk genutzt. Jeder Teilnehmer erhält eine Kopie aller bisher durchgeführten Transaktionen und kann somit Doppelausgaben erkennen und ablehnen. Das Verzeichnis aller Transaktionen nennt sich Blockchain. Die Blockchain besteht aus einer Liste von Blöcken, die jeweils ihren Vorgänger referenzieren. Dadurch können alle Blöcke bis hin zum ersten Block, dem sogenannten *Genesis-Block*, nachvollzogen werden [7].

Jeder Block beinhaltet eine oder mehr Transaktionen. Aus den Transaktionen wird mit Hilfe eines Merkle-Baums ein *Hash* erzeugt. Der resultierende Hashwert wird im Header des Blocks vermerkt. Abbildung 1 zeigt einen solchen Block mit den zugehörigen Transaktionen, dem Merkle-Baum, der aus diesen Transaktionen erzeugt wurde und den resultierenden *Root Hash*. Durch durchprobieren einer 32-Bit *Nonce* wird nun versucht ein SHA-256 Hash zu finden, dessen Wert kleiner oder gleich einem vorgegebenem Schwellenwert ist [6]. Der Schwellenwert besagt wie vielen führende Nullen ein passender Hashwert mindestens besitzen muss. Der hier beschriebene *Proof-of-Work*-Prozess wird bei Bitcoin *Mining* genannt und soll bewirken, dass ein neuer Block im Durchschnitt alle zehn Minuten erzeugt wird. Die festgelegte Zeitspanne wird dadurch gewährleistet, dass der Schwellenwert und somit die Schwierigkeit des Minings alle 2016 Blöcken angepasst wird [7].

Durch das Mining werden Transaktionen in der Blockchain verankert und gelten dadurch als ausgeführt und valide. Es kann allerdings passieren, dass zwei oder mehr Miner gleichzeitig eine Lösung finden, beziehungsweise weitere Miner eine Lösung propagieren, bevor die andere Lösung sie erreicht hat. In diesem Fall entstehen sogenannte *Forks*. Das heißt, dass mehrere Blöcke auf einen vorangegangen Block verweisen. Da die neuen Blöcke nun unterschiedliche Transaktionen enthalten können, kann es zu Diskrepanzen zwischen mehreren Benutzern kommen. Erst wenn einer der Forks länger als alle anderen ist, wird dieser allgemeingültig. Es wird angenommen, dass für diesen Fork die meiste Rechenleistung eingesetzt wurde [7]. Abbildung 2 zeigt eine Blockchain, bei der die Kette aus schwarzen Blöcken die längste Folge von Blöcken darstellt. Die lila eingezeichneten Forks sind kürzer als die schwarze Kette und gelten daher als verwaist.

Als Belohnung für die eingesetzte Rechenleistung erhält der Miner, der als erstes eine Lösung veröffentlicht hat, sowohl die Transaktionsgebühren der im Block enthaltenen Transaktionen als auch eine festgelegte Anzahl von zusätzlichen Bitcoins. Die Anzahl der Bitcoins, die ein Block enthält, halbiert sich alle 210.000 Blöcke. Während der Genesis-Block noch 50 Bitcoins enthielt, wurden im November 2012 zum ersten mal nur 25 Bitcoins ausbezahlt. Eine solche Belohnung wird es geben, bis diese unter den Wert von 10^{-8} Bitcoins fällt. Dies entspricht einem Satoshi, was die kleinste Einheit im Bitcoin-System darstellt. Danach werden Miner nur noch durch Transaktionsgebühren entlohnt [7].

C. Transaktionen

Als Bitcoin-Nutzer benötigt man mindestens ein Paar aus privatem und öffentlichen Schlüssel, wobei sich von dem öffentlichen Schlüssel anschließend eine Adresse ableiten lässt. Mit Hilfe der Adresse kann man Transaktionen erhalten. Der private Schlüssel muss in jedem Fall geheim gehalten und darf nicht verloren werden, da durch ihn der Besitz von Bitcoins nachgewiesen wird und man ihn für das Signieren von ausgehenden Transaktionen benötigt [9].

Die wichtigsten Komponenten einer Transaktion sind die Transaktions-ID (*TXID*) und je eine Liste für die Ein- und Ausgaben (*Inputs* und *Outputs*). Die TXID wird durch einen SHA-256 Hash der serialisierten Transaktion gebildet [10]. Als Inputs können ausschließlich bisher noch nicht ausgegebene Outputs von anderen Transaktionen genutzt werden. Ist die

Summe der Inputs größer, als die Summe der Outputs, handelt es sich bei dem verbleibenden Rest um die oben erwähnte Transaktionsgebühr, die an den Miner gezahlt wird, wenn er diese Transaktion in seinem Block eingefügt hat. Des Weiteren kann als ein Output auch eine eigene Adresse angegeben werden, damit die Inputs nicht komplett an andere Nutzer weitergegeben werden müssen. Insgesamt können so die einzelnen Transaktionen bis zu ihrem Ursprung zurückverfolgt werden. Dieser ist eine der Transaktionen, die Bitcoins erzeugen. Dabei handelt es sich entweder um den Genesis-Block oder eine sogenannte *Coinbase*-Transaktion. Letztere übertragen die oben beschriebene Belohnung für den Proof-of-Work an den Miner. Dies zeigt, dass bei Bitcoins, nicht wieder Name vermuten lässt, keine Münzen im traditionellen Sinne existieren. Die Transaktionen und ihre Transaktions-IDs übernehmen die Aufgaben von Münzen, Geldscheinen und Überweisungen [7].

D. Scripts

Jeder Output einer Transaktion gibt neben dem Betrag an, wem dieser Betrag angerechnet werden soll. Der Besitzanspruch wir bei Bitcoin durch *Scripts* geregelt. Bei einem Output wird im Feld `scriptPubKey` ein Script hinterlegt, mit dem der Empfänger beweisen muss, dass die angegebenen Bitcoins ihm gehören. Möchte der neue Besitzer nun die Bitcoins überweisen, referenziert er den ungenutzten Output als Input innerhalb einer Transaktion. Um seinen Besitzanspruch geltend zu machen, muss er im `scriptSig` Feld des Inputs Argumente angeben, mit denen das Script aus dem Output erfolgreich ausgeführt werden kann [10].

Das Script *"Pay-to-PubKeyHash"* (P2PKH) bildet die Basis für einfache Bitcoin-Transaktionen. Im `scriptPubKey` wird die Bitcoin-Adresse des Empfängers (`pubKeyHash`) in das Script eingefügt. Der Empfänger gibt anschließend im `scriptSig` seine Signatur (`sig`) und den öffentlichen Schlüssel (`pubKey`) an. Listing 1 zeigt eine Vorlage für ein P2PKH-Script. Bei der Ausführung des Scripts wird zunächst verglichen, ob der korrekte öffentlich Schlüssel angegeben wurde. Dies geschieht, indem aus dem übergebenen öffentlichen Schlüssel die Bitcoin-Adresse errechnet und anschließend mit der Adresse aus dem Output verglichen wird. Nachfolgend wird überprüft, ob die Signatur zu dem öffentlichen Schlüssel passt. Wenn das Script erfolgreich ausgeführt wurde, kann der Empfänger über die Bitcoins verfügen [7]. Ein weiteres Beispiel für Bitcoin-Scripte sind die *M-of-N multi-signature* Transaktionen. In diesem Fall kann im `scriptPubKey` festgelegt werden wie viele Signaturen benötigt werden, damit die angegebenen Bitcoins verwendet werden können. Zusätzlich werden die öffentlichen Schlüssel angegeben, für welche die Signaturen bereitgestellt werden müssen [11]. Somit kann beispielsweise eine 2-of-3 multi-signature Transaktion erst mit mindestens zwei gültigen Signaturen durchgeführt werden [7].

Bisher wurde das Script durch den Initiator festgelegt, wobei mindestens ein öffentlicher Schlüssel, beziehungsweise die daraus berechneten Adresse in das Script eingefügt wurden. Um jedoch komplexere Szenarien zu ermöglichen, bei denen

```
scriptPubKey: OP_DUP OP_HASH160
        <pubKeyHash> OP_EQUALVERIFY
        OP_CHECKSIG

scriptSig: <sig> <pubKey>
```

Listing 1. "Pay-to-PubKeyHash" (P2PKH) Script Vorlage [7]

```
scriptPubKey: OP_HASH160
        <redeemScriptHash> OP_EQUAL

scriptSig: [<sig> ...] <redeemScript>
```

Listing 2. "Pay-to-ScriptHash" (P2SH) Script Vorlage [7]

der Empfänger ein Script festlegen kann, mit dem auf die Bitcoins zugegriffen wird, wurden Transaktionen mit *"Pay-to-ScriptHash"* (P2SH) eingeführt. Aus dem auch *redeemScript* genannten Script des Empfängers wird ein Hash erstellt, welcher anstelle der Adresse im `scriptPubKey` eingetragen wird. In das `scriptSig` Feld muss nun neben den benötigten Signaturen das redeemScript eingetragen werden. Eine Vorlage für ein P2SH ist in Listing 2 zu sehen. Für die Überprüfung wird der bereitgestellte Hash mit dem tatsächlichen Hash des redeemScripts verglichen, bevor das Script ausgeführt wird. Eine solche Transaktion wird als Standard-Transaktion angesehen, wenn das redeemScript einem Standard-Transaktionstyp, wie zum Beispiel dem P2PKH-Script entspricht [12].

III. Angriffsvektoren und Gegenmassnahmen

Da es sich bei Bitcoin um eine Kryptowährung handelt, können Angriffe zur Folge haben, dass Bitcoin entwendet werden und dem Opfer dadurch ein finanzieller Schaden entsteht. Des Weiteren können sich Angriffe bei Bitcoin auch gegen die Privatsphäre der Nutzer richten. Dies hätte zur Folge, dass Zahlungsströme offengelegt und einzelnen Personen zugeordnet werden können. In diesem Kapitel werden mögliche Angriffe beschrieben, jeweils gefolgt von Gegenmaßnahmen, die Nutzer von Bitcoin ergreifen sollten.

A. Wallets

Die Schlüssel, die für die Nutzung von Bitcoin benötigt werden, werden mit Hilfe des Bitcoin-Clients gespeichert. Ein solcher Client wird auch *Wallet* genannt und bedarf einer gesonderten Absicherung. Der Verlust oder Diebstahl eines Wallets, beziehungsweise der in dem Wallet befindlichen Daten, bedeutet auch der Verlust der zugehörigen Bitcoins [9]. Angreifer nutzen *Malware*, um den Computer des Opfers zu infizieren und gelangen so an den oder die privaten Schlüssel, welche auf dem Computer gespeichert sind. Ein Beispiel hierfür ist die Malware Infostealer.Coinbit, welche im Juni 2011 entdeckt wurde. Sie sucht auf einem infizierten Windows Computer nach einem Bitcoin-Wallet, welches anschließend per E-Mail versandt wird [13]. Um die in einem Wallet befindlichen Bitcoins zu schützen gibt es unterschiedliche Möglichkeiten. Eine Möglichkeit ist die Art des Wallets.

Software-Wallets jeglicher Art sind anfällig für Hackerangriffe. Dazu zählen sowohl Programme, die lokal auf dem Rechner des Nutzers installiert sind als auch sogenannte Online-Wallets, bei denen die Operationen über die Webseite des Betreibers durchgeführt werden. Aus diesem Grund wurden Hardware-Wallets entwickelt, bei denen ein separates Gerät genutzt wird, welches nicht mit dem Internet verbunden ist. Weiter gehen Papier- oder Gedächtnis-Wallets. Hierbei werden die benötigten Schlüssel entweder ausgedruckt oder der Nutzer muss sich eine Passphrase merken, die dann in einen private Schlüssel transformiert werden kann [7].

Ein weiterer Ansatz besteht darin eine Art Zwei-Faktor-Authentifizierung zu implementieren. Wie oben beschrieben bietet Bitcoin die Möglichkeit M-of-N multi-signature Transaktionen durchzuführen. Neben dieser Möglichkeit bieten sich Schwellenwert-Signaturen an, bei denen der private Schlüssel aufgeteilt wird. Erst mit einer Anzahl von Teilen des privaten *Keys*, die größer oder gleich dem Schwellenwert ist, kann der gesamte private Schlüssel errechnet und so die für die Transaktion benötigte Signatur erstellt werden. Im Prinzip ist es durch die vorgestellten Verfahren aber auch möglich mehr als zwei Faktoren beziehungsweise mehrere Nutzer in den Prozess mit einzubeziehen [7]. Um durch dieses Vorgehen einen Nutzen zu erzielen müssen die privaten Schlüssel und Teilschlüssel in unterschiedlichen Wallets gespeichert werden.

B. Kryptographie

Bitcoin nutzt den ECDSA, um sicherzustellen, dass nur der rechtmäßige Besitzer über seine Bitcoins verfügen kann. ECDSA steht für *Elliptic Curve Digital Signature Algorithm*. Dabei handelt es sich um einen Signaturalgorithmus, der auf elliptischen Kurven basiert. Der private Schlüssel wird zusammen mit einem Zufallswert benutzt, um eine Signatur für eine Transaktion zu erzeugen. Der Zufallswert muss für jede Signatur zufällig und nicht vorhersagbar sein. Ist dies nicht der Fall, kann basierend auf mehreren Signaturen der private Schlüssel errechnet werden. Anfällig für diese Sicherheitslücke sind fehlerhafte Implementierungen oder Konfigurationen von Zufallszahlengeneratoren innerhalb der Wallets. Der Bitcoin-Client *bitcoin.info* zum Beispiel nutzte einen falsch konfigurierten Zufallszahlengenerator. Es wird vermutet, dass dadurch mehr als 59 Bitcoins entwendet werden konnten [7]. Insgesamt lässt sich sagen, dass eine fehlerhafte Implementierung der kryptographischen Algorithmen immer zu Sicherheitslücken führen kann.

Sicherheitsrisiken, die durch falsche Implementierung oder Konfigurationen entstehen sind für Anwender nur schwer zu erkennen. *Open Source* Software kann ein Ansatz sein, um solche Risiken zu minimieren. Solange ein Fehler allerdings auch dort nicht entdeckt wird, sind Nutzer weiterhin angreifbar.

C. Double Spending

Einen weiteren Angriffsvektor stellt das sogenannte *Double Spending* dar. Ziel ist es einen Zahlungsempfänger zu täuschen, indem zwei konkurrierende Transaktionen veröffentlicht werden, die auf dieselbe noch nicht ausgegebene Transaktion verweisen. Wie in Abschnitt II-C bereits beschrieben ist dies allerdings nicht zulässig. Nur eine der beiden Transaktionen darf tatsächlich ausgeführt werden. Im Detail läuft der Angriff wie folgt ab: Zunächst wird mit einer regulären Transaktion ein Betrag an einen anderen Nutzer überwiesen. Beispielsweise kann der Angreifer vorgeben ein Produkt von seinem Opfer kaufen zu wollen. Zusätzlich muss der Angreifer eine weitere Transaktion erstellen, die dieselben Eingaben hat, wie die bisher reguläre Transaktion. In diesem Fall ist der Empfänger allerdings der Angreifer selber. Nun muss der Angreifer warten, bis das Opfer die Gegenleistung für die Transaktion erbracht hat und anschließend die konkurrierende Transaktion veröffentlichen. Eine weiter Möglichkeit besteht darin die konkurrierende Transaktion direkt in einem anderen Bereich des Netzwerks zu veröffentlichen, damit das Opfer diese erst zu spät erhält. Schafft diese Transaktion es nun in den längsten Zweig der Blockchain, war der Angriff erfolgreich, da der Angreifer eine Gegenleistung von dem Opfer erhalten hat, ohne dass die Bitcoins den Besitzer gewechselt haben [7].

Zusätzlich kann der Angreifer seine Chancen auf einen erfolgreichen Angriff noch erhöhen. Eine Möglichkeit besteht darin, dass er sich am Mining beteiligt und basierend auf dem letzten Block einen eigenen Fork erstellt, der die konkurrierende Transaktion enthält. Sobald sein Fork länger ist und er die gewünschte Leistung von seinem Opfer erhalten hat, veröffentlicht er die von ihm berechneten Blöcke. Der längere Zweig wird nun von allen anderen Nutzern als Basis für die Validierung und das Mining genutzt, wodurch der Angriff erfolgreich war [7].

Eine Gegenmaßnahme, die ein potenzielles Opfer ergreifen sollte, besteht darin auf mehrere Bestätigungen durch die Blockchain zu warten. Zunächst muss die relevante Transaktion in einem Block enthalten sein, der in die Blockchain gelangt. Je mehr Blöcke nun an diese Kette angefügt werden, desto sicherer ist die Transaktion. Allerdings kann die Wahrscheinlichkeit für einen erfolgreichen Angriff nie auf Null reduziert werden, egal wie viele Bestätigungen abgewartet werden. Allerdings muss ein Angreifer dementsprechend immer mehr Rechenleistung investieren, um den Angriff erfolgreich durchzuführen [7]. Je nachdem wie hoch der Gegenwert ist, den er von seinem Opfer erhalten kann, lohnt sich der Angriff aus finanzieller Sicht nicht mehr. Auch wenn Double Spending mit jeglicher Hashrate möglich ist, erhöht sich die Wahrscheinlichkeit auf einen erfolgreichen Angriff mit steigender Rechenleistung. Kann der Angreifer mehr als 50% der gesamten Rechenleistung im Bitcoin-Netzwerk bereitstellen, ist sein Angriff in jedem Fall erfolgreich [14]. Abbildung 3 zeigt die Wahrscheinlichkeit eines erfolgreichen Double Spending Angriffs basierend auf der Hashrate der Angreifers q und der Anzahl der Bestätigungen n.

D. Transaction Malleability

Transaction Malleability beschreibt die Möglichkeit die ID einer Transaktion zu verändern, ohne die Transaktion insgesamt ungültig zu machen. Wie im Abschnitt II-C beschrieben, wird die TXID mit einem SHA-256 Hash der serialisierten Transaktion erstellt. Ändert sich an der Transaktion also etwas,

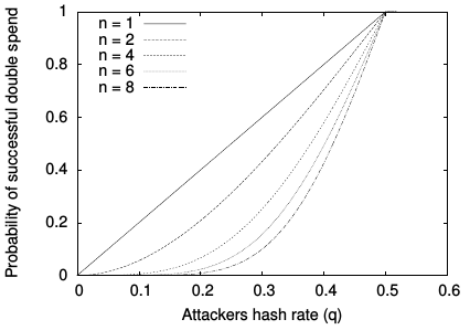

Abbildung 3. Wahrscheinlichkeit eines erfolgreichen Double Spending Angriffs [7]

verändert sich auch der Hashwert und somit die TXID. Die Signaturen im `scriptSig` Feld sollen jedoch unbemerkte Veränderungen der Transaktion verhindern. Allerdings besteht das Problem, dass die Signaturen nicht über das `scriptSig` Feld gebildet werden können, da dies zu einer zirkulären Abhängigkeit führen würde. Aus diesem Grund werden vor der Berechnung einer Signatur alle redeemScripte durch Dummy-Operationen ersetzt. Folgerichtig kann ein Angreifer ein redeemScript durch zusätzliche Operationen dahingehend verändern, dass sich die TXID verändert, die Signatur aber gültig bleibt. Weitere Variationen des Angriffs sind möglich, indem zum Beispiel die Signatur abweichend encodiert wird [10].

Alleinig die Änderung der TXID führt noch nicht dazu, dass ein Schaden entsteht [7]. Zusätzlich muss das Opfer zur Nachverfolgung von Transaktionen ausschließlich auf die TXID setzen. Transaction Malleability ähnelt dem Double Spending, in diesem Fall ist allerdings der Angreifer der Empfänger der Bitcoins. Das potenzielle Opfer muss eine Transaktion veröffentlichen, die an den Angreifer gerichtet ist. Dieser verändert nun wie oben beschrieben die Transaktion und veröffentlicht diese ebenfalls im Bitcoin-Netzwerk. Der Angriff ist erfolgreich, wenn die veränderte Transaktion in die Blockchain aufgenommen wird. In diesem Fall sieht das Opfer nur, dass die von ihm verfolgte Transaktion nicht bestätigt wurde und führt eine weitere Transaktion aus oder erstattet dem Angreifer anderweitig die aus seiner Sicht nicht übertragenen Bitcoins [10].

Um Transaction Malleability Angriffe zu verhindern gibt es unterschiedliche Möglichkeiten. Ein komplexer Ansatz besteht darin die Veränderung der Transaktionen durch zusätzliche Regeln unmöglich zu machen. Jedoch bestehen weiterhin Szenarien, in denen es möglich ist Transaction Malleability auszunutzen [15]. Die Referenzimplementierung des Bitcoin-Clients verfolgt allerdings einen anderen Ansatz und ist nicht anfällig für diesen Angriff. Zusätzlich zur TXID werden die noch nicht ausgegebenen Outputs nachverfolgt [10].

E. Mining Pools

Wenn ein Miner erfolgreich einen Block berechnet, erhält dieser alleinig die Belohnung dafür. Allerdings geschieht dies für einzelne Akteure nur in unregelmäßigen Abständen. Aus diesem Grund haben sich *Mining Pools* entwickelt, bei denen sich mehrere Miner zusammenschließen, um gemeinsam ihre Ressourcen in den Miningprozess zu investieren. Findet nun einer der beteiligten Miner mit einer Nonce den passenden Hash, wird die Belohnung unter allen Minern im Pool aufgeteilt. Durch dieses Konzept erhält ein beteiligter Miner regelmäßiger eine Belohnung, als wenn er alleine arbeiten würde. Allerdings ist die Belohnung bei Mining im Pool pro Miner kleiner, da sie unter mehreren Beteiligten aufgeteilt wird [16]. Mining Pools stellen insofern eine Gefahr da, als das Sie die Rechenleistung von mehreren Minern kombinieren und so einfacher über 50% der Rechenleistung erreichen können. *GHash.io* beispielsweise erreichte bereits diese kritische Grenze. Bisher konnten die Miner selber durch einen Wechsel des Mining Pools das Gleichgewicht wieder herstellen. Ein weiterer Lösungsansatz sind dezentralisierte Mining Pools, bei denen die kombinierte Rechenleistung nicht ausgenutzt werden kann, da eine zentrale Koordination aller Beteiligten fehlt. Die dadurch gestiegene Komplexität verringert allerdings die Attraktivität von dezentralen Mining Pools [7].

Insgesamt stehen Miner und dadurch auch Mining Pools zueinander in Konkurrenz, da jeder Teilnehmer versucht zuerst einen validen Block zu finden. Neben der reinen zur Verfügung stehenden Rechenleistung können unterschiedliche Angriffe Vorteile gegenüber anderen Mining Pools schaffen. So hat sich gezeigt, dass die zweithäufigsten Ziele für *Denial-of-Service*-Angriffe innerhalb des Bitcoin-Systems, Mining Pools sind. Ist ein Mining Pool nicht verfügbar, erhöhen sich dadurch für die verbleibenden Mining Pools die Gewinnchancen [17]. Ein weiterer Angriff im Umfeld von Mining Pools nennt sich *Selfish Mining*. Damit wird beschrieben, dass ein Mining Pool egoistisch handelt indem er neue valide Blöcke zunächst für sich behält, daraus eine eigene private Blockchain bildet und nur an dieser weiterarbeitet. Sobald sich die Länge der öffentlichen Blockchain an die Länger der privaten Blockchain annähert, werden ausgewählte, geheim gehaltene Blöcke veröffentlicht. Dadurch wird erreicht, dass die anderen Miner ihre Rechenleistung verschwenden, da sie auf Basis von Blöcken arbeiten, die später zu einem verwaisten Zweig der Blockchain gehören werden [18].

Im Detail verläuft der Angriff wie folgt. Wie bereits beschrieben, wird ein valider Block nicht direkt veröffentlicht, sondern der Mining Pool etabliert einen privaten Zweig der Blockchain. Nun gibt es mehrere Fälle, die eintreten können. Im ersten Fall wird im Anschluss von den anderen Minern ein Block basierend auf der öffentlichen Blockchain gefunden. Der angreifende Mining Pool veröffentlicht sofort seinen Block und hofft darauf, dass sein Zweig überlebt. Im zweiten Fall finden die egoistischen Miner einen weiteren validen Block, wodurch die private Blockchain zwei Blöcke Vorsprung hat. Solange der Vorsprung größer als einen Block ist, wird für jeden Block, der basierend auf der öffentlichen Blockchain gefunden wird, der nächste private Block veröffentlicht. Ist nur noch ein Abstand von einem Block vorhanden, wird die ganze private Blockchain veröffentlicht, um in jedem Fall von dem Gewinn zu profitieren. Da der vorher private Zweig länger ist, als alle bisher öffentlichen Zweige, werden alle ehrlichen Miner den Zweig des angreifenden Mining Pools übernehmen und auf diesem arbeiten [18].

Maßnahmen gegen Selfish Mining sind nur schwer umzusetzen und können den Angriff nicht vollkommen verhindern. Denkbar sind Änderungen im Bitcoin-Protokoll, wodurch ehrliche Miner durch Zufall auswählen auf welchem Block sie ihr Berechnungen durchführen, wenn konkurrierende Blöcke existieren [18]. Eine weitere Möglichkeit wäre immer nur mit dem neuesten Block fortzufahren. Da ein Zeitstempel im Bitcoin-Netzwerk aber unter anderem von benachbarten Knoten bezogen und dadurch beeinflusst werden kann, ist auch diese Möglichkeit nur begrenzt einsatzfähig [7].

F. Deanonymisierung

Da bei Bitcoin alle Transaktionen veröffentlicht werden und von der Öffentlichkeit einsehbar sind, besteht keine vollkommene Anonymität. Privatsphäre kann nur gewährleistet werden, solange die öffentlichen Schlüssel nicht mit dem jeweiligen Nutzer in Verbindung gebracht werden können. Zusätzlich sollte für jede Transaktion ein neues Schlüsselpaar genutzt werden [1]. Das Konzept von Bitcoin sollte eher als Pseudonymität verstanden werden. Bei normaler Nutzung können unterschiedliche Pseudonyme miteinander verbunden werden, wodurch alle Aktivitäten nachverfolgt werden können. Durch bestimmte Heuristiken können aus der Blockchain Transaktionsgraphen, daraus Adressgraphen und anschließend Entitätsgraphen erstellt werden. Beispielsweise wird angenommen, dass alle Input-Adressen einer Transaktion zu einer Entität gehören. Des Weiteren wird nach der Heuristik eine neue Adresse, die als Output verwendet wird, auch der jeweilige Entität zugeordnet, da diese vermutlich für das Wechselgeld genutzt wird. Kann nun ein Pseudonym (also eine Adresse) deanonymisiert werden, können durch die erstellten Graphen alle dieser Entität zugeordneten Transaktionen aufgedeckt werden [7].

Für die Deanonymisierung wird allerdings ein Seitenkanal benötigt. Teilweise reichen öffentliche Informationen schon aus, um eine Adresse einer Person zuzuordnen. Sollte ein Nutzer seine Adresse zum Beispiel in einem sozialen Netzwerk veröffentlicht haben, kann diese Information genutzt werden [7]. Eine weitere Möglichkeit bieten Dienste, bei denen mit Bitcoins bezahlt wird. Teilweise müssen dort personenbezogene Daten angegeben werden, um diese zu nutzen. Ebenfalls kann die IP-Adresse desjenigen genutzt werden, der eine Transaktion im Bitcoin-Netzwerk veröffentlicht [19].

Der Erste, der eine Transaktion innerhalb des Bitcoin-Netzwerk veröffentlicht, ist in der Regel auch der Ersteller der Transaktion. Um die IP-Adresse dieses Bitcoin-Clients zu erhalten muss zunächst eine Verbindung zu so vielen Bitcoin-Knoten aufgebaut werden, wie möglich. Im Idealfall ist ein Angreifer nun mit allen Knoten im Netzwerk verbunden. Anhand der eingehenden Transaktionen können nun Rückschlüsse auf den Initiator einer Transaktion und damit den Besitzer der privaten Schlüssel gezogen werden. Bedingt durch die Verhaltensweisen der einzelnen Bitcoin-Clients, deren Bandbreite und den damit verbundenen Verzögerungen kann es allerdings vorkommen, dass nicht alle Transaktionen eindeutig einer IP-Adresse zugeordnet werden können. Des Weiteren führt die Nutzung von Diensten, wie Online-Wallets dazu, dass gegebenenfalls nur die IP-Adresse des Dienstes ermittelt werden kann. Auch durch Anonymisierungsdienste, wie das Tor Netzwerk, lassen sich keine Rückschlüsse auf die IP-Adresse des Initiators ziehen [20].

Solange alle Bitcoin-Transaktionen mit Hilfe eines Anonymisierungsdienstes getätigt werden, ist die Privatsphäre eines Nutzers weitgehend geschützt [21]. Ein Angreifer könnte allerdings verhindern, dass ein solcher Dienst genutzt wird. Mit Hilfe des *Denial-of-Service*-Schutzes von Bitcoin können Clients, die sich fehlerhaft verhalten 24 Stunden ausgeschlossen werden. Dies geschieht nur, wenn ein Client mehrere fehlerhafte Nachrichten versendet. Am Beispiel von Tor lässt sich das Verfahren verdeutlichen. Jede Verbindung verlässt das Tor Netzwerk über einen sogenannten *Exit*-Knoten. Die nach außen hin sichtbare IP-Adresse entspricht dann der des Exit-Knoten. Ein Angreifer muss nun über alle Exit-Knoten oder mittels *IP-Spoofing* genug fehlerhafte Nachrichten verschicken, damit alle Exit-Knoten aus dem Bitcoin-Netzwerk ausgeschlossen werden. Dieses Vorgehen lässt sich auch auf andere Anonymisierungsdienste anwenden [22].

[21] beschreibt eine Variante des oben beschriebenen Angriffs. Dadurch, dass ein Angreifer eigene Exit-Knoten im Tor Netzwerk betreibt und alle anderen Exit-Knoten vom Bitcoin-Netzwerk ausschließen lässt, kann er Nutzer dazu bringen seine Exit-Knoten zu Nutzen. Nun kann der Angreifer steuern welche Transaktionen und Blöcke der Nutzer erhält. Weiter wäre es möglich durch zusätzliche Knoten im Tor Netzwerk Nutzer zu deanonymisieren. Einfacher ist allerdings ein *Fingerprinting*, bei dem alle über Tor getätigten Transaktionen einem Initiator zugeordnet werden können. Nutzt dieser mit demselben Client anschließend Bitcoin ohne Tor, können die gesammelten Informationen in Verbindung gebracht werden. Möglich ist das Fingerprinting zum Beispiel mit gefälschten Adressen, die dem Client des Opfers mitgeteilt werden, damit dieser die Adressen speichert. Erhält der Angreifer nun von irgendeinem Client bei einer Adressanfrage die gefälschten Adressen und damit den Fingerabdruck zurück, kann er das Opfer erneut zuordnen [21].

Auch die Einstiegsknoten, mit denen sich ein Bitcoin-Knoten beim Einstieg in das Netzwerk verbindet, können als Fingerabdruck dienen. In der Regel werden diese erst gewechselt, wenn sie nicht mehr verfügbar sind. Ein Angreifer muss sich erneut mit möglichst vielen Knoten, im Idealfall allen, mehrfach verbinden. Tritt ein potenzielles Opfer in das Netzwerk ein, wird dem Angreifer zuerst über die Einstiegsknoten die IP-Adresse des Opfers mitgeteilt. Diese Einstiegsknoten merkt sich der Angreifer. Werden nun Transaktionen von diesen Knoten weitergeleitet, handelt es sich mit einer hohen Wahrscheinlichkeit um eine Transaktion des Opfers. Somit können die Transaktionen einer IP-Adresse zugeordnet werden [22].

Für die beschriebenen Methoden der Deanonymisierung gilt allerdings, dass sie nicht in jedem Fall erfolgreich sind. Das Bitcoin-Protokoll erschwert durch einige seiner Vorkehrungen zwar die Analyse, macht sie jedoch nicht unmöglich. Beispielsweise geben nicht alle Einstiegsknoten die IP-Adresse eines Clients an alle anderen Nutzer weiter. Nichtsdestotrotz weisen alle beschriebenen Angriffe auf die Privatsphäre der

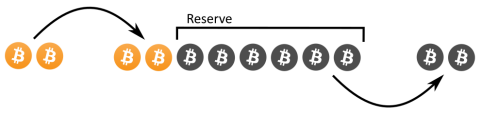

Abbildung 4. Beispielhafte Darstellung eines Bitcoin Mixing Services. [7]

Nutzer eine nicht zu verachtende Erfolgsquote auf [22], [7].

Ein einfacher Weg die Analyse von Transaktionen in der Blockchain zu erschweren, ist die Nutzung eines Dienstes, der eine Art Geldwäsche anbietet. Diese Dienste werden *Mixing Services* genannt und sollen die Nachverfolgung von Bitcoins erschweren. Der Dienst erhält vom Nutzer die zu verschleiernden Bitcoins und sendet gegen eine Gebühr andere Bitcoins an die vom Nutzer angegebene Adresse. Dadurch, dass viele Transaktionen durch denselben Dienst laufen und der Nutzer andere Bitcoins erhält, soll die Privatsphäre gewährleistet werden. Allerdings funktioniert ein solcher Dienst nur zuverlässig, wenn eine hohe Anzahl von anderen Transaktionen gegeben ist, um Ein- und Ausgänge des Dienstes nicht korrelieren zu können [7]. Abbildung 4 veranschaulicht einen solchen Mixing Service.

Um die Anonymität zu wahren sollten Bitcoin-Adressen mit Vorsicht behandelt werden. Die Veröffentlichung dieser oder die Verwendung von Bitcoin mit Diensten, bei denen personenbezogene Daten angegeben werden, können zur Deanonymisierung führen. Auch bei der Verwendung von Anonymisierungsdiensten müssen Nutzer Vorsicht walten lassen. Denial-of-Service-Attacken können hierbei nicht umgangen werden. Sind die Dienste allerdings verfügbar, verbessern sie die Anonymität. Im Fall von Tor werden zusätzlich einige Bitcoin-Knoten auch als *Hidden Services* innerhalb des Tor Netzwerks angeboten, mit denen sich ein Client verbinden kann. Zusätzlich zu einem anonymisieren Bitcoin-Client kann ein Nutzer einen weiteren Client betreiben, welcher direkt mit dem Bitcoin-Netzwerk verbunden ist. So kann sichergestellt werden, dass der anonymisierte Client über denselben Stand der Blockchain verfügt[21]. Gegen das Fingerprinting helfen zum einen die gespeicherten Adressen von anderen Knoten regelmäßig zu löschen [21] und zum anderen die Einstiegsknoten nach jeder Transaktion zu wechseln, damit die Transaktionen über unterschiedliche Knoten verteilt werden [22]. Des Weiteren ist es generell ratsam unterschiedliche Clients für Vorgänge zu nutzen, die anonymisiert und nicht anonymisiert getätigt werden.

IV. Weiterentwicklungen und alternative Ansätze

Der Erfolg von Bitcoin führte zu zahlreichen Neu- und Weiterentwicklungen von Kryptowährungen. So listet [2] zur Zeit der Ausarbeitung 643 Kryptowährungen (Stand 08.01.2017). Weiterentwicklungen, bei denen das Protokoll von Bitcoin abgewandelt wurde, werden als *Altcoins* bezeichnet [23]. Dabei lässt sich zwischen zwei Arten von Weiterentwicklungen unterscheiden. Zum einen werden mit den Altcoins Kryptowährungen entwickelt, die einige Einschränkungen von Bitcoin versuchen zu beseitigen. Zum Anderen werden auf Basis der Bitcoin-Technologie neue Bereiche erschlossen, die nicht nur den Zweck einer Kryptowährung erfüllen [24]. Insbesondere die Blockchain wird auch in Bereichen adaptiert, die keinen Bezug zu Kryptowährungen oder dem Finanzsektor haben [25]. Im Folgenden werden sowohl Altcoins, als auch andere Anwendungsbereiche näher erörtert, wobei nur Beispiele aufgeführt werden, denen eine Kryptowährung zugrunde liegt.

A. Litecoin

Ein Beispiel für eine abgewandelte Kryptowährung ist Litecoin. Es wurden lediglich einige Änderungen auf Basis des Bitcoin-Protokolls implementiert, um bestehende Einschränkungen zu umgehen. Bei Litecoin wird im Durchschnitt alle zweieinhalb Minuten ein neuer Block berechnet und es sind 84 Millionen Einheiten der Währung möglich. Im Vergleich wird bei Bitcoin alle zehn Minuten ein Block berechnet, während insgesamt 21 Millionen Einheiten möglich sind. Durch die höhere Rate von Blöcken werden Transaktionen schneller bestätigt. Da bei Bitcoin inzwischen spezialisierte Hardware für Miner unverzichtbar ist, wurde für Litecoin der Proof-of-Work-Algorithmus von SHA-256 auf *scrypt* umgestellt. Dies sollte sicherstellen, dass herkömmliche Hardware für das Mining ausreichend ist [26]. Dieses Ziel wurde allerdings verfehlt, da eine solche Hardware nicht mehr genügt [24], [7].

B. Namecoin

Namecoin ist eine weitere Abwandlung von Bitcoin. Es wurden allerdings nur wenige Änderungen am Protokoll vorgenommen. Dafür wurden zusätzlich Funktionen implementiert, um eine verteilte *Name/Value*-Datenbank zu ermöglichen. *Domain Names* werden in der Regel durch zentrale Autoritäten, wie der ICANN verwaltet. Um möglichen Missbrauch durch eine zentrale Stelle zu vermeiden, verfolgt Namecoin den Ansatz eines dezentralen Domain Name Systems (DNS). .bit-Domains können mit Hilfe von Transaktionen in der Blockchain von Namecoin registriert werden. Kompatible DNS-Programme können den Domain Name anschließend zu IP-Adressen auflösen. Neben dem DNS, wird Namecoin zum Beispiel von OneName zum Speichern von Informationen zu einem Benutzernamen genutzt [28].

C. Zerocoin

Es existieren auch Ansätzen, bei denen Weiterentwicklungen in das Bitcoin-Protokoll mit einfließen sollen. Ein Beispiel hierfür ist Zerocoin. Ziel von Zerocoin ist es die Anonymität innerhalb von Bitcoin zu verbessern. Es sollte möglich sein Bitcoins in eine Zerocoin-eigene anonyme Währung zu übertragen (*Zercoin Mint*). Durch einen *Zero-Knowledge*-Beweis ist es anschließend möglich die Geldmittel wieder in Bitcoins umzuwandeln (*Zerocoin Spend*) und mit einer anderen Bitcoin-Adresse zu verwalten. Eine Verbindung zwischen der einzahlenden und der entnehmenden Bitcoin-Adresse kann durch den Zero-Knowledge-Beweis nicht hergestellt werden [27]. Abbildung 5 verdeutlicht den Ablauf. Das

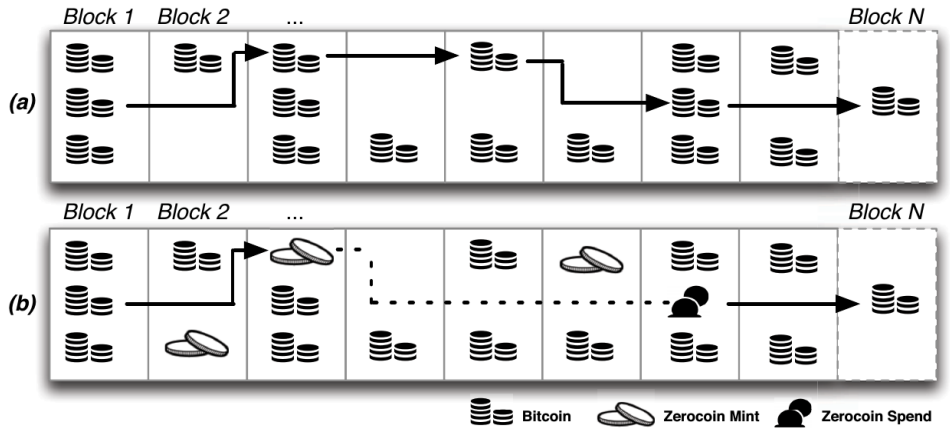

Abbildung 5. Zwei Beispiele für eine Blockchain. (a) zeigt eine normale Bitcoin Blockchain, bei der die Transaktionen miteinander verbunden sind. (b) zeigt eine mit Zerocoin angereicherte Blockchain. Die gestrichelte Linie, welche ein Beziehung zwischen den Zerocoin Mint und Zerocoin Spend darstellt, kann in der Blockchain nicht nachvollzogen werden. [27]

Vorgehen entspricht in etwa einem in Abschnitt III-F vorgestellten Mixing Service, wobei keine weitere Partei benötigt wird, um den Weg der Bitcoins zu verschleiern. Zerocoin wurde allerdings nicht in Bitcoin integriert. Anstatt dessen wurde eine verbesserte Version des Protokolls entwickelt [29] und als eigene Kryptowährung namens Zcash veröffentlicht [30].

D. Ethereum

Bei Ethereum handelt es sich um ein verteiltes System zur Entwicklung und Ausführung von dezentralen Anwendungen. In erster Linie ist Ethereum aber keine Kryptowährung, obwohl eine Blockchain zusammen mit der Kryptowährung Ether die Grundlage bildet. Nichtsdestotrotz liegt der Fokus auf den *Smart Contracts* genannten Anwendungen, die sich innerhalb der Blockchain speichern und ausführen lassen. Die Smart Contracts werden, ähnlich wie bei Bitcoin die Scripte, in einer *Bytecode*-Sprache implementiert. Der Funktionsumfang und die Möglichkeiten sind allerdings umfangreicher [31]. Smart Contracts werden in der *Ethereum Virtual Machine* (EVM) ausgeführt. Die EVM ist eine Virtuelle Maschine, die auf allen Ethereum-Knoten gleich aufgebaut ist. Der Programmcode wird von alle Knoten, die Transaktionen validieren, verteilt in ihrer EVM ausgeführt. Um Smart Contracts in der EVM ausführen zu können, müssen genug Geldmittel in Form der eigenen Währung Ether bereitgestellt werden. Der Betrag, der gezahlt werden muss, orientiert sich an der Komplexität des Programmcodes [32].

Mit Hilfe von Smart Contracts können komplexe Systeme, wie dezentrale autonome Organisationen (DAO) entwickelt werden. Eine DAO simuliert ein Unternehmen und soll weitgehend selbständig Entscheidungen treffen. Zusätzlich können aber Nutzer auch Anteile haben und über Entscheidungen abstimmten [24]. Gemeinsam können die Anteilseigner dann entscheiden wofür die Geldmittel der DAO eingesetzt werden. Waren die Investitionen profitabel, fließt der erwirtschaftete Gewinn wieder zurück in die DAO und kann gegebenenfalls an die Beteiligten ausgeschüttet werden [31].

V. Fazit und Ausblick

In der Ausarbeitung wurde beschrieben, wie dezentrale digitale Zahlungsmittel funktionieren. Insbesondere wurde dabei auf Bitcoin eingegangen, da es das erste weltweit erfolgreiche System in diesem Bereich ist und es eine Vielzahl von neuen Systemen gibt, die Konzepte von Bitcoin nutzen und diese weiterentwickeln. Die Analyse der unterschiedlichen Angriffsmethoden hat gezeigt, dass die unbedarfte Nutzung eines solchen dezentralen digitalen Zahlungsmittels Gefahren mit sich bringen kann. Die einfache Nutzung ohne die Beachtung von zusätzlichen Verhaltensregeln ist nicht zu empfehlen. Ein Nutzer sollte sich zunächst mit den Angriffen und Gegenmaßnahmen befassen. Nichtsdestotrotz verfügt Bitcoin über verschieden Sicherheitsmaßnahmen, die Angriffe erschweren. Außerdem ist anzumerken, dass in manchen Fällen nicht das Bitcoin-System an sich, sondern Dienste und Software von Drittanbietern angreifbar sind, beziehungsweise waren. Des Weiteren muss bei Angriffen, die einen monetären Nutzen verfolgen, abgewogen werden, ob es sich für einen Angreifer lohnt zum Beispiel Rechenleistung für den Angriff zu investieren oder anstatt dessen Mining zu betreiben.

Der Markt der Kryptowährungen ist zurzeit sehr unübersichtlich, da viele Alternativen zu Bitcoin entwickelt wurden. Manche davon besitzen bereits eine breite Nutzerbasis, andere hingegen werden kaum genutzt [2]. Die Alternativen bieten allerdings auch die Möglichkeit für Verbesserungen am bestehenden System. Weiterentwicklungen können auf bestimmte Anwendungsbereiche ausgerichtet sein und besitzen somit eine Daseinsberechtigung. Beispielsweise gibt es Altcoins, die auf Anonymität ausgelegt oder besser für die Zahlung kleiner Beträge geeignet sind. Defizite von Bitcoin können somit überwunden werden. Gleichzeitig können Weiterentwicklungen aber auch neue Gefahren mit sich bringen. Bitcoin als bekannter Vertreter von dezentralen digitalen Zahlungsmitteln war schon häufig Teil von wissenschaftlichen Untersuchungen, wohingegen die Alternativen bisher kaum untersucht wurden. Dies kann allerdings auch mit der großen Anzahl von unterschiedlichen Kryptowährungen zusammenhängen.

Zukünftige Arbeiten sollten die Sicherheit der Bitcoin-Alternativen analysieren und Empfehlungen benennen. Des Weiteren ist von Interesse welche Möglichkeiten die eingeführten Technologien in anderen Anwendungsbereichen in Zukunft bieten können.

Alles in allem bieten dezentrale digitale Zahlungsmittel und die verwendeten Technologien ein großes Potenzial. Sie können Beschränkungen von konventionellen Zahlungsmitteln überwinden und so neue Maßstäbe, zum Beispiel im Bereich der Privatsphäre setzen. Auf der anderen Seite ist die Akzeptanz und das Handelsvolumen im Vergleich zu den anderen Zahlungsmitteln noch relativ gering. In Zukunft wird sich herausstellen inwieweit sich welche Technologien durchsetzen und mit einer steigenden Nutzerbasis auch skalieren können.

LITERATUR

[1] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. Zugegriffen am 13.11.2016. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] CoinMarketCap. (2016) Crypto-currency market capitalizations. Zugegriffen am 08.01.2017. [Online]. Available: https://coinmarketcap.com/currencies/views/all/

[3] Blockchain Luxembourg S.A. (2016) Confirmed transactions per day - blockchain. Zugegriffen am 31.10.2016. [Online]. Available: https://blockchain.info/charts/n-transactions

[4] Heise Medien. (2016) Bitfinex-hack: 58 millionen euro gestohlen – bitcoin-kurs eingebrochen — heise security. Zugegriffen am 31.10.2016. [Online]. Available: https://www.heise.de/security/meldung/Bitfinex-Hack-58-Millionen-Euro-gestohlen-Bitcoin-Kurs-eingebrochen-3286784.html

[5] Bitcoin community. (2016) Common vulnerabilities and exposures - bitcoin wiki. Zugegriffen am 31.10.2016. [Online]. Available: https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures

[6] Bitcoin Project. (2016) Developer guide - bitcoin. Zugegriffen am 13.11.2016. [Online]. Available: https://bitcoin.org/en/developer-guide

[7] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.

[8] Theymos. (2010) File:blockchain.png - bitcoin wiki. Zugegriffen am 14.11.2016. [Online]. Available: https://en.bitcoin.it/wiki/File:Blockchain.png

[9] S. Barber, X. Boyen, E. Shi, and E. Uzun, *Bitter to Better — How to Make Bitcoin a Better Currency*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 399–414.

[10] C. Decker and R. Wattenhofer, *Bitcoin Transaction Malleability and MtGox*. Cham: Springer International Publishing, 2014, pp. 313–326. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11212-1_18

[11] G. Andresen. (2011) Bip 11: M-of-n standard transactions. Zugegriffen am 03.01.2017. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki

[12] ——. (2012) Bip 16: Pay to script hash. Zugegriffen am 03.01.2017. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki

[13] Symantec. (2011) Infostealer.coinbit — symantec. Zugegriffen am 07.12.2016. [Online]. Available: https://www.symantec.com/security_response/writeup.jsp?docid=2011-061615-3651-99

[14] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in *Proceedings of WEIS*, vol. 2013. Citeseer, 2013.

[15] P. Wuille. (2014) Bip 62: Dealing with malleability. Zugegriffen am 03.01.2017. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki

[16] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *CoRR*, vol. abs/1112.4980, 2011. [Online]. Available: http://arxiv.org/abs/1112.4980

[17] M. Vasek, M. Thornton, and T. Moore, *Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 57–71. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44774-1_5

[18] I. Eyal and E. G. Sirer, *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 436–454. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-45472-5_28

[19] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*. New York, NY: Springer New York, 2013, pp. 197–223.

[20] P. Koshy, D. Koshy, and P. McDaniel, *An Analysis of Anonymity in Bitcoin Using P2P Network Traffic*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 469–485. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-45472-5_30

[21] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 122–134.

[22] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 15–29. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660379

[23] G. Wood. (2014) Ethereum: A secure decentralised generalised transaction ledger. Zugegriffen am 31.10.2016. [Online]. Available: http://gavwood.com/paper.pdf

[24] E. Sixt, *Bitcoins und andere dezentrale Transaktionssysteme: Blockchains als Basis einer Kryptoökonomie*. Springer-Verlag, 2016.

[25] D. Bradbury, "Blockchain's," *Engineering Technology*, vol. 11, no. 10, pp. 44–47, Nov 2016.

[26] S. Sprankel. (2013) Technical basis of digital currencies. Zugegriffen am 08.01.2017. [Online]. Available: http://www.coderblog.de/wp-content/uploads/technical-basis-of-digital-currencies.pdf

[27] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *Security and Privacy (SP), 2013 IEEE Symposium on*, May 2013, pp. 397–411.

[28] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of namecoin and lessons for decentralized namespace design," in *Workshop on the Economics of Information Security (WEIS)*. Citeseer, 2015.

[29] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 459–474.

[30] Zerocash Project. (2016) Zerocash. Zugegriffen am 08.01.2017. [Online]. Available: http://zerocash-project.org/

[31] Ethereum Team. A next-generation smart contract and decentralized application platform. Zugegriffen am 08.01.2017. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[32] C. K. Frantz and M. Nowostawski, "From institutions to code: Towards automated generation of smart contracts," in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Sept 2016, pp. 210–215.

Frederik Gierschner