Blockchain und Distributed-Ledger-Technologie

Hoang Duong Nguyen
tabneib.github.io

April 2018

***Zusammenfassung*—Mit der Erfindung von Bitcoin hat die Blockchain die Aufmerksamkeit von der FinTech, den Regierungsbehörden, den Technologie-Giganten und einer Vielzahl von Start-ups auf sich gezogen. Einerseits hat sie die Distributed-Ledger-Technologie innoviert und verspricht ein breites Spektrum an Potenzialen. Andererseits wächst jedoch auch die Zahl der Risiken, die sich aus dem Einsatz der neuen Technologie ergeben. Gegenstand dieses Reports ist die Anwendung der Blockchain in der Distributed-Ledger-Technologie und die damit verbundenen Aspekte der Privatsphäre und Sicherheit.**

***Index Terms*—Blockchain, DLT, Konsensprotokoll, Privatsphäre, Sicherheit**

I. Einführung

Satoshi Nakamoto hat 2008 die Bitcoin-Kryptowährung eingeführt [1], die die Distributed-Ledger-Technologie innovierte. Zum ersten Mal wurde die Blockchain als Distributed-Ledger verwendet und das PoW-Konsensprotokoll von Nakamoto gilt als das erste Blockchain-Protokoll. Die Grundidee von Bitcoin ist ein Mittel zur Schaffung einer sicheren Währung, die keine zentrale Kontrolle benötigt. Alles wird durch einen Algorithmus bestimmt. Seitdem hat die Blockchaintechnologie die Aufmerksamkeit von der FinTech, den Regierungsbehörden, den Technologie-Giganten und einer Vielzahl von Start-ups auf sich gezogen. Im Laufe der Jahre wurden verschiedene Entwürfe und Implementierungen vorgeschlagen. Dabei unterscheiden sich Blockchainsysteme durch ihre Offenheit und können in mehrere Typen eingeteilt werden. Jeder Typ bietet seine Vor- und Nachteile. Mit dieser Vielfalt verspricht die Blockchaintechnologie ein breiteres Spektrum an Potenzialen als je zuvor. Allerdings muss sich die Technologie von Anfang an einer Reihe von Herausforderungen und Problemen im Hinblick auf die Sicherheit und Privatsphäre des Systemteilnehmers stellen. Diese könnten sich aus der Transparenz und nachträglichen Unveränderlichkeit der Blockchain oder aus den Schwachstellen verschiedener Komponenten und Schichten der Systeme ergeben. Dieser Report gibt einen kurzen Überblick über die Blockchaintechnologie als Grundlage für eine Taxonomie ihrer Probleme in Bezug auf ihre Sicherheit und Privatsphäre der Systemteilnehmer. Darüber hinaus werden mögliche Gegenmaßnahmen vorgestellt und diskutiert.

Der Report gliedert sich wie folgt: Das erste Kapitel gibt einen Überblick über das Konzept des Blockchainsystems, seine Struktur und die im Hintergrund liegenden kryptographischen Bausteine. In Kapitel II werden die zwei entscheidende Bestandteile der Anwendung von Blockchain in der Distributed-Ledger-Technologie vorgestellt, nämlich die Blockchain als eine Datenstruktur und das Konsensprotokoll. Kapitel III vertieft das Konzept des Blockchainsystems, indem ihre Kategorisierung und Anwendungspotenz eingeführt werden. Die relevanten Aspekte des Datenschutzes und der Sicherheit in Bezug auf Blockchainsysteme sind in Kapitel III bzw. Kapitel V zu finden, wo Probleme und mögliche Maßnahmen angesprochen, kategorisiert und diskutiert werden. Schließlich gibt Kapitel VI eine kurze Zusammenfassung.

A. Definition und wesentliche Bestandteile

Ein Distributed-Ledger ist ein System, an dem mehrere Teilnehmer beteiligt sind, die einen Konsens über einen Datensatz (d.h. das Ledger) erzielen und die Daten lokal speichern. Distributed-Ledger-Systeme werden unter verschiedenen Vertrauensmodellen (engl. *Trust Models*) mit unterschiedlichen Konsensprotokollen entwickelt. Es gibt zwei Haupttypen von Vertrauensmodellen: Einer geht davon aus, dass alle Teilnehmer gleichwertig sind, und einer hat Teilnehmer mit unterschiedlichen Berechtigungen für den Aufbau des Datensatzes [2]. Bitcoin mit seinem innovativen PoW-Konsensprotokoll von Nakamoto und der Hilfe der Blockchain-Datenstruktur hat es ermöglicht, dass ein Distributed-Ledger von jedermann offen und vollständig verteilt betrieben werden kann. Von da an wurde eine breite Palette weiterer Blockchainbasierter Distributed-Ledger-Systeme vorgeschlagen. In diesem Bericht wird ein solches System als Blockchainsystem bezeichnet.

Das Blockchainsystem lässt sich grob in vier Schichten aufteilen, die als wesentliche Bestandteile betrachtet werden können [3]. Die Clients stehen an der Spitze und beobachten einen abstrakten Systemzustand, wie z.B. eine Bilanz. Diese Abstraktion wird durch die Virtual-Machine-Schicht erleichtert, die Transaktionen akzeptiert und in Zustandsänderungen übersetzt. Die Konsensprotokoll- und P2P-Netzwerk-Schichten beschreiben, wie und wo die Knoten miteinander interagieren. Diese beiden Schichten koordinieren den Zustand des Systems unter einer großen Anzahl von Maschinen, so dass Teilnehmer mit jedem von ihnen interagieren und den gleichen Systemzustand beobachten können.

B. Kryptographische Grundlagen

1) Public-Key-Kryptographie: Im Jahr 1976 führten Diffie und Hellman [4] eine neue Art der Kryptographie ein, die zwischen Verschlüsselungs- und Entschlüsselungsschlüsseln unterscheidet. Klassische Kryptographie erfordert, dass Absender und Empfänger einen gemeinsamen Schlüssel teilen. Hierbei wird ein gemeinsamer Schlüssel zwischen dem Absender und

dem Empfänger wie in dem Fall klassischer Kryptographie nicht erfordert. Ein Algorithmus erzeugt ein mathematisch zusammenhängendes Schlüsselpaar, das aus einem privaten und einem öffentlichen Schlüssel besteht [5]. Der private Schlüssel wird dann geheim gehalten und benutzt, um mit dem zugehörigen öffentlichen Schlüssel verschlüsselten Nachrichten zu entschlüsseln.

2) Kryptographische Hashfunktionen: Ein wesentliches kryptographisches Primitiv der Blockchaintechnologie ist die kryptographische Hashfunktion. Eine Hashfunktion wird als eine mathematische Funktion mit den folgenden drei Eigenschaften definiert [6]:

- Die Inputs sind beliebig lange Strings.
- Die Outputs sind Strings fester Länge.
- Die Funktion ist effizient berechenbar.

Die oben genannte Eigenschaften definieren eine allgemeine Hashfunktion. Damit eine Hashfunktion kryptographisch sicher ist, werden die folgenden zwei zusätzlichen Eigenschaften benötigt [7]:

Kollisionsresistenz: Eine Kollision tritt auf, wenn zwei verschiedene Inputs den gleichen Output erzeugen. Alle Hashfunktionen besitzen Kollisionen, weil sie nicht injektiv sind [6]. Eine Hashfunktion H ist jedoch kollisionsresistent, wenn das Herausfinden einer Kollision schwer ist (d.h. es ist schwer, zwei Nachrichten M und M' zu finden, so dass $H(M) = H(M')$ [7]).

Verstecken: Wenn wir den Wert s als Output der Hashfunktion H für einen Input x erhalten, dann ist es praktisch unmöglich, herauszufinden, was der Input x war [6].

3) Digitales Signaturverfahren: Zusammen mit kryptographischen Hashfunktionen gilt das digitale Signaturverfahren als wesentlicher kryptograpfischer Baustein der Blockchaintechnologie. Eine digitale Signatur soll das digitale Analogon zu einer handschriftlichen Unterschrift auf Papier sein und wird mithilfe der Public-Key-Kryptographie realisiert. Ein digitales Signatursverfahren besteht aus den folgenden drei Algorithmen [6]:

- $(sk, pk) := generateKeys(keysize)$
  Die $generateKeys$ Methode nimmt eine Schlüsselgröße und generiert ein Schlüsselpaar. Der private Schlüssel sk is geheim aufbewahrt und zum Signieren von Nachrichten verwendet. pk ist der öffentliche Verifizierungsschlüssel. Jeder, der pk besitzt, kann die zugehörigen Signaturen verifizieren.
- $sig := sign(sk, msg)$
  Die $sign$ Methode nimmt eine Nachricht msg und einen privaten Schlüssel sk als Inputs und gibt eine Signatur sig für Nachricht msg unter sk aus.
- $isValid := verify(pk, msg, sig)$
  Die $verify$ Methode nimmt eine Nachricht msg, eine Signatur sig und einen öffentlichen Schlüssel pk als Inputs. Der zurückgegebene boolesche Wert $isValid$ ist $True$, wenn sig eine gültige Signatur für die Nachricht msg unter dem öffentlichen Schlüssel pk, und $False$ andernfalls.

II. Distributed-Ledger-Technologie

Während das letzte Kapitel einen Überblick über das Blockchainsystem gibt, betrachten wir in diesem Kapitel das System als ein Distributed-Ledger. Dabei sind zwei wesentliche Eigenschaften zu erfüllen. Zum einen, hinzugefügte Einträge können nicht mehr verändert oder gelöscht werden. Zum anderen, ein Konsensmechanismus wird zur Verfügung gestellt, der die Übereinstimmung aller Teilnehmer des Systems mit der Gültigkeit der hinzugefügten Einträge sicherstellt. Zunächst wird das Konzept der Blockchain erklärt. Anschließlich werden verschiedene Konsensmechanismen diskutiert.

A. Die Blockchain

Der Name *Blockchain* leitet sich von seiner technischen Struktur ab - eine Kette von Blöcken. Ein Block ist eine Datenstruktur, die es erlaubt, eine Liste von Transaktionen zu speichern, die von Peers des Blockchainnetzwerks erstellt und ausgetauscht werden. Jeder Block ist mit dem vorherigen Block durch einen kryptographischen Hashert verknüpft. Der Zustand der Blockchain ist damit die Reihenfolge der Blöcke zusammen mit deren Inhalt. Aufgrund der Sicherheit der Hashfunktion bietet die Blockchain die Eigenschaft von nachträglicher Unveränderlichkeit der gespeicherten Daten. Dieser Abschnitt erläutert den Aufbau dieser Bausteine der Blockchain im Einzelnen und in Verbindung mit den anderen.

1) Transaktion: Eine Transaktion besteht aus einer Reihe von Inputs bzw. Outputs und hat eine eindeutige ID [8]. Die Inputs verweisen auf Outputs gespeicherter Transaktionen, die dem Aussteller der aktuellen Transaktion geschickt wurden. Die Gültigkeit einer Transaktion ist daher u.a. von der Gültigkeit der entsprechenden vorgangenen Transaktionen abhängig und lässt sich mittels digitalen Signaturen verifizieren. Als Beispiel wird die Bitcoin-Transaktion in Abbildung 1 dargestellt. Mehrere Transaktionen werden in einem Block gespeichert, was sich anschließend verifizieren lässt. Ein als gültig verifizieter Block wird danach auf der Blockchain abgespeichert.

2) Skript: Viele öffentliche Blockchainsysteme wie Bitcoin und Ethereum und private Blockchainsysteme wie Hyperledger unterstützen die Möglichkeit, Skript für die Verarbeitung von Transaktionen zu kodieren. Diese Funktion wurde entwickelt, um den Ideen von Smart-Contract oder vollwertigen Programmen, die auf der Blockchain laufen, praktische Gestalt zu geben [9].

Zum Beispiel bieten Bitcoin-Transaktionen mit Hilfe von Skripten viel mehr Flexibilität als nur die einfache Übertragung von digitalen Werteinheiten. Das Transaktionsskript bietet einen gewissen Grad an Programmierbarkeit, was genau eine Transaktion tut. Scripting in Bitcoin wird durch eine einfache stapelbasierte Sprache realisiert. Allerdings ist sie nicht Turing-vollständig konzipiert, so dass es leichter zu benutzen und unbeabsichtigte Nebenwirkungen vermieden werden können. Im Gegensatz dazu ist die Ethereum-Skriptsprache Turing-vollständig und erleichtert so das Schreiben von Smart-Contracts auf der Blockchain (vgl. Abschnitt III-B1).

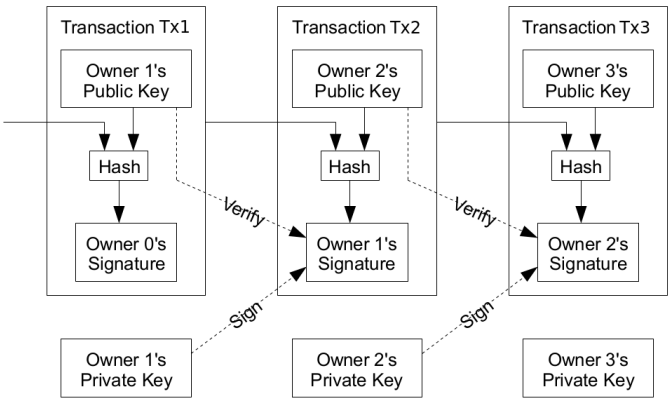


Abbildung 1: Bitcoin-Transaktion: Die Transaktionen werden mit Hilfe ihrer digitalen Signaturen verkettet, die von den entsprechenden Teilnehmern signiert werden. Die von *Owner 0* ausgestellte Transaktion Tx1 wird als Input für Tx2 verwendet, indem der Austeller *Owner 1* den Hashwert vom öffentlichen Schlüssel des Empfänger *Owner 2* zusammen mit der gesamten *Tx1* signiert. Jeder Teilnehmer kann danach ohne Weiteres mithilfe des in *Tx1* enthaltenen öffentlichen Schlüssel von *Owner 1* die $verify$ Methode des digitalen Signaturverfahrens ausführen, um zu verifizieren, dass *Tx1* als gültiger Input für *Tx2* gilt. [1]

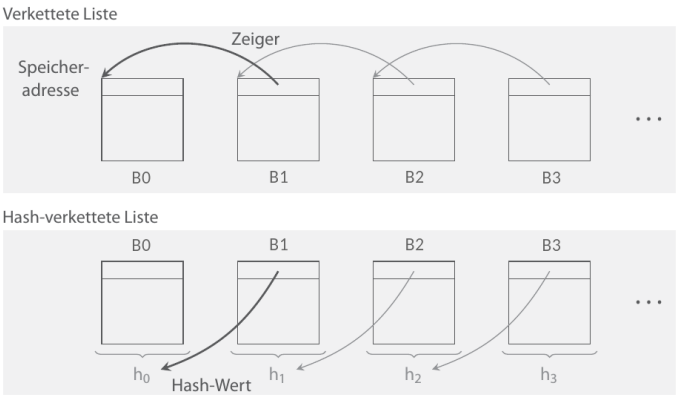


Abbildung 2: Von der verketteten Liste zur Blockchain [10]

3) Block: Die Kette von gespeicherten Blöcken könnte sich als Weiterentwicklung der verketten Liste betrachten lassen [10]. Anstelle von Zeigern auf Speicheradressen der anderen Listenelemente wie in einer verketteten Liste verweisen die Blockchainblöcke mithilfe kryptographischer Hashfunktionen auf den anderen Blöcke. D.h. der übliche Zeiger wird mit dem Hashwert des Zielblocks ersetzt (vgl. Abbildung 2). Der gesamte Inhalt eines Blocks B_1, einschließlich des (Hash-) Zeigers auf dessen vorherigen Block B_0, wird gehasht und der entsprechende Hashwert wird in den nächsten Block B_2 als Verweisung auf B_1 verwendet. Auf dieser Weise lässt sich die Blockchain sequenziell durchlaufen.

4) Nachträgliche Unveränderlichkeit: Die Konstruktion einer Blockchain erfolgt auf der Basis der kryptographischen Hashfunktion bzw. des digitalen Signaturverfahrens. Daher besitzt diese Datenstruktur die Eigenschaft von nachträglicher Unveränderlichkeit der gespeicherten Daten: Neue Blöcke können ohne Änderung von bestehenden Blöcke angehängt und nur durch Änderung gespeicherten Blöcke eingefügt bzw. gelöscht werden. Dies liegt daran, dass das Anhängen eines neuen Blocks nur den Hashwert des zuletzt angehängten Blocks erfordert, und dass eine kleine Änderung bzw. das Löschen eines Blocks auf der Blockchain zu einer Änderung dessen Hashwerts führt, was die gesamte Kette von Blöcken inkonsistent und daher ungültig macht.

Insgesamt werden Transaktionen in den Blöcken auf einer Blockchain gespeichert, die mithilfe kryptographischer Hashfunktionen erweitert und verifiziert wird. Diese Vorgehenweise garantiert die nachträglicher Unveränderlichkeit von Daten. Darüber hinaus gilt der Zustand der Blockchain als die Reihenfolge aller angehängten Blöcke zusammen mit deren Inhalt. Wie es sich ermöglichen lässt, den Zustand einer verteilten Blockchain in dezentralem System zum Konsens zu bringen, wird in dem nächsten Abschnitt beschrieben.

B. Konsensprotokolle

Im letzten Abschnitt wurde erwähnt, dass der Zustand der Blockchain als die Reihenfolge aller angehängten Blöcke zusammen mit deren Inhalt gilt. Bei der Blockchain handelt es sich allerdings um eine verteilte Datenbank, die eine ständig wachsende Liste von Datensätzen enthält. D.h. jeder Knoten des Netzwerks behält eine Kopie der Blockchain, die sich von der des anderen Knotens eventuell unterscheiden könnte. Eine Methodik ist aus diesem Grund erforderlich, damit sich der gemeinsame Zustand der Blockchain bestimmen lässt. Dies ist die Aufgabe eines Konsensprotokolls, welches sicherstellt, dass die Knoten sich auf eine eindeutige Reihenfolge einigen, in der Einträge angehängt werden. In diesem Abschnitt wird ein Überblick über das Konsensprotokoll, dessen Implementierungen für das Blockchainsystem und ein Vergleich dieser verschiedenen Implementierungen gegeben.

1) Konsensprotokoll: Konsensprotokolle ermöglichen die sichere Aktualisierung eines verteilten Shared-State. Die übliche Technik, die zur Erreichung der Fehlertoleranz in einem verteilten System verwendet wird, ist die Verteilung des gemeinsamen Zustands auf mehrere Replikate im Netzwerk. Diese Technik wird als *State Machine Replication* benannt. Dabei ermöglichen die vordefinierten Zustandsübergangsregeln die Aktualisierung des replizierten Shared State. Darüber hinaus kommunizieren die Repliken auch miteinander, um einen Konsens zu schaffen und sich auf die Endgültigkeit des Zustands zu einigen, nachdem ein Zustandswechsel durchgeführt wurde.

Allerdings können die Knoten abstürzen, sich bösartig verhalten, gegen das gemeinsame Ziel verstoßen werden, oder die Netzwerkkommunikation wird unterbrochen. Für die Bereitstellung eines kontinuierlichen Dienstes führen die Knoten daher ein fehlertolerantes Konsensprotokoll durch [11]. Mehrere Protokolle werden in der Forschungsliteratur vorgeschlagen, wobei jedes Protokoll die erforderlichen Annahmen in Bezug auf Synchronität, Nachrichtenübertragungen, Ausfälle,

bösartige Knoten, Leistung und Sicherheit der ausgetauschten Nachrichten trifft. Im Allgemeinen hat ein Konsensprotokoll drei Schlüsseleigenschaften [12]:

- *Safety*: Alle Knoten produzieren den gleichen Output und die von den Knoten erzeugten Outputs sind nach den Regeln des Protokolls gültig.
- *Liveness*: Alle nicht fehlerhaften Knoten, die am Konsens teilnehmen, produzieren schließlich einen Output.
- *Fehlertoleranz*: Das Protokoll kann nach einem Ausfall eines Knotens wiederhergestellt (engl. *recover*) werden.

Fehlertoleranz bezieht sich auf zwei Arten von Fehlern in verteilten Systemen. Fail-Stop-Fehler behandeln Knotenfehler, die dazu führen, dass Knoten sich nicht mehr am Konsensprotokoll teilnehmen. Dahingegen ist die zweite Art von Fehlern byzantinische Fehler, die dazu führen, dass sich Knoten unberechenbar verhalten. Im Kontext der Blockchaintechnologie führen die Knoten ein fehlertolerantes Konsensprotokoll durch, dass sie sich über die Reihenfolge der Einträge in der Blockchain einig sind. Die Bereitstellung eines kontinuierlichen Dienstes wird damit sichergestellt. Darüber hinaus entscheiden sich aufgrund der Natur verschiedener Arten von Blockchainsysteme zwei Kategorien von Blockchain-Konsensprotokolle. Zum einen werden in einem öffentlichen System, die an eine Kryptowährung gekoppelt sind und keine feste Anzahl von Teilnehmern hat, spezifische Konsensprotokolle eingesetzt, die auf dem Arbeitsnachweis (engl. *Proof-of-Work*, PoW) und dessen Varianten basieren [11]. Zum anderen verwendet ein privates System, im dem alle Telnehmer bekannt sind, klassisches Konsensprotoll wie z.B. verschiedene byzantinische fehlertolerante Protokolle [12].

Es ist zu beachten, dass Abschnitt III-A vor dem Rest dieses Kapitels gelesen werden sollte, falls der Leser noch nicht mit der Kategorisierung von Blockchainsysteme vertraut ist.

2) Konsensprotokolle für öffentliche Blockchains: In einem öffentlichen Blockchainsystem wie Bitcoin oder Ethereum kann jeder ein Teilnehmer werden oder einen Knoten betreiben. Jeder kann durch Aufruf von Transaktionen in den gemeinsamen Zustand schreiben und jeder kann sich am Konsensprozess zur Bestimmung des gültigen Zustandes teilnehmen. Darüber hinaus ist die Anzahl der Knoten unbekannt und es wird erwartet, dass sie groß ist Außerdem sind diese Knoten anonym und nicht vertrauenswürdig.

Konsensprotokolle für ein solches System müssen widerstandsfähig gegen Bösartigkeiten sein, und insbesondere gegen Sybil-Angriffe. Sybil-Angriffe auf ein Blockchain-Netzwerk können es einem einzelnen Teilnehmer ermöglichen, mehrere Identitäten zu generieren, um den Konsensprozess zu beeinflussen und zu manipulieren. In [1] stellte Nakamoto in seinem Bitcoin-Kryptowährungssystem zum ersten Mal ein Konsensprotokoll vor, das dieses Problem angeht, indem es die Konsensrunde rechenintensiv gestaltet. Die Knoten müssen nachweisen, dass sie als PoW für die Lösung eines schweren kryptographischen Rätsels (engl. *cryptographic puzzle*) eine beträchtliche Menge an Energie aufgewendet haben. Dieser Ansatz, auch wenn er in Bezug auf die Energiekosten teuer ist, ist notwendig, um die Sicherheit des Konsensprozesses zu gewährleisten. Von da an wurden viele neue Vorschläge vorgeschlagen, die auf dem Konsens von Nakamoto beruhen. Im Folgenden wird ein Überblick über PoW, *Proof-of-Stake* (PoS), und *Proof-of-Elapsed-Time* (PoET) gegeben.

a) PoW: In Bitcoin-PoW muss jeder der Knoten das PoW-Puzzle lösen, damit andere Teilnehmer im System seinen vorgeschlagenen Block als gültig verifizieren. Eine Nonce wird hierbei gefunden und in den Block hinzugefügt, so dass der Hashwert des Blocks kleiner als eine bestimmte Zahl ist, die als der vom Netzwerk dynamisch eingestellte Schwierigkeitsgrad gilt. Der Prozess, das PoW-Puzzle zu lösen, wird als *Mining* bezeichnet. Der erste Knoten, der die korrekte Nonce findet, kann seinen vorgeschlagenen Block an die Blockchain anhängen und auch die Mining-Belohnung beanspruchen. Aufgrund der verteilten, parallelen Natur dieses Prozesses ist es manchmal möglich, dass mehrere gültige vorgeschlagene Blockchains gleichzeitig in das P2P-Netzwerk gesendet werden und erzeugt dadurch eine temporäre Verzweigung. Ein ehrlicher Knoten wählt dabei immer die längste oder die zuerst ankommende Blockchain. Dies führt zu einer eventuellen Konsistenz zwischen allen Knoten bezüglich des Zustandes der (lokal gespeicherten) Blockchain. Die Chance eines Knotens, der nächste Knoten zu sein, der die Blockchain erweitert, ist daher proportional zu seinem prozentualen Anteil an der gesamten Rechenleistung.

b) PoS: PoS-Konsensprotokolle sollen die Nachteile von PoW-Konsensprotokolle hinsichtlich des hohen Stromverbrauchs im Mining-Prozess überwinden. PoS ersetzt den Mining-Prozess vollständig durch einen alternativen Ansatz, bei dem die Beteiligung eines Teilnehmers (ihre Eigentum an virtueller Währung im Blockchainsystem) einbezogen wird. Die Chance eines Knotens, der nächste Knoten zu sein, der die Blockchain erweitert, ist daher proportional zu seinem prozentualen Anteil an der gesamten Menge der virtueller Währung.

c) PoET: PoET verlässt sich auf *Trusted Execution Environment* (TEE). Ein Leader, der den nächsten Block erstellt, wird aus allen verfügbaren teilnehmenden Knoten zufällig durch das Protokoll gewählt und die anderen Knoten sind in der Lage, den Leader ohne Weiteres zu verifizieren. Alle Validierungs- und Mining-Knoten müssen hiebei das TEE betreiben.

Zunächst fordert jeder Knoten vom im TEE laufenden Code eine zufällige Wartezeit (engl. *Enlapsed Time*) an. Wenn ein Knoten behauptet, ein Leader zu sein, muss er auch einen im TEE generierten Nachweis erbringen, den andere Knoten leicht verifizieren können. Es muss bewiesen werden, dass der genannte Knoten die kürzeste Wartezeit hatte, und er tatsächlich für diese Zeitspanne wartete. Anschließend wird dem Knoten gestattet, den nächsten Block zu erstellen. Die Funktionen innerhalb des TEE sind so konzipiert, dass ihre Ausführung nicht durch externe Software manipuliert werden kann und damit die Sicherheit des Konsensprotokolls zu gewährleisten.

3) Konsensprotokolle für private Blockchains: Ein privates Blockchainsystem ist genehmigungsbasiert (vgl. Abschnitt

Tabelle I: Vergleich der Konsensprotokollkategorien [12]

	PoW	***PoS***	***PoET***	***BFT***	***FBFT***
Typ des Blockchain-systems	Genehmigungsfrei	Beide	Beide	Genehmigungsbasiert	Genehmigungsbasiert
Transaktionsendgültigkeit	Probabilistisch	Probabilistisch	Probabilistisch	Unmittelbar	Unmittelbar
Transaktionsrate	Niedrig	Hoch	Mittel	Hoch	HOch
Token erforderlich?	Ja	Ja	Nein	Nein	Nein
Teilnahmekosten	Ja	Ja	Nein	Nein	Nein
Skalierbarkeit des Peer-Netzwerks	Hoch	Hoch	Hoch	Niedrig	Hoch
Vertrauensmodell	Unvertrauenswürdig	Unvertrauenswürdig	Unvertrauenswürdig	Semi-vertrauenswürdig	Semi-vertrauenswürdig
Adversary-Toleranz	$\leq 25\%$	Vom verwendeten Algorithmus abhängig	Unbekannt	$\leq 33\%$	$\leq 33\%$

III-A) und wird von bekannten Knoten betrieben, die den gemeinsamen Zustand kontrollieren und aktualisieren können. Die zentrale Autorität ist in der Lage, zu kontrollieren, wer Transaktionen ausgeben kann. Genehmigungsbasiertes Blockchainsystem adressiert viele der Probleme, die im Bereich des verteilten Systems über Jahrzehnte untersucht wurden, vor allem für die Entwicklung von Byzantinischen fehlertoleranten (BFT) Systemen [11]. Nachfolgend wird ein Überblick über die BFT-Protokolle sowie die föderierte BFT-Protokolle gegeben.

a) BFT: Das Praktische BFT-Protokoll [13] war die erste praktische Lösung zur Erzielung eines Konsenses angesichts des byzantinischen Fehlers. Es verwendet das Konzept der Replicated-State-Machine und der Abstimmung durch Repliken für Zustandsänderungen. Dieser Algorithmus erfordert $3f + 1$ Repliken, um f ausfallende Knoten tolerieren zu können. Weitere Varianten sind wie z.B. das *SIEVE*-Konsensprotokoll zur Behandlung vom Nichtdeterminismus bei der Ausführung von Chaincodes (d.h. Smart-Contracts), das Cross-Fault Tolerance (XFT) Protokoll mit vereinfachtem Angriffsmodell zur Ermöglichung von effizienter Realisierbarkeit des BFT für praktische Szenarien [12].

b) FBFT: FBFT-Konsensprotokolle werden in Plattforme wie Ripple und Stellar eingesetzt. FBFT weicht von der traditionellen Sicherheitsannahme für Konsensprotokolle ab, indem es ihre Vertrauensannahmen flexibel macht. Jeder Knoten deklariert selbstständig, welchen Knoten er vertraut, anstatt eine Protokoll-weite Annahme [11]. Diese zugehörige Blockchain-Plattformen zielen auf Finanzen-Use-Cases und insbesondere auf den Payment-Bereich ab. Sie stellen Zahlungsprotokolle zur Verfügung, die grenzüberschreitende Transaktionen in Sekundenschnelle abwickeln können [12].

Tabelle I gibt einen Vergleich der Konsensprotokollkategorien, die in diesem Abschnitt vorgestellt wurde. Transaktionsendgültigkeit gibt an, ob die Transaktion, die einem Block auf der Blockchain einmal hinzugefügt wurde, als endgültig gilt. Skalierbarkeit des Peer-Netzwerks ist die Fähigkeit, einen Konsens zu erzielen, wenn die Anzahl der Peering-Knoten ständig zunimmt. Das Vertrauensmodell legt fest, ob die am Konsens beteiligten Knoten bekannt oder vertrauenswürdig sein müssen. Adversary-Toleranz ist der Anteil des Netzwerks, der kompromittiert werden kann, ohne dass der Konsens beeinträchtigt wird.

III. Blockchainsysteme

In den ersten Kapitel wurde grundlegende technische Details des Blockchainsystems beschrieben. Dieses Kapitel behandelt jedoch das System als zentrales Objekt, indem seine Einordnungen und Anwendungen dargestellt werden.

A. Einordnung von Blockchainsystemen

In dieser Sektion wird die Einordnung der Blockchainsystemen aus Teilnehmersicht vorgestellt. Die Einordnung kann durch verschiedene Kriterien erfolgen [14] [15]. Die Kernfrage ist, wer zugelassen wird, sich an dem System teilzunehmen, und welche Rechte Teilnehmer des Systems zugeteilt werden. Hierbei lassen sich in jedem Blockchainsystem zwei Teilnehmergruppen unterscheiden, welche als Schreiber und Leser beschrieben werden. Ähnlich wie in einem Datenbanksystem bezeichnen wir als Schreiber jede Entität, die in der Lage ist, den Zustand der Datenbank zu aktualisieren. Damit ergibt sich die Teilnahme desjenigen an der Verwaltung der Blockchain, d.h. am Konsensprotokoll zu beteiligen und das Wachsen der Blockchain zu unterstützen. Darüber hinaus bezeichnen wir einen Leser als jede Entität, die die Blockchain nicht verlängert, sondern sich am Prozess der Dateninputerstellung teilnimmt, indem sie die Blockchain liest und analysiert oder auditiert. Dabei dürfen die Leser neue Dateninputs (d.h. Transaktionen) vorschlagen. Als Beispiel dafür kann das Broadcast von Transaktionen in Bitcoin oder Ethereum genannt werden. Die Schreiber sind hierzu in der Lage, gültige Transaktionen innerhalb eines Blocks zu akkumulieren und diesen Block an die Blockchain anzuhängen. Dabei ist zu beachten, dass die Regulatoren und Blockchain-Software-Betreuer für außerhalb dieses Bereichs gehalten werden. [16]

a) Private und öffentliche Blockchainsysteme: Zunächst wird die Offenheit eines Blockchainsystems zum Kriterium aufgestellt. Hierbei lassen sich Blockchainsysteme darin unterscheiden, ob sie privat oder öffentlich sind, indem es entscheidend ist, durch wen sich die Systeme verwenden lassen. Die Zuteilung von dem Zugriff auf die Daten bzw. von der Berechtigung, Dateninputs vorschlagen zu dürfen,

sagt aus, welcher Katagorie sich das entsprechende System zuordnen lässt. Ist diese Verwendung jedermann gestattet, ist das Blockchainsystem als öffentlich anzusehen. Im Gegensatz, ist sie auf eine Organisation oder ein Konsortium beschränkt, handelt es sich um ein privates System [14]. Bitcoin [1] und Ethereum [17] sind Instanzen von öffentlichen Blockchainsystemen. Jeder Peer kann dem Netzwerk jederzeit als Leser beitreten. Hierbei gibt es keine zentrale Stelle, die die Mitgliedschaft verwaltet, oder die unrechtmäßige Leser verbieten könnte. Diese Offenheit impliziert, dass der geschriebene Inhalt für jedermann lesbar ist. Mit dem Einsatz von kryptographischen Primitiven ist es jedoch technisch machbar, eine öffentliche Blockchain so zu entwerfen, dass die datenschutzrelevante Informationen verbirgt werden können. Ein weitaus bekanntes anonymes System ist zum Beispiel Zerocasch [18] (vgl. Abschnitt IV-C).

b) Genehmigungsfreie und genehmigungsbasierte Blockchainsysteme: Ein weiteres mögliches Kriterium für die Einordnung von Blockchainsystemen besteht darin, ob zur Teilnahme am Verwaltungsprozess der Blockchain, d.h. die Blockchain zu erweitern, eine Genehmigung erforderlich ist [19]. In einem Genehmigungsfreien System ist jedermann in der Lage, neue Blocke an die Blockchain anzuhängen. Demgegenüber gäbe es eine Partei wie ein Konsortium oder eine zentrale Autorität, die die Berechtigung zur Unterstützung des Wachsens der Blockchain genehmigt, dann handelt es sich um ein genehmigungsbasiertes Blockchainsystem [14]. Die Vermeidung von einer zentralen Partei ist bei genehmigungsfreien Systemen von Vorteil, jedoch in praktischen Systemen wie Bitcoin oder Ethereum wird diese Berechtigung nur auf die Teilnehmer, die einen sogenannten Arbeitsnachweis gelingen können, beschränkt [1], [17]. Die bekannste Arbeitsnachweismaßnahme ist das energieaufwendige und folglich kostenintensive Bitcoin-PoW, welches zu ökonomische Barrieren führt. Diese nachteilige Barrieren sind demgegenüber hinfällig, wenn die Blockchainverwaltung nur für vorher gewälte Teilnehmer gestattet ist. Aus diesem Grund können in genehmigungsbasierten Systemen effizientere Mechanismen zur Konsensfindung implementiert werden [20]. Als Beispiel kann das Ripple-Netzwerk genannt werden. Ripple ist ein global operierendes Austauschnetzwerk mit eingebauten Krypto-Währungen. Im Gegensatz zu Bitcoin lässt sich das Mining-Prozess dabei nicht durch den Arbeitsnachweis ermöglichen, sondern durch eine genehmigte Art und Weise . Es weist eine erhöhte Effizienz auf, indem ausgewählte Netzknoten über den aktuellen Status des Systems abstimmen [21].

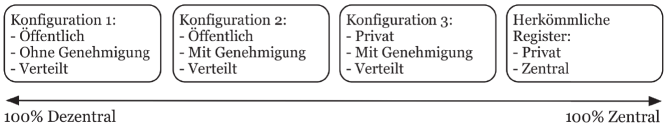


Abbildung 3: Der Grad der Zentralisierung verschiedener Blockchainsysteme [22]

Aus den obenerwähnten Einordnungskriterien können Blockchainsystemen in drei Konfigurationen kategorisiert werden [22]. Der daraus abgeleitete Zentralisierungsgrad wird in Abbildung 3 dargestellt. Darüber hinaus ist es bekannt, dass genehmigungsfreie Systeme in der Regel öffentlich sind, und dass genehmigungsbasierte Systeme hingegen in der Regel privat sind [14]. Eine angemessene Gegenüberstellung der Gewinne und Verluste zwischen Systemen unterschiedlicher Kategorien ist von großem Nutzen bei der Entscheidung, ob eine Blockchain tatsächlich die geeignete technische Lösung für ein bestimmtes Anwendungsszenario ist. Wüst und Gervais [16] untersuchen die Unterschiede zwischen genehmigungsfreien und genehmigungsbasierten Blockchainsystemen und kontrastieren deren Eigenschaften mit denen einer zentral verwalteten Datenbank. Eine strukturierte Methodik, die als eine technische Richtlinie für die erwähnte Entscheidungsfrage spielt, wird mithilfe dieser Untersuchung zur Verfügung gestellt. Der entsprechende Entscheidungsfluss wird in Abbildung 9 dargestellt. Der Einsatz von Blockchainsystemen wird in den folgenden Fällen nicht empfohlen:

1) Es müssen keine Daten gespeichert werden: Eine Blockchain als Datenbankform wird nicht benötigt.
2) Es gibt nur einen Schreiber: Eine Blockchain bietet keine zusätzlichen Garantien und eine zentrale Datenbank ist aufgrund des besseren Durchsatzes und der besseren Latenz besser geeignet.
3) Ein vertrauenswürdiger Dritter ist verfügbar und immer online: Schreiboperationen können an ihn delegiert werden und er kann als Verifizierer eingesetzt werden.
4) Alle Fälle von 1 bis 3 treten nicht auf, alle Schreiber sind bekannt und die Schreiber vertrauen sich gegenseitig.

Im Gegensatz dazu werden in den folgenden Fällen verschiedenen Typen von Blockchainsystemen empfohlen:

6) Genehmigungsfreie Blockchainsysteme: Alle Fälle von 1 bis 3 treten nicht auf und die Menge der Schreiber ist nicht fest und den Teilnehmern bekannt, wie es bei vielen Kryptowährungen wie Bitcoin der Fall ist.
7) Öffentliche genehmigungsbasierte Blockchainsysteme: Alle Fälle von 1 bis 4 treten nicht auf und öffentliche Verifizierbarkeit ist erforderlich.
8) Privates genehmigungsbasierte Blockchainsysteme: Alle Fälle von 1 bis 4 treten nicht auf und es ist keine öffentliche Verifizierbarkeit erforderlich.

Tabelle II listet die von den Autoren untersuchte Unterschiede zwischen den Systemen auf.

B. Blockchain Anwendungen

Für umfangreiche praktische Anwendungsbereiche bietet die Blockchaintechnologie viele besondere Potenziale. Einerseits lässt es sich mittels der Dezentralisierung und Transparenz des Blockchainsystems ermöglichen, Transaktionen und darauf basierte Prozesse rund um Identität, Dokumente und Assets verschiedener Art zu vereinfachen und beschleunigen. Andererseits bietet die Technologie aufgrund ihrer Eigenschaften, nämlich die nachträgliche Unveränderlichkeit von Daten und die Widerstandsfähigkeit durch Vermeidung von

Tabelle II: Vergleich verschiedener Blockchainsysteme und zentraler Datenbank

	Genehmigungsfreie Blockchain	*Genehmigungsbasierte Blockchain*	*Zentrale Datenbank*
Durchsatz*	Niedrig	Hoch	Sehr Hoch
Latenz	Langsam	Medium	Schnell
#Leser	Hoch	Hoch	Hoch
#Schreiber	Hoch	Niedrig	Hoch
#Nicht vertrauenswürdiger Schreiber	Hoch	Niedrig	0
Konsensprotokolle	Hauptsächlich PoW, einige PoS	BFT protocols	Keine
Zentral verwaltet	Nein	Nein	Ja

*Die Anzahl der Transaktionen pro Sekunde, die ein System tätigen kann

kritischen Einzelkomponenten sichere Speicherung und Validierung der Transaktionen. Momentan werden Blockchainsysteme von zahlreichen Unternehmen in verschiedenen Bereiche zum Einsatz gebracht, vor allem die Banken und Versicherungswesen. Hinter den Kulissen der Technologie sind die drei Grundbausteine ihrer praktischen Anwendung, die die Technologie zum Erfolg bringen und zunächst vorgestellt werden. Anschließend wird ein Überblick über die Einteilung der Blockchainanwendungen nach Branchen und Kategorien gegeben. Dabei wird auch diskutiert, auf welcher Art und Weise die Blockchainanwendungen vernünftig eingeordnet werden sollen.

1) Grundbausteine von Blockchainanwendungen: Zunächst werden drei Grundbausteine der Blockchainanwendungen vorgestellt. Zudem zählen Kryptowährungen, Smart-Contracts, und Dezentrale Autonome Organisationen (DAO) [22].

- *Kryptowährungen*: Der früheste Versuch, digitale Währung zu bauen, wurde 1982 von David Chaum veröffentlicht [23]. Im Jahr 1996 brachten Law et al. verschiedene elektronische Zahlungssysteme ein [24]. Diese ersten digitalen Währungen erforderten jedoch immernoch eine Bank als zentrale Autorität [25]. Das auf Blockchain basierende Bitcoinsystem [1] ist hingegen unabhängig von einer zentralen Partei und besitzt daher die Schwachstelle der Vorgängern nicht. Bitcoin etablierte zum ersten Mal ein System, in dem die kryptographische Prinzipien, in der Regel gepaart mit einem Arbeitsnachweisschema, verwendet werden, um die Währung zu erschaffen und zu verwalten. Im Lauf der Zeit werden allmählig eine Vielzahl an alternativen Kryptowährungen entworfen und realisiert [26]. Zu den Vorteile der auf Blockchain basierten Kryptowährungen zählen die Teilbarkeit in sehr kleine Einheiten, Geringe Transaktionskosten und -zeiten durch die Umgehung von Intermediären und Unmöglichkeit der Fälschung von Kryptowährungen aufgrund der Eigenschaften der Blockchain-Datenstruktur [22].
- *Smart-Contracts*: Smart-Contracts sind als Computerprogramme zu verstehen, die Entscheidungen treffen können, wenn bestimmte Konditionen erfüllt werden [22], [27]. Dazu können in einem Smart-Contract Regeln des Vertrages festgelegt werden, über die externe Informationen als Input eine bestimmte Aktion hervorrufen. Der Inhalt eines Smart-Contracts lässt sich durch ein Skript spezifizieren, das mittels einer Transaktion auf der Blockchain abgespeichert wird. Beim Auftreten eines externen Ereignises löst sich eine entsprechende Transaktion aus, die als Input für die Transaktion des Vertrages gilt [28]. Zu den Vorteilen der auf Blockchain basierten Smart-Contracts zählen die autonome Ausführung des Vertrages , die Unmöglichkeit von störenden Eingriffe dritter Parteien in der Ausführung des Vertrages, die echzeitige Vertragsausführung, die geringe Vertrags-, Durchsetzungs- und Compliance-Kosten im Vergleich zu regulären Verträgen, die Möglichkeit, die Ausführung eines Smart-Contracts von externen Ereignissen abhängig zu machen,fairer Austausch zwischen zwei Vertragsparteien ohne intermediäre Partei möglich, und die Minimierung der Interaktion zwischen den Vertragsparteien [22].
- *Dezentrale Autonome Organisationen*: Eine DAO ist als ein dezentrales Netzwerk autonomer Subjekte, denen eine leistungsmaximierende Produktionsfunktion zugrunde liegt, zu verstehen [29]. Die Blockchaintechnologie ermöglicht den Teilnehmern, DAO zu implementieren auf der Basis der Smart-Contracts. Diese DAO-Implementierungen können unter der auf den Verträge festgelegten Bedingung, ausreichende Ressourcen zu erhalten und unabhängige Handlungen durchzuführen [30]. Die Transparenz, die nachträgliche Unveränderlichkeit der Daten, und die Dezentralisierung des Blockchainsystems bietet die auf Blockchain basierende DAO Vorteile gegenüber Schwächen und Missbräuchen regulärer Organisationen [21].

2) Einordnung von Blockchainanwendungen: Zur Zeit existieren mehrere Einsätze zur Einordnung von Blockchainanwendungen. Verschiedene Kriterien werden zum Einsatz gewählt wie zum Beispiel zeitliche [29] oder technische [31] Einordnung. Diese Ansätze sind jedoch entweder nicht trennscharf oder zu kritisieren, da zur Einordnung nur inkonsistent technisch-konzeptionelle Aspekte berücksichtigt werden [22]. Dahingegen werden in [22] die Fintechs im Blockchain-Umfeld auf der Basis der Kategorisierung von William Mougayar [32] untersucht. In dieser Kategorisierung lassen sich 4 aufeinander aufgebaute bzw. sich gegenseitig bedingte Anwendungskategorien von Blockchain unterscheiden, nämlich Infrastruktur und Plattformen, Middleware-Services, Applikationen und Nebenleistungen. Den Middleware-Services liegen die Infrastruktur und Plattformen zugrunde, die die benötigten Ressourcen zur Ermöglichung der jeweiligen Services

bereitstellen. Als Beispiele für eine Infrastruktur bzw. ein Middleware-Service lassen sich Kryptowährung und das Konzept Smart-Contract nennen. Eine Blockchainapplikation wird wiederum auf der Basis eines Middleware-Services aufgebaut wie zum Beispiel ein konkretes Smart-Contract. Die Nebenleistung umfasst wie etwa Dienstleister für die Bereitstellung von Marktdaten, Fachmedien oder branchenspezifische Kapitalgeber.

In [22] wurde eine Analyse von 222 Unternehmen aus dem Umfeld der Blockchain durchgeführt. Daraus entstand ein Datensatz von 245 Datenreihen, da einige der 222 Unternehmen mehrere Produkte anboten. Die Unternehmen wurden nach Branchen und den oben erwähnten Kategorien eingeordnet. Die entsprechende Verteilungen werden in Abbildung 4 bzw. 5 dargestellt.

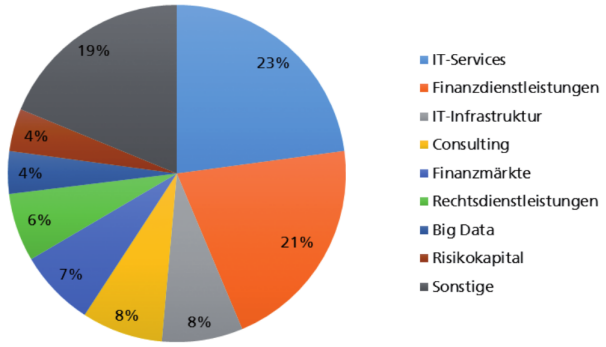


Abbildung 4: Verteilung der Unternehmen nach Branchen [22]

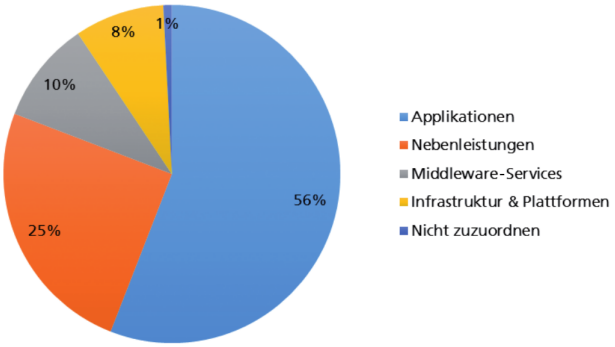


Abbildung 5: Verteilung der Unternehmen nach Kategorien [22]

IV. Privatsphäre und die Blockchaintechnologie

Die durch die Blockchaintechnologie ermöglichten Anwendungspotenziale haben in den letzten Jahren einen faszinierten Durchbruch im Einsatz von der Technologie in zahlreichen Branchen und Unternehmen geschaffen. Abgesehen von dem dabei erbrachten Gewinn sind verschiedene Probleme bezüglich des Schutzes der Privatsphäre in Blockchainsysteme zu lösen. Einerseits wird Blockchain als Mittel eingesetzt, um die Privatsphäre in anderen Systemen zu erhöhen [33], [34]. Andererseits wachsen die Studie und praktische Entwicklung vom Datenschutz im Rahmen der Blockchaintechnologie rasant. Hier sei hervorgehoben, dass öffentliche und private Blockchainsysteme, von denen aufgrund ihrer Natur, unterschiedliche Schwierigkeiten an dieser Stelle konfrontieren. Dieses Kapitel widmet sich den auftretenden Problemen des Datenschutzes in der Blockchaintechnologie, deren rechtlichen Aspekt, und der Einsicht in die herkömmlichen Gegenmaßnahmen.

A. Privatsphärenprobleme der Blockchaintechnologie

Laut [35] ist die Privatsphäre als kontrollierte Veröffentlichung von sensiblen Daten zu verstehen. D.h. Privatsphäre ist das Recht zu kontrollieren, wer bestimmte Aspekte über den Betroffene, dessen Kommunikation und Aktivitäten kennt. Dieser Abschnitt gibt einen Überblick über die Probleme in der Kontrolle der veröffentlichten sensiblen Daten in Blockchainsysteme. Wir beginnen zunächst mit den allgemeinen Probleme. Anschließend werden die Probleme, die spezifisch für öffentliches bzw. privates Blockchainsystem sind, dargestellt.

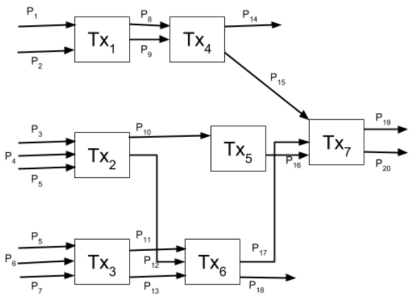
1) Allgemeine Privatsphärenproblem: Eine wichtige Eigenschaft des Blockchainsystemes ist Transparenz. D.h. die auf der Blockchain gespeicherte Daten sind zugreifbar und deren Gültigkeit ist validierbar für alle Teilnehmer des Systems. Wer diese Teilnehmer sind, ist davon abhägig, ob es sich um ein öffentliches oder privates System geht. Die Transparenz der Blockchain ergibt jedoch in beiden Fällen Schwierigkeit in dem Schutz des Privatsphäre der Blockchaindaten [3], [8]. Dazu zählen sich Identitäts- und Transaktionsprivatsphäre.

In einem öffentlichen System, wie in meisten Kryptowährungen der Fall, kann jedermann sich am System teilnehmen und dabei alle Transaktionen ablesen und beobachten. Beispielweise in Bitcoinsystem, was nur eine partielle Unverkettbarkeit bietet, ist es möglich, eine Reihe von Transaktionen mit einem einzelnen Bitcoin-Benutzer zu verknüpfen, indem der Werteinheintenfluss durch ein robustes Blockchain-Analyseverfahren verfolgt wird [8].

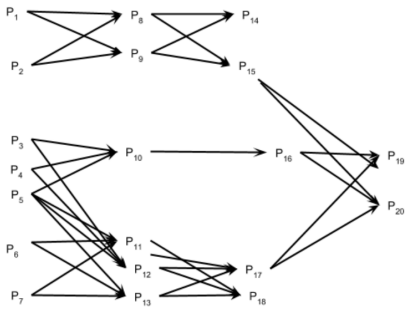
Private Blockchainnsysteme sind vor allem experimentelle und operationelle Lösungen in der Fintech-Industrie. Zwei Banken, die untereinander Gelder abwickeln, wollen möglicherweise nicht, dass eine dritte Bank diese Transaktionen verfolgen kann, sowohl aus Wettbewerbsgründen als auch aus Gründen der Privatsphäre ihrer Kunden [3].

Insgesamt tritt in Blockchainsysteme das Privatsphärenproblem bezüglich der Identitäten der Systemteilnehmer und der von denen durchgeführten Transaktionen auf. Aufgrund der Transparenz des Systems sind die Transaktionsinformationen auf der Blockchain exponiert. Jeder Systemteilnehmer könnte in der Lage sein, die Identität von anderen Teilnehmern, wer möglicherweise Konkurrenten sind, daraus abzuleiten, oder Transaktionen zu verketten.

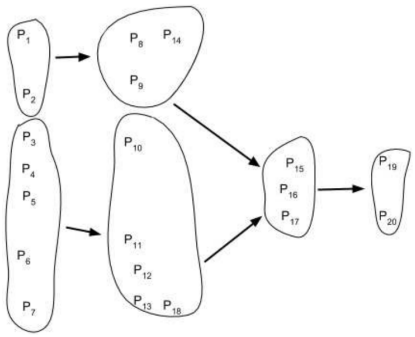
2) Spezifische Probleme des öffentlichen Blockchainsystems: Auf Grund der Offenheit konfrontieren öffentliche Blockchainsysteme spezifische Schwierigkeiten, die mit den



(a) Die gesamte Blockchain kann als ein azyklischer Transaktionsgraph betrachtet werden. Die Knoten sind Transaktionen. Input/Output-Kanten sind die entsprechenden öffentlichen Schlüssel der Input/Output-Transaktionen.



(b) Durch das Traversieren des Transaktionsgraphen lassen sich die Beziehung zwischen verschiedenen Input- und Outputadressen leicht ableiten. Dies erstellt einen Adressgraph.



(c) Durch die Verwendung des Adressgraphen zusammen mit den vom Bitcoinprotokoll abgeleiteten Heuristiken ermöglicht es, Adressen zu gruppieren, die dem gleichen Teilnehmer zu gehören scheinen.

Abbildung 6: Analyseverfahren des Transaktionsgraphen eines öffentlichen Blockchainsystems am Beispiel Bitcoin [8]

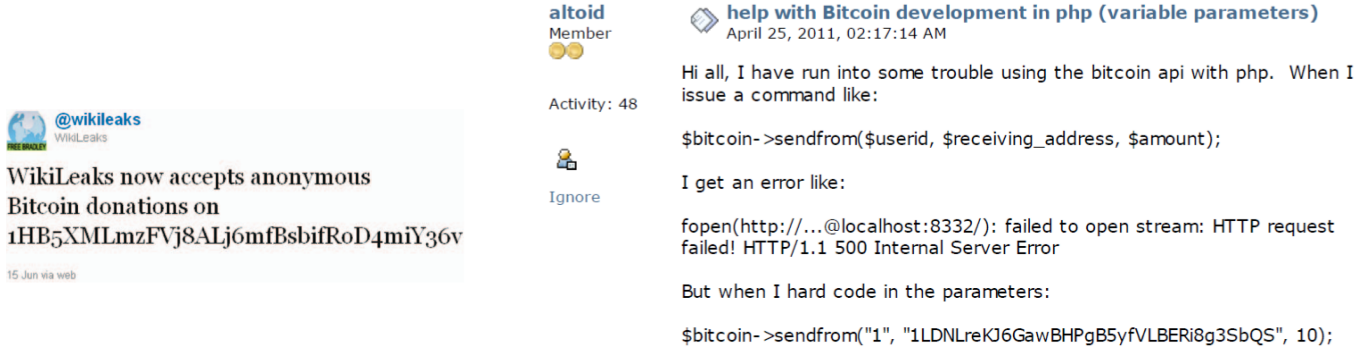
aus dem unterliegenden P2P-Netzwerk bzw. den externen Informationen stammenden Daten gekoppelt sind. Diese Daten, zusammen mit den aus der Blockchain hergeleiteten Informationen, ermöglicht es, das System zu deanonymisieren und dabei die Identitäten der Systemteilnehmern zu verraten.

Obwohl die Teilnehmeridentitäten in öffentlichen Blockchainsystemen wie Ripple und Bitcoin geschützt sind, wird das transaktionale Verhalten der Teilnehmer (wie z.B. Zeit der Transaktion und Anzahl der Werteinheiten) exponiert [36]. Unmittelbar aus den auf der Blockchain gespeicherten Transaktionen können die Teilnehmer mittels eines Analyseverfahrens mit einer Reihe von öffentlichen Adressen (Öffentliche Schlüsseln der Teilnehmern) verknüpft werden. Reid und Harrigan [37], Ron und Shamir [38] zeigten, dass sich verschiedene Netzwerke aus den öffentlichen Blockchaindaten mithilfe geeigneter Heuristiken ableiten lassen. Die topologische Struktur derjenigen Netzwerke kann komplementäre Ansichten des Blockchainsystems, die Auswirkungen auf die Anonymität haben, liefern (vgl. Abbildung 6).

Weiterhin basieren das öffentliche Blockchainsystem und dessen Konsensprotokoll auf einem unterliegenden P2P-Netzwerk [1], [8], [22], was wiederum öffentlich ist. D.h. jedermann kann sich am Netzwerk teilnehmen und Verkehrsdaten der Netzkommunikation beobachten und speichern [8], [10]. Diese Daten können Informationen über den Systemteilnehmer verraten und deren Privatsphäre dadurch verletzen. Koshy et al. [39] beschreiben eine Methodik, um öffentliche Bitcoin-Adressen direkt auf IP-Adressen, was möglicherweise als sensibel angesehen können, abzubilden. Die Autoren verwenden einen speziell angefertigten Bitcoin-Knoten, um die Netzwerkdaten zu erheben. Die Daten werden dementsprechend mithilfe anomalen Transaktionsrelaismustern (engl. *transaction relay patterns*) und geeigneten Heuristiken verarbeitet, um Paarungen von Bitcoin-Adressen und IP-Addressen zu erkennen. Weiter Methoden, die einen Super-Node einsetzen, sind auch vorgestellt [40]–[42]. Biryukov et al. zeigen, dass es möglich ist, die öffentlichen Schlüssel der Bitcoin-Teilnehmer mit ihren IP-Adressen mit einer Genauigkeit von fast 30% zu verknüpfen [41]. Darüber hinaus vermuten Conti et al., dass noch höhere Genauigkeiten erreicht werden könnten, wenn ausgeklügeltere Techniken zur Analyse des Netzwerkverkehrs eingesetzt werden [8].

Schließlich haben sich um öffentliche Blockchainsysteme externe Datenquellen wie Forumsbeiträge oder Ökosysteme von Intermediären gebildet [8], [10]. In der Tat benötigen viele Dienstleister wie Online-Verkäufer oder Austauschdienste die Teilnehmeridentität, bevor sie einen Dienst anbieten [8]. Es wird in dem Fall von Bitcoin gezeigt, dass diese Daten sich mit der auf der Blockchain gespeicherten Daten kombinieren lassen können, um einen Diebstahl von digitalen Werteinheiten durchzuführen [37]. In [43] führen die Autoren mit hoher Präzision die Verknüpfung von (Bitcoin-) Addressenclustern mit den Online-Wallets, Anbietern und anderen Dienstleistern, mit denen die Systemteilnehmer mehrere Interaktionen haben, durch. Dadurch werden Verknüpfungen von Adressclustern mit den realen Teilnehmeridentitäten enthüllt. Eine weitere Methode ist die Nutzung des Web-Crawlers, um nach Adressen der Blockchainteilnehmern, die in den sozialen Netzwerken exponiert werden, zu suchen [37], [44]. Beispielsweise verwerden u.a. WikiLeaks und Silk Road öffentlich bekannte Bitcoin-Adressen mit Absicht (vgl. Abbildung 7a), oder die Teilnehmer legen unabsichtlich ihre öffentliche Schlüsseln offen (vgl. Abbildung 7b). Diese Datenlecks wären für das Zuordnen der Addressen von großem Nutzen.

Insgesamt gesehen ergeben sich augrund der Offenheit und Transparenz des öffentlichen Blockchainsysteme verschiedene Probleme, die jedermann ausnutzen kann, im Schutz der Teilnehmer- bzw. Transaktionsprivatsphäre. Gegenmaßnahmen gegen diese Schwachstellen werden in Abschnitt IV-C diskutiert.



(a) Beabsichtigtes Leck: Bildschirmaufnahme eines Tweets von WikiLeaks, der die Annahme von anonymen Bitcoin-Spenden ankündigt [37].

(b) Unbeabsichtigtes Leck: Silk Road-Besitzer Dread Pirate Roberts enthüllt unbeabsichtigt seinen öffentlichen Schlüssel dem Online-Forum bitcointalk.org [44]

Abbildung 7: Öffentliches Blockchainsystem: Datenlecks aus externen Quellen

3) Spezifische Probleme des privaten Blockchainsystems: Private Blockchainsysteme haben einen niedrigeren Grad an Dezentralisierung als öffentliche Systeme (vgl. auch Abschnitt III-A). D.h. es gibt eine zentrale Autorität, die in der Lage ist, Lesezugriff und Schreibzugriff zuzuteilen bzw. die zugeteilte Rechte zu verwalten. Aus diesem Grund entstehen in solchem System spezifische Privatsphärenprobleme. Hierbei werden die Identitätenprivatsphäre der Teilnehmern gegen die zentrale Autorität und die optionale Offenlegung (engl. *Optional Disclosure*) berücksichtigt.

Alle private Blockchainsysteme sind genehmigungsbasiert [22]. In genehmigungsbasierten privaten Blockchainsystemen besteht das Risiko, dass die Systemteilnehmer den Grad der durch genehmigungsfreie öffentliche Systeme gebotenen Anonymität verlieren. In öffentlichen Systeme wie Bitcoin und Ethereum generieren die Teilnehmer selbst Schlüsselpaare, die in deren Transaktionen verwendet werden. Daher ist der Teilnehmer der einzige, der seine privaten Schlüsseln kennt. Diese Selbstgenerierung ermöglicht das System, partielle Anonymität (d.h. Pseudonymität) zu erreichen [8]. Im Gegensatz dazu besteht bei der Gestaltung von genehmigungsbasierten Blockchainsystemen die Versuchung, die Internet-Identität des Teilnehmers einfach mit dessen öffentlichen Schlüssel zu verknüpfen. Dadurch ist es möglich, die Zugriffskontrolle über die private Blockchain zu erzwingen. Diese Verknüpfung kann jedoch dazu führen, dass die reale Identität des Teilnehmers, der den öffentlichen Schlüssel besitzt, offengelegt wird. Dies wiederum kann die soziale Akzeptanz von genehmigungsbasieren privaten Blockchainsysteme einschränken und deren Akzeptanz auf private Organisationen oder geschlossene Konsortien beschränken [45].

In [45] sind Hardjono und Pentland der Meinung, dass neue Lösungen für die optionale Offenlegung von Identitäten bei Transaktionen, die in Frage kommen (z.B. Anti-Geldwäsche (engl. *AML*) oder Regelkonformität), erforderlich sind. Ein Systemteilnehmer ist in der Lage, mehrere nicht verknüpfbare Transaktionsschlüssel auf der Blockchain zu besitzen. Darüber hinaus ermöglicht die optionale Offenlegung-Funktion denjenigen, seinen Besitz eines dieser Schlüsseln offen zu legen, ohne dabei die Sicherheit und Privatsphäre seiner verbleibenden anderen Schlüssel auf derselben Blockchain zu beeinträchtigen. Eine solche Offenlegung ist beispielweise für den Fall einer gerichtlicher Anfechtung vorhanden.

Schließlich konfrontieren private Blockchainsysteme die Privatsphärenprobleme bezüglich der Teilnehmeridentität und der optionalen Offenlegung von Transaktionen. Die beiden Probleme stehen im Zusammenhang mit der zentralen Autorität, was als ein Merkmal eines privaten Systems angesehen wird. Vorgeschlagene Lösungen dafür werden in Abschnitt IV-C vorgestellt.

B. Datenschutzrechtlicher Aspekt

Zunächst werden verschiedene Anwendungsbereiche des Datenschutzrechts, ihre Vorgaben und die zugehörige Schwierigkeiten diskutiert.

1) Anwendungsbereich des Datenschutzrechts: Böhme und Pesch [10] widmen sich der Untersuchung von datenschutzrechtlichen Aspekten der Blockchaintechnologie und gehen der Frage nach, ob und welche datenschutzrechtliche Regelungen Anwendung auf die Verarbeitung von Blockchain-Daten finden. Hiermit lassen sich der sachlichen bzw. räumlich-persönlichen Anwendungsbereiche des Datenschutzrechts berücksichtigen.

Im Folgenden werden DS-RL als EU-Datenschutzrichtlinie [1] und DS-GVO als Datenschutzgrundverordnung [2] verstanden.

a) Sachlicher Anwendungsbereich: Privatsphäre ist die kontrollierte Veröffentlichung von sensiblen Daten [35]. Was

[1]Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr

[2]Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG, gem. ihrem Art. 99 Abs. 2 anzuwenden ab Mai 2018.

als sensible Daten und die entsprechende Kontrolle zu verstehen sind, wird durch §1 Abs.1 BDSG, Art.1 Abs.1 DS-RL, Art.1 Abs.1 DS-GVO definiert. Hierbei betrifft das Datenschutzrecht nur die Verarbeitung personenbezogener Daten. Gemäß §3 Abs.1 BDSG sind personenbezogene Daten Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person zu verstehen. Diese Definition entspricht inhaltlich der in Art.4 Nr.1 Hs.1 DS-GVO [10]. Gemäß Art.2 lit.a Hs.2 DS-RL und Art.4 Nr.1 Hs.2 DS-GVO wird eine Person als bestimmbar angesehen, wenn sie direkt oder indirekt identifiziert werden kann. Hieran wird die Frage aufgeworfen, ob die gespeicherte Daten auf der Blockchain sich potenziell auf natürliche Personen beziehen. Böhme und Pesch behaupten, dass eine Verknüpfung von kryptographischer Identität mit der dazugehörigen natürlichen Person davon abhängt, ob der für die Datenverarbeitung Verantwortlichen in der Lage ist, die Identifizierung des Betroffenen mit vernünftigerweise einsetzbaren Mitteln zu erreichen. Als einsetzbare Mittel werden von den Autoren Zusatzinformationen und die Hilfe Dritter genannt.

b) Räumlich-persönlicher Awendungsbereich: Neben der sachlichen Anwendbarkeit des Datenschutzrechts kommt seine räumlich-persönliche Anwendbarkeit auch in Frage. Zunächst werden die relevante Punkte von dem räumlichen Anwendungsbereich, der verantwortlichen Stelle und der gemeinsamen Veranwortung mit Bezug auf dem Blockchainsystem vorgestellt.

Gemäß §1 Abs.5 BDSG ist die Anwendbarkeit des Datenschutzrechtes davon abhängig, wer den jeweiligen Datenverarbeitungsvorgang veranlasst. Ist die verantwortliche Stelle im Ausland belegen, so gilt je nachdem, ob es sich um ein EU-Mitgliedstaat bzw. Vertragsstaat des Europäischen Wirtschaftsraums oder ein anderes Ausland handelt. Dazu muss auch die Niederlassung mitberücksichtigt werden [10].

Böhme und Pesh schließen aus §3 Abs.4 BDSG, dass die Blockchainknoten, die Daten an andere Knoten übermitteln und in die mehrseitig überprüfbare Datenstruktur eintragen. Gemäß §3 Abs.7 BDSG ist eine verantwortliche Stelle jede Person oder Stelle, die personenbezogene Daten für sich selbsterhebt, verarbeitet oder nutzt oder dies durch andere im Auftrag vornehmen lässt. Daher sind die erwähnte Blockchainknoten verantwortliche Stellen.

Bei einem Blockchainsystem handelt es sich um ein komplexes Datenverarbeitungssystem, in dem es jedem Teilnehmer erlaubt ist, auf seine Weise am System teilzunehmen, um zusammen mit den anderen den Zustand des Systems zu bestimmen [10]. Gemäß Art.26 Abs.1 Satz 1 DS-GVO ergibt sich eine gemeinsame Verantwortliche, wenn zwei oder mehr Verantwortliche gemeinsam die Zwecke der und die Mittel zur Verarbeitung festlegen. Daher sind die gemeinsame Verantwortung der Knoten für die Verarbeitung personenbezogener Daten zu betrachten.

Insgesamt gelten verschiedene Aspekte der Anwendarkeit des Datenschutzrechts im Kontext des Blockchainsystems zu beachten. Die Anwendbarkeit ist daher vom einzelnen Fall abhängig. Die in jedem oben erwähnten Aspekt aufgeworfenen Probleme werden in Abschnitt IV-B3 eingehen.

2) Datenschutzrechtliche Vorgaben: Zunächst werden die datenschutzrechtliche Vorgaben in Bezug auf personenbezogene Daten, die aus der Blockchain erhoben und verarbeitet werden oder Dritter an das Blockchainsystem übermittelt, aufgelistet.

- *Einwilligung:* Gemäß §4 Abs.1 BDSG gestattet eine Einwilligung der Betroffene die Erhebung, Verarbeitung und Nutzung personenbezogener Daten. In Frage kommt es, ob und inwieweit die Teilnehmer in die Verarbeitung ihrer personenbezogenen Daten einwilligen.
- *Gesetzliche Erlaubnis/Zweckbindung:* Das Erheben und Verarbeiten personenbezogener Daten ist zulässig unter bestimmten Voraussetzungen. Im Kontext des Blockchainsystems gelten die Fälle wie z.B wenn die Datenverarbeitung für die Durchführung eines rechtsgeschäftlichen Schuldverhältnisses erforderlich ist (gem. §28 Abs.1 Nr.1 Var.2); wenn die Daten allgemein zugänglich sind (gem. § 28 Abs.1 Nr.3 Var.1). Letzteres gilt insbesondere für öffentliche Blockchainsysteme.
- *Betroffenenrechte:* Die Betroffenen haben Rechte auf Auskunft (§§ 19, 34 BDSG) und auf Benachrichtigung von der Erhebung bzw. Berichtigung, Löschung oder Sperrung (§§ 20, 35 BDSG) von personbezogenen Daten.
- *Datensparsamkeit:* Gemäß §3a Satz 1 BDSG sind so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen.

3) Schwierigkeiten in der Anwendung zentraler Datenschutzgrundsätze: Aufgrund der dezentralisierten Natur des unterliegenden P2P-Netzwerks und der Eigenschaft von nachträglichen Unveränderlichkeit von Daten konfrontiert die Anwendung zentraler Datenschutzgrundsätze mit mehreren Schwierigkeiten im Kontext von Blockchainsystemen. Ob und in welchem Umfang solche Schwierigkeiten auftreten, hängt darüber hinaus davon ab, ob es sich um ein öffentliches oder privates Blockchainsystem handelt.

Es wird oben erwähnt, dass ein Blockchainknoten als verantwortliche Stelle gilt. In einen öffentlichen System werden sich die Knoten allerdings wegen der Offenheit und Dezentralisierung des Systems nicht immer einfach lokalisieren und indentifizieren lassen. Dazu ergibt sich der Zustand des Systems nicht aus einer Einigung der Knoten, sondern aus der Gesamtheit ihres unabhängigen Verhaltens. Aus diesem Grund ist der Einfluss einzelner Knoten so gering, dass ihre datenschutzrechtliche Inanspruchnahme keinen Erfolg verspricht. Im Gegensatz dazu sind die Knoten in einem privaten Blockchainsystem lokalisier- und identifizierbar. Dies ist auf die Tatsache zurückzuführen, dass die zentrale Autorität jedem Knoten die Schreibberechtigung zuteilt, und dadurch steht jeder Knoten in einer herausgehobenen Position [10].

Des Weiteren treten auch verchiedene Probleme auf beim Genügen von in Abschnitt IV-B2 aufgelisteten datenrechtlichen Vorgaben. Im öffentlichen Blockchainsystem verbergen sich die Betroffenen hinter den Pseudonymen (d.h. ihre öffentliche Schlüssel). Falls eine Benachrichtigung an den Betroffene gemäß §33 Abs.1 Satz 1 BDSG erforderlich ist,

liefe die entsprechende Identifizierung immerhin den Datenschutzinteressen Betroffenen zuwider. Andererseits ergeben sich Probleme hinsichtlich der Betroffenenrechte sowohl in privaten als auch in öffentlichen Systemen durch die nachträgliche Unveränderlichkeit der Blockchaindaten. Etwaige Betroffenenrechte wie z.B das Löschen personbezogener Daten (§35 Abs.1,2 BDSG) können nicht durch eine Inanspruchnahme von zugangsvermittelnden Intermidiären oder Knoten durchgesetzt werden, weil es nicht möglich ist, die auf der langfristigen Blockchain gespeicherten Daten zu entfernen. Letztendlich besteht ein Zielkonflikt zwischen dem Prinzip der Datensparsamkeit und der für die Validierung des Blockchainzustands erforderten Datenmenge [10].

C. Maßnahmen zum Schutz der Privatsphäre in Blockchainsystemen

Bislang werden Privatsphärenprobleme aus sowohl technischen als auch datenschutzrechtliche Aspekte diskutiert. Ausgehend von der Diskussion wird deutlich, dass die Problematik zwischen den verschiedenen Arten vom Blockchainsystem variert, trotz der allgemeinen Konzepte von Identitäts- und Transaktionsprivatsphäre. Einerseits stellt die Natur des öffentlichen Systems eine erhebliche Bedrohung für die Privatsphäre der Teilnehmer dar. Andererseits entstehen im privaten System spezifische Probleme durch die Präsenz der zentralen Autorität und die Einhaltung von Datenschutzgesetzen. Verschiedene Gegenmaßnahmen zur Bewältigung dieser Probleme werden sowohl von der akademischen Gemeinschaft als auch von der Industrie vorgeschlagen. In diesem Abschnitt wird ein Überblick über die Vorschläge gegeben.

1) Maßnahmen für öffentliche Systeme: In Bezug auf das spezifische Problem für öffentliche Blockchainsysteme (vgl. Abschnitt IV-A2) sollte eine Maßnahme zur Ermöglichung der Privatsphäre im öffentlichen Blockchainsystem die Probleme der verkettbaren Transaktionen und Teilnehmeridentitäten, der geringen Anonymität des P2P-Netzwerks und der Verwendung externer Blockchaininformationen angehen. Hierfür gibt es derzeit zwei Herangehensweisen. Die eine ist die Verbesserung der Anonymisierung. Die andere Herangehensweise besteht darin, neue Blockchainsystem zu erstellen, um die Anonymisierung auf der Protokollebene zu schaffen [46].

a) Verbesserung der Anonymisierung: Es gibt mehrere Vorschläge zur Erweiterung der Privatsphäre in einem bestehenden öffentlichen Blockchainsystem, insbesondere einer Kryptowährung. Den Lösungen liegt zugrunde die Mixing-Methode, die sich entweder um ein P2P-Protokoll oder ein verteiltes Mixing-Netzwerk, das einen Dritten benötigt, handelt [8].

- Peer-to-Peer Mixing-Protokoll: Der Ansatz wurde zuerst durch CoinJoin realisiert [47], in dem nicht vertrauenswürdige Teilnehmer Transaktionen erstellen, ohne dass ein vertrauenswürdiger Dritter erforderlich ist. Die Teilnehmer finden zunächst Peers, die sich am Mixing-Prozess teilnehmen wollen, und tauschen Ein-/Ausgangsadressen aus. Die erzeugte Transaktion wird an alle Beteiligten verschickt, so dass sie von jedem unterschrieben werden kann. Schließlich übertragen sie die Transaktion an die Blockchain. Ziel ist es zu verhindern, dass ein Angreifer, der einen Teil des Netzwerks oder einen Teil an Teilnehmer kontrolliert, eine Transaktion mit dem entsprechenden ehrlichen Absender in Verbindung bringt. Der Grad der Anonymität in P2P-Protokollen hängt von der Anzahl der Teilnehmern in der Anonymitätseinstellung ab [8].
- Verteiltes Mixing-Netzwerk: Bonneau et al. [48] stellen Mixcoin vor, in dem ein Drittanbieter-Mischprotokoll anonyme Zahlungen in Bitcoin und ähnlichen Kryptowährungen ermöglicht. Ein Teilnehmer teilt zunächst eine Anzahl von Werteinheiten mit einem Drittanbieter-Mix über eine Transaktion einheitlicher Größe. Anschließend erhält er die gleiche Anzahl von Werteinheiten aus dem Mix, die ein anderer Teilnehmer eingibt, wodurch eine starke Anonymität gegenüber externe Einträge gewährleistet wird [8].

b) Anonymisierung auf der Protokollebene: Die andere Möglichkeit, die Anonymität zu gewährleisten, besteht darin, neue Primitive direkt in das Protokoll einzubauen. Die Autoren in [49] schlagen *ZeroCoin* als eine kryptographische Erweiterung von Bitcoin vor. Hierbei wird die Anonymität durch Design mithilfe des Zero-Knowledge-Proofs (ZKP) gewährleistet. Eine Erweiterung von Zerocoin mit dem Namen *ZeroCash* wird in [50] vorgestellt. ZeroCash verwendet eine verbesserte Version von ZKP namens *SNARKs*, die zusätzliche Informationen über Transaktionen wie Betrag und Empfängeradressen verbirgt, um starke Datenschutzgarantien zu erreichen [8].

Insgesamt ist das Ziel aller oben genannten Ansätze, die Verkettungen von Transaktionen und Teilnehmeridentitäten zu verhindern und somit den Schutz gegen Blockchain-Transaktionsgraphenanalyse zu erleichtern. Da sich die externen Blockchaininformationen für die meisten Fälle mit den aus den Blockchaindaten durch diejenige Analyse abgeleiteten Informationen verknüpfen lassen, bieten diese Methoden auch in gewissem Maße Schutz vor der Verletzung der Privatsphäre in Bezug auf externe Informationen. Darüber hinaus sind Peer-to-Peer Mixing-Protokolle wie z.B Dandelion [51] in der Lage, die netzwerkfähigen Deanonymisierung von Bitcoin-Benutzern zu verhindern, damit die Anonymität des unterliegenden P2P-Netzwerks zu verbessern. In Tabelle IV werden verschiedene herkömmliche Techniken am Beispiel Bitcoin - das bekannteste öffentliche Blockchainsystem, aufgelistet.

2) Maßnahmen für private Systeme: Im Gegensatz zum öffentlichen Fall, bei dem das Netzwerk vollständig dezentralisiert ist, verursacht die zentrale Autorität in einem privaten Blockchainsystem andere Arten von Datenschutzproblemen (vgl. IV-A3). Es wurden mehrere Lösungen vorgeschlagen, die sich mit dem allgemeinen Privatsphärenproblem sowie dem Schutz der Privatsphäre der Teilnehmer gegenüber der zentralen Autorität und der optionalen Offenlegung befassen.

a) Gegen allgemeines Problem: In privaten Systemen wie Hyperledger müssen die Teilnehmer erkannt werden, auch wenn sie sich nicht zwangsläufig gegenseitig voll vertrauen. Alle Parteien haben ihre eigene Kopie des verteilten

Ledgers und sehen nur bestimmte Transaktionen, die mit ihren Geschäften zusammenhängen. Eine bekannte Gruppe von Teilnehmern baut ein gemeinsames System auf einem gemeinsamen Ledger auf und vermeidet es, einen PoW-Konsensmechanismus zu betreiben. Nur bestimmte Knoten sind in der Lage, die Validierungen durchzuführen. Aus diesem Grund ist die Privatsphäre der Teilnehmern untereinander gewährleistet [52]. Allerdings sind in privaten Systemen wie Corda [53] die als Betreiber des Konsensprotokolls gewählte Knoten in der Lage, Information über Transaktionen, die von anderen Knoten ausgegeben werden, herauszuziehen, indem sie den Transaktionsgraphen durchlaufen. Die Autoren in [54] bewältigen dieses Problem, indem sie das Konzept der *Satellite Chains* einführen.

b) Gegen die zentrale Autorität und zur Ermöglichung der optionalen Offenlegung: In [45] stellen Hardjono und Pentland das ChainAnchor-System vor, das sich mit den Herausforderungen des Schutzes der Privatsphäre der Teilnehmer gegenüber der zentralen Autorität und der optionalen Offenlegung befasst. Die Schlüsseltechnik der Lösung ist der Einsatz von einem Direct-Anonymous-Attestation-Schema namens *EPID-ZKP* [55], was es erlaubt, mehrere verschiedene private Schlüssel mit einem öffentlichen Schlüssel zu verwenden. Dies ermöglicht es jedem einzelnen Teilnehmer einzigartige EPID-private Schlüssel zu verwenden, von denen jede Signatur durch den Genehmigungsverifizierer mit dem zugehörigen einzigen EPID-öffentlichen Schlüssel verifiziert werden kann. Der Teilnehmer bleibt daher gegenüber dem Genehmigungsverifizierer anonym, da er nicht zwischen validierten Teilnehmern unterscheiden kann. Wenn der Teilnehmer die Mitgliedschaft in der Gruppe beantragt, ist der Teilnehmer dem Genehmigungsaussteller bekannt. Der Teilnehmer fügt jedoch einen geheimen Blind-Parameter hinzu, wenn er den privaten Schlüssel des Teilnehmers erstellt. Der Teilnehmer bleibt daher ebenfalls gegenüber dem Genehmigungsaussteller anonym. Letztendlich ist ein Teilnehmer in der Lage, innerhalb des Blockchainsystems beliebig viele Transaktionsschlüssel zu verteilen. Dies ermöglicht es dem Teilnehmer, dem Genehmigungsverifizierer das Eigentum des Teilnehmers an einem bestimmten Transaktionsschlüssel offenzulegen, ohne dass andere Transaktionsschlüssel betroffen sind. Dies bedeutet, dass die optionale Offenlegung gewährleistet ist.

V. Sicherheit der Blockchaintechnologie

Das letzte Kapitel hat einen Überblick über den Datenschutzaspekt der Blockchaintechnologie gegeben. Wir haben gesehen, wie sich die Transparenz und nachträgliche Unveränderlichkeit der Blockchain auf verschiedene Arten von Systemen unterschiedlich auswirkt. In diesem Kapitel diskutieren wir den Sicherheitsaspekt der Technologie und versuchen, einen guten Ansatz zu finden, um die Probleme und Herausforderungen zu kategorisieren. Dieser Ansatz sollte die intrinsischen Eigenschaften der Blockchain, die unterschiedlichen Schwachstellen, die in verschiedenen Systemen, System-Schichten und -Komponenten enthalten sind, berücksichtigen. Dazu teilen wir die Sicherheitsprobleme in zwei Gruppen auf. Die erste sind die allgemeinen Sicherheitsprobleme, die nicht nur auf Blockchainsysteme, sondern auch auf andere Systeme auftreten. Die andere ist spezifisch für Blockchainsysteme. Der nächste Abschnitt gibt einen strukturellen Überblick über diese Probleme. Anschließend werden mögliche Gegenmaßnahmen entsprechend diskutiert.

A. Sicherheitsprobleme

Die Blockchaintechnologie implementiert fälschungssichere und dezentrale Ledger durch den Einsatz von P2P-Netzwerken, Kryptographie und Konsensprotokollen. Ihre Sicherheit basiert daher auf der Sicherheit dieser Komponenten. In diesem Abschnitt kategorisieren wir die Sicherheitsprobleme in sechs Gruppen. Die ersten vier Gruppen sind allgemeiner Probleme inklusive Standardisierung, Schlüsselverwaltung, Kryptografie und Netzwerksicherheit. Die beiden letzten sind spezifisch für Blockchainsysteme, nämlich Konsensprotokollsicherheit und Smart-Contractssicherheit.

1) Standardisierung: Matsuo [56] organisiert die Technologie-Schichten aus Sicht der System- und Anwendungssicherheit. Sie bestehen aus Kryptographie-Schicht, Backbone-Protokoll, Anwendungsprotokoll, Applikationslogik, Implementierung und Betrieb. Jede Schicht hat internationale Standards, um die Sicherheit der Sicherheitsmechanismen zu analysieren. Die Standards bezüglich der Technologieschichten sind in Abbildung 8 dargestellt. Die Kryptographie-Schicht wird durch den Standardisierungsprozess von ISO, NIST abgedeckt. Die Sicherheit des Backbone-Protokolls wird mit Hilfe der formalen Analyse und des Universal Composability Frameworks und der ISO/IEC 29128 analysiert. Die Sicherheit der Implementierung ist durch Common Criteria (ISO/IEC 15408) zertifiziert und der Betrieb des Systems wird mit ISMS und dem Framework der ISO/IEC 27000 Serie definiert und auditiert. Der Autor behauptet, dass die Applikationslogik, die eine Skriptsprache für Finanztransaktionen und Verträge enthält, noch keinen guten Standard für Sicherheitsanalysen hat und dass hier weitere Forschungen notwendig sind.

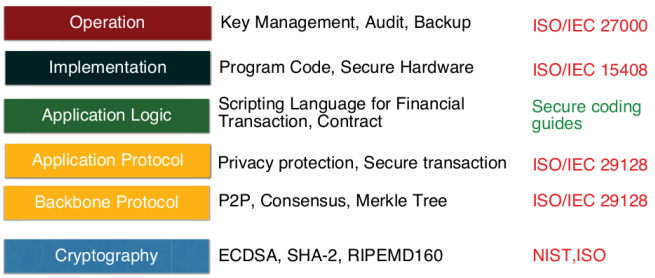


Abbildung 8: Technologie-Schichten und Sicherheitsstandards [56]

2) Schlüsselverwaltung: In öffentlichen Blockchainsystemen sind private Schlüssel das direkte Mittel, um Aktivitäten von einem Teilnehmer zu autorisieren. Bestehende Blockchain-Anwendungen verwenden in der Regel privaten Schlüssel, um die Identität der Teilnehmer zu bestätigen und

Transaktionen zu tätigen. Voraussetzung dafür, dass Informationen nicht verfälscht werden können, ist also die Sicherheit der privaten Schlüssel. Das Problem der Verwaltung von privaten Schlüsseln ist allerdings nicht innerhalb der Blockchainsysteme gelöst [57].

Im Gegensatz zu traditionellen Public-Key-Infrastrukturen sind die Teilnehmer für ihre eigenen privaten Schlüssel verantwortlich, was bedeutet, dass ein privater Schlüssel von den Teilnehmern und nicht von einem Dritten erzeugt und verwaltet wird. Zum einen wird es unmöglich sein, auf ein mit der Blockchain verbundenes digitales Asset zuzugreifen, wenn der zugehörige private Schlüssel verliert ist. Zum anderen ist der Angreifer in der Lage, das gesamte digitale Asset eines Teilnehmers zu stehlen, wenn er den zugehörigen privaten Schlüssel kennt. Die Methodik solcher Angriffe könnte beispielsweise darin bestehen, Malware zu verbreiten und/oder Social Engineering einzusetzen, um die privaten Schlüssel von der Maschine des Teilnehmers zu stehlen [58].

In Kryptowährungssysteme wie z.B. Bitcoin muss ein Teilnehmer ein Wallet auf seinem Desktop oder mobilen Gerät installieren. Das Wallet speichert die privaten und öffentlichen Schlüssel, die mit dem Besitzer des Wallets verbunden sind. Die Wallet-Diebstähle werden hauptsächlich mit Hilfe von Mechanismen durchgeführt, die System-Hacking, die Installation von Buggy-Software und die unsachgemäße Verwendung des Wallets beinhalten [8].

3) Kryptographie: Die Sicherheit der Blockchain-Datenstruktur und Konsensprotokolle wie das PoW basiert unmittelbar auf der zugrunde liegenden kryptographischen Bausteine. Eine breite Anwendung kryptographischer Algorithmen kann dabei unbekannte Backdoors oder Schwachstellen hervorrufen. Ein Risiko der Blockchaintechnologie in Bezug auf die Kryptographie ist, dass sie durch Designfehler des Algorithmus oder den Fortschritt der Rechenleistung anfällig für Kompromittierung ist. Eine solche Kompromittierung kann den Blockchainanwendungen und dem gesamten System schaden [57].

Kryptographische Algorithmen haben die Möglichkeit, innerhalb einer bestimmten Zeitspanne kompromittiert zu werden. Im Allgemeinen haben moderne kryptographische Algorithmen praktische Sicherheit. Im Falle von Public-Key-Kryptosystemen, digitalen Signaturverfahren und kryptographischen Hashfunktionen geht die Sicherheit des Algorithmus verloren, wenn der Angreifer über eine große Rechenleistung verfügt. Die Rechenleistung wächst dabei stetig nach Moore's Gesetz. Weiterhin wird intensiv an der Erforschung und Entwicklung von Quantencomputern gearbeitet. Es ist bekannt, dass Quantencomputer einen Pre-Image-Angriff gegen kryptographische Hashfunktionen durchführen können [59].

Sato und Matsuo [59] geben einen Überblick über die mögliche Bedrohungen des Blockchainsystems, die durch Kompromittierungen bei kryptographischen Algorithmen verursacht werden:

- *Impersonifizierung bei zukünftigen Transaktionen*: Wenn das digitale Signaturverfahren kompromittiert wird, ist der Angreifer inder Lage, den privaten Signierschlüssel aus dem öffentlichen Verifizierungsschlüssel zu bestimmen. Aus der Natur der Blockchain Aus der Natur der Blockchain-Datenstruktur sind die öffentliche Verifizierungsschlüssel des Opfers immer in den zugehörigen auf der Blockchain speicherten Transaktionen sichtbar. Mit Kenntnis des privaten Schlüssels hat der Angreifer die Möglichkeit, sich als Opfer auszugeben und gültige Transaktionen zu erstellen.
- *Modifizierung der Transaktion*: eine Transaktion wird gehasht, bevor sie digital signiert wird. Wenn die Sicherheit der Hashfunktion kompromittiert wird, kann der Angreifer ein zweites Urbild vom Hashwert ermitteln. Die digitale Signatur vom zweiten Urbild ist identisch mit der Signatur originaler Transaktion.
- *Modifizierung des Blocks*: Der Angreifer kann einen anderen Block und eine andere Chain für den vergangenen Zeitraum platzieren, wenn der Angreifer eine Kollision des Block-Hashwertes an einem beliebigen Punkt der Blockchain finden kann.

4) Netzwerk-Infrastruktur-Sicherheit: Das Blockchainsystem verlässt sich auf ein zugrunde liegendes P2P-Netzwerk. Daher hat eine Sicherheitskompromittierung des P2P-Netzwerks einen großen Einfluss auf die Sicherheit der Blockchainanwendungen. Es gibt verschiedene Möglichkeiten für einen Angreifer, um die Schwachstellen des P2P-Netzwerks auszunutzen. Welcher Angriffsvektor gewählt werden soll, ist jedoch abhängig vom Wissen des Angreifers über die Netzwerk-Infrastruktur und den Ressourcen, die ihm zur Verfügung stehen. Im Folgenden werden einige Beispiele für die mögliche Angriffe am Beispiel Bitcoin vorgestellt.

Partitioning-Angriff: Der Angreifer schließt sich dem Bitcoin Peer-to-Peer-Netzwerk mit so vielen Knoten wie möglich an und verwendet diese Knoten, um den Konnektivitätsgraphen zwischen ehrlichen Knoten zu verdünnen. Danach führt er DDoS-Angriffe auf einen ausgewählten ehrlichen Knoten durch, um den Konnektivitätsgraphen in mindestens verschiedene Partitionen aufzuteilen. Kommunikation zwischen den Partitionen ist nicht möglich. Ein solcher Angriff ermöglicht es dem Angreifer nicht direkt, Transaktions- oder Blockchaindaten zu manipulieren. Dennoch beeinträchtigt ein solcher Angriff die Hauptfunktionen von Bitcoin und kann zu einem Vertrauensverlust der ehrlichen Teilnehmer führen [60].

Sybil-Angriff: Der Angreifer schließt sich dem Netzwerk mit Dummy-Knoten an und versucht, einen Teil des Bitcoin-Netzwerks zu kompromittieren. Dies ermöglicht es, den Opfer-Knoten zu isolieren und die von ihm erstellten Transaktionen zu blockieren, oder der Opfer-Knoten wird gezwungen, nur die vom Angreifer bestimmten Blöcke auszuwählen [8].

Eclipse-Angriff: Der Angreifer kontrolliert alle eingehenden und ausgehenden Verbindungen des Opfers und isoliert dadurch das Opfer von den anderen Peers im Netzwerk. Der Angreifer ist dann in der Lage, dem Opfer die Sicht auf den aktuellen Zustand der Blockchain zu beeinträchtigen. Das Opfer wird auch dazu gezwungen, Rechenleistung auf einen veralteten Zustand der Blockchain zu vergeuden. Der

Angreifer kann auch die Rechenleistung des Opfers für seine eigenen bösartigen Zwecke zu nutzen [61].

5) Konsensprotokoll-Sicherheit: Neben den oben genannten allgemeinen Sicherheitsherausforderungen ist das Konsensprotokoll eine weitere Quelle für Sicherheitsproblemen, die speziell für Blockchainsysteme gilt. Der Angreifer kann Strategien verwenden, die die Schwachstellen des Konsensprotokolls adressieren, oder er kann Bugs in der Implementierung des Konsensprotokolls ausnutzen, um eine Vielzahl von Angriffsvektoren durchzuführen. Die Tatsache, dass viele bekannte Protokolle intensiv studiert wurden und ihre Implementierungen Open-Source sind, verschlimmert die Situation. Im Allgemeinen hat jedes Konsensprotokoll seine technischen Einschränkungen, die nur sehr schwer zu überwinden sind. Diese Einschränkungen führen ebenfalls zu negativen Auswirkungen auf die Sicherheit des Blockchainsystems.

a) Angriffe auf das Konsensprotokoll: Um Konsens unter einer sehr hohen Anzahl von Peers zu erreichen, wurden komplexe mathematische Berechnungen, Ressourcen, Vereinbarungen, Verhandlungs- oder Incentive-Ansätze verwendet. Die hohe Komplexität des Systems führt zu einer Vielfalt von Schwachstellen im Entwurf des Konsensprotokolls, die während der Entwurfsphase nicht berücksichtigt oder angemessen angegangen wurden. So gibt es z.B. bei öffentlichen Systeme wie Bitcoin eine Vielzahl von Angriffen auf den Mining-Prozess und die Mining-Pools. Die Angreifer können Angriffsstrategien wie Selfish-Mining oder Feather-Forking verfolgen, um Vorteile gegenüber anderen Peers zu erzielen oder eine oder mehrere Transaktionen auf die Blacklist zu setzen [8]. Das Ausmaß, in dem ein Protokoll für eine bestimmte Bedrohung anfällig ist, variiert zwischen den einzelnen Protokollen. Die Autoren in [62] haben gezeigt, dass dass 6 Blockbestätigungen bei Bitcoin widerstandsfähiger gegen Doppelausgaben sind als 6 Blockbestätigungen bei Ethereum. Und wenn die Bestätigung auf 12 Blöcke steigt, ist Ethereums Douple-Spend-Resistenz nur besser als 6 Blockbestätigungen Bitcoins für einen Angreifer mit weniger als 11% der gesamten PoW-Rechenleistung.

Darüber hinaus kann das Vorhandensein von Bugs in der Implementierung des Konsensprotokolls zu unvorhersehbaren Schwachstellen führen, trotz der Tatsache, dass Methoden und Codebase der Implementierungen durch erfahrende Entwickler überprüft werden [58]. Die Bedrohung verschärft sich noch dadurch, dass die Blockchain-Plattform als zugrundeliegende Technologie von Anwendungen der oberen Schicht die Zusammenarbeit verschiedener Anwendungen und Teilnehmer unterstützt. So können beispielsweise Industriedaten aus den Bereichen Medizin, Finanzen und Kommunikation über eine Blockchain-Plattform erzeugt, gespeichert, aktualisiert und übertragen werden. Der enorme wirtschaftliche Nutzen motiviert Hacker dazu, die Sicherheitslücken der Open-Source-Blockchain-Plattform zu suchen [57]. Andrychowicz et al. [63] zeigten, dass aufgrund eines Bugs im ursprünglichen Konsensprotokoll Bitcoins ein Transaction-Malleability-Angriff durchgeführt werden konnte. Der Angreifer ist dann in der Lage, eine unauthentifizierte Transaktion zu erstellen, die syntaktisch identisch ist (führt die gleiche Aufgabe aus), aber semantisch verschieden (erzeugt unterschiedlichen Hashwert) von einer ursprünglichen Transaktion ist.

b) Technische Einschränkungen der Konsensprotokolle: Das Konsensprotokoll eines öffentlichen Blockchainsystemes basiert auf der Annahme, dass die Mehrheit der Knoten ehrlich ist, das System zu betreiben und zu warten. Wenn die Annahme jedoch verletzt wird, könnte sich ein kooperativer Angriff auslösen lassen. Sobald ein oder mehrere Knoten mehr als 51% der gesamten Stimmgewichte (Rechenleistung bei PoW oder Eigentum von Werteinheiten bei PoS) zu Verfügung hat, können sie sich zusammenschließen, um einen Angriff durchzuführen, um den Inhalt in Blöcken zu manipulieren und störende Angriffe wie DDoS durchzuführen [57]. In Kryptowährungssysteme wie Bitcoin oder Ethereum erlaubt ein erfolgreicher 51%-Angriff es einem Angreifer, bestimmte Transaktionen abzulehnen und bereits ausgegebenen Werteinheiten wiederzuverwenden. Es besteht dem Angreifer die Möglichkeit z.B. mithilfe eines Cloud-Providers, die Rechenleistung, die benötigt wird, um den Konsensprotokoll zu hijacken, günstig genug zu gestalten [58].

Im privaten System gibt es offene Fragen bei der Entwicklung von Konsensprotokollen bzw. Algorithmen zur Erkennung bösartiger Knoten [64]. Vincent Gramoli analysierte unterdessen das Konsensprotokoll der Ethereum-Private-Blockchain, das vom R3-Konsortium[3] getestet wurde. Der Autor stellte fest, dass PoW für privates Blockchainsystem ungeeignet sein kann, wenn Anwendungen die Termination-Eigenschaft des Konsensprotokolls erfordern, d.h. den Punkt zu identifizieren, an dem das Präfix der Blockchain unveränderlich wird. Diese Eigenschaft ist entscheidend, um festzustellen, wann eine Transaktion für eine Anwendung, die den Versand von Waren sicherstellt, eine Tauschplattform, die digitale Werteinheiten in Fiatgeld umwandelt, oder eine Bank, die die Wirksamkeit der Abrechnung überwacht, verpflichtet ist [65].

In Allgemeinen haben private Blockchainsysteme mit einigen Designeinschränkungen zu konfrontieren, die von Marko Vukolić [66] aufgeführt sind:

- *Sequentielle Ausführung*: Transaktionen von Smart-Contracts werden sequentiell ausgeführt. So könnte ein Angreifer Smart-Contracts erstellen, deren Ausführung sehr lange Zeit in Anspruch nimmt, und damit einen DoS-Angriff auf das Blockchainsystem durchführen. Ethereum bewältigt dieses Problem mit der Einführung des Konzepts von *Gas*, was jedoch für viele Geschäftsanwendungen nicht ausreichend ist. Dies liegt daran, dass diese Anwendungen die Vorteile von Distributed-Ledger-Technologien benötigen, ohne dass eine Kryptowährung benötigt wird.
- *Nicht-deterministische Ausführung*: Wenn Smart-Contracts nach dem Konsens ausgeführt werden, muss ihre Ausführung deterministisch sein. Andernfalls kann die Ausführung zu divergierenden Ledgern (Forks)

[3]http://r3cev.com/.

führen. Blockchainsysteme, die für ihren Smart-Contracts eine Universalsprache (engl. *General-Purpose Language*) verwenden, stehen vor einer Herausforderung, da diese Sprachen in Bezug auf Determinismus problematisch sind.

- *Ausführung auf allen Knoten*: Smart-Contracts werden am häufigsten auf allen Knoten ausgeführt. Dies steht im Widerspruch zur Vertraulichkeit, da bei vielen Blockchain-Anwendungsfällen die Logik eines Smart-Contracts oder eines Transaktionsinputs auf bestimmte Knoten beschränkt werden sollte.
- *Flexibilität des Trust-Modells*: Die Vertrauensannahme (engl. *Trust Assumption*) des Konsensprotokolls geht über auf die Ausführung der Smart-Contracts. Allerdings kann diese Vertrauensannahme möglicherweise nicht mit dem Vertrauensmodell übereinstimmen, das ein Smart-Contract-Entwickler für sich in Anspruch nehmen muss.
- *Hard-coded-Konsensprotokoll*: Alle heutige Blockchainsysteme, unabhängig davon, ob sie genehmigungsbasiert oder genehmugiungsfrei sind, besitzen ein hard-coded Konsensprotokoll. Die Änderung des Konsensprotokolls ist sehr schwierig, wenn nicht sogar unmöglich, ohne ernsthafte Code-Umschreibungen. Allerdings kann die Anpassung eines Konsensprotokolls an eine gegebene Blockchainanwendung aufgrund des Vertrauensmodells und der Fehlerannahmen spezifisch für diese Anwendung erforderlich sein.

Zum Schluss wird der Konsens im genehmigungsbasierten System unter der Leitung eines Regulierers umgesetzt. Aus diesem Grund führt unmittelbar jede Exploitation des Regulierers zur noch bedrohlicheren Konsequenz im Vergleich zum genehmigungsfreien System. Alle Probleme, die ein Hijack des Mehrheit-basierte Konsenses erforderlich gemacht hatten, werden nun durch das Hijack einer einzigen Einheit ersetzt [58].

6) Smart-Contract-Sicherheit: Smart-Contract ist ein wesentlicher Baustein der Blockchain-Applikationen, der das Anwendungsspektrum der Technologie dramatisch erweitert. Die Tatsache, dass Smart-Contracts korrekt implementiert werden, ist eine notwendige Voraussetzung für ihre Wirksamkeit. Andernfalls könnte ein Angreifer beispielsweise einen ehrlichen Teilnehmer des Smart-Contracts das Geld stehlen. Allerdings machen zahlreiche Faktoren Smart-Contracts grundsätzlich fehleranfällig und dadurch für die Betroffenen unsicher [52]. Dazu zählen die Offenheit, die Einschränkung der Funktionsfähigkeit, und die Programmiersprache vom Smart-Contract.

a) Schwachstellen bzgl. der Offenheit : In [9] führen Loi Luu et al. aus, dass die Offenheit des Smart-Contracts zu Sicherheitslücken führen würde. Im Gegensatz zu herkömmlichen verteilten Anwendungsplattformen bietet ein öffentliches Blockchainsystem wie Ethereum eine Smart-Contract-Plattform, die in einem offenen Netzwerk arbeitet, an dem jeder teilnehmen kann. So ist die Ausführung von Smart-Contracts anfällig für Manipulationsversuche durch beliebige Angreifer. Diese Bedrohung beschränkt sich jedoch in traditionellen genehmigungsbasierten Netzwerken, wie zum Beipiel in zentralisierten Cloud-Services, nur auf unfallbedingten Fehler. Obwohl die Teilnehmer des Blockchainsystems ein vordefiniertes Protokoll befolgen müssen, besteht noch ein erheblicher Spielraum für Manipulationen bei der Ausführung eines Smart-Contracts durch die Teilnehmer. Hierbei können die Schwachstellen von Ethereum-Smart-Contracts auf der Blockchain-Ebene (vgl. Tabelle III) als Beispiele genannt werden. In Ethereum können die Miners entscheiden, welche Transaktionen sie annehmen, wie sie Transaktionen sortieren, den Block-Zeitstempel setzen, etc. Smart-Contracts, die von einer dieser Quellen abhängen, können daher von böswilligen Miners manipuliert werden.

b) Schwachstellen bzgl. der Einschränkungen von Smart-Contracts: Es gibt einige Einschränkungen in den Fähigkeiten von Smart-Contract. Dies liegt an der nachträglichen Unveränderlichkeit der Blockchain und den Einschränkungen der virtuellen Maschine, die den Byte-Code des Contracts ausführt. Die Angreifer können diese Schwachstellen ausnutzen, um verschiedene Angriffe auszuführen. z.B. In Ethereum ist der Call-Stack auf 1024 Frames begrenzt. Ein Angreifer kann den Stack zum Überlaufen bringen, um bestimmte Funktionen (wie z.B. Geld verschicken) eines Contracts zu verhindern und daraus zu profitieren. Um es noch schlimmer zu machen, aufgrund der nachträglichen Unveränderlichkeit der Blockchain lassen sich die Bugs in einem Contract nicht erheben, was ebenfalls vom Angreifer ausgenutzt werden könnten. Diese Schachstellen entsprechen die auf der EVM-Ebene von Ethereum [67].

c) Schwachstellen bzgl. der Programmiersprache: Der letzte Faktor hängt mit den Schwachstellen in der High-Level-Programmiersprache des Smart-Contracts zusammen. Viele Sicherheitslücken entstehen durch eine Fehlausrichtung zwischen der Semantik der Programmiersprache und der Eingebung der Programmierer des Smart-Contracts. Solidity - die Smart-Contract-Programmiersprache von Ethereum ist ein Beispiel dafür. Es sieht aus wie eine typisierte Javascript-ähnliche Sprache, implementiert aber einige der Features auf eine eigentümliche Art und Weise. Gleichzeitig führt die Sprache keine Konstrukte ein, die sich mit domänenspezifischen Aspekten befassen. Dies führt dazu, dass Programmierer, ob erfahren oder nicht, fehlerbehaftete Smart-VerträgeContracts erstellen könnten, die anfällig für Angriffe sind. Diese Schachstellen entsprechen die auf der Solidity-Ebene von Ethereum [67].

B. Gegenmaßnahmen

Im Laufe der Jahre wurde eine breite Palette von Gegenmaßnahmen vorgeschlagen, die sich mit den im vorhergehenden Abschnitt erwähnten Sicherheitsproblemen befassen. Dieser Abschnitt gibt einen strukturellen Überblick über diese Vorschläge. Es gibt insgesamt fünf Gruppen von Gegenmaßnahmen, die auf die im letzten Abschnitt behandelten allgemeinen und spezifischen Sicherheitsfragen abzielen. Nach unserem besten Wissen gibt es noch keine Vorschläge zur Standardisierung der Applikationslogik-Schicht.

a) Sichere Aufbewahrung von Schlüsseln: In Kryptowährungssysteme wie Bitcoin werden die Schlüssel der Teilneh-

Tabelle III: Schwachstellen von Smart-Contracts am Beispiel Ethereum [67]

Ebene	*Schwachstelle*	*Bemerkung*
Solidity	Call to the unknown	Einige der in Solidity verwendeten Primitive haben den Nebeneffekt, dass sie eine Fallback-Funktion aufrufen.
	Gasless send	Die Ausführung des Contracts geht dem Gas aus. Eine Exception wird geworfen. Dies wird vom Angreifer zusammen mit der Exception-Disorders-Schwachstelle ausgenutzt.
	Exception disorders	Solidity ist nicht einheitlich in der Art und Weise, wie es mit Exceptions umgeht.
	Type casts	In Solidity kann ein Contract typgeprüft werden. Es werden jedoch nicht alle Typenfehler entdeckt.
	Reentrancy	Fallback-Funktion ermöglicht es einen Angreifer, die Aufruffunktion aufzurufen und damit deren Termination zu verhindern.
	Keeping secrets	Die Deklaration eines Feldes als privat garantiert nicht seine Geheimhaltung aufgrund der Offenheit von Blockchainndaten.
EVM	Immutable bugs	Bugs im Smart-Contract lassen sich aufgrund der nachträglichen Unveränderlichkeit der Blockchain nicht erheben.
	Ether lost in trasfer	Wenn ein Werteinheit an eine Orphan-Adresse geschickt wird, ist sie für immer verloren.
	Stack size limit	Der Call-Stack ist auf 1024 Frames begrenzt. Ein Angreifer kann den Stack zum Überlaufen bringen, um bestimmte Funktionen (wie zum Beispiel des Versenden von Geld) eines Contracts zu verhindern
Blockchain	Unpredictable state	Böswilliger Miner könnte den Zustand, in dem ein Smart-Contract ausgeführt wird, beeinflussen.
	Generating randomness	Böswilliger Miner könnte versuchen, seinen Block so zu gestalten, dass er das Ergebnis des Pseudozufallsgenerators beeinflussen kann.
	Time constraints	Böswilliger Miner kann den Block-Timestamp anpassen.

mer in den so genannten Wallets aufbewahrt. Verschiedene Arten von Wallet-Implementierungen werden erforscht, einschließlich Software-, Hardware-, Hosted-, Papier- und Brain-Wallets [8]. In [68] wurde das Konzept des Cold-Wallets vorgeschlagen. Bei dieser Methode werden zwei Computer verwendet, von denen eines vom Internet getrennt werden muss. Ein neuer geheime Schlüssel wird generiert, an den die Werteinheiten vom Teilnehmer gesendet werden. Die Autoren behaupten dass, wenn der Computer nicht mit dem Internet verbunden ist, haben die Hacker keine Möglichkeit, die Schlüssel zu kennen, und damit die Sicherheit des Wallets gewährleistet wird.

Darüber hinaus werden auch kryptographische Techniken zum Einsatz gebracht. Zum Beispiel bietet das Online-Wallet-Service *BitGo* 2-of-3 Multi-Signatur-Transaktionen. Die Sicherheit der Schlüssel wird damit erhöht, jedoch auf Kosten der Privatsphäre [8]. In [69] schlagen die Autoren einen Threshold-Digitale-Signatur-Verfahren für die Sicherung von Bitcoin-Schlüsseln vor. Der private Schlüssel wird in Anteile (engl. *Shares*) aufgeteilt. Jede Teilmenge der Anteile, die gleich oder größer als ein vordefinierter Schwellenwert ist, kann den privaten Schlüssel rekonstruieren. Im Gegensatz dazu erhält jede Teilmenge, die kleiner ist, keine Informationen über den Schlüssel. Daher wird der Schlüssel nie offengelegt, da die Teilnehmer, die direkt eine Signatur erstellen, nichts über den privaten Schlüssel wissen.

b) Widerstandsfähigkeit gegen Kompromittierungen bei der zugrunde liegenden Kryptographie: Sato und Matsuo [59] zeigen die Möglichkeit, die Gültigkeit des Blockchainsystems durch die Anwendung vom Long-Term-Signatursverfahren zu verlängern. Das bestehende Verfahren setzt die Existenz eines von vertrauenswürdigen Dritten ausgestellten Timestamp-Tokens voraus. Das Vertrauen zu Dritten widerspricht jedoch dem Ziel eines öffentlichen Blockchainsystems. Die Autoren schlagen dabei eine Lösung vor, die das Long-Term-Signatursverfahren für den Fall vom öffentlichen Blockchainsystem anpasst. Das vorgeschlagene Verfahren vermeidet die Hard-Fork der ursprünglichen Blockchain im Falle einer Kompromittierung der Hashfunktion, muss aber im Falle einer Kompromittierung des digitalen Signatursverfahrens ein Smooth-Fork bereitstellen.

c) Sichern der Netzwerk-Infrastruktur: Eine mögliche Möglichkeit, sich gegen DDoS-Angriffe zu schützen, besteht darin, Botnets zu erkennen, die vom Angreifer verwendet werden, um diesen Teil bis zum Debuggen aus dem Netzwerk zu isolieren [8]. Der Autoren in [70] schlagen eine Technik zur Erkennung von Botnets vor, die einen maschinell lernenden Klassifikator zur Klassifizierung von Domainnamen verwendet. Danach werden die Netzwerkkommunikationen zu ähnlichen Mustern gruppiert. Schließlich wird ein graphenbasiertes Framework aufgebaut, das die Erkennung des bösartigen Teils des Netzwerks ermöglicht. Andere mögliche Methoden, um sich vor DDoS-Angriffen zu schützen, sind beispielsweise die Konfiguration des Netzwerks so, dass bösartige Pakete blockiert werden, oder die Implementierung eines DoS-Schutzsystems eines Drittanbieters [8].

Eine Überprüfung der eingehenden und ausgehenden Verbindungen eines Knotens kann den Effekt eines Eclipse-Angriffs reduzieren. Die Autoren in [61] schlugen eine Reihe von Gegenmaßnahmen vor, um zu verhindern, dass der Bitcoin-Teilnehmer von Angreifern isoliert wird. Die Teilnehmer können über ein *Intrusion Detection System* verfügen, um

das Fehlverhalten der Knoten im Netzwerk zu überwachen. Die Adressen, die sich im Netzwerk fehlerhaft verhalten, werden von Verbindungen ausgeschlossen.

d) Sichern des Konsensprotokolls: Die Leistungsfähigkeit und der Trade-off gegen die Sicherheitsgarantie öffentlicher Blockchainsysteme ist relativ gut untersucht. Gervais et al. in [62] stellten einen quantitativen Framework zur Analyse der Sicherheits- und Leistungsimplikationen verschiedener Konsensus- und Netzwerkparameter von PoW-Blockchainsystemen vor. Auf der Grundlage dieses Frameworks entwickeln die Autoren optimale Angriff-Strategien für Double-Spend und Selfish-Mining unter Berücksichtigung realer Coinstraints, wie z.B. die Auswirkungen von Eclipse-Angriffen. Bonneau et al. [71] listen eine Reihe von Vorschlägen für alternative Konsensprotokolle, die das Nakamotos PoW ersetzen können.

In [72] entwickelten Dinh et al. *BLOCKBENCH* - ein Evaluationsframework zur Analyse privater Blockchainsysteme. Es ermöglicht einen fairen Vergleich von Plattformen und ein tieferes Verständnis der verschiedenen Systemdesigns. *BLOCKBENCH* misst hierbei die Gesamt- und Komponentenperformance in Bezug auf u.a. Fehlertoleranz. In [66] wird gezeigt, dass das *Hyper Ledger Fabric* - ein modulares, universelles, genehmigungsbasiertes Blockchainsystem, in der Lage ist, die Designeinschränkungen eines solchen Systems zu lösen. Die parallele Ausführung von Smart-Contracts ist möglich und nicht jeder Peer führt alle Transaktionen aus. Es wird validiert, dass Transaktion-Updates tatsächlich von einer durch ein Endorsement-Policy festgelegten Anzahl von Peers bestätigt werden. Dies eliminiert die Effekte von Nicht-Determinismus und ermöglicht eine flexible Trennung der Vertrauensannahmen für Smart-Contracts von den Vertrauensannahmen des Konsenses. Schließlich ist Hyper Ledger Fabric modular aufgebaut in Bezug auf das verwendete Konsensptokoll.

e) Sichern des Smart-Contracts: Mehrere Vorschläge zur formalen Verifikation von Smart-Contracts wurde vorgestellt [56], [73], [74]. In [73] wurde die Ethereum Virtuelle Maschine für verschiedene interaktive Theorem-Provers wie z.B.Isabelle/HOL modelliert. Dieses Modell wurde dann verwendet, um Invarianten und Sicherheitseigenschaften von Ethereum Smart-Contracts nachzuweisen. In [74] erweiterten Amani et al. das Modell, um auch die Korrektheitseigenschaften des Smart-Contracts abzudecken, und eine Behandlung der Termination auf der Basis von Ethereums Konzept des Execution-Gases zu ermöglichen.

Eine weitere Möglichkeit ist die Entwicklung neuer Programmiersprachen, um Programmierern die Implementierung korrekter Smart-Contracts zu erleichtern. Michael Coblenz [75] schlug eine objektorientierte Sprache namens Obsidian vor. States sind in dieser Sprache first-class und daher hängen Methoden, die von einem Objekt aufgerufen werden können, vom aktuellen Zustand des Objekts ab. Der Autor behauptet, dass Programmierer Obsidian effektiv nutzen können, um Programme mit wenig Training zu schreiben, und dass es wahrscheinlicher ist, korrekten und sicheren Code zu schreiben als mit Solidity.

VI. Fazit

Wir haben gesehen, wie die Blockchain und die neue Generation von Konsensprotokollen die Distributed-Ledger-Technologie innovieren. Einerseits ermöglicht die Eigenschaft von nachträglichen Unveränderlichkeit der Blockchain eine Vielzahl neuer Anwendungen. Andererseits wirft es aber auch Fragen bezüglich der Privatsphäre der Teilnehmer, sowohl technischer als auch rechtlicher Art, und der Sicherheit des Systems auf.

Verschiedene Schichten des Systems haben unterschiedliche Schwachstellen und müssen sich unterschiedlichen Bedrohungen stellen. Die Ursache könnte in den Einschränkungen des Systems selbst liegen, wie z.B. in der Offenheit, der intrinsischen nachträglichen Unveränderlichkeit der Blockchain, der Abhängigkeit von der zugrunde liegenden P2P-Infrastruktur und Kryptographie. Darüber hinaus könnten spezifisches Systemdesigns und -implementierungen auch eine Ursache für Schwachstellen und Datenschutzprobleme sein.

Wir haben auch die Ähnlichkeiten und die Unterschiede der verschiedenen Arten von Blockchainsystemen in Bezug auf ihre Struktur sowie ihre Datenschutz- und Sicherheitsaspekte gesehen. Im Laufe der Jahre hat es eine Vielzahl von Gegenmaßnahmen gegeben, um die Privatsphäre zu verbessern und diese Systeme sicherer zu machen. Es existiert ohnehin noch viel Raum für weitere Forschungen und Entwicklungen.

Literatur

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf," 2009.

[2] L. Xu, L. Chen, N. Shah, Z. Gao, Y. Lu, and W. Shi, "Dl-bac: Distributed ledger based access control for web applications," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 1445–1450. [Online]. Available: https://doi.org/10.1145/3041021.3053897

[3] I. Eyal, "Blockchain technology: Transforming libertarian cryptocurrency dreams to finance and banking realities," *Computer*, vol. 50, no. 9, pp. 38–49, 2017.

[4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theor.*, vol. 22, no. 6, pp. 644–654, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.1976.1055638

[5] M. A. Bishop, *The Art and Science of Computer Security*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[6] J. Buchmann, *Einführung in die Kryptographie.* Springer, Berlin, 2003.

[7] B. Schneier, *Applied Cryptography (2Nd Ed.): Protocols, Algorithms, and Source Code in C*. New York, NY, USA: John Wiley & Sons, Inc., 1995.

[8] M. Conti, S. K. E, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *CoRR*, vol. abs/1706.00916, 2017.

[9] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 254–269. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978309

[10] R. Böhme and P. Pesch, "Technische grundlagen und datenschutzrechtliche fragen der blockchain-technologie," *Datenschutz und Datensicherheit - DuD*, vol. 41, no. 8, pp. 473–481, Aug 2017. [Online]. Available: https://doi.org/10.1007/s11623-017-0815-y

[11] C. Cachin and M. Vukolic, "Blockchain consensus protocols in the wild," *CoRR*, vol. abs/1707.01873, 2017.

[12] A. Baliga, "Understanding blockchain consensus models," 2017.

[13] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186. [Online]. Available: http://dl.acm.org/citation.cfm?id=296806.296824

[14] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money," *CoRR*, vol. abs/1511.05740, 2015. [Online]. Available: http://arxiv.org/abs/1511.05740

[15] T. Swanson, "Great chain of numbers: A guide to smart contracts, smart property and trustless asset management," http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf, Apr 2014, abgerufen am 03.10.2017. [Online]. Available: http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf

[16] K. Wüst and A. Gervais, "Do you need a blockchain?" *IACR Cryptology ePrint Archive*, vol. 2017, p. 375, 2017.

[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger eip-150 revision (759dccd - 2017-08-07)," 2017, abgerufen am 03.01.2018. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[18] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin." in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2014, pp. 459–474. [Online]. Available: http://dblp.uni-trier.de/db/conf/sp/sp2014.html#Ben-SassonCG0MTV14

[19] M. Walport, "Distributed ledger technology: beyond block chain," 2015, abgerufen am 03.06.2016. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf

[20] J. Mattila, "The blockchain phenomenon – the disruptive potential of distributed consensus architectures," 05 2016.

[21] T. Swanson, "Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems," http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf, 2015, abgerufen am 04.02.2018.

[22] N. U. G. F. Vincent Schlatt, André Schweizer, "Blockchain: Grundlagen, anwendungen und potenziale," 2016, abgerufen am 31.01.2018. [Online]. Available: http://publica.fraunhofer.de/documents/N-452387.html

[23] D. Chaum, "Blind signatures for untraceable payments," pp. 199–203, 01 1982.

[24] L. Law, S. Sabett, and J. Solinas, "How to make a mint: the cryptography of anonymous electronic cash," vol. 46. HeinOnline, 1996, p. 1131. [Online]. Available: http://www.aulawreview.org/pdfs/46/46-4/law.pdf

[25] 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.

[26] A. W. Baur, J. Bühler, M. Bick, and C. S. Bonorden, "Cryptocurrencies as a Disruption? Empirical Findings on User Adoption and Future Potential of Bitcoin and Co," in *14th Conference on e-Business, e-Services and e-Society (I3E)*, ser. Open and Big Data Management and Innovation, M. Janssen, M. Mäntymäki, J. Hidders, B. Klievink, W. Lamersdorf, B. van Loenen, and A. Zuiderwijk, Eds., vol. LNCS-9373, Delft, Netherlands, Oct. 2015, pp. 63–80, part 2: Adoption. [Online]. Available: https://hal.inria.fr/hal-01448070

[27] M. Kõlvart, M. Poola, and A. Rull, *Smart Contracts*, T. Kerikmäe and A. Rull, Eds. Cham: Springer International Publishing, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-26896-5_7

[28] T. D, A. J, V. I, C. N, P. M. L, U. P, and S. J, "Smart contracts: the ultimate automation of trust?" 2015, abgerufen am 06.04.2018. [Online]. Available: https://www.bbvaresearch.com/wp-content/uploads/2015/10/Digital_Economy_Outlook_Oct15_Cap1.pdf

[29] D. S, van Doorn M, van Manen Tand Bloem J, and van Ommeren E, "Design to disrupt. blockchain: cryptoplatform for a frictionless economy," 2015, abgerufen am 04.02.2018. [Online]. Available: http://www.the-blockchain.com/docs/Blockchain%20Cryptoplatform%20for%20a%20Frictionless%20%20Economy.pdf

[30] P. Forte, D. Romano, and G. Schmid, "Beyond bitcoin - part I: A critical look at blockchain-based systems," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1164, 2015.

[31] G. W. Peters, E. Panayi, and A. Chapelle, "Trends in crypto-currencies and blockchain technologies: A monetary theory and regulation perspective," *CoRR*, vol. abs/1508.04364, 2015.

[32] W. Mougayar, "The crypto-technology and bitcoin landscape," 2015, abgerufen am 04.02.2018. [Online]. Available: http://bitcoin.xyz/crypto-technology-bitcoin-landscape/

[33] G. Zyskind, O. Nathan, and A. . Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.

[34] R. AlTawy, M. ElSheikh, A. M. Youssef, and G. Gong, "Lelantos: A blockchain-based anonymous physical delivery system," *IACR Cryptology ePrint Archive*, vol. 2017, p. 465, 2017.

[35] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 3rd ed. Prentice Hall Professional Technical Reference, 2002.

[36] F. Armknecht, G. Karame, A. Mandal, F. Youssef, and E. Zenner, "Ripple: Overview and outlook," 08 2015.

[37] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," vol. 3, 07 2011.

[38] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 6–24.

[39] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Financial Cryptography and Data Security*, N. Christin and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 469–485.

[40] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 34–51.

[41] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin P2P network," *CoRR*, vol. abs/1405.7418, 2014.

[42] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," *CoRR*, vol. abs/1410.6079, 2014.

[43] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 127–140. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504747

[44] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *CoRR*, vol. abs/1502.01657, 2015.

[45] T. Hardjono, "Verifiable anonymous identities and access control in permissioned blockchains," 2016.

[46] H. Halpin and M. Piekarska, "Introduction to security and privacy on the blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, April 2017, pp. 1–3.

[47] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," 2013, abgerufen am 18.02.2018. [Online]. Available: https://bitcointalk.org/index.php?topic=279249.0

[48] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security*, N. Christin and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 486–504.

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

[50] 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.

[51] S. B. Venkatakrishnan, G. C. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *CoRR*, vol. abs/1701.04439, 2017.

[52] J. Moubarak, E. Filiol, and M. Chamoun, "Comparative analysis of blockchain technologies and tor network: Two faces of the same reality?" in *2017 1st Cyber Security in Networking Conference (CSNet)*, Oct 2017, pp. 1–9.

[53] I. G. Richard Gendal Brown, James Carlyle and M. Hearn, "Corda: An introduction. r3 cev," August 2016.

[54] W. Li, A. Sforzin, S. Fedorov, and G. O. Karame, "Towards scalable and private industrial blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, ser. BCC '17. New York, NY, USA: ACM, 2017, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/3055518.3055531

[55] E. Brickell and J. Li, "Enhanced privacy id: A direct anonymous attestation scheme with enhanced revocation capabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 345–360, May 2012.
[56] S. Matsuo, "How formal analysis and verification add security to blockchain-based systems," in *2017 Formal Methods in Computer Aided Design (FMCAD)*, Oct 2017, pp. 1–4.
[57] F. Dai, Y. Shi, N. Meng, L. Wei, and Z. Ye, "From bitcoin to cybersecurity: A comparative study of blockchain application and security issues," pp. 975–979, 11 2017.
[58] W. K. Hon, "Distributed ledger technology & cybersecurity," 2016.
[59] M. Sato and S. Matsuo, "Long-term public blockchain: Resilience against compromise of underlying cryptography," *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 42–49, 2017.
[60] T. Neudecker, P. Andelfinger, and H. Hartenstein, "A simulation model for analysis of attacks on the bitcoin peer-to-peer network," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1327–1332.
[61] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman
[62] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 3–16. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978341
[63] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "On the malleability of bitcoin transactions," in *Financial Cryptography and Data Security*, M. Brenner, N. Christin, B. Johnson, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1–18.
[64] W. T. Tsai, X. Bai, and L. Yu, "Design issues in permissioned blockchains for trusted computing," in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, April 2017, pp. 153–159.
[65] V. Gramoli, "On the danger of private blockchains," 2016.
[66] M. Vukolić, "Rethinking permissioned blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, ser. BCC '17. New York, NY, USA: ACM, 2017, pp. 3–7. [Online]. Available: http://doi.acm.org/10.1145/3055518.3055526
[67] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. New York, NY, USA: Springer-Verlag New York, Inc., 2017, pp. 164–186. [Online]. Available: https://doi.org/10.1007/978-3-662-54455-6_8
[68] M. Draupnir, "Bitcoin cold storage guide," 2016, abgerufen am 27.02.2018. [Online]. Available: https://www.weusecoins.com/bitcoin-cold-storage-guide/
[69] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security," in *Applied Cryptography and Network Security*, M. Manulis, A.-R. Sadeghi, and S. Schneider, Eds. Cham: Springer International Publishing, 2016, pp. 156–174.
[70] P. Camelo, J. Moura, and L. Krippahl, "CONDENSER: A graph-based approachfor detecting botnets," *CoRR*, vol. abs/1410.8747, 2014.
[71] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 104–121.
[72] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*, ser. SIGMOD '17. New York, NY, USA: ACM, 2017, pp. 1085–1100. [Online]. Available: http://doi.acm.org/10.1145/3035918.3064033
[73] Y. Hirai, "Defining the ethereum virtual machine for interactive theorem provers," in *Financial Cryptography and Data Security*, M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore, and M. Jakobsson, Eds. Cham: Springer International Publishing, 2017, pp. 520–535.
[74] S. Amani, M. Bégel, M. Bortin, and M. Staples, "Towards verifying ethereum smart contract bytecode in isabelle/hol," in *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, ser. CPP 2018. New York, NY, USA: ACM, 2018, pp. 66–77. [Online]. Available: http://doi.acm.org/10.1145/3167084
[75] M. Coblenz, "Obsidian: A safer blockchain programming language," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, May 2017, pp. 97–99.

ANHANG

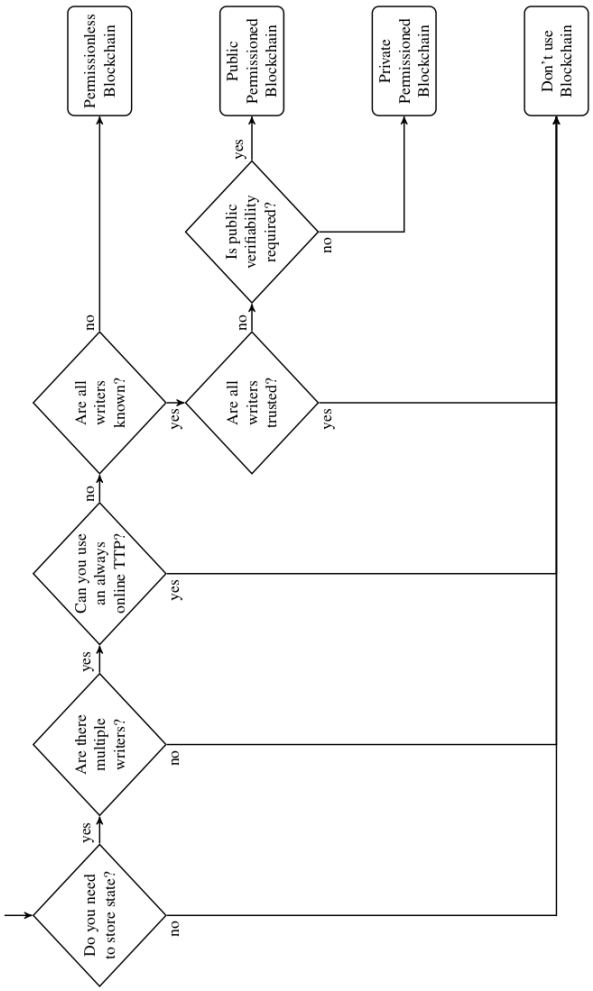


Abbildung 9: Richtlinie für die Feststellung, ob eine Blockchain die geeignete technische Lösung eines Problems ist [16].

Tabelle IV: Techniken zur Verbesserung von Privatsphäre im öffentlichen Blockchainsystem am Beispiel Bitcoin [8]

Vorschlag	***Herangehensweise***	***Merkmale und Eigenschaften***	***Vorteile***	***Nachteile***
CoinJoin	P2P	verwendet Transaktionen mit Multi-Signatur,um die Privatsphäre zu verbessern.	Verhinderung von Diebstählen, niedrigere Transaktionsgebühren glaubhafte Abstreitbarkeit	Die Anonymität ist abhängig von der Anzahl der Teilnehmer, anfällig für DoS-, Sybil- und Intersection-Angriffe, verhindert eine glaubhafte Abstreitbarkeit.
CoinShuffle	P2P	dezentrales Protokoll zur Koordination von CoinJoin-Transaktionen durch ein kryptographisches Mixing-Protokoll	interne Unverkettung, robust gegen DoS-Angriffe, Diebstahlschutz	niedrigere Anonymität und Abstreitbarkeit, anfällig für Intersection- und Sybil-Angriffe
Xim	P2P	anonymes Partnering und Multi-Round-Mixing	verteiltes Pairing, interne Unverkettung, Verhinderung von Sybil- und DoS-Angriffe	höhere Mixing-Zeit
CoinShuffle++DiceMix	P2P	basiert auf CoinJoin, optimale P2P-Mixing zur Verbesserung der Anonymität in Kryptowährungen	Geringe Mixing-Zeit (8 Sek. für 50 Peers), resistent gegen Deanonymisierungsangriffe, gewährleistet Anonymität und Terminierung des Absenders	anfällig für DoS- und Sybil-Angriffe, eingeschränkte Skalierbarkeit, keine Unterstützung für vertrauliche Transaktionen (engl. *Conficential Transaction*)
ValueShuffle	P2P	basiert auf CoinShuffle++, Mixing von CTs wird verwendet, um eine umfassende Transaktionsprivatsphäre zu erreichen.	Unverkettbarkeit, CT-Kompatibilität und Diebstahlschutz, normale Bezahlung mit ValueShuffle erfordert nur eine Transaktion.	anfällig für DoS- und Sybil-Angriffe, eingeschränkte Skalierbarkeit
Dandelion	P2P	Networking-Policies zur Verhinderung der Netzwerk-Deanonymisierung von Bitcoin-Benutzer	starke Anonymität auch in Anwesenheit mehrerer Angreifer	anfällig für DoS- und Sybil-Angriffe
SecureCoin	P2P	basiert auf CoinParty, einem effizienten und sicheren Protokoll für anonyme und nicht verknüpfbare Bitcoin-Transaktionen	Schutz vor Sabotageangriffen mit einer beliebigen Anzahl von Saboteuren, geringe Mixing-Gebühr, Abstreitbarkeit	anfällig für DoS-Angriffe, eingeschränkte Skalierbarkeit
CoinParty	partiell P2P	basiert auf CoinJoin, verwendet Threshold ECDSA und Entschlüsselung-Mixnetze, um die Vorteile von zentralen und dezentralen Mixen in einem einzigen System zu vereinen.	Verbesserung von Robustheit, Anonymität, Skalierbarkeit und Abstreitbarkeit, keine Mixing-Gebühr	partiell anfällig für Münzdiebstahl und DoS-Angriffe, hohe Mixing-Zeit, erfordert separate, ehrliche Mixing-Peers
MixCoin	Verteilt	Dritter-Mixing mit Accountability	DoS- und Sybil-Resistenz	partiell interne Unverkettung und Diebstahlschutz
BlindCoin	Verteilt	basiert auf MixCoin, verwendet blinde Signatur zur Sicherstellung der Anonymität	interne Unverkettung, DoS- und Sybil-Resistenz	partielle Diebstahlresistenz. Zusätzliche Kosten und Verzögerungen im Mixing-Prozess.
TumbleBit	Verteilt	ungerichteter, unverkettbarer Payment-Hub, der eine nicht vertrauenswürdige intermediäre Partei verwendet	verhindert Diebstahl, anonym, widersteht Intersection, Sybil- und DoS-Angriffe, skalierbar	Normale Zahlung mit TumbleBit erfordert mindestens zwei aufeinander folgende Transaktionen
ZeroCoin / Zero-Cash	Altcoin	eine kryptographische Erweiterung zu Bitcoin, unverkettbare und nicht nachvollziehbare Transaktionen unter Verwendung von ZKP	interne Unverkettbarkeit, Diebstahl und DoS-Resistenz	erfordert ein vertrauenswürdiges Setup und Annahmen von nicht-falsifizierbaren Kryptographie, Blockchain Pruning ist nicht möglich.
CryptoNote	Altcoin	verlässt sich auf Ringsignaturen, um Anonymität zu gewährleisten	starke Privatsphären- und Anonymitätsgarantien	Höhere Rechenkomplexität, nicht kompatibel mit Pruning
MimbleWimble	Altcoin	Kryptowährung mit CT	CT-Kompatibilität, Verbesserung von Privatsphäre, Fungibilität und Skalierbarkeit	anfällig für DoS-Angriffe, nicht kompatibel mit Smart-Contracts
ByzCoin	Altcoin	Bitcoin-ähnliche Kryptowährung mit hoher Konsistenz durch Collective Signing	Geringere Konsensus-Latenz und hoher Transaktionsdurchsatz, Resistenz gegen Selfish- und Stubborn-Mining, Eclipse und Delivery- Manipulationen und Double Spending Angriffe	anfällig für langsame oder temporäre DoS-Angriffe und 51% Angriffe