Bundesamt
für Sicherheit in der
Informationstechnik

Blockchain sicher gestalten

Konzepte, Anforderungen, Bewertungen

Grußwort

Im Bereich der Informationstechnik gehört Blockchain gegenwärtig zu den am häufigsten diskutierten Themen. Diese Technologie zur verteilten Datenhaltung nahm ihren Ursprung mit der Kryptowährung Bitcoin, die insbesondere durch ihre Kurshöhenflüge im Jahr 2017 berühmt wurde. Ausgehend von ihrem Versprechen, durch eine dezentrale Struktur die Manipulation von Daten rein technisch zu verhindern, größtmögliche Transparenz zu bieten und Intermediäre in Geschäftsprozessen zu ersetzen, wurden in den letzten Jahren viele Ideen zur Anwendung der Blockchain-Technologie in ganz unterschiedlichen Gebieten entwickelt.

Im politischen Diskurs wurde die Blockchain-Technologie ebenfalls verstärkt aufgegriffen. So findet sich der Begriff Blockchain mehrfach im Koalitionsvertrag der 19. Legislaturperiode des Deutschen Bundestages von 2018 wieder und die Bundesregierung hat sich die Entwicklung einer umfassenden Blockchain-Strategie bis zum Sommer 2019 zum Ziel gesetzt.

Wie bei vielen Themen mit hoher medialer Aufmerksamkeit ist es wichtig, bei den Diskussionen um Blockchain den Bezug zu den technischen Grundlagen zu wahren. Dies gilt insbesondere für den Aspekt der IT-Sicherheit, da durch die Nutzung von Blockchain häufig ein Sicherheitsgewinn erhofft wird. Zahlreiche Sicherheitsvorfälle mit Schäden in Millionenhöhe zeigen jedoch, dass Maßnahmen zur Herstellung von IT-Sicherheit auch durch Nutzung von Blockchain nicht obsolet werden.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) hat im Februar 2018 mit der Veröffentlichung seines Eckpunktepapiers einen ersten Schritt unternommen, um auf die grundsätzlichen Fragestellungen in diesem Bereich hinzuweisen. In vielfältigen Gesprächen mit verschiedenen Akteuren, die Blockchain-Lösungen anbieten oder über ihren Einsatz nachdenken, ermittelte das BSI den Bedarf nach einer tiefergehenden Analyse der Sicherheitseigenschaften.

In der Analyse auf den nachfolgenden Seiten wird die Blockchain-Technologie daher detailliert dargestellt und die aus Sicht der IT-Sicherheit relevanten Punkte werden ausführlich untersucht. Dabei wird auch analysiert, in welchem Maße die Blockchain-Technologie die mit ihr verbundenen Sicherheitserwartungen zu erfüllen vermag und wie sie sich im aktuellen Rechtsrahmen darstellt.

Dieses Dokument unterstützt somit Entwickler und potenzielle Nutzer von Blockchain-Lösungen dabei, Chancen und Risiken fundiert zu bewerten und IT-Sicherheit von Anfang an zu berücksichtigen. Die dynamische Weiterentwicklung der Blockchain-Technologie eröffnet ebenso die Möglichkeit, die Ergebnisse der Analysen als Basis für zukünftige Diskussionen auf nationaler sowie internationaler Ebene zu verwenden. Denn auch beim Thema Blockchain möchte das BSI seinem Auftrag nachkommen und die Informationssicherheit für Staat, Wirtschaft und Gesellschaft gestalten.

Ich wünsche Ihnen eine aufschlussreiche Lektüre.

Arne Schönbohm, Präsident des Bundesamts für Sicherheit in der Informationstechnik (BSI)

Arne Schönbohm

Zusammenfassung

Blockchain ist eine neue Technologie zur Datenhaltung, bei der durch die verteilte Datenspeicherung in Verbindung mit kryptografischen Verfahren und weiteren technischen Maßnahmen die Abhängigkeit von einer zentralen Stelle weitestgehend eliminiert wird. Hierdurch lassen sich einerseits gezielte Manipulationen deutlich erschweren. Andererseits wurde dieses Merkmal von Blockchains zum Anlass genommen, ihre Nutzung in verschiedenen Anwendungsfällen vorzuschlagen, bei denen mehrere Parteien mit unterschiedlichen Interessen involviert sind und sich nicht auf eine zentrale Stelle einigen können, die die Anwendung kontrolliert. Ebenso wurden auch Anwendungsfälle vorgeschlagen, in denen zwar eine zentrale Stelle existiert, man sich jedoch von einer unmittelbaren Interaktion der Parteien Effizienzgewinne verspricht.

Aufgrund der Erwartungen, die an die Blockchain-Technologie gestellt werden, ist sie momentan zu einem Trendthema geworden, mit dem sich eine Vielzahl an Akteuren aus Forschung, Wirtschaft und Verwaltung intensiv beschäftigt. Die Verwendung der Technologie wird in vielen Bereichen diskutiert und untersucht. Heute werden Blockchains aber nur in der Finanzbranche, insbesondere im Bereich der Kryptowährungen, in vergleichsweise großem Ausmaß praktisch eingesetzt.

Das BSI untersucht das Thema Blockchain in diesem Dokument vor allem unter dem Gesichtspunkt der IT-Sicherheit, betrachtet aber auch weitere Auswirkungen der technischen Grundkonzeption, z. B. auf die Effizienz und die Erfüllbarkeit datenschutzrechtlicher Vorgaben. Grundsätzlich schneiden Blockchains gegenüber klassischen zentralen Datenbanken in den Punkten Verfügbarkeit und Robustheit gegen Missbrauch positiv ab. Dem stehen auf der anderen Seite Nachteile im Bereich Vertraulichkeit und Effizienz gegenüber.

Die Nutzung von Blockchain allein löst keine IT-Sicherheitsprobleme. Vielmehr bleiben wohlbekannte Probleme wie die Sicherheit von Hard- und Software bestehen. Hinzu kommen neue Angriffsvektoren auf verschiedene Komponenten des Systems. Neben den Konsensmechanismen, mittels deren die verteilt gespeicherten Daten konsistent gehalten werden, und den Smart Contracts, die die Ausführung von Programmen im Blockchain-Netzwerk erlauben, sind hier beispielsweise externe Schnittstellen zum Einfügen und Auslesen von Daten zu nennen. Konkrete Vorfälle zeigen, dass die Angriffsmöglichkeiten nicht nur theoretischer Natur sind.

Es ist wichtig zu beachten, dass die Blockchain-Technologie ein breites Spektrum an Ausgestaltungen aufweist. Zur Differenzierung werden hier häufig die Einsehbarkeit der Daten (Leserechte) sowie die Fortschreibbarkeit der Blockchain (Schreibrechte) herangezogen. Jedoch gibt es auch für Konsensmechanismen und Smart Contracts eine Reihe von Ansätzen mit teils sehr unterschiedlichen Eigenschaften, die zu einem gewissen Grad mit der grundsätzlichen Ausgestaltung einer Blockchain zusammenhängen, aber durchaus weitere Freiheitsgrade bieten. Für konkrete Anwendungen, in denen Blockchains eingesetzt werden sollen, muss daher sorgfältig analysiert werden, welche Ausgestaltung am geeignetsten ist. Es ist davon auszugehen, dass das Modell von Bitcoin, das medial die größte Aufmerksamkeit findet, für die meisten Anwendungen nicht sinnvoll sein wird.

Im Bereich der Konsensmechanismen wird die breite Diskussion von dem Verfahren Proof-of-Work dominiert, das unter anderem von Bitcoin verwendet wird und insbesondere wegen seines immensen Energiebedarfs in der Kritik steht. Das Verfahren ermöglicht es, die Daten in einem Umfeld ohne Authentisierung der einzelnen Parteien konsistent zu halten und dabei Manipulationen zu verhindern. Unbeachtet bleibt oft die Tatsache, dass Blockchains mit strikterer Rechtevergabe und Authentisierung der Parteien, wie sie für viele Anwendungen geboten scheinen, den Einsatz von sogenannten nachrichtenbasierten Konsensmechanismen erlauben, die wesentlich effizienter und gut untersucht sind.

Verschiedene Blockchain-Systeme erlauben den Einsatz von Smart Contracts, die die manipulationssichere Abwicklung von Verträgen zwischen einander unbekannten oder misstrauenden Partnern ermöglichen sollen. Jedoch sind Smart Contracts nicht mit juristischen Verträgen gleichzusetzen, und nicht jeder Vertragsinhalt lässt sich überhaupt durch einen Smart Contract darstellen. Außerdem decken Analysen existierender Contracts eine große Zahl von Sicherheitsproblemen auf. Sie reichen von Fehlern im Code – die technologiebedingt nicht korrigiert werden können – über manipulierbare Zufallszahlen bis hin zu fehlender Authentizität der Daten, die aus der realen Welt kommend im Contract verarbeitet werden. Eine Berücksichtigung dieser Einschränkungen und Schwachstellen ist unerlässlich für einen verantwortungsbewussten Umgang mit Smart Contracts.

Da die Sicherheit von Blockchains in starkem Maße auf den verwendeten kryptografischen Algorithmen basiert, müssen diese sorgfältig ausgewählt werden, um das angestrebte Sicherheitsniveau hinsichtlich der Schutzziele Integrität, Authentizität und Vertraulichkeit zu erreichen. Detaillierte Empfehlungen hierzu finden sich in den im Text referenzierten Technischen Richtlinien des BSI. Ein bisher von den meisten Akteuren im Bereich Blockchain kaum beachteter Aspekt ist die Langzeitsicherheit. Um sensible Daten langfristig zu schützen, müssen Maßnahmen zur Verfügung stehen, die den Austausch kryptografischer Algorithmen ermöglichen, deren Sicherheitseignung abgelaufen ist. Dabei ist unbedingt zu beachten, dass ein Austausch von kryptografischen Verfahren nicht automatisch die ursprünglichen Sicherheitsgarantien für ältere Daten erhält. Neben den kryptografischen Verfahren müssen auch die Sicherheitseigenschaften der Konsensmechanismen klar verstanden und berücksichtigt werden. Diese Aspekte sollten entsprechend für Blockchain-Anwendungen von Anfang an bedacht werden.

Die rechtlichen Fragestellungen im Zusammenhang mit Blockchains resultieren unter anderem aus dem Fehlen einer zentralen rechtlich verantwortlichen Stelle im Regelbetrieb, woraus sich vielfältige Implikationen ergeben. Dieses Thema wird gegenwärtig kontrovers diskutiert. Datenschutzrechtliche Probleme, wie die Umsetzung von Vorgaben der Datenschutz-Grundverordnung (DSGVO), ergeben sich aus der auf Blockchains gerade erwünschten Transparenz und Manipulationssicherheit und sind ebenfalls Gegenstand intensiver Beschäftigung.

An Lösungen für eine ganze Bandbreite technischer Beschränkungen und Probleme der Blockchain-Technologie wird momentan ausgiebig und kreativ geforscht. Zu den aktuellen Forschungsthemen gehören beispielsweise die Aspekte Skalierbarkeit, Effizienz, Pseudonymität und Vertraulichkeit. Ob und wann sich hier signifikante Verbesserungen ergeben werden, die über den jetzigen Stand der Technologie hinausgehen, kann gegenwärtig nicht eingeschätzt werden.

Ein weiterer erwähnenswerter Punkt sind fehlende Standards im Bereich Blockchain. Dies führt zur Inkompatibilität verschiedener Blockchains und zu einer relativ unübersichtlichen Fülle an Lösungen, die die Auswahl eines konkreten Produkts für einen längerfristigen Zeithorizont für Anwender schwierig machen. Im Bereich der IT-Sicherheit liefert das BSI nun mit dem vorliegenden Dokument eine Entscheidungshilfe und fundierte Basis für zukünftige Diskussionen.

Inhaltsverzeichnis

Einleitung

Blockchains sind seit einiger Zeit nicht nur Experten, sondern durch häufige Erwähnung in den Medien auch einer breiteren Öffentlichkeit bekannt. Die Technologie kam 2009 mit der ersten erfolgreichen Kryptowährung Bitcoin auf, der bis heute viele weitere folgten. Seit einigen Jahren wird der Einsatz von Blockchains auch in zahlreichen anderen Bereichen vorgeschlagen und erprobt.

Insbesondere für Anwendungen außerhalb der Kryptowährungen handelt es sich bei Blockchain um eine vergleichsweise junge Technologie, an deren Einsatz schnell große Hoffnungen geknüpft werden, für die sich aber noch keine einheitlichen technischen Ansätze etabliert haben.

Wie bei jeder neuen Technologie sollte auch bei Blockchain das Prinzip *Security by Design* von Anfang an berücksichtigt werden. Daher veröffentlichte das BSI im Februar 2018 eine Liste von Eckpunkten, in denen es auf allgemeine Rahmenbedingungen, Anforderungen und Maßnahmen hinwies, die für den sicheren Einsatz der Technologie erforderlich sind [1]:

- Blockchain allein löst keine IT-Sicherheitsprobleme (vgl. Kapitel 5 und 7).
- Die Wahl des passenden Blockchain-Modells ist wichtig (vgl. Kapitel 1 und 2).
- Bei der Konstruktion von Blockchains müssen Sicherheitsaspekte frühzeitig berücksichtigt werden (vgl. Kapitel 3, 5, 6 und 10).
- Sensible Daten mit langfristigem Schutzbedarf müssen in einer Blockchain besonders geschützt werden (vgl. Kapitel 6).
- Einheitliche Sicherheitsniveaus für Blockchains müssen definiert und durchgesetzt werden (vgl. Kapitel 12).

Das vorliegende Dokument beschäftigt sich mit der gleichen Fragestellung, geht dabei jedoch deutlich weiter in die Tiefe. Es verfolgt nicht den Anspruch, das Thema Blockchain allgemeinverständlich aufzubereiten, sondern richtet sich in erster Linie an potenzielle Anwender, die den Einsatz von Blockchain erwägen und über fachliche Grundkenntnisse verfügen. Ziel ist es dabei, einen strukturierten und umfassenden Überblick insbesondere über diejenigen Aspekte des Themas zu geben, die einen Bezug zur IT-Sicherheit aufweisen. Den Lesern soll es damit ermöglicht werden, die für sie relevanten Fragestellungen zu identifizieren, ihre Projektideen unter dem Gesichtspunkt der IT-Sicherheit zu bewerten und gegebenenfalls konkrete Maßnahmen für den sicheren Betrieb von Blockchain-Lösungen abzuleiten.

Der Inhalt des Dokuments ist wie folgt gegliedert: Der erste Teil beginnt mit einer abstrakten Beschreibung der Technologie, ergänzt durch einige Beispiele, und führt allgemeine Fachbegriffe ein. Hieran schließt sich eine grundsätzliche Einordnung von Blockchain aus Sicht der IT-Sicherheit an, bevor die technischen Kernkomponenten Konsensmechanismen und Smart Contracts detailliert dargestellt werden. Dabei werden auch verschiedene Ansätze miteinander verglichen und Aussagen zur Sicherheit getroffen.

Der folgende Teil befasst sich intensiv mit den Sicherheitsaspekten des Themas. Hier werden allgemeine Hinweise zur Verwendung kryptografischer Verfahren gegeben. Dies betrifft einerseits die Auswahl der Verfahren, mit denen gewisse Sicherheitsgarantien erreicht werden können, aber auch konkretere Hinweise zur sicheren Umsetzung. Der nachfolgende Abschnitt widmet sich dem Thema Langzeitsicherheit in Blockchains, d. h. der Frage, ob und wie darauf reagiert werden kann, dass verwendete kryptografische Verfahren durch neuartige Angriffe ihre Sicherheitseignung verlieren. Abschließend wird eine Vielzahl bekannter Angriffe auf Blockchains aufgearbeitet und wo möglich Gegenmaßnahmen diskutiert.

Der dritte Teil behandelt rechtliche Fragestellungen. Neben einer allgemeinen juristischen Bewer-

tung fallen hierunter insbesondere Aspekte des Datenschutzes.

Im letzten Teil wird schließlich die praktische Anwendung von Blockchains näher beleuchtet. Dazu werden einige häufig genannte Anwendungsszenarien abstrakt analysiert und die aus Sicht der IT-Sicherheit relevanten Punkte hervorgehoben. Anschließend wird eine Reihe von Ansätzen und Ideen vorgestellt, die zur Weiterentwicklung und Verallgemeinerung von Blockchains unter dem Oberbegriff Distributed-Ledger-Technologie (DLT) erarbeitet und diskutiert werden. Neben Erweiterungen der Datenstruktur und Kombinationen mit anderen neuen Technologien ist hier auch die Interoperabilität ein wichtiges Forschungsthema.

Kryptografische Fachbegriffe werden am Ende dieses Dokuments in einem Glossar erklärt. Am Schluss jedes Kapitels findet sich eine Zusammenfassung der wichtigsten Aussagen.

Teil I Grundlagen

1 Definitionen und Taxonomie

Um ein gemeinsames Verständnis der Inhalte der Blockchain-Technologie und damit die Voraussetzung für die weiteren Ausführungen in diesem Dokument zu schaffen, werden in diesem Kapitel zunächst die wichtigsten Begriffe und grundlegenden Mechanismen im Bereich Blockchain erklärt und verschiedene Klassifikationsmöglichkeiten aufgezeigt. Um diese Grundlagen auch praktisch zu illustrieren, werden im Anschluss einige bekannte Beispiele für Blockchains vorgestellt.

1.1 Grundbegriffe

Die Grundidee der Blockchain-Technologie basiert auf der allgemeineren Konstruktion der sogenannten Distributed-Ledger-Technologien (*distributed ledger:* verteiltes digitales Analogon zum klassischen Journal der Buchführung). Diese beschreibt eine Technik zur verteilten Datenhaltung in einem Peer-to-Peer-Netzwerk (P2P-Netzwerk), bei der die Netzwerkknoten durch eine Übereinkunft (Konsens) gemeinsam über die Aktualisierung der Daten entscheiden. Bei den Daten kann es sich beispielsweise um Kontostände einer Kryptowährung, Herkunftsnachweise für Waren oder auch abstrakter um Vertragszustände von sogenannten Smart Contracts handeln.

Dabei gibt es keine zentrale Kommunikationssteuerung und keine zentrale Datenspeicherung. Die Netzwerkknoten verwalten jeweils eine lokale Kopie der gesamten Daten und können selbst neue Daten hinzufügen. Ein geeigneter Konsensmechanismus sorgt dafür, dass die verteilten Daten in allen Knoten aktuell sind und übereinstimmen und das Distributed Ledger als verteilte Datenstruktur damit stets in einem konsistenten Zustand gehalten wird.

Bei der Absicherung des Netzwerkzugangs, der Datenstruktur und der Konsensbildung werden kryptografische Verfahren eingesetzt, um die gewünschten Sicherheitsziele (insbesondere Integrität und Authentizität) zu erreichen. Die Regeln für die Validierung, Speicherung und Nutzung der Daten (Geschäftslogik) sind in den Datensätzen selbst codiert und werden bei der Verarbeitung automatisiert vom Netzwerk ausgeführt und durchgesetzt.

Speziell bei der Blockchain-Technologie werden alle Daten als sogenannte Transaktionen im Netzwerk validiert, zu Blöcken zusammengefasst und neue Blöcke durch eine kryptografische Verkettung manipulationssicher mit ihrem Vorgänger verbunden. Dadurch wird insbesondere eine chronologische Reihenfolge der Transaktionen festgelegt. Es entsteht eine stetig wachsende Kette von Datenblöcken, die sogenannte *Blockchain* als Spezialfall eines Distributed Ledgers (siehe Abbildung 1).

Die Blockchain-Technologie besteht also aus fünf grundlegenden Bausteinen (siehe Abbildung 2):

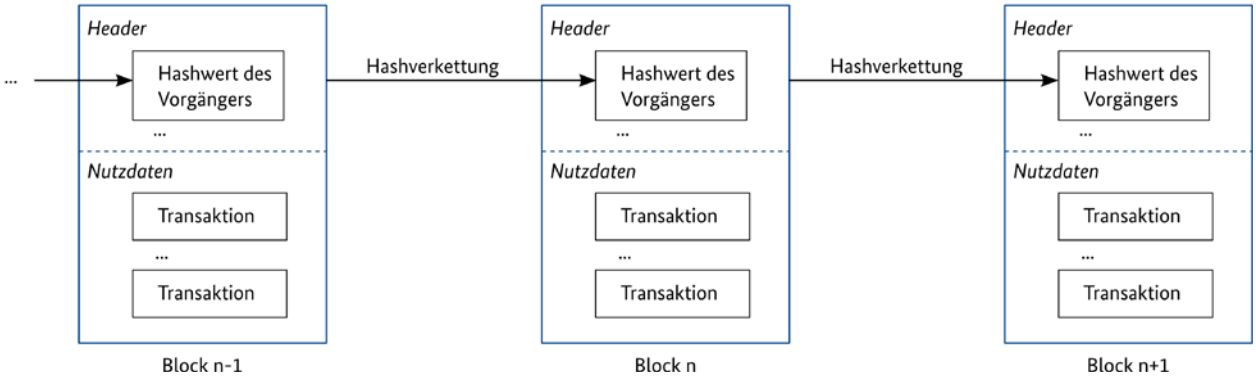

Abbildung 1: Blockchain-Datenstruktur

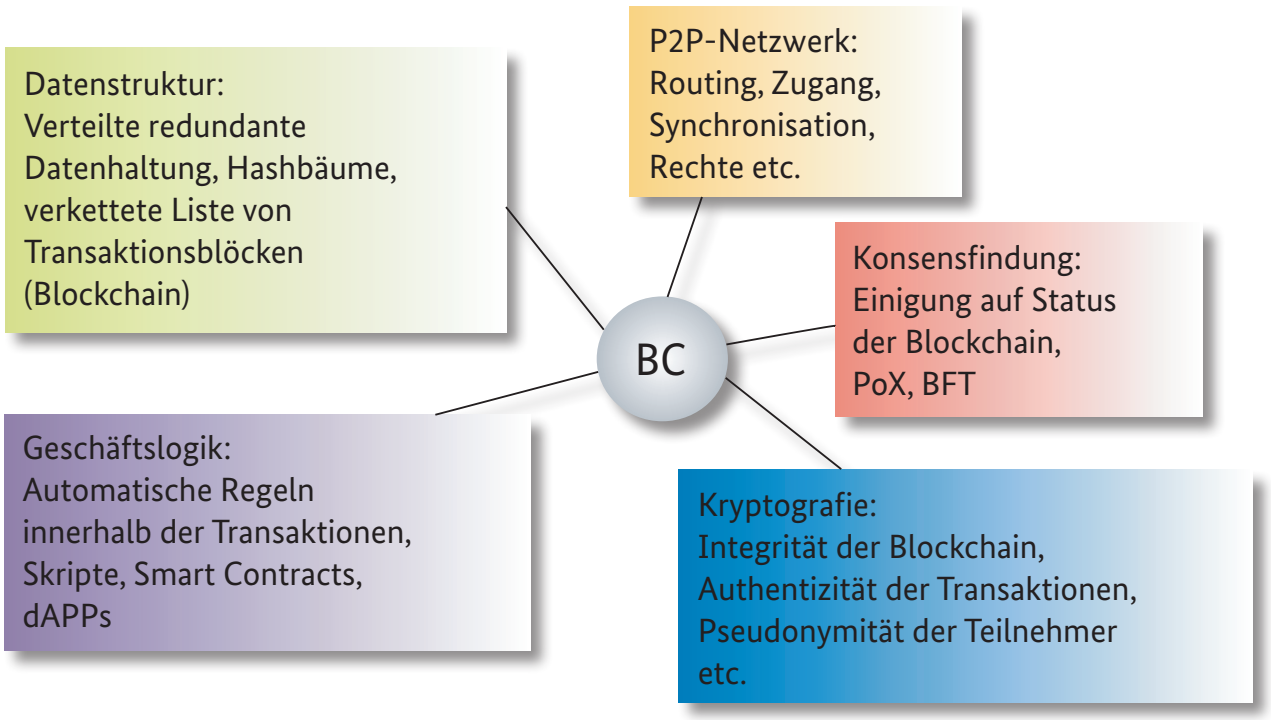

Abbildung 2: Grundbausteine der Blockchain-Technologie

- Peer-to-Peer-Netzwerk
- Datenstruktur Blockchain
- Konsensfindung
- Geschäftslogik
- Kryptografie

Diese Bausteine müssen je nach Anwendung passend modelliert werden, d. h. mit Netzwerkmodellen, Kommunikations- und Datenstrukturen, Konsensmechanismen, Regelwerken und kryptografischen Verfahren unterlegt werden. Beispielweise muss eine blockchainbasierte Kryptowährung so konzipiert werden, dass sie potenziell mit vielen wechselnden, unbekannten und möglicherweise nicht vertrauenswürdigen Nutzern zurechtkommt. Eine Blockchain zur Nachverfolgung von Luxusgütern dagegen muss insbesondere die Herkunft und Echtheit ihrer Daten zuverlässig sicherstellen. Die Auswahl und das Design eines passenden Blockchain-Modells sind entscheidend für die Erreichung der funktionalen und sicherheitstechnischen Ziele jeder Blockchain-Anwendung.

Das dabei entwickelte Blockchain-Modell kann wiederum auf unterschiedliche Art und Weise konkret umgesetzt werden. Dafür sind unter anderem die passende nachrichtentechnische Basis und die entsprechenden Kommunikationsprotokolle zu wählen, die richtigen Datenrepräsentationen und Regelsprachen auszusuchen und die passenden Algorithmen korrekt und sicher zu implementieren. Eine solche Umsetzung soll im Weiteren als Blockchain-System bezeichnet werden.

In jedem Blockchain-System (siehe Abbildung 3) gibt es bestimmte Kernkomponenten, die die Blockchain technisch definieren und die nicht austauschbar sind. Dazu zählen die Netzwerkstruktur, die Blockchain als Datenstruktur, der Konsensmechanismus und die Logik zur Systemsteuerung inklusive der zugrunde liegenden kryptografischen Mechanismen. Um diesen Blockchain-Kern herum wird das Blockchain-System durch verschiedene Infrastrukturkomponenten ergänzt, die für den Systembetrieb und die Umsetzung bestimmter sekundärer Funktionalitäten notwendig sind. Dabei handelt es sich zum Beispiel um den Netzwerkzugang, Schnittstellen in die Umgebung, Logik zur Systemverwaltung, kryptografische Zusatzfunktionen oder ein Rollen- und Rechtemanagement.

Hinter dem Begriff „Blockchain-Technologie“ – verkürzend wird oft auch einfach nur „Block-

chain“ verwendet – verbirgt sich also eine Vielzahl von theoretischen und praktischen Variationen und Kombinationsmöglichkeiten der Blockchain-Bausteine. Es kann nicht pauschal von „der“ Blockchain-Technologie oder „der“ Blockchain gesprochen werden, sondern je nach Anwendung muss ein passendes Blockchain-Modell erstellt und ein entsprechendes Blockchain-System entwickelt werden.

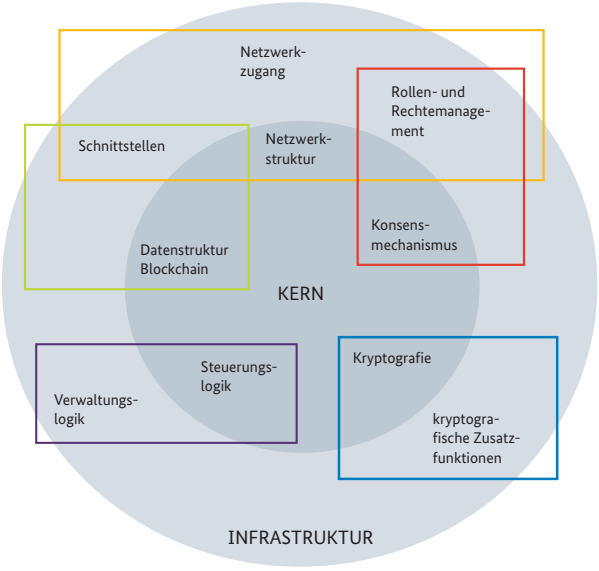

Abbildung 3: Schalenmodell eines Blockchain-Systems

1.2 Taxonomie

Aufgrund des großen Gestaltungsspielraums, den die Blockchain-Technologie bietet, können konkrete Blockchain-Modelle sehr unterschiedlich aussehen und damit auch völlig unterschiedliche Eigenschaften haben. Zur Klassifizierung haben sich dabei folgende Begrifflichkeiten durchgesetzt:

Einerseits wird unterschieden zwischen privaten *(private)* und öffentlichen *(public)* Blockchains. Hierbei geht es um die Verwendung des Netzwerks und die Einsehbarkeit der Daten. Öffentliche Blockchains erlauben uneingeschränkt das Einstellen von Daten im Netzwerk und die Einsicht in alle Transaktionen der Blockchain. Dagegen schränken private Blockchains diese Nutzung auf bestimmte Nutzergruppen, z. B. eine Organisation oder ein Konsortium, ein.

Auf der anderen Seite gibt es die Differenzierung in genehmigungsbasierte *(permissioned)* und genehmigungsfreie *(unpermissioned/permissionless)* Blockchains, die die Erlaubnis zur Fortschreibung der Blockchain betrifft. Dürfen grundsätzlich alle Netzwerkknoten an der Validierung der Transaktionen, der Bildung neuer Blöcke und an der Konsensbildung teilnehmen, so handelt es sich um eine genehmigungsfreie Blockchain. Sind diese Prozesse jedoch nur nach vorheriger Auswahl und Zulassung durch eine zentrale Autorität zugänglich, so spricht man von einer genehmigungsbasierten Blockchain.

Je nach Anwendungsbereich sollte bereits bei der Konzipierung einer Blockchain eine Einordnung in die passenden Klassen vorgenommen werden. Dabei spielt unter anderem eine Rolle, für welchen Nutzerkreis die Blockchain gedacht ist, welche Daten verarbeitet werden, aber auch wie die wirtschaftlichen und rechtlichen Rahmenbedingungen für den Blockchain-Betrieb aussehen. Diese erste grobe Einordnung hat dann auch entsprechende Auswirkungen auf die Ausgestaltung des Blockchain-Modells, insbesondere auf die Wahl des Konsensmechanismus und der kryptografischen Absicherung.

Während öffentliche genehmigungsfreie *(public unpermissioned)* Blockchains (wie zum Beispiel Bitcoin) im Normalbetrieb ohne eine zentrale Instanz auskommen, müssen bei den anderen Modellen zumindest die entsprechenden Berechtigungen bzw. Einschränkungen mehr oder weniger zentral vorgegeben und durchgesetzt werden, so dass eine geeignete Autorität für die Blockchain vorhanden sein muss.

1.3 Beispiele

Im weiteren Verlauf des Dokuments werden neben allgemeinen Ausführungen immer wieder auch Beispiele aus der Praxis genannt, um die beschriebenen Sachverhalte zu illustrieren. Vorab sollen hier deshalb einige prominente Beispiele für Blockchains kurz eingeführt werden, die aufgrund ihrer Verbreitung, ihres Reifegrads oder ihrer ökonomischen Relevanz eine besondere Rolle spielen.

Beispiel: Bitcoin

Das bestimmt bekannteste Beispiel für eine öffentliche genehmigungsfreie Blockchain-Anwendung ist die Kryptowährung Bitcoin [2], die seit 2009 ohne Unterbrechung online ist und mit etwa 67 Milliarden US-Dollar die bei Weitem größte Marktkapitalisierung aller Kryptowährungen aufweist (Stand März 2019).

Bei der Erstellung von Empfänger- und Absenderadressen sowie zur Authentisierung von Zahlungen (Transaktionen) kommt im Bitcoin-System ein Public-Key-Verfahren zum Einsatz. Die benötigten Schlüsselpaare können dabei in sogenannten Wallets („Geldbörsen“) gespeichert, verwaltet und zum Teil auch erzeugt werden. Eine Bitcoin-Adresse ist vereinfacht ausgedrückt der Hashwert eines öffentlichen Signaturschlüssels (siehe Abbildung 4 oben). Die genaue Beschreibung der Erzeugung von Bitcoin-Adressen findet sich beispielsweise in [3].

Zum Überweisen wird der zur Absenderadresse passende öffentliche Schlüssel des Überweisenden und ein mit dem zugehörigen privaten Schlüssel signierter Datensatz übertragen (siehe Abbildung 4 unten). Der Empfänger prüft dann durch Wiederholung der Hashprozedur, ob der öffentliche Schlüssel zur Absenderadresse gehört, und mit Hilfe des öffentlichen Schlüssels, ob die Signatur korrekt ist. Bei einem positiven Ergebnis beider Prüfungen ist klar, dass der Absender auch tatsächlich über das überwiesene Guthaben verfügt.

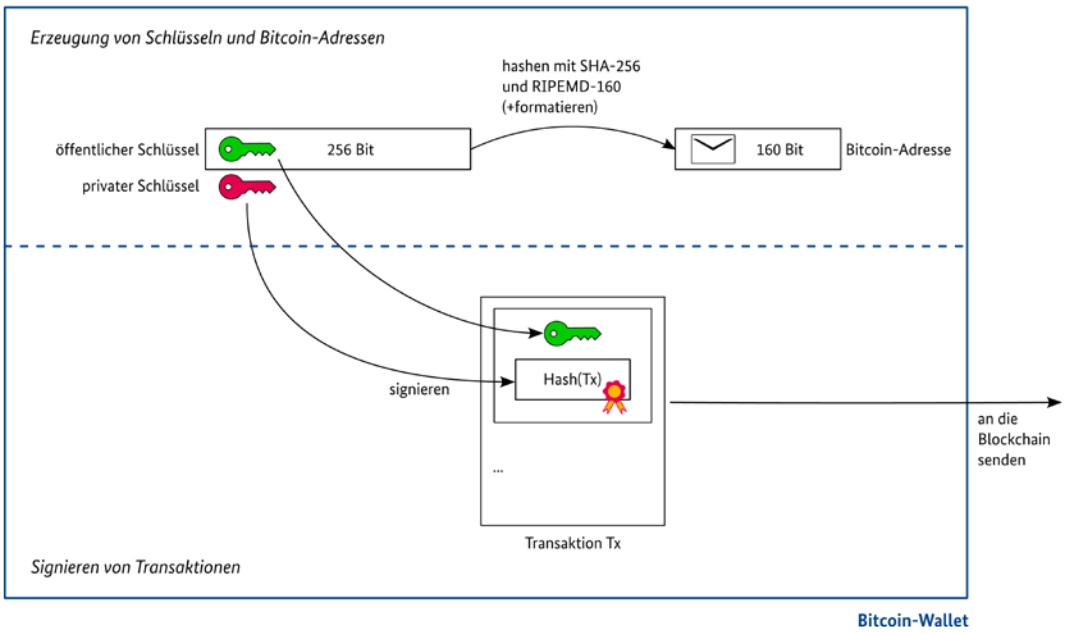

Abbildung 4: Schematische Darstellung der Adresserzeugung und Signierung von Transaktionen in einem Bitcoin-Wallet

Alle Transaktionen werden im gesamten Bitcoin-Netzwerk verteilt und dann von sogenannten Minern bestätigt, zu neuen Blöcken zusammengefasst und an das Ende der Bitcoin-Blockchain angehängt. Dieses Anhängen ist mit Hilfe einer kryptografischen Hashfunktion gesichert, was die Integrität der gesamten Kette garantiert. Zur Konsensbildung wird bei Bitcoin der sogenannte Proof-of-Work eingesetzt (siehe auch Abschnitt 3.2.3): Um einen Block hinzufügen zu können, muss ein Hashwert einer bestimmten Form berechnet werden, was im Allgemeinen sehr rechenintensiv ist. Als Anreiz erhalten die Miner für das Lösen dieser Aufgabe und die Verlängerung der Blockchain eine Belohnung in Form von (neu geschaffenen) Bitcoins und können zusätzlich Transaktionsgebühren erwarten. Dieses Verfahren wird auch Mining (also „Schürfen“) genannt.

Bitcoin als offene, pseudonyme und unregulierte Blockchain, die als Open-Source-Software verfügbar ist, verkörpert für viele die reine Lehre der Blockchain-Idee. Dieser Blockchain-Ansatz ist aber bei Weitem nicht für alle Einsatzbereiche geeignet und darf den Blick auf das breite Spektrum der Blockchain-Technologie nicht einschränken.

Viele Begriffe der Blockchain-Technologie stammen ursprünglich aus der Bitcoin-Welt und werden heute oft in einer erweiterten Bedeutung verwendet. Dazu gehört neben *Blockchain* selbst als Bezeichnung für die Datenstruktur hinter Bitcoin z. B. auch der Begriff *Transaktion*, der sich nicht nur auf die Übertragung von Bitcoins oder anderer digitaler Geldeinheiten beschränkt, sondern ganz allgemein ein Stück Information bezeichnet, das in der Blockchain verwaltet wird. Auch der Begriff *Wallet* beschreibt nicht zwangsläufig nur eine digitale Brieftasche, er bedeutet vielmehr eine generelle Benutzerschnittstelle zum Blockchain-Netzwerk, über die der Nutzer seine Zugangsdaten und bestimmte Geheimnisse verwalten und am System teilhaben kann. Ebenso kann der Begriff *Miner* allgemein einen Akteur bezeichnen, der der Blockchain neue Blöcke hinzufügen darf.

Beispiel: Ethereum

Auch Ethereum [4], [5] ist ein öffentliches genehmigungsfreies Blockchain-System. Es stellt die Kryptowährung Ether zur Verfügung, die mit einer Marktkapitalisierung von über 14 Milliarden US-Dollar auf dem zweiten Rang liegt (Stand März 2019). Vor allem aber hat Ethereum als blockchainbasierte Plattform für Smart Contracts große Bekanntheit erlangt.

Als Kryptowährung operiert Ethereum ähnlich wie Bitcoin. Die zugrunde liegenden Strukturen und Abläufe sowie auch der Konsensmechanismus sind vergleichbar. Transaktionen in Ethereum können allerdings nicht nur Wertüberweisungen enthalten, sondern auch ausführbaren Programmcode, sogenannte Smart Contracts (siehe Kapitel 4). Ein solcher Contract erhält eine eigene Adresse, die zur weiteren Interaktion benutzt wird und sich ansonsten nicht von normalen Nutzeradressen unterscheidet.

Neben den Transaktionsdaten in der Blockchain gibt es in Ethereum einen übergreifend akzeptierten Systemstatus *(world state)*, der unter anderem die aktuellen Kontostände und – im Falle von Contract-Adressen – den Hashwert des zugehörigen Bytecodes enthält. Der Systemstatus liegt nicht auf der Blockchain, sondern lokal im Speicher der einzelnen Knoten [6]. Gültige Transaktionen beschreiben den Übergang vom aktuellen zu einem neuen Status, z. B. Änderungen von Kontoständen. Um sicherzustellen, dass dabei die lokalen Kopien des Systemstatus auf allen Knoten konsistent bleiben, enthalten Blöcke unter anderem den Hashwert des aktuellen Status. Um diesen zu ermitteln, müssen beim Mining alle Smart Contracts ausgeführt werden, die durch im Block enthaltene Transaktionen initiiert oder adressiert werden. Ebenso ist es zur Verifikation eines jeden Blocks erforderlich, dass sämtliche Knoten jeden betroffenen Smart Contract ein weiteres Mal ausführen, um den hinterlegten Status-Hash zu überprüfen.

Beispiel: Hyperledger Fabric

Hyperledger Fabric [7], [8] ist ein weiteres bekanntes Blockchain-System. Es stellt keine Kryptowährung zur Verfügung, sondern ist in erster Linie eine Plattform für Smart Contracts. Im Gegensatz zu den bisher genannten ist es privat und genehmigungsbasiert, was den Einsatz von nachrichtenbasierten Konsensverfahren – insbesondere CFT- und BFT-Verfahren (siehe Abschnitt 3.2.2) – erlaubt, die den Minern keine großen Kosten erzeugen.

Ein Ziel von Hyperledger Fabric ist, ein modular aufgebautes Blockchain-System bereitzustellen, bei dem einzelne Komponenten ausgetauscht werden können. Dadurch soll eine Möglichkeit zur individuellen Anpassung des Systems an eigene Bedürfnisse gegeben werden. Insbesondere sind die Validierung der Blöcke und der Konsensmechanismus voneinander getrennt. Neben der größeren Flexibilität gelingt es dadurch, den Ressourcenverbrauch zu begrenzen (siehe Abschnitt 4.1).

Zusammenfassung.

- Die Blockchain-Technologie kann modular definiert werden und erlaubt viele unterschiedliche Ausprägungen.
- Je nach Anwendungsfall muss das richtige Blockchain-Modell gewählt werden.
- Es gibt verschiedene Möglichkeiten, die Lese- und Schreibrechte auf einer Blockchain zu gestalten.

2 Einordnung

In diesem Kapitel werden Blockchains hinsichtlich verschiedener Eigenschaften bewertet und mit ihrer klassischen Alternative in Gestalt von Datenbanken verglichen. Diese Analyse ermöglicht eine grundsätzliche Einschätzung, inwiefern der Einsatz der Blockchain-Technologie in einem gegebenen Kontext zielführend ist und welche Vor- und Nachteile gegenüber Datenbanken bestehen.

Zu den betrachteten Eigenschaften zählen einerseits die Schutzziele der IT-Sicherheit, die im nächsten Abschnitt genauer eingeführt werden, und andererseits die praktischen Anforderungen an den Durchsatz und die Skalierbarkeit der Technologie. Diese Merkmale stehen teilweise in einem Spannungsverhältnis zueinander sowie mit dem grundsätzlichen Design von Blockchains, das durch den Wunsch nach Dezentralität, Transparenz und Manipulationssicherheit motiviert ist (siehe Abbildung 5).

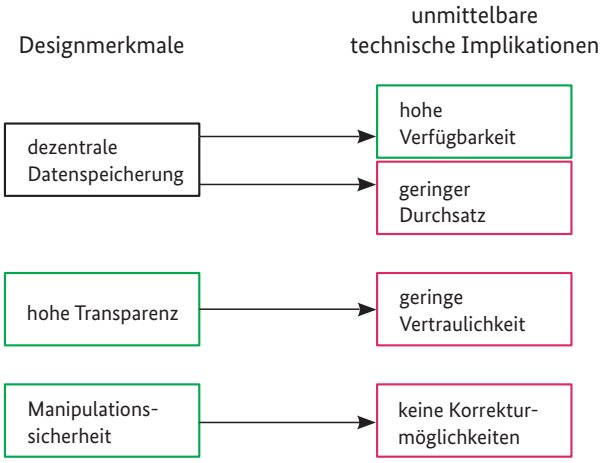

Abbildung 5: Designziele und ihre unmittelbaren technologischen Implikationen

So ist aufgrund der gewollten Transparenz die Herstellung von Vertraulichkeit und auch Anonymität in Blockchains sehr schwierig (siehe Abschnitte 5.1 und 5.4). Gleiches gilt für die Erfüllung von Datenschutzanforderungen (siehe Abschnitt 9.3).

Die sehr gute Verfügbarkeit in Blockchain-Systemen resultiert aus der dezentralen Datenspeicherung. Andererseits verringert diese wegen der auftretenden Latenzzeiten den Durchsatz des Systems gegenüber zentralisierten Lösungen und verursacht zusätzlichen Speicherbedarf sowie damit verbundene Kosten. Insofern ist immer eine Abwägung dieser verschiedenen Aspekte erforderlich.

2.1 IT-Sicherheit

2.1.1 Schutzziele

Die klassischen Schutzziele der IT-Sicherheit, anhand derer technische Lösungen bewertet werden, sind die Integrität, Authentizität, Verfügbarkeit und Vertraulichkeit. Darüber hinaus wird häufig auch die Anonymität bzw. Pseudonymität betrachtet. Die Begriffe sind dabei wie folgt zu verstehen (vgl. IT-Grundschutz [9]):

- Integrität bezeichnet die Sicherstellung der Vollständigkeit und Korrektheit (Unversehrtheit) von Daten. Ein Verlust der Integrität kann daher bedeuten, dass Daten unerlaubt verändert, Angaben zum Autor verfälscht oder Zeitangaben zur Erstellung manipuliert wurden.

- Authentizität bezeichnet die Eigenschaft, die gewährleistet, dass ein Kommunikationspartner (eine Person oder IT-Komponente oder -Anwendung) tatsächlich derjenige ist, der er vorgibt zu sein. Bei authentischen Informationen ist sichergestellt, dass sie von der angegebenen Quelle erstellt wurden.

- Die Verfügbarkeit von Dienstleistungen, IT-Anwendungen oder auch von Informationen ist vorhanden, wenn diese von den Anwendern stets wie vorgesehen genutzt werden können.

- Vertraulichkeit ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein.

- Die Anonymität einer Entität liegt vor, wenn sie nicht identifiziert werden kann. Anonymität garantiert insbesondere, dass Daten oder Handlungen der gleichen Entität nicht miteinander verknüpft werden können. Von Pseudonymität spricht man dagegen, wenn eine solche Verknüpfung über ein Pseudonym durchgeführt werden kann, gleichzeitig aber eine Zuordnung des Pseudonyms zu einer realen Identität nicht möglich ist.

Die folgende Bewertung beschränkt sich auf die Blockchain selbst. Insbesondere gelten die verschiedenen Sicherheitsgarantien für Daten erst dann, wenn sie in der Blockchain gespeichert wurden. So erstreckt sich beispielsweise der Integritätsschutz nicht auf das Einlesen von Daten über eine Schnittstelle zur realen Welt (z. B. Sensordaten, Ereignisse in der realen Welt). Hierfür wären, ebenso wie bei alternativen Technologien, zusätzliche Maßnahmen außerhalb der Blockchain erforderlich.

Wo möglich und sinnvoll differenziert die Analyse zwischen öffentlichen und privaten Blockchains.

2.1.2 Integrität

Die Integrität der in einer Blockchain abgelegten Daten wird im Wesentlichen dadurch sichergestellt, dass die einzelnen Blöcke mittels einer Hashfunktion verkettet werden. Sofern eine geeignete Hashfunktion gewählt ist, können Daten nicht nachträglich manipuliert werden, ohne dass dies auffällt.

Es ist jedoch darauf hinzuweisen, dass die Sicherheitseignung von Hashfunktionen eventuell nur über einen gewissen Zeitraum besteht und langfristig Anpassungen nötig sein können (siehe Abschnitt 6.2). Weiterhin sollte bedacht werden, dass der Schutz der Integrität für die Daten oft erst einige Zeit nach ihrer Aufnahme in die Blockchain garantiert ist. Bei verschiedenen Konsensmechanismen ist nämlich eine gewisse Wartezeit nötig, bis die Daten in Blöcken nur noch mit unrealistischem Aufwand verändert werden können. Die Wahl des Konsensmechanismus steht wiederum mit dem Blockchain-Modell in Zusammenhang (siehe Abschnitt 3.2). Grundsätzlich kann die Integrität auf privaten oder genehmigungsbasierten Blockchains schneller gesichert werden als auf öffentlichen genehmigungsfreien Blockchains.

2.1.3 Verfügbarkeit

Aufgrund der technologieimmanenten verteilten und dezentralen Speicherung sind die Informationen zu jedem Zeitpunkt mit hoher Wahrscheinlichkeit verfügbar, sofern eine ausreichende Anzahl an Knoten, die den vollständigen Datensatz vorhalten, vorhanden ist, sodass auch Teilausfälle des zugrunde liegenden Netzwerks toleriert werden können. Gezielte Angriffe auf eine einzige zentrale Stelle sind nicht möglich, in privaten Blockchains kann ein Angreifer jedoch mit größerem Aufwand durchaus die Verfügbarkeit einschränken. Grundsätzlich gilt, dass eine höhere Rate an Verbindungen die Resistenz des Netzwerks gegen Ausfälle aufgrund gezielter Angriffe oder technischen Versagens stärkt.

Anzumerken ist, dass die gute Verfügbarkeit nur für die direkt in der Blockchain gespeicherten Daten gegeben ist. Wenn Daten, z. B. aus Datenschutz- oder Vertraulichkeitsgründen (siehe Abschnitt 5.2), ausgelagert und in der Blockchain nur Verweise gespeichert werden, so liefert die Blockchain keine Verfügbarkeitsgarantien für die eigentlichen Daten.

2.1.4 Vertraulichkeit

Die Herstellung von Vertraulichkeit auf einer Blockchain ist konstruktionsbedingt als sehr schwierig anzusehen und sollte daher kein Schutzziel beim Einsatz von Blockchains sein. Ursache dafür ist, dass die Transaktionsdaten allen teilnehmenden Knoten zur Verfügung stehen

müssen und damit auch die Verschlüsselung der Daten sehr schwierig umsetzbar ist. Es existieren einige komplexe Vorschläge zur Herstellung von Vertraulichkeit, die in der Regel jedoch relativ ineffizient sind (siehe Abschnitt 5.1).

Auf privaten Blockchains ist die Anzahl der Knoten in der Regel deutlich geringer als bei öffentlichen. Vertraulichkeit ist gegenüber denjenigen gewährleistet, die keinen Zugang zum Netzwerk haben, sofern die Kommunikation innerhalb des Netzwerks angemessen geschützt wird. Zudem gibt es auf privaten Blockchains Möglichkeiten, die Sichtbarkeit weiter einzuschränken (z. B. [8]), die aber mit einem höheren Aufwand verbunden sind. Auch wenn bei diesen Lösungen die Daten immer für einige Knoten mit höheren Rechten sichtbar bleiben, sind sie womöglich für viele Anwendungsfälle hinreichend.

Eine grundsätzliche Idee, um Probleme mit der Vertraulichkeit zu umgehen, besteht in der externen Speicherung der relevanten Daten (siehe Abschnitt 5.2). In diesem Fall ist aber gemäß Abschnitt 2.1.3 insbesondere die Verfügbarkeit nicht durch die Blockchain gewährleistet.

2.1.5 Authentizität

Auf einer Blockchain werden Transaktionen in der Regel durch digitale Signaturen mittels eines Public-Key-Kryptosystems abgesichert. Bei geeigneter Wahl dieses Kryptosystems ist die Fälschung einer Signatur für gegebene Daten extrem schwierig und somit sind die Transaktionen authentisch. Jedoch sei auch hier bezüglich der Langzeitsicherheit auf den Abschnitt 6.2 verwiesen.

Wichtig ist, dass durch die Blockchain die Authentizität nur bezüglich der Identitäten innerhalb des Netzwerks sichergestellt werden kann. Die Zuordnung der Schlüssel zu einem konkreten Kommunikationspartner muss ebenso wie bei anderen Technologien, die digitale Signaturen verwenden, durch zusätzliche Maßnahmen erzielt werden. Auf einer öffentlichen genehmigungsfreien Blockchain sind dazu keine Ansätze etabliert. Auf privaten oder genehmigungsbasierten Blockchains müssen hierzu die Betreiber der Knoten bei Erzeugung der Schlüssel in geeigneter Weise identifiziert und die Zuordnung der Schlüssel dokumentiert werden, z. B. über eine Public-Key-Infrastruktur (PKI).

Um die Authentizität der Transaktionsdaten sicherzustellen, ist der Schutz der entsprechenden privaten Signaturschlüssel von größter Wichtigkeit. Insbesondere beim Einsatz von statischen (d. h. dauerhaft verwendeten) Schlüsseln erlischt mit einem Schlüsselverlust gleichzeitig auch die Garantie für die Authentizität des Knotens selbst.

2.1.6 Anonymität/Pseudonymität

Aufgrund der Transparenz von Blockchains können verschiedene Transaktionen einer Entität miteinander verknüpft werden, sodass höchstens Pseudonymität vorliegt. Auf privaten Blockchains ist dies unproblematisch, da in der Regel die Identifizierung der Knoten sogar gewünscht ist. Es wird vorgeschlagen, auf öffentlichen Blockchains für verschiedene Transaktionen verschiedene Pseudonyme zu verwenden, um die Verknüpfung von Transaktionen zu verhindern und Anonymität zu erzielen. Diese Maßnahmen stoßen jedoch an fundamentale Grenzen und lassen sich mit ausreichendem Aufwand aufheben. Auch die Aufhebung der Pseudonymität ist durch Einbeziehung weiterer Informationen, die häufig außerhalb der eigentlichen Blockchain liegen, als möglich anzusehen. Für Details wird auf Abschnitt 5.4 verwiesen.

2.2 Effizienz

Sollen Blockchains hin zu einem großflächigen Einsatz und einer großen Zahl an Transaktionen skaliert werden, so sind in der Praxis ihr Ressourcenbedarf und Durchsatz sehr wichtige Kennwerte. Zu den hier betrachteten Ressourcen zählen dabei der genutzte Speicher und insbesondere die benötigte Energie. Der Durchsatz einer Blockchain bezeichnet die Menge an Transaktionen, die in einer festen Zeitspanne in neue Blöcke aufgenommen werden können.

2.2.1 Ressourcenbedarf

Die sehr gute Verfügbarkeit der in einer Blockchain abgelegten Daten resultiert aus der vielfachen verteilten Speicherung aller Daten ab dem ersten Block. Insbesondere auf öffentlichen, großflächig eingesetzten Blockchains wäre der notwendige, schnell wachsende Speicherbedarf jedoch für privat betriebene Knoten nicht tragbar. Vor allem auf privaten Blockchains könnte sich dieses Problem durch regelmäßiges, zumindest teilweises Löschen oder Komprimieren hinreichend alter Daten abmildern.

Ein oft genanntes Hindernis für die Skalierbarkeit von Blockchains ist der Energieverbrauch, der beispielsweise bei Bitcoin gemäß Schätzungen vom März 2019 für etwa 0,2 % des weltweiten Stromverbrauchs verantwortlich und damit exorbitant hoch ist [10]. Dies liegt in erster Linie in dem verwendeten Konsensmechanismus begründet. Je nach Blockchain-Modell können jedoch Konsensmechanismen genutzt werden, deren Energieverbrauch vernachlässigbar gering ist (siehe Abschnitt 3.2), weshalb insbesondere bei privaten Blockchains der Energieverbrauch kein Problem darstellt.

2.2.2 Durchsatz

Der Durchsatz einer Blockchain hängt ebenfalls sehr stark vom Konsensmechanismus ab. Auf einer niedrigeren Ebene ist er zudem von physikalischen Eigenschaften des zugrunde liegenden Netzwerks abhängig, da die dezentrale Speicherung und Verteilung der Daten eine ständige Kommunikation der verschiedenen Knoten innerhalb des Netzwerks erfordern. Hier ist erstens die Bandbreite zu nennen, die angibt, welche Datenmenge pro Zeiteinheit innerhalb des Netzwerks verteilt werden kann. Zweitens ist die Latenzzeit zu berücksichtigen, die die Zeit zur Übertragung eines Datenpakets innerhalb des Netzwerks bezeichnet. Beide Werte unterliegen über die Zeit und in Abhängigkeit von den Netzwerkknoten, die miteinander kommunizieren, gewissen Schwankungen.

Grundsätzlich erlauben private Blockchains gegenüber öffentlichen einen deutlich höheren Durchsatz. Dies ist vor allem in der Struktur des Netzwerks begründet, das bei privaten Blockchains im Regelfall deutlich weniger Knoten als bei öffentlichen enthält. Weiterhin ermöglichen die auf privaten Blockchains einsetzbaren Konsensmechanismen einen um Größenordnungen höheren Durchsatz als die gegenwärtig auf öffentlichen Blockchains verwendeten. So lag der Durchsatz von Bitcoin im März 2019 bei etwa 7 tps (Transaktionen pro Sekunde), während Konsensmechanismen auf privaten Blockchains 20.000 tps oder mehr erreichen ([11], siehe Abschnitt 3.2).

2.3 Vergleich mit Datenbanken

Blockchains werden häufig als eine Alternative zu klassischen Datenbanken gesehen. Die Entscheidung für den Einsatz einer der beiden Technologien kann dabei anhand verschiedener Kriterien erfolgen. Als erste Entscheidungshilfe wird hier ein grober Vergleich der Technologien gegeben. Aus Gründen der Übersichtlichkeit werden lediglich die zwei wichtigsten Blockchain-Modelle, eine öffentliche genehmigungsfreie Blockchain (z. B. Bitcoin) sowie eine private genehmigungsbasierte Blockchain (z. B. Hyperledger-Instanzen), einer klassischen Client-Server-Datenbank gegenübergestellt (siehe Tabelle 1). Darin hält ein zentraler Server die Daten vor, auf die Clients per Abfrage zugreifen können. Grundsätzlich ist zu beachten, dass einzelne Aspekte je nach konkreter Ausgestaltung in einem gewissen Rahmen variieren können.

Datenbanken stellen die Integrität der gespeicherten Daten durch eine Reihe von Maßnahmen sicher. Dazu gehören in erster Linie die regelmäßige Erstellung von Sicherheitskopien und Log-Dateien sowie eine starke Zugriffskontrolle auf die Daten. Der zentrale Server stellt einen Single-Point-of-Failure dar, sodass gezielte Angriffe die Verfügbarkeit stark einschränken können. Durch redundante Datenhaltung lässt sich dieses Risiko jedoch im gewünschten Maße abmildern. Die Vertraulichkeit der Daten lässt sich in einer Datenbank durch den geeigneten Einsatz kryptografischer Routinen erreichen, wobei Zugriffsrechte nur an jeweils befugte Nutzer vergeben werden (vgl. auch IT-Grundschutz [12]).

Ein wesentlicher Unterschied besteht darin, dass eine Datenbank – auch mit redundanter Datenhaltung – unter Kontrolle eines einzelnen Betreibers steht. Im Gegensatz zu Blockchain-Lösungen kann somit die Einhaltung der Schutzziele durch missbräuchliches Verhalten dieser Entität beeinträchtigt werden, ohne dass dies technisch verhindert wird.

Durch zusätzliche organisatorische Sicherheitsmaßnahmen und aufgrund rechtlicher Rahmenbedingungen wird dieses Risiko de facto als gering eingeschätzt.

Wegen der zentralen Infrastruktur ist der Rechen- und Kommunikationsaufwand und auch der Energieverbrauch zum Betrieb einer Datenbank sehr gering. Gleichzeitig ist ein sehr hoher Durchsatz möglich. Klassische Bezahldienstleister berichteten beispielsweise 2015 von bis zu 56.000 tps im Rahmen von Stresstests [13].

Kurz gefasst bieten Blockchains gegenüber Datenbanken Vorteile in der Robustheit gegen Missbrauch und gegebenenfalls der Verfügbarkeit, wohingegen sich in den Punkten Vertraulichkeit und Effizienz deutliche Nachteile ergeben.

	Blockchain (öffentlich genehmigungsfrei)	Blockchain (privat genehmigungsbasiert)	Client-Server-Datenbank
Integrität	++	++	++
Authentizität	umsetzungsabhängig		
Verfügbarkeit	++	++	+
Vertraulichkeit	--	o*	+
Pseudonymität	o	--	--
Dezentralität	++	o	--
Robust gegen Missbrauch	++	+	o
Transparenz	++	o	--
Ressourcenbedarf	(sehr) hoch†	gering	sehr gering
Durchsatz	--	+	++

Tabelle 1: Vergleich von Blockchain und Datenbanken (vgl. Abschnitte 2.1 und 2.2).

* In einigen Blockchains gab es Fortschritte, die diese Bewertung widerspiegelt, in vielen Lösungen ist die Vertraulichkeit aber schwächer als angegeben zu beurteilen.

† Alle bedeutenden öffentlichen genehmigungsfreien Blockchains verwenden momentan den Konsensmechanismus Proof-of-Work (siehe Abschnitt 3.2). Ein Übergang zu weniger energieintensiven Konsensmechanismen ist teilweise geplant (Stand März 2019).

Zusammenfassung.

- Die Schutzziele der IT-Sicherheit, Anforderungen an die Effizienz sowie die Designziele von Blockchains stehen in einem Spannungsverhältnis zueinander.
- Es gibt große Unterschiede zwischen öffentlichen genehmigungsfreien und privaten genehmigungsbasierten Blockchains.
- Einige Sicherheitsziele, insbesondere die Authentizität, müssen durch zusätzliche Infrastrukturmaßnahmen sichergestellt werden.
- Blockchains bieten gegenüber zentralen Datenbanken Vorteile in den Punkten Robustheit gegen Missbrauch und Verfügbarkeit.
- Blockchains weisen gegenüber zentralen Datenbanken Nachteile in den Punkten Vertraulichkeit und Effizienz auf.

3 Vertrauen und Konsens

Die verteilte Datenhaltung ist eine wesentliche Eigenschaft von Blockchains. Einerseits erhöht sie auf technischer Ebene die Resistenz gegen missbräuchliches Verhalten, das daher in geringerem Maße durch organisatorische Vorkehrungen verhindert werden muss. Andererseits macht sie Maßnahmen unumgänglich, um die Übereinstimmung der verschiedenen Kopien der Daten sicherzustellen. Der hierfür verwendete Konsensmechanismus bildet somit eine Kernkomponente von Blockchains, deren Ausgestaltung für die Sicherheit des gesamten Systems grundlegend ist.

3.1 Blockchains und Vertrauen

Ein wesentliches Merkmal einer Blockchain ist das Fehlen einer zentralen Instanz, über die wie in herkömmlichen Lösungen die Kommunikation geleitet und verwaltet wird und der von den Nutzern vertraut werden muss. Die ursprüngliche Bitcoin-Veröffentlichung [2] gibt diese Eigenschaft, gemäß der die Sicherheit von Blockchains „statt auf Vertrauen auf kryptografischen Beweisen“ basiert, als grundlegende Motivation für die Konzeption von Bitcoin an. Sie wird sehr häufig als die wesentliche Neuerung der Blockchain-Technologie bezeichnet.

Im eigentlichen Betrieb der Blockchain existiert anders als bei traditionellen Lösungen wie Datenbanken keine zentrale Instanz wie oben beschrieben. Allerdings muss auch auf Blockchains einigen Akteuren und Komponenten ein erhebliches Maß an Vertrauen entgegengebracht werden.

Eine wichtige Position haben die Programmierer der Software inne, die zum Betrieb und zur Teilnahme an der Blockchain verwendet wird. Die Nutzer müssen darauf vertrauen, dass die Programmierer tatsächlich die gewünschte Funktionalität implementieren und nicht versehentlich oder absichtlich Schwachstellen in die Software integrieren. Dies gilt auch, wenn Blockchains als Open-Source-Projekte entwickelt und betrieben werden. Dabei steuern die Programmierer die allgemeine Entwicklung des Projekts, und die öffentliche Einsehbarkeit des Quellcodes muss mangels Zeit und Expertise der Nutzer nicht unbedingt zur Aufdeckung subtiler Schwachstellen führen. Dieses Vertrauen in die korrekte Programmierung ist umso wichtiger, da Updates, die bekannte Schwachstellen korrigieren, auf öffentlichen Blockchains gegebenenfalls erst nach längerer Zeit von allen Knoten eingespielt werden, sodass die Korrektur sehr langwierig sein kann.

Beim Betrieb von öffentlichen Blockchains ergeben sich im Laufe der Zeit häufig Gruppen herausgehobener Akteure. Beispiele hierfür sind bei Kryptowährungen die Tauschbörsen zum Umtausch in etablierte Fiatwährungen, z. B. Euro oder US-Dollar, sowie Mining-Pools als Zusammenschlüsse mehrerer Miner. In diesen Bereichen sind aus ökonomischen Gründen Zentralisierungstendenzen erkennbar (siehe Abschnitt 3.2.5). Den Tauschbörsen muss dabei vertraut werden, dass sie ihre Leistung tatsächlich erbringen [14] und ihre Infrastruktur gegen Angriffe schützen, für die sie herausgehobene Ziele darstellen.

Auf privaten Blockchains ähnelt das Vertrauensmodell dem klassischer zentralisierter Lösungen, ohne dass dabei die Eigenschaft der Dezentralität völlig aufgegeben wird. Aufgrund der abgestuften Rechteverwaltung müssen Nutzer den Inhabern höherer Positionen vertrauen, wobei verschiedene Arten von Fehlverhalten aufgrund der Sichtbarkeit der Blockchain für ihre Nutzer detektiert werden können. Weiterhin ist auf privaten Blockchains eine zentrale Administrationsstelle nötig, von der die Verwaltung der Rollen und Rechte durchgeführt wird. Auch wenn sie nicht direkt in den Betrieb der Blockchain eingreifen kann, könnte sie durch Vergabe oder Entzug von Rechten indirekt, aber nicht unbemerkt darauf Einfluss nehmen.

3.2 Konsensmechanismen

3.2.1 Problemstellung

Auf Blockchains muss die übereinstimmende Speicherung der Daten in der Blockchain auf den verschiedenen Knoten erreicht werden, wobei Einigkeit ebenfalls über die Ordnung der Blöcke und mindestens derjenigen Transaktionen, die logisch miteinander zusammenhängen, erzielt werden muss. Dies wird von einem Konsensmechanismus realisiert.

Die geschilderte Situation ist eine spezielle Ausprägung des sogenannten Konsensproblems aus dem Bereich der verteilten Systeme. Dabei müssen verschiedene Knoten, von denen einige fehlerhaft sein können, in einem Netzwerk Einigkeit über einen Wert erzielen. Die Knoten schlagen zu Beginn Werte vor, die unterschiedlich sein können. Mithilfe des Konsensmechanismus entscheiden sich alle korrekten, also nicht fehlerhaften, Knoten für einen der vorgeschlagenen Werte. Formal fordert man bei der Lösung des Konsensproblems die Erfüllung der folgenden Bedingungen (vgl. [15, S. 203-205]):

- Gültigkeit *(validity)*: Wenn ein Knoten sich für einen Wert entscheidet, wurde dieser von einem Knoten vorgeschlagen.
- Integrität *(integrity)*: Kein Knoten entscheidet sich mehrmals.
- Übereinstimmung *(agreement)*: Zwei korrekte Knoten treffen keine unterschiedlichen Entscheidungen.
- Beendigung *(termination)*: Jeder korrekte Knoten trifft letztendlich eine Entscheidung.

Erfüllt ein Algorithmus die ersten drei Bedingungen, so sagt man, dass er Sicherheit *(safety)* garantiert. Bei Erfüllung der vierten Bedingung stellt er Lebendigkeit *(liveness)* sicher. Informell wird dabei Sicherheit so definiert, dass nichts Schlechtes geschieht, und Lebendigkeit besagt, dass sich letztendlich etwas Gutes ereignet, nämlich dass das Verfahren terminiert.

Die tatsächliche Schwierigkeit des Problems ergibt sich aus den Nebenbedingungen, unter denen es gelöst werden muss. Die wichtigsten zwei Aspekte betreffen hierbei die Verbindungseigenschaften des Netzwerks sowie das fehlerhafte Verhalten einzelner Knoten.

Bezüglich des Netzwerks unterscheidet man die folgenden drei Modelle:

- Synchrones Netzwerk: Nachrichten zwischen zwei Knoten werden innerhalb einer bekannten Zeitspanne übermittelt.
- Asynchrones Netzwerk: Es existiert keine obere Schranke für die Zeit zur Übermittlung von Nachrichten.
- Partiell synchrones Netzwerk: Nachrichten werden innerhalb einer bestimmten, aber unbekannten Zeitspanne übermittelt oder das Netzwerk ist synchron, allerdings erst von einem unbekannten Zeitpunkt an.

Die beiden wichtigsten Fehlerarten sind:

- Crash-Fehler: Knoten stürzen ab und sind in der Regel nach einiger Zeit wieder funktionstüchtig.
- Byzantinische Fehler (vgl. Abbildung 6): Knoten können sich beliebig verhalten. Byzantinische Fehler können durch absichtlich böswilliges Verhalten eines Knotens herbeigeführt werden oder durch technisches oder menschliches Versagen auftreten.

Algorithmen, die eine gewisse Anzahl von Crash-Fehlern bzw. byzantinischen Fehlern tolerieren, werden als CFT-Algorithmen *(crash fault-tolerant)* bzw. BFT-Algorithmen *(Byzantine fault-tolerant)* bezeichnet.

Das Konsensproblem ist im Bereich der verteilten Systeme bereits seit den 1980er-Jahren intensiv und mit gutem Erfolg erforscht worden [15]. Wichtige Resultate betreffen allgemeine Aussagen, unter welchen Bedingungen das Problem grundsätzlich lösbar oder nicht lösbar ist [16], [17].

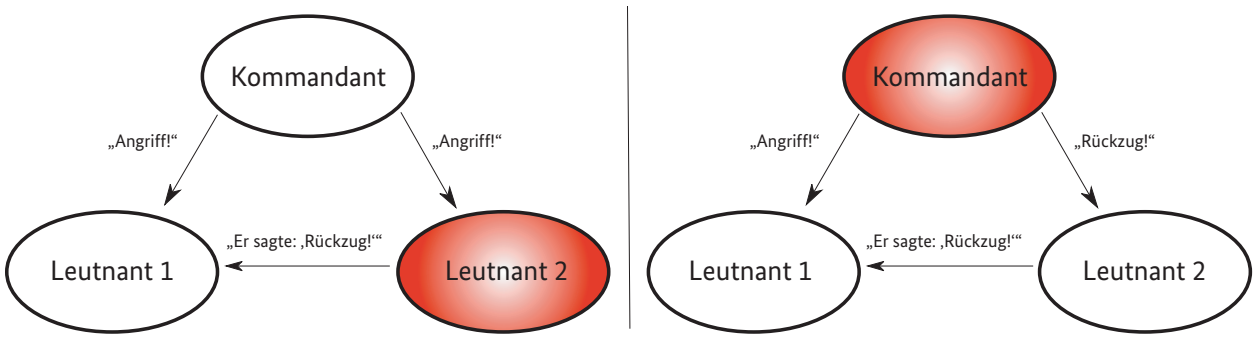

Abbildung 6: Das Problem der Byzantinischen Generale gab dem byzantinischen Fehler seinen Namen. Aus Sicht von Leutnant 1 ist nicht zu unterscheiden, welcher der beiden anderen Kommunikationspartner sich unehrlich verhält.

Auch konkrete Algorithmen, die das Problem nachweislich lösen, sind entwickelt worden und werden in sehr großem Umfang eingesetzt. Ein wichtiges Ergebnis zur Lösbarkeit besagt, dass das Konsensproblem in einem asynchronen Netzwerk nicht durch einen deterministischen Algorithmus lösbar ist, selbst wenn nur ein einzelner Crash-Fehler vorliegt [18]. Daher muss ein Konsensmechanismus, da er das Auftreten von Fehlern nicht verhindern kann, in Phasen, in denen das Netzwerk asynchron ist, eine der Eigenschaften Sicherheit und Lebendigkeit aufgeben. Weitere Aussagen betreffen die Anzahl an Knoten, die nötig ist, um eine gewisse Zahl von fehlerhaften Knoten tolerieren zu können.

Neben der Aussage, unter welchen Bedingungen ein Algorithmus das Konsensproblem lösen kann und welche Sicherheitsgarantien er bietet, ist für den großflächigen praktischen Einsatz die Effizienz von größter Relevanz. Wichtig ist hierbei der Durchsatz, den der Algorithmus erlaubt. Der Durchsatz hängt stets vom Verhalten des Netzwerks und der Knoten ab, zusätzlich ergibt sich aber teils eine starke Abhängigkeit von der Anzahl an Knoten, die am Konsensmechanismus teilnehmen. Falls Konsensmechanismen schlecht auf eine größere Knotenzahl skalieren, so ist ihr Einsatz in genehmigungsfreien Blockchains nicht sinnvoll.

3.2.2 Nachrichtenbasierte Algorithmen

Im Laufe der Jahre wurde eine Reihe nachrichtenbasierter Algorithmen entwickelt, die das Konsensproblem unter verschiedenen Bedingungen lösen. Nachrichtenbasiert bedeutet, dass der Konsens hergestellt wird, indem die Knoten untereinander gemäß einem vorgegebenen Abstimmungsprotokoll Nachrichten austauschen. Dabei übernimmt ein Knoten als sogenannter Anführer *(leader, primary)* eine zentrale Stellung, die übrigen Knoten verhalten sich passiv. Im Fall von Abstürzen (CFT-Verfahren) bzw. Fehlverhalten (BFT-Verfahren) des Anführers wird ein neuer Anführer gewählt.

Es wird ein asynchrones Netzwerk angenommen, um maximale Robustheit gegenüber ungünstigen Bedingungen zu erzielen. Zwar ist das Konsensproblem in einem asynchronen Netzwerk grundsätzlich nicht lösbar. Daher garantieren die Algorithmen zunächst nur die Sicherheit, also insbesondere die Übereinstimmung der Daten auf korrekten Knoten, nicht aber die Lebendigkeit. In der Praxis sind jedoch die asynchronen Phasen in einem Netzwerk zeitlich beschränkt, da Probleme nach einer gewissen Zeitspanne behoben werden, sodass de facto ein partiell synchrones Netzwerk vorliegt.

Da sie die gleichen Annahmen zum Netzwerk treffen, unterscheiden sich die praktisch relevanten Algorithmen aus dem Bereich der verteilten Systeme vor allem in ihrem Fehlermodell. So gibt es sowohl CFT- als auch BFT-Algorithmen, die im jeweiligen Bereich als Quasi-Standards oder zumindest als Grundlage von Weiterentwicklungen gelten.

Im Bereich der Crash-Fehler sind die Algorithmen Paxos und Raft [19], [20] am bekanntesten. Gemäß allgemeinen theoretischen Resultaten können in diesem Fehlermodell die Sicherheitseigenschaften

für die korrekten Knoten weiter gültig bleiben, wenn weniger als die Hälfte der Knoten abstürzt [16]. Beide Algorithmen erreichen diese theoretische Schranke. Wie erwähnt garantieren die Algorithmen zu jedem Zeitpunkt die Sicherheit. Lebendigkeit ist gegeben, sobald das Netzwerk ausreichend lange synchron ist. Das Nachrichtenaufkommen zur Herstellung von Konsens wächst linear mit der Anzahl der daran teilnehmenden Knoten, weshalb die Algorithmen nicht hin zu einer großen Knotenzahl skaliert werden können. CFT-Verfahren wie Paxos und Raft erlauben einen Durchsatz von etwa 50.000 tps, wenn drei Fehler toleriert werden sollen [11].

Im Bereich der byzantinischen Fehler wird der Algorithmus PBFT (Practical Byzantine Fault Tolerance) [21] als grundlegend angesehen. Die Garantien bezüglich Sicherheit und Lebendigkeit sind die gleichen wie bei Paxos und Raft und bleiben erfüllt, solange weniger als ein Drittel der Knoten byzantinischen Fehlern unterliegen, was der theoretischen Schranke in dieser Situation entspricht. Da ein byzantinischer Fehler bedeutend weitreichender ist als ein einfacher Crash-Fehler, sind deutlich mehr Nachrichten erforderlich, um die Sicherheitsgarantien erfüllen zu können. Das Nachrichtenaufkommen wächst in diesem Fall quadratisch. Aus diesem Grund ist die Skalierbarkeit noch schlechter als bei Paxos und Raft. PBFT und Weiterentwicklungen erlauben einen Durchsatz von etwa 20.000 tps, wenn drei Fehler korrigiert werden sollen [11]. Neben der Steigerung des Durchsatzes erlauben die Weiterentwicklungen es insbesondere, bei Auftreten von Fehlern den Durchsatz relativ stabil zu halten. Der ursprüngliche PBFT-Algorithmus kann zwar byzantinische Fehler tolerieren, der Durchsatz sinkt in diesem Fall jedoch sehr stark ab, was für die Praxis ungenügend ist.

In der Praxis werden in verteilten Systemen wegen ihres höheren Durchsatzes fast immer CFT-Algorithmen eingesetzt. Weiterhin wird das Auftreten byzantinischer Fehler in klassischen Umgebungen, in denen Konsensmechanismen zur redundanten Datenhaltung verwendet werden und alle Knoten unter Kontrolle desselben Betreibers stehen, als sehr unwahrscheinlich angesehen. Da diese Eigenschaft auf Blockchains nicht mehr gegeben ist, muss kritisch abgewogen werden, ob die Resistenz gegen byzantinische Fehler oder der Durchsatz wichtiger sind. Dabei sollte man sich bewusst sein, dass bei Verwendung von CFT-Algorithmen einzelne Knoten den Konsensmechanismus gegebenenfalls gezielt sabotieren können. Aufgrund der schlechten Skalierbarkeit sind nachrichtenbasierte CFT- und BFT-Algorithmen für den Einsatz auf genehmigungsfreien Blockchains nicht geeignet. Ein weiterer Grund liegt darin, dass in nachrichtenbasierten Verfahren die Identität der Knoten verifiziert werden muss, damit kein Teilnehmer im Abstimmungsprotokoll mehrere Stimmen erlangen kann. Auf privaten Blockchains werden diese Verfahren jedoch häufig eingesetzt [22].

3.2.3 Nachweisbasierte Algorithmen

Mit dem Aufkommen von Blockchains wurden zahlreiche neue Algorithmen zur Lösung des Konsensproblems vorgeschlagen. Anders als die nachrichtenbasierten Algorithmen skalieren sie im Allgemeinen sehr gut auf große Knotenzahlen und können auch auf öffentlichen genehmigungsfreien Blockchains eingesetzt werden. Den Algorithmen ist gemein, dass der Nachweis *(proof)* über eine gewisse Ressource bei der Herstellung von Konsens fundamental ist. Auf Englisch wird zuweilen zusammenfassend der Begriff Proof-of-X (PoX) verwendet.

Das bekannteste Beispiel ist der unter anderem von Bitcoin und Ethereum verwendete Algorithmus Proof-of-Work (PoW) [2]. Die Herstellung von Konsens über neue Daten geschieht bei diesem Algorithmus pro Block mithilfe eines rechenintensiven mathematischen Rätsels (Auffinden eines Strings, dessen Hashwert unterhalb einer gegebenen Schranke liegt). Die Lösung dieses Rätsels ist der eigentliche „Proof-of-Work“ (Arbeitsnachweis). Im Wesentlichen wird der von dem Knoten, der als Erster eine gültige Lösung findet, vorgeschlagene Block danach im Netzwerk verteilt und von den übrigen Knoten akzeptiert. Durch Latenzzeiten bei der Übertragung der Nachrichten im Netzwerk kann es vorkommen, dass verschiedene Knoten zunächst in ihrer lokalen Blockchain-Kopie an derselben Stelle unterschiedliche Blöcke speichern, wodurch unter-

schiedliche Zweige der Blockchain entstehen und es somit zu Inkonsistenzen kommt. Dies wird als *Fork* (Gabelung) bezeichnet. Solche Forks werden jedoch probabilistisch nach kurzer Zeit aufgelöst.

Abstrakt lässt sich PoW als Konsensmechanismus mit einem Anführer auffassen, dessen Vorschlag im Regelfall (sofern keine Forks auftreten) von allen korrekten Knoten übernommen wird. Die Berechnung des Proof-of-Work realisiert dabei die Wahl des Anführers, der nach einer zufälligen Zeitspanne (die Zeit, die er zur Berechnung einer Lösung benötigt) den von ihm vorgeschlagenen Block verbreiten kann. Im Gegensatz zu CFT- und BFT-Verfahren ist bei PoW stets Lebendigkeit gegeben. In asynchronen Phasen gibt es aber keinerlei Sicherheitsgarantien. Wegen der Latenzzeit ist die Sicherheit selbst in synchronen Phasen nicht garantiert und es kann wie beschrieben durch Forks vorübergehend zu Inkonsistenzen kommen. Man spricht hier von letztendlicher Konsistenz *(eventual consistency)*. In diesem Zusammenhang wird auch der Begriff der *Finalität* von Blöcken oder Transaktionen genutzt: Bei PoW können der Blockchain hinzugefügte Blöcke nachträglich durch Auflösen von Forks ungültig werden. Dies ist bei den vorgestellten CFT- und BFT-Verfahren nicht möglich, da sie immer die Sicherheit garantieren und die Blöcke daher final sind. Der mittels PoW mögliche Durchsatz ist vergleichsweise gering. So erreicht Bitcoin etwa 7 tps, Ethereum etwa 15 tps (Stand März 2019). Der wesentliche Grund dafür ist, dass aufgrund der Latenzzeit des Netzwerks der zeitliche Abstand zwischen verschiedenen Blöcken, also die durchschnittliche Wartezeit, bis das Rätsel zum ersten Mal gelöst wird, nicht beliebig klein gewählt werden kann, da ansonsten Forks gehäuft auftreten und kaum noch aufgelöst werden können.

Die Zeitspanne, bis ein Knoten das Rätsel gelöst hat, hängt stark von der Rechenleistung der von ihm kontrollierten Hardware ab. PoW kann langfristige Angriffe auf die Konsistenz einer Blockchain verhindern, wenn sich weniger als 50 % der Rechenleistung des Netzwerks unter Kontrolle eines (byzantinischen) Angreifers befindet (für Details siehe Abschnitt 7.1.2). Das größte Problem von PoW ist der immense Energieverbrauch (siehe Abschnitt 2.2). Da über den Energieverbrauch und dessen Kosten die Resistenz gegen Angriffe sichergestellt wird, werden korrekte Knoten durch ein Anreizsystem (siehe Abschnitt 3.2.5) dazu gebracht, am PoW teilzunehmen. Im Fall von Kryptowährungen kann der Energieverbrauch aus ökonomischen Gründen als proportional zu ihrem Kurswert angesehen werden.

Im Anschluss an PoW wurden weitere Konsensmechanismen für öffentliche genehmigungsfreie Blockchains vorgeschlagen. Ihr Ziel ist vor allem die Verringerung des Energiebedarfs und die Erhöhung des Durchsatzes.

Insbesondere ist hier der Algorithmus Proof-of-Stake (PoS) [23] zu nennen, zu dem in Ethereum seit längerer Zeit ein Übergang vorgesehen ist, der aber bisher (Stand März 2019) nicht umgesetzt wurde. Dabei ist die Fähigkeit zur Erstellung neuer Blöcke nicht an die beigesteuerte Rechenleistung, sondern an das Guthaben in der zugrunde liegenden Kryptowährung gekoppelt. Wer als sogenannter Validator an der Erstellung neuer Blöcke mitwirken möchte, muss einen Teil seines Vermögens als Pfand hinterlegen. Für jeden Block werden je nach Umsetzung einer oder mehrere Validatoren zufällig ausgewählt, um diesen vorzuschlagen. Die Wahrscheinlichkeit, ausgewählt zu werden, ist dabei abhängig von der Größe des hinterlegten Pfandes. Sofern mehrere Validatoren an der Blockerstellung beteiligt sind, stellen diese nun untereinander Konsens (z. B. mit einem BFT-Verfahren) darüber her, welcher Block der Blockchain hinzugefügt werden soll. Danach wird dieser Block auch von den übrigen Knoten übernommen. Durch die Implementierung gewisser Regeln sollen die Pfänder von Validatoren, die sich missbräuchlich verhalten, zerstört werden. Dadurch sollen mehrere Formen von Angriffen für die Angreifer bedeutend teurer und damit unwahrscheinlicher werden. Der Energiebedarf ist im Vergleich zu PoW als vernachlässigbar anzusehen. Zu beachten ist, dass PoS momentan nicht großflächig eingesetzt wird und seine Eigenschaften, insbesondere was die Erreichung von Sicherheit und Lebendigkeit angeht, stark von der Umsetzung abhängen. Insofern sind hier allgemeine Aussagen schwer zu treffen.

In Anlehnung an die abstrakte Formalisierung von PoW wird auf einigen Blockchains der Konsensmechanismus Proof-of-Elapsed-Time (PoET)

genutzt, der die zufällige Wartezeit mithilfe eines speziellen Hardware-Sicherheitsmoduls realisiert [22], [24]. Diese Lösung hat gegenüber PoW ebenfalls den Vorteil, dass der Energiebedarf vernachlässigbar ist. Gleichzeitig ist ein deutlich höherer Durchsatz möglich. Die Eigenschaften bezüglich Lebendigkeit und Sicherheit sind im Wesentlichen die gleichen wie bei PoW. Andererseits ist in diesem Fall Vertrauen in die korrekte Funktionalität des Sicherheitsmoduls erforderlich.

Eine grundsätzliche Eigenschaft der Verfahren PoW, PoS und PoET besteht konstruktionsbedingt darin, dass einzelne Knoten durch ökonomische Investitionen ihren Einfluss auf den Konsensmechanismus steigern und diesen in Extremfällen sogar in ihrem Sinne manipulieren können.

Weiterhin sind auf genehmigungsfreien Blockchains Alternativen vorgeschlagen worden, die auf Komitees basieren [25], [26], [27]. Grob gesagt wird dabei pro Block eine kleine Gruppe von Knoten ausgewählt, die untereinander Konsens herstellen, der anschließend von den übrigen Knoten akzeptiert wird. Die Herstellung von Konsens geschieht gewöhnlich mittels nachrichtenbasierter CFT- oder BFT-Verfahren. Die Auswahl des Komitees kann auf verschiedene Arten geschehen (z. B. kann man auch PoS mit mehreren Validatoren als Komiteeverfahren auffassen). Entscheidend ist, dass der Auswahlprozess vor Manipulationen geschützt ist. Von Komiteeverfahren verspricht man sich ebenfalls im Vergleich zum klassischen PoW einen sehr hohen Durchsatz bei vernachlässigbarem Energieaufwand. Es muss aber darauf hingewiesen werden, dass die Sicherheit dieser Verfahren auf dem Zusammenspiel verschiedener komplexer Algorithmen basiert, die in der Regel nicht gut untersucht sind, und daher momentan nicht beurteilt werden kann.

Tabelle 2 gibt eine Übersicht über die Eigenschaften der besprochenen Verfahren.

	Paxos/Raft	PBFT	PoW	PoS	PoET
Fehlertoleranz	CFT	BFT	BFT	vermutlich BFT	BFT
Sicherheit	ja	ja	letztendliche Konsistenz bei Synchronizität	abhängig von Umsetzung	letztendliche Konsistenz bei Synchronizität
Lebendigkeit	bei Synchronizität	bei Synchronizität	ja	abhängig von Umsetzung	ja
Authentisierung der Knoten nötig	ja	ja	nein	nein	nein
Durchsatz	sehr hoch	hoch	niedrig	mittel/hoch*	mittel
Energieverbrauch	sehr gering	sehr gering	sehr hoch	sehr gering	sehr gering
Skalierbarkeit auf große Knotenzahl	schlecht	schlecht	gut	gut	mittel
Formale Sicherheitsbeweise	ja	ja	nein†	nein	nein

Tabelle 2: Vergleich von Konsensmechanismen

* Zuverlässige Benchmarks aus tatsächlichen praktischen Anwendungen liegen nicht vor, vgl. aber [28, S. 5].

† Es gibt verschiedene Formalisierungen und Sicherheitsbeweise für PoW im Kontext von Bitcoin, die jedoch unrealistische Annahmen treffen oder gewisse Teilaspekte ausklammern [29].

3.2.4 Sicherheitsaussagen

Die Sicherheitsgarantien, die ein Konsensmechanismus bietet, stellen fundamentale Eigenschaften einer Blockchain dar. Bei der Wahl eines Algorithmus sollte daher sehr sorgfältig geprüft werden, ob er tatsächlich die angegebenen Garantien aufweist. Dazu gehören eine sorgfältige Formalisierung des Problems inklusive der Annahmen zum Netzwerkverhalten und den Fehlerarten.

Zudem sollten Sicherheitsaussagen immer durch eine breite Diskussion und Peer-Reviews durch Experten geprüft und abgesichert werden. Ein formaler Nachweis des Sicherheitsniveaus wäre wünschenswert, ist aber insbesondere bei Verfahren mit spieltheoretischen Komponenten schwierig (siehe auch Abschnitt 5.5). Grundsätzlich erfordern nachvollziehbare Sicherheitsgarantien für Blockchains eine unabhängige Evaluierung und Zertifizierung der eingesetzten Verfahren nach vorher festgelegten Standards und Prüfkriterien (siehe Abschnitt 12.2).

Bei den nachrichtenbasierten CFT- und BFT-Algorithmen enthalten insbesondere die Veröffentlichungen von Paxos, Raft und PBFT detaillierte Sicherheitsbeweise. Bezüglich der nachweisbasierten Algorithmen ist dieses Vorgehen in fast allen Fällen nicht gegeben. Neue Algorithmen wurden und werden in Whitepapern publiziert, die häufig weder formale Aussagen zu dem von ihnen gelösten Problem noch eine formale Beweisführung enthalten. Sind solche Aussagen vorhanden, so werden sie äußerst selten durch wissenschaftliche Reviews von Experten abgesichert und sollten daher zunächst als Behauptungen angesehen werden. In einer Reihe von Fällen haben sich behauptete Sicherheitsgarantien in der von ihren Entwicklern angegebenen Form als falsch herausgestellt [30], [31], [32], [33]. Ein weiterer negativer Faktor ist hier auch die unzureichende technische Dokumentation vieler neuer Vorschläge, die eine Beurteilung der Sicherheitseigenschaften noch schwieriger macht.

Das fundamentale Problem in diesem Bereich ist das folgende: Es ist vergleichsweise einfach und daher verlockend, einen neuen Algorithmus zu konzipieren, der unter günstigen Bedingungen bezüglich des Netzwerks und der auftretenden Fehler gut funktioniert. Ungleich schwieriger ist es aber, einen Algorithmus so zu konzipieren, dass er auch unter sehr ungünstigen Bedingungen und (im Fall byzantinischer Fehler) potenziell unvorhersehbaren Ereignissen seine Sicherheitsgarantien erhält (vgl. [22]).

3.2.5 Anreizsysteme

Da der Konsensmechanismus PoW sehr energieintensiv ist und eine Teilnahme dementsprechend hohe Kosten verursacht, ist ein sogenanntes Anreizsystem einer seiner integralen Bestandteile. Grundidee des Anreizsystems ist es, die am PoW teilnehmenden Knoten für erbrachte Leistung ökonomisch zu belohnen. Genauer erhält derjenige Knoten, der für einen Block als erster das mathematische Rätsel löst, einen bestimmten Betrag in der zugrunde liegenden Kryptowährung gutgeschrieben. Abhängig von deren Kurswert, der Rechenleistung des übrigen Netzwerks und den lokalen Stromkosten erhält ein Knoten somit einen ökonomischen Anreiz zur Teilnahme am PoW. Je höher der Kurswert der Kryptowährung, umso höher wird damit die Gesamtrechenleistung des Netzwerks und damit grundsätzlich der Schutz der Blockchain vor langfristigen Angriffen auf Ebene des Konsensmechanismus, auch wenn die Risiken durch eine starke Zentralisierung des Minings weiter bestehen (siehe Abschnitt 7.1.2).

Es ist anzumerken, dass das Anreizsystem alleine nicht notwendigerweise das Verhalten der Miner wie beabsichtigt beeinflusst (siehe Abschnitt 5.5). So ist es durchaus denkbar, dass einzelne Akteure zur Erreichung anderer Ziele gegen ihre ökonomischen Interessen innerhalb des Systems handeln (siehe Abschnitt 7.1.2). Weiterhin wurde z. B. für Bitcoin gezeigt, dass, selbst wenn die Annahmen zutreffen, die Anreize bereits dann, wenn ein Akteur deutlich weniger als 50 % der Rechenleistung des Netzwerks kontrolliert, nicht mehr in der gewünschten Weise wirken (siehe Abschnitt 7.1.2). Ganz grundsätzlich ist auch zu berücksichtigen, dass die korrekte und langfristige Schaffung von ökonomischen Anreizen für intendiertes Verhalten ein notorisch schwieriges

Problem ist, siehe z. B. [34], [35]. Auch bei Bitcoin sorgt das Anreizsystem zwar dafür, dass eine Kontrolle der Mehrheit der Rechenleistung und somit praktische Angriffe extrem teuer werden, hat gleichzeitig aber den Nebeneffekt des immensen Energiebedarfs und einer starken Zentralisierung des Minings, die den ursprünglichen Zielen von Nakamoto [2] diametral entgegengesetzt ist.

Zusammenfassung.

- Vertrauen in bestimmte Akteure bleibt weiter notwendig.
- Bei der Wahl eines Konsensmechanismus sollte das Netzwerkverhalten sowie die Fehlertoleranz sorgfältig modelliert werden.
- Die Frage, ob bei der Konsensbildung Sicherheit oder Lebendigkeit priorisiert wird, ist grundlegend.
- Etablierte Konsensmechanismen für öffentliche und private Blockchains unterscheiden sich in vielen Punkten.
- Sicherheitsaussagen sind in vielen Fällen unpräzise und nicht belastbar.
- Es ist möglich, dass ökonomische Anreize nicht wie beabsichtigt wirken.

4 Smart Contracts

Der Begriff *Smart Contract* wurde 1994 von Nick Szabo eingeführt für „rechnergestützte Transaktionsprotokolle, die Vertragsbestimmungen ausführen“ [36] und mit Aufkommen der Ethereum-Blockchain wieder aufgegriffen. Entwickler anderer Plattformen verwenden zum Teil andere Namen und Begriffe für ausführbare Programme auf ihrer Blockchain. Bei Hyperledger Fabric etwa kennt man sie unter dem Namen *chaincode* oder etwas weiter gefasst als *distributed applications* (dAPPs). Im Folgenden wird zur sprachlichen Vereinfachung einheitlich der Begriff *Smart Contract* oder einfach *Contract* benutzt, die Konzepte anderer Plattformen sind aber ausdrücklich in die Diskussion mit eingeschlossen. Obwohl der Begriff es suggerieren mag, sei darauf hingewiesen, dass Smart Contracts keine Verträge in streng rechtlichem Verständnis sind (siehe auch Abschnitt 8.4).

Im Blockchain-Kontext versteht man unter einem Smart Contract meist ein ausführbares Programm, dessen Code in einer Transaktion an die Blockchain geschickt und von den Netzwerkknoten im Rahmen der Validierung ausgeführt wird. Aufrufe von Contracts und Eingabe von Input erfolgen in der Regel ebenfalls über Transaktionen. Die Unveränderlichkeit der Blockchain verhindert nachträgliche Änderungen am Programmcode, und durch den automatisch ablaufenden Validierungsprozess wird verhindert, dass die Ausführung des Contracts unterbunden werden kann.

4.1 Beispiele

Das Kapitel beginnt mit der Vorstellung bekannter Beispiele, um die Vielfalt der Umsetzungsmöglichkeiten zu demonstrieren, welche im Anschluss systematischer betrachtet werden sollen.

Beispiel: Ethereum

Ein Smart Contract von Ethereum (siehe Abschnitt 1.3) wird in der Regel in der Programmiersprache Solidity geschrieben. Der compilierte Bytecode wird als eigenständige Transaktion ohne Angabe einer Empfängeradresse an das Ethereum-Netzwerk geschickt. Beim Mining wird dem Contract eine neu erzeugte Adresse zugeordnet und der Programmcode in der Blockchain veröffentlicht (z. B. [39]).

Spätere Transaktionen an Contract-Adressen bewirken, dass der entsprechende Contract ausgeführt werden muss – einmal vom Miner bei Aufnahme der Transaktion in einen Block und anschließend von jedem anderen Knoten bei dessen Verifikation. Dies ist ein Nadelöhr für Effizienz und Durchsatz der Ethereum-Blockchain, insbesondere wenn einzelne Contracts einen sehr hohen Rechenaufwand haben. Die Programmausführung erfolgt jeweils lokal in der EVM (Ethereum Virtual Machine) des betreffenden Knotens und erfordert die Bereitstellung entsprechender Ressourcen.

Um ein Steuerelement bereitzustellen, das das Aufrufen von allzu rechenintensiven Contracts ökonomisch unattraktiv macht und gleichzeitig die Miner für ihren geleisteten Aufwand entlohnt, verlangt Ethereum für jede durchgeführte Operation eine Gebühr in der eigens hierfür eingeführten internen Währung *Gas* [5]. Wer einen Contract initiiert oder ihn später aufrufen möchte, legt eine Maximalgebühr *(gasLimit)* fest, die er zu zahlen bereit ist, ferner einen Wechselkurs *(gasPrice)* zwischen Gas und Ether. Derjenige Miner, der die Transaktion erfolgreich in einem Block unterbringt, bekommt die aufgewendete Gas-Menge zum vereinbarten Wechselkurs ausbezahlt. Über diesen Wechselkurs kann also indirekt die Höhe der Belohnung für den Miner gesteuert werden (siehe aber Abschnitt 7.2.2). Überschreiten die tatsächlichen Kosten die festgelegte Maximalgebühr, wird die Ausführung abgebrochen und die Gebühr einbehalten.

Beispiel: Hyperledger Fabric

Für die Programmierung von Smart Contracts in Hyperledger Fabric (siehe Abschnitt 1.3) sollen beliebige universelle Programmiersprachen eingesetzt werden (derzeit werden nur Go, Java und Node.js unterstützt, Stand März 2019), um auf der vorhandenen Programmiererfahrung der Entwickler aufbauen zu können.

Ein wesentlicher Unterschied zu Ethereum besteht darin, dass bei Hyperledger Fabric die Konsensfindung losgelöst ist von der Ausführung der Smart Contracts, sowohl was den zeitlichen Ablauf als auch die verantwortlichen Knoten betrifft.

Jeder Contract betraut lediglich eine Teilmenge aller Netzwerkknoten mit seiner Ausführung. Von diesen simuliert jeder unabhängig voneinander zunächst lokal die Ausführung des Contracts und meldet das Ergebnis an den Aufrufer zurück, ohne es in der Blockchain zu verankern. Ist eine ausreichende Menge an übereinstimmenden Rückmeldungen eingegangen, wird eine Transaktion aufgesetzt und an eine andere Gruppe von Knoten, den *ordering service*, geschickt. Diese entscheidet mittels eines Konsensmechanismus über die Reihenfolge, in der die eingegangenen Transaktionen in die Blockchain kommen, ohne eine Validierung oder inhaltliche Bewertung der Transaktionen vorzunehmen. Diese Validierung übernehmen erst am Schluss alle Knoten, wenn sie ihre lokale Kopie der Blockchain aktualisieren. Hier werden beispielsweise solche Transaktionen nachträglich aussortiert, bei denen aufgrund von früher ausgeführten Contracts die Ausgangsbedingungen nicht mehr denen der Simulation entsprechen. Entscheidend ist aber, dass dieser letzte Schritt vollkommen deterministisch ist und keinen Fork erzeugen kann.

Von diesem Verfahrensweg verspricht Hyperledger Fabric sich eine geringere Redundanz und höheren Durchsatz, da nicht mehr alle Knoten die Contracts ausführen müssen.

Beispiel: Bitcoin

Bereits Bitcoin (siehe Abschnitt 1.3) stellt die Möglichkeit zur Erstellung kleinerer Skripte bereit. Dazu steht eine stackbasierte Skriptsprache mit eingeschränktem Befehlsumfang zur Verfügung [37]. Die hiermit erstellten Programme erlauben es, die Anforderungen an die Signaturen den eigenen Bedürfnissen anzupassen, und werden von den Minern bei der Signaturprüfung automatisch ausgewertet. Dazu wird lediglich ein Stack benötigt, dessen Größe zwar nicht explizit beschränkt ist, aber aufgrund der Maximalgröße des Skriptes de facto nicht beliebig groß werden kann [38].

Weitere Beispiele demonstrieren, welche Herausforderungen andere Smart-Contract-Plattformen zu lösen versuchen. Die Aufzählung ist nicht repräsentativ und vor allem nicht vollständig.

Quorum [40] ist eine genehmigungsbasierte Weiterentwicklung von Ethereum, die sowohl öffentliche als auch private Smart Contracts anbietet. Bei letzteren werden Transaktionen, die vertraulich behandelt werden sollen, verschlüsselt. Folglich können nur die Knoten, die den Entschlüsselungsschlüssel kennen, die Validierung dieser Contracts vornehmen, die anderen müssen sie überspringen. Es kann somit zu Diskrepanzen in der Berechnung des Systemzustands kommen. Um dieses Problem zu lösen, gibt es in Quorum zusätzlich zu dem gewöhnlichen öffentlichen Systemstatus, der von allen Knoten geteilt wird, noch einen privaten, den jeder Knoten individuell anlegt und in dem er diejenigen privaten Contracts berücksichtigt, an deren Validierung er beteiligt war.

Zu weiteren Vorschlägen für Vertraulichkeit in Smart Contracts siehe auch Abschnitt 5.1.

Ripple bot eine Zeit lang den Service Codius [41] für Smart Contracts an, der sich der Problematik der Interoperabilität (siehe auch Abschnitt 11.3) annimmt. Codius-Transaktionen sollen Smart Contracts auf unterschiedlichen Blockchain-Plattformen aufrufen können. Das Fehlen

von Standards für blockchainübergreifende Bezahlvorgänge bereitete vorübergehend Probleme, zu deren Behebung das Interledger Protocol [42] entwickelt wurde.

Bei Nxt/Ardor [43] wird die eigentliche Contract-Logik außerhalb der Blockchain auf einer API ausgeführt, und nur die Befehle, die spezifische Blockchain-Operationen betreffen – z. B. Transfer von Guthaben –, werden als Transaktion an die Blockchain weitergeleitet. Dieses Vorgehen ermöglicht eine nachträgliche Änderung von Programmcode, da dieser selbst nicht auf der Blockchain liegt. Der Grundsatz der Unveränderlichkeit wird also zugunsten der Möglichkeit zur Fehlerkorrektur aufgegeben. Das Vertrauensmodell (siehe Abschnitt 3.1) ändert sich entsprechend.

4.2 Vergleich verschiedener Ansätze

Obige Beispiele machen deutlich, dass es signifikante Unterschiede zwischen den existierenden Smart-Contract-Systemen gibt. Die folgende Übersicht soll einige maßgebliche Eigenschaften näher untersuchen, um Nutzern einen Überblick zu geben, was bei der Auswahl einer geeigneten Plattform zu beachten ist.

4.2.1 Programmiersprache

Die Programmiersprache, in der ein Contract geschrieben wird, hat Einfluss auf die mögliche Komplexität sowie auf die Fehleranfälligkeit bzw. Angreifbarkeit des Codes. Auf einigen Systemen kommen Skriptsprachen mit begrenztem Umfang zum Einsatz, mit denen sich nur in geringem Maße Programme erstellen lassen. Einschränkungen können die Skriptgröße betreffen – sowohl hinsichtlich der Bytezahl als auch der Anzahl der auszuführenden Befehle – sowie auch den zur Verfügung stehenden Befehlssatz. Mit einer Sprache, die beispielsweise die Programmierung von Schleifen ermöglicht, besteht die Gefahr, dass ein Angreifer durch gezieltes Einbringen einer Endlosschleife das gesamte System zum Erliegen bringt. Aus Sicherheitsgründen verzichten manche in Blockchains eingesetzte Skriptsprachen deshalb von vornherein auf die Implementierung missbrauchsanfälliger Sprachelemente wie Schleifen, rückwärtsgerichteter Sprünge und Random Write Access auf den Stack. Einschränkungen in der Funktionalität werden bewusst in Kauf genommen zugunsten einer besseren Überprüfbarkeit der Korrektheit der erstellten Skripte.

Andere Anbieter von Smart-Contract-Plattformen setzen dagegen auf universelle Programmierbarkeit und ermöglichen den Einsatz Turing-vollständiger Programmiersprachen. Zur Anwendung kommen zum Teil Neuentwicklungen, die oft noch Schwachstellen und Anfälligkeiten für Programmierfehler aufweisen, zum Teil aber auch ausgereifte Programmiersprachen. Mit diesen sind zwar nun beliebig komplexe Smart Contracts möglich, die Korrektheit der Programme kann dafür nicht mehr bewiesen werden. Schwachstellen wie die oben geschilderten Endlosschleifen müssen auf andere Weise verhindert werden (z. B. durch Einführung des Gas-Limits bei Ethereum, siehe Abschnitt 4.1).

4.2.2 Reihenfolge und ausführende Knoten

Ein Smart Contract wird in der Blockchain verankert, wenn durch eine entsprechende Transaktion die Initialisierung bzw. ein Programmaufruf veranlasst wird. Während der Konsensfindung werden bisher in den meisten Blockchain-Systemen die Blöcke und Transaktionen von den Minern validiert. Zur Validierung eines Blocks gehört die Ausführung aller darin enthaltenen Smart Contracts und gegebenenfalls die Aktualisierung des Systemzustands. Da die Validierung von allen Minern durchgeführt wird, ist offensichtlich, dass dies einen großen Aufwand erzeugt, insbesondere auf genehmigungsfreien Blockchains, bei denen beliebig viele Knoten an der Konsensfindung teilnehmen können.

Neuere Ansätze entkoppeln Konsens und Ausführung und bringen nur solche Smart Contracts in den Konsensprozess, die vorher von einer bestimmten Teilmenge der Knoten erfolgreich validiert werden konnten (siehe Abschnitt 4.1). Die am Konsens beteiligten Knoten werden nur

über die erfolgreiche Validierung informiert und brauchen diese nicht mehr zu überprüfen. Dieser Ansatz setzt eine genehmigungsbasierte Blockchain voraus, da eine Gruppe von Validatoren ausgewählt und ihr besonderes Vertrauen entgegengebracht werden muss.

4.2.3 Laufzeitumgebung

Für die Ausführung von Bitcoin-Skripten muss lediglich ein Stack zur Verfügung stehen, andere Systeme, deren Smart Contracts komplexere Programme sein können, benötigen eine passende Laufzeitumgebung. Die Miner führen während des Validierungsprozesses auf ihren eigenen Rechnern Smart Contracts aus, die von potenziell nicht vertrauenswürdigen Entwicklern geschrieben wurden. Es ist daher nötig, einerseits Plattformunabhängigkeit und andererseits Sicherheit für den Host-Rechner zu gewährleisten. Zu diesem Zweck werden abgesicherte Umgebungen wie beispielsweise virtuelle Maschinen oder Docker-Container eingesetzt.

4.2.4 Anreiz und Währung

Die Bitcoin-Skripte sind nur sehr kurze Programme, deren Ausführung keine nennenswerten Ressourcen verbraucht. Smart Contracts, die komplexeren Code enthalten, stehen vor dem Problem, dass jeder Knoten, der den Contract im Validierungsprozess ausführt, dafür Zeit und Rechenleistung investieren muss.

In genehmigungsfreien Blockchain-Systemen wird daher in aller Regel nur ein ökonomischer Anreiz die Knoten motivieren, einen ressourcenintensiven Validierungsprozess regelkonform durchzuführen. Ein Bezahlsystem innerhalb der Blockchain lässt sich realisieren, wenn wie bei Bitcoin und Ethereum ohnehin eine intrinsische Währung vorhanden ist. Anderenfalls könnte eine eigene Token-Währung speziell zum Zweck der Entlohnung der Validatoren eingeführt werden. Man beachte, dass die genaue Ausgestaltung der Belohnung die Sicherheit des Systems beeinflusst (siehe Abschnitt 7.2.2).

In genehmigungsbasierten Blockchain-Systemen hingegen sind die Validatoren bekannt. Man kann sie zum ordnungsgemäßen Durchführen der Validierung verpflichten oder bei Bedarf außerhalb des Blockchain-Systems für ihre Arbeit entlohnen.

4.3 Häufige Programmelemente

Hier soll kurz auf zwei Elemente eingegangen werden, die in einer Vielzahl von Smart Contracts eingesetzt werden: Orakeldienste und Zufallszahlen.

4.3.1 Orakel

Smart Contracts können nur auf Daten zugreifen, die sich innerhalb ihres Blockchain-Universums befinden. Viele Contracts sollen aber auf Ereignisse reagieren können, die außerhalb geschehen, wie z. B. Contracts für Sportwetten, die für Gewinnauszahlungen die Sportergebnisse kennen müssen. Es gibt daher den Bedarf nach einer Funktionalität, die beliebige Daten aus der realen Welt in die Blockchain importiert. Zum jetzigen Zeitpunkt wird dies durch diverse Orakeldienste angeboten, die als Schnittstelle zwischen einer Datenquelle und dem aufrufenden Contract fungieren.

Die Auswahl der Datenquelle geschieht in der Regel durch den aufrufenden Contract. Orakeldienste unterscheiden sich aber darin, ob beliebige oder nur durch den Dienst vorgegebene Quellen zur Auswahl stehen, ob nur eine oder mehrere Quellen gleichzeitig befragt werden können und ob eine nachträgliche Änderung der Quelle möglich ist. Die Übermittlung von Daten aus der Datenquelle an den aufrufenden Contract geschieht je nach Orakeldienst *on-chain* oder *off-chain*. Im ersten Fall speichert der Dienst die angefragten Werte in einem speziellen Service-Contract in der Blockchain, wo sie von Anwendern abgerufen werden können. Im zweiten Fall werden die Daten nicht in der Blockchain, sondern beispielsweise auf einer Website zur Verfügung gestellt und die Zugangsdaten per Transaktion an den aufrufenden Contract verschickt.

Ferner nutzen die Orakeldienste unterschiedliche Methoden, um den Nutzer von der Authentizität der übermittelten Daten zu überzeugen. Es gibt Dienste, die kryptografische Authentizitätsnachweise bereitstellen, die der aufrufende Contract überprüfen kann (z. B. [44]); andere setzen auf einen Abstimmungsprozess, der in der Blockchain veröffentlicht wird und den Anreiz, falsche Daten zu liefern, minimiert (z. B. [45]); wieder andere bieten die Möglichkeit, Daten anzuzweifeln und Einspruch zu erheben, wenn die Antworten des Orakels strittig sind (z. B. [46]).

Für die Programmierer von Smart Contracts sind die Orakeldienste in der Regel unkompliziert zu verwenden, aber es gibt auch hier sicherheitstechnische Aspekte zu berücksichtigen. Auch wenn manche Dienste einen Authentizitätsnachweis für die bereitgestellten Daten liefern, so bürgen sie nicht für die Integrität der Datenquelle. Das bedeutet, dass die vom Orakel gelieferten Daten bereits vorher fehlerhaft oder sogar absichtlich manipuliert sein können. Über die Vertrauenswürdigkeit der Datenquelle sollte sich der Nutzer eines Orakeldienstes im Klaren sein. Es kann die Sicherheit erhöhen, mehrere unabhängige Datenquellen zu befragen und die Mehrheitsmeinung anzuerkennen.

Probleme rein technischer Natur können auch dadurch entstehen, dass das Orakel, auf das ein Contract zugreift, oder die vorgesehene Datenquelle bei Vertragsaufruf womöglich gar nicht mehr existieren. Orakeldienste, die ihre Datenquelle schon zu Beginn festlegen und hier nicht flexibel reagieren können, wird dieses Problem besonders treffen.

4.3.2 Zufallszahlen

Viele Blockchain-Plattformen sind darauf angewiesen, dass Transaktionen von allen Knoten deterministisch ausgewertet werden, damit Blöcke bei der Validierung akzeptiert werden. Andererseits sollen Smart Contracts oftmals zufällige Elemente enthalten, beispielsweise Glücksspiele, deren Nutzer auf faire Spielchancen, also insbesondere auf faire Zufallszahlen vertrauen.

Ein Ansatz, wie man Zufall in ein deterministisches System einbringen kann, ist, deterministische Referenzen auf Werte zu verwenden, die zum Erstellungszeitpunkt noch nicht bekannt sind, z. B. Blockparameter wie die Nummer des Blocks, in den der Contract aufgenommen wird. Einige der vorgeschlagenen Werte haben aber lediglich eine geringe Entropie, so dass sie sich für bestimmte Anwendungen nicht eignen. Des Weiteren mögen sie zum Erstellungszeitpunkt des Contracts unbekannt sein, aber zum Zeitpunkt des Minings bereits feststehen. Dadurch werden sie durch die Miner beeinflussbar [47]. Auch kann es sein, dass auf diese Weise erzeugte Zufallszahlen nicht unabhängig voneinander sind: So werden alle Transaktionen innerhalb eines Blocks, die denselben Blockparameter verwenden, auch dieselbe Zufallszahl erzeugen. Dies kann ein Angreifer ausnutzen, wenn es ihm gelingt, seinen angreifenden Contract in denselben Block einzuschleusen wie sein Opfer [48].

Eine andere Methode ist die Verwendung von Commitment-Verfahren. Hierbei wählt jeder Teilnehmer eine geheime Zufallszahl und veröffentlicht zunächst nur deren Hashwert, wodurch er sich auf seine Wahl verbindlich festlegt. Anschließend werden alle Geheimnisse aufgedeckt und zur Erzeugung einer gemeinsamen Zufallszahl verwendet. Ein Angreifer kann das Ergebnis beeinflussen, indem er entscheidet, sein eigenes Geheimnis entgegen dem Protokoll nicht zu veröffentlichen. Dies soll durch Hinterlegung eines Pfandes unattraktiv gemacht werden, aber die richtige Wahl der Höhe des Pfandes ist nicht trivial. Eine Beeinflussung ist auch auf Ebene der Kommunikation möglich, indem z. B. die Verarbeitung der Transaktionen behindert wird [49].

Auch Orakeldienste (siehe Abschnitt 4.3.1) können genutzt werden, um Zufallszahlen in die Blockchain zu importieren, zum Beispiel von externen Internetseiten. Wie bei allen Orakel-Anwendungen existiert in der Regel keine Möglichkeit, die Qualität der Datenquelle und somit der importierten Zufallszahlen zu kontrollieren. Übermittelt das Orakel die Zufallszahlen in unverschlüsselten Transaktionen, kann ein Angreifer den Wert der Zufallszahl sehen, bevor sie endgültig in einen Block geschrieben wird. Wenn es ihm gelingt,

in der Blockchain eine eigene Transaktion noch vor die Orakel-Transaktion zu platzieren, kann er diese Kenntnis zu seinen Gunsten ausnutzen. Dieser Angriff wird als *Front-Running* bezeichnet [48].

Inwieweit die verschiedenen Ansätze für den Einsatz in einer speziellen Anwendung geeignet sind, hängt zum einen von dem erforderlichen Sicherheitsniveau der Zufallszahlen ab, zum anderen spielt auch das eingesetzte Blockchain-System, insbesondere der Konsensmechanismus, eine entscheidende Rolle. Wo der Einfluss der Miner auf die Verarbeitungsreihenfolge der Transaktionen groß ist oder durch gezielte Anreize von außen gesteuert werden kann, sind die geschilderten Angriffe besonders ernst zu nehmen.

Zusammenfassung.

- Die technische Ausgestaltung von Smart Contracts variiert je nach Blockchain-System mit Auswirkungen unter anderem auf Effizienz und Sicherheit.
- Orakel können ohne zusätzliche Maßnahmen nicht die Authentizität von Daten aus der Außenwelt garantieren.
- Zufallszahlen in ein deterministisches Blockchain-System einzubringen, ist nicht trivial und mit Sicherheitsrisiken verbunden.

Teil II Sicherheit

5 Datensicherheit

Sicherheit und Vertrauen in einem Blockchain-System basieren zum großen Teil auf kryptografischen Basisverfahren wie Signaturen oder Hashfunktionen. Aber auch komplexere kryptografische Mechanismen können zum Einsatz kommen, wenn anspruchsvolle Sicherheitsziele wie Vertraulichkeit oder Anonymität angestrebt werden. Das BSI pflegt seit einigen Jahren eine Technische Richtlinie [50], die eine Bewertung der Sicherheit und langfristige Orientierung für ausgewählte kryptografische Verfahren bietet und konkrete Empfehlungen für Algorithmen und Schlüssellängen enthält. Beim Einsatz von Kryptografie in Blockchains kann mit der Einhaltung dieser Empfehlungen ein angemessenes Sicherheitsniveau für die entsprechenden kryptografischen Anwendungen erreicht werden.

Im Folgenden werden die aus kryptografischer Sicht wichtigsten Aspekte für die Sicherheit einer Blockchain diskutiert und verschiedene Umsetzungsmöglichkeiten dargestellt. Nicht berücksichtigt werden hierbei spezialgesetzliche Anforderungen, wie sie sich zum Beispiel beim Einsatz rechtssicherer elektronischer Signaturen ergeben [51], [52].

5.1 Vertraulichkeit

Das Sicherheitsziel Vertraulichkeit wurde bereits in Abschnitt 2.1.4 angesprochen. Vertraulichkeit steht dem Designprinzip Transparenz der Blockchain-Technologie grundsätzlich diametral gegenüber. Beide Eigenschaften in einer Lösung zu realisieren, ist zumindest sehr herausfordernd. Sollen sensible (z. B. persönliche) Daten auf der Blockchain gespeichert oder verarbeitet werden, kommt eine Aufnahme dieser Daten in Klarlage (also ohne kryptografischen Schutz) in Transaktionen nicht in Frage.

Ein naheliegender klassischer Ansatz für den Schutz der Vertraulichkeit wäre eine Verschlüsselung der sensiblen Daten. Dann müsste aber ein entsprechendes Schlüsselmanagement für die Verschlüsselungsschlüssel zur Verfügung stehen, das konzeptionell in einem Blockchain-System zunächst nicht vorgesehen ist. Außerdem wären solche verschlüsselten Daten in besonderem Maße von den Einschränkungen der Langzeitsicherheit (siehe Abschnitt 6.2) betroffen. Wird ein Verschlüsselungsverfahren unsicher oder werden Schlüssel kompromittiert, so können betroffene Chiffrate nicht einfach zurückgezogen oder umgeschlüsselt werden, da sie im gesamten Netzwerk verteilt sind und nicht mehr unter der Kontrolle des ursprünglichen Herausgebers stehen. Außerdem sind auch verschlüsselte Daten über die Integritätsmechanismen der Blockchain vor Manipulation geschützt. Ein besonders schwerwiegendes Hindernis für die Verwendung von Verschlüsselung ist allerdings die Tatsache, dass verschlüsselte Transaktionen oder Transaktionen mit verschlüsselten Daten nicht ohne Weiteres im Netzwerk verifiziert werden können, da sie nicht für alle Knoten einsehbar sind. Ein kryptografischer Ansatz zur Lösung dieses Problems ist die Verwendung von Zero-Knowledge (ZK)-Protokollen (siehe auch Abschnitt 5.4), die allerdings in der Umsetzung komplex und häufig sehr ressourcenintensiv sind.

Eine einfache praktische Lösung für das Vertraulichkeitsdefizit liegt darin, sensible Daten nicht direkt in der Blockchain zu speichern, sondern nur Referenzen der Daten (z. B. in Form eines Hashwertes) zu verwenden. Die Daten selbst müssten dann in einer externen Struktur (z. B. in einer Datenbank) gespeichert und dort vor unbefugter Einsichtnahme geschützt werden. Bei dieser Lösung können in der Blockchain aber nur die Existenz und Integrität der Daten bescheinigt werden, ihre Verfügbarkeit für die Ausführung von Aktionen (z. B. in Smart Contracts) ist nicht garantiert. Außerdem ist die Vertraulichkeit der Daten hinter den Referenzwerten nicht immer ohne Weiteres sichergestellt (siehe Abschnitt 5.2).

In genehmigungsbasierten Blockchains kann der Umgang mit vertraulichen Inhalten auch organisatorisch gelöst werden. Dabei kommen zum Beispiel separate geschützte Datenkanäle [53]

innerhalb des Blockchain-Netzwerks zum Einsatz, auf die nur zugelassene Knoten Zugriff haben. Innerhalb dieser Gemeinschaft von Knoten (die auch sehr klein sein kann) können die Daten dann als vertraulich angesehen werden. Ähnlich gelagert ist die Konstruktion von „partieller Transaktionssichtbarkeit“ [54], [40], bei der Knoten nur Zugang zu denjenigen Transaktionen haben, in die sie selbst involviert sind oder die sie zur Verifikation der Transaktionshistorie benötigen.

Ein weiterer wichtiger Aspekt betrifft vertrauliche Berechnungen in Transaktionen oder die vertrauliche Ausführung von Smart Contracts. Neben sehr komplexen kryptografischen Konzepten wie homomorpher Verschlüsselung, die aber für den praktischen Einsatz noch kaum bereitstehen, haben sich vor allem Ansätze mit Trusted Execution Environments (TEE) herausgebildet [55]. TEEs bieten für speziell freigeschaltete Anwendungen eine isolierte Laufzeitumgebung, die andere Anwendungen, das Betriebssystem, aber auch den Host-Besitzer selbst davon abhält, den Zustand der Anwendung zu sehen oder zu manipulieren. Im Zusammenhang mit Blockchains könnten so Smart Contracts im Netzwerk ausgeführt werden, ohne dass deren Inhalt bekannt sein muss. Problematisch kann es in diesem Zusammenhang sein, wenn der Konsensmechanismus nicht final ist (z. B. bei PoW, siehe Abschnitt 3.2.3), so dass sich Angriffsmöglichkeiten durch wiederholtes Zurücksetzen eines Smart Contracts ergeben können [56], die Rückschlüsse auf geheime Inhalte erlauben. Außerdem entstehen durch zusätzliche Hardware immer auch neue Angriffsvektoren, z. B. in Form von Seitenkanälen, und das Blockchain-System muss gegen einen möglichen Ausfall der TEEs gehärtet werden.

Vertrauliche Smart Contracts lassen sich auch über secure Multi-Party Computation (sMPC) realisieren [57]. Dabei werden Berechnungen von mehreren Knoten gemeinsam ausgeführt, die jeweils nur Teile der Daten kennen und ihre Informationen nicht mit den jeweils anderen Knoten teilen. Ein Smart Contract lässt sich damit im Netzwerk ausführen, ohne dass eine der Parteien Zugang zum vollständigen Inhalt bekommt. Allerdings sind sMPC-Verfahren relativ ineffizient und damit für viele Anwendungen nicht geeignet.

5.2 Datenablage

Für die Sicherheit der Daten in einem IT-System ist ihre sichere Ablage besonders entscheidend. In einem Blockchain-System werden Daten im Rahmen von Transaktionen in der Blockchain gespeichert. Dabei zählen zu den Transaktionsdaten sowohl die Metadaten, die für die Verifikation und Administration in der Blockchain verwendet werden (z. B. Signatur, Transaktionsgebühr) als auch die Inhaltsdaten, die durch die Transaktion verarbeitet werden (z. B. Zahlungsanweisung, Bescheinigung). Normalerweise liegen diese Transaktionsdaten unverschlüsselt vor und sind damit für alle Knoten mit entsprechenden Rechten einsehbar.

Die Integrität der Daten wird mit Aufnahme der Transaktionen in die Blockchain durch die Verkettung der Datenblöcke mit Hilfe einer Hashfunktion gesichert. Das Sicherheitsziel der Datenintegrität ist jedoch nur bei Verwendung kryptografisch sicherer Hashfunktionen zuverlässig erreichbar. Diese sind sogenannte kollisionsresistente Einwegfunktionen, d. h., es ist praktisch unmöglich, zu einem gegebenen Hashwert eine passende Eingabe oder zwei Eingabewerte mit dem gleichen Hashwert zu finden. Dabei sind Hashfunktionen konstruktionsbedingt nicht injektiv, d. h., es gibt grundsätzlich die Möglichkeit, dass verschiedene Eingabewerte auf den gleichen Hashwert abgebildet werden, nur ist das in der Praxis bei den empfohlenen Hashfunktionen mit der empfohlenen Ausgabelänge [50] nach heutigem Stand extrem unwahrscheinlich und kann nicht mit Absicht herbeigeführt werden.

Es müssen aber nicht alle im System anfallenden oder verarbeiteten Daten als Transaktionsdaten in der Blockchain selbst abgelegt werden, wenn sie nicht unmittelbar zur Ausführung der Transaktion notwendig sind. Wenn es um große Datenmengen oder sensible Daten (siehe Abschnitt 5.1) geht, ist es oft sinnvoll, nur Referenzen auf die Daten in der Blockchain abzulegen. Damit kann die Größe der Blockchain reduziert und die Menge der offen zugänglichen Daten beschränkt werden. Zur Referenzierung empfehlen sich kryptografische Mechanismen mit Integritätsschutz wie zum Beispiel Hashfunktionen, falls die über die Blockchain garantierte Manipulationssicher-

heit auch die referenzierten Daten mit einschließen soll.

Bei der Verwendung von Hashfunktionen für die Referenzierung sollte ebenfalls eine kryptografisch sichere Hashfunktion ausgewählt werden. Deren Eigenschaften garantieren in diesem Fall, dass es praktisch nicht möglich ist, eine einmal verankerte Referenz nachträglich mit Daten zu hinterlegen oder Daten hinter einer Referenz zu ändern.

Wichtig im Zusammenhang mit der Verwendung von Hashfunktionen zur Referenzierung ist, dass Hashfunktionen keine Vertraulichkeit garantieren, obwohl ein Zurückrechnen des Eingabewertes aus seinem Hash praktisch unmöglich ist. Allerdings ist der Suchraum der möglichen sinnvollen Eingabewerte in der Praxis oft beschränkt, da die Daten häufig in besonders strukturierter Form (z. B. Personaldaten in einem Formular im pdf-Format) vorliegen. Dann ist es möglich, mit entsprechendem Rechenaufwand alle möglichen Eingabewerte zu testen (Brute-Force-Angriff). Das kann schon im Vorfeld eines Angriffs geschehen, indem die berechneten Hashwerte aller möglichen Eingaben strukturiert abgespeichert werden (Wörterbuch-Angriff). Will man also verhindern oder zumindest erschweren, dass über den Hashwert Informationen über die Daten durchsickern, so muss man beim Hashen Entropie hinzufügen, die Wörterbuchangriffe aufwändiger macht. Normalerweise wird das über einen sogenannten *Salt* realisiert, also einen zusätzlichen zufälligen Eingabewert. Dieser muss nicht geheim, aber unvorhersagbar sein.

Verfügen zwei oder mehr Parteien auf der Blockchain, die alle auf die referenzierten Daten zugreifen dürfen, über einen gemeinsamen geheimen Schlüssel, so kommt auch die Verwendung eines Message Authentication Code (MAC) zur Referenzierung von Daten in Frage. Mit einem MAC kann zusätzlich zur Integritätssicherung auch die Authentizität der Daten festgestellt werden. Ein Salt ist bei der Erstellung eines MAC nicht nötig, da die Referenzwerte nur bei Kenntnis des geheimen Schlüssels berechnet werden können.

Falls eine Public-Key-Infrastruktur (PKI) zur Verfügung steht, können außerdem statt eines einfachen Hashwertes Signaturverfahren auf die Daten angewendet werden und die Signaturen als Referenzen in die Blockchain eingestellt werden. Zusätzlich zur Integrität der Daten wird damit auch die Herkunft von einem über die PKI identifizierbaren Signaturersteller bescheinigt. Beim Einsatz von Signaturen ist die Verwendung eines Salts empfehlenswert, da ansonsten wieder Brute-Force-Angriffe auf die zugrunde liegenden signierten Daten möglich sind.

5.3 Kryptografische Schlüssel

In Blockchain-Systemen sind für die Erreichung der Hauptziele Dezentralisierung, Transparenz und Manipulationssicherheit nicht zwangsläufig schlüsselbasierte Verfahren notwendig. Allerdings lassen sich verschiedene Eigenschaften von Transaktionen am besten durch die Verwendung asymmetrischer kryptografischer Verfahren umsetzen. Das betrifft vor allem den Nachweis der Urheberschaft von Transaktionen (Authentizität), für den elektronische Signaturverfahren die klassische Lösung sind.

Der Einsatz von asymmetrischer Kryptografie ist immer mit der Notwendigkeit des Schlüsselmanagements verbunden. Das umfasst die Erzeugung der Schlüsselpaare, die Speicherung der privaten Schlüssel und die Verteilung der öffentlichen Schlüssel.

Die Verteilung der öffentlichen Schlüssel hängt insbesondere von den Erfordernissen der Identifizierung ab. Sollen öffentliche Schlüssel pseudonym ohne Verknüpfung zum Besitzer verwendet werden, wie es beispielsweise in Bitcoin der Fall ist (siehe Abschnitt 5.4), so können sie ungeprüft im Netzwerk verbreitet werden (z. B. als Bestandteil der Transaktionen). Ist jedoch die Identität (oder eventuell weitere Eigenschaften) hinter einem öffentlichen Schlüssel wichtig, so muss diese vertrauenswürdig bescheinigt und geprüft werden. Für diesen Fall haben sich Public-Key-Infrastrukturen als geeignete Lösung etabliert, die allerdings mit einigem organisatorischen und technischen Aufwand verbunden sind und parallel zu den Blockchain-Strukturen betrieben werden müssen. Zudem ist damit wieder eine vertrauenswürdige zentrale Stelle (Certificate Authority) notwendig.

Bei der Speicherung der privaten Schlüssel treten in der Praxis die meisten Fehler auf. Ist – wie im Bitcoin-System – der private Signaturschlüssel der einzige Nachweis über den Besitz von Eigentumswerten auf der Blockchain, so ist dieser Schlüssel hochgradig gefährdet. Private Schlüssel sollten in einem besonders gesicherten Bereich, z. B. auf einem Hardware-Token, in einem verschlüsselten Festplatten-Container oder auch offline in einem Safe, aufbewahrt werden. Zum Schutz vor Verlust sollten außerdem Backup-Kopien angelegt und getrennt verwaltet werden. Es gibt inzwischen eine Reihe von Anbietern, die Lösungen für elektronische Brieftaschen *(Wallets)* zur Aufbewahrung von Schlüsseln zur Verfügung stellen. Die Sicherheit dieser Wallets ist sehr unterschiedlich und muss in jedem Fall individuell geprüft werden. Unter anderem hängt die Sicherheit davon ab, ob es sich um ein *Hot Wallet*, also einen Speicher mit Verbindung zum Internet, oder ein *Cold Wallet* ohne Internetverbindung handelt. Mit einem Cold Wallet kann das Risiko von Hacker-Angriffen deutlich reduziert werden.

Auch die richtige Schlüsselerzeugung spielt aus Sicherheitssicht eine große Rolle. Für die Erzeugung der privaten Schlüssel wird immer Zufall benötigt, mit einem rein deterministischen Verfahren ist keine Geheimhaltung möglich. Wird hier ein schwacher Zufallsgenerator verwendet, dessen Ausgabe in irgendeiner Form vorhersagbar ist, so können Angriffe auf den geheimen Schlüssel deutlich vereinfacht werden. Gerade bei Online-Wallets ist schwer zu beurteilen, woher der benötigte Zufall für die Schlüsselerzeugung stammt und welche Qualität er hat. Auch bei der Erzeugung oder Verwendung von Zufallszahlen in kryptografischen Verfahren sollten die Empfehlungen des BSI berücksichtigt werden [50], [58], [59], [60].

Weitere Ausführungen zu praktischen Bedrohungen und Sicherheitsvorfällen im Schlüsselmanagement finden sich im Abschnitt 7.3.2.

Auch wenn die Schlüssel sicher erstellt und verwaltet werden, kann ein Kryptoverfahren über seine geheimen Schlüssel angreifbar sein.

Beispiel Bitcoin-Signaturschlüssel

In Abschnitt 1.3 wurde erklärt, wie eine Bitcoin-Adresse aus dem öffentlichen Signaturschlüssel eines Nutzers berechnet wird. Daraus ergeben sich zwei nicht intendierte Wege, um an das Guthaben hinter der Adresse zu kommen (siehe Abbildung 7).

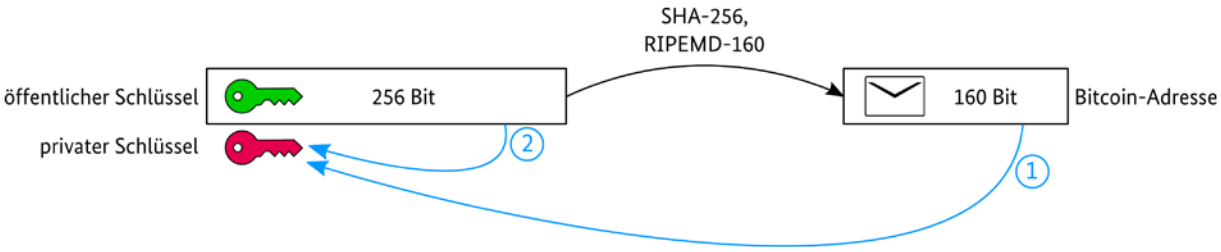

Abbildung 7: Zwei Angriffspfade auf den privaten ECDSA-Signaturschlüssel

Erstens gehören durch die Adresslänge von 160 Bit zu jeder Adresse etwa 2^{96} gültige geheime Signaturschlüssel von 256 Bit Länge, die ein Angreifer durch Probieren zu finden versuchen kann. Dazu hat er so lange Zeit, wie die Adresse noch mit Guthaben hinterlegt ist.

Wenn allerdings Adressen wiederverwendet werden, dann ist der öffentliche Schlüssel bekannt, da er bei vorherigen Transaktionen mit übermittelt wurde, und es bietet sich als zweite Möglichkeit ein Angriff auf das DLP an. Nach heutigem Forschungsstand würde man für einen solchen Angriff den Pollard-Rho-Algorithmus [61] verwenden. Oft wird behauptet, dass man für diesen Angriff nur zehn Minuten Zeit hat, da es so lange dauert, bis eine Transaktion in einen Block aufgenommen wird. Dies ist aber ein Trugschluss, da in Bitcoin häufig Adressen wiederverwendet werden und damit viele öffentliche Schlüssel schon länger bekannt sind.

Die Sicherheit von asymmetrischen Verfahren beruht darauf, dass die geheimen Schlüssel (z. B. zum Entschlüsseln oder Signieren) nicht mit realistischem Aufwand aus den öffentlichen Schlüsseln berechnet werden können. Dafür sorgt die Schwierigkeit des zugrunde liegenden mathematischen Problems (z. B. Diskreter-Logarithmus-Problem (DLP) oder Faktorisierungsproblem). Kann jedoch durch kryptoanalytische Methoden die Schwierigkeit des Problems herabgesetzt werden, so kann ein Angreifer den geheimen Schlüssel ganz oder teilweise ermitteln und damit das Verfahren aushebeln.

5.4 Pseudonymität und Anonymität

Das Bitcoin-System ist unter anderem mit dem Versprechen angetreten, die Privatsphäre der Nutzer durch anonyme Schlüssel zu schützen [2]. Adressen für Transaktionen werden aus Hashwerten der öffentlichen Signaturschlüssel des Empfängers berechnet (siehe Abschnitt 1.3), deren entsprechende private Schlüssel später eine Weiterüberweisung des erhaltenen Betrags authentisieren können. Nutzer müssen in Bitcoin also nicht mit ihrer realen Identität auftreten, sie brauchen jedoch eine Adresse im Bitcoin-System, um ansprechbar zu sein. Deshalb kann man bei Bitcoin auch nicht von Anonymität, sondern höchstens von Pseudonymität sprechen.

Doch auch die Pseudonymität ist in der Praxis nicht zuverlässig erreichbar. Transaktions- und Netzwerkanalysen können dazu genutzt werden, Nutzerprofile zu erstellen, und durch Schnittstellen zur realen Welt, z. B. bei Bezahldiensten, können die Identitäten der Nutzer aufgedeckt werden [62], [63], [64], [65], [66]. Es stehen bereits Tools mit teils hervorragenden Erfolgsraten zur Verfügung [67], [68], die von Forschern, Strafverfolgern oder auch privat genutzt werden können. Ähnliche Bemühungen zur Deanonymisierung von Teilnehmern gibt es auch für die Ethereum-Blockchain [69], [70].

Verbesserungen im Bereich der Anonymität versprechen verschiedene alternative Kryptowährungen, die Konzepte für anonyme Nutzer und/oder anonyme Transaktionen einsetzen. Dabei haben sich im Wesentlichen drei verschiedene Ansätze herausgebildet:

- Mixing: Unter Mixing versteht man Methoden, die Transaktionen verschiedener Nutzer in der Blockchain so miteinander zu vermischen, dass die ursprüngliche Quelle einer einzelnen Transaktion nicht mehr nachvollzogen werden kann. Für manche Blockchain-Systeme (z. B. Bitcoin) bieten externe Dienstleister Mixing-Services an, denen man allerdings dann seine Transaktionen anvertrauen muss. Bei anderen Systemen wie der Kryptowährung Dash [71] ist das Mixing fester Bestandteil des Protokolls.

- Ringsignaturen: Bei Ringsignaturen werden digitale Signaturen mit Hilfe einer Gruppe von Netzwerkteilnehmern erstellt, die aber nicht aktiv oder wissentlich an der Signaturerstellung beteiligt ist. Die Gruppengröße und die öffentlichen Schlüssel ihrer Mitglieder sind Bestandteil der Signatur. Dadurch können signierte Transaktionen eindeutig verifiziert werden, die Urheberschaft der Transaktionen kann aber nur bis auf die Signaturgruppe zurückgeführt werden und nicht auf den einzelnen Teilnehmer. Entscheidend für die praktische Anonymität dieser Verfahren ist eine ausreichend große Signaturgruppe. Damit wächst aber auch das Datenvolumen der Transaktionen stark an, was die Effizienz einschränkt. Ringsignaturen werden zum Beispiel bei der Kryptowährung Monero [72] eingesetzt.

- Zero-Knowledge-Verfahren: ZK-Beweise ermöglichen es einer Partei, einer anderen Partei zu beweisen, dass eine bestimmte Aussage wahr ist, ohne weitere Informationen über die Aussage selbst preiszugeben. Damit können Transaktionen in der Blockchain eingestellt werden, die weder den Absender noch den Empfänger noch den überwiesenen Betrag offenbaren müssen. Bei diesen Verfahren können die Daten in den Transaktionen vertraulich gehalten werden. Die meisten der heute schon eingesetzten Verfahren (z. B. zk-SNARKs [73]) sind relativ ineffizient, da die Komplexität der Beweise mit der Größe des Datenvolumens stark ansteigt. Außerdem

ist in einigen Verfahren eine geheime Setup-Phase zur Generierung bestimmter Parameter notwendig, die nicht öffentlich nachvollziehbar ist. Neuere Varianten (z. B. zk-STARKs [74]) versprechen aber eine bessere Skalierbarkeit und Transparenz. Praktische Anwendung finden ZK-Verfahren zum Beispiel in der Kryptowährung Zcash [75].

Allerdings sind auch bei vielen Lösungen, die diese Ansätze verwenden, bereits Probleme im praktischen Einsatz der Anonymitätsfunktionalität aufgetreten. Es gibt inzwischen eine Reihe von Untersuchungen, die trotz der implementierten Schutzmechanismen Möglichkeiten zur Nachverfolgung und Verbindung von Transaktionen gefunden haben [76], [77], [78]. Meist spielen auch hier Seitenkanalinformationen oder Metadaten eine Rolle, die außerhalb der Transaktionen im Netzwerk anfallen.

5.5 Sicherheitsaussagen

Die Sicherheit eines IT-Systems hängt maßgeblich von der Sicherheit der zugrunde liegenden kryptografischen Verfahren ab. Das gilt im besonderen Maße auch für Blockchain-Systeme, da hier (insbesondere in öffentlichen genehmigungsfreien Blockchains) oft zugunsten der Dezentralität und Disintermediation auf organisatorische Sicherheitsmaßnahmen verzichtet wird.

Die Sicherheit kryptografischer Verfahren wiederum kann in Abhängigkeit von den gewählten Parametern (z. B. Schlüssellängen) als Sicherheitsniveau in Bits angegeben werden. Ein Sicherheitsniveau von n Bits bedeutet dabei, dass der Aufwand zum Brechen des Verfahrens 2^n elementare Operationen beträgt. Damit kann die Sicherheit auf die Rechenstärke eines möglichen Angreifers zurückgeführt und quantifiziert werden. Ausgehend vom gewünschten Sicherheitsniveau bzw. der erwarteten Angreiferstärke sollten die benötigten Parameter der Kryptoverfahren gewählt werden. Die Empfehlungen des BSI [50] zielen zum Beispiel auf ein Sicherheitsniveau von mindestens 100 Bit ab. Dabei muss aber berücksichtigt werden, dass solche Ableitungen immer nur den aktuellen Wissensstand über mögliche Angriffe widerspiegeln. Werden durch technische oder kryptoanalytische Fortschritte bessere Angriffe möglich, so müssen die Parametergrößen verändert oder sogar das Verfahren aufgegeben werden (siehe Abschnitt 6.1).

Neben der Technischen Richtlinie des BSI [50] gibt es verschiedene internationale Standards zur Kryptografie (z. B. von ETSI, NIST, IEEE, ISO/IEC, IETF, ANSI), die Empfehlungen und Vorgaben für sichere kryptografische Verfahren bieten und den Stand der Technik festlegen. Die zurzeit bei den bekannten Blockchain-Implementierungen (z. B. Bitcoin, Ethereum, Hyperledger Fabric) verwendeten klassischen kryptografischen Mechanismen (z. B. die Hashfunktion SHA-256, ECDSA-Signaturen) entsprechen in der Regel den Empfehlungen und erreichen ein angemessenes Sicherheitsniveau. Bei den neueren oder spezielleren Kryptoverfahren (z. B. zk-SNARKs, sMPC) gibt es oft noch keine breiter untersuchten oder unabhängig verifizierten Sicherheitsaussagen. Das macht Bewertungen des Sicherheitsniveaus oder Empfehlungen sehr schwierig.

Grundsätzlich ist die Sicherheit eines IT-Systems höchstens so groß wie die Sicherheit seiner schwächsten Komponente und außerdem abhängig von der Komposition der einzelnen Mechanismen und ihrer Integration in das Gesamtsystem. Zusätzlich spielen die Qualität der Implementierung und die Umweltbedingungen eine entscheidende Rolle. Da Blockchain-Systeme durch ihre verteilte Struktur und ihre Schnittstellen sehr komplex werden können, sind Aussagen über die Gesamtsicherheit nur schwer zu erreichen.

Eine weitere Herausforderung insbesondere bei der quantitativen Bewertung der Sicherheit von Blockchains stellen die Konsensmechanismen (siehe Abschnitt 3.2) dar. Eine valide Konsensbildung ist essentiell für die Datensicherheit in Blockchains, insbesondere für die Manipulationssicherheit. Zur Sicherheit tragen hier neben kryptografischen Mechanismen (z. B. Hashfunktionen) vielfach auch ökonomische Anreize bei. Man spricht deshalb auch von Kryptoökonomie. Dabei spielen oft spieltheoretische Überlegungen und Erkenntnisse eine Rolle [79]. Gleichzeitig wirken sich diese Faktoren auch wieder auf das Design

von kryptografischen Verfahren und Protokollen aus, das einem veränderten Angreifermodell angepasst werden muss [80]. Der Sicherheitsbeitrag der ökonomischen Anreize lässt sich generell nur schwer in Zahlen fassen und ein Vergleich oder eine Relation zum klassischen kryptografischen Sicherheitsniveau ist kaum möglich (siehe auch Abschnitt 3.2.5).

Zusammenfassung.

- Vertraulichkeit ist in Blockchains schwer zu erreichen.
- Sensible Daten sollten nicht direkt bzw. nicht ungeschützt auf der Blockchain gespeichert oder verarbeitet werden.
- Kryptografische Verfahren sollten dem Stand der Technik entsprechen und brauchen ein gutes Schlüsselmanagement.
- Mechanismen zur Anonymisierung und Pseudonymisierung in Blockchains sind in der Praxis oft nicht zuverlässig.
- Konkrete (insbesondere quantitative) Aussagen zum Sicherheitsniveau können für die meisten Blockchain-Anwendungen nicht getroffen werden.

6 Langzeitsicherheit und Kryptoagilität

Sensible Daten mit langfristigem Schutzbedarf müssen in einer Blockchain besonders geschützt werden. Aufgrund der langen Verfügbarkeit bei gleichzeitig potenziell hoher Sensibilität von Daten in der Blockchain stellt die Erreichung von Langzeitsicherheit eine besondere Herausforderung dar. Durch ein geeignetes Konzept zur Kryptoagilität ist sicherzustellen, dass die Sicherheitsmechanismen der Blockchain bei Bedarf ausgetauscht werden können. Dabei sind insbesondere Anforderungen, die sich aus der Gefährdung durch technische und mathematische Fortschritte in der Kryptoanalyse oder potenzielle Quantencomputer ergeben, zu beachten.

Nachfolgende Ausführungen zeigen jedoch, dass sich selbst durch Kryptoagilität einige Schutzziele nicht langfristig garantieren lassen.

6.1 Kryptoagilität und Sicherheitsgarantien

Grundsätzlich ist es nicht möglich, die Sicherheit kryptografischer Algorithmen längerfristig zu garantieren – selbst wenn man Implementierungsaspekte außer Acht lässt. Das BSI nennt in seiner Technischen Richtlinie TR-02102 [50] Algorithmen, von denen es erwartet, dass sie noch für mindestens sechs bis sieben Jahre sicher sind. Die typischen Sicherheitsbeweise reduzieren die Sicherheit eines kryptografischen Verfahrens auf ein mathematisches Problem, das für schwierig gehalten wird. Prominente Beispiele aus der heute verwendeten asymmetrischen Kryptografie sind das Diskreter-Logarithmus-Problem (DLP) oder das Faktorisierungsproblem. In der symmetrischen Kryptografie weist man nach, dass bekannte Angriffe nur mit großem Aufwand möglich sind. Die zugrunde liegenden Sicherheitsannahmen können sich grundsätzlich jederzeit als praktisch falsch herausstellen. Beispielsweise ist bekannt, dass (noch hypothetische) Quantencomputer ausreichender Größe leicht das Faktorisierungsproblem oder das DLP in kryptografischen Größenordnungen lösen können. Zudem können sie eventuell verwendet werden, um klassische kryptografische Angriffe zu beschleunigen oder um Seitenkanalangriffe zu erleichtern. Der Entwicklungsstand von Quantencomputern wird durch das BSI in einer Dauerstudie [81] untersucht.

Momentan entwickelt sich das Gebiet der Post-Quanten-Kryptografie, deren Algorithmen nach heutigem Kenntnisstand sicher gegen quantencomputergestützte und klassische Angriffe sind. Die Bewertung dieser Algorithmen kann sich im Laufe der Zeit deutlich verändern. Daher sind Sicherheitsaussagen über lange Zeiträume mit einer erheblichen Unsicherheit verbunden. Dennoch können Post-Quanten-Algorithmen eine Alternative zu klassischen Algorithmen für den Einsatz in Blockchains sein.

Zumindest aber muss die Blockchain so aufgesetzt werden, dass ein Austausch von Kryptoverfahren zur Laufzeit möglich ist. Diese Eigenschaft wird als *Kryptoagilität* bezeichnet. Kryptoagilität für Blockchains ist ein aktuelles Forschungsgebiet, bei dem noch keine festen Konzepte favorisiert werden [82].

Je nach Art der Blockchain kann ein solcher Austausch von kryptografischen Verfahren – falls keine Einigung in der Entwickler- und Nutzergemeinschaft hergestellt werden kann – zur Abspaltung eines neuen Entwicklungszweiges, also einem Fork, führen. Für Kryptowährungen wurde diese Option schon diskutiert, siehe [83]. Besonders problematisch wird es dabei, wenn ein Hardfork nötig ist, also eine Abspaltung der Blockchain mit den neuen Mechanismen, die nicht mehr mit der alten Version kompatibel ist und deren Einführung deshalb ein hoch synchronisiertes Vorgehen innerhalb des Netzwerks verlangt.

Wird bei einer Kryptowährung eine feste Hashfunktion für PoW verwendet, so wird ein großer Teil der Miner Spezialhardware verwenden und damit benachteiligt, wenn ein Wechsel zu einer anderen Hashfunktion stattfindet, die vielleicht bessere Sicherheitseigenschaften bietet. Ein Wechsel zu langzeitsicheren Kryptoverfahren

kann also auch an wirtschaftlichen Gründen scheitern.

Grundsätzlich erhält aber selbst ein erfolgreicher Austausch von kryptografischen Verfahren nicht automatisch die ursprünglichen Sicherheitsgarantien für ältere Daten.

Aufgrund der dargestellten Probleme sollte Langzeitsicherheit keine Kernanforderung für eine Blockchain-Anwendung sein, und Blockchains allein erscheinen nicht zur Lösung von Fragen der Langzeitarchivierung geeignet. Im besonderen Maße gilt dies bei öffentlichen genehmigungsfreien Blockchains, bei denen prinzipiell jeder Zugriff auf die gesamte Blockchain hat. Besser stellt sich die Situation in privaten genehmigungsbasierten Blockchains dar, insbesondere wenn die Kommunikation zwischen den Partnern verschlüsselt erfolgt, die Daten exklusiv geteilt werden oder redundant in einer gesicherten externen Umgebung abgelegt sind.

6.2 Langzeitsicherheit

Daten in einer Blockchain sind typischerweise sehr lange (wenn nicht sogar unbeschränkt) zugänglich und müssen unter Umständen über ihre gesamte Lebensdauer geschützt werden. Einmal in die Blockchain eingefügt, werden sie dort für die gesamte Laufzeit der Blockchain – so zumindest das Ziel – unverändert verfügbar sein. Zudem ist jeder befugte Nutzer jederzeit in der Lage, die Blockchain bei sich abzuspeichern und später – etwa wenn sich bei den verwendeten kryptografischen Verfahren Schwächen zeigen sollten – auszuwerten.

Bei der Bewertung der Langzeitsicherheit einer Blockchain sind die folgenden Schutzziele zu berücksichtigen:

6.2.1 Integrität

Um die Integrität der gespeicherten Daten dauerhaft sicherstellen zu können, müssen Blockchain-Systeme jederzeit die Integrität der gesamten Blockchain gewährleisten, selbst wenn die verwendeten Kryptoalgorithmen zu einem späteren Zeitpunkt als nicht mehr ausreichend sicher bewertet werden. Im Standardfall einer Blockchain, die mit Hilfe von Hashwerten verkettet wird, ist dann die Sicherheit der verwendeten Hashfunktion zu gewährleisten. Hier bietet sich die Möglichkeit, alte Teile der Blockchain neu zu hashen, den Hashwert in einen neuen Block aufzunehmen und so die Integrität mit Hilfe einer neuen sichereren Hashfunktion zu gewährleisten. Dabei müssten die neuen Daten auf die ursprünglichen Daten in der Blockchain verweisen. Eine Neuverhashung setzt aber Zugriff auf die alten Daten voraus. Zudem muss ihr Inhaber eventuell beteiligt werden.

6.2.2 Authentizität

Werden Daten elektronisch signiert in die Blockchain eingestellt, etwa mit dem Ziel in rechtsverbindlicher Form eine Urheberschaft zu bestätigen, so muss dies auch gewährleistet sein, falls das zugrunde liegende Signaturverfahren angreifbar wird. Die Unveränderlichkeit der Blockchain verhindert es, nachträglich die Signaturen von Daten in ihren ursprünglichen Transaktionen durch neue Signaturen zu ersetzen. Eine neue Signatur muss daher in einen neuen Block eingefügt und in geeigneter Weise mit den Ursprungsdaten verknüpft werden. Um die Garantie für die Authentizität signierter Daten dauerhaft zu erhalten, muss die Signatur der Daten rechtzeitig mit einem sichereren Algorithmus erneuert werden. Außerdem gilt es sicherzustellen, dass die verwendeten Signaturverfahren zumindest zum Zeitpunkt der jeweiligen Signaturerstellung noch sicher waren. Blockchain-Anwendungen mit dem Ziel von Langzeitsicherheit müssen es daher erlauben, nachträglich festzustellen, wann Daten eingefügt wurden. Sie müssen also einen genügend genauen nachprüfbaren Zeitstempel verwenden.

Es kann – im Beispiel der heute typischerweise verwendeten Signaturverfahren, die nur den Hashwert von Daten signieren (siehe Abbildung 8) – nicht ausgeschlossen werden, dass das Signaturverfahren selbst noch sicher ist, aber die von ihm verwendete Hashfunktion gebrochen wurde.

Sollte es zum Beispiel möglich werden, Zweiturbilder zu erzeugen, also zu einer festen Nachricht eine andere Nachricht mit identischem Hashwert zu finden, so ist es nicht hinreichend, nur den Hashwert der ursprünglichen Nachricht signiert in die Blockchain aufzunehmen, denn ein potenzieller Angreifer könnte die zweite Nachricht vorweisen, die dieselbe Signatur besitzt. Dies gilt auch dann, wenn ein Verweis auf eine Datei, die die Nachricht enthält, in die Kette aufgenommen wird und in dieser Datei die ursprüngliche durch die zweite Nachricht ersetzt werden kann. Solche Angriffe auf die Authentizität müssen durch angemessene Maßnahmen verhindert werden, beispielsweise durch die in Abschnitt 6.2.1 angesprochene Neuverhashung alter Daten.

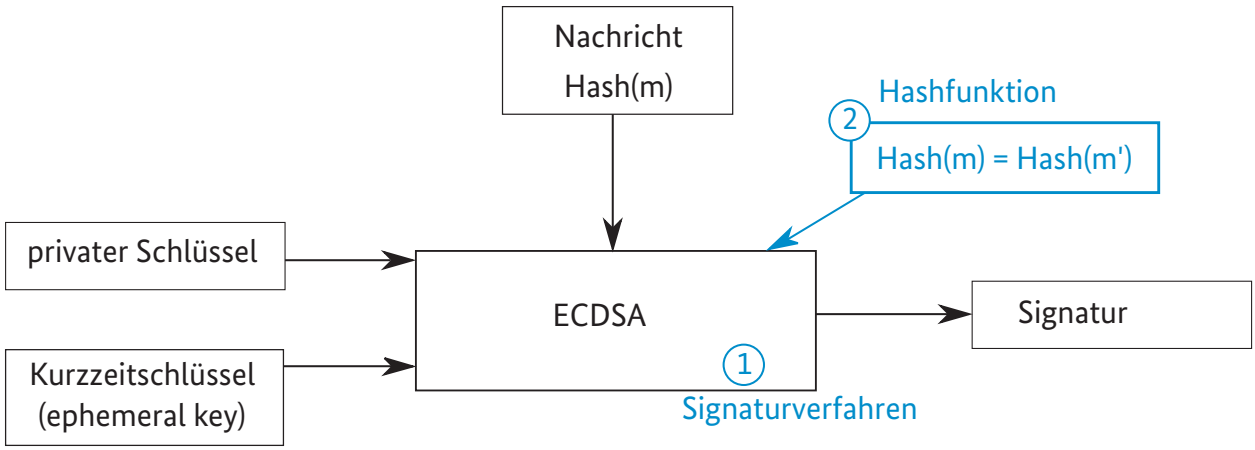

Abbildung 8: Signaturverfahren sowie Hashfunktion können langfristig angreifbar werden.

6.2.3 Vertraulichkeit

In vielen Fällen ist es nicht sinnvoll, Daten offen in eine Blockchain zu schreiben, einerseits zum Schutz der Vertraulichkeit von Daten, andererseits erhöht jede Ablage von größeren Dateien – verschlüsselt oder unverschlüsselt – die Datenmenge in einer langlebigen Blockchain stark.

Zudem ist auch das direkte signierte Ablegen von verschlüsselten Daten mit dem Ziel, Vertraulichkeit zu erreichen, mit Herausforderungen verbunden. Beispielsweise könnte der verwendete Verschlüsselungsalgorithmus zukünftig gebrochen werden und es somit gelingen, die Daten durch Entschlüsseln zu erlangen. Hier reicht ein reines Neuverschlüsseln der Daten als Gegenmaßnahme nicht, denn ein potenzieller Angreifer verfügt in der Regel über die gesamten Daten der Blockchain, die im Idealfall an vielen Stellen lokal vorhanden sind (siehe Abschnitt 5.1).

Beispiel Bitcoin-Adressen

Im Beispiel in Abschnitt 5.3 wurde erläutert, dass einem Angreifer ein erfolgreicher Angriff auf das DLP gelingen müsste, um aus einem bekannt gewordenen öffentlichen Bitcoin-Schlüssel den zugehörigen privaten Schlüssel zu errechnen. Es ist aber schon jetzt erwiesen, dass die (noch hypothetischen) Quantencomputer geeigneter Größe mit Shors Algorithmus [84] das Problem viel effizienter lösen können als alle derzeit bekannten Algorithmen. Durch entscheidende Fortschritte in der Entwicklung von Quantencomputern wären Bitcoin-Guthaben daher unmittelbar bedroht.

Man erkennt allgemein, dass alte oder wiederverwendete Bitcoin-Adressen besonders gefährdet sein können und es sinnvoll ist, Guthaben gelegentlich an neue eigene Adressen weiterzuleiten (was jedoch mit Kosten verbunden ist), sobald sich abzeichnet, dass das DLP bei Bitcoin angreifbar wird. Sollte das Bitcoin-Signaturverfahren geändert werden müssen, ist ein Wechsel theoretisch relativ einfach möglich, solange es zu den jeweiligen Bitcoins noch einen Eigentümer mit Schlüssel gibt. Es müsste jedoch in der Bitcoin-Community Einigkeit über einen Algorithmenwechsel erzielt werden, was aufgrund von Erfahrungen aus der Vergangenheit als schwierig angesehen werden kann. Der Eigentümer müsste dann eine Transaktion an eine eigene Adresse, die mit dem neuen Verfahren gesichert ist, auslösen. Führt er eine solche Transaktion nicht aus, so ist sein Guthaben diebstahlgefährdet. Es gibt schon heute Guthaben, zu denen der Signaturschlüssel verloren gegangen ist. Nach Schätzungen handelt es sich um etwa drei Millionen verlorener Bitcoins [85]. Diese Guthaben könnten (abzüglich der Kosten eines Angriffes) wieder verfügbar gemacht werden.

Zusammenfassung.

- Die Gestaltung von Blockchains, die langlebige Daten enthalten, ist besonders anspruchsvoll und erfordert ein Konzept zur Kryptoagilität.
- Ein Austausch von kryptografischen Verfahren erhält nicht automatisch die ursprünglichen Sicherheitsgarantien.
- Sensible langlebige Daten sollten nicht direkt – verschlüsselt oder unverschlüsselt – in einer Blockchain abgelegt werden.
- Die Sicherheit von langlebigen Daten kann mit der Blockchain-Technologie am ehesten in privaten genehmigungsbasierten Blockchains erreicht werden.

7 Angriffe

Wie jede Technologie bietet auch die Blockchain-Technologie Möglichkeiten für Angriffe. Neben neuen Angriffsvektoren z. B. gegen die Konsensmechanismen sind grundsätzlich auf verschiedenen Ebenen Angriffe möglich, die aus anderen Bereichen wohlbekannt sind (siehe Abbildung 12). Dieser Abschnitt zeigt die bestehenden Bedrohungen auf. Konkrete Vorfälle mit oft beträchtlichen Schäden untermauern die praktische Relevanz dieser Angriffe. Da jedes Blockchain-System eine komplexe IT-Infrastruktur ist, müssen auch hier die im Bereich IT-Sicherheit bekannten allgemeinen Schutzvorkehrungen getroffen werden, die am Ende dieses Abschnitts kurz aufgeführt sind.

7.1 Angriffe auf den Blockchain-Kern

7.1.1 Kryptografische Routinen

Die Sicherheit einer Blockchain basiert auf der Sicherheit der verwendeten kryptografischen Funktionen (siehe Abschnitt 5.3). Dies sind unter anderem Verfahren zur Authentisierung von Knoten und Transaktionen, die in der Regel durch digitale Signaturen mittels eines Public-Key-Kryptosystems umgesetzt sind, sowie Hashfunktionen, die zur Verkettung der Blöcke und damit der Integritätssicherung der in der Blockchain gespeicherten Daten dienen. Zudem werden Hashfunktionen eingesetzt, um die Transaktionsdaten für die Verkettung zusammenzubinden (Hashbäume), und bei den verbreiteten Signaturverfahren werden nicht die Transaktionsdaten selbst signiert, sondern ihre Hashwerte.

Falls ein Angreifer das verwendete Public-Key-Kryptosystem brechen und unbefugt Signaturen für vorgegebene Daten erstellen kann, kann er beliebige Transaktionen im Namen anderer Knoten veranlassen. In Kryptowährungen bedeutet dies beispielsweise, dass der Angreifer das gesamte Vermögen anderer Knoten stehlen kann.

Falls ein Angreifer für die verwendete Hashfunktion ein Zweiturbild für gegebene Hashwerte berechnen kann, so ist er in der Lage, bereits bestätigte und fest in der Blockchain verankerte Blöcke zu verändern, was die Integrität des Systems völlig zerstört. Neben der Integrität kann in diesem Fall auch die Authentizität von Transaktionen nicht mehr garantiert werden. Ein Angreifer kann eine Transaktion so modifizieren, dass ihr Hashwert unverändert und damit die ursprüngliche Signatur gültig bleibt. Details zu den genannten Angriffen finden sich in den Abschnitten 5.2 und 6.2.

In vielen bekannten Blockchains wie Bitcoin, Ethereum oder Hyperledger Fabric werden standardisierte kryptografische Verfahren verwendet, die gegenwärtig als sicher angesehen werden können (z. B. die Hashfunktion SHA-256, ECDSA-Signaturen). Auf der anderen Seite sind auch Fälle bekannt geworden, in denen in Blockchains selbst entwickelte kryptografische Algorithmen eingesetzt wurden. So verwendete die Kryptowährung IOTA anfänglich eine selbst geschriebene Hashfunktion. Diese Funktion wurde in kurzer Zeit von Kryptoanalysten gebrochen [30] und erst nach weiterer Diskussion durch eine standardisierte Hashfunktion ersetzt.

7.1.2 Konsensmechanismen

Der Konsensmechanismus stellt in einer Blockchain die übereinstimmende Speicherung eines gemeinsamen Zustands auf den verschiedenen Knoten sicher, Details finden sich in Abschnitt 3.2.1. Dort werden auch alle hier angesprochenen Algorithmen vorgestellt. Je nach Ausgestaltung sind verschiedene Angriffe denkbar, siehe dazu auch Abschnitt 3.2.4.

Die grundsätzlichen Angriffsmöglichkeiten gegen CFT- und BFT-Algorithmen wurden bereits in Abschnitt 3.2.2 dargestellt. Weitverbreitete CFT- und BFT-Algorithmen verwenden Public-Key-Kryptosysteme zur Authentisierung

von Nachrichten. Damit ermöglicht ein Angriff auf diese kryptografischen Funktionen auch einen direkten Angriff auf den Konsensmechanismus.

Der Konsensmechanismus PoW verwendet in der Regel für das zugrunde liegende mathematische Rätsel (siehe Abschnitt 3.2.3) eine Hashfunktion. Falls ein Angreifer diese Hashfunktion brechen und passende Urbilder effizient berechnen kann, so kann er die Erstellung der Blöcke völlig kontrollieren. Auch ohne diese Fähigkeit kann PoW angegriffen werden, wenn ein Angreifer mehr als die Hälfte der Rechenleistung des jeweiligen Netzwerks unter seine Kontrolle bringt, wie bereits in der Originalarbeit [2] von Nakamoto bemerkt wird. Dies wird als 51%-Angriff bezeichnet. In diesem Fall kann der Angreifer neue Blöcke mit von ihm gewünschten Transaktionen an frühere Blöcke der Blockchain anhängen. Aufgrund seiner Rechenleistung wird die von ihm erzeugte Kette nach einer gewissen Zeitspanne länger sein als die der übrigen korrekten Knoten und danach von diesen gemäß dem Protokoll übernommen. Man beachte, dass ein 51%-Angriff auf diese Weise auch bereits in der Blockchain verankerte Blöcke nachträglich ungültig machen kann. Insbesondere kann ein Angreifer durch einen erfolgreichen 51%-Angriff auf eine Kryptowährung eine Zahlung, für die er bereits eine Gegenleistung erhalten hat, nachträglich annullieren. Auf diese Weise kann er Guthaben mehrfach ausgeben, was als *Double-Spending-Angriff* bezeichnet wird.

Die praktische Durchführbarkeit von 51%-Angriffen hängt von den finanziellen Ressourcen des Angreifers ab. Schätzungen besagen, dass 51%-Angriffe für kleinere Kryptowährungen (mit Marktkapitalisierung bis zu 30 Millionen US-Dollar) maximal einige 100 US-Dollar pro Stunde kosten würden, wenn die benötigte Hardware nicht gekauft, sondern nur kurzzeitig gemietet würde (Stand März 2019) [86]. Neben anderen Kryptowährungen wurde 2018 der Bitcoin-Fork Bitcoin Gold Opfer eines solchen Angriffs, der einen Schaden in Höhe von 18 Millionen US-Dollar verursachte [87]. Bei koordinierten Angriffen bestehender Mining-Pools müsste ebenfalls keine Hardware beschafft werden. Die reinen Stromkosten für einen 51%-Angriff auf Bitcoin werden von [88] auf ca. 210.000 US-Dollar pro Stunde geschätzt. Müsste hingegen für einen solchen Angriff auch die nötige Hardware beschafft werden, so lägen die Kosten gemäß Schätzung von [88] für Bitcoin mit etwa 7 Milliarden US-Dollar um Größenordnungen höher (Stand März 2019). Die angesprochenen koordinierten Angriffe durch Mining-Pools könnten, bedingt durch ihre geografische Konzentrierung, auch durch externe Akteure angeordnet und gesteuert werden. Das häufig vorgebrachte Argument, dass Angriffe durch die Miner aufgrund des mit PoW verbundenen Anreizsystems ökonomisch nicht sinnvoll sind, lässt außer Acht, dass ein Angreifer über finanzielle Anreize außerhalb des Blockchain-Netzwerks verfügen kann. Solche *Goldfinger*-Angriffe könnten etwa die Destabilisierung einer Kryptowährung zum Ziel haben [89].

Aufgrund von zufälligen Forks, die durch Latenzzeiten im Netzwerk entstehen, liegt die Schranke für 51%-Angriffe zudem in der Praxis bei weniger als der Hälfte der Rechenleistung [90]. Denn während ein Miner die Lösung für den aktuellen Block gefunden und an das Netzwerk gemeldet hat, sind die nicht informierten Miner weiter mit dem Errechnen der Lösung beschäftigt. Auch grundsätzlich sind erfolgreiche Angriffe mit einem geringeren Anteil an der Rechenleistung möglich. Dann führen sie jedoch nur mit einer schnell sinkenden Wahrscheinlichkeit zum Erfolg. Trotzdem kann für einen Angreifer der Erwartungswert des Gewinns positiv sein, wenn die zu verändernden Blöcke sehr wertvolle Transaktionen enthalten [91].

Eine weitere als *Selfish Mining* bezeichnete Strategie konzentriert sich auf das Anreizsystem von PoW [92]. Dabei veröffentlicht der Angreifer neu gefundene Blöcke entgegen den Protokollregeln teilweise erst mit deutlicher Verzögerung. Durch den asymmetrischen Zugang zu Informationen, wobei der Angreifer die vom übrigen Netzwerk gefundenen Blöcke kennt, dies umgekehrt aber nicht der Fall ist, vergeuden die ehrlichen Knoten Rechenleistung und somit sinkt ihr erwarteter Gewinn. Sofern sie nach rein ökonomischen Gesichtspunkten handeln, ist es für sie sinnvoll, sich dem Angreifer anzuschließen, womit schließlich ein 51%-Angriff möglich wird. Wenn die in [92] genannten und im PoW-Protokoll praktisch umsetzbaren Gegenmaßnahmen getroffen werden, sind für einen erfolgreichen Selfish-Mi-

ning-Angriff lediglich mehr als 25 % der Gesamtrechenleistung erforderlich [92].

Abbildung 9 vermittelt einen Eindruck von der Aufteilung der Rechenleistung im Bitcoin-Netzwerk auf Rechenpools, wobei eine mögliche Vernetzung der Pools untereinander und des großen Bereichs „Unbekannt“ zu hinterfragen ist.

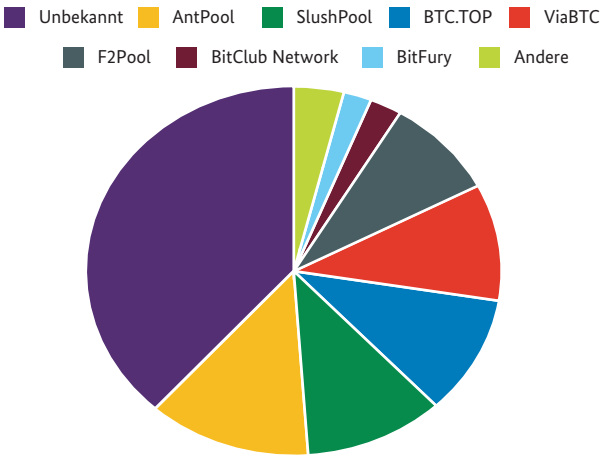

Abbildung 9: Aufteilung der Hashrate auf Mining-Pools (Durchschnitt über den Zeitraum August 2018 bis Januar 2019) [163]

Aufgrund seiner konzeptionellen Ähnlichkeit mit PoW ergeben sich für den Konsensmechanismus PoET ähnliche Angriffsmöglichkeiten. Für einen 51%-Angriff muss ein Angreifer über mindestens die Hälfte der Hardware-Sicherheitsmodule des Netzwerks verfügen. Da PoET kein Anreizsystem benötigt, ist der Selfish-Mining-Angriff nicht übertragbar. Andererseits ist das Sicherheitsmodul ein neuer Ansatzpunkt für Angriffe. Wer Schwächen an den Sicherheitsmodulen ausnutzen kann, kann somit die Blockerstellung de facto kontrollieren. Dieses Problem kann durch die Ad-hoc-Maßnahme abgemildert werden, dass Sicherheitsmodule, die viel häufiger als erwartet die niedrigste Wartezeit erzeugen, auf eine schwarze Liste gesetzt werden [24].

Auch beim Konsensmechanismus PoS ist ein Analogon zum 51%-Angriff möglich. Dafür müsste ein Angreifer einen gewissen Anteil des als Pfand hinterlegten Guthabens kontrollieren. Wie groß dieser Anteil genau ist, hängt von der konkreten Ausgestaltung ab. Er kann jedoch durchaus niedriger als 51 % sein. Auch die Frage, ob mit einem 51%-Angriff zurückliegende Blöcke verändert werden können, ist abhängig vom konkreten Design. Sofern die gewählte Ausgestaltung die Finalität von Blöcken nicht garantiert, kann ein Angreifer, der über ein entsprechendes Guthaben verfügt, ohne Zusatzkosten auch weit zurückliegende Blöcke manipulieren, sofern keine geeigneten Gegenmaßnahmen auf Netzwerkebene getroffen werden. Weitere mögliche Angriffe und Gegenmaßnahmen werden in [23] diskutiert. Grundsätzlich ist das Verfahren aufgrund seiner im Vergleich deutlich geringeren Verbreitung weniger gut untersucht als PoW.

7.1.3 Netzwerk

Eine klassische Form eines Distributed-Denial-of-Service-Angriffs (DDoS-Angriff) ist die Veranlassung einer großen Zahl von Kleinsttransaktionen, die von den Minern bearbeitet wird. Dieser DDoS-Angriff richtet sich gegen das gesamte Netzwerk und führt dazu, dass die Transaktionen ehrlicher Nutzer wesentlich langsamer bearbeitet und in die Blockchain integriert werden. Dies führt im Extremfall zum Vertrauensverlust in die Blockchain. Ein Angreifer kann dies zu seinen Gunsten ausnutzen, indem er beispielsweise auf fallende Kurse einer Kryptowährung wettet. Gegenmaßnahmen sind die Erhebung von Transaktionsgebühren und die niedrigere Priorisierung von gehäuft auftretenden Transaktionen mit demselben Absender.

Durch klassische DDoS-Angriffe oder Platzierung von Malware kann auch versucht werden, gezielt einzelne Knoten von ihrer Kommunikation mit dem Netzwerk abzuschneiden. Wird ein solcher Angriff erfolgreich z. B. gegen einen PoW-Miner angewandt, so wird dessen Rechenleistung dem Netzwerk entzogen, der Angreifer erlangt kurzzeitig einen höheren Anteil an der Gesamtrechenleistung und erhöht dadurch seine Aussicht auf einen erfolgreichen 51%-Angriff. Durch die Ausnutzung von Schwachstellen zur Platzierung einer Malware mit Remote-Code-Execution wäre sogar die Übernahme der Rechenleistung eines konkurrierenden Miners möglich. Werden Tauschbörsen oder andere mit dem Netzwerk verbundene Dienste zum Ziel, so kann ein Angreifer konkurrierenden Anbietern schaden. Im Bitcoin-Netzwerk wurden DDoS-Angriffe in der Vergangenheit mehrfach beobachtet [93].

Ein Knoten kann auch vom Netzwerk abgeschnitten werden, indem seine Kommunikation über Server des Angreifers umgeleitet wird und somit vollständig unter dessen Kontrolle gerät. Dies wird als *Eclipse-Angriff* bezeichnet. Dabei wird der Knoten bei einem Neustart der Verbindung mit dem Peer-to-Peer-Netzwerk dazu gebracht, nur Verbindungen zu Netzwerkknoten aufzunehmen, die vom Angreifer kontrolliert werden. Erreicht wird dies durch Nachrichten des Angreifers an sein Opfer, die dessen Sicht auf die verfügbaren Netzwerkknoten gezielt manipulieren.

Da dem abgeschnittenen Knoten eine funktionierende Netzwerkverbindung vorgetäuscht wird, ist er sich nicht unmittelbar bewusst, Opfer eines Angriffs zu sein. Erbringt er außerhalb der Blockchain eine Gegenleistung, kann er leicht Opfer eines Double-Spending-Angriffs werden [94]. Ein Eclipse-Angriff kann als Infrastrukturangriff oder Botnetz-Angriff gefahren werden. Diese beiden Angriffe unterscheiden sich durch die genutzten IP-Adressabschnitte. Bei einem Infrastrukturangriff hängen die IP-Adressen in Blöcken zusammen, bei einem Botnetz-Angriff ist dies nicht der Fall. Laut [94] sind 32 IP-Adressblöcke bzw. 4600 Bot-Adressen ausreichend, um in einem Netzwerk beliebiger Größe mit einer Wahrscheinlichkeit von 85 % einen erfolgreichen Eclipse-Angriff auf einen Knoten durchzuführen.

Die vorgestellten Angriffe betreffen hauptsächlich öffentliche genehmigungsfreie Blockchains. Konzeptionell ermöglicht werden sie dadurch, dass in diesem Szenario nur die Daten authentifiziert werden, jedoch die hierfür verwendeten Schlüssel keinem konkreten Kommunikationspartner zugeordnet werden (siehe Abschnitt 2.1.5). Wenn die Identität der einzelnen Knoten bekannt und durch zusätzliche Maßnahmen dokumentiert ist, sind insbesondere Eclipse-Angriffe nicht möglich.

7.1.4 Implementierung

Über die konzeptionellen technologieimmanenten Angriffsszenarien hinaus können den Entwicklern bei der praktischen Implementierung eines Blockchain-Systems Fehler unterlaufen. So wird z. B. unter [95] eine Liste bekannter Bitcoin-Schwachstellen geführt. Schwachstellen können sich aber auch absichtlich im Code befinden, etwa wenn der Betreiber einer Blockchain-Anwendung dort zu seinem Vorteil eine Backdoor einbaut wie im Fall der Kryptowährung Soarcoin [96].

Besonders kritisch sind fehlerhafte Implementierungen in den kryptografischen Routinen, da sie deren Sicherheitseignung aufheben können. So geschehen bei Bitcoin, wo Fehlimplementierungen des ECDSA es erlaubten, den privaten Schlüssel zu berechnen. Die Ursache lag darin, dass die bei der Signierung verschiedener Transaktionen verwendeten Kurzzeitschlüssel *(ephemeral keys)* zu kurz oder nicht unabhängig voneinander waren oder sogar mehrfach genutzt wurden [97], [98]. Neben solchen Erzeugungsschwächen können auch Seitenkanalangriffe während der Signaturerzeugung helfen, Schlüssel zu erlangen (siehe Abbildung 10). Bei Fehlimplementierung der Hashfunktion kann es dazu kommen, dass Angriffe auch auf eine theoretisch sichere Hashfunktion möglich sind (siehe Abschnitt 7.1.1).

Als Implementierungsschwäche kann es auch angesehen werden, wenn sicherheitsrelevanten Protokollelementen zu viel Spielraum und Beeinflussung durch den Anwender eingeräumt wird. Beispielsweise hängt die Sicherheit des PoW-Verfahrens von der Schwierigkeit des Rätsels ab, die sich üblicherweise nach den Zeitabständen zwischen den letzten Blockveröffentlichungen richtet. Ist es nun möglich, den Blockzeitstempel zu manipulieren, so kann seitens des Angreifers suggeriert werden, dass zwischen der Erzeugung zweier Blöcke mehr Zeit vergangen ist, als es der Realität entspricht. Daher wird der Schwierigkeitsgrad herabgesetzt und der Angreifer kann in kurzen Zeitabständen neue Blöcke jeweils im Gegenzug zur Belohnung erzeugen. Bei der Kryptowährung Verge wurde so beispielsweise in einem Fall drei Stunden lang jeweils ein Block pro Sekunde erzeugt und es wurden insgesamt 1,3 Millionen Euro an den Angreifer ausgezahlt [99].

Im weiteren Sinne ist auch der Transaction-Malleability-Angriff eine Implementierungsschwäche. Dieser ist möglich, wenn ein und dieselbe Operation durch unterschiedliche Befehle der Blockchain-Skriptsprache umgesetzt werden kann.

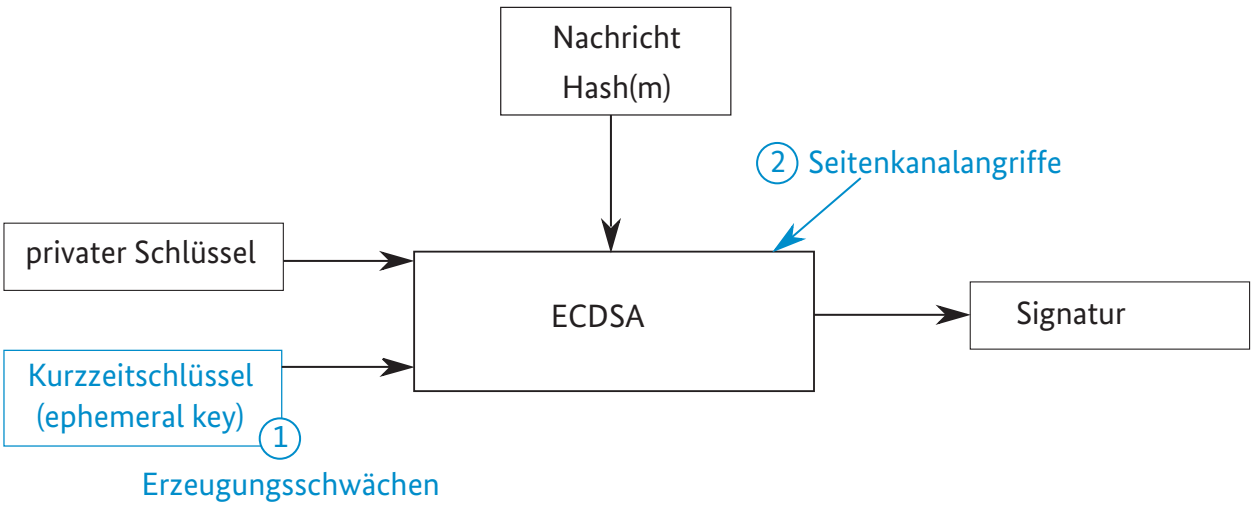

Abbildung 10: Implementierungsschwächen können kryptografische Funktionen (hier ECDSA) angreifbar machen.

Somit werden bei der Weitermeldung an andere Knoten Blöcke mit logisch gleichem Inhalt, aber unterschiedlichen Hashwerten ermöglicht, was ein Angreifer zu seinen Gunsten ausnutzen kann [100].

Auch die Firmware eingesetzter Hardware-Komponenten kann Sicherheitslücken aufweisen. Ein Beispiel bietet ein Vorfall im Bereich der Mining-Hardware bei Bitcoin, der dem Hersteller Bitmain das Abschalten von Geräten per Fernzugriff ermöglichte. Da die Firma der mit Abstand größte Produzent derartiger Hardware ist, wären durch Ausnutzung der Schwachstelle 51%-Angriffe auf die Bitcoin-Blockchain möglich gewesen [101], [102].

Die hier geschilderten Implementierungsfehler betreffen den Kern des Blockchain-Systems. Von einzelnen Anwendern programmierte Zusatzfunktionalitäten – wie Smart Contracts – müssen zusätzlich betrachtet werden (siehe Abschnitt 7.2.1).

7.2 Sicherheit von Smart Contracts

Für Smart Contracts bedeutet die Unveränderlichkeit der Blöcke, dass der einmal in der Blockchain verankerte Programmcode im Nachhinein nicht mehr modifiziert werden kann. Eine fehlerfreie Programmierung ist deshalb von essentieller Wichtigkeit. Untersuchungen zeigen aber, dass ein großer Teil der Smart Contracts mit zum Teil elementaren Fehlern behaftet ist, manche Studien beziffern ihren Anteil auf etwa 45 % bei Ethereum [103].

Die Auswirkungen fehlerhafter Contracts sind unterschiedlich. Manche ungenaue Programmierung sorgt schlicht für Unfairness zwischen den Vertragspartnern (siehe z. B. Abschnitt 4.3.2). Betroffen sind in der Regel nur die direkt beteiligten Parteien. In vielen Fällen führen Fehler jedoch dazu, dass Angreifern unbeabsichtigterweise Zugriff auf Funktionen oder Daten eines Contracts gewährt wird und so z. B. das unberechtigte Abbuchen von Geldern möglich wird. Andere Fehler können dazu führen, dass ein Contract nicht mehr ansprechbar ist und damit auch das Vertragsguthaben unwiderruflich eingefroren ist. Auf diese Weise sind in der Vergangenheit bereits Millionenschäden verursacht worden, und auch vermeintlich unbeteiligte Contracts wurden in Mitleidenschaft gezogen (siehe Parity-Hack, Abschnitt 7.2.1).

Im schlimmsten Fall können Fehler in einzelnen Contracts gesamte Systeme unterminieren, einerseits durch Einfrieren von Funktionalität – wie z. B. ein schlichter Tippfehler, der im Juni 2018 vorübergehend die gesamte Kryptowährung ICON lahmgelegt hat [104] – oder andererseits durch Vertrauensverlust in die Blockchain – wie z. B. der prominente DAO-Hack, der einen Hardfork der Ethereum-Blockchain mit Rückabwicklung vieler Transaktionen zur Folge hatte.

Eine gute Übersicht über Sicherheitsrisiken speziell von Ethereum-Contracts liefert [105], des Weiteren auch [106]; Schwachstellen speziellerer Art finden sich in [107]. Im Folgenden werden einige Sicherheitsrisiken im Lebenszyklus eines Smart Contracts erläutert.

7.2.1 Elementare Programmierfehler

Die Anfälligkeit einer Sprache für Programmierfehler hat nicht unerheblichen Einfluss auf die Sicherheit des Codes. Unausgereifte oder neu entwickelte Programmiersprachen weisen hier in der Regel größere Schwachstellen auf als etablierte. So hat die eigens für Ethereum entworfene Sprache Solidity wegen eines inkonsistenten Designs Kritik auf sich gezogen, und von der Verwendung der einst als Alternative geltenden Sprache Serpent wurde wegen zahlreicher Designmängel bereits abgeraten [108].

Neben den klassischen Fallstricken in der Programmierung ist auch das Verwenden hardcodierter Adressen oder Parameter riskant, da wegen der Unveränderlichkeit des Programmcodes keine Anpassung – etwa bei Ungültigwerden – erfolgen kann. Ein Vorfall auf der Ethereum-Blockchain im Jahr 2017 (Parity-Hack [109]) illustriert das Zusammentreffen mehrerer dieser Schwachstellen: Hier wurde durch eine fehlerhafte Initialisierung die versehentliche – oder auch bewusste – Deaktivierung eines Contracts möglich. Darauf aufbauende Smart Contracts, die seine Adresse hardcodiert beinhalteten, konnten davon nicht abrücken, waren demzufolge nicht mehr funktionsfähig und ihr Vermögen von über 150 Millionen US-Dollar ist für immer eingefroren.

Verschiedene Forschergruppen beschäftigen sich mit dem automatisierten Auffinden von Fehlern im Programmcode [110] und den Möglichkeiten einer formalen Verifikation von Smart Contracts [103], [111], [106]. Da grundsätzlich mit der Möglichkeit zu rechnen ist, dass der Compiler nicht vertrauenswürdig arbeitet oder Schadprogramme gezielt erst in den Bytecode eingebracht wurden, vertreten insbesondere die Autoren von [111] die Ansicht, dass eine Überprüfung des Bytecodes einer bloßen Analyse des Quelltextes vorzuziehen ist. Es bedarf aber noch weiterer Arbeit in diesem Bereich, vor allem auch für Blockchains jenseits von Ethereum.

7.2.2 Blockerstellung

Ist ein Contract erstellt, wird er in die Blockchain eingebracht. Dabei entscheidet nicht der Programmierer des Contracts, sondern in der Regel die Miner (oder ein *ordering service*, siehe Abschnitt 4.1) darüber, in welcher Reihenfolge und zu welcher Zeit Transaktionen einem Block hinzugefügt werden. Smart Contracts sollten deshalb nicht von einer speziellen Ausführungsreihenfolge oder dem Zeitstempel abhängen [103] (siehe auch Abschnitt 4.3.2). Ferner gilt die für Bitcoin ausgesprochene Empfehlung, eine gewisse Zeit abzuwarten, bis eine getätigte Transaktion tief genug in der Blockchain verankert ist, in gleicher Weise auch für Smart-Contract-Plattformen, wenn sie einen Konsensmechanismus verwenden, der Forks zulässt.

Contracts, deren Ausführung mit einem großen Rechenaufwand verbunden ist, sind anfällig für das sogenannte *Verifier's Dilemma* [112]. Hier stellt sich jedem Knoten, der diesen Contract validieren müsste, die Frage, ob er die nötigen Ressourcen tatsächlich aufbringen möchte oder den Contract ungeprüft akzeptiert. Bei zu hohen Kosten besteht die Gefahr, dass ein nicht unerheblicher Anteil der Knoten sich für das ungeprüfte Akzeptieren des Contracts entscheidet. Auf diese Weise könnten Contracts in die Blockchain gelangen, die z. B. unautorisierte Zahlungen vornehmen, nicht mit dem Systemstatus in Einklang stehen und die gesamte Glaubwürdigkeit des Systems untergraben. Eine angemessene Ausgestaltung des Anreizsystems (siehe Abschnitte 3.2.5 und 4.2.4), das auch Validierungsaufwände entlohnt, oder die Möglichkeit, ungeprüfte Ergebnisse anzuzweifeln [113], könnten das Dilemma abschwächen.

7.2.3 Abfangen von unerwünschtem Laufzeitverhalten

Eine große Herausforderung ist der vorausschauende Umgang mit unvorhergesehenem Verhalten,

das während der Ausführung des Contracts zu Tage treten könnte.

Der Programmierer muss sich darüber im Klaren sein, an welchen Stellen im Code er sich im Falle von Laufzeitfehlern auf ein automatisches Auslösen von Exceptions verlassen kann und an welchen Stellen ein manuelles Abfangen nötig ist. Die Antwort ist abhängig von der Programmiersprache und nicht immer offensichtlich. So ist Solidity lückenhafte Exception-Behandlung in die Kritik geraten, weil ausgerechnet die Laufzeitfehler in häufig verwendeten Operationen nicht automatisch abgefangen werden [105].

Es ist darauf zu achten, ob ein Contract an irgendeiner Stelle fremden Code aufruft und somit die Kontrolle über die Ausführung (vorübergehend) an eine andere Instanz abgibt. Wie der Fall von Solidity zeigt, kann dieser Aufruf von Fremdcode geschehen, ohne dass dem Programmierer dies bewusst zu sein braucht: Die gängigen Operationen zur Übertragung von Guthaben rufen implizit die sogenannte *fallback*-Funktion des Empfängers auf, die nicht unter der Kontrolle des aufrufenden Contracts steht. Insbesondere kann diese Funktion Programmcode enthalten, der dem aufrufenden Contract Schaden zufügt.

Unter *re-entrancy* versteht man die Möglichkeit, dass eine Funktion in einem Smart Contract erneut aufgerufen wird, bevor ihr vorheriger Aufruf abgeschlossen ist. Ermöglicht wird dies beispielsweise, wenn wie oben geschildert ein Contract die Kontrolle an einen anderen Vertrag abgibt, der die ursprüngliche Funktion erneut aufruft und sich so in eine Endlosschleife begibt. Auch wenn von Seiten des Blockchain-Systems Vorkehrungen getroffen sind, Schleifen irgendwann automatisch abzubrechen – bei Ethereum ist das der Fall, sobald das Gas-Budget aufgebraucht ist –, kann bis dahin unter Umständen schon großer Schaden angerichtet sein. Im prominenten DAO-Hack wurde genau diese Sicherheitslücke ausgenutzt [105].

7.2.4 Sichtbarkeit von Smart Contracts

Um einen Smart Contract auszuführen, müssen ihm in einer Transaktion die nötigen Inputdaten zugesandt werden. Nun sind in öffentlichen Blockchains Transaktionen in der Regel für alle einsehbar. Ohne weitere Sicherheits- und/oder Verschlüsselungsmaßnahmen kann also kein privater Aufruf des Contracts erfolgen. Direkt ausnutzen lässt sich die Vertraulichkeitslücke beispielsweise in Glücksspielverträgen, wenn der zweite Spieler den Einsatz des ersten mitlesen und zu seinem Vorteil darauf reagieren kann [114].

Etwas subtiler sind die Auswirkungen auf die Vertraulichkeit sogenannter privater Variablen, wie sie beispielsweise Solidity zur Verfügung stellt. Das sind Variablen, die Contracts in bestimmten Speicherbereichen ablegen, wo sie vor dem Zugriff von anderen Contracts geschützt sind. Auf welchen Wert eine solche Variable gesetzt werden soll, wird aber in einer Transaktion mitgeteilt, die wiederum öffentlich sichtbar in der Blockchain steht. Trotz ihres Namens ist der Inhalt privater Variablen also öffentlich nachvollziehbar.

Die Unveränderlichkeit der Blockchain verhindert auch das nachträgliche Löschen von Contracts. Zwar wird gelegentlich behauptet, in Ethereum gebe es mit dem `selfdestruct`-Befehl die Möglichkeit, Smart Contracts zu löschen, aber dabei wird nicht der Bytecode aus der Blockchain entfernt, sondern nur dessen Verknüpfung mit einer speziellen Ethereum-Adresse aus dem Systemstatus aufgehoben. Die Knoten betrachten den Contract fortan als nicht mehr existent und rufen ihn nicht auf, aber in der Blockchain-Historie lässt sich sein Bytecode nach wie vor einsehen. Dieses Konzept ist also nicht geeignet, um das im Datenschutz verankerte Recht auf Vergessenwerden (siehe Abschnitt 9.3.2) umzusetzen.

7.3 Angriffe auf die Blockchain-Infrastruktur

7.3.1 Virtuell-Real-Schnittstelle

Wie bereits in Abschnitt 2.1 angesprochen gelten die Sicherheitsgarantien einer Blockchain (wie bei anderen Technologien zur Datenhaltung) erst ab dem Zeitpunkt, zu dem die Daten in einen Block integriert werden. Sofern Daten und Ereignisse aus der realen Welt außerhalb der Blockchain ver-

wendet werden, um Änderungen des Zustands der Blockchain zu bewirken, wie dies insbesondere im Kontext von Smart Contracts der Fall sein kann, kann ein Angriff auf die Sicherheit der Daten vor Übernahme in die Blockchain für einen Angreifer attraktiv sein.

Auch Orakeldienste können dieses Problem nicht vollständig lösen, da bereits die von ihnen abgerufenen Daten manipuliert worden sein könnten (siehe Abschnitt 4.3.1). Bei der teilweise vorgeschlagenen Einspeisung von Sensordaten in die Blockchain sind zudem gezielte Manipulationen der von den Sensoren erfassten physikalischen Daten denkbar. Angriffe dieser Art sind in der Regel schwer zu detektieren.

7.3.2 Schlüsselsicherheit

Über die kryptografischen Verfahren und die technische Umsetzung hinaus stellt die Schlüsselerzeugung und -verwaltung einen wesentlichen Angriffspunkt bei Blockchain-Anwendungen dar (siehe Abbildung 11). Die Angriffe sind dabei vergleichbar zu denen auf andere Dienste wie z. B. Online-Banking. Übersichtsartikel wie [115] veranschaulichen, mit welchem Erfolg diese Schwachstellen bereits auf das Blockchain-Umfeld übertragen und ausgenutzt wurden.

Zum einen können qualitativ minderwertige Schlüssel zum Einsatz kommen (siehe Abschnitt 5.3). Beispielsweise ist ein im Bereich der Kryptowährungen häufig angewandtes Verfahren die Verwendung eines Brain-Wallets zur Erzeugung der Schlüssel. Dabei wird eine Hashfunktion auf ein natürliches Passwort angewendet. Mittels eines Brute-Force-Angriffs können die so erzeugten Schlüssel jedoch vom Angreifer aufgedeckt werden, wenn der Eingangswert für die Hashfunktion eine zu geringe Entropie hat [115].

Die Schlüsselerzeugung und -verwaltung sowie die Interaktion zwischen Nutzern und Blockchain erfolgen in der Regel über Softwareprodukte (z. B. Wallets) von Drittanbietern, zu denen meist keine Sicherheitsgarantien vorliegen. Beispiele aus dem Bereich der Kryptowährungen zeigen, dass solche Software durchaus absichtliche Hintertüren aufweisen kann. In einem solchen Fall wurde unter der Adresse iotaseed.io ein Dienst zur Schlüsselerzeugung zur Verfügung gestellt, der sechs Monate lang vermeintlich korrekt funktionierte. Während dieser Zeit wurden die erzeugten Zugangsdaten jedoch erfasst, sodass schließlich das gesamte Guthaben aller Geldbörsen an die Angreifer floss [116].

Unbeabsichtigte Schwachstellen können mittels der Platzierung von Malware ausgenutzt werden. Welche Stellen sich einem Angreifer dazu bieten, hängt von der Infrastruktur des gesamten

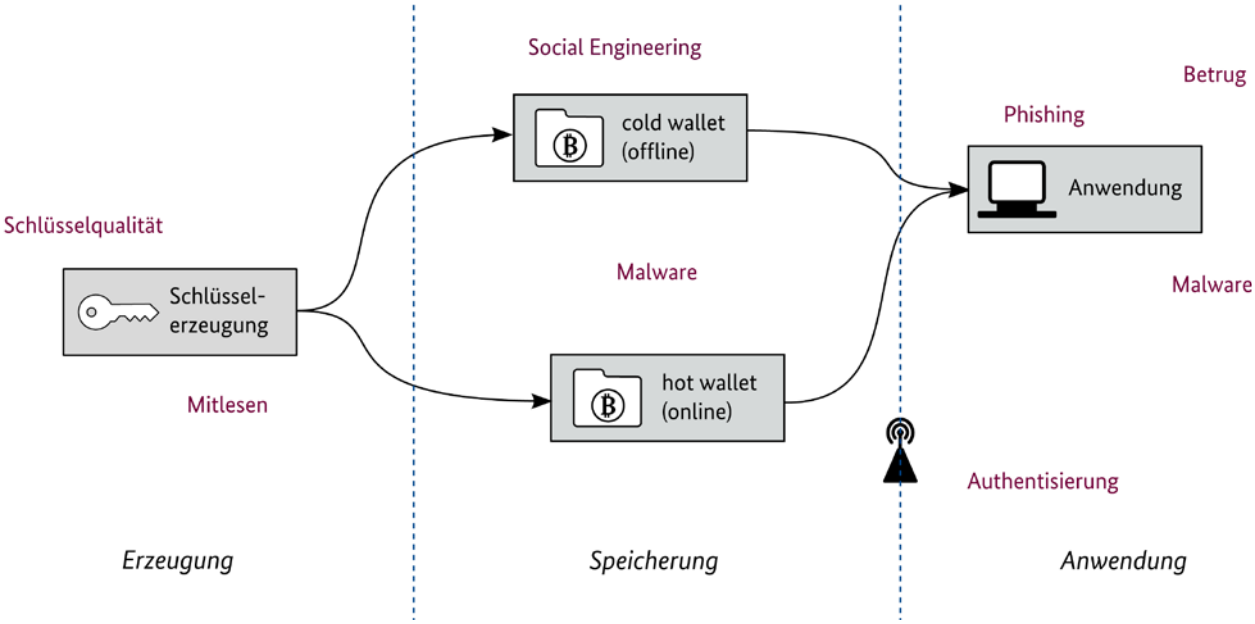

Abbildung 11: Praktische Angriffe auf Schlüssel

Blockchain-Systems ab. Je nach Ausgestaltung können sich Malware-Angriffe gegen den Nutzer oder auch direkt gegen einen Partnerdienst (wie Wallet-Betreiber, Kryptotauschbörsen) richten. Bekannte Beispiele sind die Kampagnen Dridex und Trickbot, denen Funktionen zum Diebstahl von Kryptowährungen hinzugefügt wurden. Malware wie CryptoShuffler und Evrial las die Zwischenablage aus, um hier Zugangsdaten auszuspähen [115]. Der Vorfall mit der höchsten bekannten Schadenssumme war die Platzierung eines Info-Stealers bei der Handelsplattform Coincheck. Hierbei wurden Coins der Kryptowährung NEM in einem Gegenwert von 532 Millionen US-Dollar von 260.000 Konten gestohlen [117].

Weiterhin kann bei der Übertragung der sensiblen Daten eingegriffen werden. Indem der Angreifer eine DNS-Anfrage für die Adresse einer Kryptobörse mit der Adresse seiner Website beantwortet, erhält er die Schlüssel direkt von seinem Ziel.

7.3.3 Betrug

Ein Angreifer benötigt nicht notwendigerweise Kenntnis über den privaten Schlüssel seines Opfers. Stattdessen kann ein Nutzer dazu gebracht werden, ungewollt einen legitimen Eintrag im Sinne des Angreifers in der Blockchain zu veranlassen. Dabei können klassische Betrugsmethoden zum Einsatz kommen, im Blockchain-Umfeld oft in Form sogenannter ICO-Scams. Bei diesen werden Investitionen in eine neue Kryptowährung unterschlagen. Der Vorfall mit dem größten bekannten Schaden in diesem Bereich ereignete sich im April 2018, in dem Investitionen in die Währungen Pincoin und Ifan in einem Gegenwert von insgesamt 536 Millionen Euro einbehalten wurden [118]. Auch können – z. B. durch Kompromittierung einer Website – öffentliche Schlüssel gefälscht werden, an die im Gegenzug zu einer Dienstleistung Gelder überwiesen werden sollen.

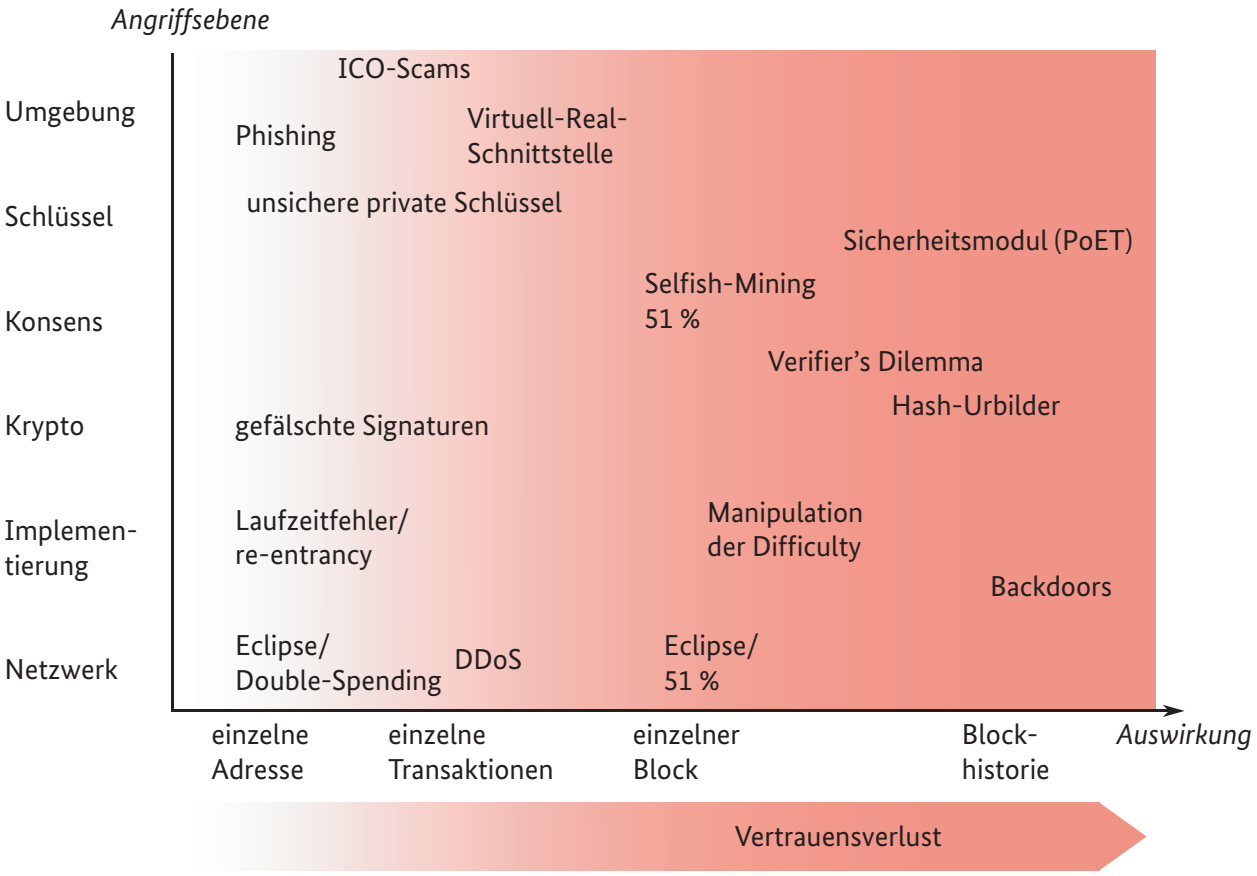

Abbildung 12: Übersicht über Angriffe auf Blockchain-Systeme

Darüber hinaus kann ein Angreifer sein Opfer z. B. durch Social Engineering oder Phishing-Angriffe zur Preisgabe des privaten Schlüssels verleiten. Alle diese Angriffe kompromittieren nicht die Sicherheit des zugrunde liegenden Blockchain-Systems.

7.3.4 Allgemeine Aspekte der IT-Sicherheit

Auch beim Einsatz von Blockchains müssen die üblichen Maßnahmen zur Gewährleistung von IT-Sicherheit umgesetzt werden. Sie werden allein durch die Verwendung von Blockchains keinesfalls obsolet, wie bereits in [1] festgestellt wurde. Es folgt eine kurze Auflistung der verschiedenen allgemeinen Aspekte und Maßnahmen, die berücksichtigt und umgesetzt werden müssen, ohne dass dabei ein Anspruch auf Vollständigkeit erhoben wird:

- Sicherheitsmanagement: Sicherheitskonzept; Dokumentation, Prüfung und Aktualisierung der Maßnahmen
- Notfallmanagement: Notfallplan mit Rollen und Sofortmaßnahmen
- Infrastruktur: Brandschutz, (redundante) Stromversorgung
- Netzsicherheit: Absicherung des Internetzugangs, Firewalls
- Computersicherheit: Zugriffsschutz, sichere Grundkonfiguration, Virenschutz
- Laufender Betrieb: Protokollierung inkl. Monitoring, Updates und Patches, Datensicherung
- Anwendungsebene (genehmigungsbasierte Blockchains): Rollen- und Rechtemanagement mit Zugriffskontrolle, Identifizierung der Teilnehmer
- Schlüsselmanagement: Sichere Erzeugung, Speicherung (mit Zugriffsschutz) und Löschung

Welche Maßnahmen genau getroffen werden sollten, hängt davon ab, wie hoch der konkrete Schutzbedarf eingeschätzt wird. Für eine ausführliche Darstellung wird auf den IT-Grundschutz des BSI [119] verwiesen.

Zusammenfassung.

- Angriffe sind auf verschiedene Komponenten des Blockchain-Systems möglich und können gravierende Auswirkungen haben.
- Die Eigenschaften des Netzwerks haben unmittelbaren Einfluss auf die Sicherheit des gesamten Blockchain-Systems.
- Die praktische Implementierung kann Algorithmen angreifbar machen.
- Die Unveränderlichkeit des Codes von Smart Contracts und ihre automatisierte Ausführung erfordern höchste Sorgfalt bei der Programmierung.
- Eine Vielzahl von Sicherheitslücken in Smart Contracts ist bekannt und in der Praxis beobachtet.
- Allgemeine Maßnahmen der IT-Sicherheit bleiben weiterhin unentbehrlich.

Teil III Recht

8 Rechtliche Aspekte

Die Blockchain-Technologie bringt neben technischen auch rechtliche Fragen mit sich. Ausgehend davon, dass die Mehrzahl ihrer Nutzer sich kaum mit den rechtlichen Rahmenbedingungen dieser Technologie auseinandersetzt, soll hier eine erste Orientierung über zivil- und strafrechtliche Aspekte gegeben werden, wie sie sich auf Basis der gegenwärtigen Rechtslage darstellen. Da die rechtliche Betrachtung allerdings keinen Schwerpunkt dieses Dokuments bildet, sind die nachfolgenden Ausführungen lediglich zur Verschaffung eines ersten Überblicks gedacht. Aspekte des Datenschutzes werden im Kapitel 9 behandelt.

8.1 Zivilrechtliche Aspekte

Eine Blockchain ist darauf angelegt, sämtliche Transaktionen unveränderlich und vor Manipulationen geschützt abzuspeichern. Hierin liegt das Potenzial dieser Technologie, gleichzeitig ergibt sich aus diesem Prinzip der Unveränderlichkeit eine Vielzahl zivilrechtlicher Fragestellungen, die bislang nicht abschließend geklärt sind.

So stellt sich beispielsweise die Frage, was bei einer fehlerhaften Transaktion passiert. Kann der fälschlich oder auch irrend transferierende Teilnehmer die Transaktion noch rückgängig machen? Kann er die Anfechtung erklären oder gar zurücktreten? Weiterhin stellt sich die Frage, gegenüber wem eine dieser Erklärungen abzugeben wäre.

Eine Lösungsmöglichkeit bestünde darin, die zivilrechtlichen Instrumente bei Verwendung einer Blockchain auszunehmen. Anfechtung, Rücktritt oder Widerruf müssten dann außerhalb des Blockchain-Systems durchgesetzt werden. Hier tritt aber auf öffentlichen genehmigungsfreien Blockchains das Problem hinzu, dass die Teilnehmer der Blockchain zumindest unter Pseudonymen arbeiten, der sich irrende Teilnehmer mithin im Zweifel gar nicht weiß, wen er als Gegenüber hat oder in welchem Rechtsraum sich dieses befindet.

Auch andere Rechtsbereiche des Zivilrechts, die den Schutz der Vertragsschließenden bezwecken, werden ausgehebelt. Gesetzliche Verbote oder die Sittenwidrigkeit eines Vertrags ziehen üblicherweise die Unwirksamkeit dessen nach sich. Ob allerdings einer der vorgenannten Fälle vorliegt, ist einzelfallabhängig und eine Wertungsfrage, die durch die Logik eines Smart Contracts nicht beantwortet werden kann [120].

Ebenfalls von einer Blockchain nicht beachtet ist der Minderjährigenschutz. Sieht das Gesetz nach §§ 107 ff. BGB bis zur Genehmigung durch einen der Erziehungsberechtigten die schwebende Unwirksamkeit des Vertrages vor, so sind einmal getätigte Transaktionen in der Blockchain verbucht. Einen Schwebezustand kennt die Blockchain indes nicht [120], [121].

Zwar sind Rückabwicklungen möglich, dies aber nur unter Mitwirkung des anderen. Verweigert indes der ursprüngliche Empfänger der fehlerhaften Transaktion seine Mitwirkung, könnte absichtlich ein Hardfork herbeigeführt werden, mithilfe dessen der fehlerhafte Block und die ursprüngliche Transaktion ersetzt werden. Hierzu müsste allerdings eine ausreichende Anzahl der Knoten, die am Konsensmechanismus beteiligt sind, davon überzeugt werden, dass ein bestimmter Block als fehlerhaft angesehen wird [122] (siehe auch Abschnitt 9.3.2). Dies könnte sich als schwierig oder unpraktikabel herausstellen, vor allem dann, wenn es mehrere Transaktionen betreffen sollte. Hinzu kommt, dass dann auch alle anderen Transaktionen, die sich zufälligerweise in denselben rückabgewickelten Blöcken befinden, ungültig werden, auch wenn sie inhaltlich gar nichts mit der angefochtenen Transaktion zu tun haben. Dieser Umstand kann auch rechtliche Auswirkungen nach sich ziehen und möglicherweise zur Aufhebung oder Rückabwicklung unstrittiger Transaktionen führen. Dies dürfte nicht nur dem Prinzip der Blockchain an sich, sondern auch dem Interesse der ebenfalls betroffenen Teilnehmer widersprechen.

8.2 Virtuell-Real-Schnittstelle

Ungeachtet der Frage, welche technischen Errungenschaften mit einer Blockchain erzielt werden können, ist der möglicherweise rechtlich tangierte Hintergrund nicht zu vernachlässigen.

So werden Blockchains insbesondere dazu eingesetzt, eine nachträgliche Veränderung der Daten auszuschließen. Was allerdings im Sinne der IT-Sicherheit kaum beachtet wird, ist die Frage, ob die aus der realen Welt in eine Blockchain eingepflegten Daten auch valide sind, also mit der Realität übereinstimmen. Zwar mögen Blockchains garantieren können, dass die Daten im Nachhinein nicht verändert worden sind. Eine Überprüfbarkeit der Validität dieser Daten bleibt der Nutzergemeinschaft allerdings vorenthalten [123]. Wie in Abschnitt 7.3.1 angemerkt, dürfen daher mögliche Angriffe auf die Virtuell-Real-Schnittstelle nicht außer Acht gelassen werden. Somit ist festzuhalten, dass eine Blockchain zwar mehr Transparenz und Nachvollziehbarkeit der Daten schafft, diese Eigenschaften aber an sich nur dann etwas nützen, wenn die eingetragenen Daten valide sind. Um das sicherzustellen, bräuchte es möglicherweise wiederum eine Instanz, die die Validität der Daten kontrolliert.

Das Problem der Validität der Daten ist eng verbunden mit der Authentisierung der Knoten im Sinne der Zuordnung der Schlüssel zu einem konkreten Kommunikationspartner (siehe Abschnitt 2.1.5). Ohne eine solche Authentisierung derjenigen Knoten, die Daten über die Virtuell-Real-Schnittstelle einspeisen oder auch die Validität von Daten kontrollieren sollen, lässt es sich vermutlich nicht lösen.

Vor diesem Hintergrund dürfte sich ein Großteil der Überlegungen zur Ersetzung staatlicher Register jedenfalls hinsichtlich der Frage erübrigen, ob es trotz Verwendung einer Blockchain eines Intermediärs bedarf. Während beispielsweise das Grundbuch selbst in Form einer Blockchain geführt werden könnte, bedarf es nach hier vertretener Auffassung weiterhin eines zuverlässigen Intermediärs, der sicherstellt, dass die Grundbucheintragungen valide sind.

Umgekehrt stellt sich ebenso die Frage nach der Rechtsgültigkeit und Durchsetzbarkeit der Einträge einer Blockchain, wenn Gegenstände in der realen Welt betroffen sind. Im Gegensatz zu zentralisierten Lösungen wie Datenbanken, in denen Missbrauch und Manipulation von Daten in erster Linie durch organisatorische Maßnahmen und rechtliche Rahmenbedingungen verhindert werden (siehe Abschnitt 2.3), adressieren Blockchains zwar diese Probleme direkt auf technischer Ebene. Genau wie bei alternativen Technologien kann aber z. B. eine in der Blockchain festgeschriebene Übertragung von Werten außerhalb der Blockchain allein technisch deren praktische Realisierung nicht sicherstellen. Vielmehr wird hierzu ein auf ausreichendem Niveau funktionierendes Rechtssystem erforderlich sein. Vor diesem Hintergrund müssen Vorschläge zum Einsatz von Blockchains in Staaten ohne funktionierendes Rechtssystem, die durch Manipulationssicherheit bei der Datenhaltung und Transparenz Abhilfe für dieses Problem versprechen, kritisch auf ihren tatsächlichen Nutzen geprüft werden.

8.3 Strafrechtliche Aspekte

Neben zahlreichen zivilrechtlichen Fragen stellt sich die Frage nach einer möglichen Strafbarkeit eines Knotenbetreibers, sofern die Blockchain zur Speicherung strafrechtlich relevanter Inhalte missbraucht wird.

Die Frage nach einer möglichen Strafbarkeit von Knotenbetreibern kam im Jahr 2018 auf, als Forscher unter anderem kinderpornografische Inhalte auf der Bitcoin-Blockchain entdeckten [124]. Das Strafgesetzbuch stellt unterschiedliche Tathandlungen, wie zum Beispiel das Zugänglichmachen oder auch nur den Besitz solcher Inhalte, unter Strafe (§§ 184 ff. StGB). So wird ein Zugänglichmachen bereits dann angenommen, wenn die Möglichkeit eines Zugriffs eröffnet wird[1]. Im Internet reicht bereits das Ermöglichen des Internetzugriffs aus, es kommt also nicht einmal darauf an, ob die Inhalte tatsächlich weitergegeben worden sind, die schlichte Möglichkeit

[1] BGH, Beschl. v. 12.11.2013 – 3 StR 322/13

genügt[2]. Da die Nutzer der Blockchain die Inhalte herunterladen und innerhalb des jeweiligen Netzwerks wieder zur Verfügung stellen, ist ein Zugänglichmachen im Sinne der vorgenannten Strafvorschriften zunächst einmal anzunehmen. Allerdings geht die Rechtsprechung davon aus, dass rechtswidrige Inhalte nur dann im Sinne der §§ 184 ff. StGB zugänglich gemacht werden, wenn es an der Überwindung einer „gewissen Barriere" fehlt[3]. Eine solche Barriere wird in der Regel angenommen, wenn die Inhalte nur mit einem nicht unbedeutenden Aufwand zu finden sind (so auch [125]). So liegt es hier. Zwar waren die Daten in der Blockchain gespeichert, sie waren aber nicht offensichtlich erkennbar. Die Verwirklichung des Tatbestandsmerkmals Zugänglichmachen dürfte daher im Ergebnis abzulehnen sein.

Auch der Besitz kinderpornografischer Inhalte dürfte im vorliegenden Beispiel ausscheiden. Rechtlich wird der Besitz als ein tatsächliches Herrschaftsverhältnis verstanden, das darüber hinaus von einem Besitzwillen getragen werden muss[4]. Ausgehend von dieser Definition kommt es also darauf an, ob der Knotenbetreiber neben dem tatsächlichen Besitz auch den entsprechenden Besitzwillen hat, der wiederum auf die Möglichkeit einer ungehinderten Einwirkung abzielen muss. Die bloße Kenntnis vom Vorhandensein einer Schrift oder das bloße Dulden des Besitzes eines anderen dürfte daher alleine nicht genügen[5], um sich im Sinne der hier genannten Vorschriften strafbar zu machen. Hier wurde der Fall der Täterschaft betrachtet, andere Beteiligungsformen blieben außen vor.

Im Ergebnis kann die Möglichkeit einer Strafbarkeit nicht pauschal ausgeschlossen werden, sie wird aber in der Regel entweder bereits an einem der erforderlichen Tatbestandsmerkmale oder aber am Vorsatz scheitern. Eine gefestigte Rechtsprechung zu diesem Thema existiert bislang nicht. Auch die bisherige Rechtsprechung zur Nutzung von Internet-Tauschbörsen wird nicht auf die Blockchain übertragbar sein. Das OLG Oldenburg hat beispielsweise mit Beschluss vom 08.05.2009 (Az.: 1 Ss 46/09) entschieden, dass ein Nutzer von Internet-Tauschbörsen nicht zwingend damit rechnen müsse, durch den Download angebotener Dateien diese zugleich wieder anderen Nutzern aufgrund der Programmierung der Tauschbörse zur Verfügung zu stellen. Nutzer einer (öffentlichen) Blockchain können dieses Argument hingegen nicht für sich beanspruchen, da die Blockchain-Technologie gerade darauf abzielt, Inhalte Dritten zur Verfügung zu stellen. Da sich einzelne Inhalte zudem nicht einfach aus der Blockchain löschen lassen (siehe Abschnitt 9.3.2), nehmen die meisten Nutzer die Weiterverbreitung der Inhalte vermutlich in Kauf, da anderenfalls eine Nutzung der Blockchain nicht mehr möglich wäre. Insofern wird abzuwarten sein, wie sich die Rechtslage zukünftig entwickelt.

Grundsätzlich betreffen Fragen im Zusammenhang mit der Speicherung strafrechtlich relevanter Inhalte vorrangig öffentliche genehmigungsfreie Blockchains. In privaten und genehmigungsbasierten Blockchains lässt sich bereits durch das Rechtemanagement das Einstellen derartiger Inhalte verhindern oder zumindest stark einschränken. Die Tatsache, dass die Strafverfolgung aufgrund der besseren Identifizierbarkeit der Teilnehmer deutlich erleichtert wird, sollte zudem abschreckend wirken. Weiterhin ist prinzipiell eine spätere Löschung vorstellbar, siehe Abschnitt 9.3.2.

8.4 Smart Contracts

Smart Contracts unterliegen der gleichen Problematik wie Kryptowährungen, was Verantwortlichkeiten, Haftungsfragen, Datenschutz (siehe Kapitel 9) etc. angeht. Wie in Kapitel 4 angesprochen, handelt es sich bei ihnen nicht um Verträge im rechtlichen Sinne, sondern um vorprogrammierte Prozesse. Allerdings sind in anderen Jurisdiktionen – z. B. in den US-Bundesstaaten Arizona und Tennessee – bereits Gesetze verabschiedet worden, die Smart Contracts die gleiche

[2] BGHSt 47, 55, 60 mit Anm. Gehrke ZUM 02, 283, Kudlich JZ 02, 310 und Lindemann/Wachsmuth JR 02, 206.

[3] BGH, Urt. v. 22.05.2003, Az.: 1 70/03; BVerwG, Urt. v. 20.02.2002 – Az.: 6 C 13/01.

[4] BT-Drs. 12/3001 S. 5 unter Hinweis auf den Besitztatbestand des § 29 Nr. 3 BtMG und dazu BGH 26, 117; 27, 380; 30, 279.

[5] Schönke/Schröder StGB, 29. Aufl, 2014, § 184 b Rn. 15 m.w.N.

Bedeutung wie juristischen Verträgen einräumen sollen [126].

Unbestimmte Rechtsbegriffe, wie sie in Verträgen oft zu finden sind (z. B. „angemessen“, „zumutbar“, „unverzüglich“), lassen sich nicht problemlos in klare Wenn-Dann-Regeln der Smart Contracts übersetzen. Durch die automatisierte Ausführung bietet ein Smart Contract auch keinerlei Spielraum in der Interpretation des Vertragsinhaltes, z. B., dass der eigentliche Wille der Vertragspartner ermittelt werden könnte (§ 133 BGB). Dies ist insofern von Bedeutung, als Smart Contracts nicht in natürlicher Sprache verfasst sind und ein normaler Nutzer ohne fundierte Kenntnis der verwendeten Programmiersprache große Schwierigkeiten haben dürfte, im Vorfeld genau zu prüfen, ob die im Smart Contract vereinbarten Klauseln tatsächlich seinem Willen entsprechen.

Zusammenfassung.

- Zahlreiche rechtliche Fragen im Zusammenhang mit Blockchain sind noch ungeklärt. Eine gefestigte Rechtsprechung existiert noch nicht.
- Zivil- und strafrechtliche Fragestellungen ergeben sich insbesondere aus der Unveränderlichkeit der Blockchain.
- Smart Contracts sind nicht mit Verträgen im rechtlichen Sinne gleichzusetzen.
- Um die Validität von über die Virtuell-Real-Schnittstelle eingebrachten Daten nachzuweisen, sind organisatorische Maßnahmen erforderlich.

9 Datenschutz und Datensouveränität

Die datenschutzkonforme Gestaltung der Blockchain-Technologie wird kontrovers diskutiert. Insbesondere die zentrale Eigenschaft der Transparenz und der Einsatz von Kryptografie in Blockchains bieten scheinbar gute Voraussetzungen für eine datenschutzfreundliche Technologie. Die systematisch angelegte dauerhafte – und grundsätzlich nicht veränderbare – Rückverfolgbarkeit von Transaktionen stellt gleichzeitig aber eine wesentliche Herausforderung im Datenschutz dar. Im Folgenden werden die wichtigsten Aspekte kurz dargestellt.

Dieses Kapitel ist in Zusammenarbeit mit dem Bundesbeauftragten für den Datenschutz und die Informationsfreiheit (BfDI) entstanden.

9.1 Ziele von Datenschutz und IT-Sicherheit

Im Gegensatz zur IT-Sicherheit, die den Schutz von IT-Systemen und der darin gespeicherten oder verarbeiteten Daten umfasst, geht es beim Datenschutz um den Schutz personenbezogener Daten vor Missbrauch. Die IT-Sicherheit ist dabei ein essentieller Baustein für die Gewährleistung des Datenschutzes. Beide Gebiete haben folglich einen großen Überschneidungsbereich, wenn personenbezogene Daten in IT-Systemen verwaltet werden. Das zeigt sich besonders bei den gemeinsamen Schutzzielen

- Integrität,
- Vertraulichkeit,
- Verfügbarkeit.

Herausforderungen bei der Erreichung dieser Ziele betreffen gleichermaßen die IT-Sicherheit wie den Datenschutz und haben oft die gleichen technischen bzw. kryptografischen Lösungen.

In Ergänzung zu diesen Sicherheitszielen wurden im Datenschutz in Deutschland bereits vor Inkrafttreten der europäischen Datenschutz-Grundverordnung (DSGVO, [127]) die zusätzlichen Ziele

- Transparenz,
- Intervenierbarkeit,
- Nichtverkettbarkeit

sowie die übergeordnete Anforderung der Datenminimierung definiert. Lösungen zur Erreichung dieser Ziele sind ebenfalls oft technischer bzw. kryptografischer Natur und damit eng verwandt mit den Ansätzen der IT-Sicherheit.

Die sechs genannten Datenschutzziele werden oft auch als Gewährleistungsziele bezeichnet (siehe auch das Standard-Datenschutzmodell [128]).

9.2 DSGVO und Blockchain

Außerhalb Deutschlands hat sich das Konzept der Gewährleistungsziele im Datenschutz nicht durchgesetzt. Diese sind nun jedoch im Grundsatzartikel 5 der DSGVO verankert oder folgen aus den in den Artikeln 15–18 formulierten Betroffenenrechten sowie unter anderem aus den Artikeln 13 und 14 (Informationspflichten) sowie Artikel 25 (Datenschutz durch Technikgestaltung und durch datenschutzfreundliche Voreinstellungen) und Artikel 32 (Sicherheit der Verarbeitung). Daher werden diese Datenschutz-Anforderungen im internationalen Kontext spätestens ab Inkrafttreten der DSGVO in der Regel direkt aus den dort definierten Grundsätzen und Betroffenenrechten abgeleitet.

Bei Blockchain-Anwendungen, die von verschiedenen Organisationen oder Personen betrieben werden, kommen außerdem gegebenenfalls die Artikel 26 (Gemeinsam für die Verarbeitung Verantwortliche) und 28 (Auftragsverarbeiter) ins

Spiel, sowie in dem Fall, dass Teile der Blockchain auch in Staaten außerhalb der Europäischen Union verarbeitet werden, die Anforderungen aus dem Kapitel V (Übermittlungen personenbezogener Daten an Drittländer oder an internationale Organisationen) der DSGVO. Generell kann es bei Blockchain-Anwendungen schwierig sein, überhaupt einen Verantwortlichen zu bestimmen. Es kommen verschiedene Akteure wie beispielsweise Entwickler, Betreiber oder Miner in Frage, die konkrete Feststellung der Verantwortlichkeit hängt aber stark vom Einsatzzweck und der Art der Blockchain ab.

Wenn bei der Verarbeitung personenbezogener Daten Blockchains eingesetzt werden sollen, so muss die Blockchain-Anwendung es insbesondere ermöglichen, die in den Artikeln 15–18 und 20 der DSGVO formulierten Betroffenenrechte umzusetzen. Dabei geht es um

- das Auskunftsrecht der betroffenen Person (Artikel 15),
- das Recht auf Berichtigung (Artikel 16),
- das Recht auf Löschung („Recht auf Vergessenwerden", Artikel 17),
- das Recht auf Einschränkung der Verarbeitung (Artikel 18) sowie um
- das Recht auf Datenübertragbarkeit (Artikel 20).

Wichtig im Kontext der Blockchain-Technologie ist die Feststellung aus Erwägungsgrund 26 der DSGVO, dass auch pseudonymisierte personenbezogene Daten noch als personenbezogene Daten anzusehen sind und damit weiterhin den Grundsätzen des Datenschutzes unterliegen. Zuverlässig anonymisierte Daten dagegen sind nicht von der DSGVO betroffen.

Je nach technischer Realisierung einer Blockchain-Anwendung kann das Erreichen der Gewährleistungsziele bzw. die Umsetzung der Betroffenenrechte eine mehr oder weniger große Herausforderung darstellen.

9.3 Erfüllbarkeit der Datenschutzanforderungen in Blockchains

Eine allgemeine Untersuchung der Sicherheitsziele Integrität, Vertraulichkeit und Verfügbarkeit bei Blockchains findet sich bereits in den Kapiteln 2 und 5. Die dort beschriebenen Ergebnisse gelten natürlich auch für den Umgang mit personenbezogenen Daten. Im Weiteren sollen deshalb nur noch die datenschutzspezifischen Anforderungen beim Blockchain-Einsatz betrachtet und technisch-kryptografische Möglichkeiten zur Realisierung der in Abschnitt 9.2 beschriebenen Betroffenenrechte untersucht werden.

Die damit verbundenen rechtlichen Aspekte (siehe auch Kapitel 8) sowie die Frage nach der Verantwortlichkeit für die Erfüllung der Betroffenenrechte (siehe Abschnitt 9.2) sind dabei nicht Gegenstand der Untersuchung, da sie nur geringe direkte Auswirkungen auf die sicherheitstechnische Umsetzung haben.

9.3.1 Auskunftsrecht

Das Auskunftsrecht erweitert im Wesentlichen die Forderung nach Transparenz, also der Nachvollziehbarkeit der Datenerhebung und Datenverarbeitung. Transparenz gilt allgemein als eine der Haupteigenschaften von Blockchains. Wenn es sich um eine öffentliche Blockchain handelt oder der Nutzer in einer privaten Blockchain entsprechende Leserechte hat, kann er die eigenen Transaktionsdaten einsehen und ihre Historie und Verknüpfungen in der Blockchain nachvollziehen. Dabei ist allerdings zu beachten, dass das nur für Daten gilt, die tatsächlich in den Transaktionen selbst gespeichert und nicht nur referenziert sind.

Schwieriger gestaltet sich die Erfüllung des Auskunftsrechts, wenn es um Informationen über verarbeitende Stellen, Empfänger, Verarbeitungszwecke oder Ähnliches geht. Hier können je nach Blockchain-Modell, besonders aber bei öffentlichen und genehmigungsfreien Blockchains, aufgrund fehlender Regulierung oft keine validen Auskünfte gegeben werden.

9.3.2 Recht auf Berichtigung und Recht auf Löschung

Sowohl das Recht auf Berichtigung als auch das Recht auf Löschung („Vergessenwerden“) (bzw. allgemein die Forderung nach Intervenierbarkeit) widersprechen dem grundlegenden Blockchain-Ziel der Manipulationssicherheit und sind in Blockchains generell schwer zu erreichen. Eine Berichtigung von Daten kann allgemein nur durch die Einstellung einer neuen Transaktion realisiert werden, die eine neue Version der Daten bereitstellt. Damit können Daten aber nur aktualisiert und nicht überschrieben werden. Die alten Versionen bleiben in der Blockchain erhalten. Für eine Löschung von Transaktionsdaten oder eine Berichtigung im Sinne von Ersetzung wäre es notwendig, bereits in der Blockchain verankerte Transaktionen im Nachhinein zu verändern, was durch die Verkettung der Transaktionsblöcke eigentlich unmöglich sein sollte.

Dennoch gibt es einige Ansätze, um eine gewollte Revision von Daten in der Blockchain zu ermöglichen:

- Rollback: In genehmigungsbasierten Blockchains kann eine privilegierte Gruppe von Knoten bestimmt werden, die das Recht erhält, unter bestimmten Bedingungen die Historie der Blockchain umzuschreiben und Transaktionen nachträglich zu ändern.

- Forks: Bei dringendem Revisionsbedarf, zum Beispiel bei gravierenden Sicherheitsvorfällen, besteht die Möglichkeit, einen Fork der Blockchain zu erzwingen, d. h., eine Abspaltung eines Blockchain-Zweigs herbeizuführen, in dem dann eine alternative Version der fraglichen Transaktionen eingebracht und bestätigt werden kann. Dazu muss sich entweder die Gesamtheit aller Nutzer oder zumindest eine gewisse kritische Masse darauf verständigen, den Fork zu unterstützen und an der Konsensbildung im neuen Zweig mitzuwirken, damit das notwendige Vertrauen in die neue Blockchain hergestellt werden kann. Gerade bei genehmigungsfreien Blockchains kann das eine Herausforderung darstellen und hat in der Vergangenheit beispielsweise dazu geführt, dass die Ethereum-Blockchain nach dem DAO-Hack (siehe Abschnitt 7.2) in verschiedene Zweige (Ethereum und Ethereum Classic) zerfallen ist, die heute unabhängig voneinander weitergeführt werden.

- Off-Chain-Datenspeicherung: Damit Daten nicht unter den Manipulationsschutz der Blockchain fallen, können sie extern gespeichert und in der Blockchain nur referenziert werden (siehe Abschnitt 5.2). Falls die Referenzierung über einen Hashwert realisiert wird, ist dieser nach einer Änderung nicht mehr gültig, der entsprechende Transaktionsblock kann aber weiterhin verifiziert werden und die Integrität der übrigen Transaktionen und der Blockchain als Ganzes bleibt erhalten. Ein solcher Ansatz wird z. B. für die Speicherung von Zertifikaten auf Blockchains verfolgt [129].

- Chameleon-Hashes: Eine mögliche kryptografische Lösung besteht im Konzept der Chameleon-Hashes [130]. Das sind Hashfunktionen, bei denen über eine „Hintertür“ *(trapdoor)* die Konstruktion von Kollisionen, also von unterschiedlichen Daten mit demselben Hashwert, möglich ist. Durch solche Kollisionen können Daten auf der Blockchain ausgetauscht werden, ohne dass der Integritätsschutz der Blockchain aufgehoben wird [131]. Damit die Manipulationssicherheit der Blockchain grundsätzlich gewährleistet bleibt, muss die Hintertür (bzw. die entsprechenden kryptografischen Schlüssel) von einer vertrauenswürdigen Instanz sicher verwaltet werden oder die Schlüssel müssen mit einem Secret-Sharing-Verfahren auf verschiedene Parteien aufgeteilt werden, die die Hintertür nur in Zusammenarbeit einsetzen können.

- Mutable Blockchains: Eine weitere Alternative, um gewünschte Änderungen in die Blockchain einzubringen, ist die Steuerung der Sichtbarkeit von Transaktionsdaten über Smart Contracts [132]. Dabei kann es zu jeder Transaktion mehrere Versionen geben, von denen jeweils nur die aktuell gültige im Netzwerk sichtbar ist, während die anderen verschlüsselt und damit nicht zugänglich sind. Auch hier muss geregelt werden, wer die Schlüssel zu den gesperrten Versionen verwal-

tet. Außerdem ergeben sich wieder die bereits im Abschnitt 5.1 diskutierten Einschränkungen von Verschlüsselung bezüglich Langzeitsicherheit und Kryptoagilität.

Allen Ansätzen zur Änderung oder Löschung von Daten in Blockchains ist gemein, dass sie in gewisser Weise die Grundideen der Blockchain-Technologie aushebeln. Das kann insofern problematisch sein, als damit neue Angriffsmöglichkeiten entstehen, z. B. wenn die Mechanismen zur Änderung von Daten missbräuchlich genutzt werden. Insbesondere bei den beiden zuletzt genannten kryptografischen Konzepten ist zudem nicht klar, wie die Konsistenz der Blockchain trotz Datenrevision erhalten werden kann. Zwar können Daten ausgetauscht oder unsichtbar gemacht werden, über die Verknüpfung der Transaktionen untereinander kann das aber zu Problemen bei der Validierung oder Verarbeitung anderer Daten führen. Außerdem muss je nach Anwendung geklärt werden, wie mit entstehenden Datenreliken umzugehen ist.

Wenn die Datenrevision für bestimmte Akteure in der Blockchain zusätzliche Rechte oder eine herausgehobene Stellung verlangt, so ändert sich damit auch das zugrunde liegende Vertrauensmodell. Durch die verteilte Datenhaltung im Blockchain-Netzwerk können zudem viele lokale Kopien unterschiedlichen Datums der Blockchain auf den einzelnen Knoten existieren, die von der Revision nicht erfasst werden und in denen die alten Daten weiterhin verfügbar sind.

9.3.3 Recht auf Einschränkung der Verarbeitung

Das Recht auf Einschränkung der Verarbeitung kann zusammen mit der Forderung nach Nichtverkettbarkeit betrachtet werden und steht ebenfalls in einem grundlegenden Widerspruch zur Konstruktion einer Blockchain. Transaktionsdaten sind grundsätzlich für jeden Blockchain-Teilnehmer einsehbar und nutzbar und die Verkettung der Blöcke macht über eine zeitliche Ordnung die logische Verknüpfung von Transaktionen möglich. Eine Einschränkung der Verarbeitung oder eine Trennung nach Zweck sind dabei nicht vorgesehen.

Ein möglicher Ansatz zur Lösung dieses Widerspruchs wären anonyme Transaktionen oder Smart Contracts (siehe Abschnitt 5.1 bzw. 5.4), mit deren Hilfe die Nutzer selbst den Zugriff auf ihre Daten steuern und Verknüpfungen für Außenstehende erschweren können. Dadurch verschiedene Verarbeitungszwecke zu kontrollieren, ist allerdings sehr aufwändig.

Wenn Daten auf der Blockchain nur referenziert, aber extern abgelegt werden, so kann über die Blockchain ein Rechtemanagement realisiert werden, das den Zugriff auf die Daten an bestimmte Voraussetzungen knüpft und zeitlich beschränkt.

In privaten Blockchains sind Möglichkeiten zur Beschränkung der Verfügbarkeit der Daten bereits per Konstruktion vorgesehen. Eine feingranulare und variable Zugriffssteuerung mit entsprechendem Rollenkonzept ist dennoch nur mit größerem Aufwand umzusetzen.

9.3.4 Recht auf Datenübertragbarkeit

Das Recht auf Datenübertragbarkeit kann je nach technischer Realisierung der Blockchain relativ einfach umgesetzt werden, zumindest dann, wenn sämtliche Daten direkt in der Blockchain gespeichert werden. Allerdings könnte die Aufgabe, die zu einer Person gehörenden Daten zusammenzustellen und sie dieser Person „in einem strukturierten, gängigen (...) Format“ (Artikel 20 DSGVO) zur Verfügung zu stellen, durchaus eine Herausforderung darstellen. Ob das Argument, bei einer öffentlichen Blockchain lägen diese Informationen ja bereits in solcher Form vor, in jedem Fall Bestand hätte, erscheint beim aktuellen Stand der Technik eher zweifelhaft. Es kann nicht unbedingt von allen Personen, von denen eventuell Daten in der Blockchain gespeichert sind, erwartet werden, dass sie die sie betreffenden Daten selbst aus der Blockchain extrahieren können. Bei einer Off-Chain-Speicherung würde sich auf jeden Fall die zusätzliche Verpflichtung ergeben, auch außerhalb der eigentlichen Blockchain gespeicherte Daten entsprechend aufzubereiten und zur Verfügung zu stellen.

Zusammenfassend kann festgestellt werden, dass der Einsatz der Blockchain-Technologie für Datenschutzanwendungen vor Herausforderungen steht. Teilweise können diese organisatorisch gelöst werden, z. B. durch die allgemeinen Maßnahmen der Off-Chain-Speicherung oder Forks, teilweise aber auch durch die Wahl eines restriktiven Blockchain-Modells. Private oder genehmigungsbasierte Blockchains bieten grundsätzlich deutlich mehr Spielraum für die Durchsetzung datenschutzrechtlicher Anforderungen. Ausgereifte technologische Lösungen liegen für die meisten Problemstellungen noch nicht vor. Nach jetzigem Stand ist Blockchain also im Allgemeinen nicht als datenschutzfördende Technologie geeignet.

9.4 Datensouveränität

Als ein Argument für die Blockchain wird häufig genannt, dass sie die *Datensouveränität* – als Teil der Digitalen Souveränität – der Nutzer ermöglicht. Da die Digitale Souveränität entscheidend durch die äußeren Gegebenheiten – insbesondere Datenschutzrichtlinien – beeinflusst wird, soll dieses Thema hier als Datenschutzaspekt im weiteren Sinne behandelt werden.

Digitale Souveränität wird hier als die Fähigkeit zu „selbstbestimmtem Handeln und Entscheiden im digitalen Raum“ verstanden [133], [134], [135]. Datensouveränität bedeutet unter anderem, dass der Nutzer jederzeit die Kontrolle über seine Daten behält und feststellen kann, wer sie bearbeitet oder auf sie zugegriffen hat. Es ist jedoch fraglich, inwiefern eine Blockchain-Anwendung diesen Ansprüchen gerecht werden kann.

Zunächst gibt es auch auf Blockchains immer Instanzen mit besonderen Rollen oder besonderem Einfluss, wie bereits in Abschnitt 3.1 angesprochen, denen zur Wahrung der Digitalen Souveränität besonderes Vertrauen entgegengebracht werden muss. Das können beispielsweise Regulierungsstellen, Miner oder die Programmierer der Blockchain-Software sein.

Miner können durch ihren Einfluss bei der Konsensbildung bestimmte Transaktionen gezielt blockieren, was die Verfügungsgewalt der betroffenen Nutzer einschränken kann. Diese Fragestellung kam beispielsweise im Zusammenhang mit dem Verkauf beschlagnahmter Bitcoins durch Strafverfolgungsbehörden auf [136]. Die Verwendung verschiedener Pseudonyme, wie sie auf vielen Blockchains empfohlen wird, schafft hier keine Abhilfe. Wie in Abschnitt 5.4 bemerkt, können nämlich solchen Pseudonymen durch Transaktionsdatenanalyse in vielen Fällen reale Identitäten zugeordnet werden. Dies kann auch zur Folge haben, dass Transaktionsinformationen eines Nutzers ungewollt für Dritte nachvollziehbar werden.

Die Programmierer der Blockchain-Software können absichtlich oder versehentlich Fehler in den Quellcode einbringen. Die adäquate Implementierung verschiedener Sicherheitskomponenten, z. B. der Erzeugung sicherer Signaturschlüsselpaare, ist sehr anspruchsvoll. Ohne eine unabhängige Evaluierung der Produkte fällt es Nutzern ohne mathematische Fachkenntnisse daher schwer, die Sicherheit von Blockchain-Lösungen einzuschätzen. Bei einer solchen Evaluierung durch eine zentrale Stelle ergibt sich aber wieder ein Spannungsverhältnis mit dem Blockchain-Ziel der Dezentralität.

Weiterhin ist die Frage ungeklärt, wie Nutzer rechtliche Ansprüche durchsetzen können, falls in Blockchains keine verantwortliche Stelle vorhanden ist oder man nicht auf diese zugreifen kann.

Schließlich sind die im Bereich der Identitätsverwaltung verbreiteten Zentralisierungstendenzen zu berücksichtigen. Dabei werden die Identitätsdaten von Nutzern durch zentrale Dienstleister verwaltet und in verschiedenen Kontexten verwendet, obwohl eine Trennung sinnvoll wäre. Als Motivation für die Nutzung von Blockchains wird gerade genannt, dass damit solche Strukturen aufgebrochen werden sollen. Eine zentralisierte Identitätsverwaltung birgt einerseits erhöhtes Missbrauchspotenzial, andererseits erhöht sie jedoch die Effizienz und wird in vielen Fällen als benutzerfreundlicher empfunden.

Zusammenfassung.

- Datenschutz und IT-Sicherheit haben große Überschneidungen bezüglich ihrer Anforderungen an die Blockchain-Technologie.
- Die Vorgaben der DSGVO sind in ihrem Anwendungsbereich auch für den Einsatz von Blockchains zu beachten.
- Die Erfüllung wichtiger Datenschutzziele ist in Blockchain-Anwendungen schwierig.
- Blockchains sind nicht ohne Weiteres dazu geeignet, die Datensouveränität der Nutzer zu befördern.

Teil IV Praxis

10 Anwendungsgebiete

Der Einsatz von Blockchains wird in den verschiedensten Bereichen untersucht. Im Sinne einer allgemeinen Hilfestellung sollen in diesem Kapitel einige abstrakte Anwendungsszenarien und ihre zugrunde liegenden Ideen vorgestellt und diskutiert werden, anstatt konkrete Beispiele besonders hervorzuheben. Dabei wird auch auf diejenigen Fragen verwiesen, deren besondere Beachtung und sorgfältige Klärung für eine sichere und sinnvolle Umsetzung nötig ist.

Über diese Fragen hinaus gilt es in jedem Fall zu bedenken, ob die von der Einführung einer Blockchain-Lösung erwarteten Ziele tatsächlich erreicht werden. Konkret wird das Fehlen einer zentralen Instanz oft als Hauptmotivation für die Umstellung auf Blockchain genannt, aber die folgenden Beispielszenarien zeigen, dass in vielen Fällen trotzdem nicht auf zentrale externe Dienste – etwa Vertrauensdienste zur Erstellung und Prüfung von Zertifikaten, Signaturen, Siegeln und Zeitstempeln – verzichtet werden kann. Grundsätzlich muss bei der Umsetzung von Lösungen vorab geklärt und Einigkeit darüber erzielt werden, welche Knoten auf der Blockchain welche Rollen und Rechte bekommen sollen. Bei dieser Entscheidung können auch gesetzliche Vorgaben relevant sein.

Auch die Frage nach den Kosten wird in der Praxis bei der Entscheidung für oder gegen den Einsatz der Blockchain-Technologie und der konkreten Ausgestaltung eine große Rolle spielen. Da sie aber stark vom Einzelfall und der gegebenen Situation abhängt, kann es hier keine pauschalen Antworten geben.

10.1 Eigentumsnachweise

Das zeitlich erste Anwendungsgebiet ist die Eigentumsübertragung von Werten. Die Blockchain dient dabei zur Dokumentation der entsprechenden Transaktionen. Neben der Anwendung in Bitcoin und weiteren Kryptowährungen ist der Einsatz auch im Energiebereich und für verschiedene öffentliche Register vorgeschlagen worden. Durch den Einsatz von Blockchains und den Verzicht auf eine zentrale Instanz für deren Betrieb erhofft man sich eine schnellere und günstigere Abwicklung von Transaktionen. Zudem wird im Rahmen der Energiewende eine dezentrale Lösung für den Handel der dezentral erzeugten Energie als vorteilhaft angesehen. Entsprechende Ideen existieren seit einigen Jahren. Die konkrete Umsetzung der Projekte im Energiebereich und für öffentliche Register umfasst bisher im Wesentlichen einige Pilotprojekte von beschränktem Umfang.

Grundprobleme sind die in den Abschnitten 2.1.4 und 2.2.2 genannten Nachteile der Blockchain-Technologie bezüglich Datendurchsatz und Vertraulichkeit. Öffentliche genehmigungsfreie Blockchains erlauben nur einen sehr geringen Durchsatz, der einen großflächigen Einsatz beispielsweise im Finanzbereich momentan ausschließt. Private bzw. genehmigungsbasierte Blockchains erlauben einen deutlich höheren Durchsatz, der aber weiterhin unter demjenigen zentralisierter Lösungen liegt. Die nicht oder nur mit weiteren hohen Einbußen bei der Effizienz herstellbare Vertraulichkeit (siehe Abschnitt 5.1) ist insbesondere bei Transaktionen, die sensible Daten umfassen, inakzeptabel. Für bestimmte Anwendungsfälle mit wenigen Knoten könnten die auf privaten Blockchains vorliegenden Möglichkeiten (siehe Abschnitt 5.1) genutzt werden, um die genannten Probleme abzumildern.

10.2 Prozesskontrolle

Auch bei der Kontrolle von Geschäftsprozessen gibt es Ansätze für die Verwendung von Blockchains. Dabei sollen Informationen über Ablauf und Zwischenschritte in den Blöcken gespeichert werden. Ein häufig genanntes Beispiel ist die Lieferkettenverfolgung im globalen Handel.

In diesem Einsatzszenario relevant ist die Virtuell-Real-Schnittstelle (siehe Abschnitt 7.3.1).

Die Authentizität, Integrität und Vollständigkeit der in der Blockchain gespeicherten Informationen, die beispielsweise als Sensordaten anfallen können, müssen sichergestellt sein. Zudem sind gegebenenfalls die Daten einzelner Parteien vertraulich oder sollen zumindest nicht für alle Knoten im Netzwerk sichtbar sein. In diesem Fall ist eine direkte Speicherung dieser Daten in der Blockchain nicht sinnvoll (siehe Abschnitt 5.1).

Die beteiligten Parteien sind in der Regel keine Privatpersonen, sondern Wirtschaftsunternehmen oder staatliche Stellen. Ihre Anzahl ist zudem begrenzt, was vermutlich auch auf die zu verarbeitende Informationsmenge zutrifft. Somit könnten durch den Einsatz privater genehmigungsbasierter Blockchains Vorteile in der Transparenz erzielt und eine nachvollziehbare, manipulationssichere Dokumentation der Prozessschritte erzielt werden.

10.3 Integritätssicherung

Ein weiterer Anwendungsfall sieht vor, die Hashwerte von Dokumenten, deren Integrität gesichert werden soll, in einer Blockchain zu speichern. Bei Bedarf kann die Integrität eines Dokuments zu einem späteren Zeitpunkt durch erneute Berechnung des Hashwerts und Vergleich mit dem hinterlegten Wert überprüft werden. Diese Anwendung wird beispielsweise für Zeugnisse (siehe Beispiel unten) und Zertifikate oder Dokumente, die urheberrechtlich zu schützendes Material enthalten, untersucht. Man beachte, dass eine erfolgreiche Verifikation lediglich die Übereinstimmung mit dem Original bestätigt, über die Authentizität desselben jedoch nichts auszusagen vermag.

Insbesondere wenn die Verifikation des Hashwertes nicht durch den Ersteller erfolgt, ist ein Nachweis der Authentizität der Dokumente entscheidend. Hier müssen also zusätzliche Maßnahmen ergriffen werden. Zum Beispiel kann eine vertrauenswürdige Instanz die Daten in die Blockchain einbringen und dabei die Echtheit der Dokumente mit einer Signatur bestätigen. Die Art der Signatur und die Anforderungen an den Signaturersteller müssen auf das geforderte Schutzniveau abgestimmt sein. Sollen etwa die hinterlegten Daten einen rechtlichen Beweiswert haben, sind qualifizierte elektronische Signaturen bzw. Siegel bzw. Zeitstempel gemäß der eIDAS-Verordnung [52] erforderlich. Ebenso muss für eine langfristige rechtssichere Aufbewahrung auch die Verkehrsfähigkeit der Daten sichergestellt sein, d. h. die Möglichkeit, die Daten in ein anderes System zu übertragen, wobei ihre Integrität und Authentizität nachweisbar bleiben.

Es sei darauf hingewiesen, dass sich die im Blockchain-Kontext gelegentlich genannten Ideen für Zeitstempel hier in der Regel nicht eignen. Gemäß dieser Vorschläge sollen beliebige Daten durch Einbettung in Blöcke mit einer Zeitangabe – wie z. B. dem Erstellungszeitpunkt des Blocks – versehen und somit ihre Existenz zu einem bestimmten Zeitpunkt nachgewiesen werden können. Als Nebeneffekt der Hashverkettung der Blöcke ergibt sich so eine total geordnete Folge solcher Blockchain-Zeitstempel. Diese muss jedoch nicht der tatsächlichen Reihenfolge der Entstehung der zugehörigen Daten entsprechen, sondern spiegelt nur wider, in welcher Reihenfolge sie in der Blockchain aufgenommen wurden. Eine Verknüpfung mit Zeitstempeln aus anderen Quellen ist nicht möglich, da die Blockchain-Zeitstempel ohne zusätzliche Maßnahmen keine Gültigkeit als absolute Werte besitzen und auch nicht notwendigerweise der wahren Zeit entsprechen müssen. Je nach Konsensmechanismus und Latenzzeit des Netzwerks ergeben sich gewisse Spielräume und Manipulationsmöglichkeiten. Größere Abweichungen von der wahren Zeit könnten vermutlich von allen Knoten detektiert, aber nicht direkt verhindert werden.

Eine Erweiterung des Vorschlags, in der durch Einbinden im Voraus unbekannter und kaum zu beeinflussender Werte (z. B. Lottozahlen, genaue Wetterkarten) die tatsächliche Erstellung eines Blocks zu einem bestimmten Zeitpunkt nachgewiesen werden soll, löst dieses Problem nicht. Sie beweist lediglich, dass der Block nach dem Zeitpunkt der Entstehung dieser Informationen erzeugt wurde. Die für praktische Anwendungen relevantere Information, dass die Daten mit dem Zeitstempel vor einem bestimmten Zeitpunkt vorlagen *(Proof-of-Existence)*, kann nicht erwiesen werden.

Für eine rechtskonforme und tragfähige Beweiswerterhaltung [137] braucht es ferner Möglichkeiten, die Integrität von Dokumenten über einen längeren Zeitraum zu schützen. Dazu müssen adäquate Mechanismen zur Erneuerung des Schutzes vorliegen, wenn die Eignung der verwendeten Hashfunktionen, Signaturen oder anderer kryptografischer Routinen oder Parameter ausläuft. Auf Blockchain-Systemen sind hierzu bisher keine geeigneten Konzepte etabliert; Details finden sich in Abschnitt 6.1.

Beispiel: Zeugnis-Blockchain

Um Fälschungen von Zeugnissen zu verhindern, wird der folgende Ansatz diskutiert und in Tests erprobt [138], [139]: Zugrunde liegt eine öffentliche genehmigungsbasierte Blockchain, bei der nur Lehranstalten über Schreibrechte verfügen. Diese signieren die Hashwerte von ihnen ausgestellter Zeugnisse und stellen sie signiert in die Blockchain. Gibt es frühere Versionen des Zeugnisses (etwa solche mit fehlerhaften Einträgen), so wird ein Verweis auf die frühere Version eingefügt. Ebenso geht man vor, wenn ein Zeugnis zurückgezogen werden muss.

Bei diesem Vorgang muss das Zeugnis um einen zufälligen Eintrag *(Salt)* fixierter Länge ergänzt werden, um das bei stereotypen Daten mögliche Rückrechnen von Zeugnisinhalten zu vermeiden. Der Salt kann beispielsweise in Form eines QR-Codes auf dem Zeugnis angebracht werden. Dem Zeugnisinhaber wird das Zeugnis nicht nur in Papierform, sondern auch elektronisch als Datei ausgehändigt. Muss er nun das Zeugnis vorweisen, hat die prüfende Instanz die Möglichkeit, den Hashwert neu zu berechnen und in der Blockchain zu prüfen. Dies ist möglich, ohne dass der Prüfer auf externe Daten – außer denen zur Validierung des öffentlichen Signaturschlüssels – zurückgreifen oder bei der zeugnisherausgebenden Instanz nachfragen muss. Diese muss also aus rein technischen Gründen keine Zeugniskopie aufbewahren, könnte allerdings aus rechtlichen Gründen weiterhin dazu verpflichtet sein.

Es kann nach heutiger Kenntnis, auch wenn die Hashfunktion gebrochen wird, dennoch schwierig sein, ein weiteres Urbild im exakt gleichen Format zu erzeugen. Dies gilt insbesondere, wenn die Zeugnisvorlage dem Prüfer authentisch elektronisch vorliegt und Daten in einem definierten Format eingetragen werden müssen. Die Sicherheit kann man noch erhöhen, indem in der Blockchain ein zusätzlicher Prüfwert wie einige Bits des Hashwertes des Salts abgelegt wird. Wahrscheinlich können Zeugnis-Blockchains zum Schutz vor Fälschungen beitragen, sie bieten aber keine absolute Sicherheit. Im Zweifelsfall ist auch eine Prüfung des Originaldokumentes in Papierform anzuraten.

10.4 Identitätsmanagement

Eine allgemeine Idee besteht darin, Blockchains zur Identifizierung im Internet und zum Zugriffsschutz (Authentisierung und Autorisierung) auf Kundenkonten zu verwenden. Ein häufig vorgestellter Ansatz liegt darin, außerhalb der Blockchain einen Hashwert über Identitätsdaten einer Person zu berechnen, der anschließend in der Blockchain gespeichert wird. Zu einem späteren Zeitpunkt kann zum Nachweis der Identität einer Person aus den entsprechenden Identitätsdaten dieser Hashwert erneut berechnet und mit den auf der Blockchain hinterlegten und vor Manipulationen geschützten Daten abgeglichen werden. Darüber wird gezeigt, dass die Person tatsächlich die vorher in der Blockchain gesicherten Identitätsdaten kennt.

Grundsätzlich ist darauf hinzuweisen, dass die Blockchain selbst nicht die Authentisierung durchführt, sondern lediglich als Datenspeicher und Hilfsmittel dient. Die Authentisierung erfolgt direkt gegenüber dem entsprechenden Kommunikationspartner.

Eine Erweiterung der Grundidee besteht darin, mittels der Blockchain die Zugriffsberechtigungen auf die entsprechenden Identitätsdaten feingranular zu verwalten. So sollen nur die jeweils

nötigen Attribute (z. B. ob eine Person volljährig ist) zur Verfügung gestellt und damit Identitätsdiebstahl erschwert werden.

Bei der Umsetzung dieses Ansatzes stellen sich eine Reihe von Fragen. Im Kontext der Online-Authentisierung liegt ein wesentliches Problem in der initialen Identitätsprüfung bzw. initialen Verifikation von Identitätsdaten und Erzeugung der zugehörigen elektronischen Identitäten, d. h. der Verknüpfung von Identitätsdaten mit einer Person und einem Authentisierungsmittel. Die Identitätsprüfung muss dabei manipulationssicher und auf einem ausreichenden Vertrauensniveau durchgeführt werden. Technische Vorgaben finden sich in [140]. Bei der Erzeugung der elektronischen Identitäten muss die Korrektheit und Authentizität der gespeicherten Identitätsdaten auf einem ausreichenden Schutzniveau sichergestellt werden. Zur Lösung dieser beiden zentralen Probleme sind zusätzliche Mechanismen außerhalb der Blockchain erforderlich. Grundsätzlich liegt dies daran, dass die Sicherheitsgarantien für Daten, insbesondere für die Authentizität (siehe Abschnitt 2.1.5), erst nach ihrer Aufnahme in die Blockchain gelten.

Weiterhin sind verschiedene Angriffe auf den vorgestellten Prozess der Authentisierung denkbar, mit denen sich ein Angreifer mit einer falschen Identität authentisieren kann und die durch passende Schutzmaßnahmen verhindert werden müssten. Hier sind unter anderem Brute-Force-Angriffe auf die Hashwerte, wenn sie direkt auf Basis strukturierter Daten berechnet werden (siehe Abschnitt 5.2), oder das Abfangen von Nachrichten mit anschließendem Missbrauch der Daten (Replay-Angriffe) zu nennen.

Einige bestehende Projekte setzen auf öffentlichen genehmigungsfreien Blockchains auf. In diesem Fall kann die zeitnahe Aufnahme der Identitätsdaten in einen Block nicht garantiert werden, zudem kann auf den gegenwärtig nutzbaren Blockchains die Höhe der anfallenden Gebühren im Zeitverlauf stark schwanken und schwer im Voraus abgeschätzt werden [141].

Grundsätzlich muss geprüft werden, inwiefern es mit der Blockchain-Technologie vereinbar ist, im Bedarfsfall Identitäten zu sperren oder zu widerrufen. Da die Identitätsdaten über einen Zeitraum von Jahrzehnten geschützt sein müssten, sind des Weiteren praktisch umsetzbare Konzepte zur Langzeitsicherheit absolut erforderlich (siehe Abschnitt 6.2.3).

Weiterhin stellen sich juristische Fragen (siehe Kapitel 8). Diese betreffen einerseits die rechtliche Verantwortlichkeit für die Inhalte und den Betrieb der Blockchain und andererseits den Datenschutz, z. B. die zur Verarbeitung berechtigten Parteien oder die Personenbeziehbarkeit von Pseudonymen (siehe Abschnitt 9.2). Eine Anonymisierung ist im vorgestellten Anwendungsfall per se nicht möglich. Außerdem muss sichergestellt werden, dass die Aktivitäten von Nutzern mithilfe der auf der Blockchain zugänglichen Daten nicht nachverfolgbar werden.

Zusammenfassung.

Je nach Anwendungsfall:

- Die Einschränkungen von Blockchains in den Punkten Vertraulichkeit und Effizienz sollten berücksichtigt werden.
- Die Authentizität von eingespeisten Daten sowie der Kommunikationspartner muss durch Maßnahmen außerhalb der Blockchain sichergestellt werden.
- Rechtliche Vorgaben z. B. zu Signaturen und Zeitstempeln müssen beachtet werden.
- Konzepte zur Langzeitsicherheit sind erforderlich.

11 Weiterentwicklungen

Die Blockchain-Technologie ist ein sehr weites Feld und umfasst viele unterschiedliche Ausprägungen. Dennoch ist Blockchain – wie bereits in Abschnitt 1.1 erwähnt – selbst nur ein Spezialfall des deutlich weiter gefassten Begriffs der Distributed-Ledger-Technologien (DLT). Im Wesentlichen wird bei DLT die Datenstruktur allgemeiner definiert als bei Blockchain und es muss weder eine Blockbildung noch eine Kettenstruktur geben.

Die zukünftige Entwicklung im Bereich Blockchain und DLT ist heute noch kaum abzusehen, es gibt aber bereits einige Ansätze, wie DLT mit alternativen Datenstrukturen aufgesetzt werden kann bzw. wie Blockchains durch andere Netzwerkstrukturen ergänzt oder mit anderen aktuellen Technologien (Quantencomputing, Quantenkommunikation, Künstliche Intelligenz (KI)) verknüpft werden können. Die meisten dieser Konzepte sind recht neu und zum Teil noch gar nicht praktisch realisiert. Gerade die Verbindung von Blockchain mit Quantentechnologien oder KI ist vielleicht eher der Tatsache geschuldet, dass man hier verschiedene aktuelle Themen zu vereinen sucht. Zu vielen Sicherheitsaspekten sind deshalb noch keine ausgereiften Konzepte vorhanden und es fehlt insbesondere eine gründliche Prüfung durch die Expertengemeinschaft. Ganz allgemein kann zum jetzigen Zeitpunkt noch nicht beurteilt werden, ob und in welchem Ausmaß die verschiedenen Ansätze tatsächlich ihre technischen Ziele erreichen werden.

Im Folgenden werden einige Weiterentwicklungen vorgestellt.

11.1 Blockgitter und doppelte Verkettung

Eine naheliegende Verallgemeinerung der Datenstruktur Blockchain ist die mehrdimensionale Erweiterung der Kettenstruktur, insbesondere die Verwendung einer Gitterstruktur. Erste Umsetzungen eines solchen Blockgitters finden sich bei der Kryptowährung Nano [142] oder der Distributed-Applications-Plattform Dexon [143].

Bei Nano unterhält jeder Teilnehmer seine eigene asynchrone Blockchain mit der Historie des eigenen Kontostandes, wobei Transaktionen zwischen Teilnehmern als Änderungen deren aktueller Kontostände in einer globalen Kette öffentlich festgehalten werden. Der Konsensmechanismus der globalen Kette ist ein sogenannter delegierter Proof-of-Stake (dPoS), eine spezielle Instanz von Proof-of-Stake (siehe Abschnitt 3.2.3). Dagegen kann bei Dexon jeder Knoten eine eigene Blockchain mit allen im Netzwerk verteilten Transaktionen parallel aufbauen, während ein blockchainübergreifender Konsens über die Gültigkeit und den Zeitpunkt einer Blockerstellung durch ein BFT-Verfahren hergestellt und auf einer globalen Kette festgehalten wird.

Mit Hilfe solcher Konstruktionen soll – insbesondere im Vergleich zu Bitcoin – die Skalierbarkeit des Systems erhöht und die Latenzzeit sowie der Energieverbrauch im Netzwerk reduziert werden, obwohl es sich dabei ebenfalls um öffentliche genehmigungsfreie Distributed Ledgers handelt.

Eine weitere Verallgemeinerung besteht darin, die einfache (gerichtete) Verkettung der Blöcke in einer Blockchain durch eine Verkettung in beide Richtungen zu ersetzen [144]. Dadurch gibt es einerseits im Gegensatz zu beispielsweise Bitcoin kein unsicheres Ende der Kette – auch der letzte Block ist durch die Verankerung im vorletzten Block schon integritätsgeschützt – und andererseits soll es möglich sein, Blöcke mit sensiblen Inhalten vor dem öffentlichen Zugriff zu verbergen bzw. Blöcke aus der Kette zu löschen.

11.2 Gerichtete azyklische Graphen

Noch weiter gefasst ist das Konzept der gerichteten azyklischen Graphen (directed acyclic graphs, DAG), das die Kettenstruktur der Blockchain zugunsten eines Geflechts von Transaktionen auflöst. Der bekannteste Vertreter ist die Kryptowährung IOTA [145], die insbesondere auf Kommunikation und Zahlungsverkehr im

Internet-of-Things (IoT) abzielt. Hier bilden die Transaktionen die Ecken eines Graphen, wobei für jede neu eingestellte Transaktion im Rahmen des Konsensverfahrens zwei andere Transaktionen und deren Konfliktfreiheit bestätigt werden müssen. Von der neuen Transaktion führen dann gerichtete Kanten im Graphen zu den bestätigten Transaktionen (siehe Abbildung 13).

Gleichzeitig muss für die Erstellung einer gültigen Transaktion ein (im Vergleich zu Bitcoin einfacher) Proof-of-Work erbracht werden, der auf der Berechnung spezieller Hashwerte basiert, die auch Informationen über die bestätigten Transaktionen enthalten. Transaktionen, die von ausreichend vielen neuen Transaktionen direkt oder indirekt (abgebildet durch eine Kante oder einen Weg in dem Graphen) bestätigt wurden, gelten als verifiziert und damit als sicher. Mit dieser Struktur soll gegenüber blockchainbasierten Kryptowährungen die Skalierbarkeit und der Durchsatz verbessert sowie die Transaktionskosten verringert werden. Die Stabilität und die Resistenz des DAG hängen allerdings in hohem Maße von der Auswahl der zu verifizierenden Vorgängertransaktionen ab, wofür komplexe wahrscheinlichkeitstheoretische Modelle entwickelt werden müssen. Außerdem gibt es durch die Graphenstruktur eine Reihe von neuen Angriffsszenarien, zum Beispiel für Double-Spending-Angriffe [146].

11.3 Interoperabilität

Im Bereich Blockchain bzw. DLT gibt es schon heute eine Vielzahl an möglichen Modellen und anwendungsspezifischen Ausgestaltungen der Technologie. Aufgrund der relativ spät angelaufenen internationalen Standardisierung haben sich in den letzten Jahren in einzelnen Anwendungsfeldern Quasi-Standards, aber auch eine große Zahl an Einzellösungen herausgebildet, die untereinander meist nicht kompatibel sind (siehe Abschnitt 12.1). Das kann dazu führen, dass dieselben Informationen in verschiedenen Blockchains unsynchronisiert verwendet werden oder Nutzer ihre Anwendungen auf verschiedenen Blockchains parallel führen müssen.

Es gibt inzwischen einige technische Ansätze, um den Wildwuchs an Blockchains wieder zu konsolidieren oder mit einem einheitlichen Überbau zu kontrollieren. Solche Lösungen werden oft unter der Bezeichnung „Cross Chain Communication“ oder „Interchain Communication“ geführt. Beispiele für entsprechende Rahmenwerke sind Cosmos [147], Polkadot [148] oder Overledger [149], die Interoperabilität und sichere Kommunikation zwischen verschiedenen Blockchains versprechen. Dabei gibt es oft eine übergeordnete Blockchain mit einem globalen Konsensmechanismus und viele untergeordnete spezialisierte Blockchains mit unabhängiger Datenverarbeitung. Die Herstellung von Vertrauen (siehe auch Abschnitt 3.1) kann bei solchen Konzepten aber sehr komplex werden.

11.4 Kanalnetzwerke

Um die Skalierbarkeit und die Privatsphäre in einer (öffentlichen) Blockchain zu verbessern, gibt es die Möglichkeit, Transaktionen aus der Blockchain auszulagern und in einem eigenen Netz-

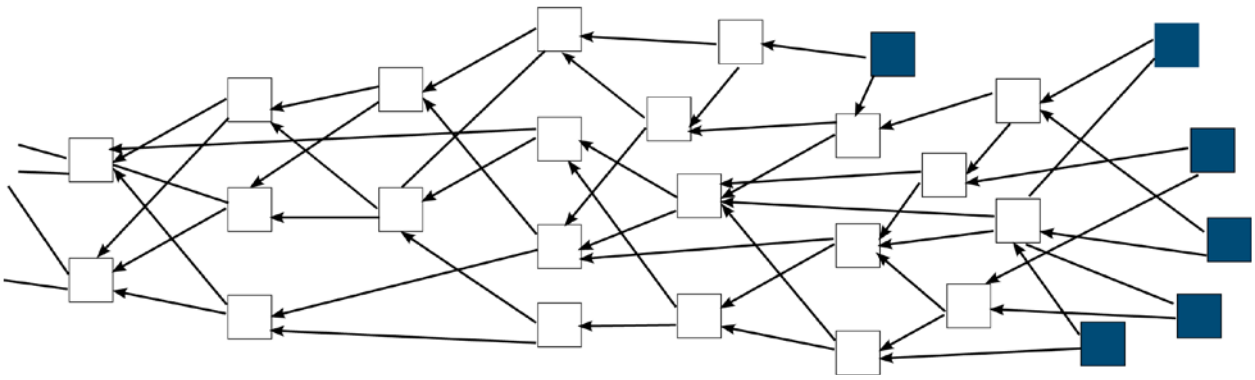

Abbildung 13: Gerichteter azyklischer Graph bei IOTA („Tangle“), vgl. [146]

werk abzuwickeln. Dazu wird ein Kanal zwischen zwei Teilnehmern in dem Netzwerk eingerichtet, über den Zahlungen oder allgemeiner Daten bis zu einem vereinbarten Limit ausgetauscht werden können. Nur die Öffnung und die Schließung des Kanals mit den entsprechenden Salden werden durch eine Transaktion in der Blockchain festgehalten. Durch einen Smart Contract auf der Blockchain kann korrektes Verhalten der Teilnehmer in dem Zahlungskanal forciert werden. Vorteile eines solchen Kanalnetzwerks sind, dass Zahlungen innerhalb der Kanäle gebührenfrei sind (wodurch auch Mikrozahlungen realisiert werden können), der Datendurchsatz im Prinzip nur durch die Kapazität des Netzwerks beschränkt ist (und nicht durch die Blockgröße oder -rate der Blockchain) und zugleich die Privatsphäre der Teilnehmer besser geschützt ist als auf der Blockchain, da nicht jede einzelne Transaktion auf der Blockchain gespeichert wird. Das bekannteste Beispiel für ein solches Netzwerk von Zahlungskanälen ist das Lightning-Netzwerk von Bitcoin [150].

11.5 Quanten-Blockchain

Im Abschnitt 6.1 wurde bereits die potenzielle Bedrohung von Blockchains durch Quantencomputer und eine mögliche Mitigation durch die Verwendung quantencomputerresistenter Kryptografie und kryptoagiler Lösungen dargestellt.

Andererseits gibt es bereits Ideen, die Blockchain-Technologie durch die Nutzung von quantenmechanischen Konzepten anzureichern. Hinter dem Stichwort „Quanten-Blockchain“ können dabei verschiedene Ansätze stecken:

- Die Blockchain baut auf einem Quantum-Key-Distribution (QKD)-Netzwerk auf. Die Blockchain-Teilnehmer können dadurch sicher Schlüssel austauschen und sich gegenseitig authentisieren. Die nachfolgende Netzwerkkommunikation wird dann über diese Schlüssel abgesichert. Ein solches System ist bereits implementiert und getestet worden [151].
- Das Blockchain-Netzwerk selbst besteht aus Quantencomputern, die Transaktionen in den Blöcken werden durch verschränkte Photonen repräsentiert und aufeinanderfolgende Blöcke werden durch zeitliche Verschränkung miteinander verknüpft [152]. Da noch keine vollwertigen Quantencomputer zur Verfügung stehen und auch die konkreten quantenphysikalischen Eigenschaften zeitlich verschränkter Photonen noch unerforscht sind, sind solche Ideen bisher rein theoretischer Natur.
- Blockchain-Transaktionen werden durch Quanten-Teleportation in einem klassischen Netzwerk übermittelt [153]. Dadurch ist kein Double-Spending-Angriff (siehe Abschnitt 7.1.2) möglich, weil Quanteninformation im Allgemeinen nicht zuverlässig geklont werden kann. Verifiziert und abgesichert werden die Transaktionen durch Quantenkryptografie (kryptografische Verfahren basierend auf quantenmechanischen Effekten, z. B. QKD). Solche Konzepte sind heute noch sehr spekulativ, da die quantenphysikalischen Grundlagen für derartige Anwendungen noch nicht ausgereift und zum Teil noch im experimentellen Stadium sind.

11.6 Blockchain und Künstliche Intelligenz

Eine weitere mögliche Verknüpfung von Technologien besteht in der Kombination von Blockchains mit Künstlicher Intelligenz.

Der in der KI-Forschung noch relativ neue Ansatz des kooperativen Maschinenlernens ermöglicht Geräten in einem verteilten Netzwerk die gemeinsame Bildung eines geteilten Vorhersagemodells, wobei die Trainingsdaten lokal auf den einzelnen Geräten verbleiben. Ein Problem in diesem Zusammenhang sind bösartige Teilnehmer, die die Korrektheit des Trainings stören können. Mit einem Blockchain-System als Grundlage könnte ein Anreiz für die regelkonforme Teilnahme am kooperativen Lernen geschaffen und eine Möglichkeit für einen zuverlässigen Audit des Systems

geschaffen werden. Mit DeepChain [154] gibt es zum Beispiel eine prototypische Umsetzung dieser Idee auf der Basis des Blockchain-Systems Corda.

Ein anderer Ansatz besteht darin, in einem Blockchain-Netzwerk Knoten zusammenzubringen, die neuronale Netze aufsetzen und trainieren oder bestehende neuronale Netze nutzen wollen (gegen Bezahlung), die ihre neuronalen Netze oder Trainingsdaten zur Verfügung stellen können (gegen Entlohnung) oder die als Miner ihre Rechenleistung für die neuronalen Netze nutzbar machen (gegen Entlohnung). In einem solchen Einsatzszenario müssen geeignete Konsensmechanismen für die Blockchain entwickelt werden, die neben dem klassischen Proof-of-Work für die Miner garantieren können, dass Trainingsdaten für das Training eines neuronalen Netzes verwendet wurden oder dass ein neuronales Netz die korrekten Schlussfolgerungen liefert. Ein Beispiel aus dem Gesundheitswesen ist das Projekt Skychain [155], das sich noch in der Testphase befindet.

Zusammenfassung.

- Es gibt verschiedene Ideen, die Blockchain-Technologie zu verallgemeinern oder mit anderen neuen Technologien zu kombinieren.
- Die meisten Weiterentwicklungen von Blockchain sind noch nicht ausgereift und sicherheitstechnisch nicht untersucht.
- Interoperabilität zwischen Blockchains ist ohne Standardisierung schwer zu erreichen.

12 Standards und Regulierung

Da Blockchain eine relativ neue Technologie ist, gibt es im Bereich der Standardisierung und Regulierung bisher noch wenige Ergebnisse. Durch den breiten Anwendungsspielraum von Blockchains wäre aber in vielen Bereichen eine Vereinheitlichung und Steuerung der Technologie sinnvoll und vorteilhaft. Neben der IT-Sicherheit stellen sich Fragen insbesondere im rechtlichen, ökonomischen und soziotechnischen Rahmen. In diesem Kapitel sollen kurz die bestehenden Bestrebungen und ersten Resultate aus Standardisierung und Regulierung im nationalen und internationalen Kontext dargestellt werden (Stand Januar 2019), wobei der Fokus auf Fragen der IT-Sicherheit liegt. Branchenspezifische Regelwerke können in diesem Zusammenhang nicht betrachtet werden.

12.1 Standardisierung

Im Bereich der Standardisierung von Blockchains ist in der letzten Zeit viel Aktivität zu verzeichnen. Die Blockchain-Technologie hat inzwischen ein gewisses Maß an Reife und Verbreitung gefunden und es wird in vielen Anwendungsbereichen über einen Einsatz von Blockchains nachgedacht. Gleichzeitig offenbart sich aber auch, dass das Verständnis über die Definitionen und Grenzen der Technologie zum Teil weit auseinander geht und es sehr unterschiedliche Umsetzungen gibt, die nicht interoperabel sind (siehe Abschnitt 11.3).

Im praktischen Einsatz haben sich inzwischen einige Quasi-Standards herausgebildet, z. B. die Open-Source-Blockchain-Plattform Corda von R3 [54] oder das Blockchain-Framework Hyperledger der Linux Foundation [8]. Auch diese Lösungen sind untereinander nicht kompatibel.

Die Entwicklung von Standards zur Blockchain-Technologie kann dazu beitragen, die Rahmenbedingungen der Technologie zu klären und den Einsatz von Blockchains zu erleichtern.

Bei der International Organization for Standardization (ISO) ist bereits Anfang 2016 eine Kommission (TC 307, [156]) zur Erarbeitung eines Standards zu Blockchain und Distributed-Ledger-Technologien unter der Leitung von Australien gegründet worden. Das Deutsche Institut für Normung (DIN) hat ein entsprechendes Spiegelgremium eingerichtet, in dem das BSI beobachtendes Mitglied ist. Fragen zur IT-Sicherheit werden in der Arbeitsgruppe „security, privacy and identity“ behandelt und für den Standard aufbereitet. Der Abschluss der Standardisierungsarbeiten ist für 2021 geplant.

Beim European Telecommunications Standards Institute (ETSI) gibt es ein Gremium (ISG PDL, [157]), das an der Standardisierung von genehmigungsbasierten Distributed Ledgers arbeitet. Das Institute of Electrical and Electronics Engineers (IEEE) führt Standardisierungsprojekte im Blockchain-Umfeld durch (Publikationsreihe P2418, [158]), in denen unter anderem Blockchain-Rahmenwerke für den Bereich IoT und Automotive entwickelt werden. Das US-amerikanische National Institute of Standards and Technology (NIST) sowie das American National Standards Institute (ANSI) arbeiten ebenfalls am Thema Blockchain. Auf der Netzwerkebene beschäftigen sich die Internet Engineering Task Force (IETF) und das World Wide Web Consortium (W3C) mit Internet-Standards für eine einheitliche Identifikation der im Netzwerk zugreifbaren Ressourcen und Adressierung für Blockchains [159], [160], die für eine Interoperabilität der Netzwerkkommunikation zwischen verschiedenen Blockchain-Systemen sorgen sollen.

12.2 Regulierung

Ein immer wieder herausgestellter Vorteil der Blockchain-Technologie ist die Möglichkeit, Transaktionen zwischen den Teilnehmern ohne eine zentrale Stelle (oder zumindest mit weniger

zentralen Abhängigkeiten, siehe Abschnitt 3.1) abwickeln zu können. Das bringt aber auch Probleme und Unsicherheiten mit sich, zum Beispiel im Schadens- oder Streitfall (siehe Abschnitt 8.1) oder bei der Reaktion auf Sicherheitsprobleme. Zudem bringt die in vielen öffentlichen Blockchains gewollte Pseudonymisierung für bestimmte Anwendungen auch Unsicherheiten mit sich. Unter anderem deshalb ist eine staatliche Regulierung für viele Einsatzbereiche sinnvoll. Da Blockchains oftmals einen transnationalen Charakter haben, ist eine internationale Abstimmung zur Regelung und Steuerung der Blockchain-Technologie in vielen Fällen notwendig.

Bei der Europäischen Union (EU) gibt es seit 2018 eine Zusammenarbeit von EU-Mitgliedsstaaten und Norwegen im Rahmen des European Blockchain Partnership (EBP, [161]), die zum Ziel hat, eine transnationale europäische Blockchain-Infrastruktur (European Blockchain Services Infrastructure, EBSI) zu schaffen, die die Einrichtung einer grenzüberschreitenden digitalen öffentlichen Verwaltung unterstützen soll. Im Rahmen dieser Zusammenarbeit sollen interoperable Rahmenwerke, standardisierte Lösungen und Regelungs- und Steuerungsstrukturen für den Blockchain-Einsatz entwickelt werden. Außerdem sollen Forschung und Entwicklung im Blockchain-Umfeld in Europa gestärkt werden. Ein Schwerpunkt liegt dabei auf Sicherheit und Datenschutz. Erste Ergebnisse werden ab 2019 erwartet.

In Deutschland erstellt die Bundesregierung – aufbauend auf dem Koalitionsvertrag 2018 – eine nationale Blockchain-Strategie, die im Sommer 2019 vorgelegt werden soll. Schwerpunkte dieser Strategie sind die Schaffung eines sicheren Rechtsrahmens, die Unterstützung von Pilotprojekten, die Identifikation von Anwendungsfeldern in der öffentlichen Verwaltung, die Stärkung der Forschung und die Vernetzung von Verwaltung, Wirtschaft und Wissenschaft. In einer öffentlichen Konsultation wird vorab die Meinung und Expertise von verschiedenen Stakeholdern im Blockchain-Bereich eingeholt, um die Bedürfnisse aller Betroffenen berücksichtigen zu können. Das BSI wird bei der Ausarbeitung der Blockchain-Strategie der Bundesregierung im Themenumfeld der IT-Sicherheit zu Rate gezogen. Zusätzlich sind in Deutschland verschiedene Behörden für die konkrete Regulierung in unterschiedlichen Einsatzbereichen der Blockchain-Technologie zuständig. So hat zum Beispiel die Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) bereits eine aufsichtsrechtliche Einordnung von Kryptowährungen und Initial Coin Offerings (ICOs) erarbeitet [162].

Im Bereich der IT-Sicherheit ist das BSI als die nationale Cybersicherheitsbehörde zuständig für die Untersuchung und Bewertung der Blockchain-Technologie. Eine erste Zusammenstellung der Potenziale und Herausforderungen von Blockchains im Hinblick auf ihre sicherheitstechnischen Eigenschaften sowie eine Einschätzung der technisch-kryptografischen Mechanismen und entsprechende Empfehlungen für den sicheren Einsatz von Blockchains liefert das vorliegende Dokument. Daneben kann zukünftig eine Sicherheitszertifizierung von Blockchain-Produkten oder ausgewählten Komponenten für bestimmte Anwendungen sinnvoll sein. Voraussetzung dafür ist die Verwendung von kryptografischen Algorithmen, die vom BSI für Blockchain-Anwendungen anerkannt wurden, da eine individuelle Bewertung proprietärer Algorithmen im Rahmen einer Zertifizierung grundsätzlich nicht möglich ist.

Zusammenfassung.

- Ein einheitliches Verständnis der Blockchain-Technologie ist wichtig für sichere und interoperable Blockchain-Anwendungen.
- Die Standardisierung im Blockchain-Bereich wird von verschiedenen Organisationen vorangetrieben, zeigt aber bisher noch wenige Ergebnisse.
- An der Regulierung von Blockchains wird sowohl auf nationaler als auch auf internationaler Ebene intensiv gearbeitet.

Glossar

Hier werden einige kryptografische Grundbegriffe eingeführt, soweit sie zum Verständnis des vorliegenden Dokuments erforderlich sind. Die Erläuterungen sind dabei oft eng an die Technische Richtlinie TR-02102 des BSI [50] angelehnt.

Asymmetrische Kryptoverfahren können zur Erstellung *digitaler Signaturen* für die Authentisierung von Daten und zur Verschlüsselung von Daten verwendet werden. Anders als bei symmetrischen Kryptoverfahren ist es nicht erforderlich, dass die kommunizierenden Parteien Kenntnis über einen gemeinsamen geheimen Schlüssel haben. Stattdessen erzeugt jeder Nutzer ein Schlüsselpaar bestehend aus einem öffentlichen und einem geheimen, nur ihm bekannten Schlüssel. Dabei darf es praktisch nicht möglich sein, den geheimen Schlüssel aus dem öffentlichen Schlüssel zu rekonstruieren. Mithilfe des öffentlichen Schlüssels können Daten für den Besitzer des privaten Schlüssels verschlüsselt und dessen Signaturen geprüft werden.

Digitale Signaturen werden zur Authentisierung von Daten verwendet. In Signaturverfahren auf Basis *asymmetrischer Kryptoverfahren* werden die zu signierenden Daten zunächst gehasht und dann aus diesem Hashwert die Prüfsumme bzw. die Signatur mit dem geheimen Schlüssel des Beweisenden berechnet. Der Prüfer verifiziert dann die Signatur mit dem entsprechenden öffentlichen Schlüssel. Eine erfolgreiche Prüfung garantiert, dass die entsprechenden Daten in der Zwischenzeit nicht durch Unbefugte verändert wurden. Für digitale Signaturen werden im Moment häufig die Verfahren *RSA* und *ECDSA* eingesetzt.

Das **Diskreter-Logarithmus-Problem (DLP)** auf einer elliptischen Kurve E besteht darin, aus den Punkten $P \in E$ und $aP \in E$ den Wert $a \in \mathbb{N}$ zu berechnen.

ECDSA (Elliptic Curve Digital Signature Algorithm) ist ein asymmetrisches Verfahren für *digitale Signaturen*. Seine Sicherheit beruht auf der vermuteten Schwierigkeit des DLP auf elliptischen Kurven.

Einweg-Eigenschaft einer Hashfunktion bedeutet, dass es für einen gegebenen Hashwert praktisch unmöglich ist, einen Bitstring mit diesem Hashwert zu finden.

Entropie von Daten bezeichnet deren mittleren Informationsgehalt. Daten mit einer niedrigen Entropie sind redundant oder enthalten statistische Regelmäßigkeiten.

Das **Faktorisierungsproblem** im Kontext von RSA besteht darin, eine große Zahl $n = pq$ in ihre Primteiler p und q zu faktorisieren.

Hashfunktionen bilden einen Bitstring beliebiger Länge auf einen Bitstring fester Länge ab. Diese Funktionen spielen in vielen kryptografischen Verfahren eine große Rolle, so zum Beispiel bei der Ableitung kryptografischer Schlüssel oder bei der Datenauthentisierung. Hashfunktionen, die in kryptografischen Verfahren eingesetzt werden, müssen je nach Anwendung die drei Bedingungen *Einweg-Eigenschaft, Zweiturbild-Eigenschaft* und *Kollisionsresistenz* erfüllen. Eine Hashfunktion, die alle drei Bedingungen erfüllt, heißt kryptografisch stark.

Homomorphe Verschlüsselung bezeichnet Verschlüsselungsverfahren, bei denen eine mathematische Operation wie Addition oder Multiplikation auf dem Chiffrat nach der Entschlüsselung zu einer entsprechenden Operation auf dem Klartext führt.

Kollisionsresistenz einer Hashfunktion bedeutet, dass es praktisch unmöglich ist, zwei verschiedene Bitstrings mit gleichem Hashwert zu finden.

Kryptoanalyse bezeichnet die Analyse von kryptografischen Verfahren, um sie zu brechen oder ihre Sicherheit zu beweisen.

Ein **Message Authentication Code (MAC)** ist ein symmetrisches Verfahren zur Datenauthentisierung. Die kommunizierenden Parteien müssen also vorab einen gemeinsamen symmetrischen Schlüssel vereinbart haben.

Eine **Public-Key-Infrastruktur (PKI)** wird eingesetzt, um in einem *asymmetrischen Kryptoverfahren* die Zuordnung der öffentlichen Schlüssel zu einem festen Kommunikationspartner zu garantieren und sogenannte Man-in-the-Middle-Angriffe zu verhindern, bei denen die Kommunikation zwischen zwei Parteien von einem Angreifer abgefangen und manipuliert wird. In einer PKI wird die Zugehörigkeit eines öffentlichen Schlüssels zu einem Kommunikationspartner durch eine zentrale vertrauenswürdige Instanz attestiert.

RSA ist ein asymmetrisches Verfahren für *digitale Signaturen*. Die Sicherheit von RSA beruht auf der vermuteten Schwierigkeit der Faktorisierung großer Zahlen $n = pq$ in ihre Primteiler p und q.

Secret-Sharing-Verfahren erlauben es, ein Geheimnis in Teilgeheimnisse aufzuteilen, sodass zur Rekonstruktion des Geheimnisses alle Teilgeheimnisse (oder eine bestimmte Teilmenge davon) erforderlich sind. Damit kann beispielsweise ein geheimer Schlüssel so auf verschiedene Anwender verteilt werden, dass eine gewisse Mindestzahl von ihnen benötigt wird, um den Schlüssel zu rekonstruieren.

Zweiturbild-Eigenschaft einer Hashfunktion bedeutet, dass es für einen gegebenen Bitstring praktisch unmöglich ist, einen anderen Bitstring mit dem gleichen Hashwert zu finden.

Index

Abkürzungsverzeichnis

API	Application Programming Interface	30
BFT	Byzantine Fault Tolerance	21
CFT	Crash Fault Tolerance	21
DAG	directed acyclic graph	72
dAPP	distributed Application	28
DDoS	Distributed Denial of Service	48
DLP	Diskreter-Logarithmus-Problem	78
DLT	Distributed-Ledger-Technologie	72
dPoS	delegated Proof-of-Stake	72
DSGVO	Datenschutz-Grundverordnung	61
ECDSA	Elliptic Curve Digital Signature Algorithm	78
EVM	Ethereum Virtual Machine	28
ICO	Initial Coin Offering	77
IoT	Internet of Things	73
KI	Künstliche Intelligenz	72
MAC	Message Authentication Code	78
P2P	Peer-to-Peer	9
PBFT	Practical Byzantine Fault Tolerance	23
PKI	Public-Key-Infrastruktur	79
PoET	Proof-of-Elapsed-Time	24
PoS	Proof-of-Stake	24
PoW	Proof-of-Work	23
PoX	Proof-of-X	23
QKD	Quantum Key Distribution	74
sMPC	secure Multi-Party Computation	36
TEE	Trusted Execution Environment	36
tps	Transaktionen pro Sekunde	18
ZK	Zero Knowledge	35

Literaturverzeichnis

[1] BSI, *Blockchain sicher gestalten – Eckpunkte des BSI*, 2018. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Eckpunktepapier.pdf.

[2] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.

[3] Bitcoin Wiki, *Technical background of version 1 Bitcoin addresses*, 2018. https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses, abgerufen: 2019-01-30.

[4] Ethereum, *White Paper: A Next-Generation Smart Contract and Decentralized Application Platform*. https://github.com/ethereum/wiki/wiki/White-Paper, abgerufen: 2018-04-17.

[5] G. Wood, *Yellow Paper: A Secure Decentralised Generalised Transaction Ledger, EIP-150 Revision*. http://paper.gavwood.com/, abgerufen: 2018-04-11.

[6] M. Dameron, *Beigepaper: An Ethereum Technical Specification, Version 0.7.2*, 2018. https://github.com/chronaeon/beigepaper/blob/master/beigepaper.pdf, abgerufen: 2018-04-16.

[7] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco und J. Yellick, *Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains*, 2018. https://arxiv.org/pdf/1801.10228.

[8] Hyperledger Fabric, *hyperledger-fabricdocs master documentation*, 2019. http://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html, abgerufen: 2019-01-30.

[9] BSI, *IT-Grundschutz-Kompendium – Glossar*, 2019. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2021.html.

[10] Digiconomist, *Bitcoin Energy Consump-tion Index*, 2017. https://digiconomist.net/bitcoin-energy-consumption, abgerufen: 2018-07-31.

[11] A. N. Bessani, J. Sousa und E. A. Pelinson, *State Machine Replication for the Masses with BFT-SMART*, in *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2014*, Atlanta, GA, USA, June 23–26, 2014, 2014, S. 355–362.

[12] BSI, *IT-Grundschutz-Kompendium – APP.4.3 Relationale Datenbanksysteme*, 2019. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/06_APP_Anwendungen/APP_4_3_Relationale_Datenbanksysteme_Edition_2021.html.

[13] Visa Inc., *Visa Inc. at a Glance*, 2015. https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf, abgerufen: 2017-08-21.

[14] R. Anderson, I. Shumailov, M. Ahmed und A. Rietmann, *Bitcoin Redux*, 2013. https://www.cl.cam.ac.uk/~rja14/Papers/bitcoin-redux.pdf.

[15] C. Cachin, R. Guerraoui und L. E. T. Rodrigues, *Introduction to Reliable and Secure Distributed Programming (2. Auflage)*, Springer, 2011.

[16] C. Dwork, N. A. Lynch und L. J. Stockmeyer, *Consensus in the presence of partial synchrony*, J. ACM, 35, Nr. 2, S. 288–323, 1988.

[17] L. Lamport, R. E. Shostak und M. C. Pease, *The Byzantine Generals Problem*, ACM Trans. Program. Lang. Syst., 4, Nr. 3, S. 382–401, 1982.

[18] M. J. Fischer, N. A. Lynch und M. Paterson, *Impossibility of Distributed Consensus with One Faulty Process*, J. ACM, 32, Nr. 2, S. 374–382, 1985.

[19] L. Lamport, *The Part-Time Parliament*, ACM Trans. Comput. Syst., 16, Nr. 2, S. 133–169, 1998.

[20] D. Ongaro und J. K. Ousterhout, *In Search of an Understandable Consensus Algorithm*, in *2014 USENIX Annual Technical Conference, USENIX ATC '14*, Philadelphia, PA, USA, June 19–20, 2014, 2014, S. 305–319.

[21] M. Castro und B. Liskov, *Practical byzantine fault tolerance and proactive recovery*, ACM Trans. Comput. Syst., 20, Nr. 4, S. 398–461, 2002.

[22] C. Cachin und M. Vukolic, *Blockchain Consensus Protocols in the Wild*, 2017. http://arxiv.org/abs/1707.01873.

[23] V. Buterin et al., *Proof of Stake FAQ*, 2018. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs, abgerufen: 2018-10-01.

[24] Hyperledger Sawtooth, *PoET 1.0 Specification*, 2017. https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html, abgerufen: 2018-09-28.

[25] D. Schwartz, N. Youngs und A. Britto, *The Ripple Protocol Consensus Algorithm*, 2014. https://ripple.com/files/ripple_consensus_whitepaper.pdf, abgerufen: 2018-01-05.

[26] D. Mazieres, *Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*, 2015. https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[27] Y. Gilad, R. Hemo, S. Micali, G. Vlachos und N. Zeldovich, *Algorand: Scaling Byzantine Agreements for Cryptocurrencies*, 2017. http://eprint.iacr.org/2017/454.

[28] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn und G. Danezis, *Consensus in the Age of Blockchains*, 2017. http://arxiv.org/abs/1711.03936.

[29] N. Stifter, A. Judmayer, P. Schindler, A. Zamyatin und E. R. Weippl, *Agreement with Satoshi – On the Formalization of Nakamoto Consensus*, 2018. https://eprint.iacr.org/2018/400.

[30] E. Heilman, N. Narula, T. Dryja und M. Virza, *IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency*, 2017. https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md, abgerufen: 2018-01-18.

[31] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri und V. Sassone, *PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain*, in *Proceedings of the Second Italian Conference on Cyber Security*, Milan, Italy, February 6th–9th, 2018, 2018.

[32] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef und E. Zenner, *Ripple: Overview and Outlook*, in *Trust and Trustworthy Computing – 8th International Conference, TRUST 2015*, Heraklion, Greece, August 24–26, 2015, Proceedings, 2015, S. 163–180.

[33] S. Thomas, *Correction to Ripple White Paper*, 2015. https://ripple.com/dev-blog/correction-to-ripple-white-paper/, abgerufen: 2018-01-05.

[34] C. A. E. Goodhart, *Problems of Monetary Management: The UK experience*, Papers in Monetary Economics, 1, 1975.

[35] H. Siebert, *Der Kobra-Effekt: Wie man Irrwege der Wirtschaftspolitik vermeidet*, Piper, 2003.

[36] N. Szabo, *Smart Contracts*, 1994. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html, abgerufen: 2018-08-13.

[37] Bitcoin Wiki, *Script*, 2018. https://en.bitcoin.it/wiki/Script, abgerufen: 2018-08-15.

[38] P. L. Seijas, S. Thompson und D. McAdams, *Scripting Smart Contracts for distributed ledger technology*, 2016. https://eprint.iacr.org/2016/1156.pdf.

[39] Etherscan, *The Ethereum Block Explorer*. https://etherscan.io/.

[40] Quorum, *Whitepaper*, 2016. https://github.com/jpmorganchase/quorum-docs/raw/master/Quorum%20Whitepaper%20v0.1.pdf, abgerufen: 2018-08-08.

[41] Codius, 2018. https://codius.org/, abgerufen: 2018-07-30.

[42] S. Thomas und E. Schwartz, *A Protocol for Interledger Payments*, 2015. https://interledger.org/interledger.pdf, abgerufen: 2019-02-08.

[43] Nxt, *The Nxt API*, 2018. https://nxtwiki.org/wiki/The_Nxt_API, abgerufen: 2018-08-03.

[44] Oraclize, *A Scalable Architecture for On-Demand, Untrusted Delivery of Entropy*. http://www.oraclize.it/papers/random_datasource-rev1.pdf, abgerufen: 2019-02-08.

[45] J. Peterson, J. Krug, M. Zoltu, A. K. Williams und S. Alexander, *Augur: a Decentralized Oracle and Prediction Market Platform*, 2018. http://www.augur.net/whitepaper.pdf, abgerufen: 2018-04-24.

[46] realitio, *Realitio – Crowd-sourced verification for smart contracts*. https://realit.io/, abgerufen: 2019-02-12.

[47] C. Pierrot und B. Wesolowski, *Malleability of the blockchain's entropy*, 2016. https://hal.archives-ouvertes.fr/hal-01364045/file/paper.pdf.

[48] A. Reutov, *Predicting Random Numbers in Ethereum Smart Contracts*, 2018. https://blog.positive.com/predicting-random-numbers-in-ethereum-smart-contracts-e5358c6b8620, abgerufen: 2018-05-22.

[49] J. Lung, *Ethereum network-level transaction spamming attack against honest participants*, 2018. https://github.com/randao/randao/issues/18, abgerufen: 2018-05-17.

[50] BSI, *Technische Richtlinie TR-02102 Kryptographische Verfahren: Empfehlun-gen und Schlüssellängen*, 2018. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.html.

[51] European Telecommunications Standards Institute (ETSI), *TS 119 312 Electronic Signatures and Infrastructures (ESI); Cryptographic Suites*, 2017. https://www.etsi.org/deliver/etsi_ts/119300_119399/119312/01.02.01_60/ts_119312v010201p.pdf.

[52] *Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG*, 2014. http://eur-lex.europa.eu/legal-content/DE/TXT/?uri=-CELEX:32014R0910.

[53] Hyperledger Fabric, *Channels*, 2018. https://hyperledger-fabric.readthedocs.io/en/release-1.3/channels.html, abgerufen: 2019-01-25.

[54] R. G. Brown, *The Corda Platform: An Introduction*, 2018. https://www.corda.net/content/corda-platform-whitepaper.pdf, abgerufen: 2019-01-25.

[55] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. M. Johnson, A. Juels, A. Miller und D. Song, *Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contract Execution*, 2018. http://arxiv.org/abs/1804.05141.

[56] M. Brandenburger, C. Cachin, R. Kapitza und A. Sorniotti, *Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric*, 2018. http://arxiv.org/abs/1805.08541.

[57] G. Zyskind, O. Nathan und A. Pentland, *Enigma: Decentralized Computation Platform with Guaranteed Privacy*, 2015. http://arxiv.org/abs/1506.03471.

[58] BSI, *AIS 46 Informationen zur Evaluierung von kryptographischen Algorithmen und ergänzende Hinweise für die Evaluierung von Zufallszahlengeneratoren*, 2013. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_46_pdf.pdf.

[59] BSI, *AIS 20 Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren*, 2013. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_pdf.pdf.

[60] BSI, *AIS 31 Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren*, 2013. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_pdf.pdf.

[61] S. D. Galbraith, *Mathematics of Public Key Cryptography*, New York, NY, USA: Cambridge University Press, 2012.

[62] A. Biryukov, D. Khovratovich und I. Pustogarov, *Deanonymisation of Clients in Bitcoin P2P Network*, in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, AZ, USA, November 3–7, 2014, 2014, S. 15–29.

[63] H. Al Jawaheri, M. Al Sabah, Y. Boshmaf und A. Erbad, *When A Small Leak Sinks A Great Ship: Deanonymizing Tor Hidden Service Users Through Bitcoin Transactions Analysis*, 2018. http://arxiv.org/abs/1801.07501.

[64] S. Goldfeder, H. A. Kalodner, D. Reisman und A. Narayanan, *When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies*, 2017. http://arxiv.org/abs/1708.04748.

[65] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker und S. Savage, *A fistful of Bitcoins: characterizing payments among men with no names*, Commun. ACM, 59, Nr. 4, S. 86–93, 2016.

[66] D. Ron und A. Shamir, *Quantitative Analysis of the Full Bitcoin Transaction Graph*, in *Financial Cryptography and Data Security – 17th International Conference, FC 2013*, Okinawa, Japan, April 1–5, 2013, Revised Selected Papers, 2013, S. 6–24.

[67] C. Kinkeldey, J.-D. Fekete und P. Isenberg, *BitConduite: Visualizing and Analyzing Activity on the Bitcoin Network, in Eurographics Conference on Visualization, EuroVis 2017*, Posters, Barcelona, Spain, 12–16 June 2017, 2017, S. 25–27.

[68] Chainalysis, *Chainalysis – Blockchain analysis*, 2019. https://www.chainalysis.com/, abgerufen: 2019-01-25.

[69] R. Klusman und T. Dijkhuizen, *Deanonymisation in Ethereum Using Existing Methods for Bitcoin*, 2018. https://pdfs.semanticscholar.org/bb20/de5fca19392e92c945685de190337bc1e0bf.pdf, abgerufen: 2019-01-25.

[70] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse und E. G. Sirer, *Decentralization in Bitcoin and Ethereum Networks*, 2018. http://arxiv.org/abs/1801.03998.

[71] E. Duffield und D. Diaz, *Dash: A Payments-Focused Cryptocurrency*, 2018. https://github.com/dashpay/dash/wiki/Whitepaper, abgerufen: 2019-01-25.

[72] Monero Research Lab, *Monero – secure, private, untraceable*. https://ww.getmonero.org/resources/research-lab/, abgerufen: 2019-01-25.

[73] E. Ben-Sasson, A. Chiesa, E. Tromer und M. Virza, *Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture*, in *Proceedings of the 23rd USENIX Security Symposium*, San Diego, CA, USA, August 20–22, 2014, 2014, S. 781–796.

[74] E. Ben-Sasson, I. Bentov, Y. Horesh und M. Riabzev, *Scalable, transparent, and post-quantum secure computational integrity*, 2018. http://eprint.iacr.org/2018/046.

[75] Zcash, *How it works*, 2018. https://z.cash/technology/, abgerufen: 2019-01-25.

[76] G. Kappos, H. Yousaf, M. Maller und S. Meiklejohn, *An Empirical Analysis of Anonymity in Zcash*, in *27th USENIX Security Symposium, USENIX Security 2018*, Baltimore, MD, USA, August 15–17, 2018, 2018, S. 463–477.

[77] J. Quesnelle, *On the linkability of Zcash transactions*, 2017. http://arxiv.org/abs/1712.01210.

[78] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan und N. Christin, *An Empirical Analysis of Traceability in the Monero Blockchain*, PoPETs, 2018, Nr. 3, S. 143–163, 2018.

[79] N. Houy, *The Bitcoin Mining Game*, Ledger, 1, S. 53–68, 2016. https://ledgerjournal.org/ojs/index.php/ledger/article/view/13.

[80] J. A. Garay, J. Katz, U. Maurer, B. Tackmann und V. Zikas, *Rational Protocol Design: Cryptography against Incentive-Driven Adversaries*, in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, Berkeley, CA, USA, 26–29 October, 2013, 2013, S. 648–657.

[81] F. K. Wilhelm, R. Steinwandt, B. Langenberg, P. J. Liebermann, A. Messinger und P. K. Schuhmacher, *Entwicklungsstand Quantencomputer*, 2017. www.bsi.bund.de/qcstudie.

[82] C. Berghoff, U. Korte, T. Kusber und S. Schwalm, *Langfristige Beweiswerterhaltung und Datenschutz in der Blockchain*, in *Dach Security 2018*, Gelsenkirchen, 2018.

[83] J. Wilmoth, *'Nuclear Option': ABC Dev Won't Rule out Changing Bitcoin Cash PoW Algorithm*, 2018. https://www.ccn.com/nuclear-option-abc-dev-wont-rule-out-changing-bitcoin-cash-pow-algorithm/, abgerufen: 2019-01-30.

[84] P. W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM J. Comput., 26, Nr. 5, S. 1484–1509, 1997.

[85] J. J. Roberts und N. Rapp, *Exclusive: Nearly 4 Million Bitcoins Lost Forever, New Study Says*, 2017. fortune.com/2017/11/25/lost-bitcoins/, abgerufen: 2019-01-30.

[86] *PoW 51% Attack Cost*, 2018. https://www.crypto51.app/, abgerufen: 2018-09-28.

[87] Trustnodes, *Bitcoin Gold 51% Attacked, $18 Million Stolen Through Double Spends*, 2018. https://www.trustnodes.com/2018/05/24/bitcoin-gold-51-attacked-18-million-stolen-double-spends.

[88] GoBitcoin.io, *Cost of a 51% attack*, 2019. https://gobitcoin.io/tools/cost-51-attack/, abgerufen: 2019-01-10.

[89] J. Kroll, I. Davey und E. Felten, *The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries*, Workshop on the Economics of Information Security, 2013.

[90] C. Decker und R. Wattenhofer, *Information Propagation in the Bitcoin Network*, 2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P), 2013.

[91] G. Bissas, B. N. Levine, A. P. Ozisik, G. Andresen und A. Houmansadr, *An Analysis of Attacks on Blockchain Consensus*, 2016. http://arxiv.org/abs/1610.07985.

[92] I. Eyal und E. Sirer, *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*, Lecture Notes in Computer Science, 8437, S. 436–454, 2014.

[93] M. Vasek, M. Thornton und T. Moore, *Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem*, Lecture Notes in Computer Science, 8438, S. 57–71, 2014.

[94] E. Heilman, A. Kendler, A. Zohar und S. Goldberg, *Eclipse Attacks on Bitcoin's Peer-to-Peer Network*, in *24th USENIX Security Symposium, USENIX Security 15*, Washington, D.C., USA, August 12–14, 2015, 2015, S. 129–144.

[95] Bitcoin Wiki, *Common Vulnerabilities and Exposures*, 2019. https://en.bitcoin.it/w/index.php?title=Common_Vulnerabilities_and_Exposures&oldid=66007, abgerufen: 2019-01-10.

[96] Coinwire, *Australian Firm Loses $6.6 Million from Soarcoin „Backdoor"*, 2018. https://www.coinwire.com/australian-firm-losses-6-6-million-from-soarcoin-backdoor.

[97] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig und E. Wustrow, *Elliptic Curve Cryptography in Practice*, Financial Cryptography and Data Security — 18th International Conference, S. 157–175, 2014.

[98] J. Breitner und N. Heninger, *Biased Nonce Sense: Lattice Attacks against Weak ECDSA Signatures in Cryptocurrencies*, 2019. https://eprint.iacr.org/2019/023.pdf.

[99] A. Roos, *Verge fällt Hacker zum Opfer – schon wieder*, 2018. https://www.btc-echo.de/verge-xvg-faellt-hacker-zum-opfer-schon-wieder/.

[100] C. Decker und R. Wattenhofer, *Bitcoin Transaction Malleability and MtGox*, in *Computer Security – ESORICS 2014*, Proceedings, Part II, 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014, 2014, S. 313–326.

[101] WhaleCalls, *A tl;dr on Antbleed*, 2017. https://medium.com/@whalecalls/a-tl-dr-antbleed-2406a5e34fcd, abgerufen: 2019-01-25.

[102] Bitmain, *Antminer Firmware Update*, April 2017. https://blog.bitmain.com/en/antminer-firmware-update-april-2017/, abgerufen: 2019-01-25.

[103] L. Luu, D.-H. Chu, H. Olickel, P. Saxena und A. Hobor, *Making Smart Contracts Smarter*, 2016. https://eprint.iacr.org/2016/633.

[104] Trustnodes, *Smart Contract Bug Nearly Freezes Transfers in $800 Million Worth of Icon Tokens*, 2018. https://www.trustnodes.com/2018/06/17/smart-contract-bug-nearly-freezes-transfers-800-million-worth-icon-tokens, abgerufen: 2018-08-22.

[105] N. Atzei, M. Bartoletti und T. Cimoli, *A survey of attacks on Ethereum smart contracts*, in *Principles of Security and Trust*, LNCS 10204, M. Maffei und M. Ryan, Hrsg., Springer, 2017, S. 164–186.

[106] S. Kalra, S. Goel, M. Dhawan und S. Sharma, *Zeus: Analyzing Safety of Smart Contracts*, Network and Distributed System Security Symposium, 2018. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_09-1_Kalra_paper.pdf.

[107] M. Bartoletti, S. Carta, T. Cimoli und R. Saia, *Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact*, 2017. http://arxiv.org/abs/1703.03779.

[108] M. Araoz, *Serpent Compiler Audit v1.0.0*, 2017. https://github.com/AugurProject/augur-audits/blob/master/serpent-compiler/Zeppelin Solutions – Serpent Compiler Audit v1.0.0.pdf, abgerufen: 2018-08-15.

[109] G. Destefanis, A. Bracciali, M. Marchesi, M. Ortu, R. Tonelli und R. Hierons, *Smart Contracts Vulnerabilities: A Call for Blockchain Software Engineering?*, 2018. https://www.researchgate.net/publication/323545752_Smart_Contracts_Vulnerabilities_A_Call_for_Blockchain_Software_Engineering.

[110] I. Nikolic, A. Kolluri, I. Sergey, P. Saxena und A. Hobor, *Finding The Greedy, Prodigal, and Suicidal Contracts at Scale*, 2018. http://arxiv.org/abs/1802.06038.

[111] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy und S. Zanella-Béguelin, *Formal Verification of Smart Contracts: Short Paper*, in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, ACM, 2016, S. 91–96.

[112] L. Luu, J. Teutsch, R. Kulkarni und P. Saxena, *Demystifying Incentives in the Consensus Computer*, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, S. 706–719, 2015.

[113] S. Jain, P. Saxena, F. Stephan und J. Teutsch, *How to verify computation with a rational network*, 2016. http://arxiv.org/abs/1606.05917.

[114] K. Delmolino, M. Arnett, A. Kosba, A. Miller und E. Shi, *Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab*, in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner und K. Rohloff, Hrsg., Springer, 2016, S. 79–94.

[115] C. McFarland, T. Hux, E. Wuehler und S. Campbel, *Blockchain Threat Report*, 2018. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-blockchain-security-risks.pdf.

[116] C. Cimpanu, *IOTA Cryptocurrency Users Lose $4 Million in Clever Phishing Attack*, 2018. https://www.bleepingcomputer.com/news/security/iota-cryptocurrency-users-lose-4-million-in-clever-phishing-attack/.

[117] Kyodo News, *Police start investigating Coincheck cryptocurrency theft*, 2018. https://english.kyodonews.net/news/2018/01/021a8a3c5844-update2-regulators-take-action-against-coincheck-after-cryptocurrency-theft.html.

[118] W. Suberg, *Vietnam: Betrügerische ICOs von Pincoin und Ifan stehlen rund 536 Mio*, 2018. https://de.cointelegraph.com/news/vietnam-pincoin-ifan-icos-exposed-as-scams-that-allegedly-stole-660-million.

[119] BSI, *IT-Grundschutz-Kompendium – Edition 2019*, 2019. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/it-grundschutz-kompendium_node.html.

[120] VDI Technologiezentrum GmbH, *Blockchain – Eine Technologie mit disruptivem Charakter, Kap. 7: Rechtliche Rahmenbedingungen der Blockchain, Version 1.0*, 2018. https://www.vditz.de/fileadmin/media/news/documents/Blockchain_-_Eine_Technologie_mit_disruptivem_Charakter.pdf, abgerufen: 2018-11-20.

[121] J. Schrey und T. Thalhofer, *Rechtliche Aspekte der Blockchain*, Neue juristische Wochenschrift, S. 1431, 2017.

[122] D. Saive, *Rückabwicklung von Blockchain-Transaktionen*, in *Tagungsband Herbstakademie 2018, Rechtsfragen digitaler Transformationen – Gestaltung digitaler Veränderungsprozesse durch Recht*, OLWIR Verlag, 2018, S. 371–380.

[123] P. J. Pesch, *Blockchain, Smart Contracts und Datenschutz*, in *Smart Contracts*, M. Fries und B. P. Paal, Hrsg., Mohr Siebeck, 2019, S. 13–23.

[124] R. Matzutt, J. Hiller, M. Henze, J. H. Ziegeldorf, D. Müllmann, O. Hohlfeld und K. Wehrle, *A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin*, in *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC), Santa Barbara*, Springer, 2018.

[125] A. Peters, *Sabotage von Blockchains durch Einschleusung strafrechtsrelevanter Inhalte?*, in *Tagungsband Herbstakademie 2018, Rechtsfragen digitaler Transformationen – Gestaltung digitaler Veränderungsprozesse durch Recht*, OLWIR Verlag, 2018, S. 381–396.

[126] S. H. Kimpel und C. Adcock, *The State of Smart Contract Legislation*, 2018. https://www.blockchainlegalresource.com/2018/09/state-smart-contract-legislation/, abgerufen: 2018-12-05.

[127] *Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung)*, 2016. https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex:32016R0679.

[128] M. Rost, *Standardisierte Datenschutzmodellierung*, Datenschutz und Datensicherheit, 6, S. 433–438, 2012.

[129] Blockcerts, *Introduction – Blockcerts: The Open Standard for Blockchain Credentials*. https://www.blockcerts.org/guide/, abgerufen: 2019-02-01.

[130] H. Krawczyk und T. Rabin, *Chameleon Hashing and Signatures*, 1998. http://eprint.iacr.org/1998/010.

[131] G. Ateniese, B. Magri, D. Venturi und E. R. Andrade, *Redactable Blockchain – or – Rewriting History in Bitcoin and Friends*, in *2017 IEEE European Symposium on Security and Privacy, Euro S&P 2017*, Paris, France, April 26–28, 2017, 2017, S. 111–126.

[132] I. Puddu, A. Dmitrienko und S. Capkun, *μchain: How to Forget without Hard Forks*, 2017. http://eprint.iacr.org/2017/106.

[133] G. Goldacker, *Digitale Souveränität*, 2017. https://www.oeffentliche-it.de/documents/10181/14412/Digitale+Souver%C3%A4nit%C3%A4t.

[134] BMWi, *Kompetenzen für eine digitale Souveränität*, 2017. https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/kompetenzen-fuer-eine-digitale-souveraenitaet.pdf.

[135] Plattform „Innovative Digitalisierung der Wirtschaft“, *Digitale Souveränität und Künstliche Intelligenz – Voraussetzungen, Verantwortlichkeiten und Handlungsempfehlungen*, 2018. https://www.de.digital/DIGITAL/Redaktion/DE/Textsammlung/digital-gipfel-plattform-digitalisierung-der-wirtschaft-fg1.html.

[136] D. Ron und A. Shamir, *How Did Dread Pirate Roberts Acquire and Protect his Bitcoin Wealth?*, in *Financial Cryptography and Data Security – FC 2014 Workshops, BITCOIN and WAHC 2014*, Christ Church, Barbados, March 7, 2014, Revised Selected Papers, 2014, S. 3–15.

[137] BSI, *Technische Richtlinie TR-03125 TR-ESOR Beweiswerterhaltung kryptographisch signierter Dokumente*, 2018. https://tr-esor.de.

[138] IT-Finanzmagazin, *Blockchain-Anwendung: Frankfurt School – fälschungssi-chere Abschlusszeugnisse für Studierende*, 2018. https://www.it-finanzmagazin.de/blockchain-frankfurt-school-zeugnis-77708/, abgerufen: 2019-01-30.

[139] regio IT, *Showcase „Zeugnisvalidierung über Blockchains“*, 2017. https://www.uni-speyer.de/files/de/Lehrst%C3%BChle/Hill/Neue_Veranstaltungen/Showcase_Niehues.pdf, abgerufen: 2019-01-30.

[140] BSI, *Technische Richtlinie TR-03147 Vertrauensniveaubewertung von Verfahren zur Identitätsprüfung natürlicher Perso-nen*, 2017. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03147/TR03147.html.

[141] F. Kirstein, M. Polzhofer und K.-P. Eckert, *Digitale Identitäten in der Blockchain – Erfahrungen aus der Entwicklung*, 2018.

[142] Nano, *Nano – an instant, zero-fee, scala-ble currency*, 2019. https://nano.org/en, abgerufen:2019-01-25.

[143] Dexon, *DEXON – Empowering the Decentralised Future*, 2019. https://dexon.org/, abgerufen: 2019-01-25.

[144] Bundesdruckerei, *Blockchain – Verbesserungspotential der Technologie*, 2018. https://www.bafin.de/SharedDocs/Downloads/DE/Veranstaltung/dl_180410_BaFinTech_2018_WS3_2.pdf?__blob=publicationFile&v=3, abgerufen: 2019-01-25.

[145] IOTA, *The Next Generation of Distributed Ledger Technology*, 2018. https://www.iota.org/, abgerufen: 2019-01-25.

[146] S. Popov, *The Tangle*, 2018. https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf, abgerufen: 2019-01-25.

[147] Cosmos Network, *Internet of Blockchains*. https://cosmos.network/, abgerufen: 2019-01-25.

[148] Polkadot, *Polkadot*, 2019. https://polkadot.network/, abgerufen: 2019-01-25.

[149] Quant Network, *Overledger*. https://www.quant.network/our-technology/overledger/, abgerufen: 2019-01-25.

[150] J. Poon und T. Dryja, *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*, 2016. https://lightning.network/lightning-network-paper.pdf, abgerufen: 2019-01-25.

[151] E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. I. Lvovsky und A. K. Fedorov, *Quantum-secured blockchain*, 2017. http://arxiv.org/abs/1705.09258.

[152] D. Rajan und M. Visser, *Quantum Blockchain using entanglement in time*, 2018. http://arxiv.org/abs/1804.05979.

[153] K. Ikeda, *qBitcoin*, 2017. http://arxiv.org/abs/1708.04955.

[154] J.-S. Weng, J. Weng, M. Li, Y. Zhang und W. Luo, *DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain-based Incentive*, 2018. https://eprint.iacr.org/2018/679.

[155] G. Popov, *SKYCHAIN – The future of artificial intelligence in healthcare!*. https://skychain.global/upload/iblock/89a/wp_english_Newest.pdf, abgerufen: 2019-01-25.

[156] International Organization for Standardization, *ISO/TC 307 Blockchain and distributed ledger technologies*. https://www.iso.org/committee/6266604.html, abgerufen: 2019-02-01.

[157] European Telecommunications Standards Institute (ETSI), *Terms of Reference (ToR) for ETSI ISG ‚Permissioned Distributed Ledger' (ISG PDL)*, 2018. https://portal.etsi.org/Portals/0/TBpages/PDL/Docs/ISG_PDL_ToR_DG_Approved_20181130.pdf, abgerufen: 2019-02-01.

[158] Institute of Electrical and Electronics Engineers, *Standards – IEEE Blockchain Initiative*, 2019. https://blockchain.ieee.org/standards, abgerufen: 2019-02-01.

[159] World Wide Web Consortium, *Blockchain – W3C Blog*, 2019. https://www.w3.org/blog/category/technology/blockchain/, abgerufen: 2019-02-01.

[160] Internet Research Task Force, *blockchain federation – IRTF wiki*, 2018. https://trac.ietf.org/trac/irtf/wiki/blockchain-federation, abgerufen: 2019-02-01.

[161] Europäische Kommission, *European countries join Blockchain Partnership*, 2018. https://ec.europa.eu/digital-single-market/en/news/european-countries-join-blockchain-partnership, abgerufen: 2019-02-01.

[162] O. Fußwinkel und C. Kreiterling, *Blockchain-Technologie – Gedanken zur Regulierung*, 2018. https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/BaFinPerspektiven/2018/bp_18-1_Beitrag_Fusswinkel.html, abgerufen: 2019-02-01.

[163] Bitcoinity.org, *Bitcoin network hashrate*, 2019. https://data.bitcoinity.org/bitcoin/hashrate/6m?c=m&g=15&t=a, abgerufen: 2019-01-10.

Impressum

Herausgeber:
Bundesamt für Sicherheit in der Informationstechnik (BSI)
53175 Bonn

Bezugsquelle:
Bundesamt für Sicherheit in der Informationstechnik (BSI)
Godesberger Allee 185–189
53175 Bonn
Telefon: +49 (0) 228 999582-0
E-Mail: blockchain@bsi.bund.de
Internet: www.bsi.bund.de

Stand:
März 2019

Texte, Redaktion und Konzept:
Dr. Christian Berghoff, Dr. Ute Gebhardt, Dr. Manfred Lochter, Dr. Sarah Maßberg,
Bundesamt für Sicherheit in der Informationstechnik (BSI)
mit Beiträgen von Julia Braam, Sandra Häberer, BSI
sowie von Thomas Häberlen, Rainer Oberweis, Dr. Tobias Stadler,
Der Bundesbeauftragte für den Datenschutz und die Informationsfreiheit (BfDI)

Gestaltung und Druck:
Appel & Klinger Druck und Medien GmbH, Schneckenlohe

Bildnachweise:
Titelbild: Getty Images, AF-studio
Abbildungen 1–13: BSI

www.bsi.bund.de