HOCHSCHULE MITTWEIDA
University of Applied Sciences

Wissenschaftliche Berichte | Scientific reports

Konferenzband zum Scientific Track der Blockchain Autumn School 2020

Nr. 1, 2020

Konferenzband zum Scientific Track der Blockchain Autumn School 2020
Impressum

Herausgeber:
Hochschule Mittweida
University of Applied Sciences
Der Rektor
Prof. Dr. phil. Ludwig Hilmer
Der Prorektor Forschung
Prof. Dr.-Ing. Uwe Mahn

Redaktion dieser Ausgabe:
Hochschule Mittweida |Referat Forschung
University of Applied Sciences

Leitung:
Prof. Dr.-Ing. Andreas Ittner
Dipl.-Volkswirt Mario Oettler

Kontakt:
Hochschule Mittweida
University of Applied Sciences
Referat Forschung
Postfach 1457
D-09644 Mittweida

Tel.: +49 (0) 3727 / 58-1264
Fax: +49 (0) 3727 / 58-21264
forschung@hs-mittweida.de
www.forschung.hs-mittweida.de

ISSN 1437-7624

Erscheinungsweise:
Unregelmäßig
Auflage:
Belegexemplare sowie bestellte Druckexemplare
Druck:
Hochschuldruckerei Hochschule Mittweida

Förderung:
Die Hochschule wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.

Titelseite: Foto: Hochschule Mittweida

Bildnachweise werden direkt am Foto bzw. im jeweiligen Artikel aufgeführt.
Im Scientific Report gelten grammatikalisch maskuline Personenbezeichnungen gleicher-maßen für Personen jeglichen Geschlechts.

Die Scientific Reports/Wissenschaftliche Berichte als Wissenschaftliche Zeitschrift der Hochschule Mittweida - University of Applied Sciences lösen die bisherigen Scientific Reports mit allen Volume I-III ab und erscheinen mit Nr.1, 1998 ab November 1998 in neuem Layout und in neuer Zählung.

Für den Inhalt der Beiträge sind die Autoren verantwortlich.

SCIENTIFIC REPORTS | WISSENSCHAFTLICHE BERICHTE
The main aspect of the Scientific Reports is to promote the discussion of modern developments in research and production and to stimulate the interdisciplinary cooperation by information about conferences, workshops, promotion of partnerships and statistical information on annual work of the Hochschule Mittweida (FH) University of Applied Sciences. This issue will be published sporadically. Contributors are requested to present results of current research, transfer activities in the field of technology and applied modern techniques to support the discussion among engineers, mathematicians, experts in material science and technology, business and economy and social work.

Die Scientific Reports der Hochschule Mittweida sind online verfügbar unter:
www.forschung.hs-mittweida.de/veroeffentlichungen/scientific-reports
Eine Veröffentlichung einzelner Beiträge erfolgt entsprechend der Open Access Strategie der Hochschule Mittweida auf dem Hochschulschriftenserver: https://monami.hs-mittweida.de

INHALTSVERZEICHNIS

WIRTSCHAFTLICHER WEITERBETRIEB VON ERNEUERBAREN ENERGIEERZEUGERN NACH AUSLAUF DES FÖRDERZEITRAUMS DES EEG MIT HILFE VON BITCOIN-MINING AM BEISPIEL EINER WINDENERGIEANLAGE

Ralf Hartig, Clemens Fröhlich
Ifem – Institut für Energiemanagement an der Hochschule Mittweida, Heinrich-Heine-Straße 25, D-09648 Mittweida

Diese Arbeit befasst sich mit dem Prozess des Minings von Bitcoin. Dabei soll erklärt werden, wie elektrische Energie genutzt wird, um neue Blöcke zur Blockchain hinzuzufügen und welche Renditen dabei zu erwarten sind. Gleichzeitig soll geklärt werden, ob das Mining von Bitcoin ein Geschäftsmodell ist, mit welchem Anlagen zur Erzeugung erneuerbarer Energie auch ohne Förderung durch das Erneuerbare-Energien-Gesetz (EEG) wirtschaftlich betrieben werden können. Es wird beschrieben, wie sich diverse Einflussgrößen auf die Wirtschaftlichkeit des Minings auswirken. Eine Auswahl an Mining-Hardware wird hinsichtlich ihrer zu erwartenden Erträge geprüft. Außerdem werden die Risiken dieses Geschäftsmodells näher betrachtet.

This paper examines the process of mining Bitcoin. It explains how electrical energy is used to add new blocks to the block chain and what returns can be expected. At the same time, it should be clarified whether the mining of Bitcoin is a business model with which plants for the production of renewable energy can be operated economically even without support from the German Renewable Energy Sources Act (EEG). It will be clarified how various factors affect the economic efficiency of mining. A selection of mining hardware is examined regarding its expected yields. In addition, the risks of this business model are examined in more detail.

1. Einleitung

Anlagen zur Erzeugung erneuerbarer Energien, welche nach dem ersten Erneuerbare-Energien-Gesetz (EEG) aus dem Jahr 2000[1] eine feste Einspeisevergütung erhalten, fallen nach dem Jahr 2020 aus der EEG-Förderung heraus, da diese auf 20 Jahre begrenzt ist[2]. Insbesondere Windenergieanlagen sind nach 20 Jahren Betrieb durchaus noch in der Lage, weitere zehn Jahre betrieben zu werden[3], jedoch ist eine Direktvermarktung der erzeugten Energie an der Strombörse nur bedingt wirtschaftlich sinnvoll. Die Nutzung der Energie zum Mining von Bitcoin dagegen könnte ein attraktives Geschäftsmodell darstellen.

Das Mining von Bitcoin ist ein Prozess, für welchen funktionsbedingt ein erheblicher Bedarf an elektrischer Energie notwendig ist. Bei einem durchschnittlichen Haushalts-Strompreis in Deutschland von ca. 30,4 Euro-Cent je Kilowattstunde (bezogen auf das Jahr 2019)[4] und einer zu erwartenden Rendite von 26 Euro-Cent je Kilowattstunde (eigene Berechnung), ist der Kostenaufwand höher als der tatsächliche finanzielle Nutzen einzuschätzen. Dennoch könnte ein wirtschaftliches Bitcoin-Mining möglich sein, wenn elektrische Energie kostengünstig produziert werden kann.
Als Beispiel für den Erzeuger elektrischer Energie dient eine Windenergieanlage, welche vom beschriebenen Ausstieg aus dem EEG betroffen ist. Anhand des Lastganges und des Kursverlaufs von Bitcoin im Jahr 2019 soll beispielhaft die theoretische Rendite für jenes Jahr ermittelt werden. Dabei wird zuerst erklärt, wie durch elektrische Energie Mining-Hardware betrieben wird, welche für das Mining von Bitcoin verantwortlich ist. Es werden die Einflussparameter für den Erfolg des Minings definiert sowie anhand der Kosten und zu erwartenden Renditen die optimale Hardwarekonfiguration gesucht.

2. Grundlagen

Die betrachtete Windenergieanlage befindet sich in der Nähe der Thüringischen Stadt Apolda. Sie besteht aus zwei Windkraftanlagen des Herstellers Enercon vom Typ E-40 (Nennleistung: 500 kW), sowie einer Windkraftanlage desselben Herstellers vom Typ E-66 (Nennleistung: 1.800 kW). Die Windenergieanlage kann also bei voller Auslastung insgesamt 2,8 MW elektrische Anschlussleistung bereitstellen.

Im Jahr 2019 lieferte die gesamte Windenergieanlage im Mittel eine elektrische Leistung von 535,92 kW. Somit wurden ca. 4.700.000 kWh (4,7 GWh) an elektrischer Energie in das Stromnetz eingespeist. Der Netto-Vergütungssatz betrug in jenem Jahr 9,5 ct/kWh für die Enercon E-40 (Marktwert + Marktprämie = 3,091 ct/kWh + 6,409 ct/kWh) und 10,1 ct/kWh für die Enercon E-66 (0,94 ct/kWh + 9,16 ct/kWh). Somit erwirtschaftete die Windenergieanlage im Jahr 2019 einen Gesamtertrag von ca. 465.000 €. Dabei handelt es sich, wie auch im Rest des Papers, um Nettoangaben.

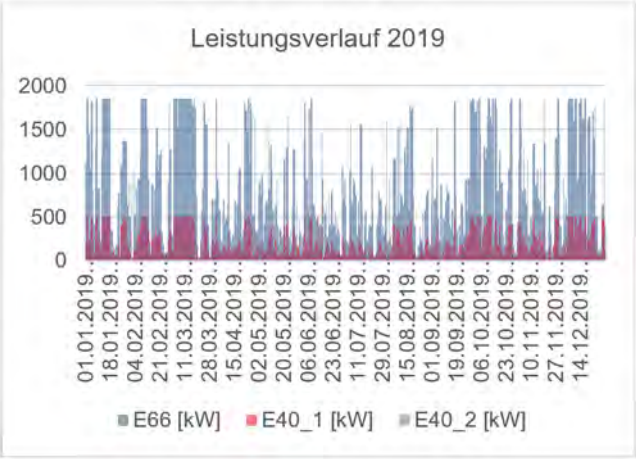


Bild 1: Verlauf der generierten elektrischen Leistung 2019. In den Sommermonaten wird insgesamt weniger elektrische Energie eingespeist als in den Wintermonaten. Im Mittel standen in jenem Jahr 536 kW elektrischer Leistung zur Verfügung.

Aufgrund des Wegfalles der Marktprämie ab 2021 wäre es nicht mehr möglich, die Windenergieanlage wirtschaftlich zu betreiben, da Betriebs- und Wartungskosten die Einnahmen übersteigen würden. Es ist also nötig, ein Geschäftsmodell zu finden, welches einen wirtschaftlichen Weiterbetrieb ermöglicht. Lösungsansätze dafür finden sich im direkten Stromverkauf ohne Zwischenhandel an den Endkunden (Community-Strom), Verkauf des Stromes an bestimmte Kunden über einen festen Zeitraum (Power Purchase Agreement, PPA) oder die direkte Nutzung des erzeugten Stromes vor Ort für Produkte oder Dienstleistungen. Letzteres beschreibt dabei beispielsweise die Produktion von Wasserstoff oder die Bereitstellung von Rechenleistung. Diese Rechenleistung könnte zum Beispiel zum Mining von Bitcoin verwendet werden.

Bitcoin-Mining ist als dezentrale Wertschöpfung zu interpretieren[5]. Dabei werden neue Blöcke erzeugt und anschließend zur Blockchain hinzugefügt. Die Blockchain lässt sich in diesem Sinne als dezentrale Datenbank verstehen, welche aus aneinandergereihten (verketteten) Blöcken besteht. Diese Blöcke beinhalten einen kryptographisch sicheren Hash des Vorgängerblocks, eine bestimmte Anzahl an Bitcoin (bis Mai 2020 12,5 und seitdem 6,25 BTC), sowie eine Bestätigung neuer bzw. noch offener Transaktionen. Ein Hash ist eine Zahlenfolge mit fester Länge, welcher nach einer bestimmten Hashfunktion, also einem Algorithmus, berechnet wird. Bitcoin nutzt den SHA256-Algorithmus (SHA = Secure Hash Algorithm). Im Bitcoin-Netzwerk wird ein Block mit einer Nonce versehen. Nonce, also „number used once“ ist eine Zahl, die nur einmalig innerhalb der Blockchain verwendet wird. Eine gültige Nonce ist jene, deren Hashwert mit einer bestimmten Anzahl von Null-Bits beginnt. Sie muss von der Mining-Hardware gefunden werden. Da dies mit einem erheblichen Rechenaufwand einhergeht, werden die Erzeuger der Blöcke mit den neu gewonnenen Bitcoin sowie den Gebühren der im Block enthaltenen Transaktionen belohnt.

Das Bitcoin-Netzwerk selbst reagiert auf eine steigende Rechenleistung. So wird die Schwierigkeit einen neuen Block zu finden erhöht, wenn die im Netzwerk vorhandene Rechenleistung steigt. Dies hat zur Folge, dass im Mittel täglich 144 Blöcke (ein Block alle 10 Minuten) gefunden werden, unabhängig davon, wie viele Rechner zu einem beliebigen Zeitpunkt ihre Kapazität bereitstellen.

Der Wert von Bitcoin bestimmt sich – im Gegensatz zu Reservewährungen – durch Angebot und Nachfrage. Im Jahr 2019 schwankte der Gegenwert der Bitcoin zwischen rund 3.000 € (Februar 2019) und 11.250 € (Juli 2019) um einen Mittelwert von ca. 6.600 €. Bereits jetzt sind ca. 18 Millionen Bitcoin gefunden wurden. Um eine Inflation zu verhindern, ist die Gesamtzahl an Bitcoin, die jemals verfügbar sein werden, auf 21 Millionen beschränkt[6]. Um das zeitnahe Gewinnen aller Bitcoin zu vermeiden findet in regelmäßigen Abständen ein Halving statt. Dabei wird die Belohnung je Block alle 210.000 gefundener Blöcke – also ca. alle vier Jahre – halbiert[7].

3. Nutzung elektrischer Energie zum Mining von Bitcoin

Das Mining von Blöcken erfordert eine Hardware, welche in der Lage ist, Hashes zu berechnen. Dafür wird diese Hardware mit elektrischer Energie versorgt und produziert eine gewisse Anzahl von Hashes pro Sekunde, die sogenannte Hashrate. Seit der Erfindung der Bitcoin stieg die Leistungsfähigkeit dieser Mining-Hardware stetig[8]. Inzwischen liefert nur noch dedizierte, d.h. anwendungsspezifische Hardware, wie Field Programmable Gate Arrays (FGPA) und Application-Specific Integrated Circuits (ASIC), die geforderte Rechenleistung. Diese bewegt sich derzeit im Bereich der Terahashes pro Sekunde [TH/s] je Miner. Für den Betreiber eines oder mehrerer Bitcoin-Miner ist es sinnvoll, sich einem Mining-Pool anzuschließen. Diese nutzen die Ressourcen aller Miner in einem Netzwerk und teilen dann die Belohnung für das Auffinden eines Blockes anteilig entsprechend des geleisteten Arbeitsaufwandes. Zwar werden dabei Poolgebühren (beispielsweise 2,5 % Pay Per Share bei AntPool) fällig, jedoch hat ein Pool aufgrund der höheren Gesamtrechenleistung eine höhere Chance, einen Block zu finden.

Generell lässt sich sagen, dass der Anteil an den gewonnenen Blöcken in etwa so hoch ist, wie der Anteil der lokal erzeugten Hashrate an der globalen Hashrate.

Befinden sich Bitcoin im eigenen Wallet, besteht die Möglichkeit, diese zu transferieren oder in eine Reservewährung (wie beispielsweise Euro oder Dollar) einzutauschen.

Der hier beschriebene Vorgang soll nun am konkreten Beispiel der Windenergieanlage simuliert werden. Als Miner wurde der Antminer S19 Pro der Firma Bitmain gewählt, da dies der derzeit energieeffizienteste Miner dieses Herstellers ist[9].

Der Bitmain Antminer S19 Pro generiert eine Hash-

rate von 110 TH/s. Dafür benötigt er eine Anschlussleistung von 3.250 W, also 3,25 kW[10]. Würde dieser Miner eine Stunde lang im Betrieb sein, würde er also eine elektrische Energie von 3,25 kWh umsetzen. Die Windenergieanlage hat eine Gesamtleistung von 2.800 kW. Damit wäre die Anlage theoretisch in der Lage, bei vollständiger Nutzung der Leistung bis zu 861 Miner gleichzeitig zu betreiben. Dabei ist anzumerken, dass für die Berechnung betreibbarer Miner stets abgerundet werden muss. Miner können nur mit Nennleistung und nicht mit einem Anteil derer betrieben werden.

Da Windkraft nicht kontinuierlich vorhanden ist und genutzt werden kann, sondern im Laufe des Jahres stark schwankt, verändert sich die durch die Windenergieanlage bereitgestellte Leistung. Damit ändert sich auch die Zahl der in einem Moment betreibbaren Miner stetig. Im Rahmen der Betrachtung für das Jahr 2019 wurden Tagesmittelwerte der elektrischen Leistung herangezogen. Es wurde für jeden Tag des Jahres untersucht, welchen arithmetischen Mittelwert die innerhalb eines Tages umgesetzte elektrische Leistung hatte. In folgendem Diagramm werden der Leistungsverlauf des Jahres 2019 sowie die dadurch mögliche Anzahl von Minern dargestellt:

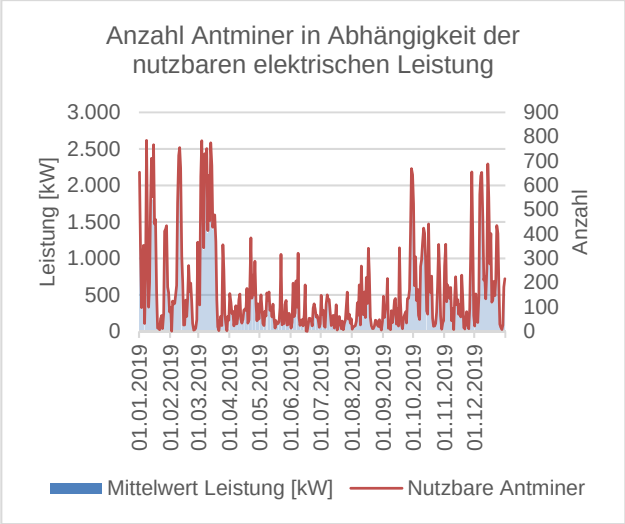


Bild 2: Die Anzahl der Miner, die theoretisch Nutzbare wäre, hängt direkt mit der zur Verfügung stellbaren Leistung zusammen.

Anhand der Anzahl der sich im Betrieb befindenden Miner lässt sich die lokal generierte Hashrate im Tagesmittel abschätzen. Diese ist wiederum in das Verhältnis zum Tagesmittel der globalen Hashrate zu setzen:

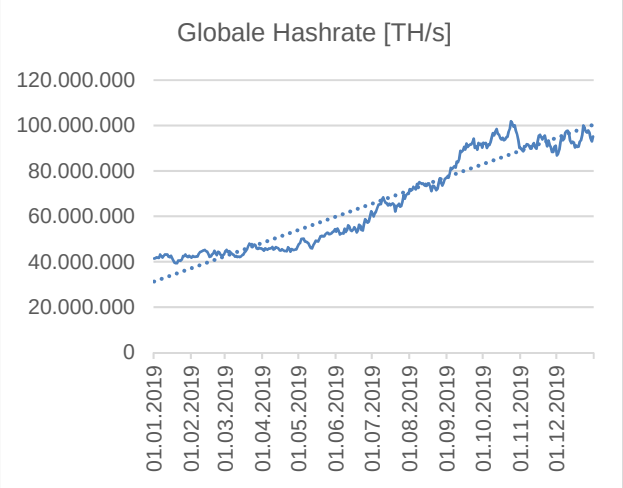


Bild 3: Die Entwicklung der Hashrate des gesamten Bitcoin-Netzwerkes im Jahr 2019[11]. Es lässt sich eine steigende Tendenz erkennen.

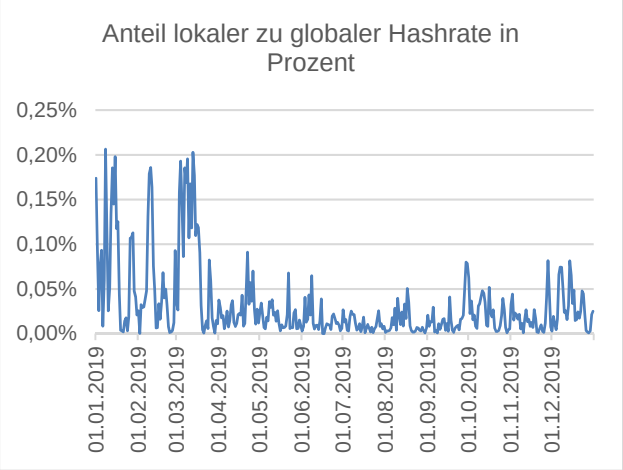


Bild 4: Besonders zu Beginn des Jahres wäre der Anteil der lokalen Hashrate an der globalen Hashrate mit bis zu 0,21 % noch sehr hoch gewesen. Dies beruht auf der zu dem Zeitpunkt noch eher geringen Hashrate im Netzwerk sowie auf der hohen Menge an nutzbarer elektrischer Energie.

Mit dem Wissen, dass am Tag im Schnitt 144 Bitcoin-Blöcke hinzugefügt werden, und dass ein Block im Jahr 2019 12,5 Bitcoin enthielt, lässt sich der lokale Tagesertrag errechnen. Dafür werden die täglich global generierten 1.800 Bitcoin mit dem Anteil von lokaler zu globaler Hashrate multipliziert. Es ergibt sich, dass im Jahr 2019 insgesamt knapp 209 BTC gewonnen worden wären.

Davon ausgehend, dass die gewonnenen Bitcoin am Ende jedes Tages verkauft, also in Euro umgewandelt werden, lässt sich ein Tagesertrag sowie eine Vergütung je Kilowattstunde, basierend auf einer Reservewährung (Euro, Dollar), berechnen. Entscheidend dabei ist der Stand des Kurses EUR-BTC zum jeweiligen Zeitpunkt[12]:

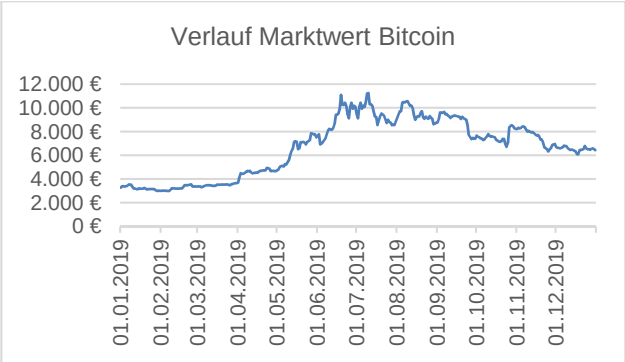


Bild 5: Im Mittel betrug der Gegenwert einer Bitcoin im Jahr 2019 6.646,09 €. Am 06.02. fiel die Bitcoin auf 2.979,51 € und erreichte am 10.07. ihren Höchstwert von 11.250,93 €.

In diesem Szenario wären im Jahr 2019 insgesamt 1,055 Millionen Euro an Vergütung für elektrische Energie erwirtschaftet worden. Das entspricht einem Mehrwert von 227 % gegenüber der tatsächlich erreichten Vergütung von 465.000 €. Der höchste Tagesertrag wäre am 08.01.2019 erreicht worden (13.071,74 €). Im Mittel lag der tatsächliche Tagesertrag bei 1.272,78 €, hätte aber durch das Mining von Bitcoin bis zu 2.891,37 € betragen können.

Je Kilowattstunde ergibt sich durch Bitcoin ein Ertrag von durchschnittlich 26,0 Cent im Vergleich zu 9,5 bzw. 10,1 Cent im bisherigen Vergütungsplan. Dabei schwankt der Ertrag durch Bitcoin von 11,9 ct/kWh am 02.02. bis 51,8 ct/kWh am 19.06.2019.

Anhand der Betrachtung lassen sich Einflussparameter identifizieren, welche den Ertrag beeinflussen:

So steigert eine höhere Anzahl der betriebenen Miner und der damit zur Verfügung gestellten lokalen Hashrate die Vergütung, genauso wie eine Verringerung der globalen Hashrate (z.B. nach einem Halving) und ein hoher Marktwert der Bitcoin.

4. Kostenbetrachtung

Mit einer Hashrate von 110 TH/s des S19 Pro ergibt sich eine Effizienz von 33,85 GH/s je Watt. Damit ist der S19 Pro der derzeitig effizienteste Miner des Herstellers Bitmain. Betrachtet man jedoch die Kostenausbeute, also die Hashrate, die man pro investierten Euro erhält, ist der S19 Pro bei einem Stückpreis von 2.021,88 € und einer damit verbundenen Kostenausbeute von 54,40 GH/s pro Euro derjenige Miner, welcher die geringste Ausbeute liefert. Am anderen Ende des Produktspektrums befindet sich der S9k mit einem Einkaufspreis von 52,92 €. Mit einer Hashrate von 13,50 TH/s liefert er die höchste Kostenausbeute von 255 GH/s pro Euro, jedoch auch die geringste Effizienz bei 11,76 GH/s je Watt. Die restlichen derzeit bei Bitmain erhältlichen Miner befinden sich dazwischen:

Miner	Hashrate [TH/s]	Leistung [W]	Stückpreis [€]	GH/s je €	GH/s je W
S19 Pro	110,00	3.250	2.021,88	54,4048	33,8462
S19	95,00	3.250	1.499,40	63,3587	29,2308
T17+	58,00	2.900	686,28	84,5136	20,0000
T19	84,00	3.150	1.469,16	57,1755	26,6667
S17e	60,00	2.700	671,16	89,3975	22,2222
S17+	70,00	2.800	1.034,88	67,6407	25,0000
S9 SE	16,00	1.280	79,80	200,5013	12,5000
S9k	13,50	1.148	52,92	255,1020	11,7596

Tabelle 1: Vergleich der aktuellen Miner-Modelle des Herstellers Bitmain hinsichtlich Effizienz und Kostenausbeute

Weiterhin lässt sich die Anzahl der verwendeten Miner hinsichtlich ihrer Auslastung optimieren. Die Windenergieanlage lieferte im Jahr 2019 eine mittlere elektrische Leistung von 535,92 kW. Das bedeutet also, dass mit einer größeren Anzahl Miner sich auch die Zahl derjenigen Miner erhöht, welche über ein Jahr hinweg sich weniger im Betriebs- als im Stand By-Zustand befinden.

Darüber hinaus ist die Verwendung von Mining-Containern zu empfehlen, da diese optimale Standortbedingungen für die Verwendung einer größeren Anzahl Miner ermöglichen (Kühlung, Monitoring, Betriebsspannung etc.). Außerdem ist zu beachten, dass bei einer reduzierten Anzahl eingesetzter Miner sich die Menge an nicht für Bitcoin-Mining genutzter elektrischer Energie erhöht.

Im Folgenden sollen die beiden vorgestellten Miner (S19 Pro und S9k) hinsichtlich ihrer optimalen Anzahl untersucht werden. Ziel dabei ist es, herauszufinden, ab wann die Miner beginnen, Erträge zu erzielen (Investition gegenüber Vergütung) und wie hoch der Ertrag je investierten Euro nach einem Jahr ist.

Für die Betrachtung wurden die maximale Anzahl (bezogen auf die Tagesmittelwerte der elektrischen Leistung), der arithmetische Mittelwert als Obergrenze und der Median herangezogen:

Miner-Modell	S19 Pro	S9k
Anzahl	785	2.223
Investition	1.587.175,80 €	117.641,16 €
Ertrag je Tag	2.891,37 €	1.006,95 €
Ertrag je Monat	87.945,84 €	30.628,16 €
Ertrag je Jahr	1.055.350,06 €	367.537,97 €
Erträge ab Tag	549	117
Ungenutzte Energie	14.235,00 kWh	4.939,32 kWh

Tabelle 2: Maximal mögliche Anzahl Miner. Obwohl der S9k eine geringere Investition voraussetzt und bereits nach knapp 4 Monaten erste Erträge generiert, erreicht er nicht den durch das EEG ermöglichten Gesamtertrag von 465.000 €.

Miner	S19 Pro	S9k
Anzahl	164	466
Investition	331.588,32 €	24.660,72 €
Ertrag je Tag	1.818,71 €	634,80 €
Ertrag je Monat	55.319,24 €	19.308,45 €
Ertrag je Jahr	663.830,92 €	231.701,43 €
Erträge ab Tag	183	39
Ungenutzte Energie	1,96 GWh	1,94 GWh

Tabelle 3: Beschränkung der Anzahl auf den arithmetischen Mittelwert. Mehr als 40 % der elektrischen Energie wird nicht zum Mining von Bitcoin verwendet. Der S9k generiert bereits nach 39 Tagen Erträge, die Vergütung beträgt ungefähr die Hälfte des durch das EEG ermöglichten Gesamtertrages.

Miner-Modell	S19 Pro	S9k
Anzahl	95	271
Investition	192.078,60 €	14.341,32 €
Ertrag je Tag	1.338,19 €	468,60 €
Ertrag je Monat	40.703,26 €	14.253,31 €
Ertrag je Jahr	488.439,18 €	171.039,73 €
Erträge ab Tag	144	31
Ungenutzte Energie	2,74 GWh	2,73 GWh

Tabelle 4: Beschränkung der Anzahl auf den Median. Der S9k generiert bereits nach einem Monat Erträge. Mit ca. 40 % der jährlich verfügbaren elektrischen Energie erzeugt der S9k 37 %, der S19 Pro 105 % des durch das EEG ermöglichten Gesamtertrages.

Der Ertrag je investierten Euro hinsichtlich der Auswahl der Miner sowie der gewählten Anzahl an Minern lässt sich durch das Verhältnis von investierter Summe zum Jahresertrag abschätzen:

Miner	Maximum	Mittelwert	Median
S19 Pro	0,6649	2,0020	2,5429
S19	0,7744	2,3315	2,9614
T17+	1,0331	3,1100	3,9453
T19	0,6988	2,1056	2,6733
S17e	1,0933	3,2875	4,1744
S17+	0,8273	2,4932	3,1562
S9 SE	2,4549	7,3831	9,3731
S9k	3,1242	9,3956	11,9264

Tabelle 5: Ertrag pro investierten Euro nach einem Jahr. Der S9k liefert bei Beschränkung auf den Median der maximal möglichen Anzahl den höchsten Ertrag gegenüber der Investition, liegt jedoch durch die geringe Effizienz trotzdem unter der Vergütung durch das EEG.

Die Betrachtung ist also wenig sinnvoll, da der S9k als Sieger hervorgeht. Der Ertrag pro Euro beim S19 Pro in maximaler Anzahl dagegen ist der geringste, obwohl diese Version langfristig den höchsten Ertrag abwerfen würde. Deshalb wird abschließend der mittlere Ertrag je Kilowattstunde betrachtet:

Miner	Ertrag [ct/kWh]
S19 Pro	26,01
S19	22,47
T17+	15,40
T19	20,50
S17e	17,11
S17+	19,25
S9 SE	9,63
S9k	9,06

Tabelle 5: Ertrag je Kilowattstunde. Mit dem S19 Pro sind die höchsten Erträge möglich, die Erträge durch den S9k liegen unterhalb der bisherigen durch das EEG ermöglichten Erträge.

5. Diskussion

Die hier dargestellten Erträge spiegeln nicht die real zu erwartenden Erträge wider. So müssen des Weiteren die Pool-Gebühren (abhängig vom gewählten Pool) sowie die Transaktionskosten beim Umwandeln von Bitcoin in Euro (abhängig von der gewählten Börse) beachtet werden. Außerdem sind die Kosten für die Anschaffung der Container, eventuelle Wartungs- und Reparaturkosten sowie die Kosten für die Schaffung der notwendigen Infrastruktur (Strom, Internet) nicht betrachtet.

Kryptowährungen wie Bitcoin weisen eine extreme Volatilität auf. Daher kann der Ertrag der Windenergieanlage von Tag zu Tag, sogar von Stunde zu Stunde stark variieren. Würde der Kurs der Bitcoin einen Jahresmittelwert von ca. 2.450 €/BTC unterschreiten, wären selbst mit dem S19 Pro keine höheren Jahreserträge als mit einer EEG-Förderung möglich.

Die steigende Tendenz der Hashrate des Bitcoin-Netzwerkes führt dazu, dass die Anteile an gewonnenen Blöcken immer geringer werden, sofern die eigene Hashrate nicht erhöht werden kann. Sollte der Wert der Bitcoin also zukünftig kein ausreichendes Wachstum mehr aufweisen, um dies auszugleichen, ist eine Wirtschaftlichkeit auch hier nicht mehr gegeben.

Ein Vorteil des Geschäftsmodells, Bitcoin zu minen, liegt in der Skalierbarkeit: spätestens, wenn sich ein Miner amortisiert hat und beginnt, Gewinne zu generieren, ist die Anschaffung eines weiteren Miners in Betracht zu ziehen. Dieser Vorteil lässt sich jedoch bei einer Windenergieanlage nur begrenzt ausnutzen, da die zur Verfügung stehende elektrische Leistung bzw. Energie beschränkt ist. Auch ist es nicht möglich, das Vorhandensein von Windkraft genau vorherzusagen. Daher lässt sich die Wirtschaftlichkeit eines weiteren Miners nur schwer kalkulieren. Schließlich ist auch die enorme Schwankung der Verfügbarkeit von Windkraft dafür verantwortlich, dass stets mit einer ungeplanten Downtime der Miner zu rechnen ist.

Weitere Bedrohungen sind ebenfalls zu berücksichtigen. So handelt es sich beim Mining von Bitcoin um einen unregulierten Markt, der in den kommenden

Jahren sicherlich mit neuen Gesetzen und Vorschriften konfrontiert wird. Es ist also in Erwägung zu ziehen, ob ein rechtlicher Beistand notwendig ist, welcher überprüft, ob sich sämtliche Tätigkeiten im Zusammenhang mit dem Mining von Bitcoin in einem rechtlichen Rahmen bewegen. Außerdem muss dieser Rahmen auch hinsichtlich der Wirtschaftlichkeit des Geschäftsmodells ständig beobachtet werden. Ein Verbot des Handels von Kryptowährungen beispielsweise würde nicht nur den Ertrag der Anlage tilgen. Auch die bis dahin getätigten Investitionen wären wertlos, da es sich bei der Mining-Hardware um eine hoch spezialisierte Hardware handelt, welche keine anderen Rechenaufgaben erfüllen kann.

Darüber hinaus muss je nach Aufbewahrung der Bitcoin für deren Sicherheit gesorgt werden, um Diebstahl oder Verlust zu verhindern. Es müssen also streng festgelegte Protokolle bzw. Verfahren umgesetzt werden, damit Probleme in diesem Zusammenhang auf ein Minimum reduziert werden können.

6. Zusammenfassung

Das Mining von Bitcoin ist ein interessantes und außergewöhnliches Geschäftsmodell, welches auch in den nächsten Jahren noch einige Entwicklungen durchlaufen wird. Das Paper hat gezeigt, dass zumindest für 2019 das Mining von Bitcoin mit nennenswerten wirtschaftlichen Erfolgen verbunden gewesen wäre. Zwar gibt es in dieser Branche eine Reihe von Risiken, doch ein qualifizierter Unternehmer, welcher diesen Markt versteht, kann auch in Zukunft beträchtliche Einnahmen erzielen.

Danksagung

Die Autoren bedanken sich für die Unterstützung durch Herrn Tobias Klein, Geschäftsführer der Firma TK-Solutions Verwaltungs GmbH, sowie beim BCCM – Blockchain Competence Center Mittweida.

Literaturverzeichnis

[1] Gesetz für den Vorrang Erneuerbarer Energien (Erneuerbare-Energien-Gesetz – EEG) sowie zur Änderung des Energiewirtschaftsgesetzes und des Mineralölsteuergesetzes vom 29. März 2000
Abgerufen von https://www.bgbl.de/xaver/bgbl/start.xav#__bgbl__%2F%2F*%5B%40attr_id%3D%27bgbl100s0305.pdf%27%5D__1525702271806

[2] §9 Abs.1 EEG 2000

[3] Schriften des Deutschen Instituts für Bautechnik (Oktober 2012), Richtlinie für Windenergieanlagen – Einwirkungen ung Standsicherheitsnachweise für Turm und Gründung
Abgerufen von https://web.archive.org/web/20160304102321/https://www.dibt.de/de/Fachbereiche/data/Aktuelles_Ref_I_1_Richtlinie_Windenergieanlagen_Okt_2012.pdf

[4] BMWI (Juli 2019), Durchschnittlicher Strompreis für einen Haushalt in Cent/kWh (Jahresverbrauch: 3.500 kWh)
Abgerufen von https://www.bmwi.de/Redaktion/DE/Downloads/I/Infografiken/durchschnittlicher-strompreis-haushalt.pdf?__blob=publicationFile&v=15

[5] Bitcoin Developer Guide, Mining
Abgerufen am 26.08.2020 von https://developer.bitcoin.org/devguide/mining.html

[6] ergibt sich aus dem Halving alle 210.000 Blöcke, siehe auch (abgerufen am 26.08.2020): https://bitcoin.stackexchange.com/questions/161/how-many-bitcoins-will-there-eventually-be/274#274

[7] Bitcoin Reference, Block Chain
Abgerufen am 26.08.2020 von https://developer.bitcoin.org/reference/block_chain.html

[8] Adrianne Jeffries (16. November 2012), Miner problem: big changes are coming for Bitcoin's working class
Abgerufen am 26.08.2020 von https://www.theverge.com/2012/11/16/3649784/bitcoin-mining-asics-block-reward-change

[9] SHA256/Bitcoin Miners
Abgerufen am 26.08.2020 von https://shop.bitmain.com/

[10] Antminer S19 Pro 110TH/s
Abgerufen am 26.08.2020 von https://shop.bitmain.com/product/detail?pid=00020200611225746542A8golxBc06C9

[11] Bitcoin.com Chart Hash Rate
Abgerufen am 26.08.2020 von https://charts.bitcoin.com/btc/chart/hash-rate#5mp0

[12] Bitcoin (BTC) and Euro (EUR) Year 2019 Exchange Rate History
Abgerufen am 26.08.2020 von https://freecurrencyrates.com/en/exchange-rate-history/BTC-EUR/2019/blockchain

BLOCKCHAIN TECHNOLOGY IN PROCUREMENT – A SYSTEMATIC LITERATURE MAPPING

Tan Gürpinar, Matthias Brüggenolte, Dennis Meyer, Philipp Asterios Ioannidis, Michael Henke
Technische Universität Dortmund, Leonhard-Euler-Straße 5, D-44227 Dortmund

Procurement processes are deemed to lack supporting digital technologies that raise efficiency and automation. Blockchain solutions are piloted in procurement in order to offer a decentralized IT infrastructure covering these needs. This paper aims at identifying current blockchain approaches in the field of procurement and presenting affected business processes. In order to get an overview of the current state of the art, a systematic literature mapping is conducted. Moreover, the out-comes are gathered and categorized in a classification scheme. Based on the analysis, systematic maps are presented to showcase relevant findings. Within the findings, several blockchain use cases in the field of procurement are identified and information about addressed challenges, utilized blockchain frameworks and affected business processes are extracted.

1. Introduction

In recent years, procurement has taken on great strategic importance due to globalization, shorter product life cycles, increased competition and a significant shift to buyer markets. Besides production and sales, procurement nowadays represents one of the three core functions of a company and forms the interface to the procurement market with its suppliers [1]. From a traditional point of view, procurement was mainly seen as an operational function that merely satisfies the demand for procurement goods and negotiates the lowest possible prices. In contrast to the sales function, the strategic and market-oriented tasks in procurement were considered to be of little importance. Hence, the current developments force companies to rethink and reorganize their traditional procurement tasks and functions. [2] In this process, two main driver can be identified, namely the outsourcing of services that were previously per-formed in-house and the increasing use of digital systems to influence the efficiency of procurement processes. [1][3]

As a result of the new role, procurement focuses on cross-company value creation and partnerships within the supply chain [2]. Through its interfaces to internal customers and external suppliers, procurement function has access to a large amount of data. The integration of this data in vertical and horizontal direction in the supply chain and in the own company offers a great potential e.g. for more transparency and more efficiency in the processes. However, the complexity of the tasks also in-creases and so do the requirements in regard to technologies in procurement. In sup-ply chain partnerships, mutual trust is an important prerequisite for successful cooperation, in addition to common interests, expectations and responsibilities [4]. Consequently, strategic areas such as supplier evaluation and management are becoming increasingly important. Here, the innovative ability of suppliers is a crucial aspect and focused by the realization of joint digital projects. [5]

The implementation of blockchain solutions and their utilization in procurement processes, are deemed to address several of the mentioned challenges [6]. Therefore, one characteristic of blockchain technology is the ability to establish trust between different parties. Hence, it is an obvious consideration to utilize this characteristic for a network of supply chain partners. By securing data in a unique way, parties that use a blockchain solution can come to consensus without any central authority involved. Furthermore, processes and transactions can be executed in a transparent manner and automatized by the utilization of smart contracts. [7]

In this paper, we want to present a systematic literature mapping and find out which blockchain use cases in procurement are existent in current literature and which business processes are addressed by them. The paper is structured as follows: In section 2, we introduce procurement as a function together with its tasks, goals and an over-view of relevant processes. Also in section 2, we define blockchain technology and describe its solutions for businesses. In section 3, we present the research methodology, in order to show the results and discuss them in section 4. In section 5, we summarize the findings, draw a final conclusion and identify future research directions.

2. Background

2.1 Procurement

The scientific literature offers a large number of different definitions when it comes to procurement [8]. Though, there is a consensus that there are important differences between its strategic activities and operational-administrative ones. In this paper, we define procurement as the management of external resources of an enterprise, with the aim of ensuring the best conditions for the availability of all goods, services, skills and knowledge [10]. In literature, the procurement process mostly starts with the step of determining requirements and extends from supplier management to ordering and processing [1][10]. Following Monczka et al. (2015) we can use the process description in Figure 1 for further analysis [13].

The procurement process can generally be divided into seven phases. In the following, the process steps are explained in more detail in order to create a uniform understanding for the subsequent literature analysis.

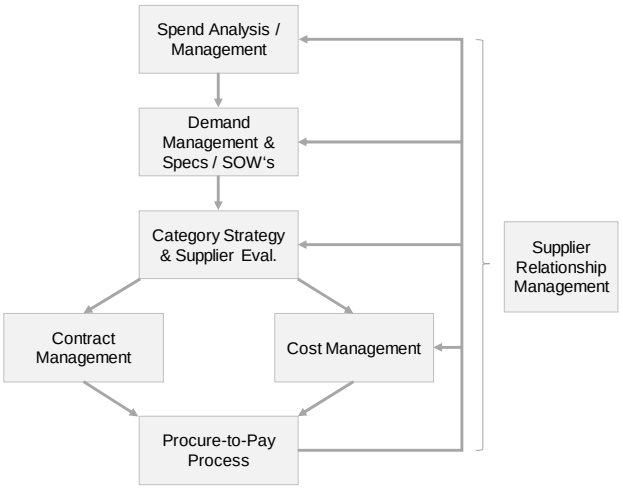


Figure 1: Procurement Process [13]

The spend analysis is concerned with the collection of historical data on goods and the demand from business sectors as well as corporate spending. The purpose of data collection is to provide a common understanding of past expenditures within an organization. The demand management on the other hand, focuses on forecasting the consumption of customers and monitoring and controlling the products that are procured. In this context, advice for optimizing consumption and information on alter-natives is provided. In category management, the aim is to gain an understanding of stakeholder requirements and to compare these with external industry information, capabilities of suppliers and potential risks. On this basis, a strategy is developed to reconcile internal requirements with the external conditions of the procurement market. Within contract management, procurement deals with supplier negotiations and the drafting of contracts. In this context, purchasing is often supported by other company departments, so that purchasing monitors the entire process, but often acts as a mediator between departments. Another important function of procurement is cost management, whereby the focus is on continuous cost improvement. Here, the physical products or services to be procured are considered over the entire product life cycle in order to continuously optimize the target costs. In the Procure-to-Pay (P2P) process, the focus is on the automation of transaction activities for the purchase of a good or service. This includes all activities from the release mechanism, order placement, orders, approval, receipt of products to the release of payment. Finally, supplier relationship management (SRM) is the end-to-end process for monitoring and evaluating suppliers throughout the procurement lifecycle. This includes all aspects and performances of suppliers, such as transactions, identification of risks and performance, opportunities for value enhancement and cost reduction. [13]

As blockchain technology is deemed to offer great potential for making the de-scribed processes in procurement more efficient or supporting in their automation, in the following we present more details on the technology basics and its application areas.

2.2 Blockchain Technology

The scientific literature offers various application areas for blockchain technology. While in many approaches blockchains solely appear in connection to cryptocurrencies, in this paper we focus on its use as an underlying technology with the purpose of optimizing different kinds of business processes. In this sense, a blockchain can be defined as a decentralized, distributed, tamper-proof and cooperatively used data storage [14]. In order to understand how this technology can be utilized as a data storage and how benefits over traditional technologies can be materialized, we have to understand the principles behind blockchain. As a matter of fact, blockchain technology is based on several core principles, namely cryptography, game- and graph theory, as well as peer to peer networks (see Figure 2, Level 1). These are utilized by the technology as follows:

Cryptography: The storage of ordered and grouped data records within a block-chain occurs in blocks that are linked by hash functions and secured by asymmetric encryption. Consensus Game Theory: Consensus algorithms are used to validate data and establish a single point of truth between several parties. Hence, it is an elementary component of decentralized approaches, since central authorities are eliminated. Graph Theory: Graphs are used to illustrate transaction histories and participants (in the form of nodes) of a blockchain network. P2P Networks: Peer-to-peer (P2P) networks are utilized for the decentralized distribution of information within a blockchain network. [15]

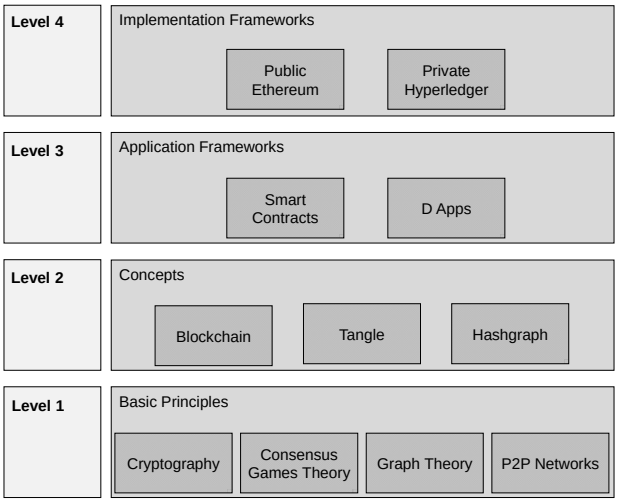


Figure 2: The Four Levels of Distributed Ledger Technologies [15]

As presented in Figure 2, blockchain technology is considered one concept (level 2) of several distributed ledger technologies, which also include directed acyclic graphs, like the Tangle or Hashgraph. Based on these concepts, decentralized applications and smart contracts can be implemented and realized on level 3. On level 4, a distinction is made between the network type (public, private) and network accessibility (restricted, unrestricted). [16]

The Blockchain types shown in Table 1 can be used

for further technical restrictions in this paper. While public blockchain solutions provide complete data transparency to the public and are used especially in connection with cryptocurrencies, private and consortium blockchain solutions present the respective counterpart and are increasingly used in business context. With these types of solutions, managed data cannot be observed by the public and access to the system is usually restricted. The assignment of reading-, writing- and administration rights is carried out either by a central instance (private blockchain) or by a consortium (consortium blockchain). [17]

According to these differentiations, also different kinds of consensus algorithms are utilized. In public blockchain solutions, we have common concepts like the Proof of Work or Proof of Stake. In blockchain solutions that are suitable for a limited number of consortium members, concepts such as the round-robin consensus algorithm, proof of authority, or practical byzantine fault tolerance are utilized. These methods, which are more common in business environments, are characterized as energy-efficient and resource-saving while achieving a higher performance than comparable concepts in the area of public solutions. [18]

Table 1: Types of Blockchain Solutions [19]

	Blockchain Types			
Network Type	Public		Private	
Access Permission	Public Permissionless	Public Permissioned	Consortium Permissioned	Private Permissioned
Read Permission	Public	Public	Authorized participants	Completely private or authorized participants
Write Permission	Anyone	Authorized participants	Authorized participants	Network operator only
Permission to Change the Setup	Anyone	Anyone or authorized participants	Anyone or authorized participants	Network operator only
Exemplary Framework	Bitcoin, Ethereum	Sovrin	Hyperledger Fabric	MulitChain

3. Research Methodology

As research methodology for this study, we follow the approach to perform a systematic literature mapping as described by Petersen et al. (2008), in order to identify approaches related to blockchain solutions in procurement [20]. Thereby, the goal of a systematic literature mapping is to present the area of interest through an overview of literature and quantify the amount of evidence. We also follow the guidelines described by Kitchenham & Charters (2007) while screening the literature for relevant papers [21]. Figure 3 illustrates the five process steps of the systematic literature mapping along with its corresponding outcomes. To be able to provide an overview of blockchain-based applications in procurement and to identify the quantity and quality of the available papers, we consider three research questions in the process of this study:

1. “What type of technical blockchain specifications are used in procurement?”

2. “Which blockchain use cases do we see in procurement and which branches are affected?”

3. “Which concrete procurement processes are addressed by blockchain solutions and what is their goal?”

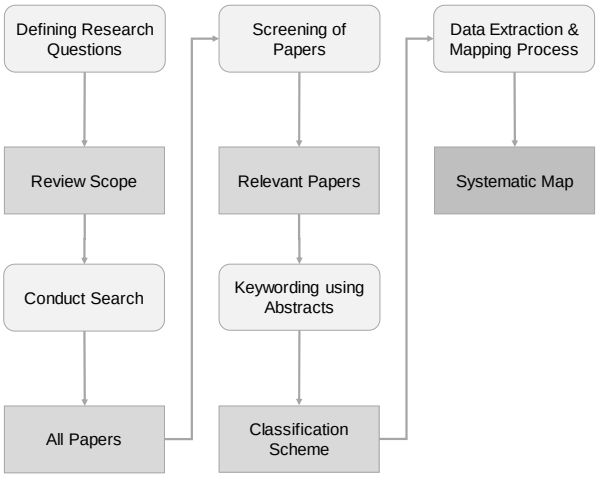


Figure 3: The Systematic Mapping Process [21]

To develop our search string, we then used the term of distributed ledger technologies and blockchain technologies, as well as their abbreviations to describe our focused technology, and used common synonyms of the procurement as our application area. As a result the following search string was used for our literature search via Scopus data base:

TITLE-ABS-KEY (((Blockchain OR BCT OR DLT OR "Distributed Ledger Technology") AND (procurement OR purchasing OR "supply management" OR sourcing OR buying OR psm)))

In order to guarantee a high quality of the considered papers, we included only peer-reviewed literature that was published between 2015 and 2020. Further on, we excluded literature that would be categorized as informal studies, as well as literature that showed an abstract divergent to our predefined topic. The total review numbers can be obtained from the table below.

Table 2: Results of the Analysis

Process step	Number of papers
Paper identified and screened	139
Remaining papers after reading the abstract	61
Remaining papers after full reading	38

The quality of the papers was assessed according to the quality guidelines of [21]. Therefore we utilized the following questions and categorized the papers as “Papers for further analysis”, “Papers that could be included additionally”, “Papers that don’t fulfill the quality criteria”.

• Does the paper clearly state their purpose and goals?

• Does the paper answers all of its predefined questions?

• Does the paper present its results in a well-structured way?

• Does the paper offer a clear conclusion that is based the prior results?

According to the quality guidelines, the obtained papers were reduced by the number of 78. The remaining papers were analyzed along our categories and by reading the full papers. During this processes another 23 papers were excluded as they did not provide enough information for further analysis. The categories for analysis of the remaining papers can be found in Table 3. [1] In the analysis procedure, the keywords (I4) and abstract (I5) served as an initial identifier to determine the type of the publications (I6) and the research category (I7). By reading the full text, all further categories were analyzed.

Table 3: Extracted Data Items

Item #	Data Item	Description
I1	Study Identifier	DOI or ISBN
I2	Author	Author of the paper
I3	Title	Title of the paper
I4	Keywords	Keywords given by the authors
I5	Abstract	Abstract of the paper
I6	Publication Type	Type of publication (e.g. conference/journal)
I7	Research Category	Type of research (e.g. review paper, method paper)
I8	Type of Blockchain	Public, private or consortium blockchain
I9	Framework	Blockchain framework (e.g. Ethereum)
I10	Consensus Algorithm	Consensus algorithm used
I11	Use Case	Description of presented use case
I12	Branch	Branch that was described by the authors
I13	Addressed Challenges	Procurement challenges that were addressed
I14	Procurement Processes	Procurement processes that were affected

4. Results and Discussion

In this section, we present and discuss the results of the systematic literature mapping and therefore the findings around blockchain solutions in procurement. We start with general information on the analyzed papers and go through technical information on the blockchain solutions, as well as identified branches, use cases, challenges and concrete procurement processes that are addressed.

4.1 Publication Type and Research Category

In Figure 4 we can see the publication type of the selected 38 papers. By publication type we mean the channel used for publication.

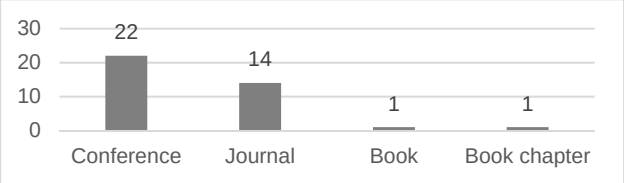


Figure 4: Publication Type

In addition to the publication type, we also consider the type of research that was performed in the papers. Most of the papers focus blockchain solutions in current enterprise pilot projects or blockchain solutions that are about to be piloted. These papers have a strong technical condition and open up new avenues for the technology. They are classified as technology and code articles and make up almost 40% of the results. A share of 29% presented and discussed concepts on a non-technical level and are classified as original research articles. 11% of the papers contribute view-points and opinions on the interpretation of recent findings in different research fields and are classified as opinion articles. Lastly, 8% are characterized as review articles and present a balanced perspective on current research activities.

4.2 Blockchain Specifications

The classification results with respect to blockchain specifications can be seen in Figure 6. We analyzed the papers according to their proposed type of blockchain, the specific framework and utilized consensus mechanism. In most of the papers, we identified a lack of blockchain specific technical information. Consequently, in all three categories we have a large share of papers not providing the respective information. Apart from that, we can see that 13 approaches introduce a private or consortium blockchain type to be utilized in procurement, while 4 approaches introduce public types. Hence, when it comes to specific blockchain frameworks, also a non-public framework, Hyperledger Fabric, is introduced with most of the hits (7). Ethereum, which can be utilized as both private or public framework, is mentioned 5 times, while other frameworks receive 3 hits. The Bitcoin blockchain is introduced in one approach. Likewise, the introduced consensus mechanisms show the proof of work concept in the case of the four public blockchain types. Apart from that, in two cases, solutions based on Ethereum are presented with a proof of stake consensus mechanism. One case is presented where the consensus mechanism of a private blockchain solution is described but not named.

[1] Here we also present the list of our analyzed papers: https://doi.org/10.5281/zenodo.4004784

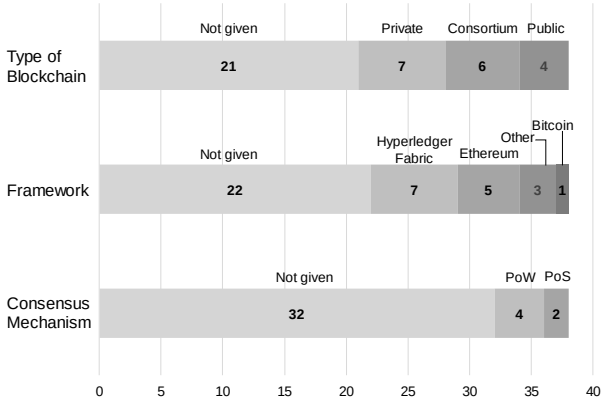


Figure 5: Classification according to Blockchain Specifications

4.3 Branches and Use Cases

The presented blockchain solutions are utilized in our identified papers for various branches. Each per branch, one to three different procurement use cases are mentioned. Exceptions are paper that present cross-sectoral approaches or reviews and therefore mention up to five different use cases.

According to our findings in agriculture and foods industry, two approaches introduce private blockchains to support and empower tracking and tracing solutions of products. In automotive, we have blockchain approaches focusing procurement, when it comes to collaborative data exchange especially with suppliers and the trading of assets. The first use case targets at making operational procurement processes more effective and efficient, while the second one opens up a new business field in introducing a blockchain based marketplace for self-maintaining machines. In the energy sector, all identified use cases are connected to platforms or mechanisms that enable peer to peer electricity trading in a sustainable manner. Both buying and selling processes are focused in this respect. In healthcare, approaches to secure products from counterfeiting and fraud are focused and supplemented by track and trace solutions. In our cross-sectoral category we have a wide distribution of different use cases. Most of them deal with the introduction of blockchain powered bidding systems or agile procurement processes that should become more efficient by block-chain solutions. In the category for other branches, we aggregated specific findings like approaches in military, real estate or public sector. Even though we have heterogeneous approaches here, most of them focus on the trading of various kinds of as-sets and introduce blockchain solutions to enable the trading in a decentralized and secure manner. Regardless whether the asset are represented by physical goods, virtual tokens, or online streaming services as examples. In Figure 7, we can see the distribution of branches and use cases.

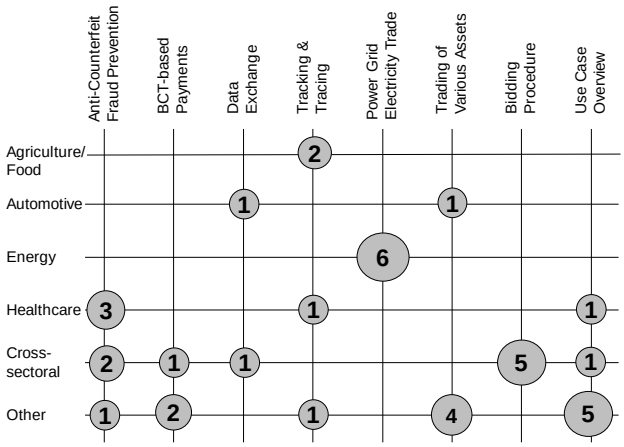


Figure 6: Use Cases and Branches

4.4 Addressed Challenges and Procurement Processes

Based on the described blockchain use cases, figure 8 shows which business challenges are addressed by the papers and in that respect, which procurement processes are affected. In our analyzed papers, we identified several challenges and differentiated between the needs of enhanced data exchange (53% of the papers), product provenance (34% of the papers), product- and data security (61% of the papers), or cost efficiency (45% of the papers). On the other side, we have our seven procurement processes that are involved in the respective projects and vary from more operational processes like procure-to-pay to more strategic processes like supplier relationship management.

According to our findings in papers aiming at an enhanced data exchange by implementing blockchain solutions, we can see that all procurement processes are mentioned at least once. Some papers present a holistic function of the blockchain solution affecting all process steps except for the spend analysis. Most of the papers give a strong focus on contract management and the procure-to-pay process. As an example, we have blockchain solutions described in the papers that build interfaces to the ERP systems of suppliers. In this context the sort and extend of data that is ex-changed needs to be contractually determined. Also in order to automatize these exchange processes along with financial transactions, smart contracts are utilized to constitute contractual information digitally. In case of papers focusing on product provenance, we can see that blockchain solutions are mentioned less in the field of operational procurement processes like procure-to-pay, or cost management and rather appear in context of strategic processes like supplier evaluation and relation-ship management. As an example, we have solutions described that target end to end traceability solutions for a whole supply chain. In this sense, not only direct partners, but also previously unknown partners, like raw material suppliers need to be involved. In this context blockchain solutions are proposed as neutral consortial solutions to engage with suppliers on an equal

footing. Papers dealing with product- and data security have a strong focus on both contract management and supplier evaluation. As an example, we have approaches targeting the detection of fraud in pharmaceutical procurement, or avoid corruption in public sectors. In the center of these approaches we have blockchain solutions that store data in an immutable and tamper-proof manner allowing parties also to use digital signatures for the consortium to identify owners of certain activities. In this sense procurement departments are involved in assessing their partners' activates or draft contracts on the basis of this data. The last category of papers targets an enhanced cost efficiency through block-chain solutions and focuses on the procurement processes of contract- and cost management. As an example, we have approaches that propose blockchain-based marketplaces, where machines receive own identities to autonomously order, pay and interact with other machines in a decentralized manner using smart contract. These approaches allow companies with a low depth of production, to depend flexibly and according to the conditions of the smart contracts, on numerous suppliers.

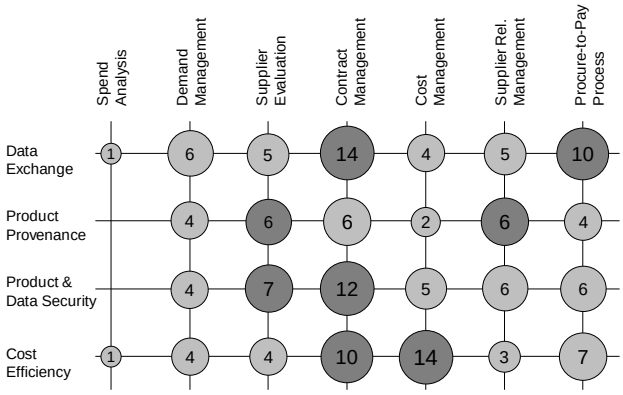


Figure 7: Addressed Challenges and Procurement Processes

5. Conclusion

In this paper, we conduct a systematic literature mapping on the application of blockchain technology in procurement processes. Our result is an overview of utilized blockchain solutions, branches that implement those solutions and a description of respective use cases. Also, we deliver the main goals and addressed challenges of the indicated approaches and present affected procurement processes. Our study has two major limitations: (i) Even though we had a relatively high share of technology articles in our scope, only a few papers reveal statements about a proposed blockchain framework or consensus algorithm. Hence, in this paper we don't present detailed information on technical blockchain specifications. (ii) In order to have a representative number of papers for the analysis, we included papers that elaborate on blockchain applications in procurement in a generic manner and either list all procurement processes to be affected or don't specify the process level at all.

As future research need, the aim should be to gather more detailed technical- and process related specifications in order to compare blockchain-based processes with traditional ones and derive a statement concerning their sense of purpose and profit-ability.

References

[1] Kummer, S., Grün, O., Jammernegg, W. (eds.): Grundzüge der Beschaffung, Produktion und Logistik, 3rd edn. Always learning. Pearson, München (2013)

[2] Piontek, J.: Beschaffungscontrolling, 5th edn. De Gruyter Studium. De Gruyter Olden-bourg, Berlin, Boston (2016)

[3] Wirtz, B.W.: Electronic Business, 5th edn. Springer Gabler, Wiesbaden (2016)

[4] Wannenwetsch, H.: Integrierte Materialwirtschaft, Logistik und Beschaffung, 5th edn. Springer-Lehrbuch. Springer Vieweg, Berlin (2014)

[5] Kleemann, F.C., Glas, A.H.: Einkauf 4.0. Springer Fachmedien Wiesbaden, Wiesbaden (2017)

[6] Nin Sánchez, S.: The Implementation of Decentralised Ledger Technologies for Public Procurement. European Procurement & Public Private Partnership Law Review (2019). https://doi.org/10.21552/epppl/2019/3/7

[7] Jakob, S., Schulte, A., Sparer, D., Koller, R., Henke, M.: Blockchain und Smart Contracts (2018)

[8] Lysons, K., Farrington, B.: Procurement and supply chain management. Purchasing and supply chain management. Always learning (2016)

[9] Schuh, G.: Einkaufsmanagement. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)

[10] van Weele, A.J., Eßig, M.: Strategische Beschaffung. Grundlagen, Planung und Umset-zung eines integrierten Supply Management. Springer Gabler, Wiesbaden (2017)

[11] Heß, G.: Strategischer Einkauf und Supply-Strategie. Springer Fachmedien Wiesbaden, Wiesbaden (2017)

[12] Werner, H.: Supply Chain Management. Springer Fachmedien Wiesbaden, Wiesbaden (2013)

[13] Monczka, R.M., Handfield, R.B., Giunipero, L.C., Giunipero, L.: Purchasing and Supply Chain Management, 6th edn. Cengage Learning, Boston, MA (2015)

[14] Knirsch, F., Unterweger, A., Engel, D.: Implementing a blockchain from scratch: why, how, and what we learned. EURASIP J. on Info. Security (2019). https://doi.org/10.1186/s13635-019-0085-3

[15] Schacht, S., Lanquillon, C.: Blockchain und maschinelles Lernen. Wie das maschinelle Lernen und die Distributed-Ledger-Technologie voneinander profitieren, 1st edn. (2019)

[16] Schacht, S., Lanquillon, C.: Blockchain und maschinelles Lernen. Wie das maschinelle Lernen und die Distributed-Ledger-Technologie voneinander profitieren, 1st edn. Springer Berlin; Springer Vieweg, Berlin (2020)

[17] Puthal, D., Malik, N., Mohanty, S.P., Kougianos,

E., Das, G.: Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems. IEEE Consumer Electron. Mag. (2018). https://doi.org/10.1109/mce.2018.2816299

[18] Pahlajani, S., Kshirsagar, A., Pachghare, V.: Survey on Private Blockchain Consensus Algorithms. In: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT). 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), CHENNAI, India, 25.04.2019 - 26.04.2019, pp. 1–6. IEEE (2019 - 2019). https://doi.org/10.1109/ICIICT1.2019.8741353

[19] Rutz, V.: Blockchain quo vadis. Springer Fachmedien Wiesbaden, Wiesbaden (2020)

[20] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic Mapping Studies in Software Engineering. Proceeding of the 12th International Conference on Evaluation and Assessment in Software Engineering (2008)

[21] Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. Technical report EBSE-2007-01. Keele University and University of Durham (2007)

TOKENIZATION AND THE SYMBIOSIS BETWEEN BLOCKCHAINS

Felix Hildebrandt
Slock.it GmbH, Markt 16, D-09648 Mittweida

The financial world of blockchains is mostly covered by Bitcoin, taking up about 210 billion dollars in market cap. Despite the huge security and independence which the technology offers to the users, it's not quite easy to adapt with upcoming applications due to the regulated infrastructure behind. For small-scale transactions, everyday use applications or the access to a variety of crypto technologies and projects, Bitcoin is relatively limited in future development. The compatibility for most of those applications is covering currencies from more development-driven blockchains like Ethereum. Those want to reach out for the user base that's already in hold of Bitcoins and offer them a seamless transition to new applications without the risk of losing their funds. Within the article, atomic swaps and tokenization are covered up and current approaches compared. Both mechanisms are used to fulfill this symbiosis between Bitcoin and Ethereum. To get a more practical view, an example on how to implement such a tokenization within an app is shown. This will give deeper insights and offers inspiration for digital identity-based app development.

1. Symbiosis between Bitcoin and Ethereum

Bitcoin was the first application realized with blockchain technology in 2009. It is an open-source, peer-to-peer technology to transfer digital money between participants. One of the key features is the static cap of the amount of currency, which means Bitcoin can't stock up its cash as it is known from fiat money. The gathering of the money itself is done by decentralized servers called nodes which solve cryptographic hash puzzles in order to get paid for created blocks. Those nodes are not only able to create, but can also spread and verify other blocks from the network. Everyone is able to participate in Bitcoin and create an address to store and transfer money. All without any central institution or customer verification. It was the first huge step for crypto assets, leading to its current market cap of 210 billion USD. [1] This is a huge opportunity for developers to create external software around it.

Where the Bitcoin blockchain is more based around a decentralized financial system, the Ethereum blockchain is not a specified chain in a certain field of usage. The project was released in 2015. However, it does not have a static hard cap like Bitcoin and in addition, everyone is able to create their own token or currency on the Ethereum chain. The chain can not only be used to execute transactions with these coins but also offers a decentralized virtual machine that can run code in order to create decentralized applications. It is valuable for developers and lead to the current market cap of around 43 billion dollars- making it the second largest cryptocurrency. [1]

Given the great potential of decentralized applications from Ethereum using tokenizing technologies, the use cases for Bitcoin and other blockchains could expand dramatically. It could resolve in much more user-friendly applications to interact with cryptocurrency. The fundamental need to tokenize or trade Bitcoin on other chains relies on the fact that Bitcoin is still the number one used digital currency for transactions. On the other side, Ethereum is the biggest blockchain for developing decentralized applications, where people want to make use out of all crypto assets. [2][3]

2. Atomic Swaps and Tokenization

Usually there is a risk of the associated counterparty when traders wish to exchange the coins between each other. Atomic swaps can solve this problem. An atomic swap can be described as a technology that introduces the trading of one cryptocurrency for another without using any centralized intermediaries. They can happen directly between blockchains [4][5]

Assuming Alice and Bob want to exchange coins. The atomic swap can be declared in three sequences in order to compliance with the security: The first sequence acts like a preparation of the transactions without any on-chain event. None of the parties needs to get any refund because they still own everything. Alice just picks a secret random number and creates a transaction which will send her coins to Bob's address if the secret number is known and signed by Bob. Not only that, Alice also creates a second, signed refund-transaction which is locked 48h in the future and will send her coins from the first transaction back to her own public address. Bob also creates both transactions on his end, the only difference being that the refund-transaction is locked 24h in the future- so Alice has enough time in order to check if the transfer was executed. Both will now send the refund-transaction to each other, sign and send them back so everyone has a signed refund transaction.

In the next sequence both will transmit their transactions on chain. In order to get the coins back, they can both publish the refund transaction which was signed from each other before. Alice submits her transaction to the network first, but Bob still can't get a hold on the coins, because the secret number isn't published. Bob then also submits his transaction to the network.

The last sequence can be called spending-phase. Both parties need to make sure they finally transmit their coins to their own addresses- otherwise the counterparty can claim their refund after a certain amount of time is over. Alice now signs and spends the transaction which was released to the network before. She cannot do this without revealing the secret number. With it, Bob can now sign and spend his transaction using the revealed secret number from

Alice. Because this process needs technological understanding, Hash Time-Lock Contracts have been developed. HTLC are time-bound Smart Contracts between parties that automate the process of atomic swaps for blockchains which support Smart Contract functionality. [4][6]

Commonly, the process for exchanging crypto-currencies is very time consuming if it's done without a middleman regarding waiting times. But even with HTLC´s or exchanges, there are several other inconveniences. For instance, not all cryptocurrency exchanges support all coins, a trader has to assign to multiple accounts or trade another crypto asset in between. Where atomic swaps can solve the exchange of currencies to use them on other blockchains, tokenization aims to convert assets so the initial value can be used on other chains.

Tokens in general refer to an asset which is created or handed out for an original value. This could be anything from rights or money, even real estate. Within the crypto space, where every blockchain has its own native currency, tokenization is mostly described as the process of an initial currency becoming an equal asset with more functionality. It is still important to know that tokenization itself does not guarantee the ability to get back the initial asset. The main problem of converting assets bound to other blockchains is that there either needs to be a verified and trustworthy middleman for centralized solutions or a technology within decentralized approaches which ensures that the initial currency or asset is backed with the same amount, making it 1:1 in scale. There are a lot of different implementations on the market covering both types- even when a fully decentralized version is always the best at the cost of more complexity.

But what is the regular order when it comes to tokenization? To be sure that the currency is really handed out for the tokens and no fraud has happened, the initial assets will be securely locked. After a verification time, new tokens for the locked assets are created. For example, if the currency from a financial based blockchain like Bitcoin is locked, it can be used afterwards to make its token accessible on a development driven chain like Ethereum. A tokenization of the original Bitcoins is initiated, and equal tokens are handed out. Now they can be used in decentralized applications or payment channels with increased liquidity. After the use of the application, the tokens can then mostly be traded in for the initial Bitcoins. Within this process, the tokens are burnt to prohibit double spending and the original currency remains. [7][8]

As a result of explaining both techniques, it would be great to compare them with each other. Atomic Swaps require price discovery by whoever starts the trading. Further, existing wallets and decentralized exchanges need to accept the atomic swap mechanisms. Tokens on the other side have the luxury to mostly be available in any ERC20 supported wallet, which is a common standard nowadays. Price discovery also doesn't need to be done for the user, because the value of the asset remains the same.

Compared in timing, Atomic Swaps are really slow. Even If there is a KYC process and no autonomous technique given during tokenization, it will still complete way faster. Further, when doing an atomic swap on a regular decentralized exchange platform, it will require a separate deposit and an atomic swap fee. This is another inconvenience of multiple exchanges.

The real benefit for atomic swaps is maintaining the initial currency and that nothing has to be locked up during the process. It is also quite handy for any single person that doesn't want to get anything else involved than the two persons. It's just not for frequent use and will also not give access back to the initial asset which tokenizing is known for. The use cases couldn't be much more apart. [5][8][9]

3. Why Tokenization Matters

Tokenization is bringing liquidity and application support. When tokenizing Bitcoins, the liquidity on decentralized exchanges will grow through Smart Contracts and bring impact on the huge decentralized financial market of Ethereum.

Also, tokens backed by fiat currencies offer a safe way for traders to keep their money within the crypto world. Because they are pegged to the real-world, price fluctuations in between won't happen. This offers a way to exchange fiat currency values in decentralized exchanges applications where no direct fiat currency can be used. Conversion rates or taxes can be saved, opening the world for digital currencies without dispense common money.

Finally, there are a lot of different projects on approaching exchanges on a decentralized fundament. Tokenizing technologies would make it easy to represent any other cryptocurrency across those projects and enhance it with new technology that offers token-support. Institutions which accept cryptocurrencies could only focus and develop on one chain, rather than multiple. [10]

4. Models of Implementation

There are two main types of implementing a tokenizing technology: either algorithmic or centralized. Within the algorithmic approach demand and supply are controlled by Smart Contracts or formulas. For example, Dai or Basis. If it is centralized, assets are stored and handed out by an organization which publishes proof of reserves. This is the case with Tether, True USD, USDC or future governmental bonds.

As for now, most tokenizing technologies are leaning on the centralized model, but instead of relying entirely on one institution, they rely on a consortium of institutions performing different roles in the network. Some approaches are even outsourcing the fee-calculation to central models. Those can be viewed as hybrid version with chosen governance on certain aspects of the technology. [10]

5. Ethereum as a Base Layout

In the following sections Bitcoin tokenization on Ethereum is viewed. But what are the actual benefits? First off, it has an increased transaction speed. Blocks are created around every 15 seconds and it is possible to have confidence in the irrevocability of a transaction in less than 5 minutes. This is faster than transacting natively on the Bitcoin blockchain. Secondly and already widely explained: Ethereum can offer much more usability. Because the ERC20 standard has been adopted by a large number of institutions and provides users with a variety of exchanges, wallets, and decentralized apps to use. It will also resolve in a greater liquidity for Bitcoin, even if the original asset barely moves on the Bitcoin chain. With it comes a greater exchangeability to other tokens, an increased transaction bandwidth and more privacy by the use of token-based technologies in general.

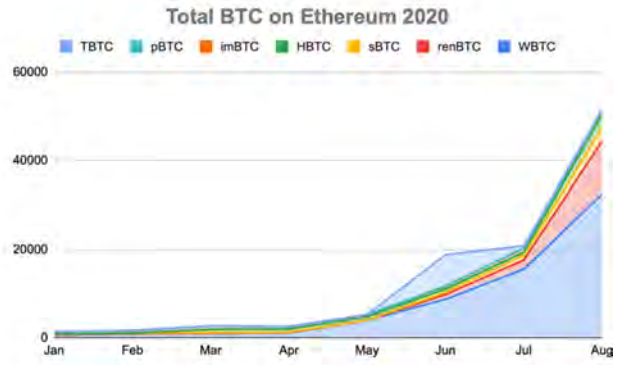


Fig. 1: Growth of tokenized Bitcoins on Ethereum 2020

Within this year, there was a huge increase in tokenized Bitcoin on Ethereum. Mainly in June and July where some of the first fully decentralized attempts were released on the main network of Ethereum. As for now, there are about 55,000 Bitcoins locked- rising nearly exponentially in the last 4 months. In the next sections, those projects will be presented. [11]

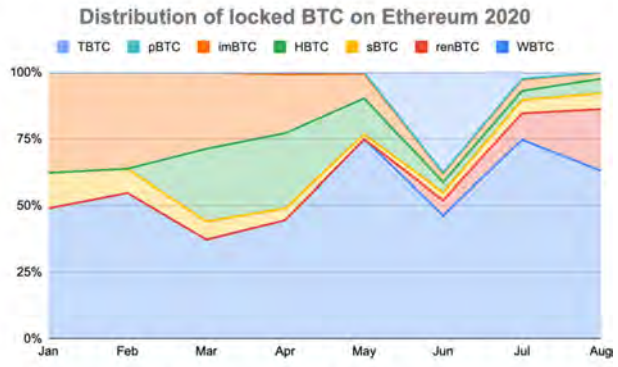


Fig. 2: Distribution of tokenized Bitcoins on Ethereum 2020

6. Keep Network: TBTC

The goal of tBTC from Keep Network is the creation of an ERC-20 token that maintains the most important property of Bitcoin- its status as hard money. But at the same time offering to use it fully decentralized on Ethereum applications. At the current point, only fixed amounts of tBTC can be traded: 0.01, 0.1, 0.2, 0.5 and 1 tBTC. Even with lot-sizes, the big advantage of this system is that anyone can convert currency without having to go through any KYC process.

The backing Keep Network already implements a token and a random beacon for signer selection, a distributed key generation protocol as well as an efficient multi-party threshold protocol. The only link between the Bitcoin blockchain and the host chain is the tBTC system itself, which runs as a set of Smart Contracts on the host chain. To acquire tokenized Bitcoin with tBTC, the user requests the creation via the Ethereum Smart Contract requiring a small amount of Ether. The TBTC network is then creating a signing group. Afterwards, multiple signers are chosen by a requested random seed from the own beacon. Then, group keys as well as a public key are created from elliptic curves through distributed key generation. The public key from the signing group is published to the host blockchain and corresponds to the Bitcoin wallet owned by the signing group. When the user requests the Bitcoin wallet address from the singing group, the wallet address is created by converting the public key. In the end, the user will deposit Bitcoins into the address, the signing group will prove the transaction block of the deposit through SPV and is assigning the user a non-fungible token. Now, the user sends the non-fungible deposit-token to the Keep Network which mints and assigns the tBTC.

Due to the mechanism being totally decentralized, there are a lot of security approaches. First off, each signer in the keep network has to deposit an amount of currency from the host chain to prevent the signers within the network from stealing Bitcoins. The percentage of how much the signers need to deposit is calculated by the number of signers per signing group. The Technology could handle up to 80 signers per group in future versions. In the first version of the TBTC Network the signer groups will consist of three signers. Because of the low number, they will need 3 out of 3 to guarantee transactions. In the beginning the collateral will be 150% from the current Bitcoin price in Ether. The signer groups will also change after 6 months. In case of a fraud, the collateral of Ether is used to buy back Bitcoins and they signers will be automatically removed from the keep network. Whats important: tBTC is not a stable synthetic coin bound to the Bitcoin price. If the collateral will be significantly lower or higher than Bitcoin, a liquidation process can be initiated. Within this process, signers close their deposit and pull out their Ether to create a new collateral with the current ratio. There is also a hard abort trigger for the developers to freeze the system for 10 days. After this option is pulled once, they can't freeze it another time. The developers can also change lot sizes, collateral threshold or delay fee rates. [12][13][14]

Because it is totally decentralized and everyone can participate, what will be the incentive to participate as a signer? First off, the keep network will choose 60 secure signers. After this, there will be six months

where anyone can apply as a signer and participate in the structure. Signers will be paid for their deposits. Per Bitcoin they deposited in Ether, they will receive 0.009375 tBTC. As each deposit has a fixed term of six months, that implies a total signing revenue of 1.875% each year. Signers also have to choose, if they only want to participate in signing groups or even in the network talking to the random beacon. If they want to participate in both, not only Ether but also KEEP as an utility-token is needed.

At the moment the solution is still in the early stage of the main network release on Ethereum. Due to a bug that was found quickly after the initial release date, the project shut down after only two days. Currently, they are on its way for a second version. [12][13][14] [15][16]

7. Provable: pBTC

The project named pTokens aims to solve liquidity and interoperability between blockchains by providing ERC-20 token versions of even non-ethereum blockchain currencies. The project is currently implementing pEOS and pBTC as ERC-20 tokens on the main network of Ethereum but also got a network for testing purposes to experiment as a developer. The way pTokens peg to the original asset is by running each involved blockchain simultaneously in a Trusted Execution Enclave. The TEE is a physical piece of hardware. Both full nodes from the blockchains of the two coins need to be involved, as well as secure enclaves running inside the TEEs and a network of validators that cooperate to jointly generate and manage the private keys for the peg-in/out process. The Enclave has access to both sets of keys and can execute transactions on both blockchains effectively linking the two assets together.

The enclave represents the secure sandbox container in which private keys for both corresponding blockchains can be generated and stored. These are then used for the transaction-signing that both mines and burns pTokens but also validate incoming blocks and their transactions. This ensures that only valid transactions from both blockchains are verified from the enclave. Decentralization will be achieved in a later stage by spreading the operation out to a federation of operators with multiple TEEs each. Currently it is still centralized. [17][18][19]

The user can make a deposit of the original asset using the pTokens deposit Smart Contract, providing their desired destination token address in the transaction. The block in which the preceding transaction takes place gets sent to the enclave along with all of its transactions. The enclave then validates the block header along with all the transactions. If the transaction is validated, the enclaves locate the pTokens transaction sent to the Smart Contract and parse out the amount and the destination address. With this data, the enclaves prepare a transaction to mint the equal number of tokens on the token-side Smart Contract. Enclaves now perform a multi-party computation to sign the transaction together with the derived private key for the address where the initial asset is held. Enclaves emit the transaction which is then broadcasted to the destination blockchain. If the transaction is mined, minted tokens will now be held by the destination address of the user. The functionality of gaining access back to the original asset is the same in reverse, the only difference being the burn function is called in the Smart Contract. At the moment, there are only about 50 Bitcoin locked into the system. That's mostly due to an early centralized version on the mainnet only using one validator. In the future, this will be replaced by DAO-like governance and a decentralized network consisting of TEE´s. [12][19]

8. TokenIon: imBTC

The idea behind imBTC is commonly known as an ERC-20 token backed 1:1 with Bitcoin from imToken which lets you manage multiple crypto assets in one wallet. imBTC can be generated by locking up Bitcoin using the imToken wallet from the company Tokenlon. Locking up Bitcoin sends BTC to a multi-signature account and simultaneously mints an equal amount of imBTC tokens. These tokens can then be used on Ethereum DeFi apps and later reimbursed again for bitcoins. While not trustless, the locking and unlocking process is fully automatic. An interesting feature of imBTC is that it bears interest by simply holding it. This interest comes from fees incurred by other users transforming and awards you about 1% annually. The locked Bitcoins are stored on a cold storage address. Users can redeem their BTC anytime or trade imBTC for other crypto assets supported by the imToken wallet. Currently, more than 1,100 Bitcoin are tokenized with the imToken wallet. [1][12][20]

9. Huobi: HBTC

Huobi is one of the world's leading crypto exchanges more common in asian regions. With HBTC they try to implement tokenized Bitcoin to the ethereum decentralized financial system. It can be viewed as a 3-step-scheme, where Huobi acts as a centralized custodian on top. From there, chosen acceptors can deposit Bitcoin and mint HBTC in return. Users can then trade Bitcoins to the acceptors to get HBTC tokens. This mechanism offers the ability to scale really well, because users only get in contact with the acceptors. But in the end, users need to trust Huobi as a centralized institution to get their Bitcoin back. Currently, there are about 2,800 Bitcoin tokenized with Huobi. [1][21][22][23]

10. Synthetix: sBTC & iBTC

Synthetix Bitcoins and Synthetix Inverse Bitcoins are synthetic assets so-called synths built on the Synthetix platform atop of Ethereum. Synths are considered to be more trustless than e.g. WBTC because they do not require the underlying asset to be held when it comes to using Bitcoins values on Ethereum. However, this makes Synthetix the only

approach without real Bitcoin-tokenization where initial currency is locked. It only appears as a trading platform for betting on assets, rather than holding them and making use with an owned initial asset on another chain. They are implemented as ERC-20 tokens and pegged against any crypto, real-world asset or other value. Synths are backed by the Synthetix Network Token, short SNX, which is staked at a ratio of 800% thus providing enough collateral to absorb large price shocks. Assets on Synthetix are assigned to an exchange rate through price feeds supplied by an oracle and can be exchanged on the Synthetix Exchange App.

There are trading pairs for SBTC against other synths such as SETH, SUSD, SEUR and even precious metal pegged tokens such as SXAU (gold) and SXAG (silver). Synthetix also allows the creation of synths that are inversely correlated to the asset they are tracking. IBTC, for example, tracks the inverse Bitcoin price and can be used to take a short position in Bitcoin by simply buying into it. Currently there are nearly 3,000 Bitcoins locked in the Synthetix ecosystem. [1][12][24][25]

11. Ren Network: renBTC

The Ren Network technology is very similar to the principle of the Keep Network. Ren is a platform with the goal of making tokens of different blockchains interoperable, allowing decentralized exchanges and decentralized financial apps to leverage the liquidity available on various blockchains. But it still has some minor changes. While tBTC uses Ether to guarantee a collateral of signers, renBTC is using their own REN-tokens to back the system. That means that the system is regulated by itself regarding demand and fees. The whole signer group will need to put down 3 times more collateral than the Bitcoin deposit which is held. That means, no Ether will be needed to be locked out of the Ethereum blockchain and the project isn't influencing the Ethereum blockchain by staking up huge collateral in Ether.

Ren also is able to use up to 90-200 singers per group which is way more than TBTC is able to achieve and offers a more frequent swap of group members. The fundamental aspect of Ren is a virtual machine which runs on a decentralized network of so-called dark nodes. RenVM allows the generation of a special address on the Bitcoin blockchain. When BTC is transferred to this address, RenVM takes custody of the coins and mints a representation of it on the host blockchain. At the moment, Ren has an application called Roundabout, to get tokenized Bitcoin quickly to Ethereum. However, Ren is the most successful solution that is fully decentralized. Only three months after release they already got about 1,200 nodes running and currently 12,000 Bitcoins locked. [1][12][26][27][28][29][30]

12. Kyber, Ren, BitGo: WBTC

Wrapped Bitcoin was initiated by a community formed out of more than 30 institutions e.g. Kyber, Ren, and BitGo. Wrapped Bitcoin was one of the first ERC20 token backed 1:1 with Bitcoin in 2019. Due to the prices of their tokens, which are reflecting the price of the asset backing them, they can also be called stable coins. WBTC posts proof of reserves on the Bitcoin chain. However, the technology is not as much decentralized, because there is a consortium out of validated custodians. It strives to promote usability, but acts like a federated governance model. WBTC is semi-permissioned, meaning there are AML and KYC processes involved, but approved merchants are incentivized to quickly initiate the minting of more tokens to users. This is similar to how Tether´s USDT has been able to massively scale with a permissioned minting and burning mechanism. The custodianship is secured by multi-sig contracts that require multiple parties to sign transactions. WBTC is currently the most significant player in the decentralized financial space with listings on platforms such as Compound, Nuo and Fulcrum. At the moment there are about 32,000 Bitcoins tokenized with WBTC. [1][12][31][32]

13. Evaluation of projects

The most important points when looking at the seven different approaches are trust and scalability. When looking at trustworthiness, there are simple factors that give a brief rating. Every project can be disassembled into a backing type, governance, custody and the price feed.

Looking at the backing type, all solutions excluding Synthetix rely on IOU, meaning that real Bitcoins are locked to gain usability of personally owned Bitcoins on Ethereum. Viewing the governance of the remaining solutions, only Ren is fully decentralized from the beginning. Keep and pTokens plan to move from federated- to decentralized in the future, but all others probably remain as trusted federation or centralized. Even by the fact that centralized institutions seem like safe custody and have no price feeds because of it, the type contradicts the principles of blockchains in general. Looking at price feeds for the two main decentral versions out there, Ren is using calculated formulas which adjust with minting- and burning-fees where Keep went with relying on the MakerDAO consortium for transaction fees.

If we look further to scaling, each project can be rated for the type of permission, peg-in/-out speed including their costs, liquidity appeal and the scalability mechanism. WBTC is the only solution which does not offer permissionless use and has a slow pegging speed compared to the others. The permissionless approach and fast verification times will be needed for most future apps to run in the autonomous backend of their machines. To be fair, the use case from WBTC is not willing to become a autonamous standard. Looking at the scalability and costs, centralized solutions can shine again, because Keep, Ren and Synthetix are limited in scale to the underlying asset meaning Ether or their own token.

Covering up the liquidy appeal for all those solutions, there is a trend for Ren, but Keep and pTokens aswell, if they can offer a stable, fully decentralized release to be picked up by developers as great tools.

All of them offer relatively low costs and great speed. However, Ren currently is the only solution out now which is able to fulfill all goals. Within three months after their release, they catched up to more then a third of WBTC holdings and may overtake them as the first fully decentralized project running tokenized Bitcoins on Ethereum. imTokens and HBTC will probably remain as long as they are used in the wallets of their big exchange companies with less potential to grow for the mass. Leaving Synthetix for crypto stock exchange traders as an user group and WBTC as the first tokenizing project and current leader on Ethereum, where you are able to lock and redeem with good conscience due to the involved KYC and AML processes.

14. Using Tokenizing in Web3

As mentioned, Ren is pushing ahead when it comes to a permissionless solution that is running on Ethereums main network. It is utilizing the web3 package of JavaScript, which lets you interact with the Ethereum blockchain and tries to be the groundwork for the development of the next generation of the internet. They also offer great assistance to make the entry with tokenizing as easy as possible.

Fig. 3: Keep Networks GatewayJS API Frontend

On their site, Ren guides you through compiling their Smart Contract with Solidity and developing a JavaScript application. Therefore, they offer two API's: GatewayJS for the seamless transition and implementation of Ren including pre-built frontend. But also RenJS which is a lightway implementation that offers more customization with its core functionalities. At the moment, not everything is fully open source, so testing is only possible within the Ethereum test network Kovan. But tied to their release plan, they will be fully open source until the end of this year. Where both solutions are mainly built for decentralized applications within the browser, there is a possibility to get web3 running on native smartphone devices. The aim in further research on developing with tokenization will be a fully working digital identity app including web3 and the ability to tokenize Bitcoin on smartphones. Therefore the Incubed client from Slock.it will be used to create one of the first super-efficient mobile digital identity applications including decentralized tokenization. [33]

Acknowledgements

Special thanks to Slock.it GmbH which supports my research, practical work, and bachelor thesis as well as giving me the opportunity to build meaningful and up to date software which will be used in future development.

References

[1] CoinMarketCap. (n. d.). Cryptocurrency Market Capitalizations. Retrieved August 31, 2020, from https://coinmarketcap.com

[2] Antonopoulos, A. M., Wood, G., & Klicman, P. (2019). Ethereum - Grundlagen und Programmierung. Weinheim, Germany: Beltz Verlag. Pa-ges 6, 9, 127 ff., 221 f.

[3] Ethereum whitepaper - whitepaper.io. (n. d.). Retrieved August 31, 2020, from https://whitepaper.io/document/5/ethereum-whitepaper

[4] What Are Atomic Swaps? (n.d.). Retrieved August 31, 2020, from https://www.investopedia.com/terms/a/atomic-swaps.asp

[5] Ks, K. (2018, November 12). Atomic Swaps BitcoinWiki. Retrieved August 31, from https://en.bitcoinwiki.org/wiki/Atomic_Swap

[6] Atomic swap - Bitcoin Wiki. (n.d.). Retrieved August 31, 2020, from https://en.bitcoin.it/wiki/Atomic_swap

[7] C. (2019, June 13). What is Tokenization? Everything You Should Know – CoreLedger. Retrieved August 31, 2020, from https://medium.com/coreledger/what-is-tokenization-everything-you-should-know-1b2403a50f0e

[8] O'Neal, S. (2019, June 2). Tokenization, Explained. Retrieved August 31, 2020, from https://cointelegraph.com/explained/tokenization-explained

[9] Token Swaps and Atomic Swaps Explained. (2018, July 25). Retrieved August 31, 2020, from https://blockwolf.com/token-swaps-and-atomic-swaps-explained/

[10] Wrapped Tokens Whitepaper. (n. d.). Retrieved August 31, 2020, from https://wbtc.network/assets/wrapped-tokens-whitepaper.pdf

[11] BTC on Ethereum. (n.d.). Retrieved August 31, 2020, from https://btconethereum.com

[12] Russo, C. (2020, April 20). It Was Almost Impossible to Keep Up With all the Bitcoin-on-Ethereum Efforts —Until Now. Retrieved August 31, 2020, from https://thedefiant.substack.com/p/it-was-almost-impossible-to-keep-26b

[13] Bitcoin on Ethereum. (n.d.). Retrieved August 31, 2020, from https://tbtc.network

[14] tBTC: A Decentralized Redeemable BTC-backed ERC-20 Tokens. (n. d.). Retrieved August 31, 2020, from https://docs.keep.network/tbtc/index.pdf

[15] Unchained Podcast. (2020, April 28). tBTC: What Happens When the Most Liquid Crypto Asset Hits DeFi? - Ep.169. Retrieved August 31, 2020, from https://www.youtube.com/watch?v=A17BdRDGbHc&feature=youtu.be
[16] ETHDenver, & Reckhow, C. (2020, March 19). BUIDL Bitcoin on Ethereum with tBTC: A Simple, Trust-Minimized Peg – Carolyn Reckhow. Retrieved August 31, 2020, from https://www.youtube.com/watch?v=KWGssyEIPa8&feature=youtu.be
[17] Things, P. (2020, April 23). pTokens launch on mainnet! - Provable. Retrieved August 31, 2020, from https://medium.com/provable/ptokens-launch-on-mainnet- 8c0a0cffa24f
[18] pTokens DApp. (n.d.). Retrieved June 19, 2020, from https://dapp.ptokens.io/
[19] Provable pTokens. (n. d.). Retrieved August 31, 2020, from https://ptokens.io/ptokens-rev5b.pdf
[20] Tokenlon - An easy-to-use cryptocurrency DEX. (n.d.). Retrieved August 31, 2020, from https://tokenlon.im/
[21] HBTC (n. d.). Huobi Bitcoin. Retrieved August 31, 2020, from https://www.hbtc.finance/static/pdf/whitepaper-en.pdf
[22] Official Launch Of Huobi BTC (HBTC) On Ethereum Network. (n.d.). Retrieved August 31, 2020, from https://huobiglobal.zendesk.com/hc/en-us/articles/900000196603-Official-Launch-Of-Huobi-BTC-HBTC-On-Ethereum-Network
[23] Huobi Center. (n.d.). Retrieved August 31, 2020, from https://www.huobi.fm/
[24] Synthetix | Decentralised synthetic assets. (n.d.). Retrieved August 31, 2020, from https://www.synthetix.io/litepaper/
[25] Synthetix. (n.d.). Retrieved August 31, 2020, from https://www.synthetix.io
[26] L. (2019, December 5). Welcome to the RenVM Developer Center - Ren Project. Retrieved August 31, 2020, from https://medium.com/renproject/welcome-to-the-renvm-developer-center-c1ade842fe07
[27] L. (2020, January 17). December Development Update - Ren Project. Retrieved August 31, 2020, from https://medium.com/renproject/december-development-update-e910df747d38
[28] Roundabout Exchange. (n. d.). Retrieved August 31, 2020, from https://roundabout.exchange
[29] Ren. (n. d.). Retrieved August 31, 2020, from https://renproject.io
[30] Ren Litepaper. (n. d.). Retrieved June 19, 2020, from https://renproject.io/litepaper.pdf
[31] WBTC Wrapped Bitcoin an ERC20 token backed 1:1 with Bitcoin. (n.d.). Retrieved August 31, 2020, from https://wbtc.network/
[32] Wrapped Tokens. (n. d.). Retrieved August 31, 2020, from https://www.wbtc.network/assets/wrapped-tokens-whitepaper.pdf
[33] Welcome. (n.d.). Retrieved August 31, 2020, from https://docs.renproject.io/developers/

LINUX-DISTRIBUTION ZUR SICHEREN ERSTELLUNG VON COLD STORAGE WALLETS

Lucas Johns
Hochschule Mittweida, Technikumplatz 17, D-09648 Mittweida

Dieses Paper beschreibt die Implementierung eines Live-Systems für die Erstellung von Cold Storage Wallets. Ziel soll es sein, einen sicheren und einfachen Erstellungsprozess von Paper-Wallets unter hohen Sicherheitsansprüchen zu ermöglichen. Der Quellcode ist abrufbar unter *https://github.com/envake/vinktar-live*.

This paper describes the implementation of a live distribution for the creation of cold storage wallets. The aim of this implementation is to simplify the process of creating paper wallets under high security requirements. The source code is available under *https://github.com/envake/vinktar-live*.

1. Einleitung

Um Kryptowährungen sicher aufzubewahren, hat sich das Speichern der Private Keys in sogenannten Cold Storage Wallets bewährt. Das sind Wallets, die zu keinem Zeitpunkt mit dem Internet verbunden sind. Die Private Keys werden dabei entweder auf spezieller Hardware gespeichert oder können alternativ auch ausgedruckt werden. Letzteres wird als Paper-Wallet bezeichnet. Da die Anschaffung eines Hardware-Wallets in der Regel mit einem Kostenaufwand verbunden ist, werden häufig einfach Paper-Wallets zur längeren Lagerung von Kryptowährungen eingesetzt. Um diese Paper-Wallets zu erstellen, muss ein gültiges Schlüsselpaar, bestehend aus Private- und Public Key generiert werden. Für alle gängigen Kryptowährungen stehen dafür längst Werkzeuge zur Verfügung, meistens in JavaScript implementiert. Unzählige Artikel und Tutorials erklären, wie beispielsweise ein Bitcoin Paper-Wallet erstellt werden kann. Immer wieder gibt es hier aber auch Angebote von Betrügern, die die Schlüsselerzeugung manipuliert haben. Sicherer ist dagegen, die Schlüsselgenerierung offline durchzuführen. Letztlich hat sich dafür die Generierung in einem schmalen Linux-System bewährt, welches offline und einmalig gestartet wird. Hier kommen meist Distributionen, wie Tails oder Ubuntu Live-USB zum Einsatz. Diese sind aber in keiner Weise auf eine so spezielle Aufgabe optimiert. Es ist davon auszugehen, dass ein Großteil der Endbenutzer eher auf die zahlreich verfügbaren Online-Angebote zur Aufbewahrung zurückgreift. Im Umkehrschluss kann die Offline-Speicherung attraktiver werden, wenn der Einrichtungsprozess einfacher ist. Daraus erschließt sich die Motivation dieses Projekts. Ziel soll es sein, eine Lösung zu entwickeln, die einen sicheren und einfachen Erstellungsprozess von Paper-Wallets ermöglicht. Verfolgt ein unerfahrener Anwender den grundsätzlich richtigen Gedanken, sein Guthaben offline zu speichern, soll ihm ein entsprechend optimiertes Werkzeug an die Hand gegeben werden können.

2. Risikofaktoren und Angriffsmöglichkeiten

Zunächst besteht das Risiko, keine seriöse Software zu verwenden. Vor allem unerfahrene Anwender sind davon betroffen. Bei Softwareprodukten, die etwas mit Kryptowährungen zu tun haben, taucht immer wieder auch Spyware auf. So beispielsweise auch bei Wallet-Anwendungen für das Smartphone [1]. Außerdem werden Phishing-Seiten verwendet, um Nutzer dazu zu bringen, sich eine Adresse dort generieren zu lassen. Die Betreiber der Phishing-Seiten haben verschiedene Methoden, um es so aussehen zu lassen, als seien die Private Keys tatsächlich zufällig generiert. Oft verfahren die Betrüger so, dass sie erst eine gewisse Zeit abwarten, bis genug Nutzer Opfer des Phishing-Angriffs wurden, um dann das Guthaben aller Wallets auf eigene Adressen zu transferieren [2]. Für so einen Angriff ist es noch nicht einmal erforderlich, dass der Rechner des Anwenders mit dem Internet verbunden ist. Es reicht aus, dass ein manipulierter Zufallsgenerator im verwendeten Programm zum Einsatz kommt. Wenn dieser keine echte Entropie liefert, sondern vorbestimmte Werte, kann die Anzahl der möglichen Adressen stark begrenzt werden. Der Angreifer kann die Erzeugung dann reproduzieren und alle privaten Schlüssel berechnen. Wenn ein Programm beispielsweise nur etwa 10000 verschiedene Adressen generiert, ist das entsprechend einfach möglich. Als Referenz hierfür dient ein Fall, durch den Betrüger etwa vier Millionen US-Dollar stehlen konnten, indem sie ein manipuliertes Tool zur Generierung von IOTA-Seeds anboten [2]. Obwohl die JavaScript-Anwendung auf GitHub veröffentlicht wurde, blieb die Manipulation unentdeckt.

Ein weiteres Risiko geht von Spyware im allgemeineren Sinne aus. Speziell Rechner, die bereits mit Malware infiziert sind und für die Generierung von Kryptoadressen verwendet werden, stellen ein sehr hohes Risiko dar. Die Chancen sind dann recht hoch, dass dies ebenfalls zu einer kompromittierten Adresse führt. Moderne Spyware, die permanent den Hauptspeicher des Opfers durchsucht, verfügt häufig auch über einen Payload zur Ermittlung möglicher Kryptoadressen. Ein Payload ist der Teil der Schadsoftware, der den tatsächlichen Schaden anrichtet. Im Falle der Spyware, stellt die Beschaffung der Informationen den Schaden dar. Ein Beispiel dafür ist Clipboard-Hijacker-Malware, die den Zwischenspeicher überwacht und bei einer entdeckten Bitcoin-Adresse,

diese mit einer eigenen austauscht [3]. Überprüft das Opfer eine eingefügte Adresse nicht noch einmal, gehen die Transaktionen auf ein Wallet des Betrügers.

Die erstellten Paper-Wallets sollen in der Regel ausgedruckt werden und hier entsteht mit der Benutzung des Druckers ein weiteres Risiko. Die Drucker können für sensitive Daten ein enormes Risiko darstellen, da die heutigen Multifunktionsdrucker meist alle über einen Netzwerk-Stack verfügen. In dem Artikel *"System & Application Security"* der Information Security and Policy von Berkeley wird über diese Problematik wie folgt berichtet.

"*Multifunction printers (MFPs) are experiencing an identity crisis: IT administrators don't always see them as the full-fledged networked computers they really are. But attackers do - and they are finding them increasingly very attractive!*" [4]

Das verwendete Programm zur Schlüssel-generierung kann dann noch so sicher sein und ordnungsgemäß funktionieren, wenn der ganze Prozess in einer unsicheren Umgebung ausgeführt wird. So, wie es in der IT-Sicherheit häufig formuliert wird, kann sich der Angreifer das Ziel aussuchen. Um an die Private Keys zu kommen, gibt es für Kriminelle demnach viele Wege. Die Umsetzung einer Lösung, die all diese Angriffsvektoren berücksichtigen soll, stellt daher keine triviale Aufgabe dar. Das Risiko, welches von klassischer Spyware ausgeht, kann mit einem Live-System sehr effizient minimiert werden. Von unsicheren Druckern hingegen geht auch in einem isolierten Betriebssystem weiterhin eine Gefahr aus.

3. Aufbau des Systems

Ziele der VINKTAR Live-Distribution:

- minimaler Aufbau
- offline, no root
- open source, auditierbar
- kryptografisch sicheres RNG-Konzept
- intuitives frontend
- hohe Druckerkompatibilität

Für die Erstellung der Distribution wird das Debian Live-Build Framework verwendet [5]. Das Debian Project hat bereits 2010 angekündigt, den Debian Kernel komplett ohne proprietäre Firmware zu entwickeln [6]. Die Basis der Distribution bildet daher ein 32-Bit Debian GNU/Linux der Release-Schiene "testing", da das "stable" Release in der Regel sehr alte Versionen von Anwendungen bezieht. Die Konfiguration der Distribution erfolgt bei Live-Build durch eine umfangreiche Verzeichnisstruktur.

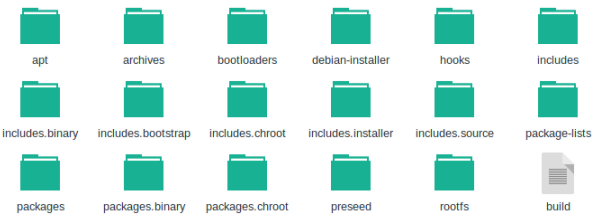


Bild 1: Konfigurationsverzeichnis von live-build, Bildquelle: L. Johns

Aus dieser Struktur kann das komplette System gebaut werden. Als Bezugsquelle der Softwarepakete sind die Debian-Repositorys voreingestellt. Diese werden auch immer für den Debian Kernel genutzt. Debian verwendet das Advanced Packaging Tool (APT) zur Paketverwaltung. Es können aber beliebige APT-Repositorys verwendet werden. In diesem Fall ist das nicht notwendig, da für das Grundsystem nur Debian Pakete verwendet werden sollen und die zusätzlichen Schlüsselgenerierungstools manuell über die live-build-Includes in das System integriert werden. Die Werkzeuge können in der isolierten Umgebung sowieso nicht aktualisiert werden. Nur einmal beim Erstellen des Abbilds werden alle Pakete von ihrer Bezugsquelle geladen. Verwendete Pakete des Live-Systems werden bei live-build in sogenannten Package-Lists organisiert. Das sind einfache Textdateien, die im config-Verzeichnis gespeichert werden. Sie beinhalten alle Bezeichnungen der Pakete, die aus den konfigurierten Repositorys von APT installiert werden sollen. Für dieses Projekt werden vier solcher Listen erstellt. Die Listen sind nach den entsprechenden Bereichen benannt. Diese Trennung ist erforderlich, um größere Änderungen am System zu erleichtern. Soll beispielsweise eine andere Desktopumgebung verwendet werden, ändern sich damit auch viele andere Pakete, die Bestandteil des Desktops sind. Dann muss einfach die *desktop.list.chroot* ausgetauscht werden. Das System beinhaltet einen Browser, Officeprogramme und printer-realted packages, die eine hohe Kompatibilität mit gängigen Druckermodellen ermöglichen sollen.

Package list	Beschreibung	Beispiel
apps.list.chroot	allgemeine Anwendungen	Firefox ESR
desktop.list.chroot	xfce4-spezifischen Pakete	xfce4-terminal
fonts.list.chroot	notwendige Schriftarten	fonts-liberation
live.list.chroot	für den live-Betrieb notwendige Pakete	user-setup
printing.list.chroot	zum Drucken erforderliche Pakete	system-config-printer
rng.list.chroot	Pakete für random number generator	rng-tools

Tabelle 1: Struktur der Paketlisten mit Beispielen

Weiterhin können benutzerdefinierte Skripte zu bestimmten Zeitpunkten ausgeführt werden (hook-scripts). Durch sogenannte *includes* werden eigene Dateien in die Live-Distribution integriert. Dadurch können Konfigurationsdateien für die im System enthaltene Software verfügbar gemacht werden. Auch die Tools zur Schlüsselgenerierung sind manuell in das Dateisystem integriert.

Eine essenzielle Softwarekomponente der Distribution stellt der Browser dar. Viele Werkzeuge zur Schlüsselerzeugung sind in JavaScript implementiert und werden im Browser ausgeführt. Speziell die kryptografischen Funktionen der Browser-API werden im Hinblick auf die spätere Verwendung im System wichtig. Für dieses Projekt wird Mozilla Firefox ESR verwendet. Dieser ist vollständig quell-offen und im Debian main repository verfügbar. Generell wird für das Live-System Wert auf ein minimalistisches Design gelegt. Sowohl die enthaltene Software als auch die Bedienung sind auf den schmalen Anwendungsfall fokussiert. Weitere Merkmale sind das Init-System systemd, der Bootloader SYSLINUX und der verwendete Desktop Xfce4. Die Xfce4-Oberfläche wird von Grund auf neu aufgebaut. Im Menü 'Paper Wallets' wird eine kategorische Zuordnung verwendet. Neue Anwendungen, die in das System integriert werden sollen, müssen so nur der Kategorie *PaperWallets* angehören und werden automatisch angezeigt. Weiterhin wurde für das allgemeine Erscheinungsbild noch ein Theme integriert mit entsprechenden Icons. Die Distribution soll sich möglichst intuitiv bedienen lassen, weshalb nur Usecase-relevante UI-Elemente vorhanden sind. Beispielsweise muss das System niemals in den Ruhezustand versetzt werden. Der Benutzer wird sich auch niemals manuell abmelden müssen. Die Konfigurationen dafür werden in XML-Dateien gespeichert. Diese befinden sich im Home-Verzeichnis des Benutzers unter *config/xfce4/xfconf/xfce-perchannel-xml/*, wobei z.B. die modulare Struktur der Panels in der Datei *xfce4-panel.xml* beschrieben wird [7]. Eine Besonderheit des Systems ist, dass es standardmäßig mit deaktiviertem Root-Benutzerkonto startet. Der Linux-Kernel kann mit der Option *noroot* gestartet werden, um die Verwendung als Root zu verhindern. Der Gedanke ist hier, dass ein Root-Benutzerkonto, welches nicht verwendet werden kann, auch keinen Schaden anrichtet. Weiterhin ist die Distribution permanent vom Netzwerk getrennt. Auch hier mittels Kernelparameter realisiert, da die Trennung möglichst weit unten im System-Stack durchgeführt werden soll.

Bild 2: Xfce4-Desktop mit allen enthaltenen paper wallet tools, Bildquelle: L. Johns

4. Random number generator

Innerhalb des Live-Systems wird eine sichere Zufallsgenerierung benötigt. Aktuell nutzen fast alle enthaltenen Tools die Browserfunktionen. Es gibt aber auch andere Implementierungen, wie zum Beispiel ein Shellscript für die Generierung von IOTA-Seeds. Hier wird der Kernel-RNG verwendet und dieser bedarf weiterer Maßnahmen. Linux stellt als Entropiequelle zwei Gerätedateien zur Verfügung. Das sind */dev/random* und */dev/urandom*. Gefüllt wird dieser Entropiespeicher mit Rauschwerten aus der Umgebung des Kernels, z.B. von Gerätetreibern [8]. Da ein Systemstart ohne zusätzliche Nutzerinteraktion in der Regel stark vorhersehbar ist, wird die gesammelte Entropie normalerweise beim Herunterfahren des Systems auf der Festplatte gespeichert. Da ein Live-System aber jedes mal mit dem exakt gleichen Datenträgerabbild startet, ist genau dieser Schritt nicht möglich.

Die Problematik von Zufallszahlen in Live-Systemen ist selbst für diesen speziellen Fall nicht neu. Das Betriebssystem Tails hat ebenfalls den Anspruch, kryptografische Werkzeuge nutzen zu können. Die Entwickler von Tails versuchen das Seeding des Linux-Kernels mit zwei zusätzlichen Entropiequellen zu steigern [9]. Speziell handelt es sich dabei um zwei Daemons, die in das System integriert werden. Der erste ist rngd, welcher bei einem vorhandenen Hardware-RNG, diesen für das Seeding des Entropiepools nutzt, bis ein definierter Grenzwert

erreicht ist. Nicht jedes System verfügt über einen Hardware-RNG, sodass rngd nicht in jedem Fall hilft. Deshalb gibt es noch eine zweite Quelle, die mit dem haveged-Daemon realisiert wird. Dieser nutzt den HAVEGE-Algorithmus (Hardware Volatile Entropy Gathering und Expansion), der aus nicht vorhersagbaren Stati des Prozessors, Entropie sammelt [10]. Es handelt sich dabei nicht um einen Ersatz für einen Hardware-RNG. Der Entwickler ordnet den Algorithmus wie folgt ein.

“One could theoretically reproduce the sequence if he/she was able to reproduce all the past events on the machine. They are not pseudo-random either since there is no (short) seed which would allow an exact reproduction of the random sequence. The randomness results instead from an inability to control or predict with sufficient accuracy the events involved in the generation process.” - [11]

Laut der Beschreibung des offiziellen Debian-Pakets ist haveged vor allem für den Einsatz auf Server- und Headless-Systemen mit eingeschränkter Benutzerinteraktion gedacht [12]. Live-Systeme, wie Tails oder die hier entwickelte Distribution haben mit Serversystemen insofern gemein, dass zum Zeitpunkt der Verwendung von Zufallszahlen, wenig Nutzerinteraktionen stattgefunden haben. Der Einsatz von haveged in virtualisierten Umgebungen wird teilweise kontrovers diskutiert [13], der HAVEGE-Algorithmus ist hier unter Umständen keine sichere Entropiequelle. Die hier entwickelte Live-Distribution wird daher generell nicht für den Einsatz in virtualisierten Instanzen empfohlen.

5. Integrierte Software

Anfangs sind Werkzeuge für 13 Kryptowährungen eingeplant. Es besteht die Möglichkeit, dies jederzeit zu erweitern. Die Auswahl der Kryptowährungen soll möglichst viele Nutzer ansprechen. Daher werden einfach die Kryptowährungen mit dem derzeit höchsten Handelsvolumen ausgewählt [14]. In Tabelle 2 sind diese entsprechend aufgelistet.

Eine Ausnahme der ausgewählten Werkzeuge stellt die Software 'IOTA-Paper-Wallet' dar. Hier wird der Private Key nicht im Browser generiert. Das Tool erwartet die Eingabe des 81-stelligen IOTA-Seeds, der aus den Zeichen A-Z und 9 generiert wird \cite{src:IIB0}. Um die Benutzung in der Distribution zu vereinfachen, wird ein kurzes Shell-Skript für die Generierung sicherer Seeds geschrieben, welches automatisch mit dem Öffnen des Browsers, in einem zusätzlichen Terminalfenster, ausgeführt wird. Im Skript werden fünf Seeds mit folgendem Shell-Befehl generiert.

cat /dev/urandom | tr -dc A-Z9 | head -c\${1: -81}

6. Typischer Usecase

Im Folgenden wird ein typisches Anwendungsbeispiel konstruiert und schrittweise durchgeführt. Es sollen zwei Paper-Wallets erstellt und ausgedruckt werden, eins für Litecoin und eins für EOS. Dafür wird ein Abbild der Distribution und ein USB-Stick mit mindestens einem Gigabyte benötigt. Zunächst wird das Abbild auf den Stick geschrieben. Bei einem Windows-System ist dafür ein Tool wie Rufus \cite{src:RRR0} sinnvoll. Unter Linux kann das Shellprogramm \texttt{dd} genutzt werden. In diesem Beispiel wird Linux eingesetzt und der USB-Stick befindet sich hier unter \texttt{/dev/sdd}, sodass sich der folgende Befehl ergibt.

sudo dd if="vinktar-live-image-i386.hybrid.iso" of="/dev/sdd"

Jetzt kann der Stick an den Rechner angesteckt werden, an dem das Live-System verwendet werden soll. Als Nächstes wird das Netzwerkkabel entfernt. Sind manuelle Schalter für WLAN oder Bluetooth vorhanden, können diese zusätzlich ausgeschaltet werden. Außerdem wird der Drucker per USB mit dem Rechner verbunden. Der Computer wird nun gestartet und soll das Live-System booten. Geschieht dies nicht automatisch, muss in der Regel eine bestimmte Taste zur manuellen Auswahl des Bootmediums während des Startvorgangs gedrückt werden. Es wird nun der BIOS-Start gewählt, der dann zum Menü des Bootloaders führt. Hier wird mit der Eingabetaste das System regulär gestartet. Nach nur wenigen Sekunden erscheint der Desktop des Live-Systems. Mit einem Klick auf das Menü *Paper Wallets* werden alle Kryptowährungen aufgelistet, für die das System Paper-Wallets erstellen kann. In diesem Fall wird zunächst EOS ausgewählt. Der Browser öffnet sich und zeigt die generierten Schlüssel inklusive QR-Codes an. Mit einem Klick auf *Print* oder mit der Eingabe von *Strg + P* kann das Paper-Wallet gedruckt werden. Ist der Drucker hier nicht gelistet, lässt er sich in der Regel über die Druckereinstellungen mit *Add* konfigurieren. Anschließend erscheint er in der Druckerauswahl. Bei bestimmten Anforderungen an das Layout des Paper-Wallets, können alternativ auch die im System enthaltenen Office Programme genutzt werden. Beispielsweise können so auch mehrere Schlüssel auf eine Seite gedruckt werden. Ist der Vorgang abgeschlossen, wird über das Menü *Paper Wallets* der Punkt *Litecoin (LTC)* gewählt. Da *liteaddress* noch die Mausbewegung als Entropiequelle hinzuzieht, erscheinen die Schlüssel nicht sofort. Ansonsten kann hier analog verfahren werden. Abschließend wird das System heruntergefahren. Als weitere Maßnahmen sind noch die Trennung vom Stromnetz für einige Sekunden und das Zurücksetzen des Druckers auf Werkseinstellungen zu empfehlen.

Kryptowährung	Software	Umsetzung	RNG
Bitcoin	bitaddress.org	JavaScript	crypto.getRandomValues, user generated entropy
Ethereum	myetherwallet	JavaScript	crypto.getRandomValues
Ripple XRP	ripply.eu	JavaScript	crypto.getRandomValues
Bitcoin Cash	bitcoin.com	JavaScript	crypto.getRandomValues, user generated entropy
EOS	eoscafe paper-wallet	JavaScript	crypto.getRandomValues
Stellar	stellar-paper-wallet	JavaScript	crypto.getRandomValues
Litecoin	liteaddress.org	JavaScript	crypto.getRandomValues, user generated entropy
Monero	monero-walletgenerator	JavaScript	crypto.getRandomValues, user generated entropy
Tron	tronpaperwallet.org	JavaScript	crypto.getRandomValues
IOTA	IOTA-Paper-Wallet	JavaScript	nicht vorhanden
Dash	paper.dash.org	JavaScript	crypto.getRandomValues, user generated entropy
NEO	Ansy	JavaScript	crypto.getRandomValues
Ethereum Classic	myetherwallet	JavaScript	crypto.getRandomValues

Tabelle 2: Integrierte Software zur kryptografischen Schlüsselerzeugung

7. Ausblick

Letztlich ist die Entwicklung einer Linux-Distribution ein sehr umfangreicher Prozess und wird für eine einzelne Person überhaupt erst möglich, durch Projekte wie Live-Build von Debian. Bei der Entwicklung bekannter Distributionen sind daher oft große Teams in diesen Prozess involviert. In Zukunft könnte das Live-System dahingehend ausgebaut werden, dass es nicht mehr nur auf das Erstellen von Paper-Wallets beschränkt ist. Denkbar ist hier eine Live-Distribution für generelle Aufgaben im Bereich Kryptowährungen, die in einer isolierten Umgebung sinnvoll sind.

Literaturverzeichnis

[1] Lookout. blog.lookout.com. 3 fake Bitcoin wallet apps appear in (and are quickly removed from) Google Play Store. [Online] 2017. https://blog.lookout.com/fake-bitcoin-wallet, abgerufen am 07.11.2018

[2] Mirko Ross. heise.de. Kryptowährung: IOTA im Wert von vier Millionen US-Dollar geklaut. [Online] 2018. https://www.heise.de/ix/meldung/Kryptowaehrung-IOTA-im-Wert-von-vier-Millionen-US-Dollar-geklaut-3952723.html, abgerufen am 05.09.2018

[3] Lawrence Abrams. bleepingcomputer.com. Clipboard Hijacker Malware Monitors 2.3 Million Bitcoin Addresses. [Online] 2018. https://www.bleepingcomputer.com/news/security/clipboard-hijacker-malware-monitors-23-million-bitcoin-addresses/, abgerufen am 12.11.2018

[4] Open Berkeley. security.berkeley.edu. Network Printer Security Best Practices | Berkeley Information Security and Policy. [Online] 2018. https://security.berkeley.edu/resources/best-practices-how-articles/system-application-security/network-printer-security-best, abgerufen am 08.11.2018

[5] Debian Live Team. debian developers' corner. Debian Live Project. [Online] 2018. https://www.debian.org/devel/debian-live/index.en.html, abgerufen am 10.10.2018

[6] Debian Projekt. Debian Nachrichten. Debian 6.0 Squeeze wird mit vollständig freiem Linux-Kernel veröffentlicht. [Online] 2010. https://www.debian.org/News/2010/20101215, abgerufen am 27.09.2018

[7] TiberiusT. forum.xfce.org. Panel config files are where?. [Online] 2012. https://forum.xfce.org/viewtopic.php?id=7671, abgerufen am 03.11.2018

[8] Michael Kerrisk Ubuntu Manpage Repository. Ubuntu Manpage: random, urandom - Kernel-Geräte zur Erzeugung von Zufallszahlen. [Online] 2018. http://manpages.ubuntu.com/manpages/precise/de/man4/random.4.html, verfügbar am 01.09.2018, 11:35.

[9] Tails project. tails.boum.org. Tails - Random numbers. [Online] 2018. https://tails.boum.org/contribute/design/random/, verfügbar am 15.11.2018

[10] Olivier Rochecouste. irisa.fr. HAVEGE Hardware Volatile Entropy Gathering and Expansion, Overview. [Online] 2006. http://www.irisa.fr/caps/projects/hipsor/, verfügbar am 15.11.2018

[11] Olivier Rochecouste. irisa.fr. Execution time of a short sequence of instructions and hardware volatile states in a modern microprocessor. [Online] 2006. http://www.irisa.fr/caps/projects/hipsor/misc.php #measure, verfügbar am 15.11.2018

[12] Debian Team. packages.debian.org. Debian -- Informationen über Paket haveged in buster. [Online] 2018. https://packages.debian.org/buster/haveged, verfügbar am 15.11.2018

[13] Nic Waller. security.stackexchange.com. Is it appropriate to use haveged as a source of entropy on virtual machines?. [Online] 2013. https://security.stackexchange.com/questions/34523/is-it-appropriate-to-use-haveged-as-a-source-of-entropy-on-virtual-machines, verfügbar am 16.11.2018

[14] CoinMarketCap. coinmarketcap.com. Cryptocurrencies Market Capitalization. [Online] 2018. https://coinmarketcap.com/, verfügbar am 19.10.2018

[15] Rufus USB. rufususb.com. Rufus - bootable USB flash drive. [Online] 2018. http://rufususb.com/, verfügbar am 13.11.2018

VERIFICATION OF BITCOIN IN THE INCUBED PROTOCOL

Tim Käbisch
Slock.it GmbH, Markt 16, D-09648 Mittweida

Abstract

To enable smart devices of the internet of things to be connected to a blockchain, a blockchain client needs to run on this hardware. With the Trustless Incentivized Remote Node Network, in short Incubed, it will be possible to establish a decentralized and secure network of remote nodes, which enables trustworthy and fast access to a blockchain for a large number of low-performance IoT devices. Currently, Incubed supports the verification of Ethereum data. To serve a wider audience and more applications this paper proposes the verification of Bitcoin data as well, which can be achieved due to the modularity of Incubed. This paper describes the proof data that is necessary for a client to prove the correctness of a node's response and the process to verify the response by using this proof data as well. A proof-object which contains the proof data will be part of every response in addition to the actual result. We design, implement and evaluate Bitcoin verification for Incubed. Creation of the proof data for supported methods (on the server-side) and the verification process using this proof data (on the client-side) has been demonstrated. This enables the verification of Bitcoin in Incubed.

1. Introduction

"The blockchain data structure is an ordered, back-linked list of blocks of transactions. Each block within the blockchain is identified by a hash, generated using a cryptographic hash algorithm on the header of the block. Each block references a previous block, known as the *parent* block, through the "previous block hash" field in the block header. The sequence of hashes linking each block to its parent creates a chain going back all the way to the first block ever created, known as the *genesis block*." [1]

At the moment the most famous use case of the blockchain technology are cryptocurrencies. Besides Bitcoin and Ethereum there are around 6,500 [2] more at the time of writing. Many more use cases are about to follow and are currently being developed.

To enable the interaction with a blockchain a device needs to install and run a blockchain client. While current notebooks and desktop computers have enough computational power, storage space and bandwidth to run a full node, smaller devices like smartphones or tablets with less powerful hardware or restricted internet connection are capable of running a light node (also known as SPV-client for Bitcoin). However, many IoT devices are severely constrained in terms of computational power and internet connection that even a light node is too "big" to run on such devices. Connecting an IoT device to a remote node still enables the connection to a blockchain. But, by using remote nodes a big advantage of a decentralized network is being undermined: not being forced to trust a single player. The risk of malfunction or an attack is very high since there is a single point of failure. Therefore, we need a blockchain client which is small enough to run on an IoT device and which can act independently in a network of players, hence not being forced to trust a single node. [3]

"The Trustless Incentivized Remote Node Network, in short Incubed, makes it possible to establish a decentralized and secure network of remote nodes and clients which are able to verify and validate the results, enabling trustworthy and fast access to blockchain for many low-performance IoT, mobile devices, and web applications." [4]

2. Incubed

Incubed solves the following problems which are preventing an IoT device to run a light node: [5]

1. **Insufficient computing power and storage space:** Incubed takes up very little storage space

(200 kB for the implementation in C).

2. **Insufficient power supply:** Incubed does not require a continuous power supply and therefore can be switched off after each usage.

3. **No continuous connection to the internet:** Incubed does not need a continuous internet connection, but only builds it if required – this is only possible because Incubed is a non-synchronizing client.

The Incubed network consists of the following components: [6]

1. **Incubed Registry:** This is a smart contract on the Ethereum blockchain. Nodes that want to participate in the network must register and store a security deposit.

2. **Incubed Node:** Full nodes of a blockchain which provide information and act as a validator.

3. **Incubed Client:** Clients that request information from Incubed nodes and can be installed on IoT devices, among others.

4. **Watchdogs:** Autonomous authorities (bots) who are responsible for the detection and punishment of fraudulent nodes.

Figure 1 shows the flow of an RPC request in the Incubed network. The client sends a request to Node B and requests signatures from Nodes A and C on the provided answer by B. Node B sends its (unsigned) response to the requested signature nodes. They check the response and answer B with their signature if the answer from B is correct. After receiving the signatures from the required nodes B will send a response to the client including the actual result and the signatures. The client assumes that the

majority of the nodes act honest and therefore considers the result as correct once he received and checked the corresponding signatures. In case B tries to send a fake answer to the client honest nodes would not sign this reply and in addition convict node B in the registry. Node B would then be removed from the registry and would lose his security deposit.

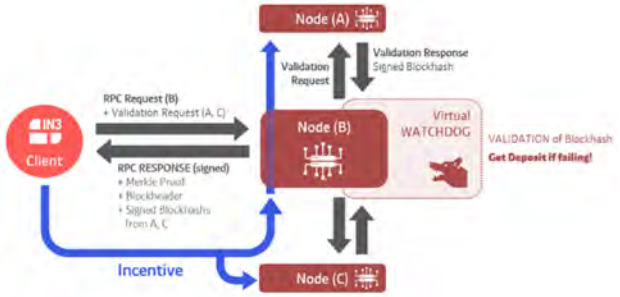


Fig. 1: Flow of an RPC request in the Incubed network

“As an incentive system for the return of verified responses, the node can request a payment. For this, however, the node must guarantee with its security deposit that the answer is correct. […] The security deposit of the node has a decisive influence on how much trust is placed in it. When selecting the node, a client chooses those nodes that have a corresponding deposit (stake), depending on the security requirements (e.g. high value of a transaction).” [7]

Lets have a look at a real world example of the usage of Incubed: a **smart bike lock**. The rental of an e-bike will be managed by a smart contract deployed on Ethereum. The lock is powered by the battery of the e-bike. It is equipped with a microchip to perform authorization checks and open the lock if necessary. Since the lock operates on the limited power of the e-bike, an internet connection is only established when needed for a check – therefore saving power in the remaining time. The installation of a blockchain client on the lock is necessary to establish a connection to the Ethereum blockchain. Installing a light node is not possible due to the limited resources (limited power supply, low computing power, no stable internet connection). Turning the light node on and off after each usage would indeed save electricity but would force the client to synchronize itself each time it comes back online – this would take too much time and requires a good internet connection. With an Incubed client running on the lock, a secure connection to the blockchain can be established at the required times only. Neither computing power is needed nor data is transferred in times when there is no rental process in action. [8]

Currently Incubed supports the verification of Ethereum data. Supporting Bitcoin, which is the largest cryptocurrency with a market cap of $210.89B and around 320,000 transactions per day [9,10], will serve a wider audience and enables many use cases for the users of Incubed. The most important use case is the verification of payments on the Bitcoin chain. Incubed clients will be able to prove the existence and correctness of a transaction (a use case here would be online shop payments). There are many more applications based on Bitcoin, for example a “proof of existence” (storing the hash of a document on the Bitcoin chain - github.com/proofofexistence).

3. Fundamentals

For the verification of Bitcoin we make use of the Simplified Payment Verification (SPV) proposed in the Bitcoin paper by Satoshi Nakamoto.

“It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he’s convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it’s timestamped in. He can’t check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker’s fabricated transactions for as long as the attacker can continue to overpower the network.” [11]

In contrast to SPV-clients an Incubed client does not keep a copy of all block headers, instead the client is stateless and only requests required block headers. We are following a simple process: A client requests certain data, the server sends a response with proof data in addition to the actual result, the client verifies the result by using the proof data. We rely on the fact that it is extremely expensive for an attacker to deliver a wrong block (wrong data) which still has following blocks referring the wrong block (i.e. delivering a chain of fake-blocks). This approach does not really work for very old blocks. Beside the very low difficulty at this time, the miner has many years of time to pre-mine a wrong chain of blocks. Therefore, we are setting some hard-coded checkpoints of hashes of bygone blocks. Proving the correctness of old blocks can be achieved by checking the linking from the requested block to a certain checkpoint (the server needs to provide the corresponding data). The only way for an attacker to fool the client would be by finding a hash collision.

3.1 Mining in Bitcoin

The process of trying to add a new block of transactions to the Bitcoin blockchain is called “mining”. Miners are competing in a network-wide competition, each trying to find a new block faster than anyone else. The first miner who finds a block broadcasts it across the network and other miners are adding it to their blockchain after verifying the block. Miners restart the mining-process after a new block was added to the blockchain to build *on top* of this block. As a result, the blockchain is constantly growing – one block every 10 minutes on average.

But how can miners FIND a block? They start by filling a candidate block with transactions from their memory pool. Next they construct a block header for this block, which is a summary of all the data in the block including a reference to a block that is already part of the blockchain (known as the parent hash). Now the actual mining happens: miners put the block header through the *SHA256* hash function and hope that the resulting hash is below the current target. If this is not the case, miners keep trying by incrementing a number in the block header resulting in a completely different hash. This process is referred to as **proof-of-work.** [12]

3.2 Finality in Bitcoin

In terms of Bitcoin, finality is the assurance or guarantee that a block and its included transactions will not be revoked once committed to the blockchain. Bitcoin uses a probabilistic finality in which the probability that a block will not be reverted increases as the block sinks deeper into the chain. The deeper the block, the more likely that the fork containing that block is the longest chain. After being 6 blocks deep into the Bitcoin blockchain it is very unlikely (but not impossible) for that block to be reverted. [13]

3.3 Difficulty Adjustment Period

The white paper of Bitcoin specifies the block time as 10 minutes. Due to the fact that Bitcoin is a decentralized network that can be entered and exited by miners at any time, the computing power in the network constantly changes depending on the number of miners and their computing power. In order to still achieve an average block time of 10 minutes a mechanism to adjust the difficulty of finding a block is required: the **difficulty**.

The adjustment of the difficulty occurs every 2016 blocks - roughly every two weeks (which is one epoch/period). Bitcoin is a decentralized network and therefore there is no central authority which adjusts the difficulty. Instead, every miner compares the expected time to mine 2016 blocks (20160 minutes) with the actual time it took to mine the last 2016 blocks (using timestamps of the first and last block). The difficulty increases when the blocks are mined faster than expected and vice versa. Although the computing power increased heavily since the introduction of Bitcoin in 2009 the average block time is still 10 minutes due to this mechanism.

What is the difference between the difficulty and the target? The difficulty is a big number used for the adjustment process. The target is used for the mining process and for the verification of a block hash. The hash of a block has to be smaller than the target to be accepted across the network (as mentioned in 3.1). The target can be calculated using the difficulty and the constant value *targetmax*:

$$target = \frac{targetmax}{difficulty}$$

$$targetmax = 0x00000000FFFF00000000000000000000000000000000000000000000000000000000$$

4. Risk Calculation

The following calculation outlines the security (in terms of $) when the client is requesting one of the newer blocks and 6 finality headers. This results in a total of 7 fake-blocks that an attacker has to calculate to fool the client. The calculation is based on assumptions and averages.

Assume that the attacker has 10% of the total mining power. This would mean he needs around 100 minutes to mine 1 block (average block time of Bitcoin is 10 minutes) and around 700 minutes to mine 7 blocks. While mining fake-blocks, the attacker loses his chance of earning block rewards. Assuming that we would have been able to mine 7 blocks, with a current block reward of 6.25 BTC and $11,400 per Bitcoin at the time of writing [14]:

$$7 * 6.25\ BTC = 43.75\ BTC$$

$$43.75\ BTC * \frac{\$11{,}400}{1\ BTC} = \$\mathbf{498{,}750}$$

Furthermore, the attacker needs to achieve 10% of the mining power. With a current total hash rate of 120 EH/s, this would mean 12 EH/s. There are two options: buying the hardware or renting the mining power from others. [15]

A new Antminer S9 with 14 TH/s can be bought for $3,000 [16]. This would mean an attacker has to pay $2,568,000,000 to buy so many of these miners to reach 12 EH/s. The costs for electricity, storage room and cooling still needs to be added.

Hashing power can also be rented online. Obviously nobody is offering to lend 12 EH/s of hashing power – but for this calculation we assume that an attacker is still able to rent this amount of hashing power. The website nicehash.com is offering 1 PH/s for 0.0098 BTC (for 24 hours). [17]

$$1\frac{PH}{s} = 0.0098\ BTC \rightarrow 12\frac{EH}{s} = 117.6\ BTC$$

Assuming it is possible to rent it for 700 minutes only (which would be 48.6% of one day).

$$117.6\ BTC * 0.486 = 57.15\ BTC$$

$$57.15\ BTC * \frac{\$11{,}400}{1\ BTC} = \$\mathbf{651{,}510}$$

Therefore, 6 finality headers provide a security of estimated **$1,150,260** in total. Figure 2 uses the assumptions and numbers from the calculation shown above to outline the costs to mine *n* fake-blocks.

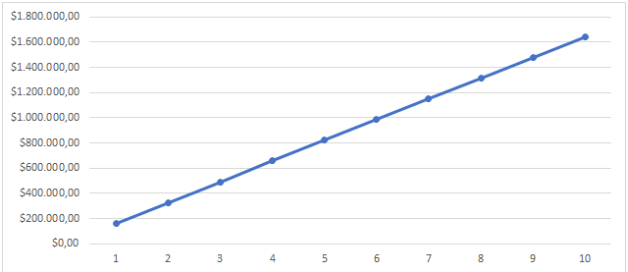


Fig. 2: Cost to mine n fake-blocks

5. Proofs

A subset of the following proofs needs to be performed to prove the correctness of certain data. The subset depends on the data itself.

5.1 Target Proof

Bitcoin uses the target for the mining process where miners are hashing the block data over and over again to find a hash that is smaller than the target (while changing the data a little each try to generate a different hash). Miners across the network can verify newly published blocks by checking the block hash against the target. The same applies for clients. Having a verified target on the client-side is important to verify the proof of work and therefore the data itself (assuming that the data is correct when someone put a lot of work into it). Since the target is part of a block header (*bits*-field) we can verify the target by verifying the block header. This is a dilemma since we want to verify the target by verifying the block header, but we need a verified target to verify the block header (explained in 5.2).

There are two options to verify a target, whereby only option one is implemented at time of writing - option two has not been implemented yet and is still being discussed.

5.1.1 Verification using finality headers

The client maintains a cache with the number of a difficulty adjustment period (dap) and the corresponding target (which stays the same for the duration of one period). This cache will be filled with default values at the time of the release of the Bitcoin implementation. If a target is not yet part of the cache it needs to be verified first and added to the cache afterwards.

We completely rely on the finality of a block. We can verify the target of a block (and therefore for a whole period) by requesting a block header (*getblockheader*) and *n*-amount of finality headers. If we are able to prove the finality using the finality proof we can consider the target as verified. The client sets a limit in his configuration regarding the maximum change of the target from a verified one to the one he wants to verify. The client will not trust the changes of the target when they are too big (i.e. greater than the limit). For such cases we implemented a special *proofTarget*-method to verify big changes of the target in smaller steps (explained in 6.7)

5.1.2 Verification using signatures

This approach uses the design of Incubed: verify a result by requesting signatures from other nodes. At time of writing this approach is still in development and discussion for Bitcoin.

Since the target is part of the block header we just have to be very sure that the block header is correct - which leads us to a correct target. The client fetches the node list and chooses n nodes which will provide a signature. Afterwards it sends a *getblockheader*-request (also containing the addresses of the selected nodes) to a random provider node. This node asks the signatures nodes to sign his result (the block header). The response will include the block header itself and all the signatures as well. The client can verify all signatures by using the node list and therefore verifying the actual result (a verified block header and therefore a verified target). The incentivization for the nodes to act honestly is their deposit which they will lose in case they act maliciously. Have a look at Fig. 1 for a better explanation of this process.

5.2 Block Proof

Verifying a Bitcoin block is quite easy when you already have a verified block hash. We take the first 80 bytes of the block data (which is the block header) and hash it with *SHA256* twice. Since Bitcoin stores the hashes in little endian we have to reverse the order of the bytes. In order to check the proof of work in the block header we compare the target with the hash. We accept the proof of work when the block hash is smaller than the target.

5.3 Finality Proof

Necessary data to perform this proof:

- Block header (block *X*)
- Finality block header (block *X+1*, …, *X+n)*

The finality for block *X* can be proven as follows: The proof data contains the block header of block *X* as well as *n* following block headers as finality headers. In Bitcoin every block header includes a *parentHash*-field which contains the block hash of its predecessor. By checking this linking the finality can be proven for block *X*. Meaning the block hash of block *X* is the *parentHash* of block *X+1*, the hash of block *X+1* is the *parentHash* of block *X+2*, and so on. If this linking is correct until block *X+n* (i.e. the last finality header) then block *X* can be considered as final. As mentioned earlier Bitcoin uses a probabilistic finality, meaning a higher *n* increases the probability of being actual final.

5.4 Transaction Proof

Necessary data to perform this proof:

- Block header
- Transaction
- Merkle proof
- Index (of this transaction)

All transaction of a Bitcoin block are stored in a **merkle tree**. Every leaf node is labelled with the hash of a transaction, and every non-leaf node is labelled with the hash of the labels of its two child nodes. This results in one single hash (the **merkle root**) which is part of the block header. Attempts to change or remove a leaf node after the block was mined (i.e. changing or removing a transaction) will not be possible since this will cause changes in the merkle root, thereby changes in the block header and therefore changes in the hash of this block. By

hashing the block header and checking this hash against the block hash such attempts will definitely be discovered. Having a verified block header and therefore a verified merkle root allows us to verify and prove the existence and correctness of a certain transaction.

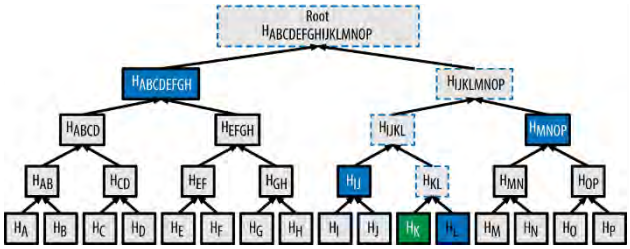


Fig. 3: Merkle Tree

In order to verify the existence and correctness of transaction [K] we use *SHA256* to hash [K] twice to obtain H(K). For this example the merkle proof data will contain the hashes H(L), H(IJ), H(MNOP) and H(ABCDEFGH). These hashes can be used to calculate the merkle root as shown in Fig. 2. The hash of the next level can be calculated by concatenating the two hashes of the level below and then hashing this hash with *SHA256* twice. The index determines which of the hashes is on the right and which one on the left side for the concatenation (Hint: swapping the hashes will result in a completely different hash). When the calculated merkle root appears to be equal to the one contained by the block header we've hence proven the existence and correctness of transaction [K]. This can be done for every transaction of a block by simply hashing the transaction and then keep on hashing this result with the next hash from the merkle proof data. The last hash must match the merkle root. [18, 19]

5.5 Block Number Proof

Necessary data to perform this proof:

• Block header

• Coinbase transaction (first transaction of the block)

• Merkle proof (for the coinbase transaction)

In comparison to Ethereum there is no *block number* in a Bitcoin block header. Bitcoin uses the height of a block, which is the number of predecessors. The genesis block is at height 0 since there are no predecessors (the block with 100 predecessors is at height 100). Therefore, you need to know the complete Bitcoin blockchain to verify the height of a block (by counting the links back to the genesis block). Hence, actors that do not store the complete chain (like an Incubed client) are not able to verify the height of a block (i.e. the number). To change that Gavin Andresen proposed a change to the Bitcoin protocol in 2012.

Bitcoin Improvement Proposal 34 (BIP-34) introduces an upgrade path for versioned transactions and blocks. A unique value is added to newly produced coinbase transactions, and blocks are updated to version 2. After block number 227,835 all blocks must include the block height in their coinbase transaction. [20]

For all blocks after block number 227,835 the block number can be proven as follows:

1. Extract block number out of the coinbase transaction:

Coinbase transaction of block #624692 [21]

```
03348809041f4e8b5e7669702f7777772e6f6b657
82e636f6d2ffabe6d6db388905769d4e3720b1e59
081407ea75173ba3ed6137d32308591495198155c
e0200000004204cb9a2a31601215b2ffbeaf1c4e00
```

Decode:

a) **03**: first byte signals the length of the block number (push the following 3 bytes)

b) **348809**: the block number in big endian format (convert to little endian)

c) **098834**: the block number in little endian format (convert to decimal)

d) **624692**: the actual block number

e) **041f4e…**: the rest can be anything

2. Prove the existence and correctness of the coinbase transaction:

To trust the extracted block number it is necessary to verify the existence and correctness of the coinbase transaction. This can be done by performing a *merkle proof* (explained in 5.4) using the provided block header and the merkle proof data. The size of the block number proof is **764 bytes** on average. [22]

6. Implementation

The implementation is split into two parts: the typescript-based server and the client based on C. The implementation in the server was done by Tim Käbisch, while Simon Jentzsch, Vice President of Blockchain Development at Blockchains LLC, did the implementation in the client. [23] An Incubed server acts as a kind of proxy server. In comparison to a standard Bitcoin full node an Incubed server is going to provide the corresponding proof data to verify the result on the client-side.

There are two options for an Incubed server to access Bitcoin data: Running a Bitcoin fullnode on the same machine as the Incubed server or connecting to a remote (trusted) Bitcoin full node. The operator of the Incubed server should make sure that he has access to *correct* Bitcoin data. This is in his own interest since he would lose his security deposit in case he delivers any wrong data. Therefore, operating a Bitcoin full node and the Incubed server on the same machine (or in the same trusted network) is highly recommended.

We found a way to prove the results of the standard Bitcoin RPC request mentioned in 6.2 - 6.6 by adding proof data and performing the corresponding proofs on the client-side. A proof object contains a subset of the following properties:

• *final* the finality headers, which are hex coded bytes of the following headers (80 bytes each) concatenated, the number depends on the requested

finality (*finality*-property in the *in3*-section of the request)

• *cbtx* serialized coinbase transaction of the block (this is needed to get the verified block number)

• *cbtxMerkleProof* merkle proof of the coinbase transaction, proving the correctness of the cbtx

• *block* a hex string with 80 bytes representing the block header

• *txIndex* index of the transaction (*txIndex=0* for coinbase transaction, necessary to create/verify the merkle proof)

• *merkleProof* merkle proof of the requested transaction, proving the correctness of the transaction

6.1 Cache

Server. The server is using a simple map with the block hash and the block number pointing to the same cache-object. A cache-object consists of the block height, block hash, block header, transactions ids and the coinbase transaction.

Client. The client is using a map with the number of a difficulty adjustment period (dap) pointing at the (verified) target of this dap. At the time of release we will fill this cache with default values for bygone daps. Every time the client can't find a needed target in his cache he is going to verify it by using the target proof (mentioned in 5.1) and stores it in the cache afterwards. Additionally, the client can use the proofTarget-method (mentioned in 6.7) to verify unusually large changes of the target.

6.2 getblockheader

Server. Returns data of block header for given block hash (returned level of details depends on *verbosity*). The server adds finality headers, the coinbase transaction and the merkle proof for the coinbase transaction to the actual result.

Client. Proves the result by performing a finality proof and block number proof.

6.3 getblock

Server. Returns data of block for given block hash (returned level of details depends on *verbosity*). The server adds finality headers, the coinbase transaction and the merkle proof for the coinbase transaction to the actual result.

Client. Uses the proof data to verify the result by performing a finality proof and a block number proof.

6.4 getrawtransaction

Server. Returns the raw transaction data (returned level of details depends on *verbosity*). The server adds the block header, finality headers, transaction index, merkle proof for the requested transaction, coinbase transaction and merkle proof for the coinbase transaction.

Client. The block header and the finality headers are used to perform a finality proof. By doing a merkle proof using the transaction index and the merkle proof for the requested transaction the correctness can be proven. Furthermore, the client is going to perform a block number proof using the coinbase transaction and its merkle proof.

6.5 getblockcount/getbestblockhash

The functionality of these two methods is the same, only the return value is a little different – the number of blocks in the longest chain for *getblockcount* and the hash of the best block in the longest chain for *getbestblockhash*. Since the server can not prove the finality of the latest block (obviously the finality headers are not existing yet) we consider the *current block count MINUS amount of finality* (set in the request) as the latest block.

Server. Returns "latest" block number/block hash of the longest chain. The server adds the block header, finality headers, coinbase transaction and the merkle proof for the coinbase transaction to the actual result.

Client. Proves the result by performing a finality proof and block number proof.

6.6 getdifficulty

Server. Returns the proof-of-work difficulty as a multiple of the minimum difficulty. Depending on the parameter the server will return the difficulty of a certain block number or the difficulty of the "latest" block (latest in the same sense as mentioned in 6.5).

Client. Uses the proof data to verify the result by performing a finality proof and a block number proof. Since the difficulty is not part of the block it can be checked by transforming it into a target (as mentioned in 3.3).

6.7 proofTarget

Whenever the client is not able to trust the changes of the target (which is the case if a block can not be found in the verified target cache *and* the value of the target changed more than the client's limit *max_diff*) he will call this method. It will return additional proof data to verify the changes of the target on the client-side. The server will provide a path of daps with corresponding proof data. By performing a finality proof and a block number proof for each dap of the path the client is able to verify the changes of the target step by step instead of having to trust a big change of the target (see Fig. 4).

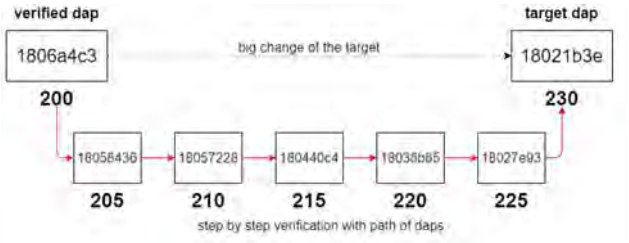


Fig. 4: Visualization of the *proofTarget*-method

7. Example

This example shows a *getblockheader*-request from the client and the corresponding response from the server including the proof data.

Request:

```
{
  "jsonrpc": "2.0",
  "id":1,
  "method": "getblockheader",
  "params":
["0000000000000000000103b2395f6cd94221b10d02
eb9be5850303c0534307220", false],
  "in3": {
    "finality": 8,
    "verification": "proof"
  }
}
```

Response:

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result":
"00e00020497f4c193dbb347c2ecfcf6169e64c7478
778340a0cb13000000000000000000cb59b377938a5
32525d52b8404390f17c35a50beb9d57292f3658cea
34a386d7e4888d5e413b14174f045476",
  "in3": {
    "proof": {
      "final": "0x00e0ff27207230...276470",
      "cbtx": "0x010000000001010...9da2fc",
      "cbtxMerkleProof": "0x6a807...52e688"
    },
  "lastNodeList": 3290882,
  "execTime": 1459,
  "rpcTime": 0,
  "rpcCount": 0,
  "currentBlock": 3290942,
  "version": "2.1.0"
  }
}
```

The client can verify the result by itself by performing the corresponding proofs. The finality headers from the final-field will be used to perform a finality proof. To verify the block number the client is going to perform a block number proof using the coinbase transaction (cbtx-field) and the merkle proof for the coinbase transaction (cbtxMerkleProof-field).

8. Conclusion

This paper has shown the basics of the Incubed protocol, fundamentals of Bitcoin, necessary proofs and the actual implementation of the verification of Bitcoin in the Incubed protocol. At the time of writing we are about to release version 1 of the Bitcoin implementation. This enables the verification of Bitcoin in the Incubed protocol. Version 2 is still in the discussion and development phase. Once implemented, it will enable the on-chain conviction for malicious Bitcoin nodes. This also enables the verification of a balance of a Bitcoin address (since the balance is the sum of all UTXOs and this can be checked by every node – nodes that deliver a wrong result will surely be convicted).

The documentation can be found on Incubed's readthedocs and as soon as version 1 of the Bitcoin implementation is released it will be available on GitHub.

Documentation: https://in3.readthedocs.org/

Client: https://github.com/blockchainsllc/in3-c

Server: https://github.com/blockchainsllc/in3-server

9. References

[1] Andreas M. Antonopoulos, Mastering Bitcoin (2017), 211

[2] CoinMarketCap (19.08.2020) https://coinmarketcap.com/de/all/views/all/

[3] Incubed Readthedocs (18.08.2020) https://in3.readthedocs.io/en/develop/intro.html

[4] Incubed GitHub (18.08.2020) https://github.com/blockchainsllc/in3

[5] Incubed Readthedocs (19.08.2020) https://in3.readthedocs.io/en/develop/intro.html#low-performance-hardware

[6] Incubed Readthedocs (19.08.2020) https://in3.readthedocs.io/en/latest/intro.html#architecture

[7] Incubed Readthedocs, Payment and Incentives (19.08.2020) https://in3.readthedocs.io/en/latest/intro.html#validated-requests

[8] Incubed Readthedocs, Smart Lock (20.08.2020) https://in3.readthedocs.io/en/latest/intro.html#smart-bike-lock

[9] Bitcoin Statistics (27.08.2020) https://www.coindesk.com/price/bitcoin

[10] Bitcoin Statistics (27.08.2020) https://www.blockchain.com/charts

[11] Bitcoin Paper (25.08.2020) https://bitcoin.org/bitcoin.pdf

[12] Learnmeabitcoin, Mining https://learnmeabitcoin.com/technical/mining

[13] Medium, Finality in Blockchain Consensus (19.08.2020) https://medium.com/mechanism-labs/finality-in-blockchain-consensus-d1f83c120a9a

[14] Coindesk (28.08.2020) https://www.coindesk.com/price/bitcoin

[15] Bitcoin Hash Rate (28.08.2020) https://charts.bitcoin.com/btc/chart/hash-rate

[16] Antminer S9 (28.08.2020) https://www.buybitcoinworldwide.com/mining/hardware/

[17] Nicehash (28.08.2020) https://www.nicehash.com/marketplace

[18] Medium, Merkle Proof (20.08.2020) https://medium.com/crypto-0-nite/merkle-proofs-explained-6dd429623dc5

[19] Wikipedia, Merkle Tree (20.08.2020) https://en.wikipedia.org/wiki/Merkle_tree

[20] GitHub, BIP-0034 (21.08.2020) https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki

[21] Blockchair, Coinbase Transaction (21.08.2020) https://blockchair.com/bitcoin/transaction/02d8cdb103f50532e2f18d9d1f85c016468ee0294908d387e38f80b99410d893

[22] Incubed Readthedocs, Block Number Proof Size https://in3.readthedocs.io/en/latest/bitcoin.html#block-number-proof

[23] Blockchains LLC (28.08.2020) https://www.blockchains.com/

MATHEMATICS BEHIND THE ZCASH

Nomana Ayesha Majeed
Hochschule Mittweida, Technikumplatz 17, D-09648 Mittweida

Among all the new developed cryptocurrencies, Zcash comes out to be the strongest cryptocurrency providing both transparency and anonymity to the transactions and its users by deploying the strong mathematics of zk-SNARKs. We discussed the zero knowledge proofs as a building block for providing the functionality to zk-SNARKs. It offers schnorr protocol which is further used in Zcash transactions where the validation of sent transaction is proved by cryptographic proof. Further, we deploy zk-SNARKs following common reference string that allows sender to prove that she knows a secret such that the proof is succinct, can be verified and does not leak the secret. Non-malleability, small proofs and effective verification make zk-SNARKs a classic tool in Zcash. We deal with NP problems therefore we have considered the elliptic curve cryptography to provide the security. Lastly, we explain Zcash transaction, the corresponding transaction completely hides the sender, receiver and amount of transaction using zero knowledge proof.

1. Introduction

As for prerequisites, reader is expected to familiar with the Bitcoin [1]**;** first decentralized cryptocurrency developed in 2008 by Satoshi Nakamoto in the world of cryptocurrencies and act as a base currency in the development of other cryptocurrencies. Bitcoin neither depend on the bank nor rely on any government instead it use a distributed ledger referred as a Blockchain which makes it completely decentralized. However, Bitcoin suffer from some limitations:

- Privacy issue.
- Functionality and scalability limitations.
- Lack of Fungibility.

Beginning from the Zerocoin; a decentralized mix which extends the Bitcoin and is the underlying academic work presented in 2014. The main flaw to Zerocoin is that it only hides the origin of the amount but do not hide the amount itself and receiver of the payment. Another complication with the Zerocoin is its performance; it depends on zero knowledge double discrete logarithm proof which is 25 kbs large and requires 10 seconds for the verification of each spending on the blockchain. It also lake functionality due to the fixed denomination of the coins.

To overcome all of the above mention problems, a team of six outstanding scientists including Zooko Wilcox developed Zcash in October 2016. The edge of using Zcash lies in the fact that Zcash provides the confidentiality and fungibility to its users and their transactions. Beside ensuring the privacy the users can split and merge their coins as well. Zcash is interchangeable due to its anonymity, so one can replace the block of Zcash with another unit of equal value of all coins.

Zcash is defined as a Decentralized Anonymous Payment scheme because it is:

- **Decentralized:** works when given any ideal global ledger like Blockchain.
- **Privacy Preserving**: Anyone can pay directly through payment transactions while keeping origin, amount and the receiver hidden.
- **Efficient:** The transactions take $< 1\,m$ to create and $< 6\,ms$ for verification of the proof.

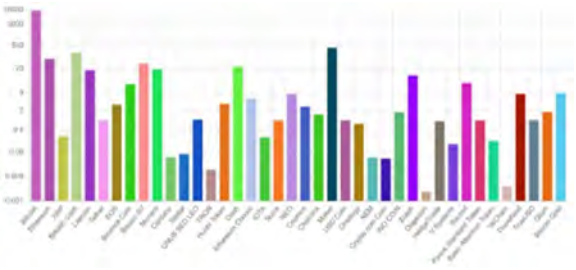
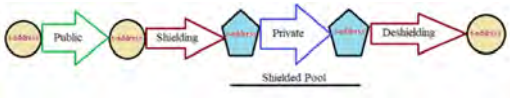


Figure 1.1: Top ranked cryptocurrencies

Beside having shielded transaction, Zcash also support transparent transaction which are same as in Bitcoin in such a way that transparent transactions reveal the pseudonymous addresses of the sender and receiver but to make sure that all the new coins have been shielded once at least it is required to pass them through the shielded pool [2].



1.1 History of Zcash

Symmetric key encryption used by the Julius Ceaser was 1950s is not useful for two parties to communicate with each other unless they are able to deliver the key securely earlier. In 1960's to 1970's Diffie, Hellman and Merkle introduced public key cryptography where Alice uses the public key of Bob to send the message while Bob uses his own private key to read it [3]**.** Later on, Goldwasser et al. (1980) came up with a third new invention of Zero knowledge Proofs. Satoshi Nakamoto (2009), developed blockchain by deploying the idea of public key cryptography where every block contains the proof-of-work and hash. The effective advantage is that one can distribute and share the data in a large group of people without relying on any central party. Finally, Zcash developed in 2016 deployed the idea of ZKP and zk-SNARKs.

1.2 Blockchain

Blockchain is a public ledger of information developed by Satoshi Nakamoto. Since we are working on digital currencies thus, the blockchain provides a digital way to store data and makes it irreversible. Every individual block consists of data, timestamp, hash value of the current and previous block.

To make the blockchain secure it is required to have the same hashes in the current and previous block which makes a chain. Following steps by [4] are required to creates the block for transaction:

1. Assume that Alice has x_1, x_2,..., x_n in her account, she wants to send a transaction to Bob let say y units.
2. Transaction is mint and represented as block.
3. The block is received by every other network.
4. The network node will check the validity of ownership, the Alice is not double spending and $y \leq \sum_{i=1}^{n} x_i$.
5. If everything works well, each participant adds this new block to their own blockchains.
6. All the network nodes make authorization.
7. Finally, transaction is added to Bob's account.

2. Zero Knowledge Proofs

Zero-Knowledge Proofs (ZKP) are used to show the verifier that some secret x belongs to some set $\mathcal{L}$ without providing any information about x except the correctness [5]. Let $\mathcal{L}$ be an NP language defined as $\mathcal{L} = \{ x \mid g^n = x \}$, it follows three properties of zero knowledge proofs as:

1. Completeness: Verifier will be convienced by the prover only if statement x ∈ $\mathcal{L}$.
2. Soundness: A prover cannot prove an honest verifier if the statement x ∉ $\mathcal{L}$ [6].
3. Zero-Knowledge: If the verifier is convienced that x ∈ $\mathcal{L}$ is correct, even though he will not gain any additional information about the secret.

2.1 From interactive to non-interactive

In 1980, Fiat and Shamir provides a method called Fiat-Shamir Heuristic where no interaction is needed between the two parties.It is simply achieved by applying the cryptographic hash function on the challenge [7].

- Alice has to prove to Bob that she knows the solution of $g^x = y$.
- Alice choose random integer $m \in \mathbb{Z}_q^*$ and compute $g^m = t$.
- She choose a challenge $c = Hash\,(g, y, t)$.
- Alice also calculate $r = m - (x * c)$ and publishes the proof (t, r).
- Bob can verify $t = g^r y^c$.

2.1.1 Schnorr Protocol

It [8] is frequently used as a proof of knowledge in DLOG problems for cyclic group $< g >$ with generator g. Suppose that the prover has to show that she knows DLOG x that belongs to $y = g^x \, mod\, p \in \mathbb{Z}_p^*$. The protocol for non-interactive version works as:

- Prover chooses $r \in \mathbb{Z}_{p-1}$ to compute message $t = g^r \,(\mathrm{mod\ p})$ and apply hash function on t to get $c = Hash(t)(mod\ p)$ and calculate $s = r + cx$.
- After hashing, prover publishes (y, c, s).
- Anyone can verify if $c = Hash\ (\ g^s\ y^{-c})$.

For interactive version, we requires interaction of two parties rather then using cryptographic hash function. Authentication of graph isomorphism problem can be taken as an example and can be prove by using [9]:

- Prover randomly chooses $\mathrm{a} \in \{1, 2\}$, a random permutation ρ and generate another graph $I = \rho\,(G_a)$. Send I to verifier.
- Verifier randomly choose $\mathrm{b} \in \{1,2\}$, send b to prover and ask him for σ with $G_b = \sigma\,(I)$.
- Consider $\mathrm{a} = \mathrm{b}$: if graph I results from G_1 or G_2 and the verifier ask prover to map graph I on either of two graphs then the prover will send $\sigma = \rho^{-1}$ to receiver.
- Consider $\mathrm{a} = 1 \,\mathrm{and}\, \mathrm{b} = 2$: if graph I is derived from G_1 and verifier ask prover to map graph I to G_2 then $\sigma = \rho^{-1}$ o π.
- Consider $\mathrm{a} = 2 \,\mathrm{and}\, \mathrm{b} = 1$: if G_2 is used to obtain graph I and verifier asks prover to send a permutation that map graph I to G_1 then prover will send $\sigma = \rho^{-1}$ o π^{-1} to verifier.

2.1.2 Sigma Protocol

Σ-protocols are 3-move honest verifier zero-knowledge proofs based on commitment, challenge and a response. A binary relation $\mathcal{R}$ for an NP relation is a pair consist of (x, w) where x denotes the instance of statement and w is called witness. Based on $\mathcal{R}$ we define $\mathcal{L}_{\mathcal{R}} = \{ \mathrm{x} \mid \exists\, \mathrm{w} : (\mathrm{x}, \mathrm{w}) \in \mathcal{R} \}$ **[10].** Sigma protocol allows prover to manifest the verifier that statement $x \in \mathcal{L}_{\mathcal{R}}$ without revealing any knowledge about w given $(\mathrm{x}, \mathrm{w}) \in \mathcal{R}$.

Interactive version: Both prover and verifier knows the instance x but private parameter w is known to prover only such that $(\mathrm{x}, \mathrm{w}) \in \mathcal{R}$.

- Prover generates a message m and integer vectors z_1 and z_2 for $z_i \in \{0,1\}^{l_z(n)}$ i-e $P_\Sigma(x, w) \rightarrow (\mathrm{m}, z_1, z_2)$ where l_n(n) is a polyn omial upper bound for security parameter n.
- Verifier chooses $\mathrm{e} \in \{0,1\}^n$ randomly and send challenge to prover.
- Prover respond with $\mathrm{z} = \mathrm{e}z_1 + z_2$ to verifier.
- Verifier return $V_\Sigma(x, m, e, z) \rightarrow \{0,1\}$.
- Verifier will reject the response if any entry $z_i \notin \{0,1\}^{l_z(n)}$ or $e \notin \{0,1\}^n$.

3. zkSNARKs

We begin by considering an idea of Zcash using illustration below:

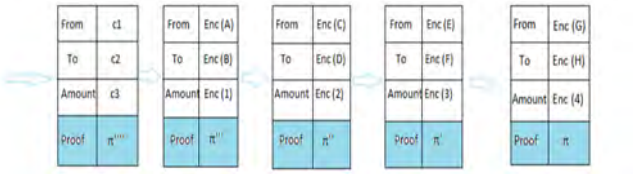


Figure 3.1: Encrypted transactions in Zcash

Suppose that Alice posted four transactions containing the ciphertexts c_1, c_2, c_3, c_4 and a proof π''' as shown in figure 3.1. Since all transactions are ciphertexts so Alice has to generate a cryptographic proof π'''' to prove that she is not double spending.

We start with the "proof" that the true statements have proofs but the false statements do not have. Then "non-interactive" providing the feature of creating & verifying the transaction without having any interaction between the two parties. "Zero knowledge" which reveal nothing else beyond the fact of statement that it is true. "Of knowledge" again which allows crypto that certain knowledge can be proven and lastly "succinct" states that proof is very brief and verification step is easy. Using all of these terms we introduce the notion of "Zero Knowledge Succinct Non-Interactive Argument of Knowledge". The strong privacy of Zcash is based on shielded transaction with are encrypted and the validation is proof by using the zk-SNARKs.

More precisely, we define an NP language as $\mathcal{L}$ [6], a non-deterministic arithmetic circuit C for an instance x. zk-SNARKs is use to prove and verify the existance of instance x in $\mathcal{L}$. Beside considering an input circuit C, a trusted setup is also needed which provides proving key pk and a verifying key vk using common reference string. Proving key enables a prover to create a proof π such that $x \in \mathcal{L}$ and vk is use to verify the proof π without making any interaction. Succinct tells that π should be verified in short time. Construction of zk-SNARKs involves the following steps:

3.1 Homomorphic encryption: For a variable x**,** [11] define homomorphic encryption $E(x)$ as a function which satisfy the following conditions:

- If $x \neq y \Rightarrow E(x) \neq E(y)$.
- It is very hard to find number x from its encrypted form $E(x)$.
- $E(x+y)$ can be computed from $E(x)$ and $E(y)$.

3.1.1 Constructing homomorphic encryption in $\mathbb{Z}_p^*$:

Construction of homomorphic encryption requires finite groups, finding the solution of discrete logarithm is still unknown in finite fields. Suppose that $\mathbb{Z}_p^*$ is obtained by applying multiplication operator over the prime numbers $\{1,2,\ldots,p-1\}$ and using a multiplication technique $ab = c\,(mod\,p)$ such that the result also lie inside $\{1,2,\ldots,p-1\}$ then;

- All elements of a cyclic group $\mathbb{Z}_p^*$ consisting of prime p and generator g can be written as $E(x) = g^x$.
- Follows from the third property of homomorphic encryption [11]:

  $E(x+y) = g^{(x+y)mod(p-1)} = g^x g^y = \mathrm{E(x)E(y)}$
- $E(x) = g^x$ can be use for linear combinations:

  $E(ax+by) = (g^x)^a (g^y)^b = E(x)^a E(y)^b$

3.2 Hidden evaluation of polynomials

Let $\mathbb{F}_p$ be a finite field of prime numbers consist of elements $\{0,1,\ldots,p-1\}$. We define polynomial P as $P(x) = a_0 + a_1 x + \cdots + a_m x^m$ of degree m. It can be very difficult to solve the polynomial for x since degree of $P(x)$ can take a very large number. Replace x with s and define P at point $s \in \mathbb{F}_p$ $P(s) = a_0 + a_1 s + \cdots + a_m s^m$.

- Alice has polynomial $P(x) = \sum_{i=0}^{d} a_i x^i \in \mathbb{F}_p$.
- Bob has a point $s \in \mathbb{F}_p$ and he wants to know $E(P(s))$ where $P(s) = \sum_{i=0}^{d} a_i s^i$.
- Bob send hidings $E(1), E(s), E(s^2), \ldots, E(s^d)$ instead of s to Alice.
- Alice compute $E\big(P(s)\big) = \prod_{i=0}^{d} E(s^i)^{p_i} = g^{p_i s^i}$ from hiding send by Bob and send $E(P(s))$ to Bob.

For non-interactive version, instead of sending $(s^0, s^1, \ldots, s^d)$, Bob will publish $(E(s^0), E(s^1), \ldots, E(s^d))$ on to the CRS where s is a secret parameter that needs to be destroyed.

3.3 Common Reference String

Non-Interactive proofs requires both sender and receiver to have a mutual access to string of random numbers encoded in common reference string. In Zcash, initial setup generates CRS called the public parameters of system, provides non-interactive and short proofs to publish on blockchain. Initial parameters are randomly chosen using pseudorandom generator based on the secret sharing scheme. The initial parameters used to generate CRS needs to be securely destroyed, otherwise CRS could be deceived. Select a random secret s of degree d and a random α so the encryption of secret s is $E(s^i) = g^{s^i}$ for $i = 0,1,\ldots,d$. Encryption of s is available to the prover as $E(s^0), E(s^1), \ldots, E(s^d)$. Based on the cryptographic pairing, we can setup the secure public parameters. Assume that the secret s and α-shifts is constructed by a single trusted party. Initial parameters should be destroyed after the encryption of α and all powers of s w.r.t α. Mainly, CRS is divided into two parts:

- proving key: $(g^{s^i}, g^{\alpha s^i})$
- verification key: $(g^{Z_m(x)}, g^{\alpha})$

Verifier can check the polynomials by using the verification key received by the encrypted polynomial evaluations g^P, g^H and $g^{p'=\alpha P}$. Now the verifier will run the check in two steps:

- According to [12],check if $P = Z_m(x).H$ as: $e(g^P, g^1) = e(g^{Z_m(x)}, g^H) \Leftrightarrow e(g, g)^P = e(g, g)^{Z_m(x).H}$
- Follows from Extended knowledge of coefficient test and assumption; check:

$$e(g^P, g^\alpha) = e(g^\alpha, g^P) = e(g^{\alpha P}, g^1) = e(g^{p'}, g)$$

Since the construction of CRS requires multi party involvement, therefore everyone has to trust that every party has deleted the initial parameters α, *s*.

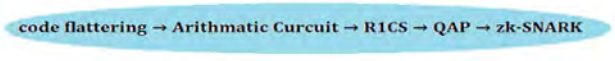
3.4 Computational to Polynomial

zk-SNARKs cannot be applied directly on computational problem, we need to convert problem to a language of polynomials called Quadratic Arithmetic Program (QAP). We work in polynomial times because it is difficult to lie in polynomial time. For example, if you want to tell the story of execution in polynomials, the true and false statements will differ a lot. Conviencing verifier that solution to the equation $x^2 + 49 = 0$ is known to prover without revealing x we proceed as follow:

$def\ qeval(x)$:

$y = x\ **\ 2\ +\ 49$

At the end, prover shows that she has correctly executed program. But there is no verification method to check if the prover has run a correct program and we have no idea how to observe the y without knowing x. We consider the following steps:

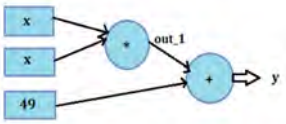


In the initial step, convert the original code into a sequence of statements called the code flattering as:

$x = y$

$x = y\ (op)\ z$

Where op denotes an arithmetic operation and y, z are variables. Further, consider these statements as gates of an arithmetic circuit:



The flattened code is:

$def\ qeval(x)$:

$out\_1 = x * x$

$y = out\_1 + 49$

Here, Prover has to prove that she knows the consistent assignment of the variable solving $x^2 + 49 = 0$. We start with Rank One Constraint System to check if all the steps are performed accurately. It is a list of triplets of vectors $\langle \vec{a_i}, \vec{b_i}, \vec{c_i} \rangle$ and its solution is vector $\vec{s}$, such that:

$$\langle \vec{a_i}, \vec{s} \rangle * \langle \vec{b_i}, \vec{s} \rangle - \langle \vec{c_i}, \vec{s} \rangle = 0 \qquad (3.1)$$

Each component of this vector will be one variable. Where vector $\vec{s}$ tells the state of variables being considered in the program. More generally, for $s = (one, x, out\_1, y)^T$ we can write 3.1 as:

$$cs = as * bs \qquad (3.2)$$

Second statement $y = out\_1 + 49$ can be written as:

$$\begin{pmatrix}0\\0\\0\\1\end{pmatrix}\begin{pmatrix}one\\x\\out\_1\\y\end{pmatrix}=\begin{pmatrix}49\\0\\1\\0\end{pmatrix}\begin{pmatrix}one\\x\\out\_1\\y\end{pmatrix} * \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\begin{pmatrix}one\\x\\out\_1\\y\end{pmatrix}$$

Logic of the code has been transferred to the triplets of constraints vector $\vec{a_i}, \vec{b_i}, \vec{c_i} \in \mathbb{F}_p^n$ for n number of variables. Thus, every statement can be represent by using the three vectors $\vec{a_i}, \vec{b_i}\ and\ \vec{c_i}$. In general, there are n variables which define the length of the vector and m statements. To complete the transformation to QAP, we switch from vectorial to polynomial representation and define polynomials as:

$A_i(x),\ B_i(x),\ C_i(x)$ for all $i\ \in \{1, n\}$ [13][14]

QAP with m statements and n variables is represented by m triples of vector $\vec{a_i}, \vec{b_i}, \vec{c_i} \in \mathbb{F}_p^n$.

$$R_1 = (a_1, b_1, c_1), \dots, R_m = (a_m, b_m, c_m)$$

Transformation is done by using the langrange interpolation which is use to find the polynomial from the set of points (x, y) that passes through all these points, since we have n variables and three constraint vectors therefore, we obtain $3n$ polynomial:

$$A_1(x), A_2(x), \dots, A_n(x),$$
$$B_1(x), B_2(x), \dots, B_n(x),$$
$$C_1(x), C_2(x), \dots, C_n(x)\ [13][14]$$

Express polynomial as $P_s(x) = A_s(x) * B_s(x) - C_s(x)$. Thus, we have:

$$P_s(x) = \begin{pmatrix}A_1(x)\\.\\.\\.\\A_n(x)\end{pmatrix}\begin{pmatrix}s_1\\.\\.\\.\\s_n\end{pmatrix} * \begin{pmatrix}B_1(x)\\.\\.\\.\\B_n(x)\end{pmatrix}\begin{pmatrix}s_1\\.\\.\\.\\s_n\end{pmatrix} - \begin{pmatrix}C_1(x)\\.\\.\\.\\C_n(x)\end{pmatrix}\begin{pmatrix}s_1\\.\\.\\.\\s_n\end{pmatrix}$$

for $x = 1, \dots, m$. Subsequently;

$$P_s(x) = A_s(x) * B_s(x) - C_s(x) = H_s(x) * Z_m(x)$$

The value of polynomial at target polynomial $Z_m(x)$ is 0 except those included at the target point. The degree of $H_s(x) \le (m\text{-}2)$, $P_s(x) \le 2(m-1)$ and $Z_m(x) = (x-1)(x-2)\dots(x-m)$ [11]. The purpose of using $H_s(x)\ and\ Z_m(x)$ is that the prover has to show that she knows legal assignment s by solving quadratic arithmetic program hence, polynomial $A_s(x) * B_s(x) - C_s(x) = 0$ iff target polynomial $Z_m(x) | P_s(x)$:

$\exists\ H_s(x) : A_s(x) * B_s(x) - C_s(x) = H_s(x) * Z_m(x)$ [13]

$$H_s(x) = \frac{A_s(x) * B_s(x) - C_s(x)}{Z_m(x)} = \frac{P_s(x)}{Z_m(x)}$$

Hence; QAP is a 4-tuple: $(\vec{a_i}, \vec{b_i}, \vec{c_i}, Z)$ with solution:

$$s = \begin{pmatrix}s_1\\.\\.\\.\\s_n\end{pmatrix}$$

3.5 Pinocchio protocol

Bob need to know whether Alice has a valid assignment or not, Bob will use method of Pinocchio protocol introduced by [11] to test the validation:

- Bob choose random point $k \in \mathbb{F}_p$ and send $E(k)$ to Alice.
- Alice choose polynomials $A_s(k), B_s(k), C_s(k), H_s(k)$ and send the corresponding hidings $E(H_s(k)), E(A_s(k)), E(B_s(k)), E(C_s(k))$.
- Bob will check if equality $E(A_s(\mathrm{k}) * B_s(\mathrm{k}) - C_s(\mathrm{k})) = E(H_s(\mathrm{k}) * Z_m(\mathrm{k}))$ holds.

If the equality holds then Bob will accept that Alice knows the valid assignments. Pinocchio protocol follows from the method of blind evaluation of polynomials. Therefore, further calculations can be solved by using hidden evaluation of polynomials.

3.7 Elliptic Curve Pairing

In Pinocchio protocol Bob need to evaluate $E(H_s(\mathrm{k}) * Z_m(\mathrm{k}))$ based on the individual hidings $E(H_s(\mathrm{k}))$ and $E(Z_m(\mathrm{k}))$ but In general,

$$E(H_s(\mathrm{k}) * Z_m(\mathrm{k})) \neq E(H_s(\mathrm{k}))E(Z_m(\mathrm{k}))$$ [11]

We define a pairing over an elliptic curve to check the quadratic constraints of the system. [15] defined an elliptic curve E over $\mathbb{F}_p$ satisfy the following equation:

$$y^2 = x^3 + ax + b \quad (3.4)$$

Group of elliptic curve consists of addition operation, point at infinity O and set of pairs (x, y) over F_{p^k}. The elliptic curve security relies on solving hard DLOG problem therefore elliptic curve cryptography provides the same security using secret key but with a very small parameter key size as compare to the other cryptosystems. The fastest algorithm to solve the DLOG in elliptic curve requires exponential time. For instance; consider an elliptic curve below.

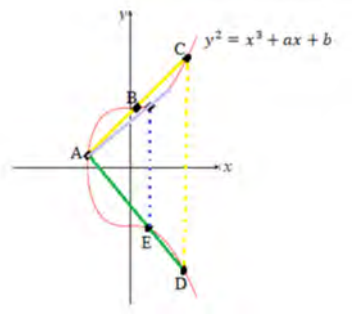


Figure 3.3: Elliptic Curve Cryptography

E over F_p from [15] showed in figure [3.3] starts with generator point $A \in \mathrm{E}/F_{p^k}$ and another point $\mathrm{Z} \in \mathrm{E}/F_{p^k}$. For an elliptic curve E, $ECDLOG$ is to find an integer n for given element A and another element Z such that $n.A = Z$. We prefer an elliptic curve over the other public key cryptosystems because it provides high security with a small bit size then that of RSA which rely on large number factorization.

As proposed by [16], prover has to show the verifier that she knows such n that results $Z = n.A$ without revealing n. The algorithm work as following:

- Prover calculate another $B = r.A$ after choosing a random $\mathrm{r} \in F_{p^k}$ and send to verifier.
- Verifier respond with binary choice $b \in \{1,2\}$.
- Prover will send r if $b = 1$ otherwise $m = n + r\ (mod\ p)$.
- Verifier will check $B = r.A$, for $b = 1$ otherwise $m.A = (n + r)(A) = Z + B$.

The algorithm will keep on repeating unless the verifier get to know that prover knows n. The number of iterations will be $\frac{1}{2^k}$. Now if the prover is a fake then she can generate a random message m and compute $m.A - Z = B$ and send B for verification but if in other case she need instance of ECDLOG r to generate B therefore, prover will be able to perform this case only if she is honest.

Definition 3.7.2 [17] defined Pairing as a bilinear and non-degenerate mapping over an elliptic curve:

$$\mathrm{e}: G_1 * G_2 \rightarrow \mathrm{G} \in F_{p^k}$$

where $G_1 \in \mathbb{F}_p$ denotes subgraph generated by a rational point on elliptic curve with order r. $G_2 \in \mathbb{F}_p$ is another subgraph obtained by a twisted curve over an elliptic curve. G define the subgroup of the field that belongs to non-zero elements of a finite field of p^k where k is an embedded degree or the minimum integer such that $\frac{(p^k-1)}{r}$. Based on information provided by pairing with elliptic curve, let G_1 and G_2 be the cyclic subgroup over F_{p^k} and G be the subgroup of $\mathbb{F}^*_{p^k}$ with order of $|G_1| = |G_2| = |G| = r$. For fix generators $g_1 \in G_1, g_2 \in G_2, g \in G$ and given hidings $E_1(\mathrm{x}) = x.g_1, E_2(\mathrm{x}) = x.g_2, E(x) = x.g$ we have:

$$E(xy) = Tate(E_1(x), E_2(y))$$

4. Zcash Transactions

For an arithmetic circuit C, [17] defined zk-SNARK as a triple of polynomial time algorithm:

- $KeyGen(1^\lambda, C) \rightarrow (a_{pk}, a_{vk})$

  Given the λ security parameter and circuit, KeyGen algorithm will generate the proving key and a verifying key called public parameter. These pp are accessible to every participant and can be use over and over to prove and verify that the statement x belongs to $\mathcal{L}_c$.

- $Prove(a_{pk}, x, w) \rightarrow \pi$:

  Prover will create a proof π and binary relationship $(x, w) \in \mathcal{R}$ such that instance $x \in \mathcal{L}_c$.

- $Verify(a_{vk}, x, \pi) \rightarrow b$:

  For the given statement x, vk along with the $non - interactive$ proof π, the receiver will return bit $b = 1$ if $x \in \mathcal{L}_c$.

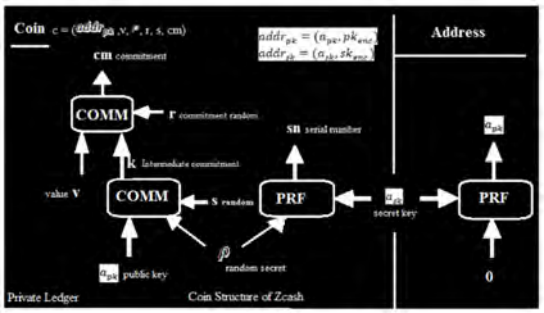


Figure 4.1: Coin Structure of Zcash

Setup: For figure [4.1], public parameters are created by the trusted setup, based on security parameters λ which is use to create the pp we have:

$$pp = \{pk_{POUR}, vk_{POUR}, pp_{enc}, pp_{sig}\}$$

Coin c consist of coin commitment cm, a string available inside the Merkle tree that makes commitment to the serial number sn of the coin when the coin is minted, value v which is the denomination of the coin ranges from 0 to v_{max}. Serial number sn is singular string devoted to the coin to avoid double spending and is hidden. Trapdoors r and s are to enhance the security. $addr_{pk}$ is use by others participats to send the transactions to the user. $addr_{sk}$ is use to obtain the payments sent to the address public key. Random secret ρ is use to obtained the sn and cm. Each user can create any number of address key pair (a_{pk}, a_{sk}). Using the pp we obtain the address key pair $(addr_{pk}= (a_{pk}, pk_{enc}), addr_{sk}= (a_{sk}, sk_{enc}))$ where $(pk_{enc}, sk_{enc}) = \kappa_{enc}(pp_{enc})$. Beside transparent transactions like in Bitcoin, Zcash mainly deals with two type of transactions:

4.1 Mint Transacton:

Mint transaction itself is a cryptographic commitment to the new coin c. Let say, Alice spend v BTC to create a value v coin with coin commitment cm.

Input: $pp, v, addr_{pk}$ to send the transaction.

Output: $c = (addr_{pk}, v, \rho, r, s, cm)$

for $cm = COMM_r(v \parallel k)$ and $k = COMM_s(a_{pk} \parallel \rho)$ with mint transaction $tX_{MINT} = (cm, v, k, r)$.

Verification: Since the unique sn of coin c is hidden so we need to validate the transaction:

Input: $pp, tX_{MINT}, Merkle\ tree$

Output: Verifier will return bit *b = 1* if:

$$\text{cm} = cm' = COMM_r(v \parallel k).$$

4.2 Pour Transaction

This algorithm [6] enables the user to split and merge the coins. User can use this method to spend the coins, to send the coins, can make change etc. It also contains the information string which tells that who is the recipient of public value and when the transaction was made.

Input: $pp, Merkle\ root\ rt$, two old Zcash coins to be consumed c_1^{old},c_2^{old} with address secret keys $addr_{sk,i}^{old}$, $path_i$ from c_i^{old}, desired input value of new ZEC coin v_i^{new}, destination $addr_{pk,i}^{new}$ for each $i \in \{1,2\}$ and public value v_{pub} that lies between $0 \leq v(c) \leq v_{max}$.

Algorithm: Alice want to consume old coins $c_i^{old} = (addr_{pk,i}^{old}, v_i^{old}, \rho_i^{old}, r_i^{old}, s_i^{old}, cm_i^{old})$ in order to create the new coins c_1^{new} and c_2^{new}. The algorithm works as follow:

- Parse c_i^{old} and yields new coins $c_i^{new} = (addr_{pk,i}^{new}, v_i^{new}, \rho_i^{new}, r_i^{new}, s_i^{new}, cm_i^{new})$ for $i \in \{1,2\}$ with $addr_{sk,i}^{old} = (a_{sk,i}^{old}, sk_{enc,i}^{old})$ and $addr_{pk,i}^{new} = (a_{pk,i}^{new}, pk_{enc,i}^{new})$.
- $tX_{POUR} = (\text{rt}, sn_1^{old}, sn_2^{old}, cm_1^{new}, cm_2^{new}, v_{pub}, \text{info}, pk_{sig}, \pi_{POUR}, h_1, h_2, C_1, C_2, \sigma)$.

Where $C_i = \mathcal{E}_{enc}(pk_{enc,i}^{new}, (v_i^{new}, \rho_i^{new}, r_i^{new}, s_i^{new}))$ is ciphertext computed by Alice to transfer the ownership of coin to Bob. $h_i = PRF_{a_{sk,i}^{old}}^{pk}(i \parallel h_{sig})$ for $i \in \{1,2\}$ works as a MACs that associates h_{sig} with address secret keys using pseudo random function. $\pi_{POUR} = Prove(pk_{POUR}, x, w)$ is prove from Alice that computations are done correctly using pk of pour transaction computed at the setup phase on instance x and witness w. Alice use the $sk_{sig}$to sign each value that is connected with the pour operation denoted by message $m = (x, \pi_{POUR}, info, C_i)$ to obtain sigma signature $\sigma = \mathcal{S}_{sig}(sk_{sig}, m)$. In order to deploy the efficiency, Zcash use zk-SNARKs and an additional requirement for avoiding the malleability attack is by using digital signature. Zcash compute the MACs to combine the secret keys with signing key, then update the instance x by adding signature verification key and MACs. Finally, each transaction is signed. Also, Bob can decrypt this message using his encrypted secret key by scanning the tX_{POUR} on public ledger.

Transaction verification:

- sn_1^{old}, sn_2^{old} do not already in ledger otherwise $b = 0$.
- If $sn_1^{old} \neq sn_2^{old}$ then $b = 1$.
- Compute $b = \mathcal{V}_{sig}(pk_{sig}, m, \sigma)$. From the given digital signature based public key, message *m* and the σ signature, if sigma is valid signature for m then return $b = 1$.
- Finally, for the given verification key derived from pp, instance x and proof π verify:

$$b' = verify(vk_{POUR}, x, \pi_{POUR}) = \begin{cases} 1, & x \in \mathcal{L}_c \\ 0, & otherwise. \end{cases}$$

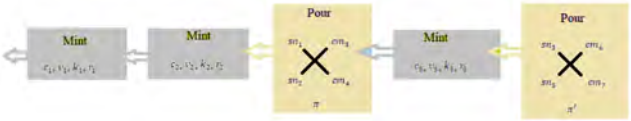


Figure 4.2: Pour transaction blockchain View

***zk-SNARK proof*:** Alice consume two input coins c_1^{old},c_2^{old} with serial numbers s_1^{old}, s_2^{old} in order to create two output coins c_1^{new}, c_2^{new} with cm_1^{new}, cm_2^{new}

and public output v_{pub}. For an $NP\ statement, zk-SNARK\ proof\ \pi$ that Alice know secret is:

Given the Merkle tree root *rt* containing *cm* of all minted coins, serial number *sn* of old coins and cm_1^{new}, cm_2^{new} of new coins. Alice know $c_1^{old}, c_2^{old}, c_1^{new}, c_2^{new}$ and the secret key a_{sk}^{old} such that:

- Coins $c_1^{old}, c_2^{old}, c_1^{new}, c_2^{new}$ are well formed.
- $a_{pk,i}^{old} = PRF_{a_{sk,i}^{old}}^{addr}(0)$.
- Revealed s_1^{old}, s_2^{old} are of old coins and are calculated correctly as $s_i^{old} = PRF_{a_{sk,i}^{old}}(\rho_i^{old})$.
- cm_i^{old} of c_i^{old} appears on CRH-based Merkle tree with rt.
- Revealed cm_1^{new}, cm_2^{new} are of c_1^{new}, c_2^{new}.
- $v_1^{new} + v_2^{new} + v_{pub} = v^{old}$.

If all the conditions stated above hold then witness w is valid for instance x.

Fetching coins: This algorithm provides the list of all those coins whose sn do not appear already on the blockchain $\mathcal{L}$.

Input: address key pair:$(addr_{sk}, addr_{pk})$ and current ledger $\mathcal{L}$.

Output: receive coins if:

- $sn_i = PRF_{a_{sk,i}^{sn}}(\rho_i)$ does not appear in ledger.
- $cm_i^{new} = COMM_{r_i^{new}}(v_i^{new} \parallel k_i^{new})$.

Transaction Anonymity:

- If Alice do not know secret key $addr_{sk}$ then based on this knowledge she cannot spend the coin. The core security lies in the secret key since $addr_{pk}$ generated by $addr_{sk}$ is required in order to create an address of coin. Given secret key $addr_{sk}$ is also apply in witness w which is requisite in creating a zk−SNARK proof π so if Alice do not know the secret then she cannot create a proof π.

5. Future Considerations

Besides having the strong mathematical algorithms used in the development of Zcash there are still some flaws in term of privacy, decentralization and efficiency that can be improve in Zcash.

- The factorization and DLOG problems on EC are at risk by the quantum computer. It is very challenging to design quantum resistant anonymous algorithms that can protect user's privacy and run anonymous authentication in quantum systems. Currently, Zcash rely on one-way-hash where user creates the digital signature using these trapdoors in order to validate the transaction. Quantum computers can be use to break these cryptographic codes as trapdoors rely on large prime numbers factorization. Based on this knowledge, we can consider two main features "trapdoors, digital signature" in Zcash. Either replace the trapdoor with other unbreakable system or upgrade to trapdoor free structure. Secondly, digital signature can be exploit by using the shor's algorithm which also deploy the idea of Fourier transform for prime number factorization based on quantum computers. One solution to secure the digital signature can be using the same idea deployed by the Monero which works using ring signature.
- Privacy can be a risk to governments as well. Anyone can use this channel for money laundering. Still considering the dark aspect of transparency everyone wants to move to privacy, a possible solution to stop using privacy for illicit acts can be by adding a security check in the pipeline connecting the two parties.
- Another possibility to raise the privacy can be by expiration of particular secret key after a certain time limit.
- The user need to pay more when select a private transaction, 20% reward of it goes to the founder which is expensive. It is required to make improvements in the shielded transaction to make it more efficient beside consuming less memory space and enhancing the security.
- Comparison between Zcash, Monero and Ethereum:

Features	**Zcash**	**Monero**	**Ethereum**
Type	Digital currency	Digital currency	DC/Blockchain platform
Supplies	21 Million	18.4 Million	18 Million annual
Block confirmation time	2.5 min	2 min	15 sec
Block size	2 MB	Dynamic	1 MB
Hashing alogorithm	Zk-SNARKs	CryptoNight	Ethash
Rank	28	10	2
Transaction per sec.	6-25	>1700	Approx. 4

Table 4.1: Zcash Comparison with other cryptocurrencies [18][6][19]

- Promotion in the decentralization is still required by allowing multiple parties to participate in Zcash development. As mentioned in CRS that initial parameters are completely destroyed but there is no way to verify it. There should be a verification method so the user can invest in Zcash without any doubt.
- Scalability need to enhance in Zcash, it should be in the access of every person. Also, it takes 40 seconds to create the transaction. Zcash need to develop a network that can handle several transactions in seconds but currently private transactions slow down the process. To make it possible, main network can be split into subnets so that

only the relevant network work when requires instead of running the whole setup. New block creation time in Zcash is 2.5 minutes while that is Monero is 2 minute and the block size is 2MB in Zcash which is high in case of private transactions. Monero takes the advantage here in term of scalability due to its dynamic block size.

Acknowledgement

This present survey is based on the author's Master's thesis. The author expresses her gratitude to Professor Klaus Dohmen for his kind supervision.

References

[1] B. S. d. Albuquerque and M. d. C. Callado, “Understanding Bitcoins: Facts and Questions,” Revista Brasileira de Economia, vol. 69, pp. 3--16, 2015.

[2] G. Kappos, H. Yousaf, M. Maller and S. Meiklejohn, An empirical analysis of anonymity in zcash, 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 463--477.

[3] S. D. Galbraith, Mathematics of public key cryptography, Cambridge University Press, 2012.

[4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Manubot, 2019.

[5] K. Balasubramanian and M. Rajakani, Algorithmic strategies for solving complex problems in cryptography, IGI Global, 2017.

[6] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer and M. Virza, “Zerocash: Decentralized anonymous payments from bitcoin,” in IEEE Symposium on Security and Privacy, San Jose, CA, 2014.

[7] T. Koens, C. Ramaekers and C. v. Wijk, “Efficient zero-knowledge range proofs in ethereum,” ING, 2018.

[8] R. Cramer, I. Damgård and B. Schoenmakers, “Proofs of partial knowledge and simplified design of witness hiding protocols,” in Annual International Cryptology Conference, 1994.

[9] E. Ayeh, “An Investigation Into Graph Isomorphism Based Zero-knowledge Proofs,” PhD thesis, 2009.

[10] P. Chaidos and J. Groth, “Making sigma-protocols non-interactive without random oracles,” in IACR International Workshop on Public Key Cryptography},, 2015.

[11] A. Gabizon, What are zk-SNARKs?, 2017.

[12] M. Petkus, “Why and How zk-SNARK Works,” arXiv preprint arXiv:1906.07221, 2019.

[13] V. Buterin, Zk-SNARKs: Under the Hood, Medium, 2017.

[14] S. D. Valentin Ganev, Introduction to zk-SNARKs, 2018.

[15] C. Paar and J. Pelzl, Understanding cryptography: a textbook for students and practitioners, Springer Science & Business Media, 2009.

[16] I. Chatzigiannakis, A. Pyrgelis, P. G. Spirakis and Y. C. Stamatiou, “Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices,” in IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011.

[17] E. Ben-Sasson, A. Chiesa, E. Tromer and M. Virza, “Scalable zero knowledge via cycles of elliptic curves,” Algorithmica, vol. 79, pp. 1102--1160, 2017.

[18] KOE, K. M. ALONSO and S. NOETHER, Zero to Monero: Second Edition, April 4, 2020.

[19] D. Vujičić, D. Jagodić and S. Ranđić, “Blockchain technology, bitcoin, and Ethereum: A brief overview,” in 17th international symposium infoteh-jahorina (infoteh), 2018.

ANALYSIS OF LARGE-SCALE DECISION MAKING TOOLS USING A DECENTRALIZED ARCHITECTURE TO GOVERN COMMON POOL RESOURCES

Tina Marquardt, Norbert Pohlmann
Institute for Internet Security, Westphalian University of Applied Sciences, Neidenburger Straße 43, D-45897, Gelsenkirchen, Germany

This paper analyses the status quo of large-scale decision making combined with the possibility of blockchain as an underlying decentralized architecture to govern common pool resources in a collective manner and evaluates them according to their requirements and features (technical and non-technical). Due to an increasing trend in the distribution of knowledge and an increasing amount of information, the combination of these decentralized technologies and approaches, can not only be beneficial for consortial governance using blockchain but can also help communities to govern common goods and resources. Blockchain and its trust-enhancing properties can potenitally be a catalysator for more collaborative behavior among participants and may lead to new insights about collective action and CPRs.

1. Introduction

Blockchain is mostly known as the technical infrastructure for cryptocurrencies and the financial industry, however, there are more applications which can have a sustainable and ethical application as well as an impact, like large-scale and collective decision making in combination with collective action theory and common pool resources (CPRs) [1]. Due to an increasing trend in the distribution of knowledge and an increasing amount of information, the combination of these decentralized technologies and approaches, can not only be beneficial for consortial governance using blockchain but can also help communities to govern common goods and resources [1]. Thomas W. Malone explores on the other side the many applications of information technology and human-machine interaction and how they enable us to decide together in order to solve complex problems and form smarter communities, markets, companies and eventually societies. Therefore, it provides the opportunity to overcome outdated inefficient economic models [2]. This paper analyses the status quo of large-scale decision making combined with the possibility of blockchain as an underlying decentralized architecture and evaluates them according to their requirements and features (technical and non-technical), potential impact and ethical implications. Further research questions and suggestions for further implementation are elaborated as a summary and final conclusion of the analysis. Besides, the well-known applications of blockchain, decentralized technologies can potentially be an engine for large-scale and collective decision-making. Hence, technical features and properties of blockchain have the potential to accelerate collective action approaches [1] that were e.g. elaborated by Elinor Ostrom in 1990 in her book about governing common goods and collective action [3]. Features of blockchain can enable new forms of decision-making and hence new forms of organizations, communities and societies. The decentralized set up and distributed storage of the database can enable trust and increase collaboration, due to transparency and immutability of the data [4]. At the same time, decentralized identities can provide solutions for compliance with data privacy laws and standards [5].The combination of these approaches and technologies may prevent depletion and conflicts over Common Pool Resources (CPRs), possibly public goods, as well, and therefore lead to self-governed, more sustainable and ethical decisions for communities, societies, including the environment [6].

2. Methodology and Limitations

Within this context the paper aims to analyze to which extent large-scale decision-making approaches are researched to govern common goods as suggested by Liv Ostrom via technological solutions such as Blockchain. Furthermore, we analyze if there are common features or requirements that can enable collective action and decision-making. Hence, our paper aims to identify the combined application of collective and large-scale decision-making with blockchain as a possible infrastructure to govern common resources in a more efficient, sustainable and moral manner. Therefore, we performed an interdisciplinary qualitative literature survey, based on surveys of each discipline, to understand if Ostrom‘s approach is already combined with these emerging topics. The literature sources examined for this survey are widely varied and include databases, research papers, book chapters, journal papers, web resources, conference papers, and whitepapers published by various blockchain, organizations, companies and forums.

To lower the complexity of the paper and increase comprehensibility, the following section provides a brief description and context of the terms used and also points out limitations of the paper and methodology.

Firstly, the terms public goods and common pool resources are distinguished. Goods and services are often classified based on two criteria: the cost and possibility to exclude free-riders (people who consume but do not pay) and the level of subtractability of a good or service (how rivalrous the good or service is) [3]

Ostrom concludes that public goods are usually underproduced and common-pool resources are usually both underproduced and over consumed. for example fish in the ocean. Reasons are missing compensations and free-riding. The initial focus of the paper is on the well researched common-pool resources (CPRs). Given the similarities between many CPRs and public

good problems, insights may be applicable to some extent to even small-scale public goods. Example of CPRs are the ocean, fishery and forestry [3] Secondly, the context of collective and large-scale decision-making is provided. Due to the nature of the problem with CPRs the scope of the paper excludes classical multi-person or Group-Decision Making (GDM) approaches for smaller groups. In current Large-Scale Decision Making (LSDM) Studies, there is no common definition to outline the key characteristics of LSDM. However, one of the most important ones can be argued to be the higher complexity of decision-making processes in the scenarios of LSDM. Moreover, these kinds of events and decisions are important for certain societies, communities and citizens living in cities. Additionally, they should include as many affected participants and stakeholders as possible. These parties are participating directly and not via a representative, as it is the case in most classical group-decision making approaches [1]. Due to the complexity and many levels of different stakeholders, LSDM can be seen as a possible approach to collectively govern CPRs. Thirdly, the taxonomy of distributed ledger technology and blockchain are explained to provide a common understanding of the terms used. Since the introduction of Bitcoin in 2008 by Satoshi Nakamoto, Blockchain Technology has been well known for its application in the financial industries and in cryptocurrency trading, it has moved far beyond this and is researched, discussed and applied by many different fields within academic discourse. industries and public services [7]. The diversity of research and development creates potential for cross fertilisation of ideas and creativity, but also the risk of fragmentation due to no common taxonomy of blockchain technologies and distributed ledger technologie. Reasons are consistent progress in engineering large and complex blockchain systems at a rapid pace [8]. In order to decrease the complexity further and since there is no uniformed definition at this point, blockchain is understood to be a part of distributed ledger technologies, which also include other technologies like Iota, which uses a directed acyclic graph (DAG or Tangle), instead of a linear blockchain data structure. In the context of this paper, these technologies are still accounted as a blockchain, even though some researchers and practitioners may refer to them as distributed ledger technologies only [8] [9]. Consequently, the following limitations of this analysis need to be pointed out.

Due to the novelty of the topics and the interdisciplinary character of this paper very strict definitions are not applied within this paper. The definitions are rather to be understood as descriptions of phenomena which are currently arising and hence are not well defined and classified in literature by a standardized manner. Since, there is not a distinguished taxonomy for the mentioned disciplines a quantitative survey of the combined topics does not seem appropriate, since there has to be a discourse about the clear distinction of certain terminology. The focus on CPRs in this paper are due to the recognition and well-discussed approach by Ostrom in combination with collective action and hence decision-making. Her work was even rewarded with the Nobel Prize in 2009 [10]. However, she also pointed out that comparable data and experiments are still missing to test the theories to make better assumptions and models that can find different rules for various scenarios, goods and stakeholders [10]. One reason may be the interdisciplinarity and thus very different terms of approaches and language. Moreover, it has to be acknowledged that neither this paper nor Ostrom claims that collective action is the only way to govern public goods and CPRs, in certain circumstances it may be one possible way. Therefore, a more common taxonomy is needed in the social-ecological framework and now combined with new technological possibilities that can help in the long-term to do more empirical research, which can refine and improve these theories to find better economic models.

3. Collective Action Theory and Self-Governance of CPRs

Within recent years many researchers started to pay attention to the approach of commons again. It refers to the research on people working commonly in the pursuit of the common good and the development of collective forms of common goods production, distribution management and ownership [11] Within this paper one of the most fundamental theories by Elinor Ostrom serves as an example for collective action. Her work can be broadly categorized within the rational-choice tradition within the fields of politics and economics and describes a collective way of CPR management to avoid the tragedy of the commons, free-riding and move beyond privatisation and government regulations. She acknowledged that agents are dealing with incomplete information and cognitive limitations, but are still agents responding to incentives [10]. Hence, she developed new game theories based on her studies that allow cooperative behavior within a non-cooperative framework. Since, she calls out for new forms of economics further classification of her theory from a heuristic point of view is not provided. The main discovery of her empirical studies was that despite recognized economic theories communities indeed create and enforce rules against free-riding and even ensure long-term sustainability of communal properties. Her ‘design principles’ explain under what conditions this happens and when it fails [10]. Similarly, she aims to provide a framework that can guide decisions about when to rely on spontaneous processes of governance and when to rely on the external generation of rules [10].

3.1 Core requirements for Collective Action

In 1990 she published her work and the eight design principles to ensure sustainability in self-organized CPRs. Since the principles have been slightly adapted over the years (The Future of the Commons, Beyond Market Failure and Government Regulation, which was published in 2012.) In the following, the eight core principles of a socio-ecological systems

and design principles for CPRS are briefly described in order to understand how distributed technologies like blockchain and large-scale collective decision-making rules are able to represent these conditions to set rules as software and protocols. The eight underlying principles for self-organized governance systems are: 1) Boundaries to facilitate exclusion; 2) The importance of internal rules 3) The importance of locally adapted rules; 4) The importance of monitoring and enforcement; 5) Dispute Resolution 6) Interaction between system of rules 7) The presumption against centralized natural planning and 8) The role of the state in environmental resource management [10]. In 1990 she summarized them as the following: "*All efforts to organize collective action. whether by an external ruler, an entrepreneur. or a set of principals who wish to gain collective benefit. must address a common set of problems. These have to do with coping with free-riding, solving commitment problems, arranging for the supply of new institutions, and monitoring individual compliance with sets of rules. A study that focuses on how individuals avoid free-riding, achieve high levels of commitment, arrange for new institutions, and monitor conformity to a set of rules in CPR environments should contribute to an understanding of how governing the commons dividuals address these crucial problems in some other settings as well"* [3]. Besides these eight principles she defined other variables and criteria that influence successful collective action such the number of participants involved, whether benefits are subtractive or fully shared, the heterogeneity of participants, face-to-face communication, the shape of the production function (factors independent from repetition) and other variables that depend on repetition. These include information about past actions, how individuals are linked and whether individuals can enter and exit voluntarily*. Moreover, she points out that the core relationships of reputation, trust and reciprocity affect cooperation. Systems where people are encouraged to cooperate seem to have a higher social outcome than the non-cooperative game theory commonly represented with the Nash-Equilibrium. She *concludes that this relationship is so fundamental that at the core of an evolving theoretical explanation of a successful or unsuccessful collective action are the links between the trust that one participant (Pi) has in others (Pj….Pn) involved in a collective action situation, the investment others make in the trustworthy reputations, and the probability of all participants using reciprocity norms* [12]. Since, she points out trust and reputation as the fundamental successor of collective action, decentralized architectures like blockchain that promise digital trust are analyzed as a possible underlying infrastructure. The possibilities of many participants to decide together via digital (collective) decision-making approaches is reviewed in the subsequent part.

4. Large-Scale Decision-making and Information Technology

Since Ostrom published her research on collective action a lot of progress has been made in communication and information technology. E-Democracy, citizen participation and social media are only a few examples of possible ways to connect and decide together these days [1]. Large-Scale Decision-Making in this context are tools which enable stakeholders to decide via computer-based systems together on different topics. In the following their application to CPRs and usage of Blockchain Technology is evaluated. Firstly, an overview of the current status quo of this discipline is presented in order to understand potential, opportunities and progress within research and real-world applications.

In their recent study Ding et al. point out the tremendous increase of Large Scale Decision-Making within the recent decade, with 2014 and 2018 as a turning point. In 2018, a high impact factor was found for the following journals and it's related publications: IEEE Transactions on Fuzzy Systems, IEEE Transactions on Industrial Informatics, IEEE Transactions on Systems, Man and Cybernetic Systems, Man and Cybernetic Systems, Information Fusion. This also proves the strong fusion of decision-making disciplines with IT and new technologies [1]. Moreover, four key elements were identified that commonly appear in literature and are in most cases stages of the Consensus-Reaching Process (CRP) within LSDM. It is essential since it helps to make consensual decisions as the number of participants and diversity can increase rapidly. and can be summarized as follows: 1) consensus measurement; 2) subgroup clustering; 3) behavior management and 4) feedback and preference modification [1].

These key elements vary through the literature and use different approaches. However, there were still trends that could be identified within the stages: "Most of the LSDM approaches analyzed within their study included a CRP in them , *with the measurement of the agreement level being a key aspect in defining such models* [1]. The consensus level CD is most widely adopted via an approach which determines the similarity degree among participants, based on a distance function. It can be understood as the supporting rate of participants for a certain alternative. This distance-based approach can be further classified into two consensus measurements: 1) CD is based on the distances to the collective assessment (aggregated collective opinion of all the participants) 2) CD is calculated by distances between pairs of DMs. If a consensus cannot be reached sub clustering can be useful to emerge to a consensus with fewer iterations. Most of these subcluserting methods are based on the distance of participants to collective opinion. Behavior Detection and Management deals with the part of establishing decision weights for participants which is however not mandatory in every LSDM scenario. Feedback and

recommendation processes are on the other side of great importance since they make CRPs more efficient [1]. The main challenges for further progress within LSDM approaches are the following: There is a growing demand for software, e.g. mobile apps, that facilitate distributed LSDM and consensus building processes in real-world scenarios where the decision group structure is decentralized. Blockchain and Distributed Ledger Technologies can provide these decision-making tools with certain advantages, since they enforce more security, integrity and and most importantly can enhance trust and represent reputation in these distributed decision making processes [1].

In this setting, blockchain technologies can provide a tool that enforces security, integrity and cost-effectiveness in these distributed federated decision processes [13]. Blockchain is a decentralized network capable of providing immutability, security, privacy and transparency without a third central authority. It allows tracking of the whole decision-making process, thereby making it more transparent. Conversely, federated blockchain technologies could also benefit from ideas underlying consensus processes in LSDM, so as to reduce the costs of their commonly used consensus algorithms [14] while ensuring security and integrity of activities occurring in distributed networks of miners configuring a blockchain system [1]
Another interesting finding by Ding et is that *In many decision situations, it is no longer realistic nor accurate to make large-scale collective decisions that solely rely on subjective pref- erences of DMs. It has been demonstrated in part of the surveyed litera- ture, that most of such large-scale decisions are rarely made by groups where DMs are socially isolated from each other. Different forms of so- cial relationships between DMs can occur, with varying strength, such as: trust, distrust, influence, reputation, etc . Social data and So- cial Network-based approaches are therefore a valuable tool to better understand the background, motivations and attitude of (Decision Makers)DMs, not only towards the problem being tackled, but also towards each other* [1]. Hence, even these large-scale collective decision-tools need an underlying trust and reputation architecture to trigger a meaningful cooperation. Other problems that need to be addressed is governance of LSDM problems with overlapping communities (most models do not represent the possibility of one participant to be involved in different subgroups), a more comprehensive analysis of decision-maker in Consensus-Reaching Process (CPR) (e.g. to identify non-cooperative behavior), visualizing of the LSDM problem (multiple evaluation criteria, diversity of stakeholders background, conflict and trust relationships etc.). They also state that there is an increasing trend to apply machine learning, computational neuroscience, deep learning, complex system simulation and decision support tools based on AI and statistical methods in order to effectively cope with scenarios where evaluation criteria is interdependent and a decision is collective and large-scale [1]. The biggest potential of innovation is seen in model validation in real-world problems, so far the nature of these tools have been far more theoretical and lack experimental studies to validate new models. They also assume that Smart Cities and Internet of Things (IOT) may influence collective decisions due to active information sharing. The aggregation and usage of the data can enable citizen data scientists to share new information and insights about the environment [1]. Similar insights are provided by Tang et al. which also point out the importance of trust and social relationships, since most theories assume that agents act indepently [15]. Empirical research suggests that trust and reputation are far more important for collaboration than so far assumed by common economic models. Hence, not only the collective action theory by Ostrom recognizes the importance of trust and reputation, but also scholars from LSDM found that trust is a core requirement to enable stakeholders to decide together. The following parts describes the core features of blockchain and explains how it can solve some of the fundamental challenges of collective action and LSDM.

5. Common features of Blockchain Technologies

This part highlights common features of blockchain and further classifies them according to contemporary approaches. A very significant innovation of blockchain technology is that blockchain technologies (not by Bitcoin itself though) have minimized two of the major risks connected to digital currency transactions: 1) The Byztantine General Problem and 2) Double Spending (in Proof of Work Consensus) [7]. Via an univerisial state layer that every actor can trust, even though they may not know each other, the need for centralized verifying third-party authority is removed. Instead protocols are used to reach consensus within the decentralized network. The technical specifications of DLT systems and their consensus mechanisms are becoming increasingly varied in nature [16]. Zwitter et al. show that often-highlighted features of blockchain technologies, such as immutability, transparency, and trustlessness, are in fact design features rather than a given property [16] In the following, the core features are explained, however not discussed in detail, since it would be out of scope. Consequently the opportunities arising with these new features are pointed out with focus on LSDM and the governance of CPRs.
A common classification of blockchain types is based on the access rights to read and write on the distributed ledger. Buterin classified “blockchain-like databases” into the following categories: 1) public blockchains 2) consortium blockchain and 3) Fully private blockchains [17]. The difference is described in the consensus process within the network. In a public blockchain anyone can read, send transactions and participate in the consensus process(e.g Bitcoin and Ethereum). In a consortium blockchain the consensus process is controlled by a preselected set of nodes (e.g. Bloxberg). Fully private blockchains are

kept centralized with one organization. Its permission to read may still be public. These different set ups lead to different levels of centralization and trust within the network and hence community. Buterin also argues that fully-private blockchains are not the best technology choice and may be better implemented with generalized zero knowledge proof technology (e.g.https://github.com/scipr-lab/libsnark). Therefore, this paper focuses on public and consortium blockchains, the latter one is a hybrid model of the low-trust public version and the high-trust single party approach [17]. Blockchain does not only provide a decentralized and transparent way to reach consensus, but can also leverage trust due these features and a commonly agreed consensus process within the entire network [18]. Compensation for the used resources to provide such an infrastructure are provided in forms of incentives via tokens (e.g. Bitcoin or Ether). Which are not only incentives for economic agents and are a missing component in the tragedy of the commons, but also according to Buterin the reason why blockchains are more robust and may solve problems other technologies cannot due to missing incentivisation layer [6]. Möhlmann et al, also analyzed trust and found that it can be one of the key drivers for more sustainable behavior and that blockchain can be an appropriate solution to represent trust digitally and therefore trigger collaboration. [19]. Moreover Blockchain can be used for digital identities which provide great opportunities in terms of data privacy for consumers and citizens. One approach was for example published by w3c which provides an entire guideline for a trust model using not only but also underlying blockchain solutions [20]. Even distributed storage solutions are becoming possible. Examples for decentralized storage solutions are using Blockchain are e.g. Ocean Protocol [21]. Hence, applications reach from industry and automation processes to more ethical and customer-centric approaches, that are not only profitable for companies but also beneficial for customers, citizens, communities, governments and NGOs [6] The following part combines the features and requirements of the presented approaches and explains why they have a great potential to trigger collective action and help to protect the environment at the same time.

6. Technology Fusion: Common Features and Requirements

By combining theories in economics and design mechanisms with progress in cryptographic mechanisms a new and interdisciplinary field emerged: Cryptoeconomics. The founder of Ethereum understood this potential of this combination early and added not only Smart Contracts to the technology stack of blockchains (which is not easily available in bitcoin) but is also one of the Pioneers in Cryptoeconomics [6]. Buterin explains within an extensive interview on better ways to fund public goods, blockchain failures, and effective giving, that a lot of problems like the increasing meat consumption and the related CO2 productions are collective actions problems that are very complex and international in their nature, which makes it impossible for all the stakeholders to meet, interact and decide [6]. Hence, he refers mostly to problems and approaches recognized already by Ostrom, but now combines these economic approaches with cryptographic features and interconnectivity. Even though Buterin talks about public goods and not CPRs, the provided description is valid for some examples as the common-resource pool and addresses problems like climate change which ultimately impact common resources. He suggests a general purpose infrastructure for funding public goods just as money is a tool for funding private goods. pool [6]. Therefore, he suggests a quadratic funding approach that computes the optimum level of social optimum with utility functions to fulfil the Nash Equilibrium. Ostrom stated that there are even more efficient ways than the Nash Equilibrium, however it may be more realistic to start with a well-known model to test different tools and approaches and then shift long-term to new economic approaches, onces insights have been validated. However, they agree in order to make these infractures work, large-scale participation is needed. Buterin also suggests a concrete enforcement or sanction mechanism that may be implemented with Escrow smart contracts [6] that can be run on Ethereum, which is in comparison to Bitcoin turing-complete. Escrow contracts may keep a certain amount of money of governments which will for example not be repaid in case of violation or war e.g. over a certain resource. On the other side, he also points out that there are certain security issues that have not been fixed [6]. Since Buterin is one of the pioneers in incentives designs to increase cooperation, aiming also to provide an infrastructure for public goods with Ethereum, the implementation of smart contracts, given the active community of Ethereum and hence it open-source character, Ethereum may be an appropriate technology choice for further research and experiments, since it is likely that technical developments will consider new forms of decision-making and recognize collective action as opportunity in combination with decentralization, transparency and interconnectivity. Not only Buterin paid attention to this topic, but also other researchers are working on this topic. The research institute of crypto economics published a report about Blockchain and its potential to enhance sustainability and contribute to Sustainable Development Goals by the UN. It is also acknowledged that also purpose-driven Token and hence a fair incentive can trigger value driven collective creation and contribution [22]. Consortial Blockchains seem to be a promising solution to create more experiments with collective action and CPRs in combination with LSDM. Compared to public blockchain they offer certain advantages that are more compliant with Ostrom's principles. The exclusion of participants is given due to the restricted character. Moreover, it provides an easier management of internal rules and less external

disturbance. The level of trust medium, since it is a hybrid solution of low-level trust public model and high-level trust private version. Dib et al. point out that they are especially suitable for highly regulated business (known identities, legal standards, etc.), since there are quite efficient transactions throughput, transactions without fees are also possible [23]. Moreover, recently the TrustoveIP Foundation was found by the linux foundation in order to provide an architecture of Internet-scale digital trust by combining cryptographic trust at the machine layer and human trust at business, legal and social layers. The biggest promises with regards to Collective Action and CPRs is that it will help to verify origins and prove every step of the journey through a supply and ownership chain and potentially help organizations to form and maintain lifetime private digital connections with customers and suppliers with full audit trails for regulatory compliance [24]. Even though there are several issues connected to these promises, it makes it more possible and feasible to test these theories empirically in order to understand which rules will trigger a greater collective outcome for different scenarios. Not only Ostrom pointed out how important trust and reputation are, also researchers in the field of large-scale decision-making, as well as researchers and foundations within the field of blockchain point out the potential of trust to enhance collaboration and hence trigger innovation. At the same time blockchain technologies do not only increase transparency about past actions but also provide a community to monitor resources, users and the network as a whole to a certain extent. Hence, it could be applied to the monitoring part of resources, users and the network even as whole. Monitoring of these networks is already a big research area within the field of blockchains and behavior detection within other consortia and communities [7].

7. Conclusion and Further Research

Since Ostrom published her theories based on observation and studies of real use cases decentralized technologies emerged and humans started to be interconnected. As pointed out by Buterin and Ostrom one of the collective action problems is that it is difficult for all included stakeholders to organize and decide together [6] [10]. Blockchain in Combination with Large-Scale Decision-Making Tools can be a possible infrastructure to test Ostorm's theory further, since technological progress made a lot of the principles less costly and hence feasible. Consortial Blockchains and Ethereum seem to be a good test infrastructure for now, since it enables smart contracts and consortial governance provides some features in alliance with the collective action theory. Therefore, more research on the potential of consortial blockchain governance in combination with CPRs has to be done. Furthermore, Ethereum as a possible test environment should be evaluated in more details with regards to its technical properties and possible implementations also in combination with other technologies and systems. It may lead to a suitable infrastructure which can help to test real world scenarios; and predict with new data analysis tools such as computational neuroscience and machine learning, the outcome while changing one variable at the time and hence refining rules suggested by Ostrom and other experts and interdisciplinary researchers. Moreover, more research needs to be done on consensus-reaching processes that are less costly, then the ones currently used in LSDM [15]. Blockchain Technologies like Ethereum provide already less costly options for Consensus Reaching Processes (e.g. Aura, Clique and IBFT) than the well-known but expensive Proof-of-Work in Bitcoin.. However, researchers, legislators and also entrepreneurs are still facing a lot of challenges within each discipline that needs to be addressed to make further progress with this fairly new field. The combination of these technologies can potentially provide solutions for decentralized self-governance within a centrally organized system that prevents free-riding and depletion of CPRs. Furthermore, it needs to be highlighted that trust is addressed as one of the core requirements for collective action. Therefore, blockchain and its trust-enhancing properties can potenitally be a catalysator for more collaborative behavior among participants and may lead to new insights about collective action and CPRs.

Acknowledgements

This work was partially supported by the Ministry of Economic Affairs, Innovation, Digitalisation and Energy of the State of North Rhine Westphalia as part of the govchain nrw project at the Westphalian University of Applied Sciences in Gelsenkirchen.

Bibliography

[1] R.-X. Ding, I. Palomares, X. Wang, G.-R. Yang, B. Liu, Y. Dong, E. Herrera-Viedma and F. Herrera, "Large-Scale decision-making: Characterization, taxonomy, challenges and future directions from an Artificial Intelligence and applications perspective | Elsevier Enhanced Reader," pp. 83-91, 1 2020.

[2] T. W. Malone, Superminds, 1 ed., Oneworld Publications, 2018.

[3] E. Ostrom, Governing the commons: the evolution of institutions for collective action, Cambridge ; New York: Cambridge University Press, 1990, p. 27.

[4] S. Huckle, R. Bhattacharya, M. White and N. Beloff, "Internet of Things, Blockchain and Shared Economy Applications," Procedia Computer Science, vol. 98, p. 461–466, 1 2016.

[5] Sovrin Foundation, "Control Your Digital Identity," 2020. [Online]. Available: https://sovrin.org/.

[6] V. Buterin, "Vitalik Buterin on effective altruism, better ways to fund public goods, the blockchain's problems so far, and how it could yet change the world - 80,000 Hours," 8 2020.

[Online]. Available: https://80000hours.org/podcast/episodes/vitalik-buterin-new-ways-to-fund-public-goods/#transcript.
[7] E. S. Boubecar, H. Gharbi and A. Jemai, "A Detailed Survey of Blockchain and Its Applications," IEEE international conference on Design & Test of integrated micro & nano-Systems., 6 2020.
[8] P. Tasca and J. C. Tessone, "A Taxonomy of Blockchain Technologies: Principles of Identification and Classification," ledgerjournal.org, p. 7, 2 2019.
[9] Iota Fundation, "Iota Foundation: Social Impact," 8 2020. [Online]. Available: https://www.iota.org/foundation/social-impact.
[10] Elinor Ostrom and Christina Chang and Mark Pennington and Vlad Tarko, “(PDF) The Future of the Commons - Beyond Market Failure and Government Regulation,” pp. 22-24,57-62, 12 2012.
[11] L. Albareda and A. J. G. Sison, "Commons Organizing: Embedding Common Good and Institutions for Collective Action. Insights from Ethics and Economics," *Journal of Business Ethics,* pp. 34-37, 8 2020.
[12] E. Ostrom and J. Walker, Eds., Trust and Reciprocity: Interdisciplinary Lessons for Experimental Research, Russell Sage Foundation, 2003, p. 15.
[13] Z. Zheng, S. Xie, H.-N. Dai, X. Chen and H. Wang, “Blockchain challenges and opportunities: A survey,” nt. J. Web and Grid Services, Vol. 14, No. 4,, 2018.
[14] F. Dalpiaz, J. Zdravkovic and P. Loucopoulos, Eds., Research Challenges in Information Science: 14th International Conference, RCIS 2020, Limassol, Cyprus, September 23–25, 2020, Proceedings, vol. 385, Cham: Springer International Publishing, 2020.
[15] M. Tang and H. Liao, "From conventional group decision making to large-scale group decision making: What are the challenges and how to meet them in big data era? A state-of-the-art survey," *Elsevier,* 10 2019.
[16] A. Zwitter and J. Hazenberg, "Decentralized Network Governance: Blockchain Technology and the Future of Regulation," *Frontiers in Blockchain,* vol. 3, 2020.
[17] Ethereum Foundation and Vitalik Buterin, "On Public and Private Blockchains," 8 2020. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.
[18] F. Hawlitschek, B. Notheisen and T. Teubner, "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy," *Electronic Commerce Research and Applications,* vol. 29, p. 50–63, 5 2018.
[19] M. Möhlmann and A. Geissinger, “Trust in the Sharing Economy: Platform-Mediated Peer Trust,” *The Cambridge Handbook on Law and Regulation of the Sharing Economy,* pp. 6-8, July 2018.
[20] w3.org, “Verifiable Credentials Data Model 1.0,” 8 2020. [Online]. Available: https://www.w3.org/TR/vc-data-model/#trust-model.
[21] Ocean Protocol, "Ocean Protocol," 8 2020. [Online]. Available: https://oceanprotocol.com/technology/roadmap#papers.
[22] S. Voshgmir, “BLOCKCHAIN, WEB3 & THE SDGs,” Research Institute for Cryptoeconomics, Austria, 2019.
[23] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea and E. B. Hamida, "Consortium Blockchains: Overview, Applications and Challenges," International Journal on Advances in Telecommunications, vol 11 no 1 & 2, 2018.
[24] T. O. IP, “Trust Over IP - Defining a complete architecture for Internet-scale digital trust,” 8 2020. [Online]. Available: https://trustoverip.org/.
[25] “Home | IOTA,” 8 2020. [Online]. Available: https://www.iota.org/.

BLOCKCHAIN TECHNOLOGIES IN THE EDUCATIONAL SECTOR: A REFLECTION ON THE TOPIC IN THE MIDDLE OF THE COVID-19 SITUATION.

Alexander Pfeiffer [*1] [*3] [*4], André Thomas [*2], Thomas Wernbacher [*3], Michael Black [*2], Lloyd Donelan [*2], Brenton Lenzen [*2], Nick Muniz [*2], Alexiei Dingli [*4], Vince Vella [*4], Stephen Bezzina [*5], Manuel Pirker-Ihl [*6]

The MIT Education Arcade, 77 Massachusetts Avenue, 14E-303, Cambridge, MA 02139, USA *1
LIVE LAB at Texas A&M University, Langford Architecture Building 3137, College Station, TX 77840, USA *2
Applied Game Studies at Donau-Universität Krems, Dr. Karl Dorrek Straße 30, 3500 Krems, Austria*3
Department for AI at University of Malta, Msida, MSD 2080, Malta *4
Ministry for Education and Employment, Great Siege Road, Floriana, Malta *5
Picapipe Gmbh, Geylinggasse 17/1, 1130 Wien, Austria *6

This paper looks at current projects in the field of Blockchain in education, their specific areas of application, possible advantages and weaknesses. Three examples developed by the team of authors are introduced in detail. First: Gallery-Defender a Serious Game, which was adapted to serve as a demonstrator in a stand-alone version to show the possibility to carry out exams directly from within the game and store the grades and meta-data on Blockchain. Second: Art-Quiz, an e-learning tool, which can be integrated into existing LMS systems and map exam results and further data using Blockchain technologies. Both were developed following an iterative design process. And third: The results of a focus group, which simulated the assignment of grades after an oral online exam. The three examples presented here are based on the Blockchain system Ardor/Childchain Ignis, but each demonstrator has a different set of features and approaches. In addition, the integration of various Blockchain solutions was conceptually designed to make a Multi-Chain model possible.

1. Introduction

The rapid changes brought about by digital technologies in education offer rich, personalised and differentiated modes of e-learning. However, the anytime, anywhere access to teaching, learning and assessment material requires a paradigm shift in the conceptualisation and implementation of validation, verification, authentication and storing of students' data.

Blockchain technologies offer an interesting and innovative approach for securing sensitive information in online educational environments. One of its main impetus is the ability, or rather the non-ability of retrospectively altering data which is stored on the Blockchain. This indelible and unalterable nature of Blockchain technologies allow for greater safeguarding when compared to conventional password-protected directories, from both within and outside the organisational e-learning environment. Furthermore, the open nature of public Blockchains, supports decentralised data verification, hence independent of any central authority and consequently valid across different programmes, departments, institutions and countries. This also extends beyond traditional formal learning institutions, such as non-formal or informal education, but more importantly, it offers an easy and inexpensive way for businesses and job providers to safely and securely verify prospective employees' credentials. The current COVID-19 situation has shown that during times of massive travel restrictions, problems with mailings and even complete lockdown, we need to have digital capabilities where secure, non-manipulable storage of data and digital identities are combined. Even during this difficult period, school grades, certificates, employment certificates and similar documents must be issued on the one hand and checked for their validity on the other.

Furthermore, the period of lockdown, months of home schooling in the school system and online distance learning at universities has shown that e-assessment formats also require technological innovations, regardless of whether the examinations are conducted using modern approaches such as Game Based Learning & Assessment, classic e-learning tools or simple video chats. The aspects of identity verification, secure assignment and storage of grades, acceptance of the grading and digital transfer to the administrative departments are of utmost importance. And some of these were arguably not guaranteed over the past few months.

With regard to related research, the fundamental work "JR science for policy reports: Blockchain in Education" by Grech and Camillieri [1] should be highlighted. In 2017 they described that Blockchain in the educational sector is still in its beginning, but they see the following use cases in the future:

- creation of digital certificates/certificates or creation of digital proof of authenticity of printed certificates;
- storage of proofs of performance after examinations including meta data;
- recognition of examination results between and within educational institutions;
- use of a personal "lifelong learning" directory (virtual CV);
- verification of the authenticity of the certificates by third parties (e.g. personnel managers authorised by applicants);
- management of intellectual property, e.g. in the context of project implementation;
- processing of payments.

They further describe various basic assumptions that need to be made in order for Blockchain to establish its place in the educational sector

- open implementations of the technology;
- use the open source software;
- use open standards for data;
- implement self-managed data management solution;
- further developments must be driven forward jointly by market participants and regulators / authorities.

Min et al. [2] discuss the Blockchain integration for games and then categorized existing Blockchain games from the aspects of their genres and technical platforms. Aini et al. [3] explore different approaches to Gamification that embed Blockchain technologies in the educational sector. In their work, Agustin et. al [4] describe the application of Blockchain technology in e-certificates in the open journal system. The study reports that issuance of e-certificates in an open journal system is a way to manage and verify, prevent duplications or even falsification of e-certificates and the reputation of the open journal system is already given. This project is based on Blockcerts by Learning Machine (originally developed at MIT). Merija and Kapenieks [5] compare Blockcerts with Ethereum Smart Contracts developed by Open University, UK, while Baldi et al [6] describe how to impersonate a legitimate issuer of Blockcerts certificates with the aim to produce certificates that cannot be distinguished from originals by the Blockcerts validation procedure.

Pfeiffer and König [7] discuss the use of blockchain technologies in educational games for assessment from a humanities perspective. The authors set up a category system for learning games and e-learning systems with the aim of also serving as examination tools. From this paper it is concluded that the game Gallery Defender and the Art Quiz fall into the category Game Based Learning & Assessment respectively E-Learning & Assessment. The learning environment corresponds to the exam setting. However, the exam takes place in a separate instance.

Serada et. al [8] analyze specific characteristics of value created through digital scarcity and Blockchain-proven ownership in cryptogames. Pfeiffer et. al [9] have identified 8 different types of Blockchain based tokens, based on their possible applications. In the further examples described in this article, the following token type is used to represent the grading or as a certificate:

> *"Non-freely tradeable utility tokens: These tokens store data, such as certificates, grades, ownership of a piece; fine art prints (e.g. limited edition prints, each with a unique number), or a last will; they can be a unique (singleton) token per record or a message attached to a specific token when sending. A separate series of tokens is generated for each different use case. Each series has its own asset ID on the respective Blockchain. (the name of the series does not have to be unique, only the asset ID). This means: The moment a message is added to one of the tokens (from a series) and this token is sent, the connection of the token with the message and the rule that the token cannot be forwarded without the knowledge of the original sender becomes a unique process, which is identified by the unique transaction ID. Messages can be attached unencrypted or encrypted. This data is usually linked to a person or a property and is not (or only under specific circumstances) tradable. It is also linked to a specific wallet (e.g. of the recipient). The Singleton/Unique Token form of this category is similar to the concept of non-fungible tokens (NFTs)."*

The 3 different approaches

- Game-based learning learning & assessment
- E-learning & assessment
- and online video education & assessment

All these approaches store the certificates on Blockchain.

2. Gallery Defender

To set up the demonstrator in an educational framework, the Serious Game is based on the requirements for art history in an introductory college level Art History Curriculum Framework:

The learning goal, which equals the game goal is defined as "All students, which means all players, will understand, analyze, and describe art styles in their historical, social, and cultural contexts."

The serious game introduces and later assess the art concepts to the player/learner. The game is inspired by the ARTé: Lumiere [10, 11] game. The player/learner slips into the role of a gallery owner. Using their profound knowledge of art history, the player must defend the artworks of the gallery from a master thief. The demonstrator utilizes Blockchain technologies in three different ways. It is important to note that 10 million of the respective tokens (for the teacher vedutt asset-ID 1653013092595194366 / for the students veduut asset-ID 1494447768104309209) were registered on the Ignis Childchain of the Ardor network.

A player/learner receives a digital token at the end of the assessment, which contains the grade, points, and time that the assessment was finished as a message. This token and the attached message are stored forever and unchangeably on the Blockchain in the player/learner's Ardor Blockchain Wallet. The certificate is encrypted and can only be decrypted by the sender and the original recipient. However, using a shared key enables the player/learner to share the results with third parties, such as their future boss, their family or another school/university. Teachers can receive a token with further information about the respective test result. After a definable time the message is deleted, and only the proof that the token

has been sent remains (and thus an assessment has been carried out).

A gamification system was also simulated on a conceptual level and in test runs using the API calls directly:

The idea is that a player/learner with particularly good results receives an additional token with which the authors of the article use to show gamification principles on Blockchain. This token can be exchanged for digital rewards, e.g. a game poster. This functionality shows off how an entire ecosystem for rewards can be built on Blockchain and included within the context of education. However, this is not yet implemented in the current version of the game. The tokens used in the token system of the demonstrator are defined as “non-freely tradeable utility tokens” (as discussed earlier), built on the Ignis child chain of the Ardor Network. It is of utmost importance that the tokens cannot be traded or sent outside of the learning/assessment environment. For this purpose we worked with approval models.

The communication between the ready-made game and the Blockchain works via API calls. These calls are set in the game engine unity during the development of the game. It was also important to us that the users do not need to have any knowledge of Blockchain and do not have to pay network fees with cryptocurrencies. Bundling accounts were used here, in case fees would be charged for the recipients.

Another essential point is to verify the identity and account of the game. In the third iteration, the game account from the Ardor main net was registered using Alexander Pfeiffer's digital citizen card. For this purpose, a document was placed in Alexander Pfeiffer's citizen e-vault and the transaction and the link to the document was stored in the wallet description of the game's Ardor account.

The game has been developed following an iterative game design process. The first two iterations took place on the testnet of the system, the third and last iteration was then implemented on the mainnet of the Ardor Blockchain.

All Blockchain transaction from the final iteration can be retrieved e.g. from https://ardor.tools/account/ARDOR-EEM6-N4UP-53XH-3LY9H

- ARDOR-EEM6-N4UP-53XH-3LY9H (Ardor Account of the Game)
- ARDOR-BN5K-N7EL-CQFD-EYD7W (Student Default)
- ARDOR-9UDW-4T5C-9D44-2Y5MX (Teacher Default)

Remark: Using the teacher account, all transaction data can be viewed as an unencrypted message.

The process is now shown in the form of a picture gallery:

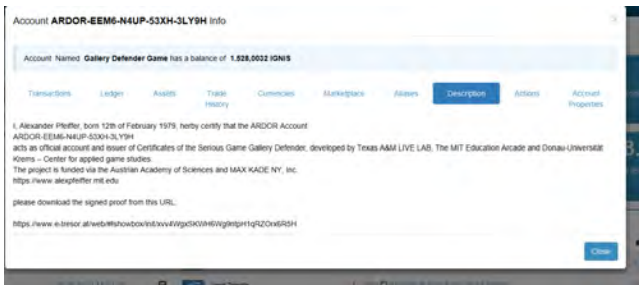


Fig. 1 - Proof of the account ownership, from the account properties of the Serious Game main Blockchain address • ARDOR-EEM6-N4UP-53XH-3LY9H

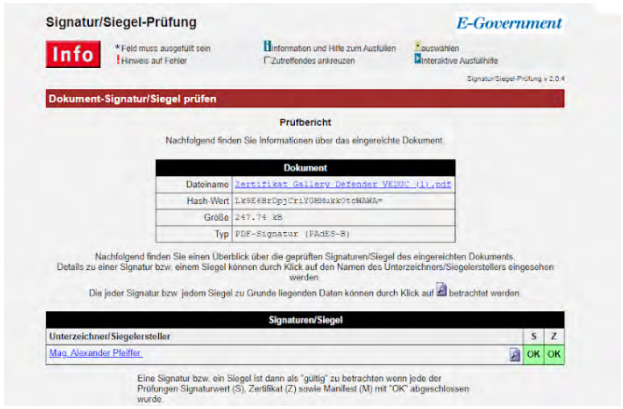


Fig. 2 - Proof of ownership retreived from the Austrian E-Government System RIS

Fig. 3 - Main screen of the game. The user can change between learning - simulation or assessment. If assessment is selected, a warning screen that the results are recorded on Blockchain pops up

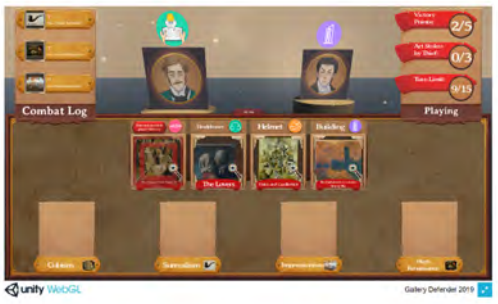


Fig. 4 - Screenshot of the actual game play

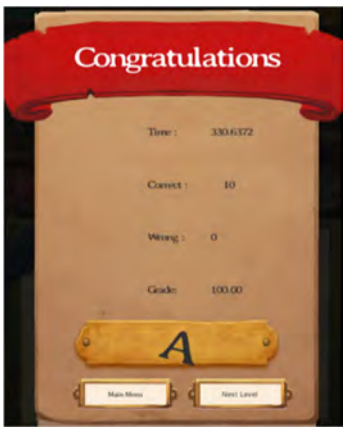


Fig. 5 - The game has been completed in 300 seconds, with an A grade and 100 points

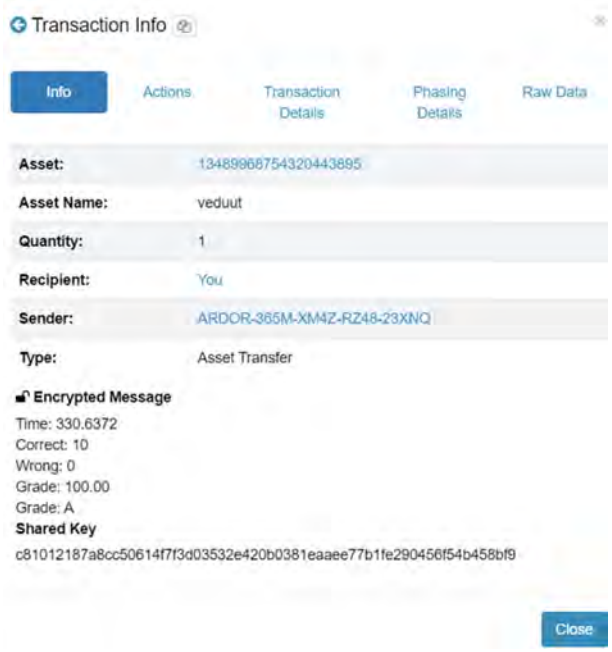


Fig. 6 - Exactly this result is sent as token (one piece of the veduut token with the message attachted) to the Blockchain address of the learner. After the learner has accessed the information with his private key, a shared key is generated to share the information with others. For example during an application. The teacher received at the same time one piece of the vedutt token, in our example with the same content, but as unencrypted message. Of course the teacher could receive additional information from the game-engine, that helps to guide the learner through his/her further learning experience

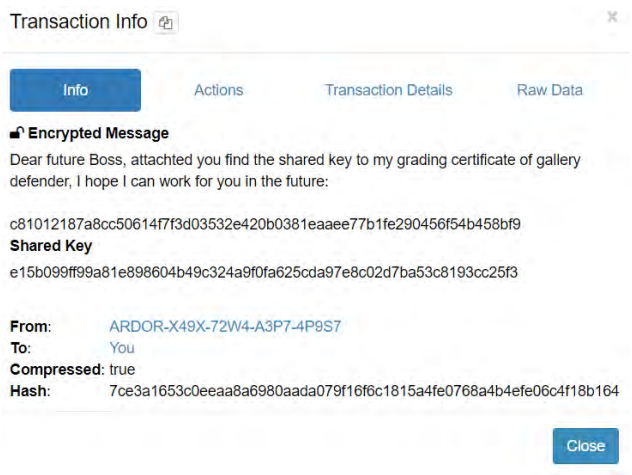


Fig. 7 - In this example the token information is shared during such application process with "the future boss"

3. Art-Quiz

The Picapipe GmbH Quizengine was enhanced for this project with a Blockchain module. In this special iteration, learning content and exam questions were developed to be an optimal addition to the Serious Game Gallery Defender by providing learners with additional information about the artworks and artists in the form of a quiz.

With the Blockchain approach chosen in this example a Singleton Token (NFT) is registered on the Blockchain after each completed test. Each exam is therefore registered on its own Asset-ID on the Ignis Childchain. Personalized information can now be provided not only as in an (encrypted) message but also in the asset properties However, the information in the Asset Properties is publicly accessible if you know the asset ID of the token. However, this is not public information per se. Lightweight Smart Contracts connect the Blockchain operations with the Learning Management System. Shared keys can be used to share exam results with others.

With the Art Quiz not only a different approach to the token structure was chosen, it was demonstrated that a web app, which can also be easily integrated into a Learning Management System (LMS), can store grades and certificates on Blockchain. Not only in the form of a 1-way-hash but, as already achieved with the Gallery-Defender game, also including meta data from the exam.

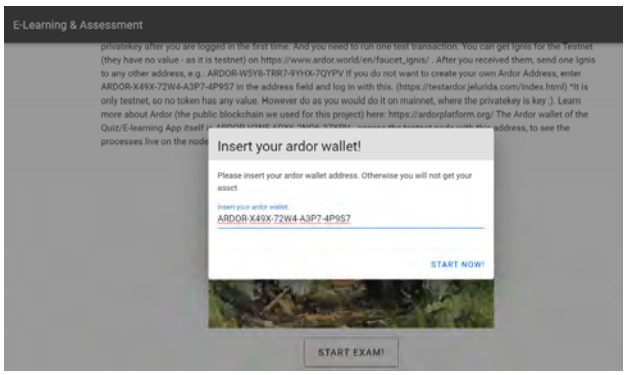


Fig. 8 - The examinee has to enter his/her ardor account address before the test starts

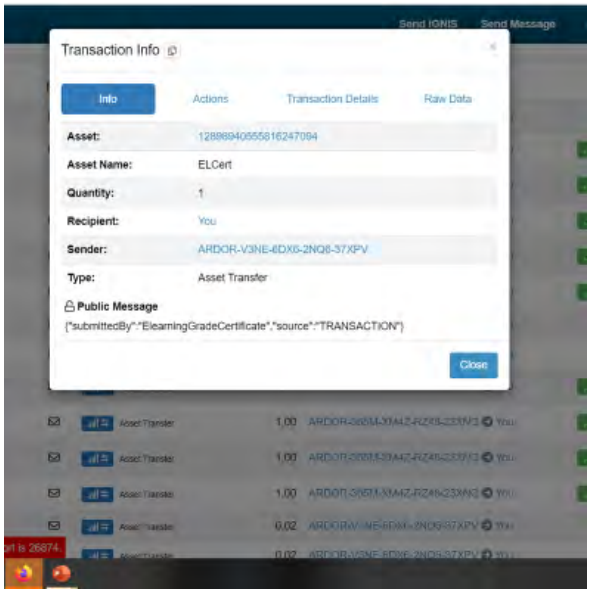


Fig. 9 - a singleton asset called "ElCert" is being created each time an exam has been finished. Each asset has its own ID on the Blockchain. A Smart Contract process the data and sends the results to the examinee

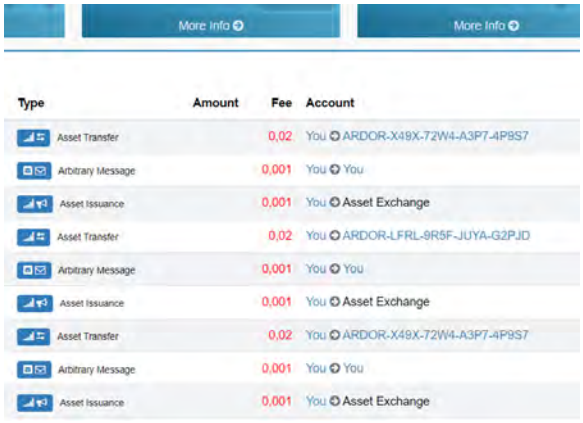


Fig. 10 - (1): Asset Creation - (2): Attachment of the meta data (grades) - 3: Transfer to the student's account

4. Online Video Education & Assessment

The goal of the third demonstrator was to conduct a role of an verbal exam, creating the tokens and transactions directly on the Ignis Childchain's Backend. Therefore this setting intended to imitate a online verbal exam situation using Zoom and the certificate transfer process via Blockchain following the successful completion.

The first step was the confirmation of the Ardor Blockchain account. To demonstrate this the account address was published via the twitter account of the applied game studies center of Donau-University Krems. The next step was to demonstrate the registration of students' Ardor addresses. This was done via a permanent text message from the university account to the same account.

The process for the verbal exam is as follows: For each exam a singleton (NFT) token is created. In the token properties the subject / university is described. When the token is sent, the exam performance is recorded as an encrypted message.

In the approval model, both the student's account and the sender's account must agree. This guarantees the acceptance of the exam grade by the student and the exam token cannot later leave the student's account. Again, shared keys can be used to share exam results with others. Smart Contracts can of course extend the functions significantly.

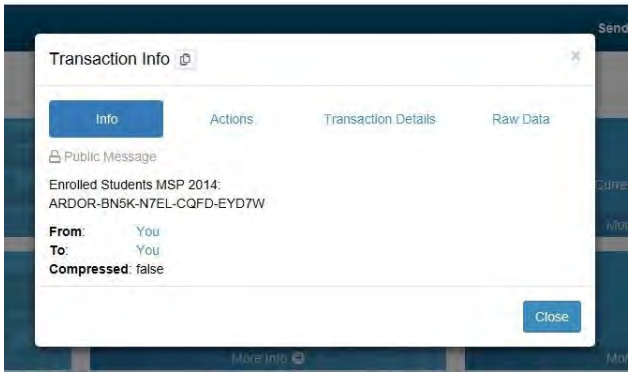


Fig. 11 - Registration of the studentes ardor accounts on Blockchain

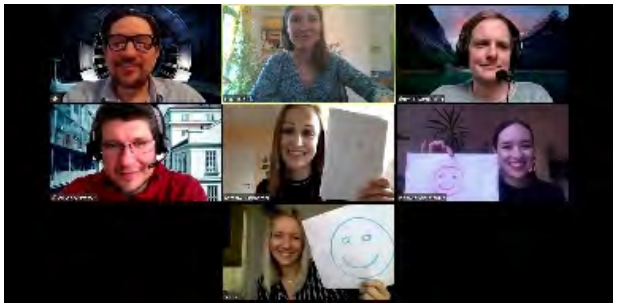


Fig. 12 – Actual online exam situation. Screenshot was taken during the Covid-19 lockdown

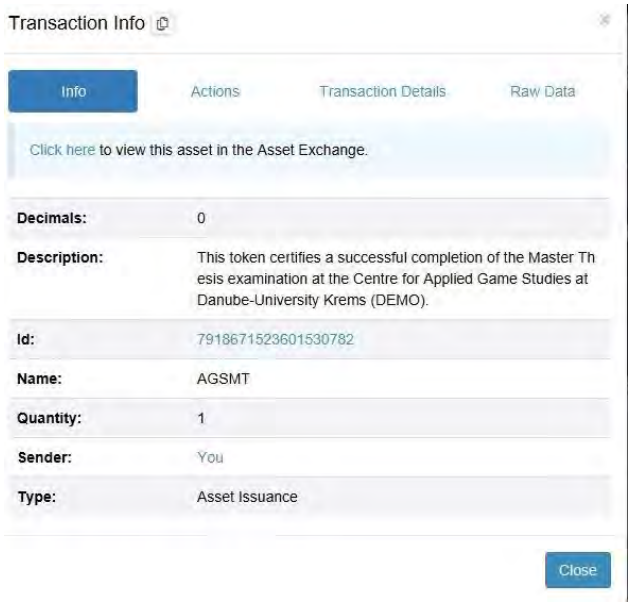


Fig. 13 - (Singleton) Token representing the successful completion of a Master Thesis Defensio (Demo)

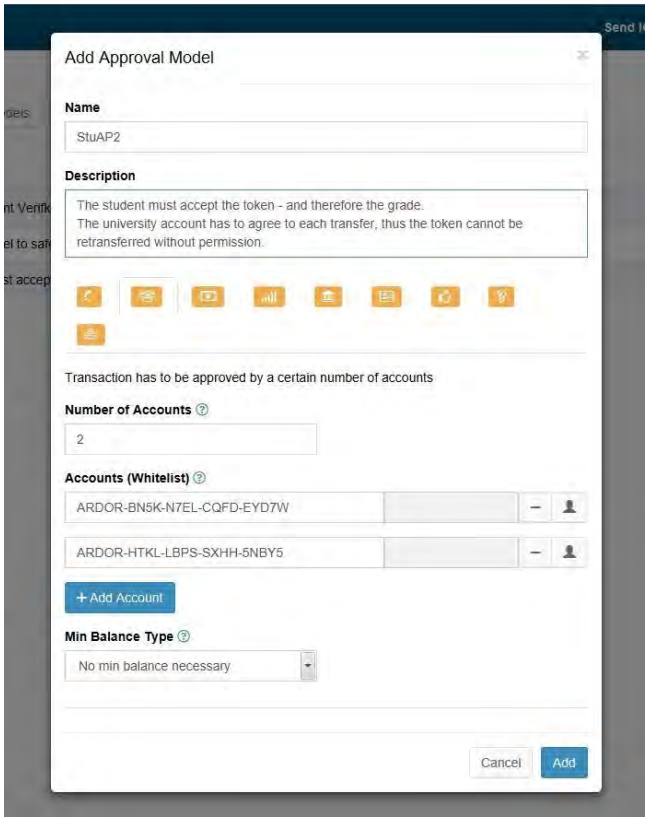


Fig. 14 - Approval model setting the rules for transfering the token. In this case both the student and the university have to agree on the transaction

5. Conclusion and further research

With these three examples the authors had the goal to show different use cases how Blockchain can be used in the educational sector. In all three cases the result should go beyond the mere proof of an original certificate. In any case, the initial findings are promising and have led to the list of the following questions, which further projects in this sector should address.

- Is a private or public Blockchain to be used?
- Which Blockchain or combination of Blockchains systems is to be used?
- Can the information-carrying tokens be sent from one account to another without this being originally intended?
- Is the private information of the token sufficiently encrypted? And if so, who has the keys to access the data?
- Can (temporary) shared-keys be created and in turn give access to (specific) information for third parties?
- Is it possible for companies / universities / government institutions to operate their own nodes in the network and thus have data sovereignty? If so, how much effort and costs are involved?
- Is it even possible for users to run their own node? And if so, with what effort and costs?
- Do the partners involved have to purchase Cryptocurrencies? If so, for what purposes? And subsequently, do the users have to purchase cryptocurrencies?
- Further: Who pays the transaction fees to the network? Can solutions be used that do not charge transaction fees? If this is the case – which network was used and what are the effects of using a transaction fee-free solution?
- Do the partners of the respective project or the users create their own Blockchain Wallets? Or is this done for them “through the app”. In the case of the latter, who and in what form has control over the private keys?

The team of authors would like to pay special attention to two aspects in the next research. The development/implementation of an internationally recognized solution in the field of identity management and the expansion of the system to include various public or private (e.g. for GDPR sensitive information) Blockchain systems. This would only be possible in practice if it involves multi-chain systems.

Acknowledgements

We would like to thank the Austrian Academy of Sciences and the Max Kade NY Foundation for making this project possible.

References

[1] A. Grech,.; A. F.Camilleri, Blockchain in Education. Luxembourg : Publications Office of the European Union 2017, (2017) 132 S. - (JRC Science for Policy Report) - URN: urn:nbn:de:0111-pedocs-150132

[2] T. Min, H. Wang, Y. Guo, and W. Cai, Blockchain Games: A Survey. In Proceedings of the IEEE Conference on Games (CoG). IEEE (2019)

[3] Q. Aini, U. Rahardja, and A. Khoirunisa. Blockchain Technology into Gamification on Education. IJCCS (Indonesian Journal of Computing and Cybernetics Systems) [n.d.]. 14

[4] F. Agustin, Q. Aini, A. Khoirunisa, E. A. Nabila, Utilization of Blockchain Technology for Management E-Certificate Open Journal System. Aptisi Transactions on Management (ATM). 4, 2(Apr2020), (2020)134-139.

[5] J. Merija, J. Kapenieks, Blockchain and the Future of Digital Learning Credential Assessment and Management. Journal of Teacher Education for Sustainability. 20.(2020) 145-156. 10.2478/jtes-2018-0009.

[6] M. Baldi, et al., Security analysis of a Blockchain-based protocol for the certification of academic credentials. arXiv preprint arXiv:1910.04622 (2019)

[7] A. Pfeiffer, N. König, Blockchain Technologies and Their Impact on Game-Based Education and Learning Assessment. In: Elmenreich W.,

Schallegger R., Schniz F., Gabriel S., Pölsterl G., Ruge W. (eds) Savegame. Perspektiven der Game Studies. Springer VS, Wiesbaden (2019)
[8] A. Serada, T. Sihvonen, J. Harviainen, CryptoKitties and the New Ludic Economy: How Blockchain Introduces Value, Ownership, and Scarcity in Digital Gaming. Games and Culture. https://doi.org/10.1177/1555412019898305. (2020)
[9] A. Pfeiffer, S. Bezzina, S, T. Wernbacher, , S. Kriglstein, Blockchain Technologies for the Validation, Verification, Authentication and Storing of Students' Data., in ECEL 20 Proceedings (2020) retrieved from: https://www.researchgate.net/publication/339254736_BlockchainTechnologies_and_Social_Media_A_Snapshot (last visitied 31.08.2020)
[10] A. Thomas, H. Ramadan, L. Campana, D. Leiderman, Sutherland,. M. Zawadzki, ARTé: Lumiere. Available electronically from https://hdl.handle.net/1969.1/188003 (2018)
[11] W. Weng, H. Ramadan, A. Thomas, Understanding Enjoyment in ARTé: Mecenas with EGameFlow. The IEEE Conference on Games (CoG) 2020 Proceedings (2020)

PROBABILISTISCHE MIKROZAHLUNGEN AUF DER BLOCKCHAIN

Marianne Poser
Hochschule Mittweida, Technikumplatz 17, D-09648 Mittweida

Dieses Paper ist eine überarbeitete Kurzfassung der Bachelorarbeit, welche 2019 von der Autorin unter dem gleichen Titel geschrieben wurde. Sie setzt sich mit der Herausforderung, kleine Zahlungen effizient mithilfe der Blockchain umzusetzen, auseinander. Ziel ist dabei, verschiedene Ansätze vorzustellen und ihr Potenzial zu prüfen. Prinzipiell hat der Einsatz von Micropayment-Schemas das Ziel, (häufige) Zahlungen von Kleinbeträgen in der Abwicklung möglichst effizient zu gestalten. Das Ungleichgewicht, dass die Kosten einer Zahlung den zu zahlenden Betrag übersteigen, gilt es insbesondere auf der Blockchain zu vermeiden. In diesem Paper werden verschiedene Ansätze für Micropayments vorgestellt und nach verschiedenen Punkten untersucht werden. Dabei wird unter anderem Wert auf die Kostenminimierung, Sicherheit und dezentrale Umsetzbarkeit gelegt. Aber auch die Anwendbarkeit und Ressourcenanforderung der verschiedenen Schemata sollen in dieser Arbeit betrachtet werden.

1. Einleitung

Die Blockchain ist als Technologie inzwischen ein Trendthema. Sowohl in der öffentlichen Diskussion in den Medien, als auch in der Finanzbranche und im Technologiebereich. Eine weitere Entwicklung, die in verschiedenen Bereichen immer wieder thematisiert wird, ist das Internet of Things (Internet der Dinge), kurz IoT. Beide Begriffe gelten inzwischen als „Buzzwords“, werden also genutzt, um Aufmerksamkeit zu erzeugen. Beide Begriffe sind auch auf dem Gartner Hype-Zyklus für neue Technologien von 2018 zu finden und sollen erst in fünf bis zehn Jahren auf ihrem „Plateau der Produktivität“ ankommen

Ein Thema, welches die Menschheit hingegen vermutlich schon immer beschäftigt, sind Zahlungen. Von der Entwicklung erster Zahlungsmittel bis zum Ablauf einer elektronischen Zahlung liegen Tausende Jahre von Weiterentwicklung und Forschung. Diese Weiterentwicklung von neuen Konzepten und der Einsatz neuer Technologien, wie der Blockchain, hält an. Aus dieser Entwicklung sind unter anderem Schemata für sehr kleine Zahlungen, sogenannte Micropayments, hervorgegangen.

In dieser Arbeit wird das Potenzial von probabilistischen Mikrozahlungen geprüft. Dafür wird ein Zahlungsschema gesucht, welches

- geringe Anforderungen an die IoT-Geräte stellt,
- nur minimale Kosten pro Zahlung verursacht,
- ein gewisses Maß an Sicherheit bietet und
- ohne den Einsatz einer zentralen Instanz umgesetzt werden kann.

Welche Ansätze dieser Herausforderung begegnen, welche neuen Probleme dabei aufkommen werden und ob sich darunter eine Lösung für das Schaffen eines Incentives befindet, wird Thema dieser Arbeit sein. Besonders die Unterkategorie probabilistischer Micropayments, welche in dieser Betrachtung besonderen Stellenwert hat, wird ausführlich behandelt werden. Es wird gezeigt werden, was diese von anderen Ansätzen unterscheidet, welches Potenzial sie haben, aber auch mit welchen Herausforderungen sie verbunden sind.

2. Begriffe und Grundlagen

Micropayments sind Zahlungen sehr kleiner Beträge. Der Hauptunterschied zwischen Micro- und Macropayments liegt in der Höhe des Betrags. Micropayments decken dabei den Bereich der Kleinstbeträge, wie wenige Cent oder Bruchteile dieser, bis zu wenigen Euro ab. Vor allem im Zuge der Digitalisierung sind sie wieder in den Vordergrund gerückt, um Zahlungsmodelle wie „Pay-per-Use“ zu ermöglichen. Damit diese Zahlungsmodelle rentabel sind, ist es von großer Wichtigkeit, dass die Gebühren für eine Transaktion geringer sind, als der eigentliche Zahlungsbetrag. Um das realisieren zu können, benötigt es Micropayment-Schemata, bei denen die Kosten pro Zahlung minimiert werden. Ein weiterer wichtiger Punkt ist, dass die Zahlungen schnell erfolgen, bzw schnell final sind. Dadurch können sie auch eingesetzt werden, wenn bezahlte Leistungen unmittelbar erbracht werden müssen. Das würde beispielsweise eine minütliche Abrechnung über den Konsum eines Stream ermöglichen.

Der Grundaufbau eines Zahlungsschemas beruht in zentralen Systemen auf drei Parteien. Es gibt einen Zahlenden, auch User, Kunde oder Sender genannt, der einen monetären Wert ausgibt. Das Ziel dieser Zahlung ist der Verkäufer beziehungsweise Lieferant. Die dritte Partei, die in verschiedenen Formen auftritt, kann eine Bank sein, wird aber auch Broker genannt. Um die Transaktionskosten zu minimieren, wird versucht, mehrere kleinere Zahlungen zu wenigen großen Zahlungen zusammen zu fassen. Die vielen kleinen Pay-per-Use Zahlungen sollen vereint werden und können entweder im prepaid oder im postpaid Ansatz final bezahlt werden.

An ein Micropayment Schema werden verschiedene Anforderungen gestellt, die je nach Nutzung variieren. Eine offensichtliche Forderung ist, dass die Verarbeitungskosten pro Zahlung minimal gehalten werden. Der Erfolg eines Schemas ist auch an weitere Faktoren geknüpft. Ein wichtiger Aspekt ist die Akzeptanz des Konzeptes. Damit der Sender ein System akzeptiert, muss es einfach zu nutzen und schnell, günstig einsetzbar sein. Ein Empfänger hingegen könnte eine höhere einmalige Investition täti-

gen, wenn die Effizienz pro Zahlung steigt. Wie umfangreich das Set-up und wie hoch die Eintrittsschwelle für das System ist, variiert nach Anwendung. Ist die Kunden-Verkäufer-Beziehung kurzweilig, sollte das Schema unabhängig davon arbeiten können.

Der Aufwand für Registrierung, das Erstellen eines Kontos und Einzahlung hat Einfluss auf die Akzeptanz aller Nutzer. Für größere Systeme sollte das eingesetzte Schema auch nach seinen Skalierungsmöglichkeiten ausgewählt werden. Der Sicherheitsaspekt sollte trotz geringer Beträge nicht vernachlässigt werden. Wenn nicht jede Transaktion online verarbeitet werden kann, wie wird dann sichergestellt, dass Angriffe entdeckt und abgewehrt oder bestraft werden können? Eine weitere Herausforderung kann der Wunsch nach Anonymität beim Sender sein. Im Folgenden sollen Erkenntnisse über verschiedene existierende Lösungen vorgestellt werden.

3. Micropayments auf der Blockchain

Auch wenn die Blockchain Technologie bereits Transaktionskosten reduziert, vor allem im Vergleich zu Überweisungen im internationalen Raum, sind sie dennoch auf public Blockchains zu hoch. Vor allem Micropayments bleiben unrentabel. Ein Ziel ist es also, die Transaktionskosten zu reduzieren. In der Blockchain-Technologie zahlt man für jede On-Chain Transaktion eine Gebühr. Diese Gebühren werden potenziell eher steigen, nämlich dann, wenn die Miner keine Coins mehr als Belohnung für den Block erhalten. Ab dann werden sie nur noch durch die Transaktionsgebühren bezahlt. Neben der Kostenreduktion benötigen aktuelle Blockchains auch Lösungen für Skalierung und Beschleunigung. Eine On-Chain Transaktion auf der Bitcoin Blockchain braucht etwa zehn Minuten, bevor sie in einem Block auf der Blockchain steht. Anschließend benötigt sie weitere sechs erfolgreiche Blöcke, um als bestätigt zu gelten. Um sicher zu sein, müsste ein Verkäufer etwa eine Stunde warten, bis er sicher sein kann, bezahlt worden zu sein und die Ware zu liefern. Das ist in vielen Szenarien nicht praktikabel. Auch die Skalierbarkeit ist eine Herausforderung. Um Blockchain wirklich weitläufig einsetzen zu können, braucht es Lösungen, um mehr als etwa zehn Transaktionen pro Sekunde verarbeiten zu können.

Alle Ansätze, die auf eine Blockchain aufbauen, sollten die Vorteile, die eine Dezentralisierung mit sich bringt, erhalten. Es sollte also auf den Einsatz von zentralen Einheiten oder Parteien, denen man vertrauen muss, verzichtet werden. Allgemein gilt zu beachten, dass es grundsätzlich zwischen den Teilnehmern keine Vertrauensbasis gibt. Es existieren verschiedene Ansätze Micropayments auf der Blockchain umzusetzen.

Mit den vermehrten Einsatzmöglichkeiten von Micropayments stehen auch Entwicklung und Weiterentwicklung der Micropayment Schemata im Fokus. Entsprechend viele unterschiedliche Ansätze wurden veröffentlicht. Das Ziel viele kleine Transaktionen zu wenigen Großen zusammenzufassen, kann auf verschiedene Arten erreicht werden. Doch weitere wichtige Aspekte wie Kosten, Sicherheit und Akzeptanz werden nicht immer vollumfänglich behandelt. Einige Erkenntnisse einer Recherche über unter Anderem Channels und Plasma sollen folgend zusammengefasst werden.

Eine Umstrukturierung eines bestehenden Systems ist immer mit Kosten verbunden und in manchen Fällen lohnt sich, trotz Kostenreduktion in der Anwendung, ein Wechsel nicht. Ein Beispiel dafür ist die Nutzung eines Tokens. Dieser benötigt ein funktionierendes Konzept, ein Broker muss eingerichtet und betrieben werden und die Einsparungen sind je nach System überschaubar. Bei der Umsetzung von Micropayment Systemen auf der Ethereum Blockchain müssen Smart Contracts entwickelt werden und auf die Blockchain geschrieben werden. Beides ist mit Kosten verbunden. Auch auf Seite der Sender kostet die Umsetzung und Programmierung der Konzepte Zeit und Geld. Dennoch ist verallgemeinert zu sagen, dass die meisten hier behandelten Lösungen, die auf Blockchain basieren, Potenzial haben Kosten zu senken.

Der Aspekt der Sicherheit wird von den existierenden Schemata unterschiedlich behandelt. In einigen wird die Absicherung der zentralen Einheit überlassen und ist nicht Teil der Betrachtung. Andere verzichten zumindest auf die Absicherung jedes einzelnen Micropayments, aufgrund des geringen Wertes. Dennoch kann jede Sicherheitslücke kritisch werden, sobald sie im großen Stil ausgenutzt werden kann. Ist dies möglich, kann ein Angriff trotz Bestrafung durch Verlust einer Kaution, rentabel werden. Es sind also andere Sicherheitsprinzipien nötig, um die verschiedenen Attacken erfolgreich abwehren zu können. Viele Sicherheitsprinzipien wirken anderen Aspekten, wie Akzeptanz der Parteien und Performance des Systems entgegen. Dennoch gilt der Sicherheitsaspekt als existenziell. Vor allem Systeme auf der Blockchain müssen hier eigene Ideen entwickeln, um der Gefahr von Double Spend, Overspend und Replay zu begegnen. Auch der Sender muss vor möglichen Attacken geschützt werden.

Eine Bestrafung auf der Blockchain muss vorbereitet werden, dies geschieht häufig durch das Hinterlegen eines Deposits. Wie hoch das Deposit ist, wird von Fall zu Fall unterschieden und kalkuliert werden. Der Verlust des Deposits soll den Angriff dabei unrentabel machen, weshalb zunächst berechnet werden muss, wie viel ein böswilliger Sender mit einer Attacke erreichen kann. Daraus leitet sich die Höhe des Deposits ab. Ist der Betrag allerdings zu hoch, sinkt der Wille des Senders dieses System zu nutzen. Außerdem könnte es nicht ausreichen, Angriffe unrentabel zu machen, wenn ein Angreifer eine andere Motivation hat. Diese Motivation könnte sein, einen Konkurrenten aus dem System zu entfernen oder zu schädigen. Es müssen also andere Schutzmechanismen gesucht und eingesetzt werden.

Die Akzeptanz von Sender und Empfänger ist abhängig von:

• Der Eintrittsschwelle, also dem Aufwand, um am System teilnehmen zu können. Dieser entsteht zum einen durch das Hinterlegen eines Deposits ergibt und zum anderen, wenn eine Registrierung oder eine Installation nötig ist.

• Der Höhe des Risikos durch das System Verlust zu machen. Das kann geschehen, weil man Opfer eines Angriffes wurde oder weil ein System fehlerhaft oder unzureichend ist. Ein System sollte beispielsweise eine abgesicherte Auszahlung des hinterlegten Deposit ermöglichen.

• Der Höhe des Vorteils durch Einsetzen des Systems. Der Vorteil kann je nach Einsatz variieren. Ein Vorteil sowohl für Sender als auch für Empfänger kann eine geringere Latenz und damit eine schnellere Finalität einer Transaktion sein. Denn umso schneller die Übertragung eines Wertes final ist, umso zeitnaher kann auch die Leistung oder die Ware anschließend erbracht werden. Abschließend sollen die vorgestellten Systeme aufgrund dieser Aspekte verglichen werden.

• Die Kosten und Anforderungen durch Einsatz des Systems. Einige Systeme setzen beispielsweise Public-Key-Verschlüsselungen ein, welche rechenintensiv sind oder erfordern das Ablegen einer Historie, was Speicherplatz benötigt.

Eine schnelle Abwicklung der Zahlungen kann durch zu viele Interaktionen beim Austausch von Zahlungen eingeschränkt werden. Neben einer Kostensenkung pro Zahlung sollte daher auch die Latenz bei den Micropayments gering bleiben und eine Zahlung schnell Finalität erreichen. Bei den meisten vorgestellten Schemata ist dies der Fall.

Der Fakt, dass Channels sowohl für Ethereum, als auch auf Bitcoin umgesetzt wurde, zeigt, wie viel Potenzial in Channels gesehen wird. Die Grundstruktur des Schemas ist simpel und die On-Chain Transaktionen können stark reduziert werden. Dies ist allerdings abhängig von der Dauer der Sender-Empfänger Beziehung ist. In einem Netzwerk von Channels werden daher Vermittler genutzt, damit keine neue Verbindung initiiert werden müssen. Nur das Speichern des Transaktionsverlaufs mindert die Skalierbarkeit. Die Performance ist als sehr gut einzuschätzen, da schnell Finalität erreicht werden kann. Auch Plasma ist ein vielversprechender Ansatz für Micropayments. Wie viel Aufwand der Einsatz dieses Systems mit sich bringt, ist allerdings schwer abschätzbar.

Eine weitere Art von Micropayments sind probabilistische Zahlungen. Diese sollen im folgenden Kapitel vorgestellt werden.

4. Grundlagen probabilistische Micropayments

Probabilistische Zahlungen (Probabilistic Payments) sind ein eigener Ansatz, um den Herausforderungen von Micropayments zu begegnen. Geprägt wurde der Begriff 1996/97 von Wheeler und Rivest. Grundlegend beruht die Idee darauf, Micropayments mit Losen/Tickets zu realisieren. Diese Tickets können mit einer gewissen Wahrscheinlichkeit p ein Gewinn sein und bewirken damit eine große Zahlung. Diese sogenannte Makrozahlung hat dabei einen Wert X. Das Gegenereignis ist eine Niete und tritt mit einer Wahrscheinlichkeit von $1- p$ ein. In diesem Fall wird keine Zahlung durchgeführt, weshalb es auch Nullpayment genannt wird. Auf lange Sicht gesehen und aufgrund des Gesetzes der großen Zahlen hat jedes Ticket einen Erwartungswert von $p*V$. Da nur jedes p-te Ticket zu einer wirklichen Zahlung führt, werden die Transaktionskosten theoretisch um ein p-faches reduziert. Die Zahlungen, deren erwarteter Wert ein Kleinstbetrag ist, werden also durch eine entsprechende Lotterie für größere Zahlungen ersetzt und damit in einer Makrozahlung zusammengefasst.

Dieser grundlegende Ansatz wurde von verschiedenen Personen aufgefasst und darauf aufbauende Schemata erstellt. In den meisten Schemata ist dem eigentlichen Ticketaustausch aus Sicherheitsgründen ein Set-up vorangestellt. Dabei „verbindet“ sich der Sender mit dem Empfänger. Dieses Set-up sollte kostenarm sein, also der Aufwand für das Aufbauen der Beziehung relativ gering. Nur dann ist eine Geschäftsbeziehung, die nur wenige Mikrozahlungen beinhaltet ökonomisch. Dadurch bleibt das Versenden einer beliebigen Menge an Zahlungen an eine beliebige Menge von Empfängern effizient. Dennoch muss insgesamt in einem System regelmäßig Ticketaustausch stattfinden, damit das Gesetz der großen Zahlen gilt. Das bewirkt, dass ein Konsument nicht über- oder unterbezahlt und auch ein Verkäufer entsprechend bezahlt wird. Da probabilistische Micropayments ihre Kostenminimierung durch Effizienzerhöhung über mehrere Zahlungen hinweg erreichen, benötigt es viele Zahlungen in dem System, wo sie eingesetzt werden. Ein System-Ansatz sollte also grundlegend mit einer hohen Kapazität einhergehen.

5. dezentrale probabilistische Mikrozahlungen

MICROPAY1 und 2

In ihrem Dokument „Micropayments for Decentralized Currencies“ von 2016 erläutern Pass und Shelat ihre Ansätze für auf Kryptowährung basierende Micropayments. In der Umsetzung konzentrieren sie sich auf die Bitcoin Blockchain. Ihre Ansätze MICROPAY1, 2 und 3 geben einen Überblick, wie ein Schema für probabilistische Micropayments aussehen kann. Allgemein können auch die meisten dezentralen Schemata in eine Vorbereitung (Set-up), eine Ticketerstellung/-übertragung und ein Einlösen des Tickets eingeteilt werden. Die ersten beiden Ansätze sollen nun im Einzelnen vorgestellt werden.

In MICROPAY1 werden in der Vorbereitung und dem Austausch die folgenden Schritte durchlaufen:

• Der Sender erzeugt zwei Schlüsselpaare, eines für ein Depositkonto mit der Adresse a^{esc} und eines für sein Strafkonto mit der Adresse a^{pen}. Auf das Deposit überträgt er einen gewissen Betrag und auf das Strafkonto ebenfalls. Der Betrag des Strafkonto sollte ein Vielfaches des Betrags des Deposits sein.

• Der Empfänger erstellt eine Zufallszahl $r_1 \leftarrow \{0,1\}^{128}$ und schreibt diese mithilfe eines geheimen Seed s in

ein Commitment $c \leftarrow Com(r_1,s)$. Der Empfänger erzeugt außerdem eine neue Adresse a_2, an welche eine Makrozahlung ausgezahlt werden soll. Er schickt c und a_2 zum Sender.

• Der nächste Schritt ist die Ticketerstellung und Übertragung, welche in MICROPAY1 sehr simpel ist. Der Sender wählt ebenfalls eine Zufallszahl r_2 und erzeugt eine Signatur sig auf c, r_2, a_2 und sendet r_2 und sig zum Empfänger.

• Der Empfänger verifiziert die Signatur und prüft, ob es ein Gewinn ist. Dafür berechnet er die XOR-Verknüpfung von r_1 und r_2. Hat das Ergebnis eine zuvor festgelegte Struktur, gilt das Ticket als Gewinn.

Beide können die Gewinnwahrscheinlichkeit nicht beeinflussen, da beide zum Ergebnis beitragen, ohne den Beitrag des anderen zu kennen. Der Empfänger wählt unabhängig seine Zufallszahl und bindet sich durch das Commitment an sie. Der Sender kennt daher diese Zahl nicht und erzeugt seine Zufallszahl ebenfalls unabhängig davon. Der Empfänger kennt erst nach der Übertragung die beiden Zufallszahlen und kann zu diesem Zeitpunkt seine Zahl nicht mehr ändern.

Dieses Schema kann an unterschiedliche Szenarien angepasst werden, so können beispielsweise die Gewinnhöhe und die Gewinnwahrscheinlichkeit für jedes System neu definiert werden. Trotz oder gerade wegen seiner Einfachheit zeigt das Schema verschiedene Herausforderungen eines dezentralen probabilistischen Micropayments auf. Die im Paper genannten Angriffsszenarien werden folgend kurz erläutert:

• Bei einer Double Spend Attacke wird ein Wert mehrfach ausgegeben, sodass mindestens eine Übertragung nicht gedeckt ist. Dabei können verschiedene Werte gemeint sein. Zum einen kann ein Sender dasselbe Ticket, von dem er nach der ersten Übertragung weiß, dass es kein Gewinn ist, mehrfach ausgegeben. Der Empfänger hingegen könnte ein Winning Ticket mehrfach einlösen wollen.

• Die Overspend Attacke beschreibt das Überziehen des hinterlegten Deposits durch den Sender. Das bedeutet, er gibt mehr aus, als er besitzt.

• Bei einer Front-Running Attacke versucht der Sender, vor der Auszahlung an den Empfänger, selbst eine Auszahlung von dem Konto zu bewirken.

Den Angriffen von der Seite des Senders soll durch das Strafkonto begegnet werden. Das Risiko mehr zu verlieren, als man durch eine Attacke erhalten würde, soll die Angriffe verhindern.

In diesem Schema wird bei Double Spending, also bei Präsentation von zwei Winning Tickets, der Wert im Strafkonto an eine invalide Adresse geschickt und dadurch „verbrannt“. Dadurch erhält auch der Empfänger keinen Vorteil durch das Aufzeigen eines Double Spends. Andernfalls könnte dieser mit dem Einlösen seines Winning Tickets warten, bis er eine Transaktion für das gleiche Konto im Netzwerk sieht und damit ein Double Spend erzwingen. Diese Verhaltensweise wird Waiting Merchant genannt. Das Problem entsteht auch durch den Ansatz, dass der Sender nicht erfährt, ob er gerade ein Winning Ticket versendet hat oder ob er gefahrenfrei weitere Tickets für sein Konto ausstellen kann.

Eine weitere Herausforderung, die außerhalb des eigentlichen Zahlungsverfahrens liegt, ist das Auszahlen des Deposits beziehungsweise des Strafkontos. Dem Sender soll es ermöglicht werden, seine Konten aufzulösen und den hinterlegten Wert wieder zurückzuerhalten. MICROPAY1 behandelt das Auszahlen (withdraw) des Strafkontos nicht im Detail. Es wird der Einsatz einer „locktime“ angesprochen, also einer Zeitspanne, in welcher es dem Sender nicht möglich ist, auf sein Konto selbst zu zugreifen.

Dieses Schema beschreibt gut die Grundstruktur der meisten existierenden Ansätze für dezentralisierte probabilistische Micropayments. Diese Lösung ist allerdings nur bedingt umsetzbar. Zum einen erhält ein betrogener Empfänger nichts, wenn er den Double Spend aufdeckt, was die Akzeptanz der Empfänger senkt. Die Akzeptanz des Senders ist unter anderem abhängig von der Höhe des Strafkontos. Je unprofitabler man den Double Spend machen möchte, um die Sicherheit des Systems zu erhöhen, desto höher muss das Strafdeposit sein.

In MICROPAY2 wird eine dritte Partei in das System eingebracht. Diese gilt als „teilweise vertrauenswürdig“ und wird daher Verifiable Transaction Service (VTS) genannt. Prinzipiell werden alle Tätigkeiten der VTS veröffentlicht und daher wird ein falsches Verhalten schnell entdeckt und dieser bestimmte VTS nicht mehr verwendet. Ein VTS wird genutzt, um Winning Tickets auszuzahlen und er ist verantwortlich für das Strafkonto. Seine Aufgabe umfasst sowohl das Zerstören des Strafkonto als Bestrafung, als auch die Rückzahlung dessen an den Sender. Das Strafkonto, welches der Sender zu Beginn erstellt, benötigt dafür eine Multisignatur 2-von-2 durch den Sender und den VTS. Der Sender erhält eine einseitig unterschriebene Transaktion des VTS zu Beginn und kann diese nach Ablauf einer gewissen locktime nutzen, um sein Strafkonto wieder aufzulösen. Im Gegenzug erhält der VTS nach dem Ticketversand ebenfalls eine einseitig unterschriebene Transaktion des Senders, was ihm das Auszahlen des Strafkontos innerhalb der nächsten k-Blöcke ermöglicht. Der Sender schickt diese Transaktion auch an den Empfänger, damit dieser sie bei Fehlverhalten des Senders auch selbst dem VTS zur Verfügung stellen kann. So ist sichergestellt, dass der Sender bei unerlaubten Verhalten bestraft werden kann.

Der Empfänger benötigt zur Auszahlung des Deposits ebenfalls zwei signierte Transaktionen. Die eine erhält der Empfänger vom VTS, wenn er diesem ein gültiges Winning Ticket zeigen konnte. Die andere erhält der Empfänger vom Sender bei der Ticketübertragung. Auf diese Weise muss ein Ticketaustausch erfolgt sein, bevor der Sender eine Bezahlung erhält, sonst könnte ein böswilliger VTS leicht mit einem Empfänger kooperieren. Der VTS wird bezüglich anderer Angriffsszenarien innerhalb des Schemas grundlegend als eine Art Sicherheitspunkt behandelt.

Er könnte auch aufgrund eines eigentlichen Nullpayment Ticket eine Auszahlung bewirken oder auch anders herum. Die einzige Sicherheitsbegrenzung ist, dass diese Taten nachvollzogen werden können, da der VTS seine Transaktionen auf einer alternativen Chain veröffentlichen muss. Dem Waiting Merchant Problem wird durch eine zusätzliche Interaktion begegnet, in welcher der Empfänger nach Ticketempfang dem Sender die Informationen zur Verfügung stellt, um die Art des Tickets zu erfahren. Verweigert der Empfänger diese Information, so muss der Sender k-Blöcke warten, bevor er wieder Tickets ausstellt und würde diesen Empfänger eventuell vermeiden.

Zusammenfassend ist dies ein etwas umfangreicherer Ansatz als MICROPAY1, dafür wird einigen Sicherheitsproblemen begegnet. Den VTS direkt als Smart Contract umzusetzen ist so nicht möglich, da dieser nicht signieren kann. Jedoch ist seine Funktionalität an sich in einem Smart Contract implementierbar. Mit erhöhter Sicherheit steigt allerdings auch der Aufwand, sowohl On-Chain, als auch Off-Chain. Die drei Parteien interagieren sehr viel miteinander, was aber die Performance des Ansatzes vermindert.

Nach Betrachtung der beiden verschiedenen MICROPAY-Ansätzen wird die größte Herausforderung bei probabilistischen Zahlungsmethoden deutlich - Sicherheit. Vor allem der Angriff durch Double Spend birgt enorme Gefahr. In jedem Schema hat der Sender die Möglichkeit (im großen Stil) eine größere Ticketmenge auszugeben, als er in seinem Deposit deckt. Die Autoren würden in diesem Fall nur den Sender bestrafen, den Empfängern aber keine Sicherheit für eine Auszahlung geben. Ihr Argument ist, dass wenn Sender einen Vorteil durch das Aufdecken von Double Spends hätten, sie dann warten könnten, ihr Winning Ticket zu veröffentlichen, bis sie selbst ein weiteres haben, oder jemand anderes eins veröffentlicht. Dieses Problem muss auch betrachtet werden, dennoch sinkt die Akzeptanz der Empfänger, wenn sie für ein Winning Ticket im Falle eines Double Spends leer ausgehen.

Zusammengefasst liefern diese Schemata die Grundlagen für probabilistische Micropayments mit vielen Denkanstößen, einigen Lösungen und viel Raum für Verbesserungen und Änderungen. Prinzipiell sind sie auf eine dezentrale Lösung ausgelegt, die Umsetzung in Smart Contracts bedarf dennoch einiger Änderungen. Das Paper hat dabei bereits Ansätze auf Grundlage von Bitcoin gefunden und auch Micro-Benchmarks aufgeführt.

CALDWELL

Eine weitere Idee, wie probabilistische Micropayments auf einer Blockchain umgesetzt werden können, wurde 2012 von Mike Caldwell in einem Bitcoin Forum (bitcointalk.org) veröffentlicht. Er bezeichnet es bereits als Ansatz für Nanopayments, da ein winziger Betrag beispielsweise ein zehntausendstel Bitcoin übertragen werden soll. Seine Erklärungen basieren auf Bitcoin, auch wenn sein Ansatz zu dem Zeitpunkt der Veröffentlichung nicht vollumfänglich auf der Blockchain von Bitcoin umsetzbar ist.

Das Set-up umfasst folgende Schritte. Zunächst informiert der Sender den Empfänger, dass er mit ihm interagieren möchte. Daraufhin erzeugt der Empfänger eine neue Bitcoin Adresse, dessen öffentlicher Schlüssel noch ungenutzt, also geheim ist. Diese Adresse teil er dem Sender mit. Der Sender hinterlegt einen Bitcoin in einer TxOut an die mitgeteilte Adresse, welche folgende Bedingungen zur Auszahlung hat: 1. Transaktion muss durch Sender signiert sein. 2. Kenntnis des öffentlichen Schlüssels zur mitgeteilten Adresse und 3. Transaktion muss die Gewinnbedingung erfüllen. Als Gewinnbedingung wird ein Wert Modulo gerechnet, wobei der Divisor die Wahrscheinlichkeit beeinflusst. Ist das Ergebnis null, so gilt die Gewinnbedingung als erfüllt. Der Wert kann beispielsweise eine vom Sender gewählte Zufallszahl sein, welche vom Empfänger signiert wird, sodass er für den Sender nicht vorhersagbar ist.

Als Micropayment schickt der Sender Transaktionen zum Empfänger, welche der Sender signiert hat, wobei die darin mitgeteilte Zufallszahl variiert. Erhält der Empfänger eine Transaktion, welche die Gewinnbedingung erfüllt, leitet er sie zur TxOut und erhält den hinterlegten Bitcoin. Um Front Running durch den Sender zu vermeiden, soll eine locktime genutzt werden. Und der Empfänger soll außerdem die Chain beobachten, um die Ausgabe „seines“ Coins festzustellen und den Dienst für den Sender einzustellen.

Der Ansatz deckt die Anforderungen an ein Micropayment ab. Es bindet allerdings den Empfänger stark an den Sender, da dieser eine Adresse eigens für den Sender erstellt. Außerdem muss der Sender für jeden Empfänger, mit dem er interagieren möchte, einen Bitcoin hinterlegen. Es entsteht somit kein Vorteil gegenüber der Nutzung von State/Payment Channels.

ORCHID

Das Orchid Netzwerk will mit einem dem Tor Browser ähnlichen Prinzip anonymes Internet ermöglichen. Dabei soll der Node, welcher Bandbreite anbietet, kontinuierlich von seinen Nutzern bezahlt werden. Innerhalb des Orchid Netzwerkes wird ein ERC20 Token namens Orchid Token eingesetzt. Allerdings ist dies kein Token im Sinne von Micropayments, sondern hat ausschließlich sozioökonomische Vorteile. Es werden also Orchid Token für die Macropayments genutzt, statt Ether, was aber an der Handhabung nichts ändert.

Die Lösung von Orchid lehnt sich an MICROPAY1 an und wurde auch durch MICROPAY2 und 3 inspiriert. Dabei wurde es insofern abgeändert, dass die Partei, welcher man in diesen Ansätzen vertrauen musste, wegfällt und ihre Funktionalität durch einen Smart Contract abgedeckt wird. Im Netzwerk werden viele Clients mit wenigen Nodes interagieren und die Clients werden unterschiedliche Nodes nutzen. Ein Client-Node-Beziehung gebundenes Set-up hätte Nachteile für beide Seiten. Ein Node müsste Informationen für jeden Client, mit dem er interagiert für eine gewisse Zeit speichern, ohne zu wissen wie lang die Beziehung sein wird. Auch für einen Client ist es unrentabel für jede Interaktion mit einem neuen Node beispielsweise ein neues Deposit zu erstellen.

Das Set-up des Clients ist aus diesem Grund unabhängig vom Node. Hierbei wird ein Deposit und ein sogenanntes penalty escrow (Strafkonto) in einem Smart Contract hinterlegt. Das Deposit wird für die Bezahlungen der Macropayments genutzt und das Strafkonto soll Double Spend unprofitabel machen. Diese Strafkaution wird im Fall eines Overspending verbrannt. Der Ticketaustausch wird folgendermaßen abgewickelt:

Der Empfänger wählt eine Zufallszahl, erstellt einen Hash zu dieser und schickt diesen zum Sender. Der Sender wählt die Werte für Gewinnwahrscheinlichkeit und Gewinnwert, aus denen er ein Ticket erstellt. Im Ticket ist außerdem der Hash der Zufallszahl, ein Zeitstempel und der Hash des Tickets abgelegt. Diesen Hash signiert der Sender mit seinem privaten Schlüssel und diese Signatur wird ebenfalls dem Ticket angefügt. Anschließend schickt der Sender das Ticket zum Empfänger. Dieser verifiziert die Korrektheit des Tickets und überprüft, ob es ein Gewinn ist. Dies ist der Fall, wenn der Hash über den signierten Ticket-Hash und seiner anfangs gewählten Zufallszahl kleiner ist als die Gewinnwahrscheinlichkeit.

Ist es ein Gewinn, wird das Ticket und die Zufallszahl an den Smart Contract übertragen und der Empfänger erhält seine Makrozahlung. Wenn das Deposit für diese Zahlung zu klein ist, wird eine Art Flag, also Zeichen in der Datenstruktur gesetzt, dass die Strafkaution zerstört werden kann. Würde der Sender allerdings Double Spend im großen Stil ausführen, könnte die Strafkaution zu klein sein, um diesen Angriff wirklich unprofitabel zu machen. Um diesem Problem und auch einem Waiting Merchant entgegenzuwirken, hat das Ticket einen Zeitstempel und der Wert des Tickets wird mit der Zeit exponentiell kleiner. Der Empfänger hat also das Ziel ein Ticket möglichst schnell einzulösen. Auf diese Weise würde bei einem Double Spend im großen Stil das Deposit zeitnah zu klein werden und die Strafkaution zerstört werden. Ab diesem Zeitpunkt würde kein Empfänger mehr Tickets von dem Sender entgegennehmen. Dieser Ansatz kann die Attacke nur einschränken und nicht wirklich verhindern. Ob und wie der Empfänger wieder Zugriff auf sein Deposit oder seine Strafkaution erhält, wird im Paper nicht beschrieben.

Zusammenfassend ist dieser Ansatz sehr klassisch, der Ticketaustausch hält sich an den Ablauf aus MICROPAY1 und die Absicherung gegen Angriffe erfolgt durch eine Bestrafung des Täters. Das Paper bietet außerdem eine Analyse über die Performance und stellt die kryptografischen Operationen als Bottleneck (Engstelle) heraus. Es werden verschiedene Maßnahmen zur Reduktion vorgeschlagen. Für den erhöhten Aufwand durch das pre-Ticket Commitment durch den Hash der Zufallszahl wird VRF als Lösung genannt.

VRF steht für Verifiable Random Function und ist eine nachprüfbare Zufallsfunktion. Der Output dieser Funktion soll also nicht vorhersagbar, aber die Richtigkeit überprüfbar sein. Dabei wird asymmetrische Verschlüsselung eingesetzt. Aus einem Input x kann der Besitzer des geheimen Schlüssels SK mit der Funktion y=F(SK, x) berechnen und einen Beweis erstellen. Dabei ist y pseudo-zufällig und jeder kann die Korrektheit mithilfe des Beweises und des öffentlichen Schlüssels überprüfen.

Obwohl es viele nützliche Anwendungen gibt, sind VRF noch nicht ausreichend erforscht und die Umsetzungen oft ineffizient. Auch wenn die EVM inzwischen in der Lage ist, VRF einzusetzen, wurde es noch nicht in Systemen mit erheblichem Wert eingesetzt und die Funktionalität und Sicherheit nachgewiesen. Aus diesem Grund verzichtet Orchid aktuell auf den Einsatz von VRF.

DAM

Decentralized Anonymous Micropayments (DAM) ist der komplexeste Ansatz. Er nutzt verschiedene Unterwährungen und verschiedene Arten der Transaktionen. Darunter befinden sich neben Makrozahlungen, Auszahlungen und Beschwerdemeldungen auch probabilistische Zahlungen. Alle Transaktionen sollen, um wirklich anonym zu sein, keine Information über Herkunft, Ziel oder Betrag von Zahlungen veröffentlichen. Es wird unterschieden zwischen deterministischen Zahlungen, also eines Macropayments, welches non-interaktiv erfolgt und einer probabilistischen Zahlung. Diese basiert auf einem 3-Nachrichten-Protokoll. Das Zahlungsschema wurde durch MICROPAY1 inspiriert und der Ansatz zur Anonymisierung durch DAP (Decentralized Anonymous Payment). Da eine einfache Kombination dieser beiden nicht ausreichend anonymisiert und auch unsicher ist, wurde sie weiterentwickelt.

Die Teilnehmer können drei verschiedene Arten von Coins erstellen - Standard Coins, Deposit-Coins und Tickets. Standard Coins werden genutzt, um Makrozahlungen abzuwickeln und sind die ursprüngliche Währung der Blockchain.

Deposit Coins werden zum Bezahlen der Macropayments bei probabilistischen Zahlungen genutzt und als Strafe eingezogen. Tickets werden für Micropayments eingesetzt und sind immer mit dem Deposit verbunden, welches sie deckt. Mit Mining Transaktionen werden Coins erstellt, welche sowohl übertragen werden können, als auch die Coin-Art wechseln können. Tickets können also erzeugt, ausgetauscht, eingelöst und aktualisiert werden. Außerdem gibt es Auszahlungstransaktionen bei denen Tickets wieder in Standard-Coins umgewandelt werden können. Außerdem kann ein Fehlverhalten gemeldet werden und die dementsprechende Bestrafung eingefordert werden. In dieser Arbeit soll sich auf die probabilistischen Zahlungen, also den Austausch von Tickets konzentriert werden.

Jedes Ticket enthält eine einmalige Kennung (Identifier), eine Gewinnwahrscheinlichkeit und den Wert des Macropayment, sowie Informationen über das verbundene Deposit. Ein Deposit ist initial valide und wird invalide, wenn ein Double Spend bei einem Macropayment auftritt. Ist ein Deposit als invalide markiert, akzeptiert der Empfänger keine damit verbundenen Tickets mehr.

Das 3-Nachrichten-Protokoll des Ticket-Austauschs

läuft wie folgt ab: Die erste Nachricht wird vom Empfänger zum Sender geschickt und beinhaltet einen Session Identifier, den öffentlichen Schlüssel der Session, eine Blacklist an Deposits der aktuellen Periode und den gewünschten Zahlungsbetrag. Der Sender erzeugt nun aus dem Ticket einen Coin und nutzt fraktionierte Nachrichtenübertragung (FMT) für eine probabilistische Nachrichtenübertragung. Neben dem Coin erstellt der Sender zwei weitere entscheidende Größen. Einen Worst-Case-Rate-Limit Tag (wcrlt) und einen Double-Spend-Tag. Mit dem Limit Tag kann der Empfänger eine Grenze für den Zahlungswert erzwingen und mit dem Double-Spend Tag kann er das Deposit erhalten, wenn ein Ticket doppelt in Macropayments ausgegeben wurde. Der Sender erstellt zwei Commitments, eines enthält unter anderem den Ciphertext c und das andere den wcrlt. Dabei sind die beiden insofern verknüpft, dass das Öffnen des ersten Commitments das Öffnen des Zweiten ermöglicht. Er schickt dann die beiden Commitments, den Inhalt des ersten Commitments, den Schlüssel davon und weitere Informationen zum Empfänger. Der Empfänger überprüft alles auf Richtigkeit und versucht, c aus dem ersten Commitment zu entschlüsseln. Wenn er das tun konnte, öffnet er auch das zweite Commitment und veröffentlicht die Transaktion, um sein Macropayment zu erhalten. Stellt er fest, dass das Ticket bereits ausgegeben wurde, erstellt und veröffentlicht er eine Bestrafungstransaktion. Zuletzt kann er dem Sender mitteilen, ob es ein Nullpayment oder ein Macropayment war.

Dieser Ablauf ist sehr theoretisch und beruht auf verschiedene Verschlüsselungsverfahren und ist somit kostenintensiv. Dies ist allerdings nötig, da das ganze Verfahren anonym bleiben soll und keine Informationen nach außen gelangen sollen. Das Konzept wirkt sehr durchdacht, wird aber viel Arbeit benötigen, um wirklich umgesetzt zu werden. Für die Anwendung im IoT-Bereich sind die Anforderungen an den Client zu hoch und die Transaktionen zu groß.

STREAMFLOW

Streamflow ist ein Konzept des Livepeer Protokolls, welches bessere Skalierbarkeit in das Livepeer Netzwerk bringen soll. Dieses Netzwerk besteht aus sogenannten Broadcastern, welche ein Video transcodieren wollen. Ihre Gegenspieler sind Orchestrators, welche segmentweise diese Videos bearbeiten. Dabei entsteht ein ständiger Austausch an Videosegmenten, der aus verschiedenen Gründen jederzeit abgebrochen werden kann. Auf der Suche nach einem passenden Zahlungsschema, welches großteils Off-Chain arbeitet und die Eigenschaften des Netzwerkes nicht aufhebt, haben sie Streamflow erstellt und den Ansatz in einem Paper veröffentlicht.

Streamflow basiert grundlegend wieder auf drei Parteien: 1. dem Broadcaster, welcher der Sender der Zahlung ist, 2. dem Orchestrator, welcher der Empfänger ist und 3. ein Broker, der in diesem Fall ein Smart Contract ist. Neben diesem Smart Contract gibt es noch zwei weitere grundlegende Smart Contracts - einer für die Reserve und ein Manager, der zur Registrierung dient. Diese Funktionalitäten könnten auch in einem Smart Contract zusammengefasst werden, sollen aber für eine bessere Übersicht im Ablauf aufgeteilt werden.

Als Set-up muss sich jeder Sender und jeder Empfänger registrieren. Der Sender erstellt außerdem ein Deposit und eine Reserve. Der Ablauf in Streamflow ist periodisiert, was bedeutet, dass es Runden gibt. Innerhalb einer Runde werden runden-spezifische Tickets erstellt, ausgetauscht und eingelöst. Außerdem wird die Reserve eines Senders virtuell auf alle in der Runde registrierten Empfänger aufgeteilt. Auf diese Weise ist jedem Empfänger ein Teil der Reserve eines Senders zugesichert, auch wenn der Sender eine Double Spend Attacke macht. Zunächst soll allerdings der Ticketaustausch erläutert werden.

Der Sender fragt den Empfänger nach den Ticket-Parametern. Zu diesen Parametern gehören Gewinnhöhe, Gewinnwahrscheinlichkeit und ein Commitment zu der Zufallszahl des Empfängers. Der Empfänger hat aufgrund dieser Anfrage des Senders die Möglichkeit den Sender zu überprüfen. Darauf basierend kann er entscheiden, ob er mit diesem Empfänger arbeiten möchte. Ist dies der Fall, schickt er ihm die gewünschten Parameter. Der Sender nutzt diese und erstellt daraus ein Ticket. Im Ticket stehen außerdem die aktuelle Rundennummer und der dazugehörige Runden-Hash.

Der Sender berechnet den Hash des Tickets und signiert ihn. Die Signatur, den Hash und das Ticket schickt er an den Empfänger. Dieser prüft alles auf Richtigkeit und ob es ein Winning Ticket ist. Das wird auch in diesem Schema durch eine Hashfunktion über die Signatur des Senders und die Zufallszahl des Empfängers geprüft. Ist dieser Hash kleiner als die Gewinnwahrscheinlichkeit, führt dieses Ticket zu einem Macropayment. In diesem Fall ruft der Empfänger eine Funktion im Broker Smart Contract auf, welcher er das Ticket, die Signatur und seine Zufallszahl übergibt. Der Broker überprüft alles und transferiert den entsprechenden Betrag vom Deposit des Senders an die Adresse des Empfängers.

Ist der Empfänger Opfer eines Double Spends, wird die Auszahlung der Reserve an den Reserve Smart Contract übergeben. Dieser prüft, ob der Empfänger für die aktuelle Runde registriert ist. Ist dies der Fall, prüft er wie hoch die Reserve des Senders ist und teilt diese auf alle registrierten Empfänger der Runde auf. Er prüft, wie viel dem Empfänger bereits zugesichert wurde und erhöht diesen Anteil entsprechend der Höhe des Gewinns und abhängig vom Anteil, den er bekommen darf. Am Ende der Runde bekommen die Empfänger die ihnen zugesicherten, eingeforderten Reserve-Anteile ausgezahlt.

Beim ersten Zugriff auf die Reserve eines Senders wird diese außerdem als eingefroren (freeze) gekennzeichnet. Dafür wird die aktuelle Runde als Freeze-Round, also die Runde, in der sie eingefroren wurde, gespeichert. Für eine gewisse Rundenanzahl (Freeze Periode), kann die Reserve durch den Sender weder aufgefüllt noch ausgezahlt werden. Eine andere Rundenanzahl (Unlock Periode) wird genutzt, um Auszahlungen durch den Sender zu ermöglichen.

Möchte er sein Deposit verringern oder an sich selbst auszahlen, so wird ebenfalls die Rundennummer verwendet. In diesem Fall wird die Runde notiert, ab welcher der Sender in der Lage ist, auf sein Deposit zuzugreifen. Der Empfänger überprüft diese Information, bevor er einem Empfänger die Ticketparameter zur Verfügung stellt. Auf diese Weise kann ein Empfänger keine Front Running Attacke durchführen.

Das Schema kann sehr variabel an seinen Einsatz angepasst werden. Zum einen kann entschieden werden, wie lange ein Ticket gültig sein soll und wie lang die Freeze- und die Unlock-Periode sind. Zum anderen kann die Anzahl der Empfänger, die auf eine Reserve Zugriff haben reduziert werden. In diesem Fall würde der Sender ein Set erstellen, in welchem er einer Auswahl an Empfängern ihren Teil der Reserve garantiert. Auf diese Weise müsste die Größe der Reserve nicht mit der Anzahl der registrierten Empfänger steigen, sondern könnte variabel reguliert werden. Das Schema liefert damit eine gute Sicherheit und eine umfassende Lösung für Double Spend.

Dies ist allerdings mit einem höheren Aufwand verbunden. Zum einen müssen sich alle Beteiligten registrieren und die Smart Contracts sind recht umfangreich. Zum anderen ist die Interaktion zwischen Sender und Empfänger auf drei Nachrichten erhöht, was auch eine höhere Latenz mit sich bringt. Die Skalierbarkeit ist dennoch gut, da keine Beziehung zwischen Sender und Empfänger erstellt werden muss, der Empfänger muss lediglich für die Reserve registriert sein. Gegen eine Replay Attacke ist die einzige genannte Lösung, das Aufzeichnen und Speichern der Hashs der genutzten Winning Tickets. Der Speicheraufwand dafür ist allerdings begrenzt, da mit Ablauf der Periode, in der sie gültig sind, die Hashs nicht weiter gespeichert werden müssen. Der Ansatz ist komplex, aber gut durchdacht und es liegen bereits Entwürfe der Smart Contracts und des Codes von Empfänger und Sender vor.

POOLLÖSUNG

In einem System, in welchem Zahlungen abgewickelt werden, gibt es oftmals eine Unterteilung in Sender (Client) und Empfänger (Server). Eine Idee, um Zahlungen weiter zusammenzufassen, beruht darauf, dass nicht ein Client einen Server bezahlt, sondern sich Gruppen (Pools) bezahlen. Das kann in einem 1-zu-n Schema umgesetzt werden, also eine Gruppe interagiert mit einem einzelnen. Es kann aber auch in einem n-zu-m Schema geschehen, bei dem eine Gruppe von m Teilnehmern an eine Gruppe von n Teilnehmern Zahlungen durchführt.

Das Absichern, wer wie viel eingezahlt hat und wer wieviel erhält, kann durch eine zentrale Instanz abgesichert werden. Die Sender Gruppe (Pool S) zahlt in ihren Pool ein und es wird zentral vermerkt, wer wie viel eingezahlt hat. Der Empfänger-Pool (Pool E) hat ebenfalls eine zentrale Instanz, die Information darüber hat, welchem Empfänger wie viel Anteil des Pools zusteht. Aus vielen kleinen Zahlungen von Mitgliedern aus Pool S an Mitglieder aus Pool E entsteht eine große Zahlung von Pool S an Pool E. Diese wird dann intern durch den Manager des Pool E aufgeteilt. Würde man diesen Ansatz beispielsweise mit probabilistischen Zahlungen umsetzen, erinnert es an eine Lotto-Tippgemeinschaft. Es werden also mehrere Tickets in einem Pool gesammelt, eingelöst und sollte eines der Tickets gewinnen, wird der Gewinn auf die Teilnehmer des Pools aufgeteilt. Diese Aufteilung könnte von der Anzahl an Lottoscheinen, die der Teilnehmer dem Pool beigesteuert hat, abhängig gemacht werden.

So könnten sich Nodes zusammenschließen und regelmäßiger Gewinne bekommen. Und vor allem können Nodes, die nur eine geringe Anzahl an Tickets erhalten, gemeinsam ihre Chance auf einen Gewinn erhöhen und eine Zahlung erhalten. Bei dieser Anwendung müsste ein zentraler Verwalter, dem alle vertrauen müssen, allerdings entfallen. Stattdessen müsste auf die zur Verfügung stehenden kryptographischen Mittel und Datenstrukturen oder Smart Contracts zurückgegriffen werden. Ein System, welches darauf aufbaut, ist Cardstack.

6. Fazit

Abschließend sollen die wichtigsten Erkenntnisse aus dieser Arbeit zusammengefasst werden. Zu Beginn der Arbeit wurden drei zentrale Themen genannt, die in dieser Arbeit vereint werden sollen: Blockchain, IoT und (Mikro-)Zahlungen. Davon ausgehend wurde nach vorhandenen Micropayment Ansätzen insbesondere probabilistischen Micropayment-Konzepten recherchiert. Das Ergebnis ist eine Übersicht über die verschiedenen Arbeitsweisen, Vorteile und Herausforderungen der Ansätze.

Es wurden verschiedene Anforderungen für ein geeignetes Micropaymentsystem definiert und für verschiedene Ansätze analysiert:

• Ablauf ohne eine zentrale Instanz - Viele Konzepte bauen auf ein zentrales System auf. Eine praktikable Anpassung dieser für eine dezentrale Umgebung ist nur bei wenigen möglich. Die Recherche ergab jedoch auch, dass bereits Ansätze für dezentrale Systeme beziehungsweise Blockchain existieren.

• Ressourcenschonend, um den Einsatz in IoT-Geräten zu ermöglichen - Diese Anforderung steht bei den meisten Systemen nicht im Fokus. Dennoch bieten einige Ansätze einfache Abläufe, die dem Sender ein Arbeiten mit begrenzten Ressourcen ermöglicht.

• Absicherung des Systems gegen verschiedene Angriffsszenarien - Inwiefern die Systeme gegen die unterschiedlichen Angriffe abgesichert wurden, unterschied sich stark. Einige zentrale Ansätze überlassen den Sicherheitsfaktor der Bank, andere komplexe Systeme konnten ein sehr hohes Maß an Sicherheit bieten. Eine Herausforderung ist es, Sicherheit und Ressourcenschonung in einem System zu vereinen.

• Verringerung der Kosten pro Zahlungen durch den Einsatz des Systems - Eine Kostenreduktion pro Mikrozahlung ermöglichen die meisten Konzepte. Eine Einschränkung ergibt sich dabei teilweise durch hohe Kosten beim Aufsetzen des Systems.

Beim Vorstellen der verschiedenen Lösungen wurden einige Herausforderungen der Dezentralisierung

und im Besonderen der Blockchain festgestellt. Von den vorgestellten Ansätzen haben nur zwei eine wirkliche Umsetzung in Form von programmierten Smart Contracts erfahren. Diese beiden sind Orchid und Streamflow. Eine zentrale Lösung auf die Blockchain zu übertragen, bedeutet neben großem Aufwand auch den Verlust des Charakters der Lösungen, die auf eine Bank als zentrale Einheit setzen. Viele zentrale Lösungen haben das Ziel, die Bank zu entlasten, während die dezentralen Lösungen das Ziel haben die Transaktionen auf Blockchain zu minimieren. Beim Verringern der Transaktionen auf der Blockchain profitieren auch die beiden anderen Parteien davon, durch die Einsparung an Transaktionskosten. Doch die Verringerung des Einsatzes von Blockchain oder Banken erhöht auch den Aufwand bei Sender und Empfänger. Sie müssen selbst prüfen, ob alles valide ist und dadurch steigt der Rechenaufwand.

Auch der Aufwand ein System so weit zu entwickeln, dass es eingesetzt werden kann, sollte beachtet werden. Orchid und Streamflow sind komplexer als die MICROPAY-Ansätze, gerade weil sie in vollem Umfang umgesetzt wurden. DAM ist bereits in der Theorie sehr umfangreich und würde viel Aufwand in der Entwicklung bedeuten. Der Ansatz von Caldwell scheint überschaubar, dennoch wird bei Micropayments für Bitcoin zunächst bitcoinj genannt, welches mit Payment Channels arbeitet.

Neben den Angriffsmöglichkeiten kann die menschliche Psyche ein weiteres großes Problem für probabilistischen Micropayments sein. Bei heutigen Angeboten existieren oft sowohl Abo-Tarife, als auch Payper-Use-Tarife und die Tendenz der Konsumenten geht laut empirischen Studien zum Abo-Tarif. Verschiedene psychologische Effekte bewegen den Konsumenten teilweise dazu, die unökonomische Entscheidung für die Flatrate zu fällen:

• Überschätzungseffekt: Der Konsument überschätzt die tatsächliche Nutzung in der Zukunft.

• Versicherungseffekte: Der Konsument kann sicher sein, dass er nicht mehr zahlen wird, als den Abo-Beitrag.

• Taxametereffekt: Durch kontinuierliche, kleine Zahlungen „verliert“ der Konsument immer wieder erneut Geld.

• Bequemlichkeitseffekt: Konsument könnte viele einzelne Transaktionen als aufwendig empfinden.

Vor allem der Versicherungseffekt wird beim Einsetzen von probabilistischen Micropayments verstärkt. Ein Konsument kann davor zurückschrecken, nicht kontrollieren zu können, wann und wie häufig er bezahlen muss. Trotz der rationalen Sicherheit durch das Gesetz der großen Zahlen, kann die Angst überwiegen und das Risiko, überzubezahlen als zu groß erscheinen. Aus der Sicht der menschlichen Psyche spricht also einiges gegen Micropayments und im Besonderen gegen wahrscheinlichkeitsbasierte Zahlungen. Auch wenn eventuell letztendlich nur IoT-Geräte interagieren, so muss das System durch Menschen eingesetzt werden, welche davon überzeugt sein müssen.

Betrachtet man die vorgestellten Ansätze aus Sicht von IoT-Geräten wird ein weiteres Problem klar: Latenz. Beim Ticketaustausch interagieren in den Ansätzen Sender und Empfänger miteinander. Bis zu drei Nachrichten werden pro Zahlung ausgetauscht. Soll allerdings beispielsweise ein Sensor in kurzer Zeit mehrere Werte in die Blockchain schreiben, so ist die Latenz durch die hohe Interaktion zu groß. Die Interaktionen im Austausch sind jedoch aus Sicherheitsgründen in den Ansätzen nötig.

Auch die Größe der auszutauschenden Nachrichten kann kritisch für ein IoT-Gerät mit geringer Performance werden. Dazu kommen kryptografische Berechnungen, die Rechenkapazität benötigen und gegebenenfalls die Latenz erhöhen. In mehreren Publikationen werden VRF als Möglichkeit genannt, um den Austausch non-interaktiv zu gestalten. Doch dieser Ansatz ist noch nicht ausreichend getestet und vor allem auf der Blockchain mit einigen Schwierigkeiten verbunden.

Diese Arbeit zeigt, dass vorhandene probabilistische Micropaymentsysteme aktuell auch für IoT-Geräte auf der Blockchain eingesetzt werden könnten. Es konnten viele Ansätze und ihre Vorteile ausgearbeitet werden. Vor allem der Ansatz der Poollösung bietet aus aktueller Sicht Potenzial und könnte weiterverfolgt werden. Eventuell bringt die anhaltende Weiterentwicklung der Blockchain-Technologie von sich aus effizientere und günstigere Zahlungen mit sich und macht damit den Einsatz spezieller Micropaymentverfahren obsolet.

Danksagung

Die Autorin bedankt sich für die Unterstützung bei Slock.it GmbH (jetzt BLOCKCHAINS, LLC) für die Betreuung der zugrundeliegenden Bachelorarbeit.

Literaturverzeichnis

[1] Ouellette, A.: These Are The 17 Top Tech Buzzwords You Need To Know, (2019). https://careerfoundry.com/en/blog/web-development/tech-buzzwords-to-learn/, abgerufen am 28.08.2020

[2] Gartner: Hype Cycle for Emerging Technologies, (2018). https://blogs.gartner.com/smarterwithgartner/files/2018/08/PR_490866_5_Trends_in_the_Emerging_Tech_Hype_Cycle_2018_Hype_Cycle.png, abgerufen am: 12.09.2019

[3] Sepp, C.; Brzoska, M.: Die Geschichte des Geldes: Von der Muschel zur Kreditkarte, (2015). https://www.br.de/radio/bayern2/sendungen/radiowissen/soziale-politische-bildung/geld-geschichte-100.html, abgerufen am: 12.09.2019

[4] Dai, X. u. Grundy, J.: NetPay Micro-Payment Protocols for Three Networks. In: Badr, Y., Chbeir, R., Abraham, A. u. Hassanien, A.-E. (Hrsg.): Emergent Web Intelligence: Advanced Semantic Technologies. Advanced Information and Knowledge Processing. (2010), S. 429–449.

[5] Chi, E.: Evaluation of micropayment schemes, (1997). https://www.hpl.hp.com/techreports/97/HPL-97-14.pdf, zuletzt geprüft am 12.06.2019.

[6] Odly, A.: The Case Against Micropayments, (2003). http://www.dtc.umn.edu/~odlyzko/doc/case.against.micropayments.pdf, zuletzt geprüft am 02.05.2019.
[7] Decker, C.; Wattenhofer, R.: A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels, (2015).
[8] Horne, L.: Generalized State Channels on Ethereum – L4 blog – Medium, (2017). https://medium.com/l4-media/generalized-state-channels-on-ethereum-de0357f5fb44, abgerufen am: 07.05.2019
[9] What is the Raiden Network?, (2019). https://raiden.network/101.html, abgerufen am: 03.07.2019
[10] Butler, A.: An introduction to Plasma – Hacker Noon, (2018). https://hackernoon.com/plasma-8bba7e1b1d0f, abgerufen am: 02.05.2019
[11] Horne, L.: What is Plasma? Plasma Cash?, (2018). https://medium.com/crypto-economics /what-is-plasma-plasma-cash-6fbbef784a, abgerufen am: 07.05.2019
[12] Poon, J.; Buterin, V.: Plasma: Scalable Autonomous Smart Contracts, (2017). https://plasma.io/, abgerufen am: 02.05.2019
[13] µRaiden: Micropayments for Ethereum – Hacker Noon, (2017). https://hackernoon.com/%C2%B5raiden-micropayments-for-ethereum-f0756cd400b3, zuletzt geprüft am 26.06.2019.
[14] Pass, R.; Shelat, A.: Micropayments for Decentralized Currencies, (2016).
[15] Simonsson, G.: Ethereum Probabilistic, (2017). https://medium.com/@gustav.simonsson/ethereum-probabilistic-micropayments-ae6e6cd85a06, abgerufen am: 02.05.2019
[16] Nanopayments - Bitcoin Wiki, (2018). https://en.bitcoin.it/wiki/Nanopayments, abgerufen am: 19.07.2019
[17] Salamon, D. L., Simonsson, G., Freeman, J., Fox, B. J., Vohaska, B., Bell, S. F.; Waterhouse, S.: Orchid: Enabling Decentralized Network Formation and Probabilistic Micro-Payments. https://www.orchid.com/assets/whitepaper/whitepaper.pdf, zuletzt geprüft am 02.05.2019.
[18] Caldwell, M.: Sustainable nanopayment idea: Probabilistic Payments, (2012). https://bitcointalk.org/index.php?topic=62558
[19] Dodis, Y. u. Yampolskiy, A.: A Verifiable Ran dom Function With Short Proofs and Keys, (2005). https://cs.nyu.edu/~dodis/ps/short-vrf.pdf, zuletzt geprüft am 21.08.2019.
[20] Goldberg, S.; Papadopoulus, D.: Verifiable Random Functions (VRFs), (2017). https://tools.ietf.org/id/draft-goldbe-vrf-01.html, abgerufen am: 15.05.2019
[21] Chiesa, A., Green, M., Liu, J., Miao, P., Miers, I. u. Mmishra, P.: Decentralized Anonymous Micropayments, (2016). https://eprint.iacr.org/2016 /1033.pdf, zuletzt geprüft am 02.05.2019.
[22] livepeer/go-livepeer. https://github.com/livepeer/go-livepeer/blob/master/cmd/livepeer/livepeer.go, abgerufen am: 08.05.2019
[23] Fu, Y.: Streamflow: Probabilistic Micropayments, (2019). https://medium.com/livepeer-blog /streamflow-probabilistic-micropayments-f3a647672462, abgerufen am: 02.05.2019
[24] Fu, Y.; Vergauwen, N.: Probabilistic Micropayments, (2019). https://github.com/livepeer/wiki/blob/master/spec/streamflow/pm.md, abgerufen am 28.08.2020
[25] Lotto-Tippgemeinschaft: gemeinsam Lotto spielen. https://lotto.web.de/tippgemeinschaften/, abgerufen am: 02.09.2019
[26] Cardstack White Paper - The experience layer of the decentralized Internet, (2018). https://resources.cardstack.com/whitepaper/resources/vision-paper-and-technical-paper, zuletzt geprüft am 28.08.2020
[27] Abdel-Rahman, H.: Scalable Payment Pools in Solidity. Paying a lot of people without paying a lot of gas, (2018). https://medium.com/cardstack/scalable-payment-pools-in-solidity-d97e45fc7c5c, abgerufen am: 02.09.2019
[28] bitcoinj, (2019). https://bitcoinj.github.io, abgerufen am: 28.08.2020
[29] Robbert, T., Priester, A. u. Roth, S.: Micropayments im Erlösmodell digitaler Serviceleistungen, (2018). In: Bruhn, M. u. Hadwich, K. (Hrsg.): Service Business Development. Wiesbaden: Springer Fachmedien Wiesbaden 2018, S. 187–209
[30] Poser, M.: Probabilistische Mikrozahlungen auf der Blockchain, (2019).

REDACTABLE BLOCKCHAIN – LEVERAGING CHAMELEON HASH FUNCTIONS FOR A GDPR COMPLIANT BLOCKCHAIN

Hauke Precht, Jorge Marx Gómez
Carl von Ossietzky Universität Oldenburg, Ammerländer Heerstraße 114-118, 26129 Oldenburg

With the increasing usage of blockchain technology, legal challenges such as GDPR compliance arise. Especially the right of erasure is considered challenging as blockchains are tamperproof by design. Several approaches investigated possibilities to weaken the tamperproof aspect of blockchains in favor of GDPR compliance. This paper presents several approaches, then focuses on chameleon hash functions by evaluating the possibility to use these specific functions in a private blockchain. The goal of the built system is to take a step towards the digitization of the bill of lading used in international trade. This paper describes the developed software as well as the core considerations around the system such as network design or block structure.

1. Introduction

As the blockchain technology is used in more and more domains, also the privacy aspect of the saved data on the blockchain draws attention. As one key feature of blockchain is the immutability of data, it clashes with General Data Protection Regulation (GDPR) requirements, i.e. the right to erasure. This must be granted if personal data is processed. As this right cannot be enforced on a blockchain, since the data is stored in immutable and linked blocks, it is currently not possible, from a legal point of view, to use blockchain technology in conjunction with personal data.

As it is not always clear if data is classified as personal data, often only hash values of actual data are stored on the blockchain. The drawback of this is that data still needs to be distributed in an old-fashioned way leading to (manual) processes of data handling around the blockchain.

The research project HAPTIK (haptik.io) makes an attempt to digitize the bill of lading via blockchain technology, which was already proposed in [1]. As the goal is to digitize the whole process around the handling of the bill of lading, an investigation was carried out whether personal data is part of a bill of lading. This led to a positive result, meaning that the GDPR must be applied [2]. Based on this result, several approaches as proof of concept were carried out, of which one is described in this paper in detail: leveraging chameleon hash functions to create a redactable blockchain, allowing the users to modify already accepted and verified blocks.

The design and prototype were developed with four students in a project group for over one year. This is part of every master students' curriculum to let them learn to work in a team and to contribute to ongoing research topics.

The following paper is structured as follows: First, an overview of related work is given, in which already existing approaches, towards a redactable blockchain, are presented, along with a short introduction of chameleon hash functions. Next, the implementation of the prototype is presented, where key design decisions are shown along with the key functions of the newly created blockchain. This paper concludes with a summary and outlook identifying further research aspects

2. Related Work

In 2017, Ateniese et. al proposed a framework enabling a redactable blockchain by using chameleon hash functions [3]. They describe a theoretical framework and a proof of concept implementation for the Bitcoin network [3]. The used Chameleon Hash functions were introduced in 2000 by Krawczyk and Tal [4]. These functions also generate a "trapdoor" key, which can be used to efficiently generate hash collisions. This is also used by Ateniese et. al with some enhancements [3]. The idea is to use this trapdoor key to calculate a collision. When redacting a block, it gets rehashed by using the trapdoor key which enables the function to calculate a collision. That way a block can be changed while maintaining its original hash. In comparison, classical hash functions aim to be collision-resistant [5]. As Ateniese et. al focuses mainly on the applicability of chameleon hash functions in bitcoin, while discussing private/consortium blockchains only shortly, we, in this paper, aim towards an evaluation of applicability in private/consortium blockchains. Especially the management of the trapdoor key is crucial as it is used to modify the block, meaning profound governance of this trapdoor key is required [3]. Next to this generally more abstract framework, Hylock and Zeng presented an application based on this framework in the specific domain of patient-centered health records [5] dealing with highly personal data. Next to using non-traditional hash functions, such as chameleon hash functions, a different approach towards a redactable blockchain is presented by Marsalek and Zefferer. These authors propose an architecture in which a second blockchain is used to track corrected blocks, named correction chain [6]. While these approaches act on a global scale, Florian et. al propose an erasing mechanism on the local scale, i.e. on node level by presenting the functionality-preserving local erasure [7]. They provided a prototype based on bitcoin as well, arguing that their solution is less invasive than, for example, the solution proposed by Ateniese et. al [7]. Another different approach is presented in [8] where the

authors propose a system leveraging multiple transaction versions representing different possible states while also encrypting non-active transactions. When a change is desired, a consensus about the new active transaction is triggered [8]. The idea is to plan possible (desired) ledger states, including no-operation, beforehand. All transaction versions are encrypted and only the key for the currently active version is distributed [8]. This way, no “real” deletion or changing of data is possible, while this approach also targets PoW-Blockchains, i.e. dealing with currency and coin amounts in wallets. Farshid et. al also proposed a “forgetting blockchain”, where they designed and implemented a pruning algorithm for the Ethereum blockchain, which deletes as much data as possible while maintaining consistency [9]. The drawback of this method is, that it is not possible to delete specific data but rather rely on the pruning algorithm to take care of it eventually.

This recap of existing approaches considering the ability to redact in blockchain shows, that this is an emerging and ongoing research topic. As it is a considerably new approach, only one paper was identified, applying a redactable blockchain to an actual real-world use case. Further, the proposed solution focuses on public blockchain such as bitcoin.

In this paper, the authors investigate, if a redactable blockchain, based on chameleon hash functions, is suitable for private/consortium blockchains for possible digitization of a bill of lading.

3. Implementation of a Chameleon Hash Function Based Blockchain

The general applicability of chameleon hash functions in private/consortium blockchains is already, shortly, discussed by Ateniese et. al, considering the management of the trapdoor key [3]. As mentioned, the usage of a chameleon hash function is invasive [7] in regards to the used blockchain, meaning, when using an existing blockchain system, the existing hash function of this blockchain must be replaced. As this requires extensive changes in the underlying code and therefore profound knowledge of this code, it can be considered challenging. The group of students first analyzed such exchange of hash functions in the consortium blockchain Corda, but after one month without significant progress decided to refrain from trying to adjust the codebase. The main reason was insufficient code documentation and time constraints. Therefore, it was decided to create a own implementation of a (simplified) blockchain system, incorporating chameleon hash functions from the beginning and building the system tailored to alternating block in the easiest way towards the use case example of a digital bill of lading.

As the most knowledge regarding programming languages were at C#, this language was chosen for the implementation. To provide the user with a user interface, the React framework was used to build a web app. The actual data is stored in a LiteDB, which is an integrated database. This way, a three-layer model is built:

- First Layer: Webapp, which servers the user for input of data
- Second Layer: Service, which processes data from user input and serves data from the blockchain to the user
- Third Layer: Blockchain, providing the necessary hash functions, assets and verifying of new blocks (consensus).

The following paper will mainly focus on the blockchain layer by shortly describing the implementation of the chameleon hash function followed by the network design, consensus algorithm as well as the defined block structure. Note, that the focus lies solely on the chameleon hash functions and its possible applicability. Therefore, no mechanism for smart contracts, living on the blockchain, was implemented yet, to further minimize complexity in the first step.

3.1. Chameleon Hash Function

Due to the complex theoretical and mathematical nature of a chameleon hash function, no own implementation was used. Instead, an existing C library [10] was used and translated into C# due to compatibility issues. It is notable, that this was, at the time of development, the only library found, explicitly focusing on an implementation of chameleon hash functions. Note, that three different approaches towards a chameleon hash function were initially implemented in the library which was used for performance measurements, comparing block creation time and block redaction time. In their implementation, they evaluated three approaches of implementation: Simple Factorization (SF), Discrete Logarithm (DL) and Advanced Factorization (AF) [11]. The third approach, AF, showed similarities in block creation as well as in calculating a collision, which is why this algorithm was chosen in this prototype as well.

3.2. Network Design

Within the process of the bill of lading, the involved participants are generally known, especially parties like ports, customs, port agents and so on and so forth. As a private blockchain is built, each of the identified participants will be represented by an own node. Since not every party is included in every bill of lading process, they only receive the data required for processes they are part of (privacy by design). To achieve this, the system supports a set of blockchains instead of a single, large blockchain. A similar concept is known from Hyperledger Fabric where the system allows to create distinct channels, each holding an own blockchain [12]. This way, shared data among the parties are limited to a minimum, following privacy by design principles. The general idea of the network infrastructure is shown in figure 1 below.

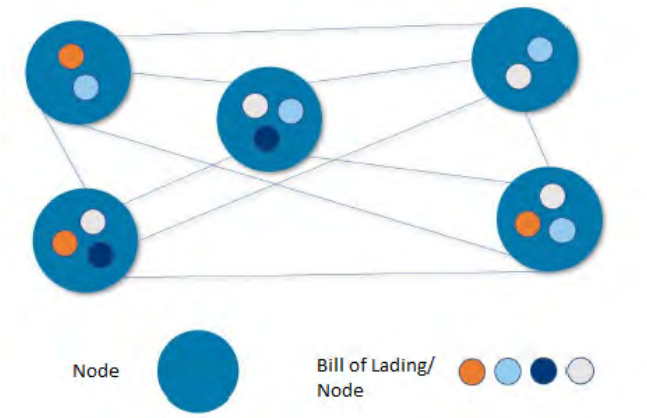


Fig. 1: Network Infrastructure

3.3. Consensus Algorithm

Since in private/consortium blockchains, the participants are generally known and access to the network is restricted, energy-consuming consensus algorithms like proof of work are not required. Instead, voting-based consensus algorithms are used [13], enabled by the identifiable parties which is a requirement, for example for the practical Byzantine Fault Tolerance (pBFT). This algorithm, which was already defined in 1999 by Castro and Liskov [14], proposes a solution to The Byzantine Generals Problem [15]. Within this prototype, a pBFT-like consensus algorithm was implemented as well requiring a 2/3 majority to finally accept actions. As the number of participants is considerably low, the possible slow performance can be neglected.

3.4. Block Structure

To this point, the general architecture of the developed blockchain is described, including used technology, network infrastructure and the used consensus algorithm. Next, the actual structure of a block is discussed.

For a starting point, the block structure of the most known blockchains, Bitcoin and Ethereum, is analyzed, starting with the Bitcoin block structure. It mainly consists of 5 objects: Magic number, block size, block header, transaction counter and a variable set of transactions [16]. These objects can be further divided into more detailed variables, for example, is the previous hash a variable in the block header [16]. A similar structure is used in the Ethereum blockchain, where the block header contains even more information in regard to Ethereum specific gas [17]. As a private blockchain is developed, also the Hyperledger Fabric block structure is taken as a reference into consideration as well. A Hyperledger fabric block consists of three sections: block header, block data and block metadata [18]. Like the block structures of bitcoin and Ethereum, the block header contains information such as the previous hash or the block number. The block data is also similar, containing transaction data. The block metadata, however, is used to store the certificate and signature of the creator of the block [18]. Block committer also adds valid/invalid indicators for each transaction into a bitmap. Note that the metadata is not taken into consideration for the block hash computation [18]. An abstraction of a block structure is given in [13] where the authors defined two major parts which every block structures have in common: the header, storing metadata such as the timestamp or hash of the previous block, and the block content, storing actual (transaction) data.

Based on the different existing block structures, a similar approach is developed in this prototype meaning that block metadata must be stored as well as the actual data. To simplify the structure in this prototype as much as possible, no distinct header object is modelled but typical header information is included in the block. As chameleon hashes are used, a block must also store a checksum as well as the public/trapdoor key pair. Note that the actual trapdoor key is only present on the node, which created the block. Therefore, also the creator of the block is modelled in the block structure for other parties to know whom to contact for a possible redact request. Furthermore, each involved node must sign the block. As a pBFT-like algorithm is implemented, a block requires 2/3 acceptance (by signing the new block) which also holds true when redacting a block. To be able to include the redacting of a block as part of the consensus, the mentioned checksum is included. In case the creator of a block changes the block without an existing request for his own good, it would be noticed by the other participants as the checksum changes. Further, the other nodes must manually accept any given redaction action, providing another layer of security preventing arbitrary changes by the creator. The newly created block structure in general block structure is shown in figure 2.

Fig. 2: Block Structure

In this given use case, the data of the block represents the parts of the bill of lading. Note that each block only contains newly added data, so the bill of lading will be constructed by reading and merging the data of each block since the calculation of a collision is time-consuming. Therefore, the goal is to minimize the number of blocks which require a change.

4. Block Creation and Redaction Walkthrough

With the described system, a private blockchain leveraging chameleon hash function is built. Next, the

process of block creation, as well as the process of redacting a block, is described, starting with the creation of a new block, which is shown in figure 3. In a first step, the initiating node (source node) sends the new block for verification to its known partners. The nodes evaluate the block and send either a verification signature or a rejection. The source node must collect 2/3 of verification signatures to consider the votes as an acceptance, then sending the newly accepted and signed block to the network. If this threshold is not reached, i.e. no majority voted for acceptance, no block will be created.

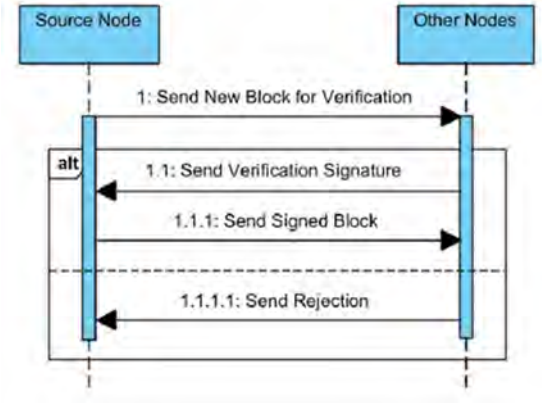


Fig. 3: Block Creation Process

Note that the accepted and signed block is sent to the other nodes, who then also verify that 2/3 majority signatures are present. This is possible as every node knows the other parties. The other nodes then add the new block to their own blockchain.

If the need to change data in an already accepted and appended block appears, the redaction function comes into play. The process is shown in figure 4 and is explained in the following. In the first step, the node who requests a change of data notifies the owner node (i.e. the node which created the respective block as this is the only one in possession of the trapdoor key) by sending a redact request. The node performs the requested change by modifying the block and rehashing the block via the chameleon hash function. The redacted block is sent to the other participants of the process for manual verification. At this point, it is up to the users to decide if a change is justified and allowed. Like the block creation process, 2/3 of the participants must accept the change by manually accepting the request via the provided web app. In case they decline the request, the changed block will be discarded, and a reject request message is sent to the owner node. The owner node verifies the signed verification/rejection signatures and in case a 2/3 majority of acceptance is reached, the newly redacted and signed block is sent to the network. In case no 2/3 majority is reached, the changes are discarded, and a rejection notification is sent to the requesting node. As this process involves the manual action of users, waiting time must be taken into consideration as well as possible timeouts. To deal with these issues, a request object is stored each time a block is sent, storing information of the state of the request. This serves as a basis to determine a possible timeout in which the nodes must answer. If this timeout is reached, the request is sent again. This is done until every node has sent either an accept or reject.

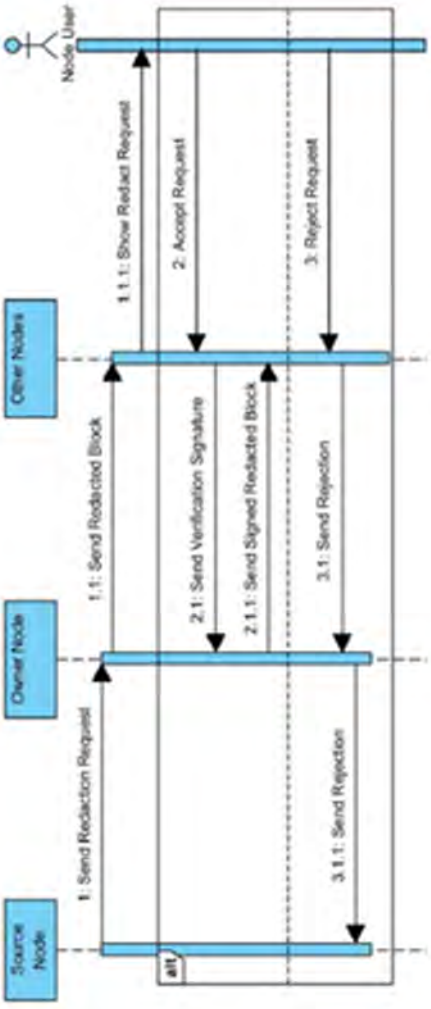


Fig. 4: Block Redaction Process

5. Conclusion and Outlook

This paper shows a newly implemented blockchain around the key feature of the redaction of existing blocks. It is shown that several approaches exist in terms of modifiability of blockchain. Some of these approaches use the chameleon hash function which is also used in this presented prototype. The prototype was developed by a group of students over the duration of one year as part of their master studies. It is shown that the block structure consists of additional fields in comparison to other block structures, e.g. a checksum which is specific for the chameleon hash function. Further, a walkthrough in terms of block creation and block redaction is given, showing how consensus is also applied when changing blocks. Note that the acceptance of change must be given by users and is not automated yet, so the last instance of decision making is a person. In the presented prototype, only the creator of the block can perform a change as only this creator is in possession of the required trapdoor key providing a very simple and basic solution of the problem of key

management. These challenges described by [3] are still valid and must be taken into consideration in further development.

So far, the application has only been tested under laboratory conditions. As a next step, this system should be evaluated by practitioners and business experts. Further, no performance tests could have been realized in the time being, so no findings in terms of performance of the chameleon hash function in this implementation can be made yet. As noted, several key aspects of blockchains were simplified in the development process, e.g. the designed block structure. Those simplified aspects must be revised to develop a more profound blockchain. This also affects possible transaction models or smart contract support. Nevertheless, this project helped students to understand blockchain technology while also enabling them to apply such technology in a real-world use case contributing to ongoing research.

Acknowledgements

The paper was written within the research project HAPTIK (www.haptik.io) at the University of Oldenburg. The research goal is the digitization of the bill of lading using blockchain technology.

References

[1] S. Wunderlich and D. Saive, “The Electronic Bill of Lading,” in Advances in Intelligent Systems and Computing, Blockchain and Applications, J. Prieto, A. K. Das, S. Ferretti, A. Pinto, and J. M. Corchado, Eds., Cham: Springer International Publishing, 2020, pp. 93–100.

[2] D. Saive and T. Janicki, “Datenschutz in elektronischen Frachtdokumenten,” RdTW - Recht der Transportwirtschaft Zeitschrift für Transportrecht und Schifffahrtsrecht mit dem Recht des Überseekaufs sowie Versicherungsrecht, Zollrecht und Außenwirtschaftsrecht, no. 6, pp. 201–207, 2019.

[3] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, “Redactable Blockchain – or – Rewriting History in Bitcoin and Friends,” in 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, Apr. 2017 - Apr. 2017, pp. 111–126.

[4] H. Krawczyk and T. Rabin, “Chameleon Signatures,” in Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA, 2000. [Online]. Available: https://www.ndss-symposium.org/ndss2000/chameleon-signatures/

[5] R. H. Hylock and X. Zeng, "A Blockchain Framework for Patient-Centered Health Records and Exchange (HealthChain): Evaluation and Proof-of-Concept Study," Journal of medical Internet research, vol. 21, no. 8, e13592, 2019, doi: 10.2196/13592 .

[6] A. Marsalek and T. Zefferer, “A Correctable Public Blockchain,” in 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, Aug. 2019 - Aug. 2019, pp. 554–561.

[7] M. Florian, S. Henningsen, S. Beaucamp, and B. Scheuermann, “Erasing Data from Blockchain Nodes,” in 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS & PW), Stockholm, Sweden, Jun. 2019 - Jun. 2019, pp. 367–376.

[8] Ivan Puddu, Alexandra Dmitrienko, and Srdjan Capkun, “μchain: How to Forget without Hard Forks,” IACR Cryptol. ePrint Arch., vol. 2017, pp. 106–127, 2017.

[9] S. Farshid, A. Reitz, and P. Roßbach, “Design of a Forgetting Blockchain: A Possible Way to Accomplish GDPR Compatibility,” in Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[10] V. Patil, Chameleon hash function implementation: C Code. [Online]. Available: http://wwwusers.di.uniroma1.it/~patil/projects/cham/code.html (accessed: Aug. 19 2020).

[11] V. Patil, Chameleon Hash Function Implementations: Experimental setup and results. [Online]. Available: http://wwwusers.di.uniroma1.it/~patil/projects/cham/results.html (accessed: Aug. 19 2020).

[12] Hyperledger Fabric, Channels. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/channels.html (accessed: May 13 2020).

[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in 2017 IEEE International Congress on Big Data - BigData Congress 2017: 25-30 June 2017, Honolulu, Hawaii, USA : proceedings, Honolulu, HI, USA, 2017, pp. 557–564. Accessed: Mar. 6 2019.

[14] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” in Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999, 1999, pp. 173–186. [Online]. Available: https://dl.acm.org/citation.cfm?id=296824

[15] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Trans. Program. Lang. Syst., vol. 4, no. 3, pp. 382–401, 1982, doi: 10.1145/357172.357176 .

[16] Bitcoin Wiki, Block. [Online]. Available: https://en.bitcoin.it/wiki/Block (accessed: Aug. 21 2020).

[17] Dr. Gavin Wood, “Ethereum Yellow Paper: a formal specification of Ethereum, a programmable blockchain,” Accessed: Mar. 6 2019.

[18] Hyperledger Fabric, Ledger (accessed: Aug. 21 2020).”

THE CONTINUOUS MATERIALITY OF BLOCKCHAIN

Alesja Serada
The University of Vaasa, PL 700, 65101 Vaasa, Finland

Both cryptocurrency researchers and early adopters of cryptocurrencies agree that they possess a special kind of materiality, based on the laborious productive process of digital ‘mining’ [1]. This idea first appears in the Bitcoin White Paper [2] that encourages Bitcoin adopters to construct and justify its value in metaphoric comparison to gold mining. In this paper, I explore three material aspects of blockchain: physical infrastructure, human language and computer code. I apply the concept of 'continuous materiality' [3] to show how these three aspects interact in practical implementations of blockchain such as Bitcoin and Ethereum. I start from the concept of ‘digital metallism’ that stands for ‘fundamental value’ of cryptocurrencies, and end with the move of Ethereum to ‘proof-of-stake’, partially as a countermeasure against ‘evil miners’. I conclude that ignoring material aspects of blockchain technology can only further problematize complicated relations between their technical, semiotic and social materiality.

1. Introduction

Blockchain technology further complicates the already problematic divide between software and hardware. In a way, it is an answer to ‘digital immateriality’ of online services and transactions. Digitization of records, including accounting journals and ledgers, has led to new challenges to prove their authenticity. A cryptographic record on blockchain has been proposed as one of such solutions, because it is validated by an unintermediated, even if often costly, consensus between many participants of a network.

Architecture of a blockchain platform does not require a central server to keep the records. Results of each transaction are validated by a majority of nodes in a network or by a reasonable share of selected representatives on a digital platform, and then recorded as the next block on each node. In the common process of validation, computing takes place at many machines at once, which makes blockchain platforms particularly robust and, ideally, affords their democratic self-regulation. For example, Filipe Calvão believes that "the work of digital mining... enables the formation of democratic communities" [1], and compares mining pools to trade unions, even though empirical results of his own research show heavy ‘capitalization’ of industrial ‘mining’.

Material conditions of blockchain platforms have been the subject of many researchers since the early years of Bitcoin studies [4]–[6]. Many turn to the ‘material’ value of cryptocurrencies: for instance, in their studies of early Bitcoin hype, Garcia et al. suggest that the “fundamental value” of one Bitcoin equals at least the cost of its production [4]. From this perspective, the exchange price of mined cryptocurrencies is tied to the material conditions of ‘mining’, although this existing relation has only been further complicated with time. At an early stage of blockchain adoption, Henrik Karlstrøm calls for more attention to the 'material embeddedness' of Bitcoin, or its connection to specific material and institutional arrangements [5]. This paper relies on development and specification of this approach within technology studies.

Taking the previous research in blockchain studies into account, I counterpose it with factual implementations of roadmaps for Bitcoin and Ethereum. To integrate the history and the genealogy of blockchain into a wider perspective of information and communication technologies, I turn to the “material history of bits” [7], and to the critical study of ‘continuous materiality’ of computer code [3]. Blockchain-related discourse has many features of ‘rupture talk’ [8], and I propose to look at the specific material conditions of decentralization that may have been overshadowed by it.

2. The Rigs, the Fees and the Lags of Blockchain Technology

There are many ways to build a blockchain solution today, and most of them do not require dedicated hardware. Still, the initial ‘proof-of-work’ protocol popularized by Bitcoin and the first version Ethereum, involves so-called collective ‘mining’ of a cryptographic hash. The material technology behind blockchain solutions is represented by physical 'rigs' - stacks of equipment for industrial ‘mining’ - and the non-trivial amount of electricity spent on it.

Initially, bitcoins were ‘mined’ on CPUs of personal computers. Since around 2011, mining was mostly performed on dedicated GPUs due to growing complexity of calculations [6]. Some miners’ continued to use GPUs late into 2017, mostly to mine various (and often highly speculative) ‘altcoins’ [1], but production of bitcoins has mostly moved to industrial facilities as early as in 2013 [5].

A basic unit for professional or industrial mining is a dedicated ASIC (Application Specific Integrated Circuit), also simply called a 'miner'. A large farm can have thousands if not hundreds of thousands, of ASICs. As an illustration, a documentary from a cryptocurrency-related YouTube channel VoskCoin invites its viewers to visit one of the biggest farming facilities in the USA, located in North Carolina. Its power is around 100 megawatt, which is comparable to a large data center. Around 90% of the mining equipment is owned by clients who rent the facility. As of 2020, some of them still use GPUs, due to their relatively low cost. [9]. The software company that owns the farm states that it uses 80% renewable energy; however, the magnitude of industrial mining defies the notion of energetic efficiency.

At a certain point in history, industrial 'mining' expanded to an almost planetary scale. Same as major data centers and server farms, economic efficiency of big 'mining' farms depends on the climate at their location. In addition to that, they gravitate towards cheap sources of energy such as hydroelectric power plants in geologically diverse regions. In the golden days of Bitcoin, a lightweight version of its software could run on any personal computer; now, the principal hardware is to be found among picturesque mountains of China, not far from the controversial Three Gorges Dam. On certain days, the speed of a cryptocurrency transaction literally depends on the weather in China. Mining capacities are regularly damaged by seasonal floods [10], [11]. Although not directly related to material conditions, we should also consider the sociopolitical environment of China, where cryptocurrencies have been effectively banned since 2017 [12], but the state simultaneously heavily invested in blockchain technologies and even designed a state digital currency [13].

The unprecedented energetic cost of Bitcoin validation has led to very material ecological concerns. "You are a miner, you are destroying the world!" - the host of VoskCoin playfully teases the farm keeper in the documentary [9]. The inefficiency of this process is so jarring that Alexander Galloway even compares cryptocurrency farms to the XIX century steam machines "that run on heat and energy" [14]. In his critical essay Anti-Computer, he applies the Marxist perspective to Bitcoin farms, which makes them "essentially large batteries for value" in the same way as the machines used to produce steel or textile: both "burn fuel to release value" (ibid.). Symbolically, the farm visited by VoskCoin occupies several buildings of a former textile factory [9]. This makes Bitcoin rather steampunk than cyberpunk.

3. 'Digital Metallism': the Semiotic Materiality of Bitcoin

'Bitcoin mining' is a primary metaphor used to explain how a cryptocurrency works. This rhetorical tool first appears in the Bitcoin White Paper: "The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation" [2]. The metaphor defined the language of blockchain adopters for the years to come, spawned countless memes and even influenced narratives of many 'crypto games' such as Ether Kingdoms and My CryptoHeroes, which could be considered educational in this regard.

Rhetorically, the metaphor of 'mining' has everything to do with material existence and circulation of gold as a metal. Building from Nakamoto's statements, blockchain aficionados justify the fundamental value of Bitcoin and several other cryptocurrencies by referring to their limited supply and presumable scarcity, and comparing them to the 'gold standard'. Comparison of Bitcoin to gold is sometimes dramatically reversed, so gold is compared to Bitcoin [15], especially after the recent surge in price of the former.

Such comparisons highlight a specific form of an 'authentic' value that also relies on natural scarcity. However, it is problematic to speak about the scarcity of digital tokens that are mined in large quantities on an industrial level, can be divided into almost infinitely small parts and, most importantly, have very little use value outside of professional trading. The problem of 'fundamental value' of cryptocurrencies circulates not only in research circles [4], but also in online communities of traders and early adopters. According to their views, Bitcoin, as well as many following cryptocurrencies, derives its value from the computational work put into 'mining'. Semiotically, mining is "an algorithmic imitation of the limited supply of metallic currencies" such as golden dollar coins [15, p. 72]. Elizabeth Ferry even noticed that the rhetorics of Bitcoin adherents is similar to those who invest in physical gold, especially in their political stance. This principle of value creation in cryptocurrencies has been described as 'digital metallism' [6].

Does 'digital metallism' make Bitcoin more material? Even though it is a discursive construction, it has real implications in the real world. It creates an additional level of conceptualizing and comprehending blockchains. The current level of public understanding would be impossible without this interpretive discourse.

Blockchain technologies are often seen as too complex and difficult to comprehend. This is often mentioned as the reason for relatively slow adoption, although the absence of actual use cases might be the real reason. However, metaphoric interpretation affects not just the human users and developers of blockchain applications, but also the machines they build, the money they invest, and the code they write. Looking at countless visual and verbal representations of 'miners' in online discussions, we cannot simply discard their image as 'immaterial', even knowing that the real owners and workers of cryptocurrency farms are nothing like that.

4. How 'Miners' Became Evil: Blockchain as a Sociotechnical Object

Contrary to its initial ambition as a global currency, the current design of Bitcoin does not allow for scalability, which is an inherent problem of blockchains in general. This failure of the seemingly immaterial code involves not only limitations of hardware, but also existing socioeconomic arrangements. One simple example is storage, even though it is not as obvious as processing power. Hosting full nodes, which would participate in the global verification process, would require more and more storage: as of August 2020, recommended disk space for hosting a full node is 350 Gb [16]. Some voices in the community would point out that the size of a full node would eventually surpass the technical capacities of an ordinary Bitcoin user and leave verification to a dedicated and wealthy few. Exactly

this is likely to happen to the second major cryptocurrency, Ether.

The block size problem is the most discussed scalability problem of Bitcoin. The size of a single block that contains records of new transactions is limited to 1 Mb. This limits the number of transactions that can be performed and verified throughout the whole network. A number of solutions have been presented since 2015, but the problem generally remains unsolved due to the lack of consensus between developers, miners, investors and other representatives of the community [17]. Unfortunately, all efforts to establish a democratic procedure for reasonable decision-making in the Bitcoin community were futile.

The impressive scale of the scalability debate does not allow to follow it in this paper, but I would like to draw attention to 'miners' who unexpectedly appeared as independent social agents in the dramatic process of Bitcoin infrastructuring [18]. 'Miners', or, more specifically, owners of facilities for industrial mining, became important actors behind decentralization after the rapid industrialisation and the following centralization of Bitcoin mining. Also, this is when 'miners' became 'evil' in ordinary discourse of blockchain adopters.

The initial vision of 'miners' among Bitcoin portrays them as agents of digital democracy. Miners not only contribute to the algorithmic consensus on the validity of every next transaction, they also represent the interests of the Bitcoin community, accept or reject the changes to the code. This idea originates in Nakamoto's writing: as of 2008, he suggested that "proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote" [2]. This idea quickly became outdated as mining moved to GPU rigs and then to industrial-level ASICs, and the computational power was organized in pools. The promise of decentralization has been broken many times since then [19]. Democracy or not, 'miners' as social agents can influence decisions about the future of blockchain.

Another event that involves the code, the hardware and the community is the 'halving' of Bitcoin. 'Halving' is decreasing a reward for mining in half. This is a pre-programmed event that happens after mining every 210,000 blocks. It also raises the 'fundamental value' of Bitcoin, as twice more resources are required to mine the same quantity of bitcoins [4]. It usually leads to a noticeable surge in Bitcoin price, although the consequences of mining are unique for each time. The difficulty of mining has been constantly growing since the introduction of Bitcoin, although it can be algorithmically adjusted to match the total mining power. After the reward to miners of Bitcoin had been halved on May 11, 2020, its volatility decreased and difficulty of mining increased [20], which led to concerns about economic unsustainability of 'mining'. These concerns return every time the price of Bitcoin approaches its 'fundamental value', as it was during its crash in December 2018, when mining difficulty temporarily dropped -15% and many miners left the business [21]. Bitcoin still would be impossible to use without the work of miners. This work is becoming less and less rewarding, and relations between the market, the code and the hardware are still far from reaching a long-lasting equilibrium.

5. The Challenges of Ethereum

The Ethereum platform, fueled by the most used 'altcoin' Ether, arrived in 2015 as a revitalizing solution to realize a variety of use cases on blockchain. It reached the limits of its scalability after 5 years, effectively freezing all activities made possible by a massive and dedicated community of developers. As a response to this situation, in summer 2020, it is moving from proof-of-work, which required miners, to proof-of-stake where transactions are validated by cryptocurrency holders who own stakes worth at least 32 ETH per validator.

The new platform, Ethereum 2.0, postulates energy efficiency as one of its major advantages - finally, ecological concerns were addressed - but it comes at the cost of partial centralization. Now validation of transactions and other matters that require consensus are under control of a limited group of wealthy individuals (32 ETH roughly amounted to USD12,000 since April 2020). The community-written resource EthHub suggests that the proof of stake is fairer than the proof of work: "$10 million of coins will get you exactly 10 times higher returns than $1 million of coins, without any additional disproportionate gains because at the higher level you can afford better mass-production equipment" [22]. This statement represents the platform economy of Ethereum as a 'fair game' where everyone is rewarded proportionally to their input. It still remains blind to the fact that the initial distribution of wealth may not have granted most Ether to the most honest, or even the most reasonable individuals.

As such, the proof-of-stake protocol goes against the initial crypto-anarchist beliefs of Bitcoin adopters, because it replaces democracy with plutocracy. This also affects miners on both industrial and 'artisan' scale: even before the staking of Ethereum, Calvão suggested that "private-led blockchain-based initiatives based on the stake in the network may push small-scale (crypto) miners and, by extension, artisanal miners out of the system of rewards and incentives in place" [1]. However, it is still presented as a measure against centralization in the discourse of Ethereum supporters, mostly because the proof-of-stake protocol reduces involvement of big 'mining' companies in decision-making and safeguards against 51% attacks.

The algorithmic basis of value is also affected. Ethereum 2.0 allows 'sharding', or breaking down blocks. It decreases demand for Ether to fuel transactions on Ethereum, which leads to concerns about its artificial scarcity. However, the exchange rate of Ethereum to Bitcoin is on the rise in 2020:

more importantly, Ethereum 2.0 affords passive income by hosting the nodes with the locked value of 32ETH or more, which means more predictable return of investments in the long run.

Finally, let us look at the physical materiality of Ethereum 2.0. Hosting a node requires a server that remains online 24/7. Of course, it can be a rented server, but the usual rhetorics of 'the cloud' should not prevent us from remembering that even so-called 'cloud services' are not hosted in the thin air. To the contrary, they usually require large scale server farms, as in the case of the leading service from Amazon. Even though such facilities are much more energy saving, they are still basically the same kind of 'power plants' as 'mining farms'.

6. 'Continuous Materiality' of Blockchain

Exploration of technicalities behind the distributed architecture of Bitcoin, Ethereum and other blockchain solutions may make us wonder whether electronic communications have ever been 'immaterial'. Computer code does not exist without the hardware to run on, and it also needs people to make use of it. Actions of these people have material consequences in the real world, and this is also the side of blockchain technologies that is somehow underdeveloped due to limited adoption.

Materiality of technology is seen as threefold in studies of technology and society. Firstly, it is material technology behind blockchain solutions: physical 'rigs' and the electricity spent on 'mining'. Secondly, it is the semiotic level that reveals itself in metaphors like 'mining' ground the code in material reality. Thirdly, practical implementations of blockchain become embedded into active human networks of early adopters, miners, developers, investors and other actors such as researchers (who sometimes go no further than the semiotic level). This corresponds to three definitions of materiality: matter, significance and practical instantiation [23]. The latter, which is the social dimension of materiality, describes how software exists as a part of a social practice that "compels people to follow the abstract plan" [23], for instance, to trade cryptocurrencies as a part of the real-world economy.

'Continuous materiality' of electronic communications can be understood as multi-level amalgamation of computer code, human language and physical entities: "a wide spectrum of materiality activated by a hierarchy of codes" [3]. Such assemblages also include social codes of behavior, which becomes visible, for example, when Bitcoin developers blame miners for violating such code in their refusal to update the Bitcoin software. In the end, such arrangements become solidified in the legal code: acknowledgement of cryptocurrencies as a specific type of assets, and, in different areas of application, property rights and personal identification based on blockchain. This is how records on blockchain become hard institutional facts that directly define the rules for the material world.

Eventually, before solidifying the code in legal and institutional relations, it is important to consider the basic level of its (im)materiality. The very real physical matter of decentralized calculations often remains hidden beyond the promise of decentralization. As Blanchette writes, "a focus on materiality highlights that computation is a mechanical process based on the limited resources of processing power, storage, and connectivity" [7, pp. 1042–1043]. These exact resources were exhausted by blockchain technologies in just over 10 years. This is another sad confirmation of Blanchette's thesis that "Yet we today have neither technical language nor intuition for something akin to the tensility, durability, or density of computing resources" [7, p. 1055]. While decentralized blockchain records can account for great durability, the underlying infrastructure can never ensure the required plasticity and flexibility. Of course, there have always been rightful warnings about exactly this problem: as early as in 2013, Karlstrøm noticed that "the materially embedded features of the currency, such as its reliance on very specific physical technologies, can point towards an underlying tension within the rhetoric behind Bitcoin" [5]. This tension between the material and the discursive reality of blockchain has only intensified during the following five years of public adoption.

Practical implementations of blockchain may rely on problematic assumptions that have more to do with the discourse than with the technology itself. For example, in a much publicized partnership with IBM, the national clearing house of Poland, Krajowa Izba Rozliczeniowa (KIR), developed a blockchain solution for 'durable medium' on blockchain [24], even though such records are still much more of a message that may or may not be delivered depending on the state of the network. To prevent unwanted disruptions, developers should consider that "the boundary of software is always affected by the limitation of hardware" [3].

Blockchain technologies are usually presented as 'disruptive', which makes the blockchain discourse yet another example of 'rupture talk'. It is not uncommon in the history of technology when "the sharp breaks proclaimed by elites masked profound continuities"[8, p. 692], and the rupture in fact 'conjugated' the same sociotechnical relations it was supposed to abruptly end. The usual promise of cryptocurrencies to 'bank the underbanked' has evolved into an abundance of investment schemes for those who already had, or were lucky enough to quickly acquire, enough 'digital wealth' to invest.

Metaphors are an excellent tool to understand new technologies and to form meaningful connections with them. However they should not replace the material reality that makes them work. For example, the metaphor of 'cyberspace' influences the way we envision the internet, sometimes in a confusing way. "'Cyberspace', in this sense, is conceived of as both an ethereal alternate dimension which is simultaneously infinite and everywhere (...), and as fixed in a distinct location, albeit a non-physical one

(...)" [25, p. 179]. Such a view remains blind to the physical infrastructure of electronic networks, which is costly and often vulnerable. Another example is the discourse of ‘regulating cyberspace’, against which so many early blockchain adopters have argued, - and yet, the blockchain-related discourse never acknowledges the fact that blockchain platforms function within the existing (or non-existent, or temporarily unavailable) material infrastructures of the internet.

7. Conclusion

Does decentralization mean dematerialization? Could it be that, by taking the blockchain agenda at face value, we are following the same path as with still unfulfilled promises of ‘cyberspace’? Constraints and affordances of software can shape the material world “in much the same way as physical artifacts do” [23]. Software is nothing without hardware, and decentralization does not free the participants of an electronic network from material constraints. It merely obfuscates the role of hardware and material expenditures such as the cost of electric energy.

The material aspect of digital currencies is well represented in ‘digital metallism’. Images of mining rigs resting under buzzling coolers represent the materiality of Bitcoin and allow to treat it as authentic ‘digital gold’. However, time has shown that ‘digital metallism’ does not guarantee the future of digital currencies. Integrity of the network and authenticity of records on blockchain can be safeguarded by a sheer amount of computational power, but this is still not enough to impose integrity on the community or create value beyond expenditures.

After all, blockchain networks are only a superficial layer over the existing internet infrastructure, Due to their unprecedented speed, electronic communications were expected to overcome the limitations of time and space, but laborious verification of data and the need to update it at each node have brought these limitations back into the equation. Almost unsolvable scalability issues are the final reminder about the fundamental fact that blockchain technologies are never ‘synchronous’, but inherently ‘historical’. While the data kept and transferred on blockchain is discrete, the time to calculate and verify the results is continuous, and usually very long. As a result, operations on blockchain tend to slow down to an almost full halt, as in the case of Ethereum, especially when ‘mining’ is a part of the process. Furthermore, blockchain platforms are simultaneously electronic and social networks, which brings interactions between embodied, and often very passionate, human agents into the equation. To avoid inefficiency and stagnation, developers and investors should not forget about material limitations of immaterial blockchains. The revolutionary and disruptive potential of electronic communications should be evaluated by taking technological, semiotic and social aspects into account.

Acknowledgements

This paper would not be possible without the support from the Evald and Hilda Nissi Foundation for students engaging in commercial studies.

References

[1] F. Calvão, Crypto-miners: Digital labor and the power of blockchain technology, Econ. Anthropol., 6, Nr. 1 (2019), 123–134, doi: 10.1002/sea2.12136.

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

[3] K. J. Knoespe and J. Zhu, Continuous Materiality: Through a Hierarchy of Computational Codes, Fibreculture J., Nr. 11 (2008), Accessed: Aug. 26, 2020. [Online]. Available: http://eleven.fibreculturejournal.org/fcj-076-continuous-materiality-through-a-hierarchy-of-computational-codes/.

[4] D. Garcia, C. J. Tessone, P. Mavrodiev, and N. Perony, The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy, J. R. Soc. Interface, 11, Nr. 99 (2014), doi: 10.1098/rsif.2014.0623.

[5] H. Karlstrøm, Do libertarians dream of electric coins? The material embeddedness of Bitcoin, Distinktion J. Soc. Theory, 15, Nr. 1 (2014), 23–36, doi: 10.1080/1600910X.2013.870083.

[6] B. Maurer, T. C. Nelms, and L. Swartz, ‘When perhaps the real problem is money itself!’: the practical materiality of Bitcoin, Soc. Semiot., 23, Nr. 2 (2013), 261–277, doi: 10.1080/10350330.2013.777594.

[7] J.-F. Blanchette, A material history of bits, J. Am. Soc. Inf. Sci. Technol., 62, Nr. 6 (2011), 1042–1057, doi: 10.1002/asi.21542.

[8] G. Hecht, Rupture-Talk in the Nuclear Age: Conjugating Colonial Power in Africa, Soc. Stud. Sci., 32, Nr. 5–6 (2002), 691–727, doi: 10.1177/030631270203200504.

[9] MASSIVE Crypto Mining Farm Tour | Bitcoin, Dash, and GPU Mining!, Deeper in the mines, VoskCoin, Feb. 10, 2020.

[10] A. Marshall, Local Media: Floods in China Heavily Damage Major Crypto Mining Operation, Cointelegraph, Jul. 01, 2018.

[11] J. Redman, Flooding Threatens China’s Bitcoin Miners, Chinese Billionaire Says ‘Three Gorges Dam Collapse Imminent,’ Bitcoin News, Aug. 05, 2020. https://news.bitcoin.com/flooding-threatens-chinas-bitcoin-miners-chinese-billionaire-says-three-gorges-dam-collapse-imminent/ (accessed Aug. 28, 2020).

[12] R. Xie, Why China Had to Ban Cryptocurrency but the U.S. Did Not: A Comparative Analysis of Regulations on Crypto-Markets between the U.S. and China, Wash. Univ. Glob. Stud. Law Rev., 18, Nr. 2 (2019), [Online]. Available: https://heinonline.org/HOL/Page?handle=hein.journals/wasglo18&id=469&div=&collection=.

[13] M. A. Peters, B. Green, and H. (Melissa) Yang,

Cryptocurrencies, China's sovereign digital currency (DCEP) and the US dollar system, Educ. Philos. Theory (2020), doi: 10.1080/00131857.2020.1801146.
[14] A. R. Galloway, Anti-Computer, Mar. 19, 2018. http://cultureandcommunication.org/galloway/anti-computer (accessed Sep. 27, 2019).
[15] E. Ferry, On Not Being a Sign: Gold's Semiotic Claims, Signs Soc., 4, Nr. 1 (2016), 57–79, doi: 10.1086/685055.
[16] Requirements and Warnings - Bitcoin Core, Bitcoin.org. https://bitcoin.org/en/bitcoin-core/features/requirements (accessed Aug. 30, 2020).
[17] O. Williams-Grut and R. Price, A Bitcoin civil war is threatening to tear the digital currency in 2 — here's what you need to know, Business Insider, Mar. 26, 2017.
[18] Y. M. Kow and C. Lustig, Imaginaries and Crystallization Processes in Bitcoin Infrastructuring, Comput. Support. Coop. Work CSCW, 27, Nr. 2 (2018), 209–232, doi: 10.1007/s10606-017-9300-2.
[19] G. Vidan and V. Lehdonvirta, Mine the gap: Bitcoin and the maintenance of trustlessness, New Media Soc., 21, Nr. 1 (2019), 42–59, doi: 10.1177/1461444818786220.
[20] W. Zhao, "Bitcoin Mining Difficulty Sets New Record High 2 Months After Halving," CoinDesk, Jul. 13, 2020.
[21] K. Moskvitch, How to make sense of bitcoin's unrelenting death spiral, Wired UK, Dec. 10, 2018.
[22] Ethereum Proof of Stake, EthHub. https://docs.ethhub.io/ethereum-roadmap/ethereum-2.0/proof-of-stake/ (accessed Jul. 26, 2020).
[23] P. M. Leonardi, Digital materiality? How artifacts without matter, matter, First Monday, 15, Nr. 6 (2010), doi: 10.5210/fm.v15i6.3036.
[24] KIR: Easing banking compliance, reinforcing trust and accelerating services with IBM Blockchain Services," IBM, May 20, 2020. https://www.ibm.com/case-studies/kir-blockchain-financial-services-payments (accessed Aug. 25, 2020).
[25] M. Graham, Geography/internet: ethereal alternate dimensions of cyberspace or grounded augmented realities?, Geogr. J., 179, Nr. 2 (2013), 177–182, doi: 10.1111/geoj.12009.

DISTRIBUTED LEDGER TECHNOLOGIES IN LOGISTIK UND SUPPLY CHAIN MANAGEMENT IM KONTEXT VON DATENSICHERHEIT UND DATENQUALITÄT

Maximilian Stange
Fraunhofer Institut für Werkzeugmaschinen und Umformtechnik IWU,
Reichenhainer Straße 88, D-09126 Chemnitz

In der Anfangszeit der Distributed Ledger Technologies (DLT) waren die hauptsächlichen Betrachtungswinkel die der Disruption des Bank- und Finanzwesens. Mit dem Aufkommen des Systems Ethereum im Jahr 2015, hat die Auseinandersetzung mit der Anwendung von Blockchain in weiteren Branchen, an Bedeutung gewonnen. Eine davon ist die Logistik und das Supply Chain Management (SCM). Gerade in Deutschland spielt der Logistiksektor eine große Rolle, nach der Beschäftigtenzahl ist er die drittgrößte Branche und erzielt einen Umsatz von rund 258 Milliarden Euro. Im Beitrag werden konkrete Anwendungsfelder identifiziert und gezeigt welche potentiellen Vorteile sich dort, durch den Einsatz von DLT, erzielen lassen. Ein Schwerpunkt liegt dabei auf der Einschätzung der Technologie hinsichtlich ihrer Sicherheitseigenschaften. Im Beitrag wird den Fragen nachgegangen, ob Datensicherheit mithilfe von DLT verbessert werden kann und auf welchem Wege.

1. Einleitung

Bitcoin hat es im Jahr 2017 geschafft der breiten Öffentlichkeit ein Begriff zu werden. Das liegt vor allem an der enormen Wertsteigerung, die die Kryptowährung zu einem begehrten Investment gemacht haben. Die den Kryptowährungen zugrundeliegende Technologie ist als Blockchain bekannt. Sie ist, vereinfacht ausgedrückt, „ein Medium für digitale Werte“, das diese durch kryptografische Verfahren schützt.

Anhand des Aufkommens von digitalen Währungen ist ersichtlich, dass ein großes Potential besteht, die Finanzbranche grundlegend umzuwälzen. Aber auch andere Branchen stehen, aufgrund der Technologie, vor einem Wandel. Dabei werden immer wieder die Bereiche Logistik und SCM genannt. Gerade in Deutschland spielt der Logistiksektor eine große Rolle, nach der Beschäftigtenzahl ist er die drittgrößte Branche und erzielt einen Umsatz von rund 279 Milliarden Euro [1].

Ein Beispiel für die Anwendung in diesem Bereich ist die Rechnungsstellung. Derzeitige Warenwirtschaftssysteme erfassen heute in Sekundenbruchteilen Änderungen im Bestand bei einer Bestellung. Die Rechnungsstellung findet aber oft noch manuell statt und dauert dementsprechend länger. Es ist denkbar diesen Prozess mithilfe eines Smart Contracts zu automatisieren und somit zu beschleunigen, womit beispielsweise die Liquidität einer Unternehmung erhöht werden kann. Daneben existieren eine Reihe weiterer angedachter Anwendungen von DLT in der Branche, die zum Teil auch auf Smart Contracts basieren. Diese sollen hier aufgezeigt und deren Potential abgeschätzt werden. Ein Schwerpunkt liegt dabei auf der Einschätzung der Technologie hinsichtlich ihrer Sicherheitseigenschaften. Denn das Thema IT-Sicherheit gewinnt immer mehr an Bedeutung, gerade im Hinblick auf die gleichbleibend hohe Gefährdungslage im Bereich der Cyberspionage zu Ungunsten deutscher Unternehmen [2].

2. Begriffseinordnungen

Distributed-Ledger-Technologien stellen ein relativ neues Forschungsfeld dar, daher gibt es auch keine einheitliche Definition und verschiedene technische Ausführungen werden darunter subsumiert [3]. Verallgemeinert handelt es sich um Systeme zur verteilten Kontoführung, bei denen die Daten von allen, oder zumindest mehreren Computern im beteiligten Netzwerk genutzt, weitergegeben und verifiziert werden [4]. Hauptmerkmale von DLT sind:

a) Im Vergleich, zu den bisher üblichen Shared-Ledgers sind Distributed-Ledger-Systeme nicht auf eine übergeordnete, zentrale Instanz angewiesen. Informationen in digitaler Form können zwischen Parteien, die sich untereinander nicht vertrauen, ausgetauscht und gespeichert werden.

b) DLT stellen sicher, dass es zu keinem Double-Spending kommen kann. Double Spending bezeichnet dabei das mehrfache Ausgeben eines digitalen Guts beispielsweise eines Bitcoins.

Allgemein ausgedrückt werden DLT dazu genutzt den Besitz von digitalen Gütern nachzuverfolgen. Beispiele für DLT sind das Bitcoin- und das Ethereum-Netzwerk.

Smart Contracts (SC) können, wenn sie mithilfe einer DLT implementiert wurden, als unveränderliche Computerprogramme bezeichnet werden, die „rechtlich relevante Handlungen […] in Abhängigkeit von digital prüfbaren Ereignissen steuer[n], kontrollier[en] und/oder dokumentier[en] […] [5].

Smart Contracts, die für ihre Ausführung auf Informationen außerhalb der Blockchain angewiesen sind, werden als **non-deterministic Smart Contracts** (NDSC) bezeichnet. Diese Art von Smart Contracts erweitert das Spektrum an möglichen Anwendungsszenarien beträchtlich [6]. Ihr Einsatz birgt jedoch auch Gefahren, da hier der Rückgriff auf Daten einer dritten Partei nötig ist, die nicht in jedem Fall fehlerfrei agiert und nicht immer völlig vertrauensvoll ist. Dem gegenüber stehen sogenannte **deterministic Smart Contracts** (DSC). DSC sind solche Smart Contracts die nur aufgrund der Informationen, die in dem DL

vorhanden sind funktionieren.

Da eine direkte Anbindung an eine externe Datenquelle (z.B. ein aktueller Börsenkurs) einen nicht deterministischen Zustand des Ledgers auslösen könnte, werden **Oracles** als vermittelnde Instanz verwendet. Grundsätzlich sind Oracles auch Smart Contracts. Sie dienen lediglich als vermittelnde Instanz zwischen externen Datenquellen und anderen Smart Contracts. Smart Contracts, die auf externe Informationen angewiesen sind, um zu funktionieren, fragen ein Oracle ab, anstatt dies direkt bei der originären Datenquelle zu tun. Dafür sendet die externe Datenquelle dem Oracle Updates über den abzufragenden Zustand, dadurch werden Inkonsistenzen vermieden, da nun ein Smart Contract einen anderen abfragt [7]. Das bedeutet, dass das an sich geschlossene System eines Distributed Ledgers durch Oracles um eine Anbindung an die Außenwelt erweitert wird, was die Nutzung externer Datenquellen erlaubt.

3. Anwendungsfelder von Smart Contracts

Die Anwendungsfelder von Smart Contracts werden durch die Vorteile definiert, die diese Technologie mit sich bringt. Zu den größten Vorteilen von Smart Contracts gehören:

- Sehr geringe Wahrscheinlichkeit, dass die in den Smart Contract festgelegten Vereinbarungen nicht durchgesetzt werden → Durch die dezentrale Ausführung und Validierung des Smart Contracts
- Genauigkeit → z.B. Reduzierung der Fehlerquote, dort wo sonst manuell Daten übertragen werden
- Höhere Geschwindigkeit durch Automatisierung von bisher manuellen Prozessen → z.B. Vergleich mehrerer Quellen ob ein Flugzeug Verspätung hat und sofortige Freigabe der Versicherungssumme, sollte der Versicherungsfall eintreten.
- Reduzierung der Zahl an Intermediären → Vertrauen unter den Vertragsparteien ist nicht nötig, daher werden Dritte, die für die Einhaltung von vertraglichen Vereinbarungen garantieren, nicht benötigt
- Geringere Kosten → Durch die vorher genannten Punkte [8]

Use Cases für Smart Contracts sind somit dort zu finden, wo die Digitalisierung und Automatisierung von Prozessen bisher, aufgrund des Fehlens fälschungssicherer digitaler Dokumente, nicht möglich war. Neben der Prozessoptimierung werden durch Smart Contracts auch gänzlich neue Geschäftsmodelle möglich. Decentralized Autonomous Organizations (DAO) sind ein Beispiel für ein neues Geschäftsmodell auf Basis von Smart Contracts. DAOs sind Organisationen ohne Vorstände und Geschäftsführer, die sich selbst verwalten und in denen Nutzer Stimm- und Eigentumsrechte erwerben können, um beispielsweise über zukünftige Investitionen zu bestimmen [9].

4. Limitierungen von Smart Contracts

Die zuvor genannten Vorteile der Smart Contracts lassen sich aufgrund von Limitierungen hinsichtlich der Sicherheit, Anwendbarkeit und weiterer Faktoren, noch nicht in umfänglicher Weise in reale Anwendungen übertragen beziehungsweise verhindern deren Ausdehnung auf größere Benutzergruppen.

Zu den gewichtigsten Faktoren, die den Einsatz von Smart Contracts hemmen, gehören problematische Aspekte bei der Sicherheit von Smart Contracts. Eine vollumfängliche Darstellung gibt es auch hier noch nicht, unter anderem wegen der noch geringen Verbreitung von Smart Contracts [7]. Trotzdem ist bereits bekannt, welche gravierenden Folgen durch Sicherheitslücken entstehen können. Bekanntestes Beispiel dafür ist die bereits angesprochene DAO, eine Investmentfirma ohne menschliches Personal. Durch einen Programmierfehler wurden digitale Token im Wert von 53 Millionen Dollar entwendet. Nur durch eine Protokolländerung (Hard Fork) der gesamten Ethereum-Blockchain konnte der Schaden rückgängig gemacht werden. Dadurch existiert aber seitdem neben Ethereum noch die Kryptowährung Ethereum Classic [10].

Eine der ersten Publikationen zur Sicherheit von Smart Contracts auf Basis von Ethereum stammt von DELMOLINO ET AL. (2016) und bezieht sich auf die Besonderheiten in der Programmierung von Smart Contracts und häufig begangenen Fehlern, die auf die Eigenheiten von DL zurückzuführen sind.

Im Wesentlichen stellen sie drei Fehlerarten fest:

1. Smart Contracts, die gesendeten Beträge einschließen und nicht wieder freigeben
2. Speichern von Nutzerinformationen in Klartext, wodurch eine andere Partei einen Vorteil erzielen kann
3. Falsch gesetzte Anreize können dazu führen, dass sich Nutzer nicht wie beabsichtig verhalten, da ein Fehlverhalten nicht oder nicht stark genug bestraft wird [11].

LUU ET AL. (2016) zeigen mit ihrem Tool Oyente, dass 45 % aller Smart Contracts in Ethereum nach ihrer Definition Bugs besitzen, die von Angreifern ausgenutzt werden könnten [12]. Die meisten Fehler treten dabei bei der Behandlung von Ausnahmen auf. Das kann unter anderem dazu führen, dass Gelder unwiderruflich in einem Smart Contract verbleiben, ohne dass der rechtmäßige Besitzer darauf Zugriff erhält [12].

NIKOLIC ET AL. (2018) definieren in ihrer Untersuchung drei Arten angreifbarer Contracts:

Verschwenderische Smart Contracts → Sind solche Verträge, die Gelder willkürlich an andere Adressen schicken können.

1. Suizidale Smart Contracts → Können von anderen Nutzern als dem Ersteller oder an-deren berechtigten Personen zerstört werden.
2. Gierige Smart Contracts → Behalten eingezahlte Gelder ein, ohne dass eine Möglichkeit besteht

sie wieder freizugeben [13].

Von knapp einer Million untersuchter Smart Contracts können rund 2,5 % in einer der von den Autoren definierten Kategorien eingeordnet werden. Die Diskrepanz zwischen den Zahlenangaben zu Smart Contracts mit sicherheitsrelevanten Schwachstellen (45% zu 2,5 %) zeigt, dass einheitliche Bewertungsmaßstäbe gefunden werden müssen. Beide Untersuchungen zeigen jedoch unabhängig davon, dass ein nicht vernachlässigbarer Anteil von Smart Contracts über potentiell gefährliche Fehler verfügt, die die Gelder von Nutzern gefährden. Eine Lösung hierfür ist die Überprüfung von Smart Contracts mit den von den Autoren entwickelten Tools, bevor sie eingesetzt werden, um solche Sicherheitslücken zu vermeiden. Natürlich gibt es auch in jeder anderen Software Bugs, Anwendungen auf Basis von DLT sind jedoch durch drei Eigenschaften besonders gefährdet:

1. **Finalität von Transaktionen**
   Eigentlich ein Vorteil von Distributed Ledgers, jedoch können beispielweise bei einer erfolgreichen Attacke die Beträge nicht zurückgebucht werden oder ähnliches. Nur durch die Änderung des gesamten Protokolls, der alle Knoten zustimmen müssen kann eine Entschädigung stattfinden. Im Falle von The DAO wurde dies nur durchgeführt, da ein großer Anteil von Ether in The DAO investiert wurde, dadurch war die Anwendung „too big to fail“ [10]. Für Smart Contracts mit kleineren Summen ist diese Art der Rettung so gut wie ausgeschlossen. Aber auch Hacks und Bugs, deren Größenordnung, gemessen an den involvierten Summen an Kryptowährungen, die des DAO-Hacks übersteigen, können nicht grundsätzlich auf eine Rettung hoffen. Beispiel hierfür ist der als Parity Bug bzw. Hack bezeichnete Fall, indem die Gelder einer Wallet Software nicht mehr zugänglich waren, da ein unberechtigter Nutzer eine notwendige Bibliothek gelöscht hatte (Suizidaler Smart Contract).
2. **Starker monetärer Anreiz Smart Contracts anzugreifen**
   Da fast die Hälfte aller Smart Contracts Finanzgeschäfte abwickeln und somit Kryptowährungen verwalten, ist der potentielle finanzielle Gewinn eines Angriffes hoch.
3. **Noch keine Rechtssicherheit**
   Fehlende Regulierung im Bereich der DL erhöht das Risiko für Nutzer im Falle eines Angriffs keine Entschädigung außerhalb der DL-Infrastruktur zu erhalten etwa über ein Gerichtsverfahren.

5. Problematik Oracles

Die Probleme, welche sich mit der Nutzung von Oracles auftun, sind offensichtlich, da sie die Zielsetzungen von DL konterkarieren zu scheinen. Distributed Ledger werden genutzt, um revisionssichere Transaktionen durchzuführen, ohne dass sich die Teilnehmer untereinander vertrauen. Wird jedoch für ein Smart Contract ein Oracle genutzt, das die Daten aus einer externen Quelle bezieht wird Vertrauen wieder nötig [14]. Zum einem Vertrauen in die Validität der Daten aus der externen Quelle, zum anderen Vertrauen darin, dass die Daten vom eingesetzten Oracle nicht verändert wurden.

Allgemein, sind die als CIA-Triade bezeichneten Begriffe, confidentiality (Vertraulichkeit), integrity (Integrität) und availability (Verfügbarkeit), ein fundamentales Modell für Betrachtungen in der Computersicherheit, auch für Smart Contract Oracles gültig. Vertraulichkeit steht für den Schutz, dass Daten nicht unautorisiert veröffentlicht werden. Integrität ist der Schutz vor der unerlaubten Änderung von Daten und Verfügbarkeit bezeichnet den Schutz vor unautorisierter Dienstverweigerung (beispielsweise durch Distributed-Denial-of-Service-Attacken) [15]. Oracles können diese Eigenschaften ohne zusätzliche Maßnahmen nur schlecht erfüllen. Dieses konzeptionelle Problem, sich widersprechender Eigenschaften von Smart Contracts und Oracles, ist ein in der Forschung und Industrie bekanntes.

Ein Ansatzpunkt den Widerspruch aufzulösen ist die Nutzung von externen Quellen denen ohnehin vertraut wird, beziehungsweise denen vertraut werden muss. Anwendungen die beispielsweise die Validierung von staatlichen Stellen benötigen, können und müssen von eben diesen mit Daten in Form von Oracles gespeist werden, womit keiner zusätzlichen Entität vertraut werden muss [16]. Dies ist jedoch nur für eine begrenzte Anzahl von Anwendungen der Fall und mit Hinblick auf die Zielsetzung von DL, dass sichere Transaktionen zwischen Parteien abgewickelt werden sollen, die sich nicht vertrauen, auch keine primäre Anwendung.

ZHANG ET AL. (2016) setzen für das Problem der sicheren und nicht manipulierten Übertragung von Daten, von externen Quellen zu einem Oracle auf ein von ihnen entwickeltes Protokoll mit dem Namen Town Crier [17]. Town Crier setzt hierbei stark auf die Nutzung von Software Guard Extensions von Intel, wie auch der Konsensmechanismus Proof of Elapsed Time, um Code in einer sicheren Umgebung auszuführen. Dadurch soll die Authentizität und Integrität der Daten eines Oracles sichergestellt werden. Daneben soll durch die Verschlüsselung von so genannten „datagrams“ Vertraulichkeit in den Abfragen an ein Oracle gewährleistet werden. Dies ist insofern wichtig, weil in einem öffentlichen DL alle Transaktionen für jeden Teilnehmer einsehbar sind. Für viele Anwendungen ist jedoch ein gewisser Grad an Vertraulichkeit notwendig.

Ein weiterer Ansatz, der zum Beispiel von der Firma ChainLink angewandt wird, ist die Verteilung von Datenquellen und Oracles. Einfach ausgedrückt, werden für ein Oracle verschiedene Datenquellen genutzt und deren Ergebnisse zu einer Abfrage aggregiert, weitere Oracles mit teilweise anderen Datenquellen führen auch eine Abfrage durch und aggregieren die Ergebnisse. Die Ergebnisse aller Oracles werden am Ende ebenfalls aggregiert und bilden somit den Input für den Smart Contract, der einen bestimmten Parameter abfragt [18].

Mithilfe aller dieser Ansätze lassen sich Risiken in der Nutzung von Oracles für Smart Contracts minimieren. Jedoch kann, durch die ständige Weiterentwicklung des Feldes, noch kein abschließendes Urteil zur Sicherheit von Oracles getroffen werden.

6. Anwendung von DLT in Logistik und SCM

Nachdem in den vorherigen Abschnitten aufgezeigt wurde, wo die Potentiale von DLT liegen, aber auch ihre Limitierungen, soll geprüft werden, wo diese Technologien in Logistik und SCM angewandt werden. Dazu wurden 37 verschiedenen Case Studies bzw. Projekte ausgewertet, die alle öffentlich zugänglich sind. Die Anwendungsbereiche wurden in sechs Kategorien zusammengefasst. Diese sind in Tabelle 1 zusammengefasst.

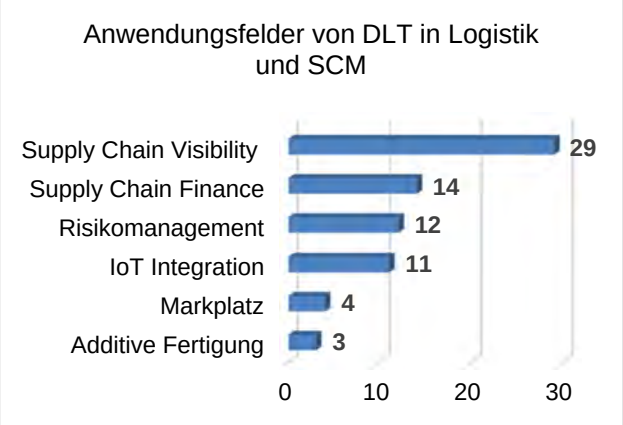


Abbildung 1 - Anwendungsfelder von DLT in Logistik und SCM

Die Case Studies und Projekte wurden zu denen in Tabelle 1 definierten Anwendungsfeldern zugeordnet. Eine Mehrfacheinordnung wurde teilweise vorgenommen. Die Ergebnisse sind in Abbildung 1 zu sehen. Das Anwendungsfeld Supply Chain Visibility, worunter auch das Tracking und Tracing fällt ist dominant in der Anwendung der DLT im SCM. Supply Chain Finance folgt mit deutlichem Abstand, was überraschend ist, da die ganze Technologie als ein neuartiges Finanzinstrument gestartet ist und in vielen Berichten primär auch noch als dieses gesehen wird. Die Integration von IoT-Geräten und Sensoren sowie Risikomanagement sind weiterhin wichtige Anwendungsfelder. Marktplätze für logistische Dienstleistungen und die Integration in die Additive Fertigung stellen derzeit Nischenanwendungen dar.

Kategorie	Supply Chain Visibility	Supply Chain Finance
Merkmale	Tracking und Tracing	Digitalisierung von Frachtdokumenten
	Herkunftsnachweise für Kunden und Endverbraucher und Behörden	Automatische Abwicklung von Zahlungen über Token, Kryptowährungen und Smart Contracts
	Audits	Know Your Customer
	Bestandsmanagement	
Kategorie	**IoT Integration**	**Additive Fertigung (AF)**
Merkmale	Integration von IoT Geräten und Sensoren zur Datenerfassung & Automatisierung	Sichern der Datenübertragung in der AF + Herkunfts-nachweise in DLT
Kategorie	**Risikomanagement**	**Marktplatz**
Merkmale	Reaktionsvermögen durch besseren Zugriff auf Daten	Sicherer Marktplatz zum Handel von Gütern und Dienstleistungen
	Sichere Qualitätsnachweise der Lieferanten	Automatische Verhandlungen M2M
	Einfachere Streitlösung	
	Schutz vor Produktfälschungen	

Tabelle 1 - Kategorisierung der Anwendungsfelder von DLT in SCM und Logistik

Daneben wurde den untersuchten Fällen ein Anwendungsbereich zugeordnet. Die Ergebnisse sind dazu in Abbildung 2 zusammengefasst.

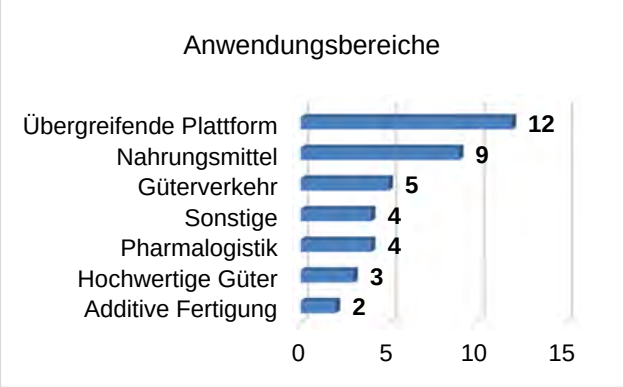


Abbildung 2 - Anwendungsbereiche von DLT in Logistik und SCM

Eine Branche in der der Einsatz von DLT stark diskutiert wird, ist die Nahrungsmittelbranche. Hier geht es primär darum das Vertrauen der Verbraucher, das durch die zahlreichen Lebensmittelskandale stark gelitten hat, wieder zu gewinnen. Sichere Herkunftsnachweise sollen den Verbrauchern eine Garantie über die Quelle ihrer Produkte geben und zwar über die ganze Supply Chain hinweg. Eine ähnliche Thematik spielt für die Anwendung in der Pharmalogistik eine Rolle. Auch hier gab es in den vergangenen Jahren Skandale gerade in Asien. Weitere Anwendungsbereiche sind der Güterverkehr und die Herstellung und Handel mit hochwertigen Gütern, worunter beispielsweise die Firma Everledger fällt, die Diamanten registriert.

In den Case Studies häufig genannte Vorteile von DLT in Logistik und SCM waren: höhere Transparenz, Sichere Herkunftsnachweise, Erhöhung von Datensicherheit und -integrität, Prozessautomatisierungen, Kostensenkungen.

Um die in den Case Studies identifizierten Vorteile weiter zu untersetzen, wird ein Literature Review durchgeführt.

7. Literature Review

Als Grundlage der eigenen Methodik dienen die Ausführungen von DURACH ET AL. (2017), die Literature Reviews im Bereich der Forschung zum Supply Chain Management systematisierten [19]. Im ersten Schritt der Methode geht es um die möglichst konkrete Formulierung der Forschungsfragen. Die vordergründig zu beantwortenden Forschungsfragen sind hierbei:

1. Welche Vorteile werden durch den Einsatz von DLT erzielt und wie sind sie quantifizierbar?
2. Welche Implikationen ergeben sich daraus auf die Datensicherheit im Speziellen und die Sicherheit der Unternehmung im Allgemeinen, beispielsweise bezogen auf die Einhaltung rechtlicher Vorgaben?

Darauffolgend werden potentiell relevante Quellen durch eine Suche in einschlägigen Datenbanken ermittelt. Neben der bloßen Suche in den Datenbanken werden auch die Querverweise der ausgewählten Publikationen untersucht. Die Suchmethode orientiert sich dabei an den von DENYER UND TRANFIELD (2009) gemachten Angaben [20]. Gefundene Quellen werden anhand der in Schritt zwei bereits definierten Kriterien, entweder in die Untersuchung aufgenommen oder davon ausgeschlossen.

8. Vorteile durch DLT in Logistik und SCM

Im Bezug zu den Vorteilen, die durch DLT erreicht werden können, decken sich die Ergebnisse der Auswertung der wissenschaftlichen Quellen zum Großteil, mit denen aus der Auswertung der Case Studies. Die Ergebnisse sind dazu in Tabelle 3 festgehalten. Eine Quantifizierung der Vorteile, die durch DLT erbracht werden, findet sich jedoch in keiner Quelle.

Von vielen Autoren werden die Automatisierung von Prozessen und die Schaffung von Transparenz als die größten Vorteile der Implementierung von DLT gesehen. Die Automatisierung bezieht sich dabei vor allem auf Finanztransaktionen, Streitresolution und Dokumentation von Prozessen. Diese Prozesse benötigen heute noch viele manuelle Interventionen, da sie nicht digitalisiert ablaufen und papierbasierte Dokumente benötigen. Bei vielen Prozessen ist dies der Fall, weil digitale Dokumente noch nicht genügend Sicherheit gegenüber Manipulation und Vervielfältigung aufweisen. Digitale Originale werden jedoch durch DLT möglich und eröffnen so neue Nutzungsmöglichkeiten und Wege zur Automatisierung von Prozessen, welche in der Regel durch Kosteneinsparung und einer höheren Güte der Prozesse verbunden sind.

Ein Beispiel, dass die Fähigkeit von DLT, Prozesse zu automatisieren, gut beschreibt ist die Digitalisierung der Bill of Lading (BoL), die auch als Konnossement bezeichnet wird. STAHLBOCK ET AL. (2018) haben sich damit auseinandergesetzt [21].

Die BoL ist ein wichtiges Dokument im maritimen Warenverkehr. Ihre hauptsächlichen Funktionen sind: „Beleg des Erhalts oder Verschiffens von Gütern, Beweis eines abgeschlossenen Frachtvertrags und die Repräsentation des Besitzrechtes der Güter.“ [21]. Durch den letzten Punkt wird die Digitalisierung des Dokumentes erschwert, denn es stellt auch ein Wertpapier dar. Also muss sichergestellt werden, dass eine digitale Repräsentation dieses Dokuments nicht vervielfältigt werden kann, dass also nur ein digitales Original besteht.

Warum ist eine Digitalisierung sinnvoll? Durch die Involvierung vieler verschiedener Parteien, beim Transport von Seefracht, kann es dazu kommen, dass die BoL nicht immer dort ist, wo sie gerade benötigt wird, beispielsweise um Waren freizugeben, die am Zielort angekommen sind. So kann die BoL, die nicht vervielfältigt werden darf, noch bei der Hafenverwaltung sein, obwohl sie vom Zoll benötigt wird. Dadurch kann es zu Verzögerung beim Löschen der Ware kommen, was in der Regel zu finanziellen Einbußen führt (z.B. könnten empfindliche Nahrungsmittel verderben). Daneben ist eine BoL als Papierdokument fehleranfällig, sie könnte beispielsweise verloren gehen.

Eine digitale BoL muss einige Anforderungen erfüllen. Die digitale BoL muss einzigartig und übertragbar sein. Die Integrität der Dokumente muss sichergestellt sein und, es muss ein Mechanismus bestehen, mit dem der Besitz des Dokumentes nachgewiesen werden kann, nach Ende des Geschäfts muss das Dokument seine rechtliche Wirkung verlieren [21]. Diese Eigenschaften lassen sich mit DLT erreichen. Ein Nachweis über den Besitz des Dokumentes lässt sich beispielsweise von jedem Teilnehmer, beim Blick auf die Transaktionen im Ledger, nachvollziehen. Durch eine mit DLT digitalisierte BoL werden manuelle Schritte wie das Übergeben der BoL nach dem Festmachen des Schiffes an die jeweiligen Autoritäten unnötig. Sie könnte gleich bei der Einfahrt in den Hafen automatisch übermittelt werden. Somit könnten Wartezeiten vermieden und die Ware schneller gelöscht werden.

Weitere Anwendungen im Bereich der Automatisierung gehen stark auf die Abwicklung von Finanzflüssen ein. Ein Ziel im SCM ist die Integration von Waren-, Informations- und Zahlungsflüssen. Die Abwicklung von Zahlungsflüssen ist jedoch noch stark von den eigentlichen Warenflüssen entkoppelt. Wie bereits beschrieben ist dies vor allem auf die weitverbreitete Nutzung papierbasierter Dokumente zurückzuführen. Analog zum Beispiel der BoL sind in diesem Bereich auch Vorteile durch die Digitalisierung solcher Dokumente zu erzielen. Daneben entstehen durch die automatisierte Abwicklung von Zahlungsströmen, auch in M2M-Beziehungen, neue Ge-

schäftsmodelle, wie dezentralisierte autonome Organisationen oder P2P-Netzwerke zum Handeln von Gütern z.B. elektrische Energie aus privaten regenerativen Quellen.

Vorteile durch DLT	Anzahl	Quellen
Automatisierung	12	[21]; [26]; [27];[28]; [29];[30]; [31]; [32]; [33];[34]; [35];[36]
Transparenz (unternehmensintern)	8	[26]; [32]; [34]; [35]; [37]; [38]; [39]; [40]
Transparenz (extern für Endverbraucher)	5	[25]; [37]; [41]; [42]; [43]
Neue Geschäftsmodelle	4	[28]; [29]; [44];[45];
Kontrollebene in IoT-Systemen	3	[27]; [39]; [43]
Produktfälschungen verhindern	3	[39]; [42], [46]
Interoperabilität von Systemen	2	[38]; [47]
Besserer Zugang zu Finanzierungsinstrumenten für KMU	1	[31]

Tabelle 2 - Auswertung Literature Review Vorteile durch DLT

Ein letzter Punkt der Automatisierung beschreibt die Streitresolution. Mithilfe von Smart Contracts können Logiken festgelegt werden, nach denen Algorithmen zwingend handeln, ohne dass eine Partei darin eingreifen könnte. Ein Beispiel hierzu sind Smart Contracts, die die Kühlung eines Medikamentes überwachen, sollten die Temperaturen über einen vorher definierten Zeitraum, nicht eingehalten werden, muss der Logistikdienstleister dem Hersteller eine Entschädigung zahlen, da das Produkt möglicherweise nicht mehr verkäuflich ist. Mit Smart Contracts wird sichergestellt, dass die Entschädigung auch gezahlt wird, da diese eine Zahlung automatisch veranlassen können, sobald der gewünschte Transportzustand nicht mehr erfüllt wird. Verhandlungen mit dem Transportdienstleister über solche Entschädigungen sind dann nicht mehr nötig.

Die Erzielung von Transparenz in Supply Chains ist, laut der Auswertung der Literature Reviews, neben der Automatisierung der bestimmende Vorteil von DLT im SCM. Es kann grob zwischen dem Ziel nach der Schaffung von unternehmensinterner und von externer Transparenz unterschieden werden. Externe Transparenz bezieht sich dabei auf die Schaffung von Transparenz für den Endkonsumenten eines Produktes. Das heißt dem Kunden wird die Möglichkeit geboten die Herkunft seines Produktes und die Produktionsbedingungen nachzuvollziehen. Dies ist eine Möglichkeit, Vertrauen zwischen Unternehmen und Kunden aufzubauen und so die Kundenbindung zu stärken. Einher geht dieser Ansatz auch oft mit dem Ziel, Produktfälschungen zu verhindern, beziehungsweise zu erschweren, dadurch, dass sich nur genuine Produkte über eine Abfrage in einem DL verifizieren lassen.

Interne Transparenz bezieht sich ausschließlich auf das Unternehmensnetzwerk. Der Grund hierfür liegt in der Absicherung der Unternehmen hinsichtlich der Herkunft der von Lieferanten bezogenen Teile oder Rohstoffe. So ist ein wichtiger Ansatzpunkt im Risikomanagement, der ausschließliche Bezug von Teilen und Rohstoffen aus bekannten Quellen. Dass ein bestimmtes Teil tatsächlich von einem bestimmten Lieferanten stammt, lässt sich mithilfe von DLT anhand der getätigten Transaktionen leicht überprüfen.

Grundlegend bei fast allen der diskutierten Anwendungen ist eine Datenbasis aus vielen Sensoren und Geräten. Somit sind diese Anwendungen von einer primären IoT-Infrastruktur abhängig. Um diese sicherer zu machen werden DLT in einigen Publikationen als eine Art Kontrollinstanz gesehen, die Informationen bzw. Transaktionen validiert und diese Netzwerke so zusätzlich schützt. Somit sind IoT und DLT fundamentale Bausteine für eine Verbesserung der Visibility, des Risikomanagements und der Abwicklung von Zahlungsflüssen in Supply Chains.

9. Sicherheitsimplikationen

Der Einsatz von DLT wird oft unter dem Aspekt einer Verbesserung der Sicherheit diskutiert. Daher wurden die Quellen, die sich mit der Anwendung der Technologie beschäftigen, auch dahingehend untersucht, wie DLT eine erhöhte Sicherheit von Anwendungen erzielen und welche Punkte sich möglicherweise negativ darauf auswirken.

Das am häufigsten angebrachte Argument dafür, dass DLT mehr Sicherheit schafft, ist dass dadurch die Manipulierbarkeit von Daten stark reduziert wird oder sogar, mit vertretbaren Mitteln, ganz ausgeschlossen ist. Gerade in Bezug auf IoT-Geräte, die allzu oft leicht angreifbar sind und bei Finanztransaktionen, ist eine Verbesserung der Manipulationssicherheit gewünscht.

Die Forderung nach Manipulationssicherheit ist im Grunde genommen, dieselbe wie die Forderung nach Datenintegrität, einer der drei Säulen der Informationssicherheit, nach dem Modell der CIA-Triade [15]. In vielen wissenschaftlichen Quellen wird davon ausgegangen, dass DLT grundsätzlich sicher sind. Begründet wird dies mit den genutzten kryptographischen Verfahren und den verteilten Konsens. Anhand von Smart Contracts wurde bereits gezeigt, dass Anwendungen auf Grundlage von DLT nicht zwangsläufig sicher sind. Daher ist es wichtig an dieser Stelle zu prüfen, ob die oft getätigte Aussage „Blockchains sind sicher“ wirklich zutrifft und somit auch ihre Eignung Daten manipulationssicher zu verarbeiten.

Sicherheitsimplikationen durch DLT	Anzahl	Quellen
Manipulierbarkeit von Daten verringern	12	[21]; [26]; [27]; [31]; [32]; [33]; [35]; [36]; [40]; [43], [44]; [47]; [51];
Produktsicherheit für Verbraucher	6	[25]; [37]; [41]; [42]; [46]; [48]
Sicherheit in IoT Anwendungen (z.B. verhindern von DDOS-Attacken, Schutz von Updates)	4	[45]; [49]; [50]; [51]
Kein Single-Point of Failure	4	[30]; [32]; [44] [49];
Klärung von Haftungsfragen	2	[38]; [40]
Manuelle Eingaben gefährden das System	1	[25]

Tabelle 3 - Auswertung Literature Reviews Sicherheitsimplikationen durch DLT

Aufgrund des sich schnell entwickelnden Feldes an DLT gibt es bislang keine allgemeine Übersicht über die Sicherheit aller Systeme. Das liegt auch daran, dass zwar viele DLT ähnliche Konzepte nutzen, die sich jedoch in ihrer Implementierung stark unterscheiden können. Grundsätzlich gilt, dass zwar in vielen Publikationen behauptet wird, dass DLT sicher sind, jedoch kann es für kein System eine einhundertprozentige Sicherheit geben. Einschätzungen zur Sicherheit können daher nur für weitverbreitete Systeme wie Bitcoin oder Ethereum abgegeben werden. Implementationen von konsortialen oder Private Blockchains, können aufgrund der individuellen Anpassung nur im konkreten Anwendungsfall auf ihre Sicherheit hin untersucht werden.

Gefährdet waren Bitcoin und andere Blockchains bisher dort, wo Berührungspunkte zur Außenwelt bestehen. Gab es bis jetzt noch keinen erfolgreichen Angriff auf die Bitcoin-Blockchain, so wurden doch schon durch Hackerangriffe Bitcoins von Währungsbörsen oder anderen Applikationen gestohlen. Smart Contracts können, wie bereits angesprochen, ein weiterer Schwachpunkt sein. So wurden schon mehrmals durch falsch konfigurierte Smart Contracts auf Ethereum Angriffe möglich, die schwerwiegende finanzielle Schäden angerichtet haben [22]. Die Blockchains der beiden größten Netzwerke Bitcoin und Ethereum waren jedoch bis jetzt noch nicht betroffen. Somit ist bei der Betrachtung der Sicherheit zwischen der Technologieebene und der Anwendungsebene zu unterscheiden.

Trotzdem gibt es theoretische Angriffsmöglichkeiten auf die Blockchain an sich. Die bekannteste ist die sogenannte 51%-Attacke. Der Konsens wird verteilt getroffen, das heißt, dass immer ein zufälliger Knoten, anhand einer bestimmten Ressource, ausgewählt wird, der den nächsten Block propagieren darf. Erreicht eine Entität im Netzwerk mehr als die Hälfte der Ressourcen zu kontrollieren, bei einem PoW-Verfahren ist dies die Hashrate, so kann diese erfolgreich Double-Spends ausführen oder Transaktionen modifizieren [23]. Wie schwer es ist in einem Netzwerk 51% der Ressourcen zu sammeln ist von dem Aufbau und der Nutzerzahl abhängig. Dass eine einzelne Person 51% der Hashrate im Bitcoin-Netzwerk kontrolliert ist sehr unwahrscheinlich, da dies mit hohen Kosten verbunden wäre. Mining Pools können jedoch sehr nahe an diese Grenze stoßen, was ein Risiko darstellt. Daneben existieren andere mögliche Attacken, die an dieser Stelle nicht erörtert werden sollen, LI ET AL. (2017) geben dafür eine gute Übersicht. Das zeigt, dass Angriffe auf Blockchains möglich sind, verglichen mit zentralen Datenbanken verfügen sie jedoch über weitaus stärkere inhärente Schutzmechanismen, zumindest auf der Technologieebene [23].

Grundsätzlich eignen sich also DLT, um manipulationssicher Daten zu verarbeiten. Somit können DLT einen Beitrag zu Datenintegrität leisten und die Informationssicherheit stärken. Wenn von Datenintegrität die Rede ist, bezieht sich dies nicht nur auf den Schutz vor unerlaubter Veränderung der Daten von außen. Auch eine zentrale Instanz, die eine Datenbank verwaltet kann Einträge manipulieren, um daraus Vorteile zu ziehen. So könnte ein Logistikdienstleister die Datenbankeinträge über die Kühlung eines Produktes so manipulieren, dass ein Ausfall der Kühlung nicht mehr in den Einträgen auftaucht, um so Schadensersatzforderungen aus dem Weg zu gehen. Ist das System jedoch dezentral reicht es nicht aus, wenn eine Partei einen Manipulationsversuch startet, somit ist das System als Ganzes sicherer vor Manipulation.

Dadurch kann die Glaubhaftigkeit und Reputation von Informationen gesteigert werden, welche nach WANG UND STRONG (1996) Eigenschaften der Datenqualität darstellen [24]. Wie bereits geschildert korreliert die Datenqualität in vielen Untersuchungen mit der Performance der Supply Chain. Wodurch die Einführung von DLT einen positiven Beitrag zur Verbesserung der Leistung der ganzen Supply Chain leisten kann. Zu beachten gilt jedoch, dass andere Faktoren der Datenqualität nicht von der Blockchain beeinflusst werden. Ungenaue Eingaben, die auf menschliche Fehler zurückzuführen sind, können die Vorteile der Technologie wieder ausgleichen, da den Daten dadurch wieder nicht vertraut werden kann. Eine Nutzung der Technologie ist also nur ohne manuelle Dateneingabe sinnvoll [25].

Neben einer höheren Datenintegrität schaffen DLT mehr Sicherheit dadurch, dass sie keinen Single-Point-of-Failure aufweisen. Das heißt, dass wenn eine zentrale Datenbank angegriffen wird und dadurch nicht verfügbar ist, können die darin enthaltenen Daten nicht genutzt werden, was zu negativen wirtschaftlichen Konsequenzen führen kann. Fällt jedoch ein Knoten eines dezentralen Netzwerkes aus, können die Daten immer noch von den anderen Knoten abgerufen werden.

Ein letzter wichtiger Punkt, wie DLT Sicherheit verbessern kann bezieht sich indirekt auf die Akteure der Supply Chain. DLT kann es ermöglichen Verbrauchern mehr Produktsicherheit zu geben. Das bezieht sich vor allem auf Produkte wie Medikamente oder Nahrungsmittel. Wird ein Problem offenkundig z.B. eine Kontamination eines Nahrungsmittels, so kann durch die erhöhte Transparenz nachvollzogen werden, wo das Problem auftrat und Kunden können feststellen ob sie betroffen sind. Somit können Unternehmen schneller reagieren und geeignete Maßnahmen treffen das Problem zu beseitigen, was die Produktsicherheit erhöht.

10. Zusammenfassung

DLT scheinen eine Gruppe an Technologien zu sein, die Vorteile für SCM und Logistik bergen. Sie sind durch eine geringe Reife und Verbreitung gekennzeichnet, die sich zum einen an den Umsetzungsgraden von relevanten Projekten, zeigen. Zum anderen ist die Anzahl an wissenschaftlichen Veröffentlichungen, speziell im Spannungsbereich zwischen SCM und DLT, sehr gering. In der Auswertung der Literatur zum Thema wurde deutlich, dass teilweise unterschiedliche Auffassungen über die Definitionen und Eigenschaften von DLT bestehen. Daher sollten zukünftige Forschungsarbeiten weiter an einer Standardisierung der Definitionen arbeiten. Dies ist gerade nötig, da mittlerweile unter den Begriff DLT eine Vielzahl verschiedener Ausprägungen zusammengefasst werden. Außerdem finden sich in der Literatur keine Aussagen über die Quantifizierung der Vorteile, die DLT schaffen sollen. Auch hier sollte in der Zukunft ein Forschungsschwerpunkt liegen.

In der Untersuchung haben sich vier Bereiche herausgestellt, in denen sich Vorteile durch DLT erreichen lassen. Ein Bereich, der ganz klar mit der Herkunft von DLT als Zahlungssystem zusammenhängt, ist die Supply Chain Finance. Hier können DLT dazu beitragen, Prozesse zu automatisieren und zu digitalisieren. Dies ist dadurch möglich, dass es mit der Technologie erstmals möglich ist genuine digitale Originale zu schaffen. Dadurch wird eine stärkere Integration von Waren- und Zahlungsflüssen möglich.

Weitere Bereiche sind das Risikomanagement, SCV und die Integration von IoT-Netzwerken. Dabei dient der letztere Bereich vor allem als Basis für die vorher genannten. Das Internet-of-Things soll für die umfassende Abbildung der realen Welt im Digitalen sorgen, jedoch sind diese Anwendungen mit zahlreichen Sicherheitsproblemen behaftet. DLT wird als eine Art Kontrollebene gesehen, die Sicherheitsprobleme solcher Anwendungen mildern kann, was zu einer weiteren Verbreitung in der Industrie führen könnte. Zudem können DLT eine finanzielle Transaktionsebene zwischen einzelnen Geräten und Maschinen schaffen, womit neue Geschäftsmodelle ermöglicht werden.

Die Bereiche Risikomanagement und SCV profitieren von einer soliden Datenbasis aus IoT-Netzwerken, da sich dadurch ein schnelleres Reaktionsvermögen auf Ereignisse in der Supply Chain ergibt. Die Schaffung von Transparenz über Prozesse, Bestände, Nachfragen und Ereignisse trägt zu einer höheren Supply Chain Performance bei. Diese ist jedoch von der Qualität der Daten abhängig, wichtige Metriken sind hier die Glaubhaftigkeit, Reputation und Zugänglichkeit von Daten. DLT können diese Kategorien stärken, da sie es ermöglichen in einem Umfeld von Akteuren, die sich nicht vertrauen, Vertrauen zu schaffen. Dies ist durch die Dezentralität und der kryptographischen Absicherung der Anwendung der Fall. Es ist also für einen einzelnen Akteur nicht möglich Daten nachträglich zu manipulieren, die Datenintegrität und somit die Datensicherheit werden gestärkt. Dies vergrößert das Vertrauen in den Datenbestand, womit es für Unternehmungen einfacherer ist auf Basis der Daten Entscheidungen zu treffen. Des Weiteren wird durch die Dezentralität eine höhere Zugänglichkeit zu den Daten gewährt, da der Ausfall eines Knotens für das Gesamtsystem verkraftbar ist, weil der Datenzugriff über die anderen Knoten weiterhin möglich ist.

Literaturverzeichnis

[1] Bundesvereinigung Logistik (BVL) (2017): Logistikumsatz und Beschäftigung. Online verfügbar unter https://www.bvl.de/service/zahlen-daten-fakten/umsatz-und-beschaeftigung, zuletzt geprüft am 31.08.2020.

[2] Bundesamt für Sicherheit in der Informationstechnik (Hg.) (2019): Die Lage der IT-Sicherheit in Deutschland 2019. Bonn. Online verfügbar unter https://bit.ly/3lGBwer.

[3] J. Sürmeli, U. Der, S. Jähnichen, A. Vogelsang (2017): Ein Rahmenwerk zur Protokollierung von Transaktionen in Distributed Ledgers. In: Informatik-Spektrum 40 (6), S. 595–601.

[4] V. Brühl (2017): Bitcoins, Blockchain und Distributed Ledgers. In: Wirtschaftsdienst 97 (2), S. 135–142.

[5] M. Kaulartz, J. Heckmann (2016): Smart Contracts – Anwendungen der Block-chain-Technologie. In: Computer und Recht 32 (9).

[6] V. Morabito (2017): Business Innovation Through Blockchain. Cham: Springer International Publishing

[7] M. Bartoletti, L. Pompianu (2017): An Empirical Analysis of Smart Contracts. Platforms, Applications, and Design Patterns. In: Michael Brenner, Kurt Rohloff, Joseph Bon-neau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague et al. (Hg.): Financial Cryptography and Data Security. Cham: Springer International Publishing, S. 494–509.

[8] J. Ream, Y. Chu, D. Schatsky (2016): Upgrading blockchains: Smart contract use cases in industry. Hg. v. Deloitte University Press. Online verfügbar unter https://bit.ly/32F5A17.

[9] C. Jentzsch (2016): Decentralized Autonomous Organization To Automate Govern-ance. Online verfügbar unter https://bit.ly/2QG79X6, zuletzt geprüft am 20.06.2020.

[10] M. Biederbeck (2016): Der DAO-Hack. Ein Blockchain-Krimi aus Sachsen. Online verfügbar unter

https://bit.ly/34PORuS, zuletzt geprüft am 22.06.2020.
[11] K. Delmolino, M. Arnett, A. Kosba, A. Miller, E. Shi (2016): Step by Step Towards Creating a Safe Smart Contract. Lessons and Insights from a Crypto-currency Lab. In: J. Clark, S. Meiklejohn, P. Y.A. Ryan, D. Wallach, M. Brenner und K. Rohloff (Hg.): Financial Cryptography and Data Security. FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers. Berlin: Springer. (Lecture Notes in Computer Science), S. 79–94.
[12] L. Luu, D. Chu, H. Olickel, P. Saxena, A. Hobor (2016): Making Smart Contracts Smarter. In: E. Weippl und S. Katzenbeisser (Hg.): CCS '16. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, S. 254–269.
[13] I. Nikolic, A. Kolluri, I. Sergey, P. Saxena, A. Hobor, (2018): Finding The Greedy, Prodigal, and Suicidal Contracts at Scale.
[14] W. Blocher (2018): C2B statt B2C? Auswirkungen von Blockchain, Smart Contracts & Co. auf die Rolle des Verbrauchers. In: P. Kenning und J. Lamla (Hg.): Entgrenzungen des Konsums. Dokumentation der Jahreskonferenz des Netzwerks Verbraucherforschung. Wiesbaden: Springer Fachmedien Wiesbaden, S. 87–108.
[15] Y. Cherdantseva, J. Hilton (2013): A Reference Model of Information Assurance & Security. In: 2013 International Conference on Availability, Reliability and Security. ARES 2013. Regensburg, 02.-06.09.2013. The Institute of Electrical and Electronics Engineers, Inc. Piscataway: IEEE, S. 546–555.
[16] X. Xu, C. Pautasso, L. Zhu, V. Gramoli,A. Ponomarev, A. Tran, S. Chen (2016): The Blockchain as a Software Connector. In: H. Muccini und K. E. Harper (Hg.): WICSA 2016. 2016 13th Working IEEE/IFIP Conference on Software Architecture: proceedings. Venedig. Piscataway: IEEE, S. 182–191.
[17] F. Zhang, E. Cecchetti, K. Croman, A. Juels, E. Shi (2016): Town Crier. An Authenticated Data Feed for Smart Contracts. In: E. Weippl und S. Katzenbeisser (Hg.): CCS '16. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, S. 270–282.
[18] S. Ellis, A. Juels, S. Nazarov (2017): ChainLink. A Decentralized Oracle Network. Online verfügbar unter https://link.smartcontract.com/whitepaper.
[19] C. F. Durach, J. Kembro, A. Wieland (2017): A New Paradigm for Systematic Literature Reviews in Supply Chain Management. In: J Supply Chain Manag 53 (4), S. 67–85.
[20] D. Denyer, D. Tranfield (2009): Producing a Systematic Review. In: D. A. Buchanan und A. Bryman (Hg.): The Sage handbook of organizational research methods. Los Angeles: Sage, S. 671–689.
[21] R. Stahlbock, L. Heilig, S. Voß (2018): Blockchain in der maritimen Logistik. In: HMD Praxis der Wirtschaftsinformatik, S. 1–19.
[22] M. Orcutt (2018): How secure is blockchain really? It turns out "secure" is a funny word to pin down. MIT Technology Review. Online verfügbar unter https://bit.ly/3gGej8p.
[23] X. Li, O. Jiang, T. Chen, X. Luo, Q. Wen (2017): A survey on the security of blockchain systems. In: Future Generation Computer Systems.
[24] R. Wang, D. M. Strong (1996): Beyond accuracy. What data quality means to data consumers. In: Journal of Management Information Systems 12 (4), S. 5–33.
[25] T. K. Agrawal, A. Sharma, V. Kumar (2018): Blockchain-Based Secured Traceability System for Textile and Clothing Supply Chain. In: S. Thomassey und X. Zeng (Hg.): Artificial Intelligence for Fashion Industry in the Big Data Era. Singapore: Springer, S. 197–208.
[26] H. R. Hasan, K. Salah (2018): Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters. In: IEEE Access 6, S. 46781–46793.
[27] T. Bocek, B. B. Rodrigues, T. Strasser, B. Stiller, (2017): Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. In: P. Chemouil (Hg.): Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network Management. Lissabon. Piscataway: IEEE, S. 772–777.
[28] A. Bahga V. K. Madisetti (2016): Blockchain Platform for Industrial Internet of Things. In: Journal of Software Engineering and Applications 9, S. 533.
[29] M. Kupferberg, P. Sandner, M. Felder (2018): Blockchain-basierte Abrechnung der IoT-registrierten Stationshalte: ein Proof-of-Concept auf Basis von Ethereum. Frankfurt am Main.
[30] K. Korpela, J. Hallikas, T. Dahlberg: Digital Supply Chain Transformation to-ward Blockchain Integration. In: Hawaii International Conference on System Sciences 2017 (HICSS-50). Hawaii, January 4-7, 2017. Hawaii International Conference on System Sciences; HICSS, S. 4182–4191.
[31] E. Hofmann, U. M. Strewe, N. Bosia (2018): Supply Chain Finance and Blockchain Technology. Cham: Springer International Publishing.
[32] Y. Omran, M. Henke, R. Heines, E. Hofmann, (2017): Blockchain-driven supply chain finance. Towards a conceptual framework from a buyer perspective. In: IPSERA 2017. Budapest, S. 1–15.
[33] M. Witthaut, H. Deeken, P. Sprenger, P. Gadzhanov, M. David (2017): Smart Objects and Smart Finance for Supply Chain Management. In: Logistics Jounal: referierte Veröffentlichungen 2017 (10).
[34] B. Nicoletti (2018): Agile Procurement. Volume II: Designing and Implementing a Digital Transformation. Cham: Springer International Publishing.

[35] S. Tönnissen, F. Teuteberg (2018): Using Blockchain Technology for Business Processes in Purchasing. Concept and Case Study-Based Evidence. In: W. Abramowicz und A. Paschke (Hg.): Business Information Systems. Cham: Springer International Publishing (320).

[36] S. C. Eickemeyer, T. Halaszovich, C. Lattemann, (2018): Blockchain Technologien für die Sicherung von Material-, Informations- und Geldflüssen in der Logistik – Erfolgsfaktoren für die chinesische „Belt-Road“ Initiative. In: HMD Praxis der Wirtschaftsinformatik, S. 1–14.

[37] Y. Cui, H. Idota (2018): Improving Supply Chain Resilience with Establishing A Decentralized Information Sharing Mechanism. In: Proceedings of the 5th Multidisciplinary International Social Networks Conference - MISNC '18. Saint-Etienne, 16-18. Juli 2018. New York: ACM Press, S. 1–7.

[38] A. Imeri, C. Feltus, D. Khadraoui, N. Agoulmine, D. Nicolas (2018): Solving the trust issues in the process of transportation of dangerous goods by using blockchain technology. In: P. Reinecke, P. Burnap, N. Moradpoor, A. Elçi, G. Theodorakopoulos, O. Rana und K. Karabina (Hg.): Proceedings of the 11th International Conference on Security of Information and Networks - SIN '18. Cardiff, 10/9/2018 - 12/9/2018. New York: ACM Press, S. 1–2.

[39] B. Alangot, K. Achuthan (2018): Trace and Track: Enhanced Pharma Supply Chain Infrastructure to Prevent Fraud. In: N. Kumar und A. Thakre (Hg.): Ubiquitous Communications and Network Computing. First International Conference, UBICNET 2017, Bangalore, India, August 3-5, 2017, Proceedings, S. 189–195

[40] B. Yahsi (2017): Financial Supply Chain Management. Erfolgsfaktoren der Gestaltung von Finanznetzwerken. Dissertation, Darmstadt, Technische Universität Darmstadt, 2017.

[41] Q. Lu, X. Xu (2017): Adaptable Blockchain-Based Systems. A Case Study for Product Traceability. In: IEEE Softw. 34 (6), S. 21–27.

[42] V. A. J. Boehm J. Kim, J. W. Hong (2018): Holistic Tracking of Products on the Blockchain Using NFC and Verified Users. In: B. Kang und T. Kim (Hg.): Information security applications. 18th international conference, WISA 2017, Jeju Island, Korea, August 24-26, 2017 : revised selected papers. Cham: Springer (Lecture Notes in Computer Science, 10763), S. 184–195.

[43] F. Tian (2017): A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In: J. Tang, J. Chen und X. Cai (Hg.): The 14th International Conference on Services Systems and Services Management (ICSSSM2017). June 16-18, 2017, Dalian, China : proceedings. Piscataway: IEEE, S. 1–6.

[44] A. Reyna, C. Martín, J. Chen; E. Soler, M. Díaz, (2018): On blockchain and its integration with IoT. Challenges and opportunities. In: Future Generation Computer Systems 88, S. 173–190.

[45] K. Christidis, M. Devetsikiotis (2016): Blockchains and Smart Contracts for the Internet of Things. In: IEEE Access 4, S. 2292–2303.

[46] K. Toyoda. T. Mathiopoulos, I. Sasase, T. Ohtsuki (2017): A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain. In: IEEE Access 5, S. 17465–17477.

[47] Y. Yang, Y. Yang, J. Chen, M. Liu (2018): Application of Blockchain in Internet of Things. In: X. Sun, Z. Pan und E. Bertino (Hg.): Cloud Computing and Security. 4th International Conference, ICCCS 2018, Haikou, China, June 8-10, 2018, Revised Selected Papers, Part II. Cham: Springer International Publishing (11064), S. 73–82.

[48] H. L. à Nijeholt,J. Oudejans, Z. Erkin (2017): DecReg. A Framework for Preventing Double-Financing using Blockchain Technology. In: S. Lokam, S. Ruj und K. Sakurai (Hg.): Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts - BCC '17. Abu Dhabi. New York: ACM Press, S. 29–34.

[49] N. Kshetri (2017): Can Blockchain Strengthen the Internet of Things? In: IT Prof. 19 (4), S. 68–72.

[50] D. Minoli, B. Occhiogrosso (2018): Blockchain mechanisms for IoT security. In: Internet of Things 1-2, S. 1–13.

[51] M. A. Khan, K. Salah (2018): IoT security. Review, blockchain solutions, and open challenges. In: Future Generation Computer Systems 82, S. 395–411.

EXCLUSIVE MINING OF BLOCKCHAIN TRANSACTIONS

Elias Strehle[1], Lennart Ante[1, 2]
[1] Blockchain Research Lab, Max-Brauer-Allee 46, D-22765 Hamburg
[2] Universität Hamburg, Von-Melle-Park 5, D-20146 Hamburg

After creating a new blockchain transaction, the next step usually is to make miners aware of it by having it propagated through the blockchain's peer-to-peer network. We study an unintended alternative to peer-to-peer propagation: Exclusive mining. Exclusive mining is a type of collusion between a transaction initiator and a single miner (or mining pool). The initiator sends transactions through a private channel directly to the miner instead of propagating them through the peer-to-peer network. Other blockchain users only become aware of these transactions once they have been included in a block by the miner. We identify three possible motivations for engaging in exclusive mining: (i) reducing transaction cost volatility ("confirmation as a service"), (ii) hiding unconfirmed transactions from the network to prevent frontrunning and (iii) camouflaging wealth transfers as transaction costs to evade taxes or launder money. We further outline why exclusive mining is difficult to prevent and introduce metrics which can be used to identify mining pools engaging in exclusive mining activity.

1. Introduction: What is exclusive mining?

Every blockchain user can create new transactions. These transactions are regarded as unconfirmed until they have been mined, i.e. included in a new block. In principle, every user can mine new blocks and thus confirm his own transactions. In practice, however, mining on popular blockchains like Bitcoin and Ethereum has become so resource-intensive that it is only performed by a handful of miner collectives, known as mining pools.[1] The vast majority of blockchain users must therefore rely on miners to have their transactions confirmed.

How do miners become aware of new transactions? In the absence of a central coordinator, blockchains rely on their peer-to-peer network to transmit new transactions. In a peer-to-peer network, every network node maintains connections with a number of peer nodes. Whenever a node receives or creates new information (e.g. a new transaction), it forwards the information to its peers, which forward it to their own peers, and so on. In this way, information is propagated through the network quickly and reliably.

To make transactions attractive to miners, they typically include a fee. The miner who confirms the transaction can redeem the fee. This approach – propagating a transaction through the peer-to-peer network and offering a fee to whoever confirms it – enables users to have their transactions confirmed without interacting with a miner directly, even without knowing who the miners are. In principle, every blockchain user has a shot at confirming the transaction and collecting the transaction fee. This maximizes the probability that the transaction is confirmed quickly. It also limits the power of every individual miner to censor transactions or demand excessive fees.

Every blockchain node is a black box to its peers, characterized only by the information it chooses to share. As a result, most blockchains cannot enforce full compliance with their protocol. The Bitcoin protocol, for example, prescribes that all blockchain nodes should forward all new transactions to their peers.[2] But the absence of a central coordinator and the inherent unreliability of a peer-to-peer network on the internet makes misbehaviour difficult to detect. Slow connections and failing nodes occur on a regular basis, meaning that some nodes might become aware of a transaction very late or not at all. It is therefore not possible for other nodes to determine whether a suspicious node deliberately withheld a transaction or simply did not receive it.

This non-enforceability of protocol opens the door to an alternative way of having transactions confirmed, which we refer to as *exclusive mining*.

In exclusive mining, a transaction initiator and a miner set up a private communication channel outside the blockchain network. Through this channel, the initiator sends transactions directly to the colluding miner. Neither the initiator nor the miner propagates the transactions through the peer-to-peer network; no other network members can become aware of the unconfirmed transactions. The miner then confirms the transactions by including them in new blocks, collecting the associated transaction fees in the process. All other members of the network only become aware of the exclusively mined transactions as part of the blocks in which the miner has confirmed them. Table 1 contains a systematic comparison of the two approaches. Figure 1 compares the information flows of regular mining and exclusive mining.

To our knowledge, exclusive mining has not been discussed in the academic literature until now. A close relative, however, is studied in Babaioff et al. [1]. The authors observe that blockchain nodes have no incentive to forward new transactions to their peers. In fact, miners have an incentive to do the opposite and keep transactions secret in the hope of

[1] We use the term miner to refer to both individual miners and mining pools.

[2] For details, see Chapter 7 of Antonopoulos [25].

Table 1. Comparison of regular mining and exclusive mining.

	Regular mining	Exclusive mining
Cost for initiator	Transaction fee	Off-chain payment + Transaction fee (can be lower or higher than in regular mining)
Gain for colluding miner	If colluding miner confirms transaction first: Transaction fee Otherwise: Zero	Off-chain payment + Transaction fee (guaranteed)
Visibility of unconfirmed transaction	Network aware of unconfirmed transaction	Network unaware of unconfirmed transaction
Time until confirmation	Depends on size of transaction fee	Depends on hashrate of coll. miner

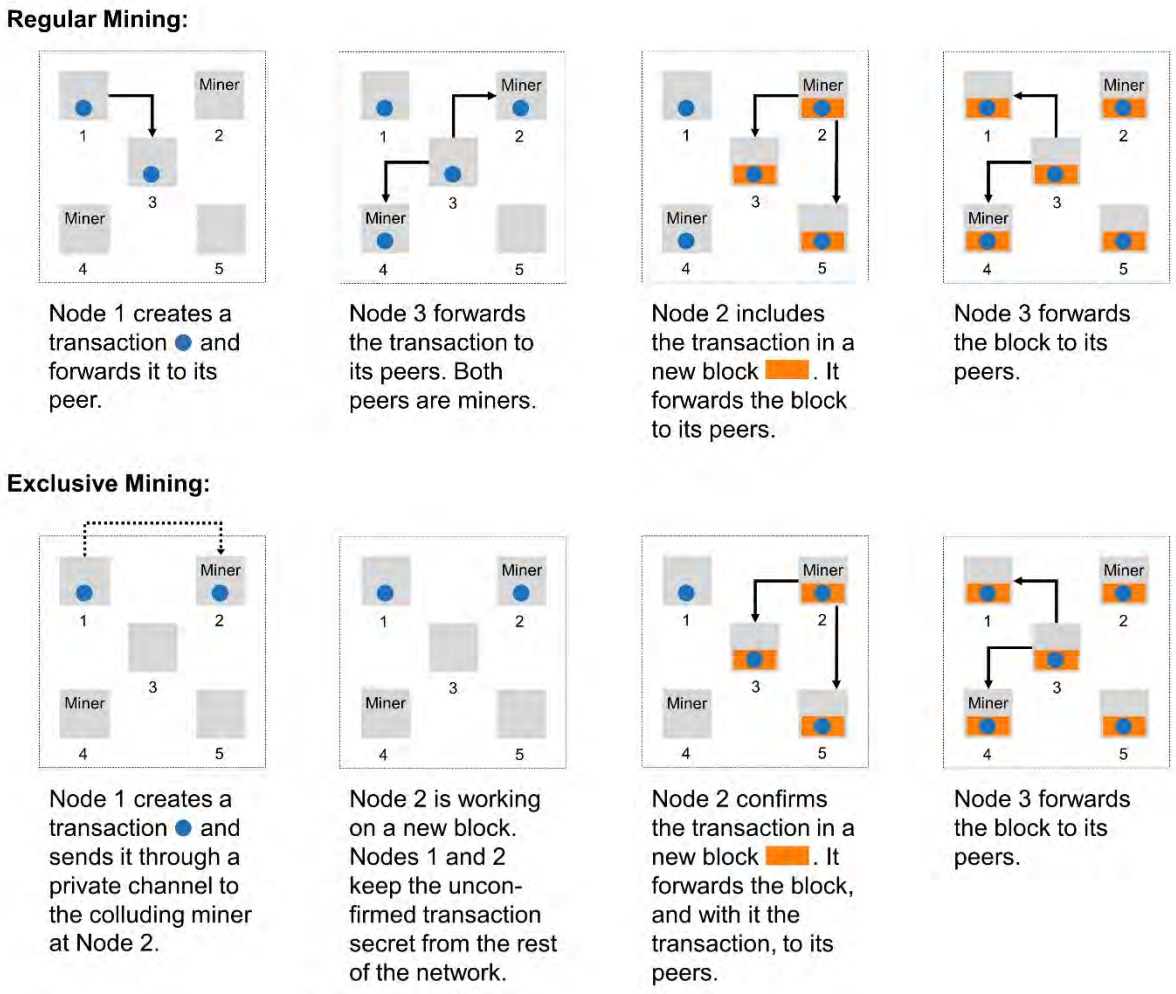


Figure 1: Information flow in regular mining and exclusive mining.

being the only one who can earn the associated transaction fees. While this observation has resonated in academia (see e.g. a proposed solution in Ersoy et al. [2]), it appears to be irrelevant in practice. Propagation through a peer-to-peer network is highly robust to misbehaving nodes, unless there are very many of them or they eclipse a part of the network. As long as the majority of blockchain nodes forwards transactions as prescribed, it hardly makes a difference whether a miner forwards transactions or not. Exclusive mining, on the other hand, is guaranteed to succeed. The transaction initiator and the miner share new transactions only through the private channel, therefore ensuring that no-one except the colluding miner can confirm them.

Another mechanism which at first glance shares similarities with exclusive mining is selfish mining, which was first described in Eyal and Sirer [3]. In selfish mining, a miner does not immediately share successfully mined blocks with the network but secretly generates a competing chain. Once the competing chain is long enough, it is revealed to the network and has a chance of becoming the new main chain, effectively putting the mining effort of competing miners to waste. Thus, both exclusive mining and selfish mining rely on a miner holding

back information. The difference is that in the case of exclusive mining, a transaction is held back, not a block. Exclusive mining does not aim to fork the blockchain. The intention and the results are therefore very different. Unlike selfish mining, exclusive mining is not an attack on the network; it is merely an unintended way of confirming new transactions.

In Chapter 2, we explain why transaction initiators and miners would employ exclusive mining. We describe how miners can use it to offer "confirmation as a service" or offer users protection from frontrunning, but also how criminal entities might utilize it in money laundering and tax evasion schemes. In this way, we illustrate that exclusive mining does have useful and potentially desirable applications, but also characteristics that can make it highly problematic. In Chapter 3, we describe how other members of the network can detect exclusive mining activity. Our results contribute to the literature on the mining of blockchain transactions and on the incentives and the behaviour of blockchain users [4–7].

2. Applications of exclusive mining

In the following, we discuss three potential applications of exclusive mining. First, transaction processing agreements between miners and entities regularly generating transactions, such as crypto-currency exchanges ("confirmation as a service"). Second, bypassing the mempool of unconfirmed transactions to hide activity from frontrunning bots. Third, money laundering or tax evasion by means of transaction fees in exclusively mined transactions.

2.1 Confirmation as a service

On popular blockchains like Bitcoin or Ethereum, transaction fees have been highly volatile, creating significant cost uncertainty especially for "power users" like cryptocurrency exchanges or services which regularly write information to a blockchain, e.g. supply chain tracking services. Significant unexpected changes in transaction fees on the blockchain could endanger the profitability or even the viability of these users. This danger is far from hypothetical, as e.g. the "CryptoKitties incident" shows, in which the popularity of a game on the Ethereum blockchain led to a significant increase in transaction fees [8]. Against this background, an exclusive mining agreement can provide a safety mechanism to ensure that critical processes are shielded from extreme situations.

When exclusive mining is employed as a safety mechanism against volatile transaction fees, we refer to it as "confirmation as a service." The colluding miner promises to confirm all transactions of the transaction initiator as quickly as possible. For this, he receives an off-chain fee. The initiator can be sure that his transactions are confirmed within a certain time (depending on the agreement and the miner's hashrate). In effect, exclusive mining works as a hedge for both the initiator and the miner: The initiator has his transactions confirmed at a pre-agreed cost; the miner secures a predictable source of income. Thus, both parties reduce their exposure to the volatility of blockchain transaction fees.

Until now, fee volatility has been less impactful for miners than for transaction initiators, as the majority of miner income has come from fixed block rewards. But most blockchains are reducing block rewards over time (Bitcoin's block reward, for example, is halved every four years), such that miners will increasingly need to secure a reliable stream of well-paying transactions to remain profitable. Indeed, researchers have argued that declining block rewards might drive blockchain miners towards protocol violations in search of profit [9].

In principle, different arrangements of confirmation as a service are conceivable, depending on the agreement reached by the parties. In the case of fully exclusive mining, the transaction initiator sends transactions exclusively to the colluding miner; neither the exchange nor the miner propagates them through the blockchain's peer-to-peer network before they are confirmed. It is also conceivable that an initiator enters an exclusive mining agreement with a miner but also propagates transactions to the rest of the network. This increases the chances of having the transactions confirmed quickly. The agreement with the miner acts as a safeguard – if the transaction is not confirmed by non-colluding miners, the colluding miner will do so eventually.

2.2 Anti-frontrunning strategies

While an unconfirmed transaction is propagated through the network, more and more blockchain nodes become aware of it. The fact that other users learn about a transaction before it is confirmed and thus executed can have undesirable or even catastrophic consequences for the transaction initiator. Users who closely monitor their mempool of unconfirmed transactions can exploit the information leaked by these transactions and attempt to profit from frontrunning.

Frontrunning involves replicating or countering an unconfirmed transaction with another transaction and ensuring that the latter is executed first, usually by offering a higher transaction fee to miners. While frontrunning of large transfers on Bitcoin and other single-purpose blockchains is possible under certain circumstances, the issue is particularly problematic for multi-purpose ("Turing-complete") blockchains. In a ground-breaking study, Daian et al. [10] observed a large number of highly profitable arbitrage bots on the Ethereum blockchain. These bots engaged in frontrunning by jumping the queue in decentralized exchanges (DEXes) or exploiting erroneous transfers (e.g. transfers with typos or misplaced decimal points). Robinson and Konstantopoulos [11] provide an especially vivid example in which an attempt to "rescue" misplaced funds is spotted and pre-empted by a predatory bot.

Through exclusive mining, a transaction initiator can hide transactions from the network until they are

confirmed and thus effectively prevent frontrunning – unless it is conducted by the colluding miner. Miners could therefore offer exclusive mining to transaction initiators who want to avoid frontrunning. Miners could do this for profit or to gain reputation within the blockchain community by acting as "white-hat hackers."

Is exclusive mining also attractive to the frontrunners themselves? Possibly, because collusion with a miner (or being a miner oneself) gives power over the selection and ordering of transactions in a block. On proof-of-work blockchains, however, it is uncertain which miner will mine the next block. This might make exclusive mining too unpredictable for frontrunners.

Frontrunning attacks on blockchains do not only relate to decentralized exchanges but also to double-spending attacks, decentralized applications (dApps), initial coin offerings (ICOs), decentralized auctions, blockchain name services and other on-chain activity [12,13]. While modifications of blockchain protocols with regard to confidentiality or transaction ordering may prevent frontrunning in the future, exclusive mining is a safety mechanism that is applicable right now.

2.3 Money laundering and tax evasion

On June 10, 2020, an Ethereum transaction sent 0.55 Ether (ETH), worth around $136 at the time, for a record-breaking transaction fee of 10,669 ETH, worth around $2.6 million.[3] One day later, 350 ETH were sent from the same address, again for a transaction fee of 10,669 ETH.[4] The intention behind these transactions remained unclear. The extreme fees might have resulted from a typo or a software bug. This had happened before. In that case, the mining pool which confirmed the transaction, SparkPool, agreed to reimburse half of the fee [14].

However, the high fees may also have been deliberate, which could have various reasons. One reason could be that access to the address had been compromised but transfers could only be made to certain whitelisted addresses. In this scenario, the high transaction fees would be a kind of blackmail. The hijacker demands ransom money and lends weight to his demand by "burning" funds through transaction fees [15].

Another reason might be that these and similar transactions were part of a money laundering or tax evasion scheme. Exclusive mining allows miners to retain transactions and integrate them exclusively into their own blocks. In this way, a miner can ensure that a transaction with very high transaction costs will never be confirmed by other miners. Since transaction costs represent regular income for miners, significantly increased transaction costs could be used to launder money by colluding with a miner.

Money laundering in the context of cryptocurrency markets has been assessed by a variety of studies. Yet, none of the studies we identified explicitly describe or analyse the mechanism of money laundering via exclusive mining [16–18].

Figure 2 shows a schematic overview of the money laundering process, where an initiator is transferring funds to a colluding miner via exclusive mining. We show two different entities – a transaction initiator and

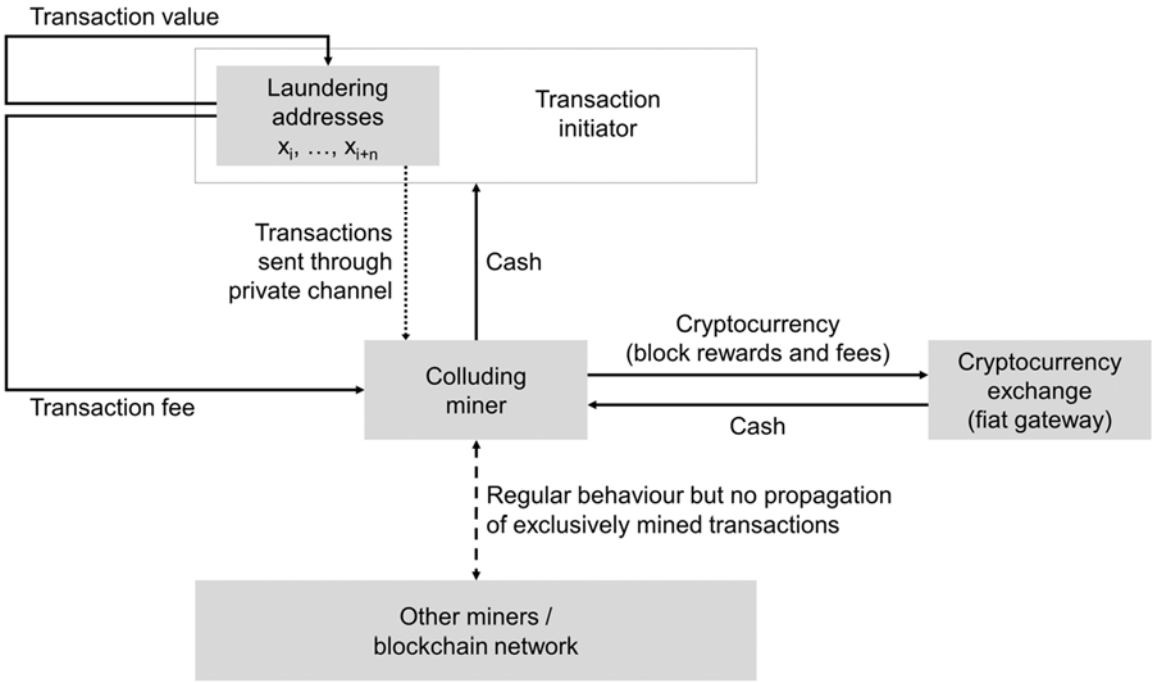


Figure 2: Schematic model of money laundering through transaction fees in exclusive mining.

[3] https://etherscan.io/tx/0xca8f8c315c8b6c48cee0675677b786d1babe726773829a588efa500b71cbdb65.

[4] https://etherscan.io/tx/0xc215b9356db58ce05412439f49a842f8a3abe6c1792ff8f2c3ee425c3501023c.

a miner that come to terms about laundering funds. Both entities could however be controlled by the same actor.

During the process of laundering money through exclusive mining, the initiator always retains control of all transferred funds since all sender and receiver addresses are under his control. Transaction costs are deducted from the initiator's blockchain asset balance and redeemed by the colluding miner. Thus, this mechanism works for all blockchains where transaction fees are directly transferred to the miner of the transaction. Depending on the size of the respective fee per transaction and the quantity of funds to be laundered, all of the initiator's funds can be transmitted to the miner via transaction fees. The latter in turn can declare these fees – together with the "clean" fees earned as a miner – as regular income and exchange them to fiat currency on cryptocurrency exchanges. In the case of tax evasion, the initiator deducts the transaction fees as costs whereas the miner declares them as income. In the case of money laundering, the miner transfers the laundered funds back to the initiator as fiat currency.

On Bitcoin, some mining pools distribute earned transaction fees to their members while others do not [19]. Arguably, the risk of systematic money laundering is much lower in pools which distribute fees. Indeed, it might be easier for the initiator to run its own mining pool. This would eliminate the complexity of transferring fiat money and reduce the costs and risks of colluding with a third-party miner.

If money laundering processes are to be hidden from the rest of the network, the colluding miner must take care not to be seen as untrustworthy by the blockchain network. If it became obvious that the miner engaged in illegal activity, cryptocurrency exchanges could block his access to fiat currency. We describe two obfuscation mechanisms which the initiator and the miner could use to conceal their activity: initiate a small number of high-fee transfers and send cheap quality signals, or initiate a large number of average-fee transfers.

In the first scenario, a small number of transactions with very high transaction costs is initiated. To hide his involvement, the colluding miner can then announce that he would like to reverse the transactions or reimburse some of the fees. From the perspective of signalling theory [20], this acts as a positive quality signal to the market – even though it is a morally hazardous or cheap (to fake) signal [21]. The transaction initiator, of course, does not step forward. The colluding miner then declares the fees as regular mining income and completes the laundering process. This scenario is likely to attract attention from the community, miners, researchers and law enforcement, thus it cannot be repeated indefinitely.

In the second scenario, many transactions with average or slightly higher fees are created. This makes it easier to hide the exclusive mining activity, making the approach potentially viable over a long time. However, various metrics must be considered to ensure that such a system does not attract attention. Since blockchains are transparent, every network participant can see the total and the average transaction fee per mined block. Therefore, the transaction initiator and the miner must make an effort to obfuscate their activity in the best way possible. In the next chapter we describe how such obfuscation techniques can be countered to nonetheless uncover exclusive mining activity.

3. Detection of exclusive mining

Exclusive mining is not easy to detect. Private channels between nodes are easily kept secret, and there is no public record of when or how a node became aware of transactions. Thus, it is rarely possible to obtain definite proof of exclusive mining activity from public information alone.

In the absence of proof, one must resort to evidence. Exclusive mining activity creates characteristic patterns which can be observed by other blockchain nodes. In view of the role exclusive mining may play in tax evasion or money laundering schemes, it is important to be aware of these patterns and develop methods to detect them. In this section, we describe how any blockchain node can utilize the information it receives from peers to monitor the network for exclusive mining activity.

We introduce some clarifying notation. Let T denote the set of all transactions on the blockchain which were confirmed during a given time period. Any individual transaction will be denoted by $\tau \in T$. In principle, the aim of our analysis is to determine the set of all transactions which were exclusively mined:

$$T_{exclusive} := \{\tau \mid \tau \; was \; exclusively \; mined\}.$$

As argued above, $T_{exclusive}$ cannot be determined from publicly available information alone. Finding every exclusively mined transaction would require the cooperation of all miners or complete knowledge of the inner workings of all mining nodes. Neither seems realistic. Nonetheless, every blockchain node is able to determine a set of "suspicious" transactions; this set can then be perused for evidence of exclusive mining.

Let n denote "our" blockchain node. As part of the peer-to-peer network, n is made aware of new (unconfirmed) transactions and new blocks by its peers. We assume that the node is capable of timestamping incoming information, i.e. that it has a local clock and knows when it first became aware of any given transaction or block. Notice however that this clock need not be synchronized with the clocks of other nodes.

For a transaction τ, let $t_{received}(\tau)$ denote the time when n first became aware of the transaction, and let $t_{confirmed}(\tau)$ denote the time when n first became

aware of the block containing the transaction.[5] Since becoming aware of a block also means becoming aware of the transactions it contains, it always holds that $t_{received}(\tau) \leq t_{confirmed}(\tau)$.

The defining characteristic of exclusively mined transactions is that uninvolved nodes only become aware of them once they have been confirmed: If τ has been exclusively mined, then[6]

$$t_{received}(\tau) = t_{confirmed}(\tau). \qquad (*)$$

We refer to transactions which satisfy condition (*) as late transactions. Define the set of all late transactions:

$$T_{late} := \{\tau |\ t_{received}(\tau) \ = \ t_{confirmed}(\tau)\}.$$

Every exclusively mined transaction is a late transaction. The converse statement does not hold. Indeed, even in the absence of exclusive mining, it is not unusual that a node only becomes aware of a transaction through the block which confirms it. Figure 1 illustrates this: Even in the case of regular mining, the bottom right node remains unaware of the transaction until it becomes aware of the block in which it is confirmed. In other words, condition (*) is necessary, but not sufficient, for exclusively mined transactions; or equivalently:

$$T_{exclusive} \subseteq T_{late}.$$

Be aware that $T_{exclusive}$ is "objective" while T_{late} is "subjective." A transaction τ was either mined exclusively or not; given full information, different nodes would not disagree over this fact. The timestamps $t_{received}(\tau)$ and $t_{exclusive}(\tau)$, on the other hand, are different for every node in the network.[7] Indeed, one could denote T_{late} as T_{late}^{n} to clarify that the set pertains to node n only. We omit the n only to avoid visual clutter.

The advantage of T_{late} over $T_{exclusive}$ is that it can be determined using only the information available to node n. Thus, T_{late} can be considered a noisy but observable approximation of the desired but unobservable set $T_{exclusive}$. Because T_{late} also contains transactions which were late by chance and not because of exclusive mining, one must resort to statistical methods to detect unusual patterns within the set.

Let M be the set of miners. For $m \in M$, define the set of transactions contained in blocks which were mined by miner m:

$$T^m := \{\tau \mid \tau \ was \ confirmed \ by \ m\}.$$

Furthermore, define the set of late transactions mined by m:

$$T_{late}^m := \{\tau \mid \tau \ was \ confirmed \ by \ m \ and \ t_{received}(\tau) \ = \ t_{confirmed}(\tau)\}.$$

Notice that $T_{late}^m = T^m \cap T_{late}$.

If m engages in exclusive mining, he will have a larger share of late transactions than the average miner. Therefore, a first metric which correlates with exclusive mining activity is the share of late transactions for a given miner:

$$\alpha^m := \frac{|T_{late}^m|}{|T^m|} = \frac{|T^m \cap T_{late}|}{|T^m|}.$$

A second metric derives from the fees associated with late transactions. When exclusive mining is employed as part of tax evasion or money laundering schemes, the intention is to transfer a significant amount of value to the miner through transaction fees. As described in Chapter 2, this may be accomplished through a small number of high-fee transactions or a high number of average-fee transactions. Regardless of the number of transactions, however, the exclusive mining activity will show up as a large amount of transaction fees earned through late transactions. For a transaction τ, let $f(\tau)$ denote its fee. Define the total fees earned by miner m through late transactions:

$$\beta^m := \sum_{\tau \in T_{late}^m} f(\tau).$$

The variable β^m is the sum of fees earned by m through exclusive mining plus the fees earned by m through other late transactions. It is therefore an upper bound on the amount transferred to m as part of tax evasion or money laundering schemes. However, one must keep in mind that a high value of β^m does not in any way prove that m engages in money laundering or tax evasion. It may, however, hint towards unusual activity.

Two additional metrics can be used to study the fee structure of late transactions. One is the share of fees earned by miner m through late transactions:

$$\gamma^m := \frac{\sum_{\tau \in T_{late}^m} f(\tau)}{\sum_{\tau \in T^m} f(\tau)} = \frac{\beta^m}{\sum_{\tau \in T^m} f(\tau)}.$$

While γ^m does not provide an upper bound on suspicious fees earned by miner m, it has the advantage of being independent of the miner's total hashrate. In particular, under the assumption that there is no exclusive mining, the expected value $E[\gamma^m]$ is the same for small and large miners.

Another metric is the average fee of late transactions relative to the average fee of non-late transactions:

$$\delta^m := \frac{\sum_{\tau \in T_{late}^m} f(\tau)/|T_{late}^m|}{\sum_{\tau \in T^m} f(\tau)/|T^m|}.$$

The metric δ^m can provide valuable insight but should be interpreted with care. Miners engaging in

[5] For simplicity, we ignore blockchain forks and thus the possibility that a transaction is confirmed in more than one block. When analyzing historical blockchain data, this is easily accomplished by considering only the main chain.

[6] A miner may try to obfuscate exclusive mining activity by propagating the transaction after finding, but before propagating the new block. In this case, the equality would not hold. However, by doing so he risks that another miner finds a new block first, causing him to lose all gains associated with his block.

[7] Bitcoin's blocks contain a timestamp. One might therefore argue that there is an objective time when transactions were confirmed. The block timestamp, however, is set by the miner of the block. In the presence of clock drift, it cannot be compared to another node's timestamps. Thus, a comparison between the time when a transaction was received and the time it was confirmed is only sensible when both times have been determined by the same node.

confirmation as a service are likely to earn lower fees for their exclusively mined transactions (because they receive an additional off-chain fee), which would result in δ^m being significantly smaller than 1. Miners engaging in money laundering or tax evasion through exclusive mining are likely to earn higher fees for their exclusively mined transactions (because these transactions need to transfer significant value to the miner), which would result in δ^m being significantly larger than 1. For exclusive mining employed in anti-frontrunning strategies, the effect on δ^m is unclear: The miner may demand higher fees as remuneration, or lower fees plus an off-chain fee. In any case, it is easy to "whitewash" δ^m by offsetting high-fee transactions with a number of low-fee transactions or vice versa. The metrics α^m, β^m and γ^m appear less susceptible to such attempts of hiding exclusive mining activity.

Additional insight may be gained from studying the transactions in T_{late} in more detail. Transactions which are part of tax evasion or money laundering schemes are self-transfers, i.e. the transaction initiator controls all input and output addresses. While self-transfers can be obfuscated by using a large number of addresses and emulating realistic transaction behaviour, sophisticated pattern recognition may be able to detect such self-transfers and thus uncover entities which potentially engage in exclusive mining.

The metrics α^m, β^m and γ^m are positively correlated with exclusive mining activity. Large values suggest irregular mining behaviour and can thus be interpreted as possible evidence of exclusive mining of miner m. But how large is large, exactly? To obtain quantitative results, it is helpful to view the metrics as random variables. Under the null hypothesis that miner m does not engage in exclusive mining, and under appropriate assumptions on the propagation of information through the peer-to-peer network, it should be possible to derive the stochastic distributions of these random variables in dependence of m's share of total hashing power and the arrival rate of new transactions and blocks. Once the distributions are known, one can conduct statistical hypothesis testing for exclusive mining. We leave this to future research.

4. Conclusion

We have provided an overview of the concept of exclusive mining. Transactions are sent via a private channel to a colluding miner who confirms them in new blocks. The unconfirmed transactions are not propagated through the blockchain's peer-to-peer network, neither by the initiator nor the miner. Exclusive mining can be employed for various reasons, ranging from innocuous hedging of transaction fee volatility to money laundering and tax evasion.

Considering that it is difficult to identify exclusive mining and therefore the motivation behind it, we have outlined ways for node operators to identify evidence of exclusive mining and suggested a direction for future research into statistical testing for exclusive mining.

How realistic is it that money is being laundered through exclusive mining? As mentioned above, transactions with extremely high fees have attracted considerable attention from the media and the blockchain community. Some mining pools have offered to reimburse excessive or accidental fees [22,23]. This speaks in favour of self-regulation of the market, although one should not automatically assume that these reimbursements actually occur.

Of course, honest miners have an interest in the long-term success of their blockchain ecosystem. This may contribute to a miner's decision to reimburse fees. But miners are not necessarily honest. Arguably, comprehensive regulation of blockchain mining is the only measure that could fully prevent money laundering and tax evasion through exclusive mining. Considering that cryptocurrencies are decentralized networks whose miners are located all over the world; however, such regulation seems out of reach. It should also be borne in mind that over-regulation hampers innovation. Any regulation of mining should be designed with appropriate caution.

What if one abolished transaction fees altogether? In its current form, Bitcoin will eventually evolve into a fee market without any other rewards for miners. By contrast, parts of the Ethereum community argue that the current first-auction fee market is inefficient. The Ethereum Improvement Proposal (EIP) 1559 proposes the introduction of a base fee which adjusts based on network demand. When the transaction is confirmed, the base fee is burned, i.e. destroyed. The proponents of EIP 1559 argue that this would increase price efficiency and avoid unnecessarily high transaction fees [24]. While the proposal still allows for small tips for miners, the above-described ability to launder money via large transaction fees would become much more difficult. It is unrealistic, however, that all existing blockchains will adapt their fee mechanism. The associated risk of illegal activity based on exclusive mining must be assessed for each blockchain infrastructure individually.

It should be noted that exclusive mining is possible on a large number of blockchains. For instance, it should be possible to engage in exclusive mining on a proof-of-work blockchain as well as on a proof-of-stake blockchain. The incentive structure, however, might look entirely different. On proof-of-stake blockchains, the miner of the next block is typically known in advance, which could make confirmation as a service as well as anti-frontrunning strategies much more attractive [10]. We hope that future research builds on our study to determine to what extent different blockchains display evidence of exclusive mining activity and if associated risks differ based on underlying technologies and mechanisms.

It can also be useful to examine market characteristics such as trading volume or liquidity. Money laundering only works if the funds can reliably be redeemed for fiat currency. While Bitcoin and Ethereum are currently most suitable in this regard,

they are also the two blockchain where it is most difficult to mine blocks at regular intervals. In addition, both Bitcoin and Ethereum are observed closely by blockchain enthusiasts, researchers and the media. Criminals may see smaller blockchain networks as more suitable vehicles for money laundering or tax evasion via exclusive mining.

In summary, exclusive mining can be both a blessing and a curse for blockchains. It is in the best interest of blockchain communities and concerned authorities to develop appropriate monitoring tools which can detect exclusive mining activity, at least when it is employed towards criminal ends. We hope that our work serves as the foundation for further research and a heightened awareness of exclusive mining, its potential and its perils.

References

[1] M. Babaioff, S. Valley, S. Dobzinski, S. Oren, On Bitcoin and Red Balloons, in: Proc. 13th ACM Conf. Electron. Commer. - EC '12, 2012: pp. 56–73. https://doi.org/10.1145/2229012.2229022.

[2] O. Ersoy, Z. Ren, Z. Erkin, R.L. Lagendijk, Transaction Propagation on Permissionless Blockchains: Incentive and Routing Mechanisms, in: 2018 Crypto Val. Conf. Blockchain Technol., 2018. https://doi.org/10.1109/CVCBT.2018.00008.

[3] I. Eyal, E.G. Sirer, Majority Is Not Enough: Bitcoin mining is vulnerable, in: Eighteenth Int. Conf. Financ. Cryptogr. Data Secur., 2014.

[4] E. Strehle, F. Steinmetz, Dominating OP Returns: The Impact of Omni and Veriblock on Bitcoin, 2020.

[5] A. Gervais, H. Ritzdorf, G.O. Karame, S. Čapkun, Tampering with the delivery of blocks and transactions in Bitcoin, in: Proc. ACM Conf. Comput. Commun. Secur., 2015: pp. 692–705. https://doi.org/10.1145/2810103.2813655.

[6] J.A. Kroll, I.C. Davey, E.W. Felten, The Economics of Bitcoin Mining or, Bitcoin in the Presence of Adversaries, in: Proc. WEIS, 2013: pp. 1–21.

[7] O. Schrijvers, J. Bonneau, D. Boneh, T. Roughgarden, Incentive Compatibility of Bitcoin Mining Pool Reward Functions, in: Int. Conf. Financ. Cryptogr. Data Secur., Springer Berlin Heidelberg, 2016: pp. 477–498.

[8] BBC, CryptoKitties craze slows down transactions on Ethereum, (2017). https://www.bbc.com/news/technology-42237162 (accessed August 12, 2020).

[9] M. Carlsten, H. Kalodner, S.M. Weinberg, On the instability of bitcoin without the block reward, in: Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur., 2016: pp. 154–167. https://doi.org/10.1145/2976749.2978408.

[10] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, A. Juels, Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability, in: 2020 IEEE Symp. Secur. Priv., 2020: pp. 910–927. https://doi.org/10.1109/SP40000.2020.00040.

[11] D. Robinson, G. Konstantopulos, Ethereum is a Dark Forest, (2020). https://medium.com/@danrobinson/ethereum-is-a-dark-forest-ecc5f0505dff (accessed August 20, 2020).

[12] S. Eskandari, S. Moosavi, J.C. B, SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain, in: Int. Conf. Financ. Cryptogr. Data Secur., Springer International Publishing, 2019: pp. 170–189. https://doi.org/10.1007/978-3-030-43725-1.

[13] G.O. Karame, E. Androulaki, Double-Spending Fast Payments in Bitcoin Categories and Subject Descriptors, in: Proc. 2012 ACM Conf. Comput. Commun. Secur., 2012: pp. 906–917. https://doi.org/10.1145/2382196.2382292.

[14] Coindesk, Ethereum mining pool Sparkpool has located and verified the accidental sender of an unusually high miners' fee and agreed to split the amount., (2019). https://www.coindesk.com/sparkpool-splits-2100-ether-mining-fee-with-accidental-sender (accessed June 13, 2020).

[15] Decrypt, Hackers blackmail exchange with $5 million of Ethereum fees - report, (2020). https://decrypt.co/32145/hackers-blackmail-exchange-with-5-million-of-ethereum-fees-report (accessed June 13, 2020).

[16] L. Ante, Cryptocurrency, Blockchain and Crime, in: K. McCarthy (Ed.), Money Laund. Mark. Regul. Crim. Econ., Agenda Publishing, 2018: pp. 171–198. https://doi.org/10.2307/j.ctv5cg8z1.10.

[17] M. Möser, R. Böhme, D. Breuker, An Inquiry into Money Laundering Tools, in: 2013 APWG ECrime Res. Summit, 2013: pp. 1–14. https://doi.org/10.1109/eCRS.2013.6805780.

[18] R. van Wegberg, J.J. Oerlemans, O. van Deventer, Bitcoin money laundering: mixed results?: An explorative study on money laundering of cybercrime proceeds using bitcoin, J. Financ. Crime. 25 (2018) 419–435. https://doi.org/10.1108/JFC-11-2016-0067.

[19] Bitcoin Wiki, Comparison of mining pools, (2020). https://en.bitcoin.it/wiki/Comparison_of_mining_pools (accessed June 15, 2020).

[20] M. Spence, Job Market Signaling, Q. J. Econ. 87 (1973) 355–374. https://doi.org/10.1055/s-2004-820924.

[21] L. Ante, I. Fiedler, Cheap Signals in Security Token Offerings (STOs), (2019). https://doi.org/10.2139/ssrn.3356303.

[22] Bitcoin.com, Mining Pool BTC.com Finds Accidental 80 BTC Fee – Offers a Refund, (2017). https://news.bitcoin.com/mining-pool-btc-com-80-btc-fee-refund (accessed June 15, 2020).

[23] Bitcoin.com, Bitcoin Miner Repays Customer Who Accidentally Paid 2.5 Bitcoins Transaction Fee, (2017). https://news.bitcoin.com/bitcoin-

miner-repays-customer-who-accidentally-paid-2-5-bitcoins-transaction-fee/ (accessed June 15, 2020).
[24] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norder, Ethereum Improvement Proposal 1559 (EIP-1559), (2020). https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md (accessed June 16, 2020).
[25] A.M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, 2014.

IMMANENTES SYSTEMVERTRAUEN DER BLOCKCHAIN FÜR INTERNET OF THINGS – ERGEBNISSE EINER SYSTEMATISCHEN ÜBERPRÜFUNG

Stefan Tönnissen, Frank Teuteberg
Universität Osnabrück, Katharinenstr. 1, D-49069 Osnabrück

Mehr als 50 Milliarden physische Objekte sollen bis 2020 mit dem Internet verbunden sein. Diese reichen von kleinen und rechenarmen RFID-Systemen bis zu komplexen Geräten wie Smartphones, intelligenten Geräten und Fahrzeugen. Für dieses Internet of Things (IoT) bedarf es eines Systemvertrauens, da die Nutzung intelligenter Dienste über das Internet ohne menschliches Eingreifen geschieht. Heute vorhandene zentrale Vertrauensinstanzen für IoT verlieren aufgrund von Hacker- und Cyberangriffen ihr Vertrauen. Mit der Blockchain existiert eine Vertrauensarchitektur, die es dem Menschen erlaubt, einem System und seinen Komponenten, und nicht einer zentralen Instanz zu vertrauen. Kann die Symbiose von Blockchain und IoT Vertrauen generieren? Dieser Artikel präsentiert eine systematische Literaturübersicht zum Konzept Vertrauen im Kontext der Blockchain-Technologie für das IoT und deren Geschäftsmodelle. Das Ziel dieses Beitrags ist die Darstellung der aktuellen Entwicklungen im Zusammenspiel von Blockchain und IoT, um diese als Blaupause für die Schaffung von Vertrauen in weiteren Anwendungsfeldern nutzen zu können.

More than 50 billion physical devices will be connected to the Internet by 2020. These range from small and rake-poor devices such as RFIDs to complex devices such as smartphones, smart devices and vehicles. This Internet of Things (IoT) requires system trust, as the use of intelligent services over the Internet is done without human intervention. Today's central IoT line of businesses (LOBs) lose their confidence due to hacker and cyber-attacks. Blockchain is a trust architecture that allows people to trust a system and its components, not a central entity. Can the symbiosis of blockchain and IoT generate trust? This article presents a systematic literature review of trust in the context of blockchain technology for the IoT and its business models. The aim of this paper is to present the current developments in the interplay between Blockchain and IoT in order to use it as a blueprint for the creation of trust in other fields of application.

1. Einleitung

„Vertrauen ist der Schlüssel für die digitale Wirtschaft“ [43]. Der Economist beschrieb bereits 2015 die Blockchain als eine Technologie zur Schaffung von Vertrauen für Menschen, die kein besonderes oder besonders hohes Vertrauen ineinander haben, jedoch miteinander arbeiten müssen ohne eine neutrale zentrale Instanz zu nutzen. Walterbusch et al. (2014) heben hervor, „…trust is a fundamental cornerstone in the business context“. Die Sharing Economy erfährt eine Verlagerung von einer Infrastruktur, die Menschen voreinander schützt, zu einer Infrastruktur, die Vertrauen zwischen Menschen schafft [16]. In einer aktuellen Studie von Bitkom bestätigen Blockchain-Experten, dass die Blockchain Geschäftsbeziehungen zwischen Unternehmen schaffen kann, die bisher aufgrund fehlenden Vertrauens nicht zustande gekommen wären [4]. In diesem nicht vertrauenswürdigen Umfeld schafft die Blockchain die Voraussetzungen zur Speicherung von Informationen [9]. Die Blockchain als Distributed Ledger Technologie schafft somit Vertrauen in die Funktionalität und Manipulationsfreiheit und gibt dem Nutzer das Vertrauen, dass Inhalte nicht geändert werden. Sie erreicht dies durch eine dezentrale, redundante und manipulationssichere Speicherung von Daten [29]. Jedoch ist Vertrauen zunächst an Personen und Interaktionen zwischen Personen gebunden, daher sprechen wir hinsichtlich des Vertrauens in die Blockchain von Systemvertrauen und damit in die Funktionsfähigkeit von Systemen [5]. Dieses Systemvertrauen lässt sich nach Heidt et al. (2019) in die Dimensionen Vertrauen in Code, Vertrauen in Daten, Vertrauen in die Vision eines Projektes sowie systemisches Vertrauen einteilen.

Für bestehende und vertrauensvolle Beziehungen zwischen Geschäftspartnern ist demnach eine Blockchain unnötig, es sei denn, andere Aspekte wie die Schaffung von Transparenz treten in den Vordergrund [52]. „Vertrauen ist gut, Blockchain ist besser.“ [44] Die Blockchain ist jedoch eine komplexe Technologie, die laut einer Studie von pwc aus dem Jahre 2016 nur von 19% der Menschen in Deutschland verstanden wird [39]. Wie kann unter diesen Voraussetzungen die Blockchain das Systemvertrauen herstellen beziehungsweise in welchem Kontext wird das Systemvertrauen hergestellt? Denn das ebenbürtige Äquivalent zu Vertrauen ist Misstrauen, und fordert daher von einem Individuum eine permanente Entscheidung zwischen diesen beiden Möglichkeiten. Dem Misstrauen immanent ist eine geringere Abhängigkeit von Informationen [23]. Neben technischen Aspekten und der Frage der Finanzierung sollten sich erfolgreiche Geschäftsmodelle auch mit dem Aspekt des Vertrauens beschäftigen, um nicht zu scheitern [16]. Sollte das Vertrauen der Öffentlichkeit in ein bestimmtes Unternehmen schwinden, so kann dies „…to an immediate and short-term loss in customers or a dip in the share price“ führen [42]. In einer Untersuchung von Rodig (2017) über IoT-basierte Geschäftsmodelle zeigen 51% der untersuchten IoT-Projekte, dass die Schaffung von zusätzlichen Einnahmen durch neue Services oder Produkte für bereits adressierte Zielgruppen im Vordergrund steht.

Unsere vorherigen Überlegungen führen zu folgen-

den Forschungsfragen (FF), die wir im Rahmen dieses Beitrags anhand eines systematischen Literaturreviews beantworten:

- FF1: Mit welchen technischen oder konzeptionellen Lösungen wird das Systemvertrauen in die Blockchain hergestellt?
- FF2: Welche Dimensionen von Vertrauen werden dabei angesprochen?
- FF3: Welche Muster der Vertrauensgenerierung für IoT-basierte Geschäftsmodelle sind bisher entwickelt worden?

Unser Beitrag ist wie folgt aufgebaut: Zunächst führen wir in die Grundlagen über Vertrauen, die Blockchain-Technologie und das IoT ein, um im nächsten Kapitel die methodische Vorgehensweise unserer Arbeit zu erläutern. Daran schließt sich die Suche nach Literatur und deren Auswertung an, um dann im nächsten Kapitel die Ergebnisse zu präsentieren und die Schlussfolgerungen hinsichtlich unserer Forschungsfragen zu präsentieren. Zum Ende werden Limitationen und ein Ausblick dargestellt.

2. Grundlagen

2.1 Vertrauen

Vertrauen wird als Enabler für soziale Interaktionen gesehen und hat ihren Ursprung der Forschung außerhalb des Bereichs der Informationssysteme. Seit vielen Jahren nimmt jedoch die Bedeutung des Vertrauens durch den Fortschritt der Technologien, wie zum Beispiel dem elektronischen Geschäftsverkehr, zu. Denn durch virtuelle Teams, Online-Märkten und Plattform getriebenen Geschäftsmodellen nehmen die Interaktionen und der Handel zwischen Fremden zu [50]. Vertrauen ist von besonderer Bedeutung, wenn Unternehmen ihre Prozesse oder Daten in elektronische Märkte oder an Cloud-Computing Anwendungen übergeben [53]. Vertrauen bezeichnet die subjektive Überzeugung von der Richtigkeit, von der Wahrheit von Handlungen, von Einsichten und Aussagen. Es tritt in unsicheren Situationen oder bei Handlungen mit einem risikohaften Ausgang auf, bedingt jedoch immer eine Grundlage [28]. Vertrauen braucht zum einen Grundlagen und muss zum anderen stets über gute Gründe hinausgehen und Ungewissheit aufheben [35]. Es ist darüber hinaus ebenfalls die Erwartung, nicht durch das Handeln anderer benachteiligt zu werden und schafft somit die Grundlage jeder Kooperation [49]. Neben dem interpersonalen Vertrauen zwischen Personen ist das Systemvertrauen als Vertrauen in die Systemzuverlässigkeit zu berücksichtigen. Hierbei wird dem Funktionieren des Systems ein Vertrauen geschenkt, und dies nicht durch ein einzelnes Individuum, sondern durch deren Masse. Des Weiteren ist Kontinuität im Vertrauen an die Funktionsfähigkeit des Systems notwendig, ohne dass das einzelne Individuum verstehen muss, wie das System funktioniert. Dass es funktioniert, reicht für das Vertrauen aus [23].

Im Kontext der Blockchain haben Heidt et al. (2019) folgende Vertrauensdimensionen unterschieden, die wir in unsere Arbeit übernehmen:

• **Vertrauen in Code**: Benutzer vertrauen darauf, dass das System keine schwerwiegenden Programmierfehler enthält.

• **Vertrauen in Daten**: Benutzer vertrauen darauf, dass die in das System eingegebenen Daten korrekt und überprüfbar sind. Dies ist von größter Bedeutung, da Daten direkt die Grundlage für Entscheidungen bilden.

• **Vertrauen in die Vision**, die das Projekt befeuert: Benutzer vertrauen darauf, dass das System genügend Schwung erhält, um ein digitales Ökosystem zu schaffen, das die fragliche Plattform unterstützen kann.

• **Systemisches Vertrauen**: Um auf ein bestimmtes System oder eine bestimmte Plattform vertrauen zu können, müssen mehrere Elemente zusammenwirken. Ob Benutzer einem System oder einer Plattform vertrauen oder ihnen misstrauen, hängt vom erfolgreichen Zusammenspiel der oben genannten Elemente ab [22].

• **Authentifizierung**: Sie sorgt dafür, dass die Identität eines Benutzers gegenüber einem System nachgewiesen und verifiziert werden kann [32].

• **Autorisierung**: Nach der Authentifizierung ist die Identität vom System bestätigt und mit der Autorisierung erfolgt die Zugriffsberechtigung [32].

Vertrauen ist allgemein eine entscheidende Komponente für erfolgreiche Transaktionen, unabhängig davon, ob sie in einem physischen oder virtuellen Raum ausgeführt werden [51].

2.2 Blockchain

Die Blockchain als Distributed Ledger ist in ihrer allgemeinen Form eine verteilte Transaktionsdatenbank in einem Peer-to-Peer-Netzwerk, die Transaktionen als digitale Ereignisse aufzeichnet [2]. Die global verteilten Knoten in dem Netzwerk sind mit einer eigenen Protokollschicht für die Kommunikation zwischen den Knoten verbunden. Die Identifizierung geschieht durch die IP-Adresse des Knoten und über die öffentlichen Schlüssel der Benutzer [14]. Innerhalb der Teilnehmer in dem Peer-to-Peer-Netzwerk der Blockchain sind die gespeicherten Transaktionen unveränderlich und werden über einen Konsensmechanismus konsistent gehalten [37]. Jeder Teilnehmer dieser Blockchain kann alle Transaktionen einsehen sowie die chronologische Kette der Blöcke nachvollziehen [2]. Die Blockchain ist eine Kette von Blöcken, von denen jede eine Reihe von Transaktionen eines bestimmten Zeitraumes enthält. Die Unveränderlichkeit dieser Kette von Blöcken geschieht durch die Sicherung der Transaktionsdaten durch kryptografische Hash-Funktionen. Hierbei werden die Blöcke mit dem vorherigen Block dadurch verbunden, dass der Hash des vorherigen Blocks in den Hash des aktuellen Blocks einfließt. Somit ist es praktisch unmöglich, Transaktionen der Blockchain rückgängig zu machen oder zu manipulieren. Alle Teilnehmer eines Blockchain Peer-to-Peer-Netzwerks verfügen über einen

persönlichen Schlüssel zur Signierung der Transaktionen. Aufgrund des über das Netzwerk verteilten Hauptbuches verfügt jeder Teilnehmer über das Wissen, wie die digitalen Assets unter den Teilnehmern verteilt sind. Dieses Wissen verhindert die doppelte Ausgabe oder Weitergabe von digitalen Assets, die Vermeidung des double-spending. Als Mehrbenutzersystem ist die Blockchain konzipiert worden, um kontinuierliche und nicht zentral durch einen Intermediär gesteuerte Interaktionen zwischen heterogenen Teilnehmergruppen zu ermöglichen. Die Blockchain kann entweder als öffentliche oder private Blockchain verwendet werden. Eine öffentliche Blockchain steht jedermann zur Nutzung frei und erlaubt die Einsicht in alle Daten. Die private Blockchain hingegen ist nur für ausgewählte Benutzer zugänglich und bedingt eine entsprechende Berechtigung für den Zugang.

Die systemimmanente Vertrauensgrundlage von Blockchain (siehe hierzu auch Kapitel 2.1 Vertrauen) setzt sich zusammen aus:

a) **Smart Contracts**. Sie fördern die Effizienz, Sicherheit und Unparteilichkeit bei der Ausführung eines Vertrages auf der Blockchain und schaffen somit das Vertrauen zwischen den Parteien [25].

b) Dem Proof-of-work **Konsensmechanismus**, der die Inhalte der Blockchain vor Manipulationen [1] schützt und Transaktionen ohne Konsens ablehnt [29].

c)Die **Unveränderlichkeit** der Daten und die Nachvollziehbarkeit von Änderungen erhöht die Transparenz und schafft somit Vertrauen [46].

d) Dem **Peer-to-Peer Netzwerk**, in dem jeder Teilnehmer im Mining Netzwerk als vertrauenswürdiger Dritter für Clients fungieren kann [3].

e) Dem **Distributed Ledger** zur Vermeidung eines Single Point of Failure [3].

f) Die Gewährleistung der **Integrität** der in der Blockchain gespeicherten Daten [29].

g) Anhand der **Kryptographie** mit öffentlichen Schlüsseln [29].

Während es verschiedene Varianten und Ausprägungen von Blockchain gibt, konzentrieren wir uns in diesem Beitrag auf öffentliche Blockchain mit den nachfolgenden Eigenschaften in Tabelle 1.

Tabelle 1: Eigenschaften einer öffentlichen Blockchain [37]

Eigenschaft	*Erläuterung*
Open Source	Jeder kann eine Blockchain einrichten.
Dezentralisierung	Es gibt keine zentrale Instanz beziehungsweise Indermediären; die Blockchain ist Teil eines Peer-to-Peer-Netzwerkes.
Konsens	Die Aufnahme einer Transaktion geschieht nur durch einen Konsens. Es gibt nur einen Single Point of Truth.
Manipulations-sicherheit	Neue Einträge in die Blockchain werden nur akzeptiert, wenn sie auf unveränderten Einträgen basieren.
Gültigkeit	Neue Einträge in die Blockchain werden nur akzeptiert, wenn sie einem vordefinierten Protokoll entsprechen.

2.3 Internet of Things

Mit IoT wird die Verknüpfung eines physischen Objekts beziehungsweise Gegenstandes mit einer digitalen Repräsentation verstanden [30], oder auch die Verschmelzung von physischen Produkten und digitalen Services zu hybriden Lösungen [11]. Die Basistechnologie des IoT sind drahtlose Sensornetzwerke [45]. Mit dem IoT wird die Vision verfolgt, das Internet durch die Einbindung physischer Gegenstände in die reale Welt hinein zu verlängern [11]. Das Potential liegt laut Analysten bis 2020 bei mehr als 50 Milliarden Geräten [24]. Mit Hilfe von Minicomputern werden Gegenstände und Orte zu smarten Dingen, die Informationen aus der Umwelt verarbeiten und mit dem Internet kommunizieren [11]. Zahlreiche Gegenstände der täglichen Verwendung werden mit elektronischen Geräten ausgestattet, um sie miteinander und mit dem Internet zu verbinden. Der Datenaustausch erfolgt ohne einen menschlichen Eingriff [20]. Die sich hieraus ergebenden Herausforderungen und Probleme basieren hauptsächlich auf dem allgegenwärtigen Zugang zum Internet, der enormen Menge an verbundenen Geräten sowie der Heterogenität der entsprechenden Komponenten [6]. Aufgrund der Autonomie der Vernetzung als auch des Datenaustausches dieser Geräte liegt eine weitere Herausforderung in der Authentifizierung sowie der Integrität der ausgetauschten Daten [20]. Die heutigen IoT-Systeme sind so konzipiert, dass Sicherheit und Datenschutz einem vertrauenswürdigen Dritten übertragen werden [34]. Der Austausch von Informationen und die Vernetzung der Geräte erfordert in diesem Kontext ein hohes Maß an Vertrauen [26]. Die größte Herausforderung liegt demnach in der Überbrückung der aktuell stark fragmentierten Vertrauensdomänen [38]. Hierbei sind die fünf Wertschöpfungsstufen einer IoT-basierten Anwendung zu beachten, die sich in eine Ebene der physischen Welt als auch eine Ebene der digitalen Welt einteilen lassen (siehe Bild 2). Die physische Welt besteht aus der physischen Ebene auf Ebene 1 und der Sensoren und Aktuatoren auf Ebene 2. Die Verbindung der physischen mit der digitalen Welt des Internets geschieht auf der Ebene 3 mit Konnektoren. Darauf baut die Ebene 4 mit der Analytik der Daten sowie Ebene 5 mit den Webservices oder mobilen Applikationen auf [11].

3. Methodische Vorgehensweise

3.1 Ablauf

Für die Beantwortung unserer Forschungsfrage führten wir zunächst ein Literaturreview mit 369 Treffern durch (siehe Abbildung 1). Diese Treffer wurden dann einer Konformitätsprüfung anhand der Merkmale Kontext, Zweck und praktischer Beitrag unterzogen, um nur die für unsere Forschungsfragen relevanten Beiträge bestimmen zu können. Das Ergebnis sind 79 Treffer, die anhand der Vertrauensdimensionen (siehe Kapitel 2.1) sowie der systemimmanenten Vertrauenseigenschaften der Blockchain (siehe Kapitel 2.2) analysiert wurden. Mit dem Ergebnis können wir die Forschungsfragen FF1 und FF2 beantworten. Für die Beantwortung der Forschungsfrage FF3 führten

wir zunächst auf der Grundlage der zuvor erhobenen Daten eine hierarchische Cluster-Analyse mit der Ward-Methode durch, um anhand der unterschiedlichen Cluster die Muster der Vertrauensgenerierung ableiten zu können.

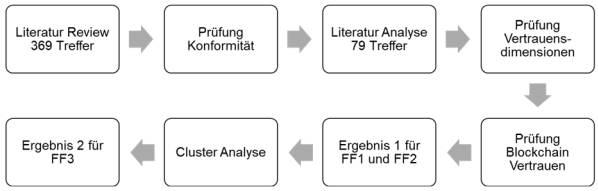


Bild 1: Ablauf der Forschung

3.2 Literaturreview

Das Literaturreview der Beiträge und Artikel wurde unter Verwendung der in Tabelle 2 und Tabelle 3 angegeben Quellen im Juli 2019 durchgeführt.

Die Suche nach relevanten Beiträgen ist in zwei Stufen durchgeführt worden:

1.Stufe: Suche nach Beiträgen zu „Blockchain Trust IoT“, „Blockchain Trust Internet of Things“ sowie „Blockchain Vertrauen IoT“ und „Blockchain Vertrauen Internet of Things“.

2.Stufe: Suche nach Beiträgen zu „Blockchain Trust“ sowie „Blockchain Vertrauen“. Manuelle Sichtung der zusätzlichen Beiträge anhand des Titels und des Abstracts und Entscheidung für eine Aufnahme in unsere Analyse.

Der Grund für diese Vorgehensweise ist, dass wir auch Beiträge wie zum Beispiel „BARS: A Blockchain-Based Anonymous Reputation System for Trust Management in VANETs” in unsere Analyse einbeziehen wollen, obwohl der Begriff IoT oder Internet of Things fehlt. Jedoch sind Vehicular Ad Hoc Network (VANet) ein Teil von Internet of Things und damit relevant für unsere Analysen.

Die Auswertung von Journals führt zu der in Tabelle 2 aufgeführten Ergebnisse.

Tabelle 2: Ergebnisse unserer Literaturanalyse in Journals

Quelle:	*Ergebnis*
IEEE Xplore,	58 Treffer, davon 28 als relevant eingestuft.
Web of Science	37 Treffer, davon 10 als relevant eingestuft.
Google Scholar	26 Treffer, davon 21 als relevant eingestuft.
Springer Link	220 Treffer, davon 29 als relevant eingestuft.
Science Direct	21 Treffer, davon 5 als relevant eingestuft.

Neben den Journalen haben wir drei gemäß Verband der Hochschullehrer für Betriebswirtschaft (VHB) Jourqual3 Rating hoch bewertete und renommierte Konferenzen der Wirtschaftsinformatik und deren Proceedings nach relevanten Beiträgen mit folgendem Ergebnis durchsucht (siehe Tabelle 3).

Tabelle 3: Ergebnisse unserer Literaturanalyse in Konferenzen

Quelle:	*Ergebnis*
ICIS	3 Treffer, davon 0 als relevant eingestuft.
ECIS	4 Treffer, davon 0 als relevant eingestuft.
WI	0 Treffer, davon 0 als relevant eingestuft

Nach der Bereinigung der Ergebnisse anhand von Doppelungen (14 Beiträge) konnten wir 79 Beiträge zur Beantwortung unserer Forschungsfragen selektieren.

Die Konformität der ausgewählten Beiträge wurde anhand der Titel und der Abstracts nach den folgenden Einschlusskriterien untersucht:

- **Kontext**: Die Studien sollten ihre Beiträge im Kontext der Blockchain-Technologie definieren, die in den Anwendungsbereich von IoT zielen.
- **Zweck**: Der Zweck dieser Studien muss sich auf konkrete Lösungsvorschläge zur Generierung von Vertrauen durch die Blockchain-Technologie im Kontext von IoT beziehen.
- **Praktischer Beitrag**: Die Studien sollten mindestens eines der folgenden Elemente von Design Science Research enthalten: praktische Umsetzung, Tests, kritische Analyse, Bewertung oder Diskussion [13].

Die wichtigsten extrahierten Merkmale sind die Eigenschaften des systemimmanenten Vertrauens der Blockchain (siehe Kapitel 2.2) sowie die Dimensionen des Vertrauens (siehe Kapitel 2.1).

3.3 Verwandte Arbeiten

Anhand des zuvor durchgeführten Literaturreviews konnten wir die nachfolgenden thematisch verwandten Arbeiten finden (siehe Tabelle 4).

Tabelle 4: Verwandte Arbeiten

Titel des Beitrags	*Autoren*
The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy.	Hawlitscheka, F. et al. (2018)
Vertrauen ist gut, Blockchain ist besser – Einsatzmöglichkeiten von Blockchain für Vertrauensprobleme im Crowdsourcing.	Schütz, A.E. et al. (2018)
The issue of user trust in decentralized applications running on blockchain platforms.	Bracamonte, V. und Okada, H. (2017)
Blockchain-Based Traffic Event Validation and Trust Verification for VANETs.	Yang, Y.-T. et al. (2019)
Blockchain based trust & authentication for decentralized sensor networks.	Moinet, A. et al. (2017)
TrustChain: Trust Management in Blockchain and IoT supported Supply Chains.	Malik, S. et al. (2019)
Research on trust mechanism of cooperation innovation with big data processing based on blockchain.	Liu, Q. und Zou, X. (2019)

Titel des Beitrags	*Autoren*
Trust management in social Internet of vehicles: Factors, challenges, blockchain, and fog solutions.	Iqbal, R. et al. (2019)

Hawlitscheka et al. (2018) untersuchen in ihrem Beitrag anhand einer Literaturrecherche das Potenzial der Blockchain-Technologie zur Lösung des Vertrauensproblems in einer Sharing Economy. Sie stellen fest, dass das Konzept des Vertrauens zwischen der Blockchain-Technologie und der Sharing Economy erheblich variiert, die Blockchain-Technologie jedoch bis zu einem gewissen Grad geeignet ist, dass Vertrauen in Plattformanbieter zu ersetzen. Schütz et al. (2018) setzen ihren Fokus auf das Crowdsourcing und deren aktuellen Vertrauensprobleme und entwickeln ein Reputationssystem auf Basis der Blockchain-Technologie. Mit Hilfe von Smart Contracts auf der Ethereum Blockchain werden Profile von Auftraggebern und Arbeitnehmern mit einem Reputationsscore verknüpft, um dadurch die notwendige Vertrauensgrundlage für die Geschäftsbeziehung zu erreichen [44]. Bracamonte und Okada (2017) untersuchen in ihrem Beitrag das Thema Vertrauen und vertrauensbezogene Faktoren im Zusammenhang mit dezentralen Anwendungen, die auf öffentlichen Blockchain-Plattformen ausgeführt werden. Sie differenzieren Vertrauen in soziales und technologisches Vertrauen und analysieren, inwieweit diese Anwendungen als von Dritten nicht kontrollierbar angesehen werden. Ihre Ergebnisse zeigen, dass die Websites dezentraler Anwendungen zwar auf die Konzepte Dezentralisierung, Vertrauenswürdigkeit und Autonomie verweisen, diese jedoch nicht in gleicher Weise definieren [7]. Yang et al. (2019) entwickeln ein Vertrauensmodell mit Hilfe der Blockchain-Technologie und einem Proof-of-event Konsensmechanismus, um die Legitimität und das Verhalten anonymer Knoten in einem Fahrzeugnetz zum Austausch von Verkehrsinformationen zu bewerten. Für die Sicherstellung der Integrität von kryptografischen Authentifizierungsdaten in Sensornetzwerken schlagen Moinet et al. (2017) ein Blockchain-basiertes Protokoll zur Gewährleistung eines Peer-Trust Levels vor. Auf der Grundlage der Blockchain-Datenstruktur entwickeln sie ein Modell zur dezentralen Authentifizierung. Malik et al. (2019) sehen ein Vertrauensproblem hinsichtlich der Daten in einem Audit-Trail für Lieferkettenereignisse und die Lösung in einem Reputationssystem. Sie entwickeln ein dreistufiges Trust-Management-Framework auf der Grundlage einer Konsortium-Blockchain, um die Interaktionen zwischen den Lieferkettenteilnehmern zu verfolgen und dynamisch Vertrauens- und Reputationswerte zu generieren. Liu und Zou (2019) stellen die neuesten Forschungsergebnisse des Vertrauensmechanismus in einem Peer-to-Peer Netzwerk der Blockchain vor. Der Blockchain Vertrauensmechanismus basiert auf der Beteiligung aller Mitglieder an der Überwachung, Kontrolle und Prüfung des Vertrauenswertes für den gesamten Lebenszyklus von Adressen, Schlüsseln und Daten [31]. Der Beitrag von Iqbal et al. (2019) stellt die Schlüsselfaktoren für die Konzeptionierung eines Vertrauensmodells für „…social Internet of vehicles“ (SloV) vor. Diese sind Reputation, Kontext, Umfeld, Ziele, Nutzererwartungen, soziale Beziehungen, Verbindungsbereitschaft und zeitnahe Bewertung. Sie zeigen in ihrem Beitrag, dass die Blockchain-Technologie zur Lösung der heute vorhandenen Probleme mit diesen Schlüsselfaktoren geeignet ist.

4. Ergebnisse

Die Analyse der Beiträge erfolgte auf der Grundlage eines ausführlichen Coding Handbuches, in dem sowohl die Vertrauensdomänen nach Heidt et al. (2019) sowie die systemimmanenten Vertrauenseigenschaften der Blockchain ausführlich und mit Beispielen dokumentiert wurden. Anhand dieses Coding Handbuches wurde zunächst der Abstract des Beitrags analysiert, und darüber hinaus bei Unklarheiten der gesamte Beitrag. Für jede in dem Beitrag angesprochene Vertrauensdomäne als auch Vertrauenseigenschaft der Blockchain wurde jeweils eine 1 notiert. Beispielsweise wird für den Beitrag „BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network“ von Rathore et al. (2019) eine 1 für die Vertrauenseigenschaft Integrität der Blockchain notiert, da die im Beitrag adressierte Lösung anhand der drei Kerntechnologien Software Defined Networking (SDN), Blockchain, Fog und Mobile Edge Computing die Gewährleistung der Integrität der Daten erreichen soll. Die durch den Beitrag angesprochene Vertrauensdomäne ist Vertrauen in Daten, hierfür wird ebenfalls eine 1 notiert.

Die nachfolgende Statistik (siehe Tabelle 5) zeigt das Ergebnis unserer Zuordnung der Beiträge zu den angesprochenen Vertrauensdomänen (siehe Kapitel 2.1) sowie den involvierten systemimmanenten Vertrauenseigenschaften der Blockchain (siehe Kapitel 2.2). Mit diesen Ergebnissen beantworten wir unsere Forschungsfragen FF1 und FF2.

Tabelle 5: Anzahl der Zuordnungen zu Vertrauensdomänen und Vertrauenseigenschaften

Vertrauensdomäne	*Anzahl der Beiträge*	*Vertrauenseigenschaften der Blockchain*	*Anzahl der Beiträge*
Vertrauen in Code	4	Smart Contracts	9
Vertrauen in Daten	12	Konsensmechanismus	6
Vertrauen in die Vision	0	Unveränderlichkeit	3
Systemisches Vertrauen	11	Peer-to-Peer Netzwerk	19
Authentifizierung	41	Distributed Ledger	34
Autorisierung	34	Integrität	10
		Kryptographie	24

Die Summen übersteigen die Anzahl der Beiträge von 81, da Mehrfachzuordnungen möglich waren. Im Bereich der Vertrauensdomänen wird deutlich, dass der Fokus der Beiträge und der avisierten Lösungen hinsichtlich der Authentifizierung und Autorisierung liegt.

Singh et al. (2018) beschreiben in ihrem Beitrag das Problem einer sicheren Kommunikation für intelligente Fahrzeuge und sehen wie Guo et al. (2019) die Lösung in einer sicheren Authentifizierung mit Hilfe der Blockchain. Das Distributed Ledger Konzept der Blockchain-Technologie ist die dominierende Eigenschaft für die Bildung von Vertrauen, gefolgt von den Eigenschaften Kryptographie und dem Peer-to-Peer Netzwerk. Auffällig ist, dass im Kontext von Internet of Things die Unveränderlichkeit der Daten in einer Blockchain anscheinend für die Bildung von Vertrauen keine gravierende Rolle spielt. Ebenso scheint der Konsensmechanismus in diesem Kontext eine eher untergeordnete Rolle zu spielen.

4.1 Deskriptive Statistik und Clusteranalyse

Die Clusteranalyse ist eine Gruppe multivariater Techniken, deren Hauptzweck darin besteht, Objekte auf der Grundlage ihrer Eigenschaften zu gruppieren [19]. Nach Eckstein (2016) besteht die Grundidee einer Clusteranalyse darin, eine definierte Menge von Objekten so zu gruppieren, dass die Objekte innerhalb einer Gruppe möglichst homogen bezüglich der Menge der Clustermerkmale und die Objekte unterschiedlicher Gruppen möglichst heterogen bezüglich der Menge der Clustermerkmale sind. Die Verwendung der Clusteranalyse hat in den letzten Zeiträumen erheblich zugenommen und ist weit verbreitet [27].

Bevor wir die Clusteranalyse durchführen, untersuchen wir die Pearson-Korrelation zwischen unseren Merkmalen (siehe Tabelle 7). Die Ergebnisse lassen aufgrund ihres niedrigen Niveaus eine weitere Clusteranalyse zu. Hierfür nutzen wir IBM SPSS Version 24 und verwenden die hierarchisch-agglomerative Klassifikation des Ward-Verfahrens. Diese Varianz-Methode arbeitet mit dem kleinsten Zuwachs der Fehlerquadratsumme bei einer Clusterfusion [8]. Als Abstandsmaß in der Clusteranalyse wurde der quadratische euklidische Abstand gewählt, der für binäre Variablen verwendet werden kann. Für die Bestimmung der besten Anzahl an Clustern „gibt es keine „harten“ Regeln, die für eine statistisch und sachlogisch plausible Deutung der erzielten Ergebnisse hilfreich sind“ [8]. Daher haben wir zunächst mit Hilfe des Dendrogramms eine Bestimmung der Anzahl der Cluster durchgeführt. In einem weiteren Schritt haben wir die Entscheidungsregel von Eckstein (2016) angewandt. Dazu wird der Fusionsschritt gesucht, der sich durch eine übermäßige Steigerung des Heterogenitätskoeffizienten auszeichnet. Die optimale Anzahl der Cluster ist dann die Anzahl der Fusionsschritte gesamt abzüglich des Fusionsschritts mit der übermäßigen Steigerung des Heterogenitätskoeffizienten. In unserem Fall ergeben 78 Fusionsschritte abzüglich dem 75. Fusionsschritt, der eine übermäßige Steigerung des Heterogenitätskoeffizienten zeigt, 3 Cluster. Damit wird das Ergebnis von drei Clustern der visuellen Analyse des Dendrogramms bestätigt.

Tabelle 6: Ergebnis der Cluster-Analyse

Cluster	*Frequency*	*Percent*	*Cumulative Percent*
1	30	38,0	38,0
2	32	40,5	78,5
3	17	21,5	100,0
Total	79	100,0	

Die unterschiedlichen Cluster zeigen in Tabelle 8 die Muster der Komposition der Vertrauensdomänen mit den Vertrauenseigenschaften der Blockchain.

Tabelle 8: Vertrauenseigenschaften und –domänen je Cluster

Anzahl der Vertrauenseigenschaften der Blockchain sowie Vertrauensdomänen je Cluster	*Cluster 1*	*Cluster 2*	*Cluster 3*
Smart Contracts	8	0	1
Konsensmechanismus	6	0	0
Unveränderlichkeit	2	0	1
Peer-to-Peer Netzwerk	0	10	9
Distributed Ledger	4	14	16
Integrität	6	2	2
Kryptographie	6	18	0
Vertrauenseigenschaften der Blockchain - Teilsumme :	32	44	29
Vertrauen in Code	1	0	3
Vertrauen in Daten	12	0	0
Vertrauen in die Vision	0	0	0
Systemisches Vertrauen	7	0	4
Authentifizierung	3	29	9
Autorisierung	5	28	1
Vertrauensdomänen - Teilsumme :	28	57	17
Gesamtsumme:	60	101	46

Die Vertrauenseigenschaften Smart Contracts, Konsensmechanismus, Integrität und Kryptografie führen im Cluster 1 zu einer Aktivierung der Vertrauensdomänen „Vertrauen in Daten“ und „Systemisches Vertrauen“. Im Vergleich zu den anderen Clustern ist die Bedeutung von Smart Contracts, dem Konsensmechanismus sowie der Integrität hervorzuheben. Dies begründet sich darin, dass der Konsensmechanismus der Blockchain-Technologie dessen Inhalte vor Manipulationen schützt, und damit die Vertrauensdomäne „Vertrauen in Daten“ aktiviert. Mit Smart Contracts werden Sicherheit und Unparteilichkeit bei der Ausführung von vereinbarten Transaktionen gewährleistet. Die Blockchain gewährleistet sowohl die Ausführung dieser Transaktionen als auch deren Protokollierung und schafft somit Vertrauen. Die hierauf aufbauenden Geschäftsmodelle, wie zum Beispiel im Kontext von Industrie 4.0, enthalten als wesentliche technische Grundlage die Übertragung von Daten zwischen Kunden und Anbietern über das Internet [11]. Aufgrund der hohen Automatisierungen in diesen Geschäftsmodellen ist ein systemisches Vertrauen des Kunden unabdingbar. Sowohl der Konsensmechanismus als auch Smart Contracts schaffen hierfür die notwendigen Grundlagen.

Der Cluster 2 zeigt einen deutlichen Schwerpunkt bei den Vertrauenseigenschaften Peer-to-Peer Netzwerk, Distributed Ledger sowie Kryptographie, und

führt zu einer Fokussierung auf die Vertrauensdomäne Authentifizierung und Autorisierung. Dieses Muster der Vertrauensgenerierung spricht Geschäftsmodelle an, die die Vernetzung von cyber-physischen Systemen und Menschen wie zum Beispiel mit Konsumentenelektronik im Fokus haben [40]. Mit dem Peer-to-Peer Netzwerk der Blockchain wird die Vertrauenswürdigkeit jedes Teilnehmers erreicht, verbunden mit der Vermeidung eines Single-point of failure durch die Distributed Ledger Konzeptionierung. Der dezentrale und mobile Einsatz der Konsumentenelektronik erfordert darüber hinaus ein starkes Vertrauen sowohl in die Authentifizierung als auch in die Autorisierung der Benutzer.Dieses Vertrauen wird durch die Kryptographie innerhalb der Blockchain-Technologie erreicht.

Der Cluster 3 ist dominiert von der Vertrauenseigenschaft Distributed Ledger, die wiederum zu den Vertrauensdomänen Authentifizierung sowie Systemisches Vertrauen führt. Hervorzuheben ist die Differenzierung zu den vorangegangenen Clustern hinsichtlich des Vertrauens in Code und systemisches Vertrauen. Die auf diesen Eigenschaften basierenden Geschäftsmodellen sind so einzurichten, dass den Benutzern sowohl ein hohes Vertrauen zu den programmierten Anwendungen als auch zu der Plattform beziehungsweise dem System vermittelt wird. Die in diesem Kontext relevanten IoT-Anwendungen sind beispielsweise IoT-Apps im Bereich Home Automation.

4.2 Diskussion der Ergebnisse

In diesem Beitrag wurde der Wissensstand zum Vertrauen im Kontext von IoT im Zusammenspiel mit der Blockchain-Technologie dargelegt. Eine der wesentlichen Eigenschaften der Blockchain-Technologie ist die Fähigkeit, vertrauenslose Interaktionen zwischen Menschen und Technologien zu ermöglichen [12]. Basis dieser vertrauenslosen Interaktionen ist ein Konsensalgorithmus wie beispielsweise Proof-of-work, der für die Integrität der Daten in einer Blockchain sorgt [1]. Diese Vertrauenseigenschaften der Blockchain-Technologie ergeben zusammen mit den Vertrauensdomänen drei unterschiedliche Cluster (siehe Tabelle 6), die wir nachfolgend den Wertschöpfungsstufen einer IoT-Anwendung nach Fleisch et al. (2015) zuordnen. Mit dieser Zuordnung wird deutlich, auf welchen Ebenen der Wertschöpfungsstufe die Komposition aus Vertrauenseigenschaften der Blockchain und den Vertrauensdomänen eine Konfiguration der Elemente eines Geschäftsmodells darstellen. Der Cluster 1 beispielsweise ist der Ebene 4 Analytik zugeordnet, in der Daten von Sensoren gesammelt, gespeichert, plausibilisiert und klassifiziert werden [11]. Geschäftsmodelle in diesem Kontext schaffen Mehrwert für den Kunden durch die Verarbeitung von Daten und Gewinnung von Erkenntnissen aus diesen Daten. Die Blockchain-Technologie kann an dieser Stelle mit Hilfe der Smart Contracts und den Konsensmechanismen für ein hohes Vertrauen in die Daten sorgen und damit ein systemisches Vertrauen beim Kunden erzeugen.

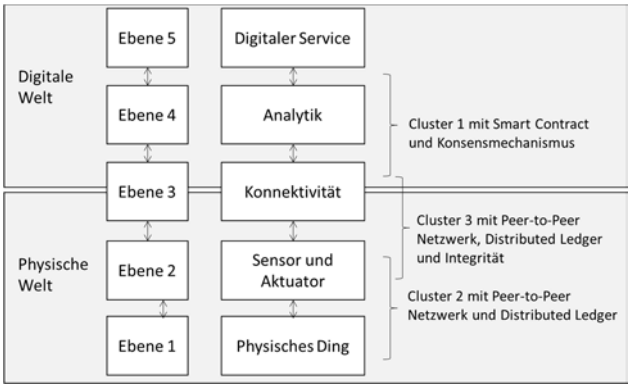


Bild 2: Wertschöpfungsstufen einer IoT-Anwendung [11]

Der Cluster 2 ist den physischen Ebenen 1 und 2 zugeordnet. Die Geschäftsmodelle mit dem Fokus auf diesen Ebenen streben die Digitalisierung von physischen Gegenständen an. Die Blockchain kann hierbei aufgrund ihres Peer-to-Peer Netzwerkes und der Distributed Ledger Technologie verbunden mit der Kryptographie für ein Vertrauen in die sichere Authentifizierung sowie Autorisierung dieser physischen Geräte sorgen. Die physische Ebene sowie die Verbindung zur digitalen Ebene ist die Zuordnung von Cluster 3, die mit Hilfe des Peer-to-Peer Netzwerkes und der Distributed Ledger Technologie die Grundlage für darauf basierende Geschäftsmodelle bietet. Diese Geschäftsmodelle benötigen neben einem hohen Vertrauen in die Authentifizierung ebenfalls ein Vertrauen sowohl in die programmierten Anwendungen als auch in das System. Eine sichere IT-Infrastruktur verbunden mit einer sicheren Konnektivität zu den Kunden schafft Vertrauen [17].

Die Ergebnisse aus unserer Cluster Analyse (siehe Tabelle 8) als auch die Einordnung der Cluster in die Wertschöpfungsstufen (siehe Bild 2) zeigen die Muster der Vertrauensgenerierung für IoT-basierte Geschäftsmodelle. Dieses Ergebnis beantwortet die Forschungsfrage FF3.

4.3 Mögliche Forschungsthemen

Die zuvor dargestellten Ergebnisse werfen ethische, rechtliche und soziale Fragestellungen (ELSI – Ethical, legal and social issues) auf und führen zu folgenden möglichen Forschungsthemen (siehe Tabelle 9).

Tabelle 9: Forschungsthemen nach ELSI-Dimensionen

ELSI Dimension	*Relevante Forschungsthemen*
Ethische Implikationen	Die Autonomie der Entscheidung wird über das durch die Blockchain generierte Vertrauen im Kontext von IoT auf die Anwendung verlagert. Die Mensch-IoT-Blockchain-Interaktion führt aufgrund ihrer Komplexität zu einer enormen Anpassungsleistung für den Menschen. Zukünftige Forschungen sollten sich diesen Veränderungen und den Auswirkungen für den Menschen widmen (z.B. Autonomieverlust, Algorithmen-Ethik).
Rechtliche Implikationen	In 2018 sind mit Art. 16 DSGVO das Recht auf Berichtigung und mit Art. 17 DSGVO das Recht auf Löschung personenbezogener Daten in Kraft getreten. Hinsichtlich der personenbezogenen Daten auf einer Blockchain wird unterschieden

ELSI Dimension	*Relevante Forschungsthemen*
	in personenbezogene Daten gespeichert auf der Blockchain sowie öffentlichen Schlüsseln der Teilnehmer einer Blockchain. Wie können in einer öffentlichen und globalen Blockchain für IoT-Applikationen die Art. 16 und 17 der DSGVO eingehalten werden? Wie ist die Blockchain zu konzipieren, um einem Betroffenen das Recht auf Berichtigung und das Recht auf Löschung seiner Daten zu gewähren?
Soziale Implikationen	Die Generierung von Vertrauen im Kontext von IoT durch die Blockchain und deren Komplexität führt zu einer Verantwortungsdiffusion, in dem für den Menschen nicht mehr klar erkennbar ist, von wem und an welcher Stelle das notwendige Vertrauen stattgefunden hat. Die Zuordnung der Vertrauensgenerierung zu einer verantwortlichen Stelle ist nicht mehr gegeben. Die sich hieraus ergebenden Auswirkungen im Verhalten der Menschen mit IoT sollten analysiert werden sowie Handlungsfelder zur Problemlösung ermittelt werden.

5. Limitationen und Ausblick

Die Generierung von (System-) Vertrauen im Kontext von IoT mit Hilfe der Blockchain-Technologie bleibt den Beweis einer verlässlichen Bewertung mangels einer verlässlichen Messung schuldig. Die Kernfrage, wie sich Vertrauen in der digitalen Welt verlässlich messen lässt, konnte mit diesem Beitrag nicht beantwortet werden. Sie stand allerdings auch nicht im Fokus unserer Analysen. Dennoch schafft die Blockchain-Technologie anhand ihrer systemimmanenten Vertrauenseigenschaften eine Grundlage zur Bildung von Vertrauen im Kontext von IoT. Die bisherigen Lösungen setzen auf zentrale Vertrauensinstanzen, die jedoch aufgrund der Vielzahl und Heterogenität der Geräte keinen verlässlichen und überzeugenden Schutz gegenüber Cyberkriminalität bieten können [15]. Die Distributed Ledger Technologie der Blockchain in einem Peer-to-Peer-Netzwerk hat das Potential zur Schaffung von Vertrauen im Kontext von IoT. Jedoch gilt zu bedenken, dass die komplexe Blockchain-Technologie in der Breite der Gesellschaft in Deutschland bisher eine unverstandene Technologie ist [39]. An dieser Stelle sind weitere Forschungen dahingehend notwendig, ob ein Vertrauen in eine technologische Lösung unabhängig vom Wissensstand über die Technologie generiert werden kann.

Neben den in Bild 1 dargestellten systemimmanenten Eigenschaften der Blockchain zur Generierung von Vertrauen im Kontext von IoT sind weitere Eigenschaften der Blockchain geeignet, das Vertrauen in dezentrale Anwendungsgebiete wie Logistik, Gesundheit oder Medizin zu erzielen. Die Blockchain findet in einer vernetzten Welt ihre Anwendungsmöglichkeiten, jedoch gibt es nicht die „eine“ Blockchain, sondern eine Vielzahl von Blockchain Varianten mit unterschiedlichen Anwendungsgebieten. An dieser Stelle sind weitere Forschungen hinsichtlich der unterschiedlichen Vertrauenseigenschaften sinnvoll.

6. Literatur

[1] Azaria, Asaph, Ekblaw, Ariel, Vieira, Thiago, Lippman, Andrew (2016), MedRec: Using Blockchain for Medical Data Access and Permission Management, in: 2016 2nd International Conference on Open and Big Data, DOI 10.1109/OBD.2016.11.

[2] Beck, Roman, Müller-Bloch, Christoph (2017), Blockchain as Radical Innovation: A Framework for Engaging with Distributed Ledgers, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2017).

[3] Benshoof, Brendan, Rosen Andrew, Bourgeois, Anu G., Harrison, Robert W. (2016), Distributed Decentralized Domain Name Service, in: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, DOI 10.1109/IPDPSW.2016.109 DOI 10.1109/IPDPSW.2016.109.

[4] Bitkom (2019), Blockchain in Deutschland – Einsatz, Potenziale, Herausforderungen. 2019, abrufbar unter: https://www.bitkom.org/Bitkom/Publikationen/Blockchain-Deutschland-Einsatz-Potenziale-Herausforderungen, Stand: 19. Juli 2019.

[5] Bohn, Ursula (2007), Welchen Einfluss haben Reorganisationsmaßnahmen auf Vertrauensprozesse? Eine Fallstudie, Inaugural-Dissertation 2007, Ludwig-Maximilians-Universität München.

[6] Bordel, Borja, Alcarria, Ramon, Martin, Diego, Sanchez-Picot, Alvaro (2019), Trust Provision in the Internet of Things Using Transversal Blockchain Networks, in: Intelligent Automation and Soft Computing, Jahrgang 25, Ausgabe 1, S. 155-170.

[7] Bracamonte, Vanessa, Okada, Hitoshi (2017), The issue of user trust in decentralized applications running on blockchain platforms, in: 2017 IEEE International Symposium on Technology and Society (ISTAS), Sydney, NSW.

[8] Eckstein, Peter P. (2016), Angewandte Statistik mit SPSS. Praktische Einführung für Wirtschaftswissenschaftler, 8. Auflage, Springer-Gabler, Wiesbaden.

[9] Engelschall, Ralf S. (2019), Blockchain. Suchen wir nur das Problem zur Lösung?, in: Informatik Spektrum, Jahrgang 42, Ausgabe 3, S. 205–210.

[10] Fleich, Elgar, Weinberger, Markus, Wortmann, Felix (2014), Geschäftsmodelle im Internet der Dinge, in: HMD Praxis der Wirtschaftsinformatik, Jahrgang 51, Ausgabe 6, S. 812-826.

[11] Fleisch, Elgar, Weinberger, Markus, Wortmann, Felix (2015), Geschäftsmodelle im Internet der Dinge, in: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, Jahrgang 67, S. 444-464.

[12] Foroglou, Georgios, Tsilidou, Anna Lali (2015), Further applications of the blockchain, Conference Paper, in: Proceedings of the 12th Student Conference on Managerial Science and

Technology, Athens, Greece.
[13] Frauchiger, Daniel (2017), Anwendungen von Design Science Research in der Praxis, In: Portmann, Edy (Eds.) Wirtschaftsinformatik in Theorie und Praxis, Festschrift zu Ehren von Prof. Dr. Andreas Meier, Springer Fachmedien, Wiesbaden 2017.
[14] Glaser, Florian (2017), Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis, in: Proceedings of the 50th Hawaii International Conference on System Sciences 2017.
[15] Goncalves, Nicholas (2018), Can decentralised networks influence the level of digital trust in eCommerce sites? DOI: 10.13140/RG.2.2.21778.71364.
[16] Green, Charles H. (2019), Trust and the Sharing Economy: A New Business Model, abrufbar unter: https://trustedadvisor.com/trust-and-the-sharing-economy-a-new-business-model, Stand: 18. November 2019.
[17] Grünert, Lars, Sejdic, Goran (2017), Industrie 4.0-getriebene Geschäftsmodellinnovationen im Maschinenbau am Beispiel von TRUMPF, In: Seiter, Mischa, Grünert, Lars, Berlin, Sebastian (Hrsg.), Betriebswirtschaftliche Aspekte von Industrie 4.0, ZfbF-Sonderheft 71.17, Springer-Gabler.
[18] Guo, Shaoyong., Hu, Xing, Zhou, Ziqiang, Wang, Xinyan et al. (2019), Trust access authentication in vehicular network based on blockchain, in: China Communications, Jahrgang 16, Ausgabe 6, Juni 2019.
[19] Hair, Joseph F., Black, William C., Babin, Barry J., Anderson, Roph E. (2014), Multivariate Data Analysis, 7. Auflage, Pearson Education Limited.
[20] Hammi, Mohamed Tahar, Hammi, Badis, Bellot, Patrick, Serhrouchni, Ahmed (2018), Bubbles of Trust: A decentralized blockchain-based authentication system for IoT, in: Computers & Security, Jahrgang 78, S. 126-142.
[21] Hawlitscheka, Florian, Notheisena, Benedikt, Teubnerb, Timm (2018), The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy, in: Electronic Commerce Research and Applications, Jahrgang 29, S. 50-63.
[22] Heidt, Michael, Berger, Arne, Bischof, Andreas (2019), Blockchain and Trust: A Practice-Based Inquiry, In: Nah, Fiona Fui-Hoon, Siau, Keng (Hrsg.), HCI in Business, Government and Organizations. eCommerce and Consumer Behavior, HCII 2019, Lecture Notes in Computer Science, Springer.
[23] Hörler, Salomon (2015), Vertrauen im Zeitalter der digitalen Moderne. Ein Mechanismus der Reduktion digitaler Komplexität, abrufbar unter: http://www.effibeisst.com/portfolio_page/vertrauen-im-zeitalter-der-digitalen-moderne/, Stand: 14. November 2019. 14.11.2019)
[24] Huber, Daniel, Kaiser, Thomas (2015), Wie das Internet der Dinge neue Geschäftsmodelle ermöglicht, in: HMD Praxis der Wirtschaftsinformatik, Oktober 2015, Heft 52, Ausgabe 5, S. 681-689.
[25] Idelberger, Florian, Governatori, Guido, Riveret, Regis, Sartor, Giovanni (2016), Evaluation of Logic-Based Smart Contracts for Blockchain Systems, in: Conference Paper, July 2016, DOI: 10.1007/978-3-319-42019-611.
[26] Iqbal, Razi, Butt, Talal Ashraf, Afzaal, Muhammad, Salah, Khaled (2019), Trust management in social Internet of vehicles: Factors, challenges, blockchain, and fog solutions, in: International Journal of Distributed Sensor Networks, Jahrgang 15, Ausgabe 1, DOI: 10.1177/1550147719825820.
[27] Kettenring, Joan R. (2006), The Practice of Cluster Analysis, Journal of Classification, Jahrgang 23, S. 3–30.
[28] Kratz, Hans-Jürgen (2017), Erfolgreich führen von A-Z. Für gute Vorgesetzte und zufriedene Mitarbeiter, Metropolitan, Regensburg 2017.
[29] Lewin, Marcus, Dogan, Alaettin, Schwarz, Jonas, Fay, Alexander (2019), Distributed-Ledger-Technologien und Industrie 4.0. Eine Untersuchung der Relevanz für Industrie 4.0, in: Informatik Spektrum, Band 42, Heft 3, Juni 2019. Seite 166-173.
[30] Linnhoff-Popien, Claudia (2018), 1. Internet of Things (IoT), in: Digitale Welt, Jahrgang 3, https://doi.org/10.1007/s42354-018-0102-6.
[31] Liu, Qi, Zou, Xiao (2019), Research on trust mechanism of cooperation innovation with big data processing based on blockchain, in: Journal on Wireless Communications and Networking, https://doi.org/10.1186/s13638-019-1340-5.
[32] Luber, S., Schmitz, P. (2019), Was ist Authentifizierung? abrufbar unter: https://www.security-insider.de/was-ist-authentifizierung-a-617991/, Stand: 18. Juli 2019.
[33] Malik, Sidra, Dedeoglu, Volkan, Kanhere, Salil S., Jurdak, Raja (2019), TrustChain: Trust Management in Blockchain and IoT supported Supply Chains, in: IEEE Blockchain 2019, arXiv:1906.01831 [cs.CR].
[34] Mkpa, Akpanakak, Chin, Jeannette, Winckles, Adrian (2019), Holistic Blockchain Approach to Foster Trust, Privacy and Security in IoT based Ambient Assisted Living Environment, abrufbar unter: https://www.researchgate.net/publication/333132719_Holistic_Blockchain_Approach_to_Foster_Trust_Privacy_and_Security_in_IoT_based_Ambient_Assisted_Living_Environment, Stand: 19. Juli 2019.
[35] Möllering, Guido (2019), Grundlagen des Vertrauens: Wissenschaftliche Fundierung eines Alltagsproblems, abrufbar unter: https://www.mpg.de/451610/forschungsSchwerpunkt, Stand: 16. Juli 2019.
[36] Moinet, Axel, Darties, Benoît, Baril, Jean-Luc (2017), Blockchain based trust & authentication for decentralized sensor networks, in: IEEE Security & Privacy, Special Issue on Blockchain, arXiv:1706.01730 [cs.CR].

[37] Naerland, Kristoffer, Müller-Bloch, Christoph, Beck, Roman, Palmund, Søren (2017), Blockchain to Rule the Waves - Nascent Design Principles for Reducing Risk and Uncertainty in Decentralized Environments, in: Proceedings ICIS 2017.

[38] Pietro, Robert Di, Salleras, Xavier, Signorini, Matteo, Waisbard, Erez (2018), A blockchain-based distributed Trust System for the Internet of Things, SACMAT'18, Juni 13-15, 2018, Indianapolis.

[39] Pwc (2016), Privatkundengeschäft der Zukunft. Juli 2016, abrufbar unter: https://www.pwc.de/de/finanzdienstleistungen/digital/pwc-befragung-privatkundengeschaeft der-zukunft.pdf, Stand: 19. Juli 2019.

[40] Ranz, Fabian, Guldin, Marc (2018), Geschäftsmodelle für die Industrie 4.0, Erfolgsfaktoren, Hindernisse und Anwendungsbeispiele, abrufbar unter: https://www.esb-business-school.de/fileadmin/user_upload/Fakultaet_ESB/Forschung/Wertschoepfungs-_und_Logistiksysteme/ESB_Business_School_GENI40_Studie_Geschaeftsmodelle_fuer_die_Industrie_40.pdf, Stand: 19. November 2019.

[41] Rodig, Jan (2017), Erfolgreiche IoT-Geschäftsmodelle, Chancen im Internet der Dinge und der Industrie 4.0 nutzen, abrufbar unter: http://fs-media.nmm.de/ftp/ITI/ITP/files/vortraege/2017/jan-rodig-tresmo.pdf, Stand: 18. November 2019.

[42] saïd Business school (2019), rebuilding trust in Business, A collaborative research project between DLA Piper, the Oxford University Centre for Corporate reputation, saïd Business school, University of Oxford; and Populus, abrufbar unter: https://www.sbs.ox.ac.uk/sites/default/files/2019-04/Rebuildingtrustinbusiness.pdf, Stand: 18. November 2019.

[43] Schubert, Manuel (2014), Vertrauensmessung in der digitalen Welt. Übersicht und Ausblick, DIVSI Diskussionsbeiträge 06, 2014, ISSN 2196-6729.

[44] Schütz, Andreas E., Fertig, Tobias, Weber, Kristin et al. (2018), Vertrauen ist gut, Blockchain ist besser – Einsatzmöglichkeiten von Blockchain für Vertrauensprobleme im Crowdsourcing, in: HMD Praxis der Wirtschaftsinformatik, Jahrgang 55, Ausgabe 6, S. 1155-1166.

[45] Schwabe, Gerhard (2017), Blockchain-Enhanced Trust in International Trade, In: Beck, Roman, Becker, Christian, Lindman, Juho, Rossi, Matti (Hrsg.), Opportunities and Risks of Blockchain Technologies, Report from Dagstuhl Seminar 17132. 2017, 10.4230/DagRep.7.3.99.

[46] Schwarzkopf, Julia, Adam, Katarina, Wittenberg, Stefan (2018), Vertrauen in nachhaltigkeitsorientierte Audits und in Transparenz von Lieferketten – Schafft die BlockchainTechnologie einen Mehrwert? In: Khare, Anshuman, Kessler, Dagmar, Wirsam, Jan (Hrsg.), Marktorientiertes Produkt- und Produktionsmanagement in digitalen Umwelten, Festgabe für Klaus Bellmann zum 75. Geburtstag, SpringerGabler, S. 171-180.

[47] She, Wei, Liu, Qi, Tian, Zhao, Chen, Jian-Sen, Wang, Bo, Liu, Wei (2019), Blockchain Trust Model for Malicious Node Detection in Wireless Sensor Networks, in: IEEE ACCESS, Jahrgang 7, S. 38947-38956-

[48] Singh, Madhusudan, Kim, Shiho (2018), Trust Bit: Reward-based intelligent vehicle commination using blockchain paper, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT).

[49] Suchanek, Andreas (2019), Vertrauen, abrufbar unter: https://wirtschaftslexikon.gabler.de/definition/vertrauen-50461, Stand: 16. Juli 2019.

[50] Söllner, Matthias, Benbasat, Izak, Gefen, David, Leimeister, Jan Marco, Pavlou, Paul A. (2016), Trust, An MIS Quarterly Research Curation, MISQ Research Curation, abrufbar unter: https://misqresearchcurations.files.wordpress.com/2016/10/trust-research-curation_oct-31-20161.pdf, Stand: 21. November 2019.

[51] Son Jai-Yeol, Tu Lingling, Benbasat Izak (2006), A descriptive content analysis of trust-building measures in B2B electronic marketplaces, in: Communications of the Association for Information Systems, Jahrgang 18, https://doi.org/10.17705/1CAIS.01806.

[52] Treiblmaier, Horst, Önder, Irem (2018), 1. The Impact of Blockchain on the Tourism Industry: A Theory-Based Research Framework, In: Treiblmaier, Horst, Beck, Roman (Hrsg.), Business Transformation through Blockchain, Volume II, palgrave macmillan.

[53] Walterbusch, Marc, Teuteberg, Frank, Gräuler, Matthias (2014), How Trust is Defined: A Qualitative and Quantitative Analysis of Scientific Literature, AMCIS 2014.

[54] Yang, Yao-Tsung, Chou, Li-Der, Tseng, Chia-Wei, Tseng, Fan-Hsun, Liu, Chien-Chang (2019), Blockchain-Based Traffic Event Validation and Trust Verification for VANETs, in: IEEE Access, Jahrgang 7, DOI: 10.1109/ACCESS.2019.2903202.

STATISTICAL ANOMALY DETECTION IN ETHEREUM TRANSACTION GRAPHS

Kevin Wittek[1], Neslihan Wittek[2], Andrei Ioniță[3], Norbert Pohlmann[1]

The set of transactions that occurs on the public ledger of an Ethereum network in a specific time frame can be represented as a directed graph, with vertices representing addresses and an edge indicating the interaction between two addresses. While there exists preliminary research on analyzing an Ethereum network by the means of graph analysis, most existing work is focused on either the public Ethereum Mainnet or on analyzing the different semantic transaction layers using static graph analysis in order to carve out the different network properties (such as interconnectivity, degrees of centrality, etc.) needed to characterize a blockchain network. By analyzing the consortium-run bloxberg Proof-of-Authority (PoA) Ethereum network, we show that we can identify suspicious and potentially malicious behaviour of network participants by employing statistical graph analysis. We thereby show that it is possible to identify the potentially malicious exploitation of an unmetered and weakly secured blockchain network resource. In addition, we show that Temporal Network Analysis is a promising technique to identify the occurrence of anomalies in a PoA Ethereum network.

1. Introduction

Ethereum is a popular technology in the blockchain space that combines a rich shared-state model (rich referring to the state history being a core part of the system) with a quasi-Turing complete transaction-based state machine [1]. The default Ethereum protocol uses a Proof-of-Work (PoW) consensus mechanism, that shares some similarities with Bitcoin's Hashcash based PoW as proposed by Nakamoto [2] and Back [3]. However Natoli and Gramoli [4] have shown that the inherently forkable nature of PoW based blockchain protocols makes them vulnerable to e.g. double-spending attacks, especially in the context of consortium run blockchain networks. In addition, there has been a rising scepticism about blockchain technology with regards to the energy consumption and sustainability aspects of PoW based protocols, often equalizing blockchain in general with high energy consumption [5].

The Proof-of-Authority (PoA) consensus mechanism is a proposed alternative to PoW for certain blockchain network topologies and use cases. Instead of proving the investment of computing resources, it uses a set of authority nodes (often called validators) that are in charge of creating new blocks, which is called sealing in contrast to mining. Confirmations happen as soon as a certain threshold of authorities agree and sign the respective transactions. Among its advantages are the relatively short block confirmation times, due to fixed block creation times. In fact, the good distribution of authorities in the network accounts for security, especially against malicious 51% attacks. Furthermore, PoA networks are more predictable, as blocks are issued at constant time intervals. PoA is particularly effective for public-permissioned networks.

The bloxberg network [6] is a global blockchain network established by an international consortium of research organisations for scientific purposes. Its mission is to build applications in the network that promote collaboration in all research areas while remaining decentralized and robust to accommodate future requirements of the research community. bloxberg's governance is based on on-chain voting from the consortium members, while the ensuing actions are executed by the Iron Throne holder, a position that is voted for off-chain once a year. bloxberg uses a PoA consensus based on the Authority Round (AuRa) algorithm [7], that ensures availability, consistency and performance, apart from the aforementioned security properties. The bloxberg network provides a faucet application that enables members to acquire bloxberg's cryptocurrency, called bergs (which is functionally equivalent to Ethereum's Ether), to pay for the gas costs to deploy and use their smart contracts and applications. At the same time, bergs are acquired automatically while participating as a validator in the consensus. In fact, in practice, there has been a low demand for bergs from the members as soon as they collected a starting amount, by which their decentralized application (DApps/dApps) could be deployed for the first time.

The faucet application is accessible for human users as a web application and secured against fraud and abuse using Google's reCAPTCHA service in version 2 [8]. However, general operational monitoring of the faucet application, as well as random sample investigations of faucet usage in the past, have demonstrated not only suspicious and potentially malicious patterns but also exploitative faucet usage patterns. The following analyzes show that potentially exploitative activities did indeed occur, while also identifying potential heuristics that can be used for future security monitoring setups.

[1] Institute for Internet Security, Westphalian University of Applied Sciences, Neidenburger Straße 43, D-45897 Gelsenkirchen, Germany
[2] Faculty of Psychology, Department of Biopsychology, Ruhr University Bochum, Universitätsstraße 150, D-44801 Bochum, Germany
[3] Fraunhofer Institute for Applied Information Technology FIT, Fraunhofer Society for the Advancement of Applied Research, Schloss Birlinghoven 1, D-53757 Sankt Augustin, Germany

2. Model

The Ethereum transaction ledger can be modelled as a directed multigraph [9] - [11], containing edges with identity (identity properties of edges are block number and transaction value), with multiple edges being allowed but not required. The nodes of the graph represent Ethereum addresses. A transaction from address A to address B creates a directed edge from node A to node B. The graph G is, therefore, an ordered pair G = (N, E), with N being the set of nodes and E being the set of ordered pairs of nodes, i.e. edges, representing a transaction between those nodes. Looking at the ledger at a certain block number results in a certain graph. By analyzing the evolution of the graph over time, certain events can be inferred and clusters of transactions that happened in a short period of time can be identified.

Bai et al. [12] constructed three different types of graphs with the goal to uncover fundamental properties of Ethereum transaction: user-to-user graphs (UUG), contract-to-contract graphs (CCG) and user-contract graphs (UCG). UUGs describe a directed graph where the direction of the edges is dictated by the transfer of Ether between externally owned accounts (EOAs). For CCG the edges represent a creation or call action towards a smart contract, while UCG reveals how externally owned accounts use smart contracts in Ether transfers. For analysing the graph dynamics sliding windows and incremental windows are employed. It is observed that a sliding window of 180 days is suitable for analysis, as 70% of the nodes have a lifetime of below 180 days. The granularity is of about a quarter of the window size, i.e. 45 days. The incremental window expands from 180 days to 1260 days with the same granularity. As far as degree distribution is concerned, it was observed that about a quarter of the nodes (23.58%) have transactions with a single address, while 97.45% have transactions with less than ten addresses. Furthermore, patterns of interaction between node triplets are identified and counted. It was found that closed triplets, i.e. 3-node graphs where the unoriented edges describe a triangle and hence signify that all pairs of nodes are in a relationship, are negligible relative to the open triplets, the rest of the 3-node graphs.

3. Observation

For this paper, all transactions since the genesis block (which was sealed in January 2019) up until the current block number 9527285 (which was sealed in August 2020) are analyzed. In this period of time, a total number of 9527286 transactions have been recorded (see Fig. 1), which results in a transaction frequency of approximately 0.2 transactions per second.

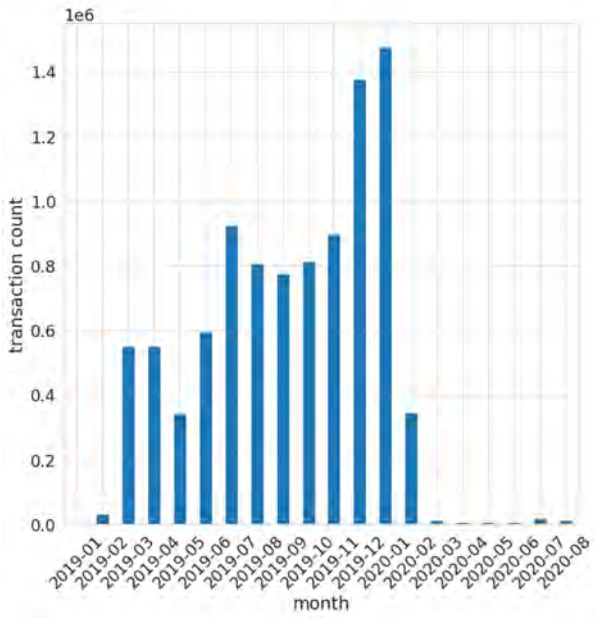


Fig. 1: Transaction count in bloxberg since the genesis block up until August 2020

The drastic decrease in monthly transactions beginning in February 2020 is due to the fact that the bloxberg network did perform the Ethereum Istanbul hard fork in this period of time which also deactivated the use of an internal monitoring Smart Contract, that was used to observe and monitor the behaviour of validators. If we filter those transactions, we get a much more accurate overview of the network traffic (see Fig. 2). When these transactions are excluded, a total number of 114256 transactions occur in the same period of time, which results in a transaction frequency of approximately 0.002 transactions per second.

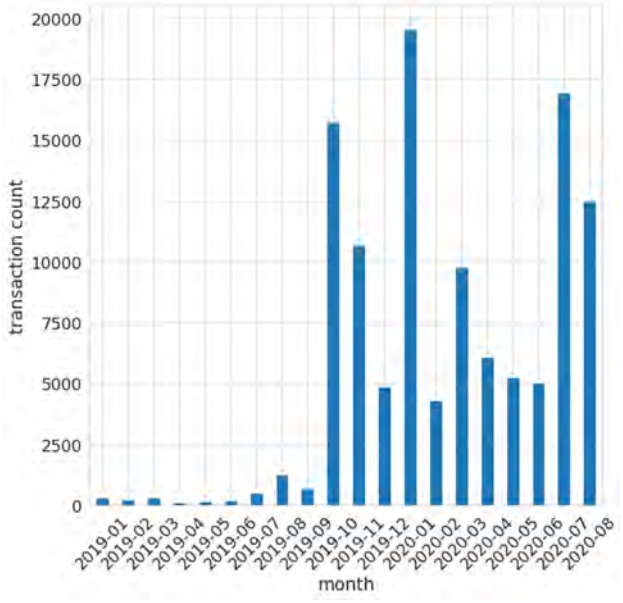


Fig. 2: Filtered transaction count in bloxberg since the genesis block up until August 2020

If we compare this to a public Ethereum network such as the Ethereum Mainnet, which possesses a transaction frequency of approximately 1.3 transactions per second in its initial 2 years beginning in 2015 [13], bloxberg can be considered a low traffic

network. It is important to consider this for further analyzes.

Of all addresses receiving funds from the faucet account (which accounts for 235 addresses in total), 38.3 % receive a single transaction (which accounts for 90 addresses in total), with 19.6 % receiving over 10 transactions (which accounts for 46 addresses in total). In general, the distribution shows the absolute peak for low transaction numbers, with high transaction numbers being seemingly suspicious (see Fig. 3).

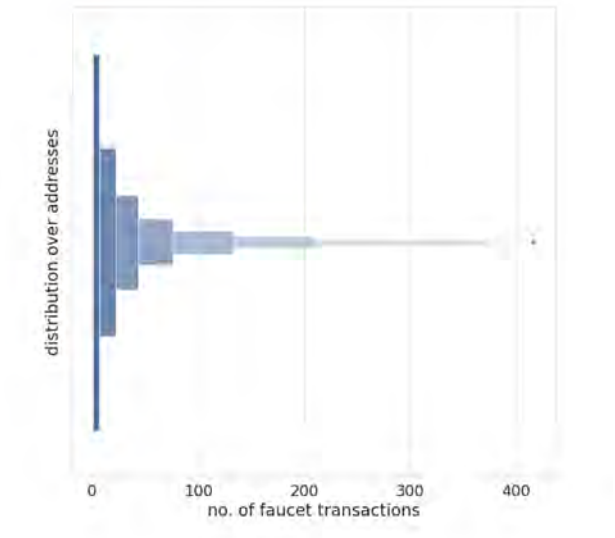


Fig. 3: Distribution of faucet transaction over addresses in bloxberg

If we focus our observations exclusively on those suspicious addresses, we can still see a peak distribution for lower transaction counts, while we also see further local maxima in the distribution for higher transaction counts (see Fig. 4).

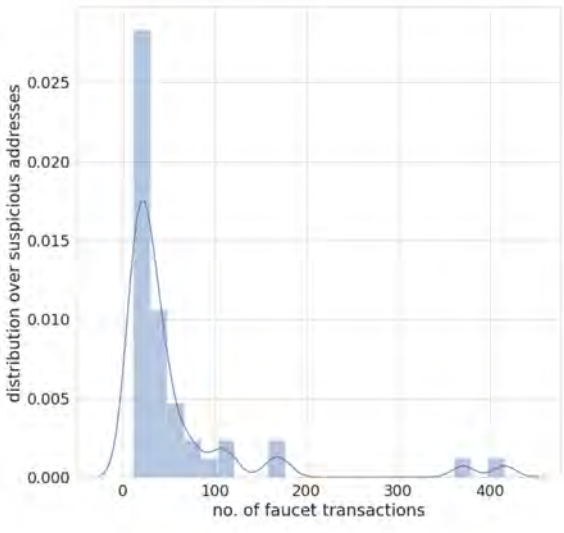


Fig. 4: Distribution of faucet transaction over addresses in bloxberg for addresses with faucet transactions count > 10

These observations with regards to faucet transaction distribution characteristics assured us in the hypothesis, that a high number of faucet transactions for a single address can be a potential heuristic for identifying suspicious addresses.

4. Results

Using the seemingly arbitrary but promising heuristic of faucet transactions counts with a cut-off value of 10 transactions, 46 suspicious addresses have been identified for which we compared the resulting transaction network graph with the transaction network graph of regular addresses. Note that transactions by validators, as well as by the faucet address, have been excluded.

We defined the connectivity of a node in the transaction graph N_{con} as the ratio between connected nodes, i.e. neighbours, N_n and the number of ingoing and outgoing edges E_{sum}:

$$N_{Con} = N_n / E_{sum}$$

We can observe a highly different connectivity distribution when comparing suspicious and regular accounts (see Fig. 5). To confirm this interpretation, a one-way between-subjects ANOVA revealed a significant difference of node connectivity between suspicious and regular addresses. ($F(1, 1281) = 120$, $p < 0.001$).

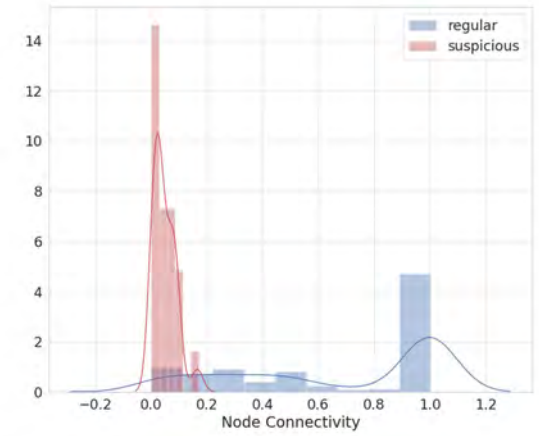


Fig 5: Comparison of node connectivity distribution between regular and suspicious addresses

Therefore we can conclude that those suspicious addresses show a transaction behaviour that differs from regular accounts, leading us to further heuristics for better classification of suspicious addresses. Consecutively, we performed a manual forensic analysis of the top 10 suspicious accounts with the highest number of faucet transactions. Simultaneously, the list of suspicious addresses has been forwarded to the bloxberg consortium, with one address in the top 10 being identified as a false positive (FP). In addition, another of the addresses seemed to indicate an FP, while for the others, 4

different patterns of exploitation could be identified: Manual, Automated, Mule & Mule Receiver.

Manual describes a seemingly manual faucet usage by a human user. Automated shows similar characteristics to Manual, but the transaction timing hints at a machine aided triggering of faucet transactions. Mule & Mule Receiver can be considered a pattern tuple, with the Mule accumulating a certain amount of bergs, which are then transferred to the Mule Receiver in a single transaction. This behaviour pattern is interesting since it shows more creativity and potential goal orientation in the exploit.

5. Temporal Network Analysis

While the previous analyzes showed promising results employing statistical techniques without considering time-based dynamics, the manual forensic analysis revealed different time-based transaction patterns which were identifiable for a human observer. It is therefore reasonable to further investigate possible techniques that allow for more precise and accurate detection of malicious activities.

Temporal network analysis has been traditionally employed in fields such as social network analysis [14], trade patterns [15] - [17] and even network neuroscience [18]. However, the time-related nature of blockchain transactions seems to make a blockchain transaction graph well suited for temporal network analysis as well.

Teneto [19] is an open-source Python package for temporal network analysis, which allows the generation of temporal network objects, the computation of various graph measures and provides specialized functions for neuroimaging use cases. Temporal network objects can be constructed from either NumPy arrays [20], pandas DataFrames [21], or built-in Python data structures, such as dictionaries or lists of edges. Centrality measures provide a way to quantify the attributes of a node in the network, such as temporal degree centrality, temporal betweenness centrality, temporal closeness centrality or a burstiness coefficient. Other measures such as burstiness, node neighbourhood, and edge properties are available as well.

Teneto has built-in plotting capabilities when used in conjunction with matplotlib [22]. A time unit can hold multiple transactions when the time axis is split into intervals. In order to eliminate empty columns, we have discretized time and assigned it the transaction index. In its simplest form, a time unit therefore corresponds to a transaction. The rows ids have been simplified too and have been assigned the addresses' index. For the bloxberg transaction set, Fig. 6 shows the plotted temporal network for the first 25 transactions. When the time unit was set to span 10 transactions, bloxberg's first 200 transactions were represented as in Fig. 7.

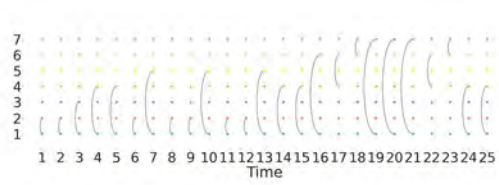


Fig. 6: Temporal network of the first 25 bloxberg transactions with time unit size equal to a single transaction

While the plotting of a temporal network might be helpful for network exploration and pattern discovery for a human analyst, future work is needed to identify classification functions for anomaly detection and malicious behaviour, which consecutively can be applied on a bigger scale via automated methods. With regards to faucet exploitation, the burstiness coefficient seems to be a promising property for further investigations.

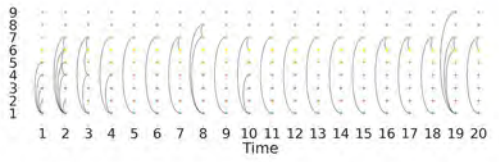


Fig. 7: Temporal network of the first 200 bloxberg transactions with time unit size of 10

6. Discussions and Mitigations

We've shown that by applying a statistical approach for analyzing the transaction graph of a PoA Ethereum network, potentially malicious and exploitative activities can be detected. Since this approach provides rather broad heuristics, it has to be supported by additional manual forensics work. However, future implementations might incorporate additional heuristics originating from temporal network analysis, which could lead to better automated classification results.

All findings have been shared with the bloxberg consortium and possible mitigations and countermeasures were discussed. Although bloxberg is not fully decentralized, since it is PoA based, the degree of decentralization makes it functionally impossible to block malicious actors from using the network, even after being identified. However, they could be restricted from using the faucet application, thereby cutting them off from collecting further funds. Future countermeasures are planned to include better monitoring and alerting of malicious faucet behaviour, with subsequent automated updates of a deny list for malicious addresses as part of the faucet application.

Acknowledgements

We like to thank all members of bloxberg for the operation and governance of the bloxberg infrastructure.
This work was partially supported by the Ministry of Economic Affairs, Innovation, Digitalisation and Energy of the State of North Rhine-Westphalia as part of the connect.emscherlippe project at the Westphalian University of Applied Sciences in Gelsenkirchen.

References

[1] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” EIP-150 REVISION.
[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
[3] A. Back, “Hashcash - A Denial of Service Counter-Measure”, 2002.
[4] C. Natoli and V. Gramoli, “The Balance Attack or Why Forkable Blockchains are Ill-Suited for Consortium,” in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Jun. 2017, pp. 579–590, doi: 10.1109/DSN.2017.44.
[5] C. Schinckus, “The good, the bad and the ugly: An overview of the sustainability of blockchain technology,” Energy Research & Social Science, vol. 69, p. 101614, Nov. 2020, doi: 10.1016/j.erss.2020.101614.
[6] F. Kleinfercher, S. Vengadasalam, J. Lawton, “Bloxberg - The Trusted Research Infrastructure”, Whitepaper 1.1, Last Accessed: Aug. 28, 2020 [Online]. Available at https://bloxberg.org/wp-content/uploads/2020/02/bloxberg_whitepaper_1.1.pdf.
[7] Aura Consensus Protocol Audit, Last Accessed: Aug. 28, 2020. [Online] Available at: https://github.com/poanetwork/wiki/wiki/Aura-Consensus-Protocol-Audit.
[8] “reCAPTCHA,” reCAPTCHA. https://www.google.com/recaptcha/about/ (accessed Aug. 30, 2020).
[9] F. Harary, Graph theory, 15. print. Cambridge, Mass: Perseus Books, 2001.
[10] J. L. Gross and J. Yellen, Graph theory and its applications, 2nd ed. Boca Raton: Chapman & Hall/CRC, 2006.
[11] S. V. Pemmaraju and S. S. Skiena, Computational discrete mathematics: combinatorics and graph theory with Mathematica. Cambridge, U.K. ; New York: Cambridge University Press, 2003.
[12] Q. Bai, C. Zhang, Y. Xu, X. Chen, und X. Wang, “Evolution of Ethereum: A Temporal Graph Perspective“, arXiv:2001.05251 [cs], Jan. 2020, Zugegriffen: Aug. 25, 2020. [Online]. Verfügbar unter: http://arxiv.org/abs/2001.05251.
[13] etherscan.io, “Ethereum Daily Transactions Chart | Etherscan,” Ethereum (ETH) Blockchain Explorer. http://etherscan.io/chart/tx (accessed Aug. 30, 2020).
[14] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, “Temporal distance metrics for social network analysis,” in Proceedings of the 2nd ACM workshop on Online social networks, New York, NY, USA, Aug. 2009, pp. 31–36, doi: 10.1145/1592665.1592674.
[15] B. L. Dutta, P. Ezanno, and E. Vergu, “Characteristics of the spatio-temporal network of cattle movements in France over a 5-year period,” Preventive Veterinary Medicine, vol. 117, no. 1, pp. 79–94, Nov. 2014, doi: 10.1016/j.prevetmed.2014.09.005.
[16] H. H. K. Lentz et al., “Disease Spread through Animal Movements: A Static and Temporal Network Analysis of Pig Trade in Germany,” PLOS ONE, vol. 11, no. 5, p. e0155196, May 2016, doi: 10.1371/journal.pone.0155196.
[17] X. Fan, X. Li, J. Yin, and J. Liang, “Temporal Characteristics and Spatial Homogeneity of Virtual Water Trade: A Complex Network Analysis,” Water Resour Manage, vol. 33, no. 4, pp. 1467–1480, Mar. 2019, doi: 10.1007/s11269-019-2199-2.
[18] W. H. Thompson, P. Brantefors, and P. Fransson, “From static to temporal network theory: Applications to functional brain connectivity,” Network Neuroscience, vol. 1, no. 2, pp. 69–99, Jun. 2017, doi: 10.1162/NETN_a_00011.
[19] William Hedley Thompson, granitz, Vatika Harlalka, and lcandeago, wiheto/teneto: 0.5.0. Zenodo, 2020.
[20] S. van der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy Array: A Structure for Efficient Numerical Computation,” Computing in Science Engineering, vol. 13, no. 2, pp. 22–30, Mar. 2011, doi: 10.1109/MCSE.2011.37.
[21] W. McKinney, “Data Structures for Statistical Computing in Python,” Proceedings of the 9th Python in Science Conference, pp. 56–61, 2010, doi: 10.25080/Majora-92bf1922-00a.
[22] Thomas A Caswell et al., matplotlib/matplotlib: REL: v3.2.2. Zenodo, 2020.

DECENTRALIZING SMART ENERGY MARKETS - TAMPER-PROOF DOCUMENTATION OF FLEXIBILITY MARKET PROCESSES

Zeiselmair, Andreas[1]; Guse, Miguel[1]; Yahya, Muhammad[2]; Förster, Felix[2]; Okwuibe, Godwin[2]; Birgit Haller[3]

[1] Forschungsstelle für Energiewirtschaft e.V., Am Blütenanger 71, 80995 München, www.ffe.de, azeiselmair@ffe.de

[2] OLI Systems GmbH, Silberburgstr. 112, 70176 Stuttgart, www.my-oli.com

[3] Dr. Langniß - Energie & Analyse, Silberburgstr. 112, 70176 Stuttgart, www.energieanalyse.net

The evolving granularity and structural decentralization of the energy system leads to a need for new tools for the efficient operation of electricity grids. Local Flexibility Markets (or "Smart Markets") provide platform concepts for market based congestion management. In this context there is a distinct need for a secure, reliable and tamper-resistant market design which requires transparent and independent monitoring of platform operation. Within the following paper different concepts for blockchain-based documentation of relevant processes on the proposed market platform are described. On this basis potential technical realizations are discussed. Finally, the implementation of one setup using Merkle tree operations is presented by using open source libraries.

Das Energiesystem ist zunehmend geprägt von steigender dezentraler Erzeugung und kleinteiligen Strukturen, welche neue Herausforderungen für einen effizienten Netzbetrieb schaffen. Daher werden neue Werkzeuge, sog. Flexibilitätsmärkte benötigt, die Plattform-basiert marktbasiertes Engpassmanagement bereitstellen können. In diesem Kontext ist es notwendig ein sicheres, zuverlässiges und manipulationsresistentes Marktdesign zu gewährleisten. Daher ist eine transparente und unabhängige Überwachung des Plattformbetriebs notwendig. Im folgenden Beitrag werden verschiedene Konzepte zur Blockchain-basierten Dokumentation relevanter Prozesse auf der vorgeschlagenen Marktplattform beschrieben. Auf dieser Grundlage werden mögliche technische Umsetzungsvarianten diskutiert. Abschließend wird die Implementierung einer Variante unter Verwendung von Merkle tree Operationen anhand von Open-Source-Bibliotheken vorgestellt.

1. Introduction

The energy system is already subject to fundamental change. Increasing penetration of renewable energy combined with an increased electrification of the heat and mobility sector leads to new challenges. Finally, these aspects lead to stress on the grid infrastructure. The evolving granularity and decentralization by the growing number of units and actors makes new coordination tools necessary. So called “Local Flexibility Markets” or "Smart Markets" are platform concepts currently under development in order to efficiently operate the electricity grid. [1] [2] As their main goal is to provide new approaches for market-based congestion management there is a distinct need for reliability but also tamper-resistance, so transparency and monitoring of correct platform operation is needed. Historical incidences like [3] but also current discussions in this field of research (see [4] or [5]) prove this need. The status quo of market monitoring through authorities is mainly report-based (i.e. “EU Regulation on wholesale Energy Market Integrity and Transparency”, REMIT) making it necessary for each market participant to provide all transaction and fundamental data.

Blockchain provides specific value propositions that could cover some of these needs and provide a more automated approach. On the one hand system-inherent data integrity through tamper-proof, time-specific documentation can increase trust to these newly created platforms. On the other hand, it can provide transparency through traceability of processes. Nevertheless, it also holds drawbacks regarding privacy protection and limited scalability depending on the actual setup. Therefore, different design configurations need to be assessed for specific use cases. [6] [7] [8]

2. Smart Markets

Developing a digitalized energy system providing data and controlling flexible energy units of prosumers by Smart Meters as well as measuring the physical network state by Smart Grid technologies is already an ongoing process [9] [10]. The consequential continuation to these finally provides the possibility to establish Smart Markets in order to coordinate and allocate the available flexibility to the needs of the grid [11]. Flexibility therefore can be understood as the “technical ability of a unit to change its current and/or predicted power [P, Q]” [12] [13], [14]. These flexible energy units include for example power-to-heat, distributed energy resources or energy storage systems.

Flexibility therefore is also a commodity that can be traded. In contrast to (wholesale) electricity or balancing power markets, trading flexibility for grid relief has to consider the local component to it. Congestions manifest themselves in current overload or voltage limit violation at a specific grid point. Depending on the grid topology the loads within the network have an impact on the congested element. This fact makes them not only part of the problem but eventually also part of the solution as long as they can adjust their power consumption or generation and therefore offer their flexibility. The allocation or

matching of flexibility demand at a congested grid spot to the offered flexibility by relevant flexible energy assets therefore is done by the proposed Smart Market platform [15] [16]. Fig. 1 illustrates relevant interactions between demanders and providers of flexibility to the Smart Market platform.

Involved parties consist of flexibility providers, i.e. operators of flexible energy units that offer their flexibility to the Smart Market platform and flexibility demanders, i.e. grid operators that want to contract flexibility within their grid in order to solve (predicted) congestions. Further, there is the role of the platform operator that is responsible for the correct market processes and flexibility allocation. Finally, there is also the regulatory authority that needs to control and observe correct market operation. [17] [18]

Fig. 1: Interactions of relevant Smart Market platform users

As such a Smart Market platform – as soon as in productive state – involves ten thousands up to millions of active actors (e.g. flexibility providers [19]) uploading daily datasets of relevant size and containing sensitive information, the aspects of privacy and scalability play a major role in platform design.

3. Decentralization Potentials

The decentralized character and ongoing trend in the development of the energy system involving increasing numbers of participants but also the local component of flexibility demand and provision on Smart Markets raises the question of also decentralizing the corresponding platform architecture. Taking a closer look reveals that there are three different dimensions of platform decentralization. **Operational decentralization** refers to the organizational operation of the platform. This includes both the provision of the necessary hardware and the allocation of responsibilities within the network. On the other hand **structural decentralization** addresses the platform structure itself, which is designed, for example, according to regional effectiveness, limited or defined reach or target groups. **Technical decentralization** is aimed at the actual implementation and realization of the platform or of individual functions of the platform. Different options exist starting from a jointly operated platform to complete decentralization without the need of an intermediary, e.g. using distributed ledger technology (DLT). As there is not per se an inherent value in technical decentralization it is necessary to take a closer look at potential added values provided to relevant functions, processes and finally stakeholders' needs.

4. Platform Environment incl. User Stories of Involved Parties

As part of decentralized applications, DLT in general or blockchain technology in specific aim to replace or support traditional, centralized databases, promising transparency, tamper-resistance and a high degree of availability [20], [21]. Regarding a Smart Market, this could finally lead to increasing credibility to the platform and therefore be a competitive advantage compared to alternative platform designs. Therefore, the following user stories of potential parties involved were identified regarding their need for transparency and trust:

1. **Flexibility providers** want to ensure that their flexibility offers are considered correctly on the Smart Market platform. Their demand bids should be documented immutably and time discrete to avoid conflicts. The flexibility provider should only be able to see his own offers and if applicable, corresponding contraction.
2. The **grid operator** places flexibility demands and as such wants to ensure that its demand bids are considered correctly on the market platform. The demand bids should be documented immutably and time discrete to avoid conflicts. The grid operator should only see his own demand bids, as well as (anonymized) allocated flexibility offers.
3. The **platform operator** receives flexibility demand and offers and conducts the matching algorithm. It wants to provide transparency to users by proving the correctness of registered demand and offers as well as to fulfill its reporting duties to certain authorities.
4. The **Federal Network Agency** (regulatory authority supervising electricity market) needs to ensure the correct function of the market [22]. It wants to check that all flexibility offers are considered without discrimination. Thus, it needs to be able to inspect all in- and output data (in pseudonymized form), results and version of the matching algorithm provided by the platform operator to spot-check on request.

On top of these stakeholder perspectives there are external requirements evolving from legal and regulatory frameworks. Besides safe, efficient and trusted operation one very relevant aspect is compliance of GDPR-related data privacy.

5. Concepts for Documentation of relevant Processes

In order to cover the identified needs for transparency and data security there is another challenge regarding the initial proof of correct data input. Data can be stored very securely on a blockchain but there is no impact to the correct provision of data. Especially (but not only) in the energy sector this fact shows a fundamental problem in realizing feasible end-to-end use cases. Input sources can be manifold including:

- Measurement gear that need to provide trustable sensor data to the blockchain.
- User interaction, i.e. data input coming from a user interface, e.g. providing an offer bid to a Smart Market
- External data sources, like information from third parties, e.g. weather prognosis data to a Smart Market platform.
- Computational results, i.e. solving complex problems that need to be computed off-chain, i.e. the allocation optimization of a Smart Market considering a high number of bids including constraints.

Nevertheless, there are already different approaches available to address the challenge of trusted data provision.

The most obvious approach is to regulate technical connections and the data providers themselves by a central authority. In the energy sector, available standardized and secure Smart Metering infrastructure including trusted metering point operators regulated by the Federal Network Agency and the Federal Cyber Security Authority provides a certain advantage and trust compared to other sectors [23].

A second one is to provide the possibility of checking the validity by each single user. This can be done by redundant offline storage of user-specific data and ex-post verification. This approach will be further evaluated in the following chapter. [24]

A third option is to enable different, redundant pathways to the blockchain and using consensus oracle operations as well as verifiable multi-party computation to validate the correctness of data provision [25].

Zero-Knowledge-Proofs are possibly the most elegant way of providing trusted data and especially correctly computed results without revealing all input data [26] [27]. Nevertheless, currently there are still limitations regarding scalability.

Finally, the correct application of these approaches needs to be decided on a use-case-specific point of view. Applied to the Smart Market platform an appropriate validation process could be considered in the following platform steps:

1. Provision of basic operational platform data (e.g. grid topology, boundary conditions, market area)
2. Provision of flexibility demand (by the grid operator)
3. Provision of flexibility offers (by operators of flexible energy units)
4. Optimization and provision of allocation results (through the platform-operator(s))
5. Proof of flexibility provision (through measurement data from Smart Meters)
6. Settlement information (provision of billing and payment information)
7. Revision-safe documentation

Within this paper the focus was put on the validation of flexibility offer bids (step 3) which was also realized in a proof-of-concept (see chapter 7). Besides this, the proposed setup is also applicable for steps 1 and 2.

6. Evaluation of Data Storage and Validation Options

Blockchain platforms like Ethereum provide the possibility of storing any type of data through the use of smart contracts [21]. As illustrated in Fig. 2, data can be stored openly as “plain text” within a smart contract transaction.

Storing all application data on a blockchain comes with limitations, mainly regarding scalability and data privacy. In general, scalability of blockchains is limited in terms of storage capacity and throughput. Furthermore, the cost of storage is high [28]. Current developments such as alternative consensus mechanisms, sharding or state channels aim to solve the scalability issue, but still have overhead compared to traditional databases [29], [30], [31].

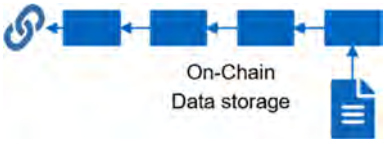


Fig. 2: On-Chain data storage

Storing private data is especially a problem on public blockchains, where data are openly accessible to anyone. Approaches to preserve confidentiality on blockchains include the use of private networks or encryption of stored data. Because encryption algorithms are susceptible to future vulnerabilities, it is questionable whether public storage of encrypted private data is compliant with regulations such as the EU’s GDPR. In addition, GDPR compliant data privacy also requires the possibility of erasing data upon request, which conflicts with the immutability of data stored on a blockchain. [32]

Considering these limitations, an alternative is to store only data hashes on-chain and storing data themselves off-chain. [33] This approach is illustrated in Fig. 3. The integrity of off-chain data can then be proven using the on-chain hash. Due to the constant length of a hash, this approach requires less storage capacity on-chain, improving scalability. The pre-image resistance of a hash function prevents private data to be inferred from its hash and thus

provides the required confidentiality [33]. As the data are stored off-chain it is also possible to erase them upon request, improving data sovereignty. Nevertheless, this approach sacrifices the blockchains improved availability and limits transparency, as data themselves are still provided off-chain.

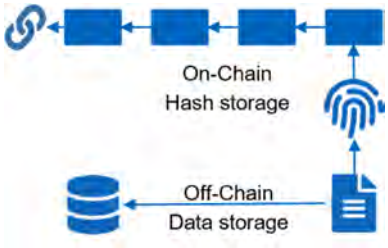


Fig. 3: On-Chain hash storage

In order to evaluate different options, relevant criteria range from privacy, scalability, accessibility, availability, data sovereignty to transaction costs depending on data volumes, as contrasted in table 1.

Table 1: Comparison of different data storage options

On-Chain storage	Plain data	Encrypted data	Hash
Scalability	×	×	✓
Privacy	×	○	✓
Data sovereignty	×	○	✓
Low transaction costs	×	×	✓
Decentralized availability	✓	✓	×
Full Transparency	✓	○	×

In order to choose a suitable approach, the requirements for the documentation of the Smart Market processes were analyzed, yielding the following results:

- Functional requirements: The data storage option must offer enough storage capacity to document the entire process and enough throughput to document it in time.
- Non-Functional requirements: Because the Smart Market also processes private data, data should be modifiable, erasable and stored confidentially, in order to comply with the GDPR. In addition, inspections by the Federal Network Agency require process data to be traceable and secured against manipulation.

Because of the scalability and privacy requirements, storing data on-chain is not an option for documentation of Smart Market processes. For this reason, the hash storage approach has been further investigated.

As an additional measure to improve scalability, process documentation data can be gathered to create a Merkle tree, resulting in a single root hash and thus less storage capacity required on the blockchain. Due to the Merkle tree's properties, this root hash alone is enough to verify the integrity of individual data entries. As mentioned in section 5, a key issue of using a blockchain for tamper-proof process documentation is to assure the correct provision of data to the blockchain. In the bidding process input data are user-provided and as such the correctness of data is determined by the user. The most efficient use of a Merkle tree structure, is to gather all input data, which however is only possible for the platform operator and not a single user. As a consequence, three different options for creating a Merkle tree and storing its root hash on a blockchain have been identified.

In the first option, illustrated in figure 4a), all Smart Market user input data for a given time frame are collected by the platform operator and then gathered to create a Merkle tree. The platform operator then stores the root hash of this Merkle tree on the Blockchain, leaving users the ability to validate the integrity of their input ex-post. With this option however, market regulators can only verify whether data supplied by the platform operator have not been manipulated since the Merkle trees creation. It is not possible to check if supplied input data are correct from a user's perspective.

In the second option, illustrated in figure 4b), platform users store their input data hash on the blockchain themselves, ensuring the correctness of the hash. Input data are supplied to the platform separately. Previous input data hashes, that are already stored on the blockchain, can be combined by the user with its own hash to create a Merkle Tree. The resulting root hash of this Merkle Tree can be stored on the blockchain by the user. This way, one root hash needs to be stored on the blockchain for each user input. Therefore, this option is less scalable as the number of transactions on-chain increases with the number of platform users.

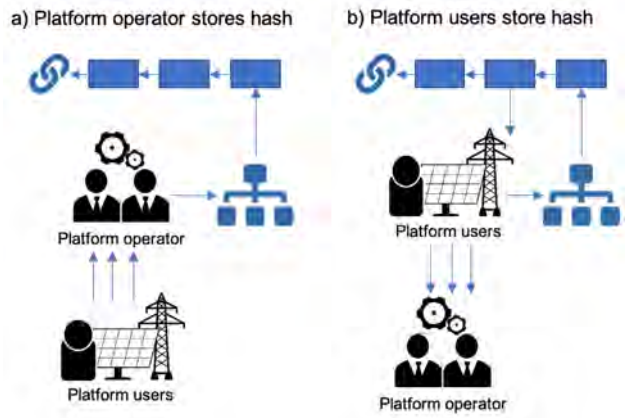


Fig. 4 a), b): Hash storage by platform operator and user

The third option, illustrated in figure 4 c), brings together both benefits of the previous options. All user input data for a given timeframe are gathered by the platform operator to create a Merkle Tree. The platform operator submits the root hash to a smart contract and requests users to verify the correctness of the root hash. A majority of users then need to sign the transaction using a multi-party consensus to

ensure its correctness, before for the smart contract stores it on the blockchain.

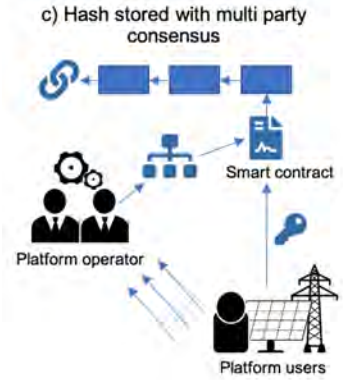


Fig. 4 c): Hash storage with multi-party consensus

While this option is scalable through its Merkle tree use and provides a check for correctness, it requires the availability of users for the consensus process. Difficulties arise from situations, where no majority consensus can be achieved or when users find their input data to have been manipulated.

Finally, the first option was chosen for the implementation described in the following section as it offers the benefits of scalability, while being more user-friendly as it requires less user interaction.

Regardless of what option would be chosen, once the root hash of a Merkle tree is stored on the blockchain it can be used for validating the integrity of data. Assuming the correctness of data used for the construction of the root hash stored on the blockchain, any data provided by the platform at a later moment can be considered untampered with, if they can be used to reconstruct an identical root hash. In the case of the aforementioned first option of storing a root hash on the blockchain, the correctness of input data used by the platform can be validated by the user ex-post. The user does this by receiving a list of hashes by the platform, which together with the user's own input data can be used to locally reconstruct the Merkle tree's root hash. If this local root hash matches the one stored on the blockchain, this proves that the user's input data have been considered correctly by the platform.

7. Implementation of Merkle tree based Proof-of-Concept

Based on the chosen concept described in the previous chapter, a proof-of-concept in the form of a prototype has been implemented for the process of provisioning flexibility offers.

In the current implementation of the ALF Smart Market, the users upload their flexibility offer as a .csv file via a dashboard on a publicly accessible domain after successful registration. The market users submit their offer one day before the actual activation of the flexibility. After the offers have been collected and the gate closure time has passed, the market operator will calculate the market result. This determines which flexibilities are being activated later on and ultimatively results in money flows.

The additional layer that is now being added to the provision process is using the blockchain along with Merkle trees. The implemented concept is illustrated in Fig. 5. In the first step, user place their offers and submit them to the platform. While the market operator is storing the data on its side, the users themselves are also creating and storing hashes corresponding to their offers locally. This technical redundancy is later used to execute the proof.

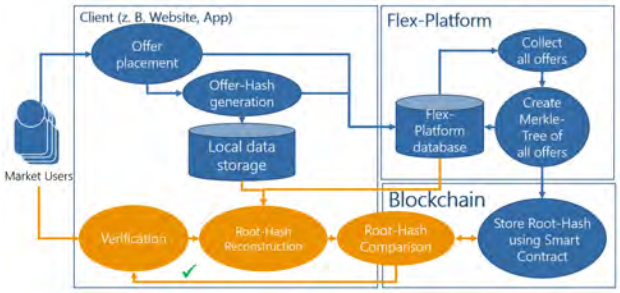


Fig. 5: Verifiable Offer Placement on Flex-Platform

After all offers have arrived at the platform, their corresponding hashes are calculated, and used to create a Merkle tree and its respective root hash. This root hash is then stored in a smart contract on the blockchain by the market operator.

Later on, in case the user wants to verify if the offer has been included correctly by the market operator, the user can request all necessary leaves for recreating the Merkle tree root hash from the platform. Using these leaves, the user can recreate a local root hash with the locally stored offer hash on the client-side. The local result can be compared with the root hash that has been stored on-chain by the market operator. This process can have the following outcomes:

- In case the root hashes match, the market operator has correctly received, included and not tampered with the offer from the user.
- In case the root hashes do not match, further investigation is required.

In theory, the approach above could be used in a diverse set of circumstances and also in other commercial sectors. In market processes, where some sort of bidding, offering or tendering involved and the market operator wants to obtain and retain a certain level of credit of trust, the operator might use this option.

As explained before, this process has been implemented as an actual technical prototype inside a standalone application. The prototype can be divided into the following components:

- **Server-side functions**, that are run at the premises or the cloud of the market operator
- **Client-side functions**, that are being executed by the browser locally on the device of the market users
- **Blockchain functions** implemented as Smart Contracts to hold the root hashes, that are being

stored by the market operator and read out by the market users

On the **client-side**, a web-application is implemented with the *vue.js* Framework. The frontend is a dashboard containing all the necessary functionality. It can be accessed via a publicly available URL. Important libraries in the web-application are:

- *Web3.js* in order to specifiy the contract address and ABI[1] of the Smart Contract. Also, web3.js features a toolset that helps connecting to a RPC-Endpoint[2] or node in order to write to or read from Smart Contracts on a designated blockchain.
- *Merkletree.js* that features a toolset to create and interact with Merkle trees and corresponding objects and attributes

Fig. 6 shows the **Create Offer Screen** which enables the user to send their offer as a .csv to the market operator. The authentication in the context of the prototype is being done with a simple username. Also, the user can select the market date, which is later used to identify which hashes from which day are supposed to be requested and compared.

The .csv file is sent to the market operator in stringified form via HTTP and a REST-API on the server.

In parallel, the hash is also created with merkletree.js and the corresponding market date is also stored in the browser storage locally for later use.

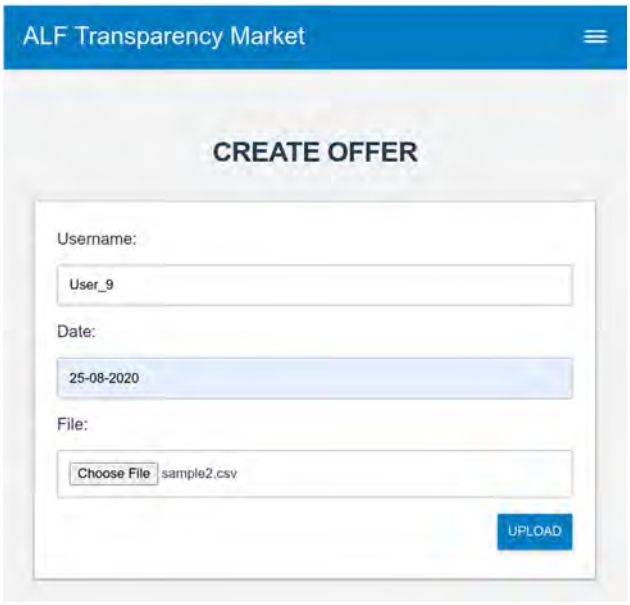


Fig. 6: Create Offer Screen for the market user

Besides the **Create Offer Screen**, there is also a need for displaying the local hashes from the browser storage. The **Show Hashes Screen** (see Fig. 7) queries the browser storage for the entries.

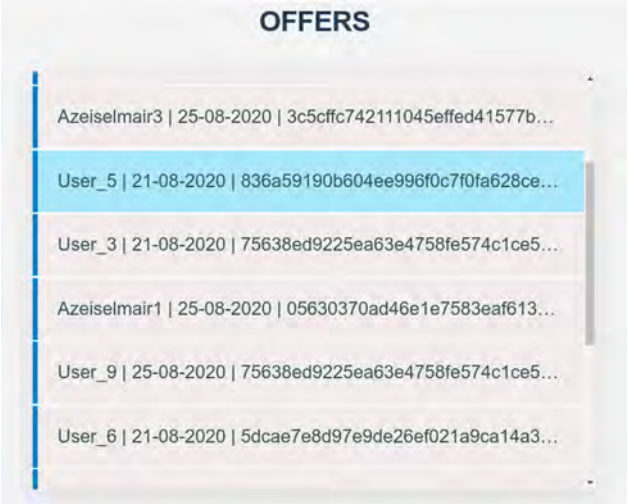


Fig. 7: Show Hashes Screen for the market user

Finally, there is also the **Verification Screen** (Fig. 8). Here, the user inputs their username and the market date. The underlying scripts will query the local hash and also request from the server, again via the REST-API, all the necessary leaves from the server in order to recalculate the hash locally. The root hash for that market date is also queried from a blockchain node and its RPC-endpoint. In case the two hashes match, a simple to understand traffic light will show the color green. In case there has been an undetermined error in the process, it will light up yellow. And in case there is a clear mismatch between the hash on the blockchain and the one created locally, the traffic light will show the color red. This concept is supposed to abstract the rather complicated processes in the background and give the user an easy-to-understand indication on the result.

Fig. 8: Verification Screen for the market user

On the **market operator (server) side**, additional functions also have been implemented. A microservice using *node.js* as its engine is placed close to the other services that are being run by the market operator. Multiple scripts, also using the merkletree.js library, not only create the hashes, Merkle trees and root hashes from the many offers that are being received, but also provide the necessary leaves back to the users. Therefore, a database and the aforementioned *REST-API* are needed. For the prototype, we decided on *MongoDB*

[1] Application Binary Interface

[2] Remote Procedure Call

in combination with *Express.js* to operate the API, endpoint and database.

Web3.js, as realized on the client-side, is used to communicate with the blockchain. As the platform operator also has to send actual transactions to the blockchain, a key-pair with enough tokens to execute the transaction is needed.

The last and central component is the **blockchain**, which is being used in this context as a tamper-proof database that also enforces consensus. For the prototype the Volta test-blockchain was used. A public blockchain with proof-of-authority consensus algorithm (Parity Aura) operated by the Energy Web Foundation [34]. The reasons for this are:

- Only root hashes are being stored on-chain, no personal information is being put into the public domain. Therefore, the advantages of public networks can be used in full.
- PoA uses the existing extended hierarchy in the energy-sector to reduce the energy consumption in comparison to proof-of-work blockchains by orders of magnitudes. [35]

After creating a key-pair and receiving the necessary token through a faucet, the smart contract *ALFtransparency.sol* was deployed with the Truffle Framework on Volta via an open Ethereum Node operated by OLI Systems. The Contract accepts the root hashes only from the key-pair that originally deployed it. It also takes care of additional safety measures like timestamp creation on-chain.

Via the *web3.js* library, both the client-side and server-side functions can interact with the contract through an RPC, although only the market operator can successfully send transaction to it.

The whole code is open source and publicly available at *https://github.com/olisystems/alf-transparency*.

8. Critical Review and Outlook

Several blockchain-based options were analyzed as possible concepts for tamper-proof documentation of Smart Market processes with the aim of providing increased trust for platform user as well as transparency to regulatory authorities. Scalability and privacy were identified as key issues. Finally, one approach combining on- and off-chain storage using Merkle tree hashes turned out to be the most promising option, providing scalability while preserving GDPR compliant data protection. Within a proof-of-concept this approach was realized using open-source libraries including *merkletree.js*, *web3.js*, the *vue.js* frontend framework. Also, the Volta test-network from the Energy Web Foundation was chosen as the blockchain component.

Besides the achieved value propositions the following options for improvement and need for further research were identified:

- The correctness of documented data can only be verified ex-post by the users. Therefore, regulatory authorities depend on users' validation to prove the correctness of data. Data still needs to be provided by the platform operator. Already proposed possibilities of data provision using multi-party consensus could provide additional security, but further research on reducing the need for user interaction is required.
- The implementation still considers a centralized approach involving the platform operator as intermediary. Full decentralization would require a secure, scalable and privacy-preserving method for distributed computation. Current research includes the use of zero knowledge proofs or multi computation approaches solving increasingly complex calculations.
- Blockchains and their use for documentation still have to be approved by regulatory authorities as trusted resources. Therefore, further proof-of-concepts and research projects have to prove the applicability.
- Within the proposed implementation, usability was always in focus. In order to reach a productive system further automation needs to be provided.
- A detailed evaluation of synergies to other energy platforms (including smart metering infrastructure) needs to be conducted in order to reach the state of an energy business ecosystem.

Acknowledgements

The presented works are part of the project C/sells funded by the Federal Ministry of Economics and Energy (BMWi) as part of the "Schaufenster intelligente Energie - Digitale Agenda für die Energiewende" (SINTEG) funding program (funding code: 03SIN121).

References

[1] Ropenus, Stephanie: Smart-Market-Design in deutschen Verteilnetzen. Berlin: Agora Energiewende, 2017

[2] Radecke, Julia et al.: Markets for Local Flexibility in Distribution Networks - A Review of European Proposals for Market-based Congestion Management in Smart Grids. Berlin: Hertie School of Governance, 2019.

[3] Weare, Christopher: The California Electricity Crisis: Causes and Policy Options. San Francisco, California: Public Policy Institute of California, 2003.

[4] Hirth, Lion et al.: Market-Based Redispatch in Zonal Electricity Markets - Inc-Dec Gaming as a Consequence of Inconsistent Power Market Design (not Market Power). Berlin: Neon Neue Energieökonomik GmbH (Neon), 2019.

[5] Zeiselmair, Andreas et al.: Market power assessment in regional smart markets. In: 17th International Conference on the European Energy Market. Stockholm: Forschungsstelle für Energiewirtschaft e.V., 2020.

[6] Bogensperger, Alexander; Zeiselmair, Andreas, Hinterstocker, Michael: Die Blockchain-Technologie - Chance zur Transformation der Energieversorgung? - Berichtsteil Technologiebeschreibung. München: Forschungsstelle für Energiewirtschaft e.V. (FfE), 2018.

[7] Bogensperger, Alexander; Zeiselmair, Andreas; Hinterstocker, Michael; Dufter, Christa: Die Blockchain-Technologie - Chance zur Transformation der Energiewirtschaft? - Berichtsteil: Anwendungsfälle. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[8] Strüker, Jens et al.: Blockchain in der Energiewirtschaft - Potenziale für Energieversorger. Berlin: Bundesverband der Energie- und Wasserwirtschaft e.V. (BDEW), 2017.

[9] Gesetz zur Digitalisierung der Energiewende. Ausgefertigt am 2016-08-29; Berlin: BMWi, 2016.

[10] Bogensperger, Alexander; Estermann, Thomas; Samweber, Florian; Köppl, Simon; Müller, Mathias; Zeiselmair, Andreas, Wohlschlager, Daniela: Smart Meter - Umfeld, Technik, Mehrwert. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[11] Ramos, Ariana et al.: Realizing the smart grid's potential: Defining local markets for flexibility. In: Utilities Policy Vol. 40, p. 26 - 35. Michigan: Katholieke Universiteit Leuven, 2016.

[12] Müller, Mathias et al.: Dezentrale Flexibilität für lokale Netzdienstleistungen - Eine Einordnung des Flexibilitätsbegriffs als Grundlage für die Konzipierung einer Flexibilitätsplattform in C/sells. In: BWK - Das Energie-Fachmagazin 6/2018. Düsseldorf: Verein Deutscher Ingenieure (VDI), 2018.

[13] Flexibilität im Stromversorgungssystem - Bestandsaufnahme, Hemmnisse und Ansätze zur verbesserten Erschließung von Flexibilität - Diskussionspapier Stand 03. April 2017. Bonn: Bundesnetzagentur, 2017

[14] E-Bridge Consulting GmbH: Sichere und effiziente Koordinierung von Flexibilitäten im Verteilnetz. Bonn: E-Bridge Consulting GmbH, 2017.

[15] Köppl, Simon et al.: Altdorfer Flexmarkt – Decentral flexibility for distribution networks. In: Internationaler ETG-Kongress 2019. Esslingen: VDE ETG, 2019.

[16] Zeiselmair, Andreas et al.: Netzdienlicher Handel als Element des zellulären Energiesystems am Beispiel des Altdorfer Flexmarkts (ALF) - 11. Internationale Energiewirtschaftstagung (IEWT). Wien: Technische Universität Wien, 2019.

[17] Zeiselmair, Andreas et al.: Altdorfer Flexmarkt (ALF) - Konzeptbeschreibung, Zielsetzung, Funktionsweise und Prozesse des Altdorfer Flexmarkts. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[18] Zeiselmair, Andreas et al.: Altdorfer Flexmarkt (ALF) - Use Case Beschreibung. München: Forschungsstelle für Energiewirtschaft e.V., 2020.

[19] Müller, Mathias et al.: Regionales Flexibilitäts-Potenzial dezentraler Anlagen - Modellierung und Bewertung des regionalen Flexibilitäts-Potenzials von dezentralen Flexibilitäts-Typen im Verteilnetz. Berlin: Conexio GmbH, 2019.

[20] Ali, Robleh et al.: Distributed Ledger Technology: beyond block chain. London: Government Office for Science, 2016.

[21] Buterin, Vitalik: A Next Generation Smart Contract and Decentralized Application Platform - Ethereum White Paper. Switzerland: Ethereum Foundation, 2014.

[22] VERORDNUNG (EU) Nr. 1227/2011 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 25. Oktober 2011 über die Integrität und Transparenz des Energiegroß-handelsmarkts. Brüssel: Europäisches Parlament und Rat, 2011

[23] Technische Richtlinie BSI TR-03109. Bonn: Bundesamt für Sicherheit in der Informationstechnik (BSI), 2015.

[24] Eberhardt, Jacob et al.: On or Off the Blockchain? Insights on Off-Chaining Computation and Data. Berlin: TU Berlin, 2017.

[25] Aitzhan, Nurzhan et al.: Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. Abu Dhabi: Masdar Institute of Science and Technology, 2016.

[26] Hasan, Jahid: Overview and Applications of Zero Knowledge Proof (ZKP). Nanjing: Nanjing University of Posts and Telecommunications, 2019.

[27] Ben-Sasson, Eli et al.: Scalable, transparent, and post-quantum secure computational integrity. Haifa, Israel: Zerocash, 2018.

[28] Mazlan, Ahmad et al.: Scalability Challenges in Healthcare BlockchainSystem—A Systematic Review. Kuala Lumpur: Universiti Teknologi Malaysia, 2020.

[29] Buterin, Vitalik: A Proof of Stake Design Philosophy. In: https://medium.com/@Vitalik Buterin/a-proof-of-stake-design-philosophy-506585978d51. (Abruf am 2018-01-24); (Archived by WebCite® at http://www.webcitation.org/6whn5uuin); San Francisco,

[30] Zamyatin, Alexei: On sharding blockchains. In: https://github.com/ethereum/wiki/wiki/Sharding-FAQ. (Abruf am 2018-01-01); Zug, Switzerland: Ethereum Foundation, 2017.

[31] Poon, Joseph et al.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. San Francisco: Lightning Network, 2016.

[32] Eichler, Natalie et al.: Blockchain, data protection, and the GDPR - Positionspapier. Berlin: Blockchain Bundesverband e. V., 2018.

[33] Rogaway, Phillip et al.: Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. Davis, CA: Dept. of Computer Science, University of California, 2004.

[34] Hartnett, Sam et al.: The Energy Web Chain - Accelerating the Energy Transition with an Open-Source, Decentralized Blockchain Platform. Energy Web Foundation, 2019.

[35] Sedlmeir, Johannes et al.: The Energy Consumption of Blockchain Technology: Beyond Myth. Bayreuth: Project Group Business and Information Systems Engineering of the Fraunhofer FIT, 2020.

ISSN 1437-7624