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).