Blockchain (Bitcoin) - Skalierbarkeit und Sicherheit

Thomas Mühlegger - 1223504
Rene Thaler - 1220587

Universität Salzburg
Computerwissenschaften
Jakob-Haringer-Str. 2
5020 Salzburg, Österreich

Projektbetreuer:
Prof. Dipl.-Ing. Dr. Wolfgang PREE

28. Juli 2016

Abstract: In der heutigen Zeit, in der Computer und Handys fast schon unseren Alltag für uns regeln, gewinnen auch Kryptowährungen immer mehr an Bedeutung. Die bekannteste Währung in diesem Bereich ist der Bitcoin. Dieses Paper behandelt zum einen Aspekte zum Thema Sicherheit des Bitcoin Protokolls, zum anderen werden mit Bitcoin-NG und dem Bitcoin Lightning Network zwei Ansätze betrachtet, die die Skalierbarkeit des Bitcoin Protokolls deutlich verbessern können.

Inhaltsverzeichnis

1 Einleitung

Bitcoin ist eine verteilte, dezentrale Kryptowährung, welche von Satoshi Nakamoto, dessen Identität bis dato nicht geklärt ist, im Paper Bitcoin: A Peer-to-Peer Electronic Cash System [1] initial definiert und implementiert wurde. Bitcoin verwendet das Blockchain Protokoll um die Transaktionen des Netzwerks zu serialisieren, sowie dadurch die Kontostände der einzelnen Nutzer zu verwalten. Sogenannte Schürfer (Miner) bestätigen Transaktionen, indem sie diese an das globale Register, die Blockchain, anfügen. Somit kann ein Nutzer einen Betrag nicht mehrfach versenden und nur der richtige Empfänger über diesen verfügen. Der Ursprungsblock am Anfang der Blockchain wurde im Protokoll definiert. Weitere Blöcke enthalten eine eindeutige ID und die ID des vorhergehenden Blockes. Ein gültiger Block enthält die Lösung eines Kryptographieproblems, den Hash des vorhergehenden Blocks, sowie den Hash der gültigen Transaktionen im aktuellen Block. Der Aufwand der Schürfer wird durch eine weitere Transaktion im Block, die sogenannte Coinbase, entlohnt, deren Empfänger sie selbst definieren dürfen. Weiters erhält der Schürfer die Transaktionsgebühren, deren Höhe die Nutzer bei Tätigen einer Transaktion selber festlegen können. Dieser Bestätigungsvorgang, also das Lösen des Kryptopuzzles, wird Block Mining genannt. Die Schwierigkeit dieses Puzzles wird durch das Protokoll dynamisch so verändert, dass im Schnitt alle zehn Minuten ein Block an die Blockchain angefügt wird.

Lösen Schürfer gleichzeitig das Kryptopuzzle und erzeugen somit einen gültigen Block, so können sie diesen an die Blockchain anfügen und es entstehen Verzweigungen. Im Bitcoin Protokoll ist definiert, dass die Schürfer neue Blöcke an die längste Kette von Blöcken anfügen sollen. Die längste Kette ist gültig, während andere, kürzere Zweige verworfen werden. Transaktionen von verworfenen Blöcken können wieder in Blöcke des Hauptzweiges inkludiert werden, insofern nicht bereits andere Transaktionen den Betrag des Kontos verbraucht haben. Durch das lange Schürfintervall von zehn Minuten ist eine Verzweigung relativ selten und tritt nur etwa alle 60 Blöcke auf.

Auf den folgenden Seiten werden die Aspekte Sicherheit und Skalierbarkeit des Bitcoin-Protokolls näher beleuchtet.

2 Sicherheit von Bitcoin:

2.1 Allgemeines

Im Allgemeinen ist zu berücksichtigen, dass sämtliche Cryptowährungen gegen herkömmliche Angriffe keinen Schutz bieten. Dies gilt aber ebenfalls für "vertrauenswürdige"Zahlungsmethoden, wie z.B. das Netbanking der Sparkasse. Ist ein Rechner eines Nutzers infiziert, haben Angreifer durchaus die Möglichkeit, Passwörter zu klauen oder Transaktionen von fremden Benutzerkonten durchzuführen. Ein sicherer Clientrechner ist somit eine zwingend notwendige Voraussetzung für das sichere Abwickeln von Geldtransaktionen.

Weiters ist das Bitcoin-Protokoll nicht anonym und Transaktionen können nicht rückgängig gemacht werden. Daher ist es wichtig, dass man Zahlungen nur an vertrauenswürdige Partner tätigt, denn immer öfter wird der Bitcoin mit illegalen Geschäften, wie Drogen bzw. Waffenhandel, in Verbindung gebracht.

2.2 Bekannte Angriffe

Folgende Angriffe [2] auf das Bitcoin-Netzwerk sind bekannt:

- **Illegales Botnetz zum Minen:** Zwei Deutsche entwickelten gemeinsam einen Trojaner, der die infizierten Rechner zu einem Botnetz zusammenschloss. Die Rechenleistung dieses Botnetzes wurde für das Minen von Bitcoins verwendet. Die Täter flogen auf und es wurden Bitcoins im Wert von 700.000 € beschlagnahmt.

- **Scheinfirma aus China:** Die chinesische Handelsplattform GBL mit 1.000 Investoren lockte Benutzer an und bezahlte die Anteile der Nutzer nie aus. Die Betreiber der Plattform tauchten unter, wurden aber später gefasst. Die chinesische Regierung verbot daraufhin den Banken im Land den Handel mit Bitcoins.

- **Lösegeldtrojaner:** Die Entwickler so genannter Lösegeldtrojaner fordern Lösegeld in Form von Bitcoins. Der Trojaner verschlüsselt private Daten am Rechner und entsperrt diese erst bei Zahlungseingang des geforderten Betrages.

Die oben geschilderten Szenarios sind aber keine Angriffe auf das Bitcoin-Protokoll an sich, sondern auf die Rechner von Benutzern. Da derzeit also keine Angriffe auf das Bitcoin-Protokoll bekannt sind, ist die Währung grundsätzlich als sicher einzustufen. Deshalb werden im Folgenden mögliche Angriffe auf das Protokoll vorgestellt, die eine Sicherheitslücke in dessen Cryptoprimitiven voraussetzt.

2.3 Verwendete Kryptoprimitiven

2.3.1 Hashfunktionen

Eine Hashfunktion $h(x) = y$ ist eine Funktion, die eine Eingabe beliebiger Länge x auf eine Ausgabe fixer Länge y abbildet. Unter schwierig versteht man im Folgenden, dass es eigentlich unmöglich ist, in vertretbarer Zeit eine Lösung zu finden. Eine sichere Hashfunktion sollte also folgende Eigenschaften aufweisen:

- **(First)Pre-Image Resistent:**
  Es ist schwierig ein x zu finden, sodass $h(x) = y$.

- **Second Pre-Image Resistent:**
  Bei existierendem x_1 ist es schwierig ein $x_2 \neq x_1$ zu finden, sodass $h(x_1) = h(x_2)$.

- **Kollisision Resistent:**
  Es ist schwierig ein $x_2 \neq x_1$ zu finden, sodass $h(x_1) = h(x_2)$.

Main Hash - $H_M(x) = SHA256(SHA256(x))$
Die Main Hashfunktion [3] wird beim Bitcoin-Protokoll für das Minen verwendet. Sie basiert auf dem SHA256 Algorithmus und hat einen Output von 256 Bit. Diese Hashfunktion ist dafür zuständig, dass ein Miner einen entsprechenden Rechenaufwand aufbringen muss, um die Gültigkeit einer Transaktion zu bestätigen und somit die Transaktionsgebühr einzuheben $\Rightarrow$ Proof-of-Work. Für eine gültige Lösung muss der Miner ein x berechnen, sodass $H_M(x) < vorgegebnenenHashwert$.

Address Hash - $H_A(x) = RIPEMD160(SHA256(x))$
Die Address Hashfunktion wird für das Pay-To-Public-Key-Hash und das Pay-To-Script-Hash verwendet. Sie basiert auf dem RIPEMD160 Algorithmus sowie auf dem SHA256 Algorithmus und hat einen Output von 160 Bit.

2.3.2 Elliptic Curve Digital Signature Algorithm (ECDSA)

Der Elliptic Curve Digital Signature Algorithm ist ein Algorithmus für das sichere Signieren des Main Hash H_M von Transaktionen und für kritische Alertnachrichten von den Entwicklern des Protokolls. Genauer gesagt benötigt eine Transaktion mit i Eingängen und o Ausgängen eine unterschiedliche Signature für jeden Eingang. Folgende Eigenschaften sollte ein Algorithmus für das erstellen von digitalen Signaturen mit sich bringen.

- **Fälschungssicherheit:**
  Eine Nachricht m die vom Public Key p bestätigt wurde, kann nicht ohne den Secret Key s unterschrieben werden.

- **Integrität:**
  Eine gültige Unterschrift m_s bestätigt keine andere Nachricht $m' \neq m$.

- **Nachweisbarkeit:**
  Eine gültige Unterschrift m_s bestätigt keinen anderen Public Key $p' \neq p$.

2.4 Mögliche Angriffsszenarien

2.4.1 Szenario 1 - Kombination Address Hash und Unterschrift

Fälschungssicherheit - Ist beispielsweise die Fälschungssicherheit der digitalen Signatur nicht gegeben, dann könnten in Kombination mit einer Lösung von First bzw. Second Pre-Image der Address Hashfunktion, alle unverbrauchten Bitcoins gestohlen werden. Außerdem kann eine Zahlungstransaktion mit einer beliebigen Adresse getätigt und diese erfolgreich unterschrieben werden.

Integrität - Die Integrität des Signaturalgorithmus hat im Zusammenhang mit dem Address Hash keine Auswirkung, da die Nachrichten für Alerts und Transaktionen den Address Hash nicht verwenden.

Nachweisbarkeit - Kann eine gültige Unterschrift m_s von einem Public Key p gefunden werden, sodass diese auch für p' gültig ist, dann können im Zusammenhang mit einer Lösung von Second Pre-Image bzw. Kollision der Address Hashfunktion, Transaktionen ersetzt werden.
Wenn p', sodass $H_A(p) = H_A(p')$ und die Signatur s, bestätigt $H_A(p)$, dann bestätigt s ebenfalls $H_A(p')$.

2.4.2 Szenario 2 - Kombination Main Hash und Unterschrift

Fälschungssicherheit - Ist der Signaturalgorithmus nicht Fälschungssicher, dann können im Zusammenhang mit einer Lösung von Second Pre-Image oder Kollision der Main Hashfunktion, Bitcoins zerstört oder unter gewissen Umständen sogar gestohlen werden.

Integrität - Gibt es eine Lösung von Second Pre-Image bzw. Kollision des Main Hashes, dann kann durch eine nicht geltende Integrität des Signaturalgorithmus ein Public Key p mit m und m' erzeugt werden, sodass die Unterschrift von $H_M(m)$ auch für $H_M(m')$ gültig ist. Es können also Transaktionen ersetzt und somit doppelt verwendet werden.

Nachweisbarkeit - In Kombination mit dem Main Hash, hat die Nachweisbarkeit des Signaturalgorithmus keine Auswirkungen.

2.4.3 Auswirkungen im Überblick

In Abb. 1 wird dargestellt, wie sich ein geknackter Signaturalgorithmus in Kombination mit einer Sicherheitslücke vom Main Hash bzw. Address Hash auswirken könnte. Wie bereits erwähnt, gibt es allerdings auf diese Primitiven noch keine bekannten Angriffe, darum sind diese Angriffe so derzeit noch nicht möglich.

	Signature Property		
Hash Property	**Selective forgery**	**Integrity break**	**Repudiation**
Address Hash (H_A)			
Collision	Repudiate transaction	-	Change existing payment
Second pre-image	Steal all coins	-	Change existing payment
Pre-image	Steal all coins	-	-
Main Hash (H_M)			
Collision	Steal coins	Steal coins	-
Second pre-image	Steal coins	Double spend	-
Pre-image	-	-	-

Abb. 1: Mögliche Folgen von gebrochenen Kryptoprimitiven

Da die aktuell verwendeten Algorithmen gegebenenfalls ersetzt werden können (siehe 2.5), zeigt die folgende Grafik (Abb. 2) die Auswirkungen von Angriffen, auf die konkreten Algorithmen der derzeitigen Bitcoinimplementierung.

Breakage	**Effect**
SHA256	
Collisions	Steal coins
Second pre-image	Double spend
Pre-image	Complete failure
RIPEMD160	
Any of the above	Repudiate payments
ECDSA	
Selective forgery	Steal coins, Send fake alerts
Integrity break	Claim payment not received
Repudiation	Send fake alerts

Abb. 2: Auswirkungen von gebrochenen Algorithmen

2.5 Maßnahmen für den Ernstfall

Im Falle eines erfolgreichen Angriffs, auf die Algorithmen der Bitcoinimplementierung existieren Maßnahmen, um im Ernstfall einschreiten zu können. Da ein Angriff auf den Signaturalgorithmus ohnehin nur in Kombination mit dem Address Hash bzw. Main Hash sinnvoll ist, wurde dagegen keine relevante Sicherheitsvorkehrung getroffen. Existiert ein Angriff auf den SHA256 Algorithmus, kann ein Hard Fork (Abspaltung der Blockchain) durchgeführt werden und der SHA256 Algorithmus durch einen sicheren Algorithmus ersetzt werden. Dafür muss jedoch die gesamte Blockchain neu gehasht werden. Ein neuer Algorithmus kann ebenfalls die Effizienz des Bitcoin-Protokolls senken. Wird der RIPEMD160 Algorithmus geknackt, gibt es keine Pläne um den Algorithmus zu tauschen. Da dieser Algorithmus ohnehin nur in Kombination mit dem SHA256 Algorithmus angewendet wird, besteht dadurch aber keine ernstzunehmende Gefahr.

3 Skalierbarkeit des Bitcoin Protokolls:

Beim Thema Skalierbarkeit im Bezug auf Bitcoin und dessen Datenbank Blockchain stellt sich die Frage:
Ist es möglich, alle Finanztransaktionen der Welt mit einem Protokoll wie Bitcoin zu realisieren? Weiters, wie kann die aktuelle Zeit von ca. zehn Minuten für eine Transaktionsbestätigung verkürzt werden? Das aktuelle Bitcoinprotokoll schafft ca. 3,5 Transaktionen pro Sekunden, während beispielsweise Visa zu Spitzenzeiten 47.000 Transaktionen pro Sekunde bewerkstelligt. Das größte Problem des Bitcoin Protokolls ist, dass alle Transaktionen an jeden Nutzer übertragen werden. Würden alle global getätigten Transaktionen in die Blockchain eingefügt, wäre der Platzbedarf enorm. Bei 47000 Transaktionen pro Sekunde würde die Blockchain um 400TB pro Jahr wachsen und die Blockgröße müsste von 1MB auf 8GB erhöht werden. Dies würde zu einer Zentralisierung führen, da einzelne Nutzer nicht mehr in der Lage wären, diese Datenmengen zu bewerkstelligen und den Ansatz einer dezentralen Währung ad absurdum führen.
Bitcoin NG [4] versucht mit Verbesserungen am Protokoll höhere Antwortzeiten und weniger Locks und dadurch höheren Durchsatz an Transaktionen zu erreichen. Ein anderer Ansatz stellt das Bitcoin Lightning Netzwerk [5] dar, welches versucht ein Off-Chain Protokoll zu realisieren, indem Micropayments in sogenannten Micropaymentkanälen abseits der Blockchain abgewickelt werden. Dadurch soll die Anzahl der Transaktionen, die in die Blockchain eingefügt werden, drastisch reduziert werden.
Die Autoren des Bitcoin-NG A Scaleable Blockchain Protocol Papers [4] haben in ihrer Implementierung besonders auf Effizienz geachtet. Die Skalierbarkeit des Bitcoin Protokolls wird grundlegend durch zwei Parameter beschränkt:

- Block Size
- Block Intervall

Im Bitcoin Protokoll wird die Anzahl der Transaktionen durch die Blockgröße von einem Megabyte je Block und die Latenz von zehn Minuten beschränkt. Bitcoin-NG versucht diese Beschränkung aufzuheben, sodass die Latenz nur noch von der Ausbreitungszeit der Nachrichten im Netzwerk und der Bandbreite der Knoten abhängt. Hierbei werden die Blockchain Operationen in zwei Teile aufgespalten:

- Leader Election

– Transaction Serialization

Im Bitchoin Protokoll wird die Leader Election zufällig und selten durchgeführt, zwischen diesen es zu langen Locks kommt. Im Gegensatz dazu wird im Bitcoin-NG Protokoll die Zeit in sogenannte Epochen mit je einem Leader aufgeteilt. Ein Leader hat die Aufgabe in seiner Epoche die Transaktionen zu serialisieren. Dieser Ansatz unterscheidet sich wesentlich zum Bitcoin Protokoll, aber es werden die gleichen Sicherheitseigenschaften erfüllt. Leader Election im Bitcoin-NG Netzwerk ist vorrausschauend und stellt sicher, dass die Transaktionen nicht unterbrochen werden.

Bitcoin-NG kennt zwei Arten von Blöcken:

– Keyblocks
– Microblocks

Die Keyblocks werden verwendet, um einen Leader zu wählen. In diesem Blocktyp gibt es wie im Bitcoin Protokoll eine Referenz auf den vorhergehenden Block, Zeitstempel, eine Coinbase Transaktion sowie einen Public Key mit dem folgende Mircoblocks signiert werden. Damit dieser Block gültig ist, muss auch hier ein Kryptopuzzle gelöst werden.
Die Mircroblocks dürfen vom gewählten Leader erzeugt werden, bis eine definierte Maximalzahl erreicht wird. Weiters ist ein Mircroblock nur gültig, wenn sein Zeitstempel nicht in der Zukunft liegt und der Zeitabstand zum vorhergehenden Block ausreichend groß ist. Dadurch kann der Leader nicht in gieriger Absicht das Netzwerk mit Microblöcken überschwemmen. Microblöcke enthalten einen Header mit der ID des Vorgängers, einen Zeitstempel, einen Hash der Transaktionen, sowie eine Signatur des Headers, welche vom Leader mit dem Public Key des Keyblocks erzeugt wurde. Da Microblöcke das Gewicht der Kette nicht verändern, müssen sie auch nicht überprüft werden.
Durch die Ausbreitungszeit von Blöcken im Netzwerk ist es wahrscheinlich, dass ein Schürfer beim Erzeugen eines Keyblocks noch nicht alle Microblöcke des vorhergehenden Keyblocks erhalten hat und es zu einer Verzweigung kommt. Erhalten Knoten den neuen Keyblock, so werden die vom neuen Leader noch nicht erhaltenen Microblöcke verworfen. Deshalb sollten Nutzer zumindest die Ausbreitungszeit im Netzwerk abwarten. Da Microblöcke hochfrequent erzeugt werden, entsteht eine Verzweigung bei fast jedem Keyblock. Jedoch gibt es im Vergleich zu Bitcoin die Regel, dass Schürfer die schwerste Kette erweitern sollen, dadurch bleiben Verzweigungen kürzer bestehen. Auch wenn es möglich ist,

dass Verzweigungen mit neuen Keyblocks entstehen, sind lange Verzweigungensketten nicht gefährlich, da die richtige bekannt wird, sobald die Informationen die Nutzer erreichen.
Der Leader wird für seine Aufwände ähnlich wie beim Bitcoin Protokoll mit einer Coinbase Transaktion und den Transaktionsgebühren vergütet, jedoch werden die Beträge aufgeteilt: 40% erhält der Leader, 60% erhält der nächste Leader. Die Vergütung kann erst nach einer Reifeperiode von 100 Keyblöcken verwendet werden.
Im Bitcoin NG Protokoll wird die Bearbeitungszeit durch die einzelnen Nutzer und die Bestätigungszeit durch die Ausbreitungszeit von Blöcken im Netzwerk beschränkt.

3.1 Sicherheitsanalyse:

Schürfer mit einer Größe kleiner 1/4 des Netzwerkes sind angereizt, dem Protokoll zu folgen, weil

1. sie ihre Transaktionen in die Microblöcke einfügen wollen
2. sie die schwerste Kette erweitern wollen
3. sie die längste Kette erweitern wollen

Hierbei unterscheidet sich Bitcoin NG in Punkt zwei und drei vom Bitcoin Protokoll.
(2) Die Motivation, die schwerste Kette zu erweitern, ist gleich wie bei Bitcoin. Microblocks haben kein eigenes Gewicht und auch keinen eigenen Index, deshalb erhöht sich durch sie auch nicht die Gefahr für egoistisches Schürfen. Eine Minderheit, die sich für eine falsche Kette entscheidet, wird sich nicht durchsetzten können.
(3) Die Motivation, die längste Kette zu erweitern, wird wie folgt gegeben: Der Schürfer könnte versuchen, den Microblock, welcher die Transaktion erhält, zu übergehen und den vorherigen Microblock zu schürfen, und auf Basis dieses einen Keyblock zu erzeugen. Jedoch wäre der Ertrag kleiner, als wenn er den richtigen schürfen würde.

3.2 Angriffsmöglichkeiten:

3.2.1 Transaktionsgebühren:

Ein Leader könnte versuchen, 100% der Transaktionsgebühr, von der ihm eigentlich nur 40% zustehen, abzugreifen, indem er den gesamten Betrag in einer

Transaktion in einen Microblock einfügt, diesen aber nicht veröffentlicht. Daraufhin könnte er versuchen einen neuen Keyblock zu schürfen, während andere Schürfer dies am vorherigen Microblock probieren. Hat der Schürfer weniger als 1/4 der Schürfkraft, macht dies keinen Sinn, da er nicht der nächste Leader werden wird.
Ein Nutzer könnte versuchen, den Leader direkt zu bezahlen, um sich Transaktionsgebühren zu sparen. Dabei könnten sie die Coinbase Adresse des Keyblocks nutzen. Dies könnte die Motivation des Leaders steigern, da er nun anstatt 40% der Transaktionsgebühr 100% von dieser bekommen würde. Schafft der Leader es jedoch nicht die Zahlung zu inkludieren bis der nächste Keyblock erzeugt wird, so verfällt die Transaktion, und der Nutzer verliert die direkt gezahlte Transaktionsgebühr.

3.2.2 Zensurresistenz:

Ein zentrales Ziel von Bitcoin ist die Unabhänigkeit von dritten Instanzen. Ein bösartiger Schürfer soll nicht die Möglichkeit haben Transaktionen eines Nutzers zu verwerfen. Da in Bitcoin-NG ein Leader nur für eine Epoche die Führung übernimmt, ist seine Macht auf diese Epoche beschränkt. Ein bösartiger Leader könnte zwar versuchen per DoS Attacke keine Transaktionen in die Microblocks zu schreiben. Ist die Mehrheit von 3/4 ehrlich, so kann man davon ausgehen, dass eine Transaktion nach 4/3 Keyblöcken, also 13,33 Minuten bestätigt wird. Bitcoin-NG verbessert die Zensurresistenz zwar nicht, aber das Keyblockinvervall könnte verkürzt werden um sie zu erhöhen. Ein weiters Problem sind Schwankungen der Schürfkapazität. Bei Wechselkursschwankungen oder Schwierigkeitssteigerungen der Kryptopuzzle kann es sein, dass sich Schürfer einer rentableren Kryptowährung zuwenden. Durch die verlorene Schürfkapazität dauert das Bestätigen eines Blocks länger. Bei Bitcoin-NG betrifft dies jedoch nur die Keyblöcke, die Verarbeitung der Transaktionen in Microblöcken läuft ungehindert weiter.

3.2.3 Double Spending Attacken:

Da Microblöcke nicht bestätigt werden müssen, könnte ein Leader sogenannte Double Spending Attacken durchführen, indem er verschiedenen Nutzern verschiedene State Machine States der Blockchain überträgt. Um ihn davon abzuhalten, wird ein Blockchain Register verwendet, welche Einträge, sogenannte Poison-Transaktionen, enthält, mit denen vom Leader unrechtmäßig erzeugte

Transaktionen ungültig gemacht werden können. Poison-Transaktionen müssen nach dem nächsten gültigen Keyblock und vor Ablauf der Reifeperiode von 100 Keyblöcken in die Blockchain eingefügt werden. Werden Transaktionen ungültig gesetzt, so verliert der betrügende Leader den gesamten Betrag, der aktuelle Leader erhält 5% davon.

3.3 Kennzahlen:

3.3.1 Fairness:

$$fairness = \frac{\frac{\text{Transaktionen nicht vom größten Schürfer}}{\text{alle Transaktionen}}}{\frac{\text{Schürfkapaziät nicht vom größten Schürfer}}{\text{gesamte Schürfkapazität}}}$$

Optimal wäre ein Wert von 1.

3.3.2 Schürfkapazitätsnutzung:

Das Verhältnis der Schürfleistung zur Absicherung des Systems (Energie, die nicht für die Arbeit an der Blockchain verwendet wird) zur gesamten Schürfleistungen. Da in Bitcoin-NG Kryptopuzzles nur in Keyblocks vorhanden sind, wirken sich Microblöcke und auch Verzweigungen von Microblöcken nicht auf die Schürfenergieverwertung aus.

3.3.3 Zeit um eine Verzweigung zu verwerfen (Time to Prune):

Die Zeit, die benötigt wird, damit ein Nutzer erfährt, dass ein Zweig ungültig ist. Siehe Abb. 3. Anders: Die Zeit, die ein Nutzer warten muss, um sicher zu sein, ob eine Transaktion auch wirklich gültig übernommen wurde.

3.3.4 Zeit zu gewinnen (Time to Win):

Die Differenz zwischen dem Zeitpunkt, an dem ein Nutzer glaubt, einen Keyblock zu haben, welcher noch nie verworfen wurde und dem Zeitpunkt, an dem ihm ein anderer Knoten wiederspricht. Siehe Abb. 3.

3.3.5 Propagationszeit:

Da die Latenzzeit zwischen Knoten unbekannt ist, ist es schwierig, die wirkliche Größe des Bitcoin Netzwerks abzuschätzen. Es ist aber wichtig zu wissen, wie lange es dauert, bis ein Block durchs Netzwerk bis hin zu Schürfhardware übertragen wurde. Dies gilt insbesondere für sogenannte Mining Pools, welche

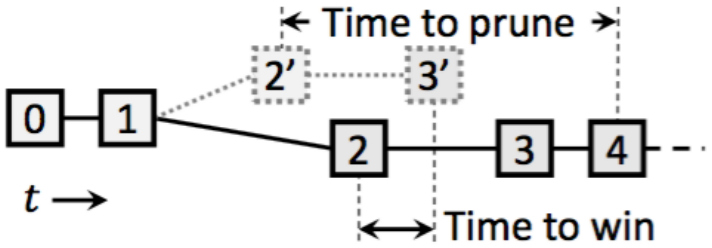

Abb. 3: Time to Prune, Time to Win

auf verteiltes Schürfen setzen. Die Autoren des Bitcoin-NG Protokolls haben bei ihren Tests einen linearen Zusammenhang zwischen Blockgröße und Propagationszeit erkannt, wie er auch bereits im Bitcoin Netzwerk durch Decker und Wattenhofer festgestellt wurde.

3.4 Test des Bitcoin-NG Protokolls:

Zur Analyse des neuen Protokolls wurde ein Netzwerk mit 1000 Knoten, 15% der Größe des aktuellen Bitcoin Netzwerks, realisiert.Es wurde mit unveränderten Bitcoin Clienten (Version 0.10.0) sowie mit realen Internetlatenzzeiten getestet. Anschließend wurde der Bitcoin Client auf das Bitcoin-NG Protokoll erweitert (ohne Transaktionsgebührenverteilung und Microblock Signaturüberprüfung) und beide Clienten verglichen. Bei den Tests wurden Netzwerkbandbreite, Latenz, Blockfrequenz und Blockgröße variiert und geprüft. Bei jedem Test wurden 50-100 Bitcoin Blöcke oder Bitcoin-NG Microblöcke erzeugt. Weiters wurde die Frequenz der Erzeugung von Bitcoin Blöcken bzw. Bitcoin-NG Microblöcken so gewählt, dass sie proportional zum laufenden Bitcoinsystem mit 1MB Blockgröße zu je zehn Minuten ist.
Dabei wurde festgestellt, dass es möglich ist, die Verzögerung und Bandbreite des Bitcoin Protokolls durch Optimierung zu verbessern, ohne die Sicherheitsmetriken zu schwächen. Das Bitcoin-NG Protokoll übertrifft das Bitcoin Protokoll bei allen getesteten Einstellungen.
Das Experiment zeigte, dass eine höhere Blockfrequenz die Bestätigungsdauer sowie die Zeit zum Verwerfen von Verzweigungen erhöht. Verzweigungen in Bitcoin bleiben länger bestehen, als im Bitcoin-NG Protokoll. Bei hoher Blockfrequenz sinkt bei Bitcoin die Fairness stark, da im schlechtesten Fall nur der größte Schürfer Blöcke an die Blockchain anfügen kann. Obwohl bei Bitcoin-NG das

Schürfen auf Keyblöcke beschränkt ist, können bei hoher Blockfrequenz auch bei den Microblöcken ähnliche Probleme auftreten. Hohe Microblockfrequenz verbessert die Bestätigungszeit und die Zeit zum Verwerfen von Verzweigungen. Weiters wurde mit verschiedenen Blockgrößen experimentiert, um die Skalierbarkeit der Bandbreite zu untersuchen. Es wurde die Bitcoin Blockfrequenz und Bitcoin-NG Microblockfrequenz auf zehn Sekunden sowie die Keyblockfrequenz auf 100 Sekunden gesetzt. Wie erwartet erhöht sich die Transaktionsfrequenz mit der Blockgröße. Größere Blockgrößen erhöhen die Propagationszeit und somit das Risiko der Verzweigungen.
Folgend die Resultate aus dem Paper für ausgewählte Werte, in denen Bitcoin-NG (grün) große Vorteile gegenüber Bitcoin (blau) erreichen könnte.

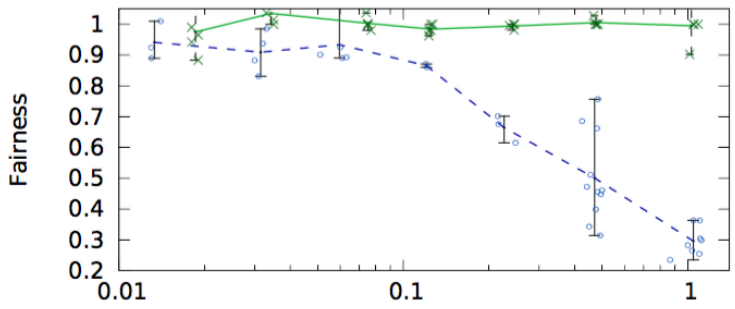

Abb. 4: Fairniss bei Veränderung der Latenzzeit (1/sec)

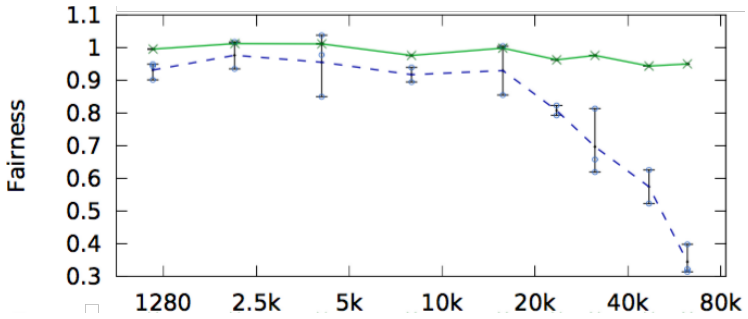

Abb. 5: Fairniss bei Veränderung der Blockgröße (Byte)

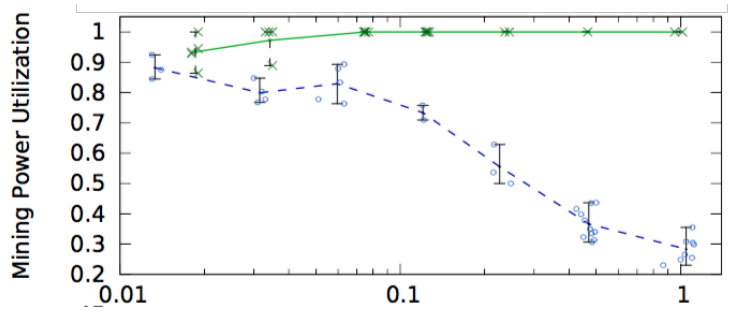

Abb. 6: Schürfkapazitätsnutzung bei Veränderung der Latenzzeit (1/sec)

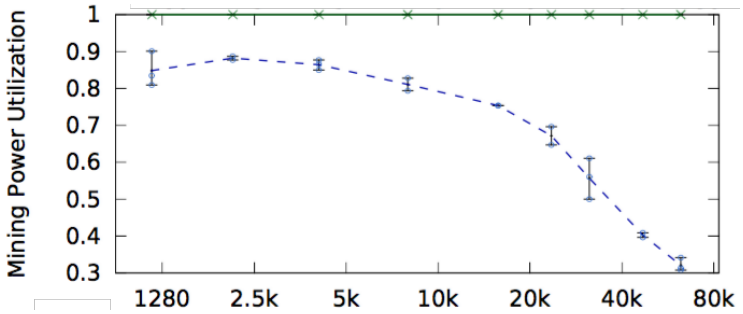

Abb. 7: Schürfkapazitätsnutzung bei Veränderung der Blockgröße (Byte)

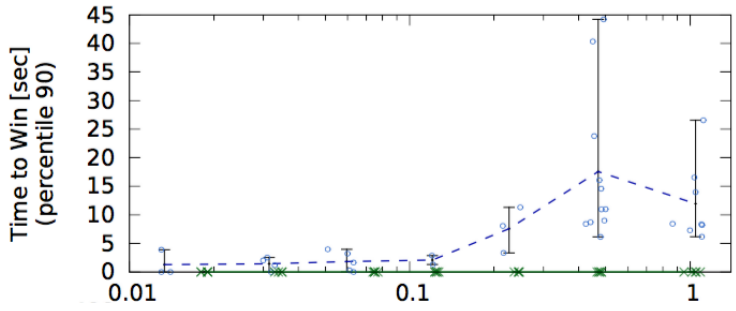

Abb. 8: Time to Win bei Veränderung der Latenzzeit (1/sec)

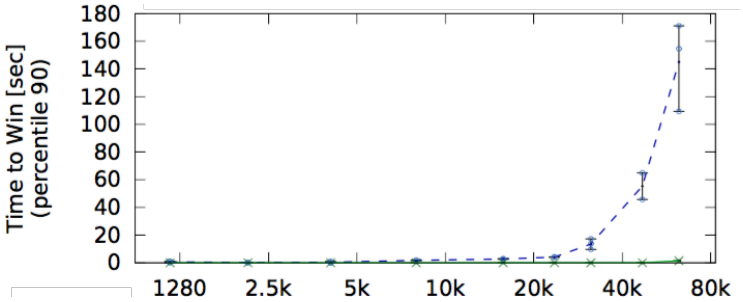

Abb. 9: Time to Win bei Veränderung der Blockgröße (Byte)

Es gibt einen Trade-Off zwischen Bandbreite und Latenz. Eine hohe Latenzzeit erhöht die Zeit um zu gewinnen, sowie die Zeit, um eine Verzweigung zu verwerfen. Dabei entstehen neue Sicherheitsprobleme, insbesonders da die Verzweigungen im Bitcoin Netzwerk einen signifikanten Anteil an Schürfenergie (bis zu 80%) verbrauchen. Dies erhöht die Angreifbarkeit des Systems. Abgesehen davon sinkt die Fairniss und somit der Anreiz für kleinere Schürfer, was zu einer Zentralisierung der Schürfenergie führt. Bitcoin-NG zeigt eine deutliche Verbesserung und auch bei großen Blockgrößen keine Einbußen bei Fairness und Schürfkapazitätsnutzung. Außerdem wurde bei hoher Bandbreite keine Verschlechterungen der Bestätigungszeit und der Zeit zum Verwerfen von Verzweigungen entdeckt.

3.5 Conclusion Bitcoin-NG

Bitcoin und ähnliche Kryptowährungen haben Skalierbarkeitsprobleme. Mit Bitcoin-NG haben die Autoren gezeigt, dass es möglich ist, die Skalierbarkeit des Bitcoin Protokolls deutlich zu erhöhen, sodass das Netzwerk nur noch durch den Durchmesser und die Rechenleistung der Nutzer beschränkt wird. Dies ermöglicht die Bearbeitung einer großen Menge von Transaktionen.

3.6 Bitcoin Lighting Network

Ein anderer Ansatz ist das Bitcoin Lightning Network [5], bei dem sogenannte Micropaymentkanäle erzeugt werden, in denen Zahlungen außerhalb der Blockchain durchgeführt werden. Da viele Micropayments regelmäßig zwischen zwei Nutzern gesendet werden, benötigen diese weniger Vertrauen. Micropaymentkanäle erzeugen eine Beziehung ausschließlich zwischen zwei Nutzern, die den

Eintrag einer Transaktion in die Blockchain auf einen späteren Zeitpunkt verzögert. Sind sich zwei Nutzer über die Balance des Kanals einig, so ist dies der korrekte Kontostand. Beide Nutzer zahlen in eine 2-zu-2 Multisignaturadresse ein und signieren gegenseitig Auszahlungen zu je 50%, welche sie jedoch nicht in die Blockchain übertragen. Um dieses Verhältnis zu verändern, müssen sich die Nutzer auf neue Auszahlungstransaktionen einigen. Hierbei gibt es jedoch ein Zeitstempelproblem, da die Blockchain nicht wissen kann, welche der beiden Transaktionen nun korrekt ist. Dies ist jedoch kein komplexes Problem, da es nur zwei Stati von Transaktionen gibt:

- der aktuelle korrekte Kontostand
- alle verworfenen Kontostände.

Die Autoren zeigen, dass es möglich ist, ein Protokoll zu etablieren, welches alte Kontostände auf ungültig setzt, sodass nur die neue Transaktion gültig ist. Dieses Protokoll sieht vor, dass sich Nutzer, wenn der andere versucht einen verworfenen Kontostand einzulösen, den gesamten Kontostand des Kanals als Strafe auszahlen lassen. Micropaymentkanäle an sich lösen jedoch nicht das Skalierbarkeitsproblem, wenn jeder mit jedem Kanäle erzeugt. Die Skalierbarkeit kann durch ein großes Netzwerk dieser Kanäle verbessert werden. Der Gedanke hierbei ist, dass nur Zahlungen von unkooperativen Nutzern in die Blockchain übertragen werden müssen. Die Blockchain wird als dezentrales Zeitstempelsystem für die Überprüfung der Gültigkeit von Zahlungen genutzt.
Die Autoren des Lightning Networks beschreiben zwei Arten von Verträgen, den Revocable Sequence Maturiy Contract (RSMC), welcher auf dem Sequenznummeralter von Transaktionen basiert, sowie den Hash Time Lock Contract (HTLC), welcher auf der Lieferung von Bytes, welche einen passenden Hash liefern müssen, sowie einem Time Lock, aufbaut.

3.7 Transaktionstypen:

Es gibt 5 Typen von Transaktionen:

- Funding - Einzahlungstransaktion
- Commitment - Verpflichtungstransaktion
- Delivery - Lieferungstransaktion
- Breach Remedy - Straftransaktionen
- Exercise Settlement - Vertragsabschlusstransaktion

3.8 Der RSMC (Revocable Sequence Maturiy Contract):

Bei der Erstellung des Kanals erstellen beide Nutzer eine Einzahlungs- und Auszahlungstransaktionen, jedoch ohne sie zu signieren. Beide Nutzer tauschen ihre Signaturen nicht aus, solange sie keine Auszahlungstransaktion erstellen, um ihre Einzahlung zurückzuerhalten. Sollten die Nutzer nicht koopoerieren und später die Signaturen der Einzahlungstransaktionen nicht austauschen, bleibt die Einzahlung für immer gesperrt. Die Nutzer tauschen einen Schlüssel aus, um die Transaktionen später signieren zu können. Dieser Schlüssel wird für die 2-zu-2 Auszahlung verwendet.
Um von einer unsignierten Transaktion Geld auszahlen zu können, wird Bitcoin um einen Transaktionstyp `SIGHASH_NOINPUT` erweitert. Dadurch ist es möglich, Kindtransaktionen basierend auf der ID nicht signiertet Transaktionen vom Typ `SIGHASH_NOINPUT` zu erstellen. Um diese Einzahlungen ausgeben zu können, müssen die Nutzer kooperieren und ihre Signaturen austauschen und die Einzahlungstransaktion signieren.

3.8.1 Erstellung eines Kanals:

1. Einzahlungstransaktion erstellen
2. Verpflichtungstransaktion erstellen
3. Verpflichtungstransaktion signieren
4. Signaturen für die Verpflichtungstransaktion austauschen
5. Einzahlungstransaktion signieren
6. Signaturen der Einzahlungstransaktion austauschen
7. Einzahlungstransaktion der Blockchain hinzufügen

Möchten die Nutzer nun das Verhältnis eines Kanals ändern, also eine Zahlung durchführen, müsste zuerst die Verpflichtungstransaktion veröffentlicht werden, da sonst jeder Teilnehmer, die für sich günstigere Verpflichtungstransaktion nutzen könnte. Deshalb ist ein Mechanismus nötig, um zu verhindern, dass alte Verpflichtungstransaktionen veröffentlicht werden können.

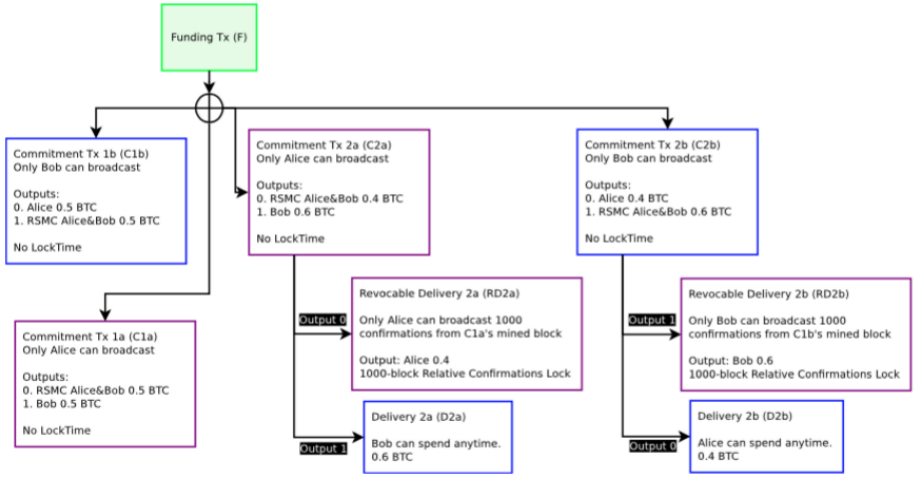

Abb. 10: Neues Kanalverhälntis

Hier wurde ein Konstrukt gewählt, ähnlich einer Kautionsversicherung, welche einen Nutzer bei unrechtmäßiger Veröffentlichung einer alten Transaktion mit einer Strafe belegt und der anderen Partei den gesamten Betrag des Kanals zuspricht. Da die Verpflichtungstransaktionen von der jeweils anderen Partei unterschrieben wurden, ist ersichtlich, wer die Transaktion veröffentlicht hat. Alle Auszahlungen (Kindtransaktionen) der Einzahlungstransaktion sind Verpflichtungstransaktionen, welche zwei halb-signierte Transaktionen haben.

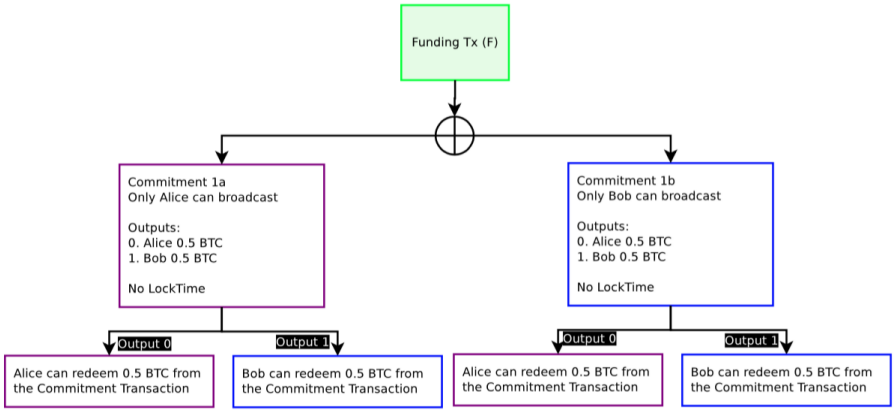

Abb. 11: Einzahlungs- und Verpflichtungstransaktionen

Der Aufbau ist wie folgt:
Damit es möglich ist, Zahlungen zu widerrufen, wird eine eindeutige, fortlaufende und nicht überschreibbare Sequenznummer eingeführt. Durch können Time Locks eingefügt und mit der Funktion `OP_CHECKSEQUENZVERIFY` überprüft werden. Um Transaktionen abseits der Blockchain mit Sequenznummern durchzuführen, wird eine Wartezeit eingeführt.
Die widerrufbare Transaktion enthält zwei Rückgabepfade, bei der beim einen die Transaktion sofort, beim anderen nach einer bestimmten Anzahl an Bestätigungen durchgeführt werden kann. Dies wird dadurch erreicht, dass die Verpflichtungstransaktion eine bestimmte Anzahl an Bestätigungen der Einzahlungstransaktion benötigt (Sequenznummer). Innerhalb dieser Periode von 1000 Bestätigungen (ca. 7 Tage) hat die andere Partei die Möglichkeit, die Widerufstransaktion in der Blockchain zu platzieren und damit sich sofort den kompletten Betrag des Kanals als Strafe zu sichern. Wird die Anzahl an Bestätigung erreicht, wird aus der Einzahlungstransaktion mit einer Auszahlung eine verpfändete Einzahlung ohne Widerrufsmöglichkeit.

Der Ablauf ist wie folgt:

1. Beide Nutzer zahlen in einen Kanal ein, und erzeugen Auszahlungstransaktionen

2. Beide Nutzer können eine widerrufbare Ausgabenverteiltung mit einer bestimmten Wartezeit wählen (z.B. 1000 Bestätigungen)
3. Eine oder beide Nutzer können sich dazu entschließen, die Veröffentlichung der Transaktion in die Blockchain auf einen späteren Zeitpunkt zu verschieben.
4. Hat kein Nutzer die Transaktion veröffentlicht, können Auszahlungen nur durchgeführt werden, indem beide Nutzer kooperieren und sich auf eine neue Auszahlungstransaktion einigen.
5. Wurde die Transaktion veröffentlicht und das neue Verhältnis nicht widerrufen, so liegt die Verantwortlichkeit wieder bei beiden Parteien zu kooperieren.

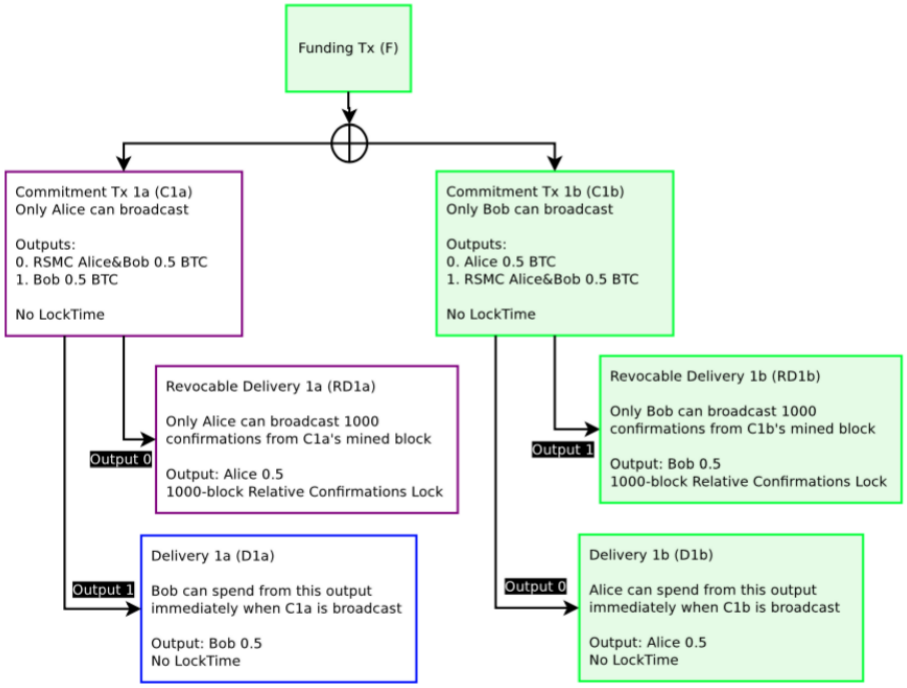

Abb. 12: RSMC - Verpflichtungstransaktion mit Auszahlungspfaden

Eine Kindtransaktion kann also erst nach 1000 Bestätigungen der Elterntransaktion eingelöst werden. Innerhalb dieses Zeitraumes kann die Transaktion mithilfe der Sequenznummer widerrufen werden. Um Kindtransaktionen zu widerrufen,

bzw. das Verhältnis des Kanals weiter zu verändern, muss eine neue Kindtransaktion erzeugt werden. Dies funktioniert nur, wenn beide kooperieren. In dem Zeitfenster der 1000 Bestätiungen müssen beide Teilnehmer die Blockchain beobachten, um zu prüfen, ob nicht der andere Teilnehmer eine Widerufstransaktion eingefügt hat. Dadurch ist sichergestellt, dass keiner der beiden ein Interesse daran hat die Transaktion zu veröffentlichen.
Eine solche Konstruktion ermöglicht jedem eine Transaktion zu erstellen, sie nicht zu veröffentlichen und später mit Strafzahlungen Anreize zu schaffen, sie weiterhin nicht zu veröffentlichen und dadurch viele Transaktionen vom Eintrag in die Blockchain auszusetzen.
Ein Ansatz, um die von einem böswilligen Angreifer eingefügte Flut an Transaktionen abzuschwächen, ist ein Timestop. Hierbei können Schürfer ein Flag setzen, z.B. das letzte Bit der Versionsnummer oder ein eigenes Flag im Blockheader. Bei 1 wird die Sequenzanzahl dieser Transaktionen nicht mehr für die Anzahl an nötigen Bestätigungen berücksichtigt und der Block als verstopft deklariert. Es könnte ein Automatismus angedacht werden, der bei im Durchschnitt zu hohen Transaktionsgebühren einen Block automatisch auf verstopft setzt.

3.8.2 Breach Remedy Transaktionen - Straftransaktionen:

Breach Remedy Transaktionen geben der jeweils anderen Partei einen begrenzten Zeitraum lang die Möglichkeit den gesamten Betrag als Strafe zu erhalten, wenn man selbst eine alte Transaktion veröffentlicht.
Ein Ansatz ist, eine dritte Partei zu beauftragen, die Blockchain zu überwachen. Dazu könnte man diesem Überwacher im Output der Breach Remedy Transaktion eine Gebühr zugestehen und ihm dadurch einen Anreiz geben, sie zu veröffentlichen, falls die andere Partei eine alte, ungültige Verpflichtungstransaktion veröffentlicht hat. Da die dritte Partei nur bei einer ungültigen Verpflichtungstransaktion reagieren kann, entstehen dadurch keine Sicherheitsprobleme.
Der Prozess, widerrufbare Verpflichtungstransaktionen zu bearbeiten, benötigt ein geignetes Kanal Konstrukt. In diesem Ablauf muss festgelegt werden, welche Public Keys für neue Verpflichtungen verwendet werden, da `SIGHASH_NOINPUT` Transaktionen eindeutige Public Keys für jede Verpflichtungstransation des RSMC (oder HTLC) benötigen. Dazu gibt es beispielsweise die BIP 0032 [6] HD Wallet Konstruktion, bei der die Master Public Keys bei der Erzeugung des Kanals ausgetauscht werden. Verpflichtungstransaktionen, welche älter als die aktuelle sind, werden mit Straftransaktionen (Breach Remedy) ungültig gesetzt. Um Verpflichtungstransaktionen auf ungültig zu setzen, können die Nutzer dem jeweils

anderen den Private Key der jeweiligen Transaktion senden. Da nun das Gegenüber den Private Key kennt, könnte bei unrechtmäßiger Veröffentlichung der kompletten Betrag beanspruchen qweswn. Deshalb ist es im eigenen Interesse, diese alte, ungültige Transaktion zu vernichten.

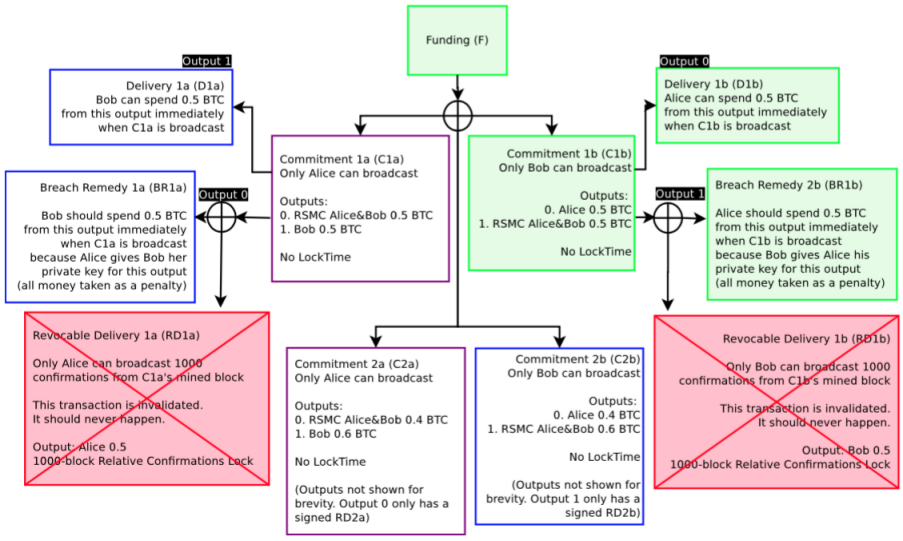

Abb. 13: Kanal mit alten Commitment Transaktionen mit Straftransaktion (Breach Remedy)

3.8.3 Auszahlung:

Beim unkooperativen Schließen eines Kanals kann ein Nutzer die aktuelle Transaktion in die Blockchain schreiben, müsste dann jedoch den Bestätigungszeitaum abwarten. Durch kooperatives Schließen kann diese Wartezeit vermieden werden. Dabei wird eine Exercise Settlement Transaktion erzeugt, welche die Gelder direkt aus der Einzahlungstransaktion entnimmt.

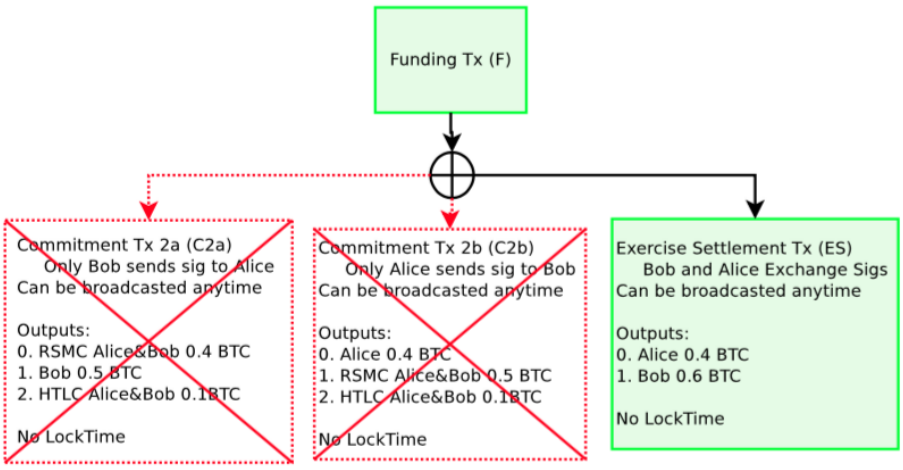

Abb. 14: Kooperatives Schließen eines Kanals

3.9 Der Hashed Timelock Contract (HTLC):

Ein bidirektionaler Zahlungskanal ermöglicht nur sichere Transaktionen innerhalb des Kanals. Damit es möglich ist, Transktionen auch zwischen Kanälen durchzuführen, wird ein weiteres Konstrukt benötigt, der HTLC. Auch dieser Vertrag benötigt keine dritte Partei. Ein HTLC ermöglicht es sichere Transaktionen zu erstellen, welche nur nach einem bestimmten Zeitstempel gültig sind. Dazu wird nLockTime eingeführt. Weiters muss es einem Nutzer möglich sein, einen HTLC zu widerrufen. Dies kann wieder per Eintrag in die Blockchain erreicht werden.
Ablauf wie folgt:

1. Bob soll Alice unbekannte 20Byte Daten `R`, mit einem bekanntem Hash `H` senden. Erfüllt Bob dies, so bestätigt Alice den Vertrag durch Zahlung von 0.1 BTC an Bob.
2. Sind die 432 Blöcke (3 Tage) vergangen, verfällt der Vertrag und keine Partei beansprucht Zahlungen.
3. Jeder Nutzer sollte sich die Gelder wie im Vertrag vorgesehen auszahlen lassen, und der Vertrag wird geschlossen.
4. Die Verletzung des Vertrags wird mit einer Strafe in Höhe des gesamten Betrags an die Gegenpartei geahndet.

Ein Teilnehmer entscheidet sich also, eine Zahlung auf Basis eines innerhalb eines Zeitraum zu übertragenden `R`'s zu schließen.
Dazu werden Verpflichtungstransaktionen wie im RSMC abgeschlossen. Kennt Bob `R` innerhalb von 3 Tagen, kann er sich die Gelder auszahlen lassen, indem er die Transaktion in die Blockchain einfügt. Das Script ist also als zusätzliche Bedingung in einer Verpflichtungstransaktion zu sehen:

```
 if HASH(R) = H
  <Alice2> <Bob2> OP_CHECKMULTISIG (Zahlung an Bob)
 else
  <Alice1> <Bob1> OP_CHECKMULTISIG (Rueckzahlung an Alice)
}
```

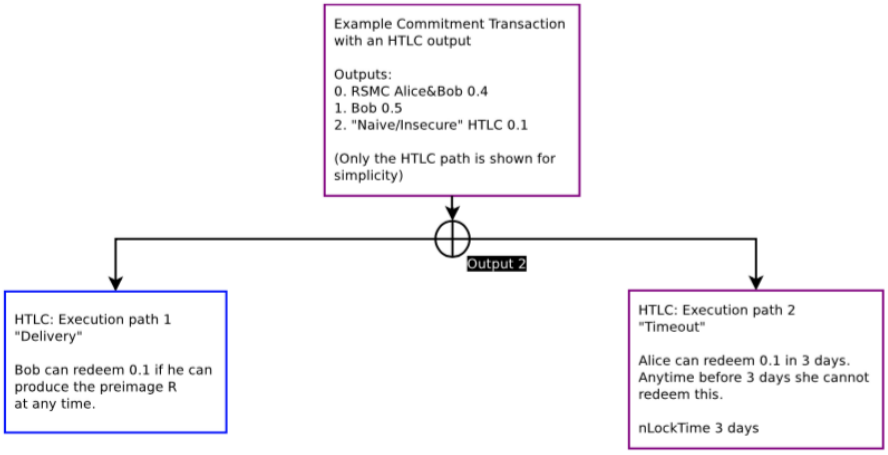

Abb. 15: HTLC Verpflichtungstransaktion

Es gibt also zwei Auszahlungspfade. Der Erste basiert darauf, dass Bob `R` mit `H=hash(R)` haben muss, der Zweite ist durch den Timelock von 432 Blöcken geschützt.
Hier gilt jedoch das gleiche Problem wie beim RSMC, dass nicht erkannt werden kann, welche Transaktion gültig ist, wenn die Teilnehmer das Verhälnis des Kanals ändern und neue Verpflichtungstransaktionen erstellen. Beispielsweise könnte `R` später bekannt werden und jemand könnte versuchen die Gelder zu stehlen. Deshalb ist es beim HTLC nötig, dass der Vertrag am Ende ge-

schlossen wird, da beide Pfade später gültig werden können. Bei unkooperativen Teilnehmern ist ein Eintrag der Verpflichtungstransktion bei Abschluss in die Blockchain deshalb nötig. Damit dies nicht nötig wird, kann der RSMC mit dem HTLC kombiniert werden, um widerrufbare Off-Chain HTLCs zu ermöglichen.

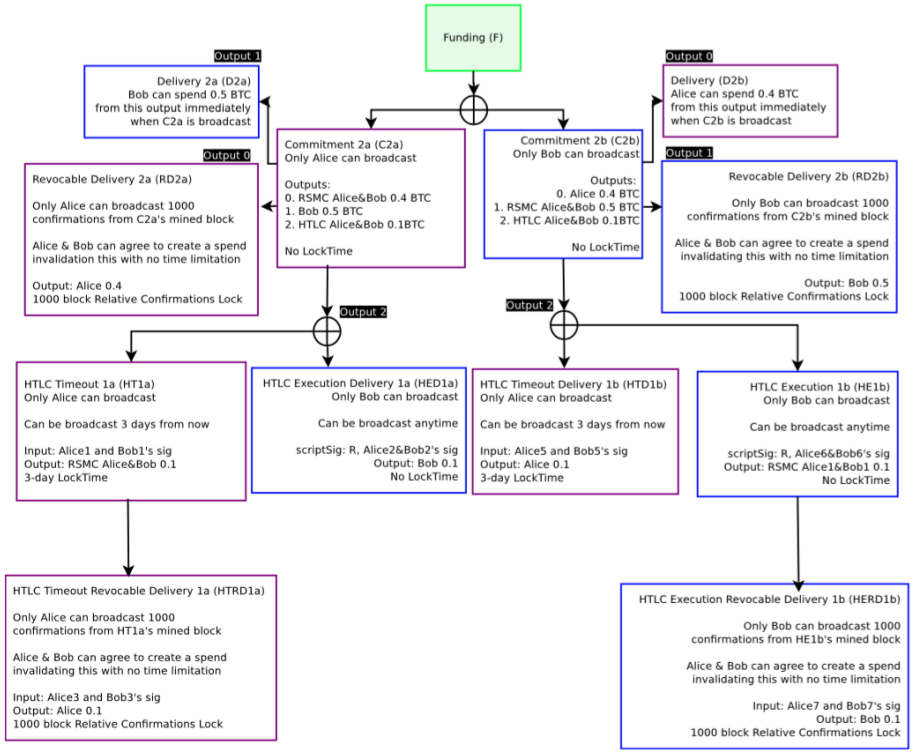

Abb. 16: Widerrufbarer Off-Chain HTLC

3.10 Lighting Network:

3.10.1 Schlüsselverwaltung:

Die Schlüssel im Lighting Network werden mit BIP 0032 Hierarchical Deterministic Wallets [6] bei Erstellung eines Kanals generiert. Dazu wird ein Hash-Baum genutzt, welcher eine Vielzahl an Schlüssel erzeugt. Dabei wird z.B. ein Schlüssel als Master-Key für einen Tag genutzt, und seine Kind-Schlüssel für die Transaktionen. Durch Bekanntgeben des Master-Keys an den anderen Nutzer

können mit der Übertragung eines einzigen Schlüssel viele Transaktionen auf einmal ungültig gesetzt werden.

3.10.2 Blockchain Transaktionsgebühren für bidirektionale Kanäle:

Da klar ist, welcher Nutzer die Transaktion in die Blockchain einfügt, könnte ein Service genutzt werden, um Gebühren einer 2-zu-3 `MULTISIG` einzubehalten. Möchte nun ein Nutzer anstatt eines kooperativen Schließens des Kanals die Transaktion in die Blockchain schreiben, wird der Service kontaktiert und der Eintrag in die Blockchain erfolgt.

3.10.3 The Bitcoin Lightning Network

Durch Micropaymentkanäle, welche mit Verträgen durch Hash Locks und Timelocks gesichert sind, ist es möglich, Transaktionen zwischen Kanälen ohne eine Kontrollinstanz durchzuführen, indem eine Serie von absteigenden Timelocks verwendet wird. Im traditionellen Bankenwesen werden Transaktionen gecleart, indem sie an einem zentralen Punkt bestätigt werden. Beim Lighting Network werden Zahlungsverpflichtungen in einer Kette von einem Nutzer zum nächsten weitergeben. Dabei wird jeweils ein HTLC geschlossen, welcher einen kürzeren Timelock hat.

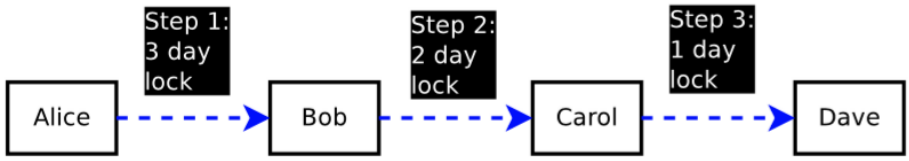

Abb. 17: Übertragung von Zahlungsverpflichtungen

Dabei werden Zahlungen über andere Personen gesendet. Beispiel: Alice will Dave einen Betrag senden, so kann sie dies über Bob und Carol machen, indem HTLC mit einer absteigenden Timelockzeit geschlossen werden. Jeder Vertrag wird Off-Chain geschlossen und wenn alle kooperieren, erfolgt kein Eintrag in die Blockchain.
Erfüllen alle Parteien in den entsprechenden Zeiträumen ihre HTLCs, kann die Transaktion abgeschlossen werden, ansonsten wird die Verpflichtungstransaktion von dem jeweiligen Partner in die Blockchain geschrieben. Das Risiko, Transak-

tionsgebühren zu bezahlen, besteht also nur gegenüber der direkt verbundenen Parteien, die anderen haben nur das Risiko eines Verzugs, wenn der HTLC nicht eingehalten wurde. Sollte ein Teilnehmer dieser Kette die Verbindung verlieren, ist die andere Partei verantworlich, die aktuelle Verpflichtungstransaktion in die Blockchain zu übertragen. Nur diese Stati werden in die Blockchain übertragen.

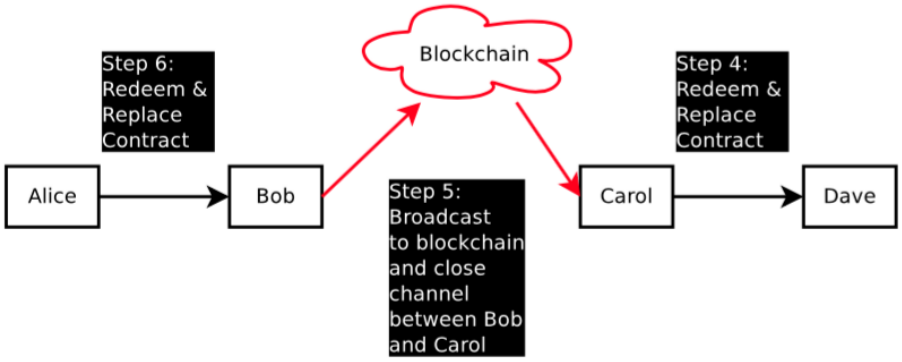

Abb. 18: Eintrag in Blockchain bei Verbindungsverlust

Sollten extrem hohe Beträge gesendet werden, ist es möglich, dass die Zahlung nicht bis zum Ende durchgeführt werden kann, deshalb wird empfohlen kleine Beträge zu senden. Sollte ein Teilnehmer unkooperativ sein, müsste der Sender den Timelock abwarten, bis er den Betrag zurück erhält. Hier gibt es einen Trade-Off zwischen den niedrigen Gebühren und möglichst kleinen Transaktionen, da der prozentuale Anteil wieder höher ist als in einem direkten Kanal.
Weiters ist es möglich eine Transaktion, die ihr Ziel nicht erreicht, über einen anderen Nutzer umzuleiten.

3.10.4 Routing:

Theoretisch können Routing Maps erstellt werden, indem die 2-zu-2 `MULTISIGS` der Blockchain in eine Routingtabelle eingetragen werden. Dies ist jedoch mit Pay-to-Script Hash Transaktionen nicht möglich, könnte aber durch Routing Service einer dritten Partei gelöst werden. Dabei könnte ähnlich wie bei ISP und IP Paketen eine Route vorgeschlagen werden, z.B. über Teilrouten zu bekannten Knoten.

3.10.5 Gebühren:

Die Gebühren des Lightning Netzwerks unterscheiden sich von den Blockchaingebühren, da diese direkt zwischen den Kanalteilnehmern bezahlt werden. Die Gebühren werden für die Zeitsperre der Gelder eines Kanals abgegolten, ähnlich wie Zinsen, sowie für das Risiko eines unkooperativen Partners.

3.11 Risiken:

3.11.1 Time Lock:

Das primäre Risiko ist der Time Lock. Wurde die Dauer falsch gewählt, so können geglaubte Transaktionen gültig werden, was einen Diebstahl ermöglichen würde. Hier gibt es einen Trade-Off zwischen langen Timelocks und dem Zeitwert des Geldes. Beim Schreiben der Wallet Software und der Lightning Network Applikation ist es notwendig sicherzustellen, dass genügend Zeit gewählt wird, damit Nutzer die Möglichkeit haben, ihre Transaktionen in die Blockchain einzufügen, falls Nutzer unkooperativ oder mit böswilliger Absicht handeln.
Das größte Risiko sehen die Autoren bei erzwungenem Ende vieler Transaktonen. Werden von bösartigen Teilnehmen viele Kanäle erzeugt und diese gleichzeitig beendet, so könnte es passieren, dass die Kapazität des Bitcoin Netzwerks nicht ausreicht, um die Transaktionen der Nutzer in die Blockchain schreiben zu können, bevor die Lock Zeit ausläuft. Das könnte abgeschwächt werden, indem mit einer Transaktion alle offenen Transaktionen erstetzt werden können. Eine solche Attacke wäre jedoch auch ein sehr großes Risikio für den Angreifer, da ein falsches Veröffentlichen einer Verpflichtungstransaktion den vollen Betrag als Strafe vorsieht.
Vergessen Nutzer eine Transaktion innerhalb der Lock Time zu veröffentlichen, könnte die andere Partei Gelder stehlen. Dies könnte durch einen Service einer dritten Partei gelöst werden. Ebenso kann dies durch die Funktion `OP_CHECKSEQUENCEVERIFY` abgeschwächt werden.

3.11.2 Diebstahl via Cracking:

Da Nutzer online sein müssen und ihre privaten Schlüssel zum Signieren verwenden, wäre es möglich, diese Rechner zu kompromitieren und die Gelder zu stehlen. Die Autoren sehen hier Möglichkeiten dies abzuschwächen, indem z.B. verschiedene Arten von Wallets Checking Account, Saving Account verwendet werden, und in der online verfügbaren Hot Wallet nur wenig Geld verfügbar ist.

3.11.3 Datenverlust:

Verliert eine Partei ihre Daten, so ist es der anderen möglich, diese Gelder zu stehlen. Dies könnte durch einen Service einer dritten Partei abgeschwächt werden, an welche entsprechende Daten verschlüsselt übertragen werden können.

3.11.4 Attacken im Zusammenspiel mit Schürfern:

Schürfer könnten das Einfügen von Transaktionen (z.B. der Straftransaktionen) verzögern, damit ein Timeout erreicht wird und Gelder gestohlen werden können. Diese Attacke ist jedoch für den Angreifer sehr unattraktiv und risikoreich.

3.11.5 Veränderungen am Bitcoin Protokoll - Blocksize Erhöhung:

Ein Problem sehen die Autoren darin, dass nötige Veränderungen wie Timestops nicht ins Bitcoin Protokoll übernommen werden können, falls das Lightning Netzwerk populär wird. Dadurch würden sich systembedingte Sicherheitsrisiken ergeben.

3.12 Einsatzbereiche:

3.12.1 Instant Transactions:

Durch Soforttransaktionen ist es mit dem Lightning Network möglich z.B. für einen Kaffee zu bezahlen, mit einer direkten nicht widerrufbaren Zahlung innerhalb von Millisekunden.

3.12.2 Exchange Arbitrage:

Geldwechsel - Mit dem Lightning Network ist es möglich, Geldwechsel nahezu ohne Verzögerung durchzuführen.

3.12.3 Micropayments:

Bei Micropayments sind im Bitcoin Netzwerk die Transaktionsgebühren zu hoch. Das Lightning Network kann diese Miropayments kostengünstig mittels Bitcoins ohne den Nachteil eines Treuhänders abwickeln. Z.B. Zahlung für Internetdatenvolumen oder Gebühren für Onlinezeitungsartikel.

3.12.4 Finanzverträge und Treuhändervereinbarungen:

Zeitkritische Finanzverträge haben höhere Anforderungen an die Blockchainberechnung. Mit dem Lightning Network können diese schnell Off-Chain abgewickelt werden und möglicherweise ohne Eintrag in die Blockchain.

3.12.5 Cross-Chain Payments:

Zahlungen können mit unterschiedlichen Regeln über verschiedene Ketten geroutet werden. Der Sender muss dazu nicht vertrauen oder über diese Ketten Bescheid wissen, nicht einmal über das Ziel. Analog muss der Empfänger nichts über den Sender wissen. Es muss nur darauf geachtet werden, dass die Regeln des Vertrages mit dem jeweiligen Nutzer in der gleichen Kette eingehalten werden. Ebenso wäre es möglich, eine Verkettung über verschiedene Kryptowährungen durchzuführen. Angenommen: Alice (Bitcoin), Bob (Bitcoin und X-Coin), Carol (X-Coin), so könnte Alice über Bob Carol bezahlen, ohne die X-Coin Regeln zu kennen.

3.13 Conclusion - The Lightning Network:

Würden alle 7 Milliarden Menschen zwei Transaktionen pro Tag durchführen, würde Bitcoin 24GB große Blöcke alle 10 Minuten (250 Bytes pro Transaktion) benötigen. Diese wären von den Knoten im Bitcoin Netzwerk nicht bearbeitbar und würden zwangsläufig zu einer Zentralisierung führen.
Mit dem Ansatz des Lightning Networks würde eine Blockgröße von 133MB (500 Bytes pro Transaktion) alle 10 Minuten ausreichen, um diese Menge an Zahlungen durchführen zu können. Dies würde einen Datenzuwachs von 7TB pro Jahr bedeuten. Die Autoren wollen alte Blöcke entfernen und den nötigen Speicherbedarf auf 2TB senken.
Ein Netzwerk an Micropaymentkanälen verleiht Bitcoin Skalierbareit, kostengünstige Micropayments und nahezu Soforttransaktionen. Die Transaktionen in den Micropaymentkanälen sind echte Bitcoin Transaktionen, welche abseits der Blockchain ohne einen Treuhänder durchgeführt werden können. Mit Hilfe des Lightning Networks kann Bitcoin skalieren, indem Transaktionen Off-Chain durchgeführt werden, ohne die Risiken einer Zentralisierung oder der Notwendigkeit von Treuhändern.

Literatur

1. Bitcoin: A peer-to-peer electronic cash system (2008)
2. Die dubiose masche der bitcoin-betrüger (December 2013)
3. Giechaskiel, I., Cremers, C., Rasmussen, K.B.: On bitcoin security in the presence of broken crypto primitives. - (2016)
4. Eyal, I., Gencer, A.E., Sirer, E.G., van Renesse, R.: Bitcoin-ng: A scalable blockchain protocol. CoRR **abs/1510.02037** (2015)
5. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments. (2016)
6. Bip 0032: Hierarchical deterministic wallets. (February 2012)