Kryptographische Uhren und Anwendungen[1]

Giulio Malavolta[2]

Abstract: In dieser Dissertation definieren und konstruieren wir kryptographische Objekte, die das *Vergehen von Zeit* simulieren. Im Einzelnen bauen wir unter der Annahme, dass bestimmte mathematische Probleme unlösbar sind, die folgenden Grundelemente auf:

(1) *Inkrementelle Beweise für sequentielle Arbeit* ermöglichen es einem Beweisführer, einen Prüfer davon zu überzeugen, dass eine bestimmte Menge sequentieller Arbeit verrichtet wurde.

(2) *Homomorphe Zeitschlossrätsel* verbergen Geheimnisse für einen bestimmten Zeitraum und ermöglichen es jedem, Funktionen über Klartexte öffentlich zu berechnen, ohne die Rätsel zu lösen.

(3) *Mehrschrittschlösser* ermöglichen es Anwenderpaaren Verträge abzuschließen, bei denen Münzen überwiesen werden, wenn eine bestimmte Bedingung innerhalb einer bestimmten Zeit erfüllt ist.

Einleitung

Wir betrachten das Problem, das Vergehen von Zeit in Gegenwart böswilliger Parteien zu messen. Wir untersuchen jene Szenarien, in denen sich gegenseitig misstrauende Individuen, Bedingungen durchsetzen oder Entscheidungen treffen möchten, die darauf beruhen, ob eine bestimmte Zeitspanne verstrichen ist oder nicht. Man betrachte als ein höchst anschauliches Beispiel den Fall, dass eine Person ein Geheimnis erst nach Ablauf der Zeit T, für ein beliebiges T, veröffentlichen möchte. Wie stellt man sicher, dass das Geheimnis verborgen bleibt, möglicherweise auch nach dem Tod eines solchen Individuums? Das Thema dieser Dissertation ist es, kryptographische Objekte zu identifizieren, zu definieren und zu konstruieren, die als Lösung für diese Klasse von Problemen eingesetzt werden können. Eine Lösung wird als kryptographisch bezeichnet, wenn ihre Sicherheit auf Mathematik beruht und sie keine physischen Eigenschaften ausnutzt oder auf das Vertrauen einer externen Entität angewiesen ist. Diese Dissertation schlägt Lösungskandidaten für die folgenden Fragen vor, welche auf mathematischen Problemen beruhen, bei welchen man davon ausgeht, dass sie sehr schwer zu lösen sind.

Beweis vergangener Zeit. Angenommen, ein Beweisführer möchte einen misstrauischen Verifizierer davon überzeugen, dass (ungefähr) eine bestimmte Zeitspanne T verstrichen ist, um die Echtheit eines bestimmten Zeitstempels zu beweisen. Eine Möglichkeit dies umzusetzen, besteht darin, den Beweisführer vom Zeitpunkt der Erzeugung des Objekts an, eine bestimmte (inhärent sequentielle) Berechnung starten und später nachweisen zu lassen, dass ein angemessener Teil der Berechnung auf seinem Computer ausgeführt wurde.

[1] Englischer Titel der Dissertation: „Cryptographic Clocks and Applications“

[2] giulio.malavolta@hotmail.it

CC BY SA

Im Idealfall sollte der Nachweis aus einer einzigen Nachricht des Beweisführers an den Verifizierer bestehen und öffentlich überprüfbar sein. In der Literatur ist dies als Beweis für sequentielle Arbeit bekannt. In dieser Dissertation erweitern wir dieses Konzept auf inkrementelle Beweise für sequentielle Arbeit, bei denen Rechenthreads ohne zusätzlichen Aufwand aufgeteilt oder an neue Beweisführer weitergegeben werden können. Dies trägt sehr lange Berechnungen oder möglichen Hardwareausfällen Rechnung, die die Lebensdauer eines einzelnen Beweisführers überschreiten. Vor dieser Arbeit waren Beweise für sequentielle Arbeit durch die Zeitkomplexität des Beweisführers von $T \quad \sqrt{T}$ und die Platzkomplexität $\sqrt{T}$ gekennzeichnet. Im Gegensatz dazu sind die hier angegebenen Konstruktionen im Wesentlichen optimal in dem Sinne, dass der Beweisführer eine in T lineare Zeitkomplexität und eine in T polylogarithmische Platzkomplexität aufweist. Unsere Schemata basieren ausschließlich auf symmetrischen Grundelementen und sind in der Praxis effizient.

Zeitschlossrätsel. Ein Zeitschlossrätsel erlaubt es, Geheimnisse für eine bestimmte Zeit T zu verkapseln, d.h. es ist möglich, Nachrichten so zu verschlüsseln, dass der Klartext nach Zeit T (für beliebiges T), aber nicht vorher wiederhergestellt werden kann. Dies kann als ein feinkörnigerer Begriff der probabilistischen Verschlüsselung angesehen werden, bei dem T immer exponentiell groß gesetzt wird. Die Anwendungen von Zeitschlossrätseln reichen von elektronischen Wahlen bis hin zu fairen Verträgen. Unglücklicherweise erfordern Szenarien, in denen mehrere Parteien beteiligt sind, dass man im schlimmsten Fall alle Rätsel, die während des Protokolls generiert werden, lösen muss. Abhängig von der Anzahl der Teilnehmer kann es unsinnig sein, anzunehmen, dass eine solche parallele Leistungsfähigkeit zur Verfügung steht. Um dieses Problem zu lösen, führen wir das Konzept der homomorphen Zeitschlossrätsel ein, bei denen es möglich ist, öffentlich Funktionen auf den Geheimnissen zu berechnen, ohne die entsprechenden Rätsel zu öffnen. Danach muss nur noch das Rätsel, das die Funktionsausgabe kodiert, gelöst werden, wodurch der Rechenaufwand der Parteien stark reduziert wird. In dieser Arbeit werden Zeitschlossrätsel definiert und instanziiert, die die homomorphe Auswertung verschiedener Funktionsfamilien zulassen. Im Anschluss wird gezeigt, wie homomorphe Zeitschlossrätsel als Eckpfeiler für Protokolle für elektronische Wahlen, Auktionen mit versiegelten Geboten, Mehrparteienmünzwurf und Vertragsunterzeichnung dienen.

Atomare Transaktionen. Gegeben sei das Szenario, in welchem eine Partei einige Münzen über eine Kette von Mittelsmännern an eine andere Partei weitergeben möchte. Offensichtlich ist es keine gute Strategie, die nächste Partei in der Kette zu bezahlen und zu hoffen, dass sie die Münzen schließlich vorwärts gibt. In dieser Dissertation wird das Konzept der Mehrschrittschlösser als Lösung für dieses Problem vorgestellt. Ein Mehrschrittschloss ermöglicht es, in einer Kette von Benutzern eine Reihe von Verträgen abzuschließen, die besagen:„Wenn eine bestimmte Bedingung vor der Zeit T erfüllt ist, zahlt A eine Münze an B.“ Der Empfänger (d.h. der letzte Knoten in der Kette) kennt eine Lösung, die er freigibt, wenn die Kette von Verträgen bei ihm ankommt. Dies löst eine Kettenreaktion aus, bei der jeder Knoten die Lösung berechnen kann, um seine linke Verbindung freizugeben, sobald die Bedingung für seine rechte Verbindung erfüllt ist. Mehrschrittschlösser werden in Verbindung mit einer Blockchain verwendet, um indirekte und schnelle Zahlungen zu ermöglichen und die

Atomarität der Transaktion in dem Sinne zu gewährleisten, dass entweder die gesamte Kette aktualisiert wird oder alle Verbindungen unangetastet bleiben. Darüber hinaus garantieren Mehrschrittschlösser ein gewisses Maß an Anonymität aus der Sicht der Zwischenknoten. Es wird gezeigt, wie man Mehrschrittschlösser aus einfachen Annahmen so instanziiert, dass sie mit der großen Mehrheit der Kryptowährungen, einschließlich Bitcoin, kompatibel sind.

Hintergrund und Tragweite dieser Dissertation

Diese Dissertation basiert auf Veröffentlichungen bei führenden Konferenzen der Kryptographie (EUROCRYPT, CRYPTO) sowie der Computersicherheit (NDSS). Des Weiteren wurden die Arbeiten bei diversen weiteren Workshops präsentiert und zur Präsentation bei dem Charles River Crypto Day at Microsoft Research (2019) sowie bei der Konferenz Advances in Financial Technologies (2019) als ein Höhepunkt der Forschung eingeladen. Eine der Veröffentlichungen befand sich unter den Finalisten im CSAW Wettbewerb für angewandte Forschung (2019). Die in dieser Dissertation entwickelte kryptographische Maschinerie wurde anschließend vom Lightning Network sowie KZen Network, zwei auf der Blockchain basierenden Bezahlnetzwerken, implementiert.

Inkrementelle Beweise für sequentielle Arbeit

Man stelle sich vor, jemand habe eine mögliche Lösung für die Goldbachsche Vermutung. Diese Person wäre sich fast sicher, dass ihr Ansatz fundiert ist, jedoch noch nicht bereit für eine Veröffentlichung, da weitere Prüfung notwendig wäre. Auf der anderen Seite bestünde die Gefahr, dass eine andere Partei das Resultat für sich beanspruchte, weshalb die Person einen Zeitstempel für ihre Entdeckung benötigte. Eine Lösung besteht darin, sich auf eine Drittstelle zu verlassen, die die Authentizität des Dokuments bescheinigt, wobei das Vertrauen in einen solchen Nachweis jedoch an das Vertrauen gebunden ist, das man in eine solche Stelle hat. Eine wünschenswertere Alternative ist ein universell verifizierbares kryptographisches Protokoll, das sich auf die Schwierigkeit ausgiebig untersuchter Probleme verlässt.

Ein Beweis für sequentielle Arbeit (engl. Proof of Sequential Work) (PoSW) [MMV13] erlaubt es einem Beweisführer $\mathcal{P}$, bei Eingabe einer bestimmten Aussage χ und eines Zeitparameters T, einen zweifelnden Verifizierer $\mathcal{V}$ davon zu überzeugen, dass seit der Festlegung von χ eine bestimmte Zeit $(1-\alpha)\cdot T$ (für einige feste Konstante α) verstrichen ist. Eine PoSW wird als nicht-interaktiv bezeichnet, wenn die Kommunikation zwischen $\mathcal{P}$ und $\mathcal{V}$ aus einer einzigen Nachricht $\mathcal{P} \xrightarrow{\pi} \mathcal{V}$ besteht. Eine PoSW muss *vollständig* sein, d. h. ein ehrlicher Beweisführer $\mathcal{P}$ muss immer in der Lage sein, den Verifizier $\mathcal{V}$ zu überzeugen, und *zuverlässig*, was bedeutet, dass keine bösartige Maschine in der Lage ist, $\mathcal{V}$ 1 in einer Zeit $< \alpha T$ ausgeben zu lassen. Um Effizienz zu gewährleisten, wird verlangt, dass der Beweis π in (möglicherweise paralleler) Zeit in etwa T und mit wenig Speicher, idealerweise $\log(T)$, berechenbar ist. Die Größe von π und die Zeit für die Verifizierung muss ebenfalls höchstens logarithmisch in T sein.

Inkrementeller PoSW. Eine natürliche Frage ist, ob Beweise für sequentielle Arbeit monolithische Objekte sind oder ob sie eine gewisse Flexibilität zulassen, z.B. die Verlagerung der Arbeitsbelastung auf einen anderen Beweisführer oder die Aufteilung der Berechnung eines Beweises zwischen zwei unabhängigen Parteien. Dies ist von besonderem Interesse, wenn Hardwareausfälle berücksichtigt werden, die bei der Berechnung von langlebigen Beweisen auftreten können. Für wichtige Berechnungen kann es sinnvoll sein, parallele Prozesse zu erzeugen und je nach den Umständen von einem zum anderen zu migrieren.

Die vorliegende Dissertation schließt diese Lücke, indem sie den Begriff der inkrementellen Beweise sequentieller Arbeit (engl. incremental Proofs of Sequential Work) (iPoSW) einführt. Alle diese Ergebnisse basieren auf der in [DLM19] veröffentlichten Arbeit. Vereinfacht gesagt, erlaubt ein iPoSW einem Beweisführer, den Beweis π für eine bestimmte Aussage χ und den Zeitparameter T zu berechnen und später die Berechnung fortzusetzen, um einen Beweis π' für den Zeitparameter $T \quad T'$ zu berechnen. Die Zeitkomplexität des zweiten Beweisführers ist unabhängig von T und seine Speicherkomplexität ist durch eine logarithmische Funktion in der Gesamtzeit begrenzt. Außerdem verlangen wir, dass π und π' identisch verteilt sind.

Aktueller Stand. Wir haben folgende triviale Konstruktion eines iPoSW aus beliebigem PoSW. Ausgehend von einem Beweis π für eine Aussage χ und dem Zeitparameter T, kann man diesen auf die Zeit $T \quad T'$ inkrementieren, indem man eine neue Aussage χ' ableitet (z.B. durch Hashen $\chi' \leftarrow \mathsf{H}(\chi,\pi)$) und einen neuen Beweis π' für χ' mit dem Zeitparameter T' berechnet. Der neue Beweis besteht aus dem Paar (π,π') und kann leicht überprüft werden, indem die Aussage χ' wieder berechnet wird und zwei parallele Instanzen des Verifikationsalgorithmus auf (χ,T,π) und (χ',T',π') ausgeführt werden. Diese naive Lösung erfüllt jedoch keine Kompaktheitsanforderung in dem Sinne, dass die Größe des Beweises linear mit der Anzahl der Inkremente wächst. Insbesondere gibt es keine Hoffnung, dass inkrementierte Beweise die gleiche Verteilung haben wie (nicht inkrementierte) Standardbeweise. Das ist vor allem dann nicht wünschenswert, wenn der Beweis stark fragmentiert ist oder wenn er an viele neue Beweisführer weitergegeben wird.

Außerdem lassen bekannte Instanzen von PoSW keine optimalen Parameter für die Zeit- und Speicherkomplexität des Beweisführers zu. Es sind zwei Hauptkandidaten bekannt:

(1) Das ursprüngliche Schema von Mahmoody, Moran und Vadhan [MMV13] basiert auf tiefenrobuste Graphen und Beweise können in Zeit und Raum proportional zu T berechnet werden.

(2) Die neueste Arbeit von Cohen und Pietrzak [CP18] führt ein neues kombinatorisches Objekt ein und schlägt eine Konstruktion vor, die es dem Beweisführer erlaubt, Speicher gegen zusätzliche Rechenzeit einzutauschen. Bei optimalen Parametern werden die Beweise in der Zeit $T \quad \sqrt{T}$ und mit dem Speicher $\sqrt{T}$ berechnet.

Unsere Ergebnisse. In dieser Dissertation definieren wir inkrementelle Beweise sequentieller Arbeiten und zeigen, wie sie mit im Wesentlichen optimalen Parametern instanziiert werden können, wobei die Berechnungskomplexität des Beweisführers linear in T wächst und seine Raumkomplexität polylogarithmisch in T. Dies schließt die Speicherlücke für den

Beweisführer und löst ein bislang offenes Problem. Wir schlagen zwei nicht-interaktive Konstruktionen vor, die auf der Kombination von Cohen-Pietrzak-Graphen [CP18] mit neuen Techniken für die Auswahl von Herausforderungsknoten basieren.

(1) Unser erstes Schema geht davon aus, dass der Beweisführer Zugang zu einem einzigen Prozessor hat und liefert Beweise der Größe $O(\log(T)^2)$.

(2) Unser zweites Schema geht davon aus, dass der Beweisführer $\log(T)$-viele parallele Prozessoren hat. Die Beweisgröße ist nur um einen Faktor von 9 größer als die von [CP18].

Wir betonen, dass die Annahme, dass $\log(T) \approx 100$, eine sichere Obergrenze für die Anzahl der für unsere zweite Konstruktion benötigten Prozessoren ist. Dies entspricht der Annahme der Existenz von 100 unabhängigen Kernen für die Berechnung einer Hashfunktion, was für moderne GPUs angemessen ist. Beide Schemata sind einfach zu implementieren und machen nur einen Blackbox-Aufruf an eine Hashfunktion (modelliert als ein Zufallsorakel).

Homomorphe Zeitschlossrätsel

Ein Zeitschlossrätsel [RSW96] erlaubt es, Nachrichten für die Zukunft zu verschlüsseln. Das bedeutet, dass ein Rätsel bezüglich eines geheimen Inputs s und eines Zeitparameters T generiert und sichergestellt wird, dass s vor den Augen eines polynomiell großen Angreifers versteckt wird, der in einer Zeit von (deutlich) weniger als T läuft. Das entscheidende Merkmal von Zeitschlossrätseln ist, dass die Anzahl der parallelen Prozesse des Angreifers keinen Einfluss auf die Geschwindigkeit hat, mit der s wiederhergestellt wird, d. h. das Lösen eines Rätsels ist eine inhärent sequentielle Operation. Dies steht im Gegensatz zur Standardauffassung von probabilistischer Verschlüsselung [GM82], bei der eine Senkung des Sicherheitsparameters des Schemas schnellere Algorithmen ermöglichen kann, die eine massive Parallelisierung ausnutzen. Ein weiterer wichtiger Aspekt von Zeitschlossrätseln ist, dass die Zeit, die zum Erstellen eines Rätsels benötigt wird, im Wesentlichen unabhängig von T ist. Das ist besonders dann wünschenswert, wenn ein Geheimnis über einen längeren Zeitraum, z.B. 10 Jahre, versteckt werden muss.

Eines der motivierenden Beispiele für die Nützlichkeit von Zeitschlossrätseln sind elektronischen Wahlen: Der Standardansatz, um (zumindest in der Anfangsphase) die Vertraulichkeit von Stimmen zu gewährleisten, besteht darin, dass sich jeder Benutzer auf seine Wahl commitet und alle Commitments gesammelt werden. In einer zweiten Auszählungsphase werden die Benutzer aufgefordert, den Inhalt ihrer Commitments (zusammen mit den privaten Zufallsmünzen) offenzulegen, was die Berechnung und Bekanntgabe des Wahlergebnisses ermöglicht. Dies lässt jedoch die Frage offen, wie mit Benutzern umgegangen werden soll, die in der ersten Phase gültige Commitments senden, aber in der folgenden Auszählungsphase nicht öffnen. Man könnte (i) den gesamten Vorgang wiederholen, wenn dies geschieht, oder (ii) solche Stimmen ignorieren. Die erstere Option ist eindeutig nicht durchführbar, da sie es einem böswilligen Nutzer ermöglicht, die Wahl im Wesentlichen kostenlos zu verhindern. Andererseits könnte letztere Lösung ausgenutzt werden, um das Endergebnis zu manipulieren: Ein Angreifer, der

den Netzwerkverkehr kontrolliert, könnte die Eröffnungen blockieren, die einem unerwünschten Kandidaten entsprechen, und dadurch der bevorzugten Seite einen Vorteil verschaffen.

Zeitschlossrätsel lösen diesen Konflikt auf elegante Weise auf, indem sie die Commitments als Versteckmechanismus für Stimmen ersetzen. Die Stimme von Benutzern, die ihre Zufallsmünzen nicht veröffentlichen, kann einfach durch Knacken des Zeitschlossrätsels bestimmt werden. Wenn der Zeitparameter so gesetzt wird, dass er um einen sicheren Wert länger als die Abstimmungsphase ist, wird sichergestellt, dass die Wahlen geheim gehalten werden, bis eine solche Phase vorbei ist. Leider ist diese Lösung nicht ohne zusätzliche Kosten möglich: Wenn eine große Anzahl von Wählern ihr Rätsel nicht öffnet, so muss die Auszählbehörde vor der Bekanntgabe des Wahlsiegers die gleiche Anzahl von Rätseln bruteforcen, was bedeutet, dass eine massive Anzahl an Parallelberechnungen erforderlich ist, um die Wahl in angemessenem Zeitraum abzuschließen. Bedenkt man, dass es Millionen von Stimmen geben kann, so wird das Problem offensichtlich. Es ist völlig plausibel, dass die für den Zeitschlossansatz benötigten Ressourcen von der Entscheidung für diese Lösung wegführen können. Sicherlich ist das Szenario der elektronischen Wahlen ein gutes Beispiel für die Skalierbarkeit von Zeitschlossrätseln, jedoch nicht der einzige Fall, in dem sie auftauchen. Jede andere Anwendung, an der eine große Anzahl von Benutzern beteiligt ist, wie z.B. versiegelte Auktionen oder Mehrparteienmünzwurf, stößt auf ähnliche Probleme. Wir vermuten, dass solche Einschränkungen eines der Haupthindernisse darstellen, die bisher die umfassende Einführung von Zeitschlossrätseln verhindert haben.

Homomorphe Zeitschlossrätsel. Das Hauptproblem bei auf Zeitschlossrätseln basierenden Lösungen ist, dass eine Partei viele Rätsel öffnen muss und dann eine Funktion über die Geheimnisse berechnen muss. Was wäre, wenn Funktionen zuerst *homomorph* berechnet und dann ein *einziges* Zeitschlossrätsel, das den Funktionsoutput beinhaltet, geknackt werden könnte? Dies würde die Hauptquelle der Ineffizienz der Protokolle, die Zeitschlossrätsel verwenden, auslöschen. In obigem Beispiel könnte die Auszählbehörde einfach den Gewinner der Wahl homomorph berechnen und dann ein einziges Rätsel lösen.

Von diesem Ziel motiviert, schlagen wir den Begriff des homomorphen Zeitschlossrätsels (engl. Homomorphic Time-Lock Puzzle) (HTLP) vor: Ein HTLP ist ein erweitertes Zeitschlossrätsel, das es jedem erlaubt, eine Funktion F über eine Menge an Rätseln homomorph zu berechnen, ohne unbedingt die kodierten Geheimnisse $(s_1,\ldots,s_n)$ zu kennen. Das resultierende Rätsel enthält die Funktionsausgabe $F(s_1,\ldots,s_n)$ und seine Schwere ist nicht von der Größe von F abhängig. Dieser Begriff kann als Gegenstück zur vollständig homomorphen Verschlüsselung [Ge09] für schwache Gegner gesehen werden, bei denen die Schwierigkeit mit abnehmendem Zeitparameter stark abnimmt. Wir betonen, dass die *Kompaktheit* des Algorithmus auch für HTLP eine entscheidende Voraussetzung ist: Wenn man sie ignoriert, so ist die triviale Lösung, die Eingaberätsel zu lösen und dann die Funktion über die Geheimnisse im Klartext zu berechnen, ausreichend.

Unsere Ergebnisse. In dieser Dissertation stellen wir den Begriff des homomorphen Zeitschlossrätsels vor, wir präsentieren formale Definitionen, schlagen verschiedene Schemata für verschiedene Klassen von Funktionen vor und zeigen ihre Nützlichkeit anhand

einiger realer Anwendungen auf. Alle folgenden Ergebnisse basieren auf den vom Autor veröffentlichten und in [MT19] vorgestellten Ergebnissen.

(1) Ein HTLP für lineare Funktionen, das die inhärente Sequentialität der wiederholten Quadrierung in Gruppen zusammengesetzter Ordnung annimmt. Dies ist die gleiche Annahme, die von Rivest, Shamir und Wagner für die Konstruktion des ersten (nicht-homorphen) Zeitschlossrätsels [RSW96] verwendet wurde.

(2) Ein multiplikativ-homomorphes Zeitschlossrätsel aus Standardannahmen in Gruppen zusammengesetzter Ordnung.

(3) Ein vollständig homomorphes Zeitschlossrätsel (oder HTLP für jede Funktion polynomieller Größe F) unter der Annahme, dass es eine ununterscheidbare Verschleierung für probabilistische Funktionen gibt [Ca15].

Die beiden erstgenannten Schemata basieren auf Standardannahmen (in Bezug auf die Faktorisierung großer Zahlen) und sind einfach und effizient genug, um in realen Anwendungen eingesetzt zu werden, während das letztere von einer stärkeren Maschinerie (der Existenz eines Programmverschleierers) ausgeht und als ein Machbarkeitsergebnis von HTLP für jede Funktion interpretiert werden soll. Wir untersuchen verschiedene Erweiterungen unserer grundlegenden Schemata, z. B. die Möglichkeit der Wiederverwendung des Setups für eine unbegrenzte Anzahl von Rätseln und Algorithmen zur Berechnung von Funktionen über Rätsel, die mit unterschiedlichen Härteparametern erzeugt wurden. Wir schlagen auch ein halbkompaktes HTLP für Programme mit unbegrenzter Verzweigungstiefe vor. Schließlich zeigen wir, wie verschiedene Arten von HTLP als Hauptbaustein verwendet werden können, um völlig sichere und *faire* Protokolle zu konstruieren, wie z. B.:

(1) Ein elektronisches Wahlverfahren und ein Protokoll für Auktionen mit versiegelten Geboten, welches eine beliebige Anzahl von Parteien unterstützt, die nach den ersten Phasen offline gehen, ohne dass ein erheblicher Rechenaufwand entsteht oder das Ergebnis des Protokolls verändert wird.

(2) Ein effizientes und skalierbares Mehrparteienmünzwurfprotokoll, das gegen selektive Fehler- und Dienstbehinderungsangriffe resistent ist. Ein solches Protokoll kann auf heutigen Maschinen ohne größeren Rechenaufwand ausgeführt werden.

(3) Eine Mehrparteienvertragsunterzeichnung, bei der ein kürzlich erzieltes Ergebnis zu aggregierten Unterschriften genutzt wird [HW18]. Die Fairness des Protokolls spiegelt sich darin wider, dass bei vorzeitigem Ausstieg einer Partei der Wiederherstellungsaufwand für ehrliche Parteien (unabhängig von deren Anzahl) etwa doppelt so groß ist.

Mehrschrittschlösser

Vollständig dezentralisierte Kryptowährungen wie Bitcoin [Na08] entstanden kürzlich als Alternative zu traditionellen Zahlungssystemen. Anstatt Zahlungen zu protokollieren und Kontostände auf einem zentralen Server zu verwalten, wird eine Datenbank über alle Anwender im

Netzwerk repliziert. Aktualisierungen werden in einer Blockchain gespeichert und durch einen Konsensalgorithmus validiert, an dem alle (möglicherweise misstrauischen) Maschinen im Netzwerk beteiligt sind. Dies begrenzt die Transaktionsrate des Systems auf natürliche Weise und schränkt die Verwendung dezentralisierter Währungen für kleine Transaktionen stark ein.

Zahlungskanalnetzwerke (engl. Payment Channel Networks) (PCNs) [Ma17] haben sich als die praktischste Lösung zur Abmilderung dieser Skalierbarkeitsprobleme erwiesen, da sie es ermöglichen, den Großteil der Zahlungen zwischen zwei Anwendern außerhalb der Blockchain durchzuführen. Vereinfacht gesagt öffnen zwei Anwender einen gemeinsamen Kanal, indem sie einige Münzen an einer gemeinsamen Adresse hinterlegen, die durch einen intelligenten Vertrag abgesichert ist. Spätere Zahlungen zwischen ihnen können dann durch die Vereinbarung eines neuen Saldos durchgeführt werden, ohne die Notwendigkeit, Informationen in die Blockchain hochzuladen. Dies erhöht den Transaktionsdurchsatz der Blockchain drastisch, da die einzige weitere Interaktion, die benötigt wird, das Schließen des Kanals und das Freigeben der entsprechenden Gelder ist. Die Verwendbarkeit des Netzwerkes kann weiter erhöht werden, indem man Transaktionen zwischen Nutzern erlaubt, die nicht direkt einen gemeinsamen Kanal teilen, aber über mehrere Schritte verbunden sind. Es gibt heute mehrere Umsetzungen dieser Idee, wobei der prominenteste Vorschlag der des Lightning Networks ist [PD16]. Jüngste Ergebnisse [Ma19b] haben jedoch die Unzulänglichkeit der derzeitigen Designprinzipien für PCNs offenbart, indem sie einen Angriff auf alle wichtigen Systeme demonstrieren, der es erlaubt, die Gebühren von ehrlichen Mittelsmännern auf einem Zahlungsweg zu stehlen. Dies drängt zu einer fundierten Untersuchung der zugrunde liegenden Mechanismen und zur Erlangung eines tieferen Verständnisses der Sicherheits- und Datenschutzgarantien für ein PCN.

Mehrschrittschlösser. Wir schlagen Mehrschrittschlösser (engl. Multi-Hop Locks) (MHL) vor, ein neues kryptographisches Primitiv, das als Eckpfeiler für die Gestaltung von PCNs dient. Eine MHL ermöglicht es einer Kette von Anwendern, ihre Verbindungen schrittweise zu sperren, bis der letzte Anwender erreicht ist. Ein solcher Anwender kann dann nach seinem Ermessen eine Kaskadenfreigabe der Sperren auslösen, z. B. wenn eine bestimmte Menge von Bedingungen erfüllt ist. Eine MHL sollte drei grundlegende Eigenschaften erfüllen.

(1) Atomarität: Wenn die ausgehende Kante eines Mittelsmannes in der Kette freigegeben wurde, dann kann ein solcher Anwender seine eingehende Kante jederzeit freigeben.
(2) Konsistenz: Jeder Zwischenanwender in der Kette kann seine eingehende Kante nur dann freigeben, wenn seine ausgehende Kante bereits vom vorherigen Anwender freigegeben wurde.
(3) Anonymität: Jede Sperre sollte keine Informationen über die Kette, zu der sie gehört, preisgeben.

MHLs erfassen den Kernmechanismus, der jedem PCN zugrunde liegt, und ermöglichen einen modularen Aufbau und das Verständnis größerer Systeme.

Unsere Ergebnisse. In dieser Dissertation motivieren, definieren und konstruieren wir MHLs. Genauer modellieren wir die Sicherheit der MHL im Rahmen der universellen Kombinierbarkeit (engl. Universal Composability) (UC) [Ca01], was erlaubt, über die

Verwendung von MHLs in einer größeren Umgebung nachzudenken. Dann schlagen wir zwei Instanziierungen für MHLs vor:

(1) Wir zeigen, dass linear homomorphe Einwegfunktionen ausreichen, um MHLs zu konstruieren, und positionieren damit dieses Primitiv in einem Bereich gut studierter Annahmen.

(2) Wir konstruieren MHLs in einer vertragsfreien Umgebung, wobei die Gültigkeit einer Einwegfunktion durch einfache Verifizierung einer ECDSA-Signatur überprüft werden kann. Dieser Ansatz ist von besonderem Interesse, da er die Verwendung von MHLs in Verbindung mit Blockchains ermöglicht, die keine turingmächtige Skriptsprache (wie z.B. Bitcoin) unterstützen. Vor unserer Arbeit wurde dies als ein offenes Problem in diesem Bereich betrachtet.

Schließlich untersuchen und diskutieren wir verschiedene Anwendungen unseres Primitivs: Das MHL kann in einer Blackbox-Art verwendet werden, um effizientere PCNs mit konkreten Datenschutzgarantien zu konstruieren und atomare Münztauschaktionen, sogar über verschiedene Währungen hinweg, zu ermöglichen. Eine Version unseres Schemas wurde durch das Lightning Network [Fr18] implementiert, um der in [Ma19b] aufgedeckten Sicherheitsverletzung entgegenzuwirken. Die oben genannten Ergebnisse basieren auch auf der (vom Autor) in [Ma19b] veröffentlichten Arbeit.

Literatur

[Ca01] Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols. In: 42nd FOCS. IEEE Computer Society Press, S. 136–145, Okt. 2001.

[Ca15] Canetti, R.; Lin, H.; Tessaro, S.; Vaikuntanathan, V.: Obfuscation of Probabilistic Circuits and Applications. In (Dodis, Y.; Nielsen, J. B., Hrsg.): TCC 2015, Part II. Bd. 9015. LNCS, Springer, Heidelberg, S. 468–497, März 2015.

[CP18] Cohen, B.; Pietrzak, K.: Simple Proofs of Sequential Work. In (Nielsen, J. B.; Rijmen, V., Hrsg.): EUROCRYPT 2018, Part II. Bd. 10821. LNCS, Springer, Heidelberg, S. 451–467, Apr. 2018.

[DLM19] Döttling, N.; Lai, R. W. F.; Malavolta, G.: Incremental Proofs of Sequential Work. In (Ishai, Y.; Rijmen, V., Hrsg.): EUROCRYPT 2019, Part II. Bd. 11477. LNCS, Springer, Heidelberg, S. 292–323, Mai 2019.

[Fr18] Fromknecht, C.: tpec: 2P-ECDSA Signatures, Github repository, `https://github.com/cfromknecht/tpec`, 2018.

[Ge09] Gentry, C.: Fully homomorphic encryption using ideal lattices. In (Mitzenmacher, M., Hrsg.): 41st ACM STOC. ACM Press, S. 169–178, Mai 2009.

[GM82] Goldwasser, S.; Micali, S.: Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In: 14th ACM STOC. ACM Press, S. 365–377, Mai 1982.

[HW18] Hohenberger, S.; Waters, B.: Synchronized Aggregate Signatures from the RSA Assumption. In (Nielsen, J. B.; Rijmen, V., Hrsg.): EUROCRYPT 2018, Part II. Bd. 10821. LNCS, Springer, Heidelberg, S. 197–229, Apr. 2018.

[Ma17] Malavolta, G.; Moreno-Sanchez, P.; Kate, A.; Maffei, M.; Ravi, S.: Concurrency and Privacy with Payment-Channel Networks. In (Thuraisingham, B. M.; Evans, D.; Malkin, T.; Xu, D., Hrsg.): ACM CCS 2017. ACM Press, S. 455–471, Okt. 2017.

[Ma19a] Malavolta, G.: Cryptographic Clocks and Applications, doctoralthesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2019, S. 108, URL: https://nbn-resolving.org/urn:nbn:de:bvb:29-opus4-112571.

[Ma19b] Malavolta, G.; Moreno-Sanchez, P.; Schneidewind, C.; Kate, A.; Maffei, M.: Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability. In: NDSS 2019. The Internet Society, 2019.

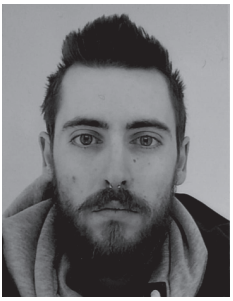
[MMV13] Mahmoody, M.; Moran, T.; Vadhan, S. P.: Publicly verifiable proofs of sequential work. In (Kleinberg, R. D., Hrsg.): ITCS 2013. ACM, S. 373–388, Jan. 2013.

[MT19] Malavolta, G.; Thyagarajan, S. A. K.: Homomorphic Time-Lock Puzzles and Applications. In (Boldyreva, A.; Micciancio, D., Hrsg.): CRYPTO 2019, Part I. Bd. 11692. LNCS, Springer, Heidelberg, S. 620–649, Aug. 2019.

[Na08] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf, 2008.

[PD16] Poon, J.; Dryja, T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, Technical Report, https://lightning.network/lightning-network-paper.pdf, 2016.

[RSW96] Rivest, R. L.; Shamir, A.; Wagner, D. A.: Time-Lock Puzzles and Timed-Release Crypto./, 1996.



Giulio Malavolta ist derzeit Postdoc an der UC Berkeley und der Carnegie Mellon University. Zuvor war er als wissenschaftlicher Mitarbeiter am Simons Institute for the Theory of Computing tätig. Im Juni 2019 schloss er seine Promotion an der Friedrich-Alexander-Universität Erlangen-Nürnberg ab. Er hat einen Master in Informatik an der Universität des Saarlandes, den er im Dezember 2016 abgeschlossen hat. Sein Forschungsinteresse erstreckt sich auf die Theorie und die Anwendungen der Kryptographie und Computersicherheit. Seine Arbeit wurde auf führenden Konferenzen in den Bereichen Kryptographie (CRYPTO, EUROCRYPT), Berechenbarkeitstheorie (FOCS) und Systemsicherheit (S&P, CCS, NDSS) veröffentlicht.