universität innsbruck

Prinzipien von Blockchain-Systemen

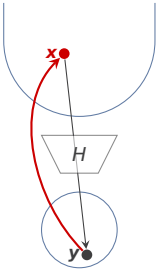
Infrastruktur für Blockchain-Systeme

Rainer Böhme

Kryptographische Hash-Funktionen

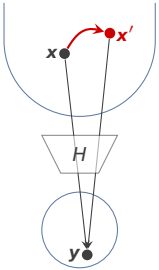
$$H : \{0,1\}^* \to \{0,1\}^\ell$$

Urbildresistenz



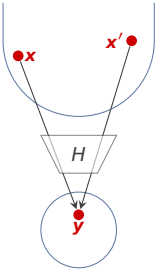
Schwer: Gegeben $H(\boldsymbol{x})$, finde $\boldsymbol{x}$.

2.-Urbild-Resistenz



Schwer: Gegeben $\boldsymbol{x}$, finde $\boldsymbol{x}'$ sodass $H(\boldsymbol{x}') = H(\boldsymbol{x})$.

Kollisionsresistenz



Schwer: Finde $(\boldsymbol{x}, \boldsymbol{x}')$ sodass $H(\boldsymbol{x}) = H(\boldsymbol{x}')$.

Praktische Hash-Funktionen

Aktuell empfohlen

- SHA-3-Familie („Keccak“, Bertoni et al. 2011) mit $\ell = 256$ — Einsatz in Ethereum
- SHA-2-Familie (NIST 2002), z. B. SHA-256 mit $\ell = 256$ — Einsatz in Bitcoin
- RIPEMD-160 (Dobbertin et al. 1996) mit $\ell = 160$ — Einsatz in Bitcoin

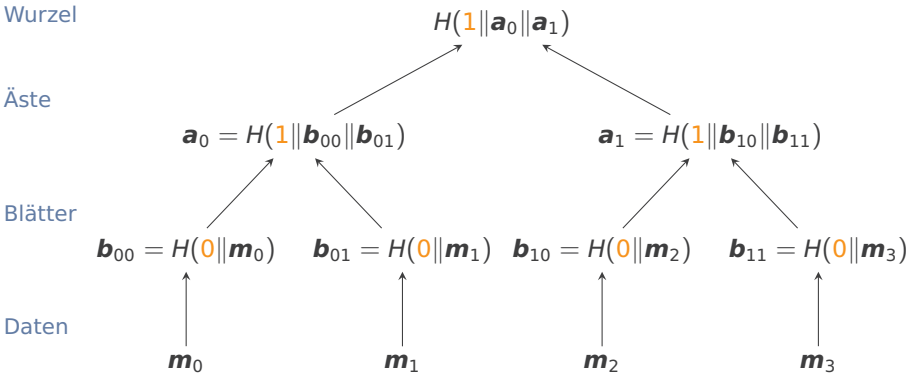
Bekannt aber unsicher

- MD4 (1990, $\ell = 128$)
- MD5 (1991, $\ell = 128$)
- SHA-0 (1993, $\ell = 160$)
- SHA-1 (1995, $\ell = 160$)

Prinzip Hoffnung: Die Sicherheit dieser Hash-Funktionen ist nicht bewiesen.

Merkle-Baum

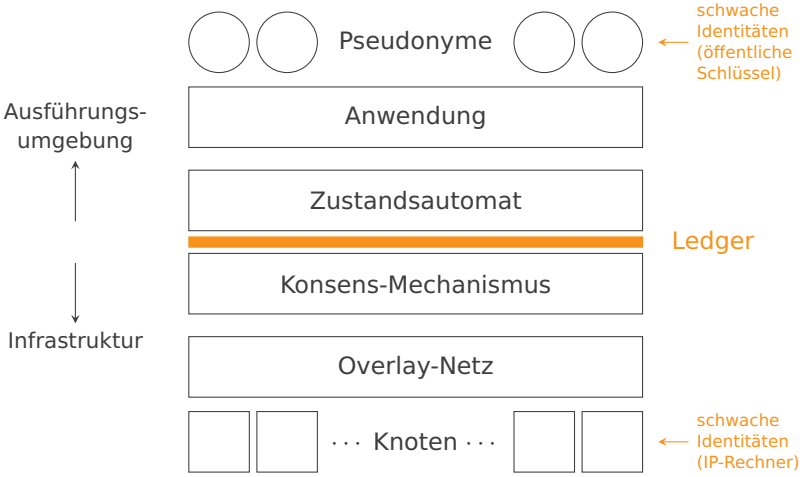
Eine authentisierte Datenstruktur auf Basis von kryptographischen Hash-Funktionen



Die Überprüfung, ob $\boldsymbol{m}_i$ Teil des Baums ist ($0 \leq i < n$), benötigt $O(\log n)$ Schritte.

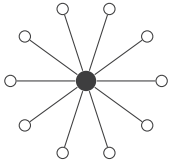
Merkle 1979

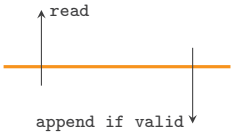
Schichtmodell für Blockchain-Systeme



Ideale Infrastruktur als Zielvorstellung

(auch als Abstraktionsmodell zum Studium der Ausführungsumgebung geeignet)



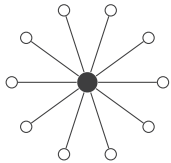


Annahme eines zentralen Koordinators:

1. verfügt über einen sicheren (vertraulich, nicht manipulierbar) Kanal mit jedem Knoten;
2. pflegt eine öffentliche Datenbank von zeitlich geordneten Transaktionen, für die er die einzige autoritative Quelle darstellt;
3. empfängt neue Transaktionen von allen Knoten, serialisiert, verifiziert und schreibt gültige Transaktionen in die Datenbank;
4. arbeitet unparteiisch, vollkommen altruistisch und ist stets völlig ohne Latenz verfügbar.

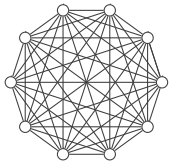
Realistisches Kommunikationsmodell

Simulation des Koordinators mit schwach identifizierten Knoten



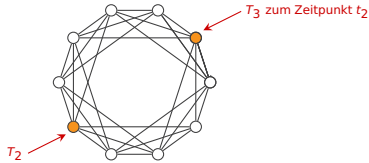
Stern-Topologie

(starke Abstraktion)



synchroner
Broadcast-Kanal

(schwächere Abstraktion)



asynchrones
Overlay-Netz

(realitätsnah)

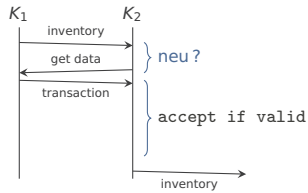
→ Signallaufzeiten im Netz sind eine Ursache für Inkonsistenz.

Vgl. VU Prinzipien von Blockchain-Systemen, Kapitel „Einführung, Grundlagen, Schichtmodell“, 5. März 2020, S. 16

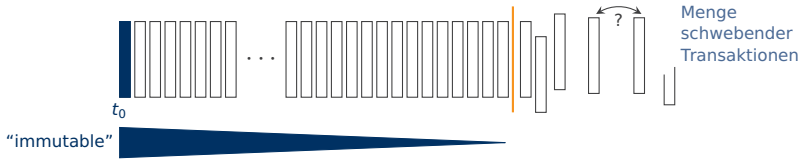
Zustandsreplikation

Ziel: eventual consistency (abgeschwächte Konsistenz)

Ein Gossip-Protokoll repliziert die Datenbank in jedem Knoten. (Deshalb muss sie öffentlich sein.)

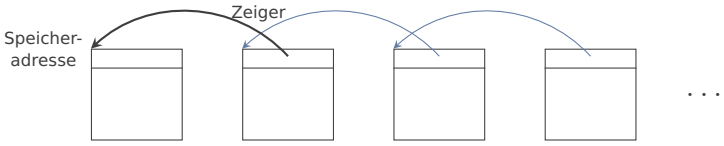


Eine authentisierte Datenstruktur hält alte Einträge überprüfbar konsistent.

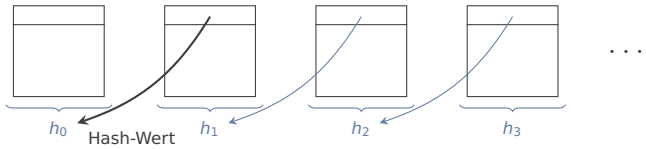


Authentisierte Datenstruktur

Verkettete Liste

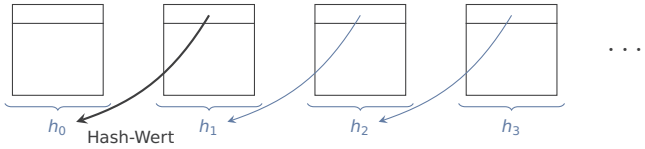


Hash-verkettete Liste

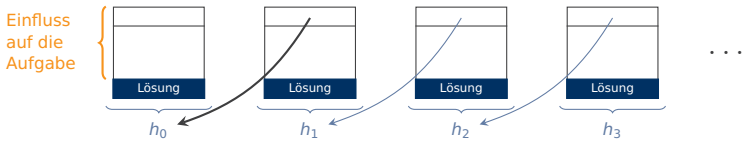


Authentisierte Datenstruktur

Hash-verkettete Liste (W)

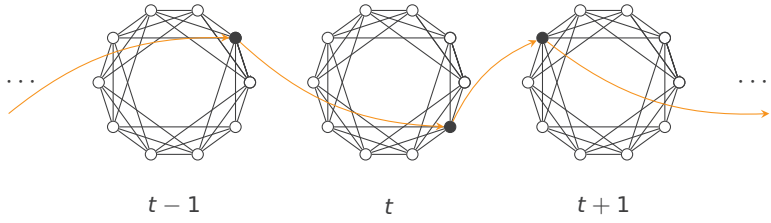


Hash-verkettete Liste mit Arbeitsnachweis



Konsens-Mechanismus

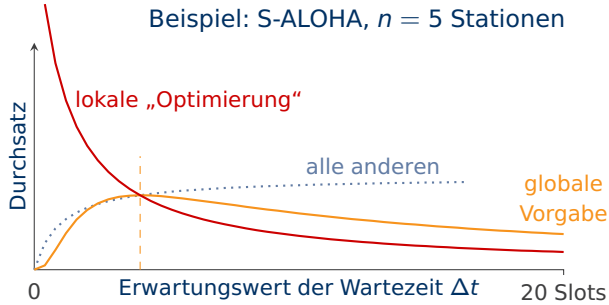
Einigung auf einen Zustandsübergang mittels “leadership selection”



Aufgaben des Leaders:

1. Blockbildung: Serialisierung schwebender Transaktionen in gültige Folge
2. Den neuen Block mit Hash-Verkettung und Arbeitsnachweis an Historie anfügen
3. Den neuen Block über das Gossip-Protokoll im Netz bekannt machen

Self-enforcing Rate Limit – Exkurs



Analoges Problem: Medienzugriffskontrolle

- Jeder möchte zuerst senden.
- Einhaltung der zufälligen Wartezeit nicht durchsetzbar.

Vgl. VO Rechnernetze und Internettechnik, Kapitel „Multiplexing und Medienzugriff“, 28. März 2019, S. 11 ff.

Self-enforcing Rate Limit

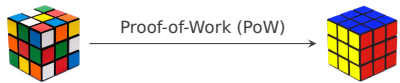
(dt. selbstdurchsetzende Sendedrosselung)

Problem: Der Leader ist mächtig. Er kann z. B. Transaktionen bevorzugen.

- Jeder möchte Leader werden.
- Schwache Identitäten $\rightarrow$ Teilnahme mit vielen Knoten (Sybil-Angriff)

Lösung

Drosselung der Teilnahme an der Wahl des Leades durch Koppelung an eine in der Realwelt knappe Ressource (Rechenkapazität): Leistung eines Arbeitsnachweises

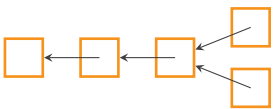


Vgl. VU Prinzipien von Blockchain-Systemen, Kapitel „Einführung, Grundlagen, Schichtmodell“, 5. März 2020, S. 7

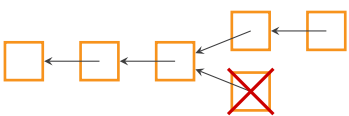
Auflösung von Konflikten auf Blockebene

Leader lösen Konflikte auf Transaktionsebene im Zuge der Blockbildung.

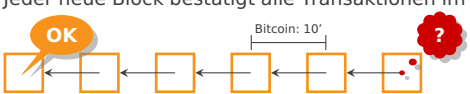
Aber: Konflikte können auf Blockebene auftreten, wenn mehrere Blöcke gleichzeitig gefunden werden . . .



. . . und verschwinden, wenn alle Knoten nur den Ast akzeptieren, der insgesamt die meiste Arbeit nachweist.



Jeder neue Block bestätigt alle Transaktionen im Ast und macht sie schwerer ersetzbar.



Ein **Regelkreis** passt die Schwierigkeit des Arbeitsnachweises an die gesamte Rechenleistung des Netzes an. Die Blockrate muss deutlich langsamer sein als die Signallaufzeit.

Auswahl geeigneter Arbeitsnachweise

Grundsätzliche Anforderungen

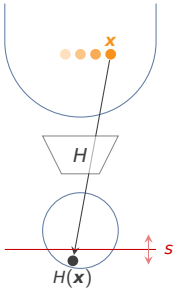
1. Leicht adjustierbare Schwierigkeit
2. Aufgabe muss datenabhängig sein
3. Lösung muss leicht überprüfbar sein ($\rightarrow$ Ansatz über Trial-and-error)
4. Aufgabe und Lösung sollten nicht viel Speicherplatz benötigen
5. Schwierigkeit darf nicht offensichtlich von Wissen abhängen (d. h. keine „Falltüren“)

Gegenstand aktueller Forschung

- Geringer Vorteil durch Spezialhardware (ASICs) Bsp. Ethash
- Schwer delegierbar
- Nützlich (über die Sicherung des Ledgers hinaus)
- An andere Ressource als Rechenleistung bzw. Energie gekoppelt

Arbeitsnachweis bei Bitcoin

Partielle Invertierung der Hash-Funktion: $H(\boldsymbol{x}) \triangleq \text{SHA256}\,(\text{SHA256}\,(\boldsymbol{x}))$



Konstruktion einer moderat-harten Funktion

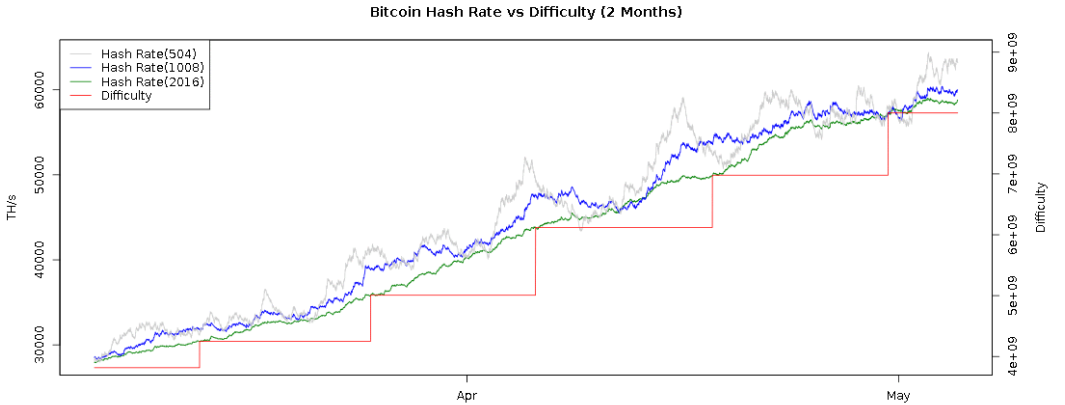
Für einen aus der **Schwierigkeit** gegeben Schwellwert s, finde nonce_i, sodass $H(\boldsymbol{x}_i) < s$ mit

$$\boldsymbol{x}_i = (\text{SHA256}\,(\boldsymbol{x}_{i-1}) \,\|\text{Block-Konstanten}\|\text{nonce}_i).$$

- i: Blockhöhe
- nonce: “number used once”, 32 Bit
- Die Block-Konstanten enthalten s und die Wurzel eines Merkle-Baums aller Transaktionen in Block i.
- Beste bekannte Lösung durch Ausprobieren.

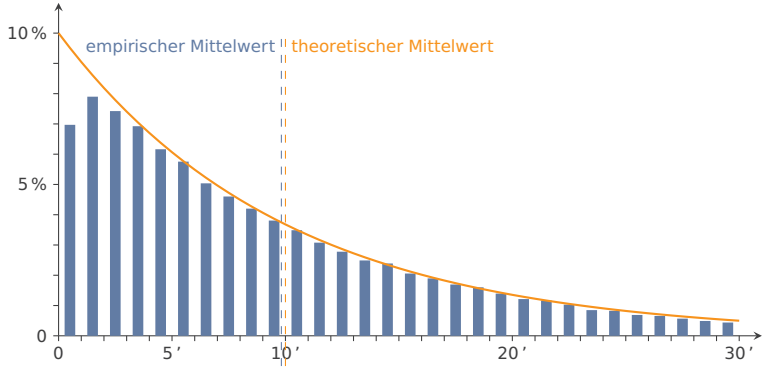
Schwierigkeit des Arbeitsnachweises

Daten zur Beobachtung des Regelkreises



Quelle: https://bitcoinwisdom.com/bitcoin/difficulty vom 4. Mai 2014

Verteilung der Zeitintervalle zwischen Blöcken



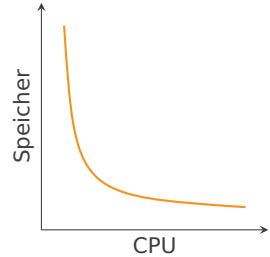
Risiko eines Konflikts auf Blockebene: $P(\Delta t = 5\,'') = 0.8\,\%$, $P(\Delta t = 12\,'') = 2.0\,\%$, ...

Daten: Timestamps in der Bitcoin-Blockchain, Jan–Dez 2014

Arbeitsnachweis bei Ethereum

Entwurfsziel für den Algorithmus Ethash: ASIC-unfreundlich

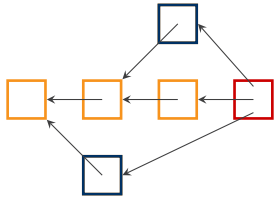
- **Idee:** Lösung der Aufgabe erfordert viel Speicher
- Alle 30.000 Blöcke wird ein 1 GB großer directed acyclic graph (DAG) erstellt.
- Arbeitsnachweis durch Nachschlagen und Ausprobieren
- **Kniff:** Verifikation ohne DAG effizient möglich



Zielgröße für Zeitintervall zwischen zwei Blöcken: **15 Sekunden**

Behandlung von Konflikten auf Blockebene

Ethereum nutzt eine Modifikation des GHOST-Protokolls
um den Schaden durch lange Signallaufzeiten zu begrenzen.



„Onkel“

„Kind“ bzw. „Neffe“

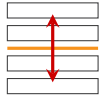
„Onkel“

Spezielle Regeln

- Maximal 2 Onkel pro Block
- Onkel können maximal 7 Blöcke alt sein
- Anreiz zur Berücksichtigung von Onkeln

Zohar, A., Sompolinsky, Y.. Accelerating Bitcoin’s Transaction Processing. Fast Money Grows on Trees, Not Chains. *IACR Cryptology ePrint Archive*, Nr. 881, 2013. https://eprint.iacr.org/2013/881.pdf

Anreize zur Bereitstellung der Infrastruktur



Ein öffentlicher Ledger hat die Eigenschaften eines **öffentlichen Gutes**.

- **Kosten**: insb. Arbeitsnachweis, getragen von den Knoten
- **Nutzen**: anwendungsspezifisch, realisiert von den Pseudonymen
- **Diskrepanz** in Wert, Zeitpunkt und Parteien !

Ökonomische Beziehung zwischen den Schichten

Blockchain-Systeme benötigen ein Zahlungsmittel, damit die Pseudonyme die Knoten kompensieren können.

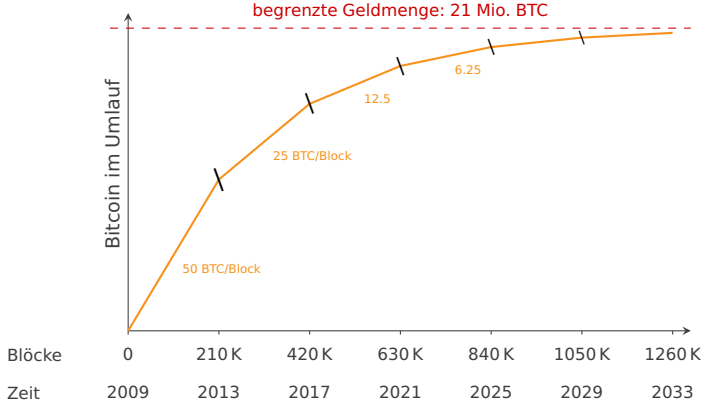
Zwei Varianten (auch in Kombination):

1. Geldschöpfung (“minting”) $\rightarrow$ alle Konten bezahlen indirekt durch Verwässerung
2. Transaktionssteuer (“fee”) $\rightarrow$ Individuen bezahlen für Schreibzugriff

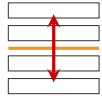
Bemerkung Geldschöpfung ist oft im Protokoll festgelegt, während die Transaktionssteuer (im Prinzip) einem Marktmechanismus zur Laufzeit unterliegt.

Belohnung beim „Block-Minen“

Beispiel: Bitcoin



Zusammenspiel



Die Knoten der Infrastruktur überprüfen regelmäßig:

1. Die Integrität des Ledgers: Hash-Verkettung + Arbeitsnachweis
2. Die Gültigkeit aller Blöcke, typischerweise definiert als
   - Korrektheit der Block-Konstanten und
   - Gültigkeit aller Transaktionen in der vorliegenden Reihenfolge

Technische Beziehung zwischen den Schichten

Um Transaktionen zu verifizieren **implementiert** die Infrastruktur die Ausführungsumgebung und **evaluiert** die in den Transaktionen kodierte Logik.

Logische Konsequenzen

- Die Ausführungsumgebung muss in jedem Zustand deterministisch sein.
- Aufgaben, die Zufall benötigen (Schlüsselgenerierung) müssen „off-chain“ bleiben.

Existenz von Mining-Pools

„Solo-Mining lohnt sich nicht mehr . . . “

Phänomen

- Miner schließen sich zusammen um PoW-Aufgabe gemeinsam zu lösen
- Aufteilung der Gewinne proportional zur beigetragenen Leistung
- Organisation i. d. R. durch zentralen Pool-Manager als Teil des Ökosystems

Erklärungsansatz

- Pool-Manager erhält Anteil als Gebühr und hält Reserve
- Warum akzeptieren Miner einen geringeren Ertrag ?
- Weil sie sich geringere Varianz der Auszahlung etwas kosten lassen.
  (Der gleiche Grund, aus dem wir Versicherungen abschließen.)

→ **Modellierung dieses Verhaltens als Risikoaversion**

Entscheidungsproblem

Alternative 1

mit Sicherheit

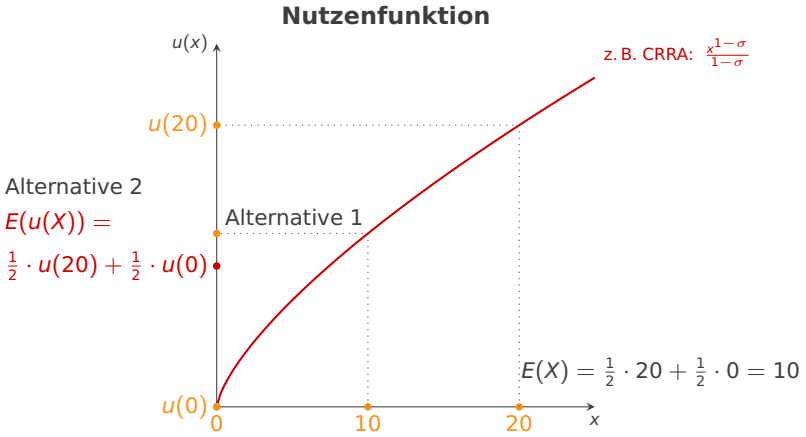
Alternative 2

mit 50 % Wahrscheinlichkeit

(sonst nichts)

Risikoaversion

Ökonomen modellieren Präferenzrelationen auf einer imaginären Nutzen-Skala



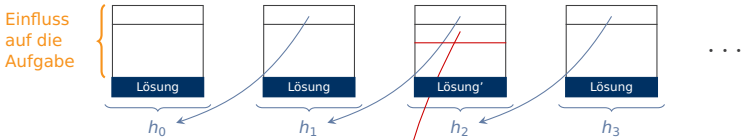
Merged Mining

Recycling des Mining-Aufwands zur Sicherung mehrerer Blockchain-Systeme

Host-Blockchain

nimmt Commitment z. B. in Transaktion auf

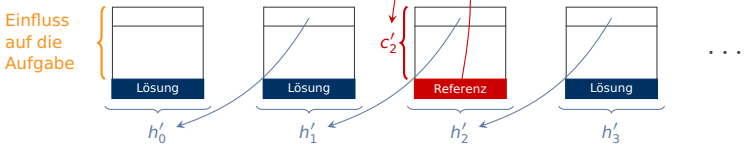
Bsp. Bitcoin



Gast-Blockchain

akzeptiert Referenz auf passende Lösung

Bsp. Namecoin



Syllabus

05.03.20	1. Einführung und Grundlagen	
23.04.20	2. Infrastruktur für Blockchain-Systeme	
30.04.20	3. Transaktionslogik in Bitcoin und Ethereum	
07.05.20	Übung: Blockchain-Analyse mit BlockSci	(Martin Plattner)
14.05.20	4. Datenschutz und Sicherheit	
28.05.20	5. Skalierbarkeit, Off-Chain-Transaktionen, Governance	
04.06.20	6. Wiederholung, Fragestunde	
18.06.20	Übung: Ethereum-Programmierung mit Solidity	(Michael Fröwis)
25.06.20	Klausur	

Stand: 23. April 2020. Änderungen vorbehalten.