22. Nachwuchswissenschaftler*innenkonferenz (NWK)

https://doi.org/10.52825/ocp.v2i.174

Publiziert: 15. Dez. 2022

Existenznachweise für Daten in unternehmensübergreifenden Blockchain-Netzwerken

Erik Neumann[1*][https://orcid.org/0000-0002-4483-589X]

[1] Fakultät Angewandte Computer- und Biowissenschaften, Hochschule Mittweida

* Korrespondenz: neumann3@hs-mittweida.de

Kurzfassung. Das Projekt *safe-UR-chain* befasst sich mit der manipulationssicheren Speicherung von Produktionsdaten und deren Austausch über Unternehmensgrenzen hinweg. Das System basiert auf „verstrickten“ Blockchains, wobei jedes Unternehmen eine eigene Blockchain führt und in diese sporadisch Block-Hashes aus den Blockchains anderer Unternehmen integriert. In diesem System soll der Austausch von Produktionsdaten ermöglicht werden. Um die Manipulationsfreiheit dieser Daten nachzuweisen, wird ein effizienter, kryptografischer Akkumulator auf Block-Basis verwendet, der Mitgliedschaftsbeweise für Blöcke innerhalb der Blockchain ermöglicht. Somit kann die Existenz von Blöcken zwischen den zuvor Synchronisierten Block-Hashes belegt werden. Auf Basis dieses Beweises kann gezeigt werden, dass Daten aus der externen Blockchain in einem gewissen Block enthalten sind.

1. Einleitung

Die schrittweise Vernetzung und Digitalisierung von Produktionssystemen hat es Unternehmen ermöglicht, entlang der Wertschöpfungskette effizienter miteinander zusammenzuarbeiten. Neben der damit verbundenen Produktivitätssteigerung, ermöglicht diese Vernetzung jedoch auch neue Angriffe durch Cyberkriminelle [1]. Einer dieser möglichen Angriffe besteht in der Manipulation produktionsnaher Daten, durch die beispielsweise spätere Rückrufaktionen gestört werden können.

Um zu erforschen, wie gegen diese Kategorie von Angriffen vorgegangen werden kann, wurde 2019 das Projekt *safe-UR-chain* ins Leben gerufen. In diesem arbeiten Unternehmen und Forschungseinrichtungen an einem Blockchain-basierten System zur Speicherung von produktionsnahen Daten, sowie deren Austausch über Unternehmensgrenzen hinweg. Ziel des Projektes ist, Daten in diesem System durch die Verwendung der Blockchain-Technologie gegen nachträgliche Veränderung zu schützen [2].

2. Bestehendes System

Um die nachträgliche Manipulation von Daten zu verhindern, werden die Produktionsdaten in einer Blockchain (s. Abb. 1) gespeichert. Dafür werden die Daten, die innerhalb eines Unternehmens anfallen, zuerst in dessen Netzwerk verteilt. Diese Daten werden dann regelmäßig in sog. *Blöcken* zusammengefasst. Diese Blöcke enthalten einen Hash-Wert über alle in ihnen enthaltenen Daten, der durch die Verwendung eines *Merkle Trees* [3] errechnet wird. Mit diesem Hash-Wert kann die Mitgliedschaft jeglicher Daten im Block nachgewiesen werden. Mit dem Hash-Wert der Daten und einigen Metadaten (z.B. Zeitstempel), kann ein Hash-Wert für den gesamten Block erzeugt werden. In diesen Block-Hash geht auch der Hash des vorhergehenden Blockes ein. Diese Verknüpfung macht die Manipulation von Daten innerhalb der

Blockchain unmöglich, da sich hierdurch der Daten-Hash eines Blockes ändern würde, was ebenso zur Änderung des Block-Hashes und damit zum Bruch der Kette führen würde, weil Nachfolger-Blöcke auf den Hash ihres Vorgängers verweisen [4].

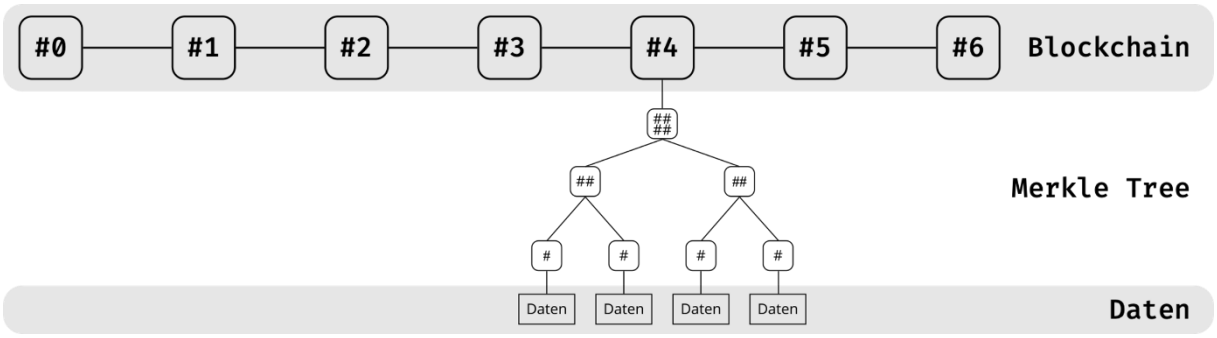

Abb. 1: Blockchain-Struktur mit Nutzdaten innerhalb eines Merkle-Trees

Die Blockchain wird im Unternehmensnetzwerk redundant auf vielen Computern (sog. *Nodes*) gespeichert und mithilfe eines Konsensverfahrens wird sichergestellt, dass die Blockchains auf all diesen Nodes langfristig die gleichen Blöcke enthalten.

Da die beteiligten Unternehmen physisch voneinander getrennt sind, ist die Nutzung einer zwischen Unternehmen geteilten Blockchain nicht möglich, da hierfür ggf. sehr große Datenmengen zwischen den Unternehmen synchronisiert werden müssten. Aus diesem Grund führt jedes Unternehmen eine eigene Blockchain für Produktionsdaten. Um sicherzustellen, dass Daten in den lokalen Blockchains nicht verändert werden können, synchronisieren Unternehmen einige der lokal erzeugten Block-Hashes untereinander und schreiben externe Block-Hashes in ihre lokale Blockchain. Damit wird die Veränderung von Daten immer bis hin zum letzten synchronisierten Block-Hash verhindert [5] (s. Abb. 2).

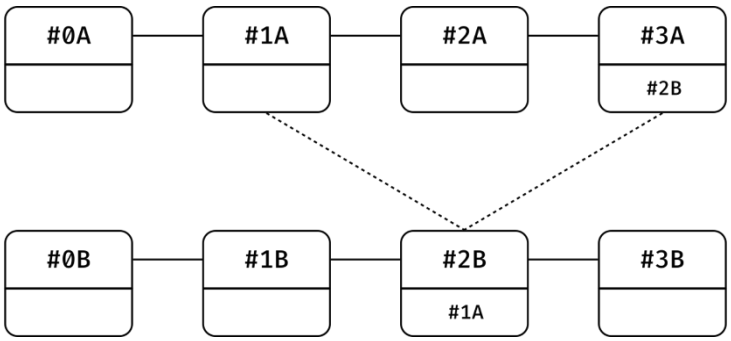

Abb. 2: „Verstrickte“ Blockchains zweier Unternehmen; Externe Block-Hashes werden als Daten in der jeweils lokalen Blockchain gespeichert

Eine nachträgliche Veränderung von Produktionsdaten kann im System somit erkannt werden. Nun soll ein manipulationssicherer Datenaustausch zwischen den Unternehmen ermöglicht werden. „Manipulationssicher“ bedeutet in diesem Kontext, dass das Unternehmen, das die Daten bereitstellt, beweisen kann, dass die Daten in seiner lokalen Blockchain existieren.

3. Problembeschreibung

Die Existenz von Daten innerhalb eines Blockes kann bewiesen werden, indem zuerst der Mitgliedschaftsnachweis innerhalb des Merkle Trees für den Block geführt wird. Danach kann mithilfe des Block-Hashes belegt werden, dass die Daten in dem Merkle Tree enthalten sind, der mit diesem Block assoziiert ist. Der Beweis, dass Daten in Blöcken enthalten sind, deren

Block-Hashes zuvor über Netzwerkgrenzen hinweg geteilt wurden, ist einfach möglich, da jegliche Daten für die Überprüfung des Beweises bereits in der anderen Blockchain enthalten sind (Block-Hashes).

Beweise für Daten, die in Blöcken mit nicht-synchronisierten Hashes enthalten sind, gestalten sich schwieriger. Hierfür muss zuerst gezeigt werden, dass der Block-Hash, der für die Überprüfung des Daten-Beweises notwendig ist, tatsächlich in der Blockchain des beweisenden Unternehmens enthalten ist.

Für den Beweis der Existenz von Daten wird somit ein Schema benötigt, dass anhand der vorher synchronisierten Block-Hashes einen Existenznachweis für andere Blöcke in der Blockchain eines bestimmten Unternehmens ermöglicht.

4. Lösung

Um die Existenz eines Blockes innerhalb der Blockchain nachzuweisen, können die Verknüpfungen zwischen Blöcken herangezogen werden. Da jeder Block auf seinen Vorgänger verweist, kann für jeden synchronisierten Block-Hash belegt werden, welcher Block-Hash ihm vorangeht. Und da dieser wiederum auf seinen vorhergehenden Block-Hash verweist, kann die Mitgliedschaft jedes Blockes in der Kette belegt werden. Dieser Vorgang kann wiederholt werden, bis die Kette rückwärts, bis zu einem weiteren bereits synchronisierten Block-Hash nachgewiesen ist. Jeder Hash zwischen diesen beiden Blöcken kann dann zum Nachweis der Existenz von Daten in den jeweiligen Blöcken genutzt werden. Dieses Vorgehen erlaubt einen nachvollziehbaren Beweis, allerdings verlangt es auch die Übertragung von ggf. sehr vielen Daten zwischen den Unternehmen, was durch die getrennten Blockchains ursprünglich verhindert werden sollte.

Um einen effizienteren Beweis zu ermöglichen, kann ein sog. *Akkumulator* verwendet werden. Eine solche Datenstruktur erlaubt den Mitgliedschaftsbeweis für Daten innerhalb eines Sets, wobei der Akkumulator selbst kleiner ist als das Set selbst. Mit einem Akkumulator kann also auch dargelegt werden, dass ein bestimmter Block im Set aller Blöcke (d.h. der Blockchain) enthalten ist. Wenn die Block-Datenstruktur so erweitert wird, dass jeder Block einen Akkumulator-Wert enthält, mit dem sich die Mitgliedschaft aller seiner Vorgänger in der Blockchain beweisen lässt, können auch Beweise für die Manipulationsfreiheit von übertragenen Daten erbracht werden.

Ein Beispiel für einen Akkumulator ist der bereits erwähnte Merkle Tree, dieser kommt jedoch für den Nachweis von Blöcken nicht infrage, da dieser pro Block über die gesamte Blockchain neu erstellt werden müsste. Stattdessen kommt ein Akkumulator zum Einsatz, der analog zum „Hochzählen“ einer Binärzahl funktioniert und intern Wurzeln von Merkle-Trees mit immer höherer Kapazität verwendet.

Der Akkumulator wird mit einem ersten Element initialisiert, dessen Hash an Position 0 gespeichert wird. Kommt ein zweites Element hinzu, wird sein Hash gebildet und mit dem Hash des ersten Elementes zusammengelegt. Aus dem resultierenden Daten wird wieder ein Hash gebildet und an der nächsten Position gespeichert. An Pos. 1 ist nun im Grunde ein Merkle Tree (bzw. dessen Wurzel), der das erste und zweite Datum enthält. Die Merkle Trees haben immer die Kapazität von 2^n (n = Position/Index), also die doppelte Kapazität ihres Vorgängers [6].

Weitere Elemente werden auf die gleiche Art eingefügt, Bäume werden immer zusammengeführt, um einen Baum mit der doppelten Kapazität zu erhalten (s. Abb. 3). Die Wurzeln der Merkle-Trees bilden den *Akkumulator-Wert*, mit dem Beweise über die Mitgliedschaft und Position von Blöcken innerhalb der Blockchain möglich werden. Die zugrundeliegenden Daten werden nicht gespeichert, nur die Wurzeln der Merkle Trees.

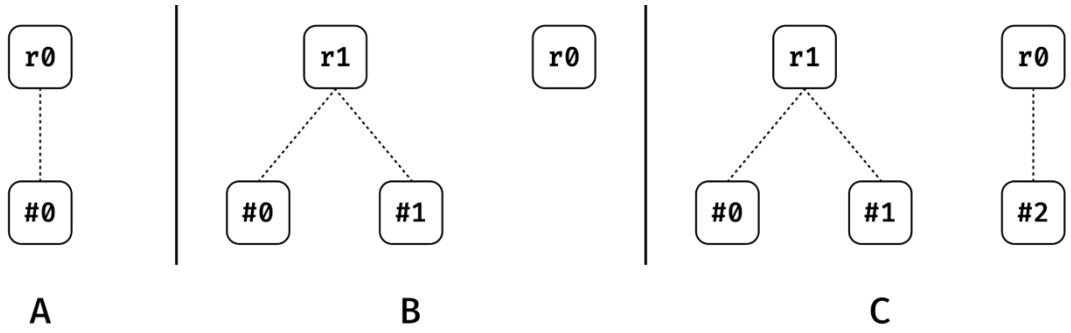

Abb. 3: Hinzufügen von Elementen zum Akkumulator; **A:** Initialisierung mit dem ersten Element; **B:** Hinzufügen eines zweiten Elementes, es wird ein Baum mit einer Kapazität von 2 erzeugt; **C:** Hinzufügen eines dritten Elementes an der nun freien Position 0

Beweise lassen sich analog zu „normalen“ Merkle Trees führen. Jedoch ist die Erstellung der Beweise vereinfacht, da die Hashes, die für den Beweis notwendig sind, in vorhergehenden Akkumulator-Werten auffindbar sind.

5. Ergebnis

Das beschriebene Schema erlaubt den effizienten Mitgliedschaftsnachweis für Blöcke innerhalb der Blockchain. Basis dafür ist eine angepasste Block-Datenstruktur (s. Abb. 4), die den zum Zeitpunkt ihrer Erstellung aktuellen Akkumulator-Wert enthält. Dieser kann zum Beweis der Zugehörigkeit jedes Vorgängerblockes zur Blockchain genutzt werden. Mithilfe dieser Anpassung kann die Existenz von Daten innerhalb einer unternehmensexternen Blockchain zweifelsfrei nachgewiesen werden. Der hierfür genutzte Beweis besteht zuerst aus einem Nachweis der Mitgliedschaft eines Blockes innerhalb der Blockchain, mit dessen Block-Hash ist der Beweis für die Existenz von Daten dann möglich.

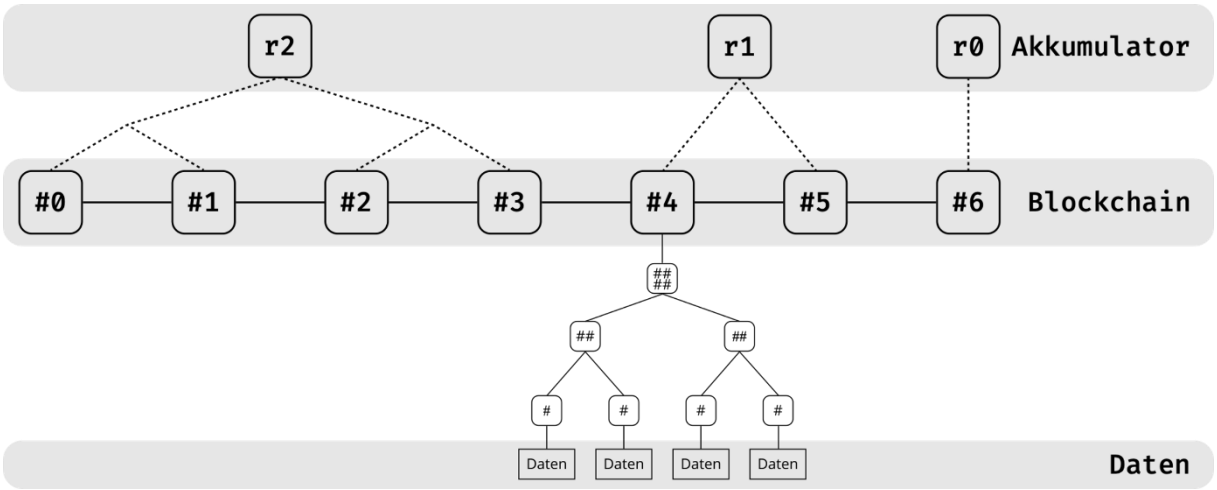

Abb. 4: Neue Struktur mit Akkumulator auf Block-Basis. Dargestellt ist die Beweisstruktur für Daten in Block #4, mit dem Akkumulator-Wert aus Block #6 (Beweis über *r1* und den Hash von Block #4)

Datenverfügbarkeit

Der Beitrag erklärt ein Schema; es müssen keine extra Daten zur Verfügung gestellt werden.

Interessenskonflikte

Es gibt keine Interessenkonflikte.

Förderung

Das Vorhaben wird mit Mitteln des Bundesministeriums für Bildung und Forschung im Rahmen der Bekanntmachung „Zivile Sicherheit – Kritische Strukturen und Prozesse in Produktion und Logistik“ unter den Förderkennzeichen 13N15150 bis 13N15153 gefördert.

Literaturverzeichnis

1. Bundeskriminalamt (2021): Cybercrime Bundestagsbild 2020
2. Bundesministerium für Bildung und Forschung (2019): Sicherheit und Nachverfolgbarkeit in zivilen Produktions- und Wertschöpfungsnetzwerken durch Blockchain (safe-UR-chain)
3. Merkle, Ralph (1980): Protocols for Public Key Cryptosystems, 1980 IEEE Symposium on Security and Privacy: 125–127
4. Nakamoto, Satoshi (2008): Bitcoin: A Peer-to-Peer Electronic Cash System
5. Neumann, Erik et al. (2021): A High-Performance Solution for Data Security and Traceability in Civil Production and Value Networks through Blockchain
6. Reyzin, Leonid/ Yakoubov, Sophia (2015): Efficient Asynchronous Accumulators for Distributed PKI