UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR SOFTWARE ENGINEERING
AND PROGRAMMING LANGUAGES

isp

Aktuelle Themen SSE
Martin Leucker, Martin Sachenbacher, Philipp Bende February 4, 2019

Übungsaufgabe

Eine Blockchain ist eine kontinuierliche Liste von Daten, die mittels kryptografischer Verfahren zu Datenblöcken zusammengefasst und miteinander verknüpft sind. Meist wird eine Blockchain als dezentrales verteiltes Konsenssystem genutzt, bei dem Teilnehmer weder einer zentralen Instanz, noch sich gegenseitig vertrauen müssen. Die Richtigkeit und Unveränderbarkeit bestehender Eintragungen werden dabei durch die Verknüpfung aufeinanderfolgender Blöcke garantiert. Anwendung finden Blockchains beispielsweise in verteilten Datenbanken, Kyptöwährungen wie Bitcoin, oder Smart Contracts.

In dieser Übung soll eine einfache Blockchainarchitektur mit folgenden Funktionen implementiert werden:

- Jeder Block ist in einer separaten UTF-8 codierten Plaintext-Datei gespeichert.
- Die genutzte Hash-Funktion ist SHA-256. Diese erhält beliebige Daten als input und errechnet daraus eine 64-Stellige Hexadezimal-Zahl.
- Eine beliebige Anzahl bereits bestehender Blöcke kann gelesen, und auf ihre Richtigkeit überprüft werden.
- Ein neuer Block kann einer leeren oder bestehenden Blockchain hinzugefügt werden.
- ein Block enthält die folgenden Daten:

```
block=Blocknummer
previous_hash=.....
data=beliebige Daten
```

- Die Blocknummer ist fortlaufend zweistellig (00, 01, 02, ...) und `previous_hash` bezieht sich auf den Hash des kompletten Vorgängerblocks.
- Der initiale Block hat Blocknummer 00 und `previous_hash=null`

Aufgabe 1 Implementation der Blockchain

1.) Implementiere die oben beschriebene Blockchain in einer beliebigen Sprache.

2.) Verifiziere die Implementation anhand von der gegebenen Blockchain im Archiv ex1 (Moodle Download).

3.) Füge dieser Blockchain 2 weitere Blöcke mit beliebigen Daten hinzu.

Aufgabe 2 Verifikation von Blöcken

Eine Blockchain ist kryptografisch richtig, wenn Blöcke den Hash-Wert des Vorgängerblocks enthalten. Eine Änderung der Daten eines bereits eingetragenen Blocks mit wenigstens einem Nachfolger verändert den Hash-Wert dieses Blocks. Somit stimmt der `previous_hash` im Nachfolger-Block nicht mehr mit dem Hash des veränderten Blocks überein.

Lade das Archiv ex2 aus dem Moodle herunter und überprüfe alle enthaltenen Blockchains auf ihre Richtigkeit.

1.) Falls die Blockchain fehlerfrei ist, füge einen Block mit den Daten "Blockchain richtig." hinzu.

2.) Welche Blockchains sind falsch, und wo liegen die Fehler?

3.) Was müsste ein Angreifer tun, um bestehende Daten einer Blockchain zu verändern, sodass die Veränderung nicht als Fehler erkennbar ist?

Aufgabe 3 Proof of Work

Der sogenannte "Proof of Work" ist ein System, was z. B. bei der Bitcoin Blockchain zum Einsatz kommt. Es ist ein rechnerisch sehr aufwändiger Prozess, der jeweils durchgeführt werden muss, wenn ein Block der Blockchain hinzugefügt werden soll. Für den Proof of Work in dieser Aufgabe kommt eine sogenannte `Nonce` zum Einsatz. Die Nonce ist dabei ein frei wählbarer Wert, der dem Block hinzugefügt wird.

Der Proof of Work ist erbracht, wenn eine Nonce gefunden wurde, sodass der Hash-Wert des gesamten Blocks einen bestimmten Wert nicht Überschreitet.

1.) Welche Vor/Nachteile hat der Proof of Work im Bezug auf die Sicherheit und Effizienz der Blockchain?

2.) Erweitere die Implementation der Blockchain um die Möglichkeit Blöcke mit Noncen im folgenden Format hinzuzufügen:

```
block=Blocknummer
previous_hash=.....
data=beliebige Daten
nonce=.....
```

3.) Füge der Blockchain im Archiv ex3 einen Block mit beliebigen Daten und Nonce hinzu, sodass der Hash-Wert des Blocks kleiner als 16^{58} ist (6 Nullen, gefolgt von 58 beliebigen Hex-Ziffern).

ABGABEKRITERIEN

Es muss ein Archiv im Format .zip, .7z, .tar oder .tar.gz im Moodle hochgeladen werden, welches folgende Dateien enthält

- ein PDF-Dokument, welches die Fragestellungen in Aufgabe 2.2 , 2.3 und 3.1 beantwortet.
- die Block-Dateien der Lösungen zu Aufgabe 1.3, 2.1 und 3.3
- den Quell-Code des Programms
- eine Readme-Datei, die erklärt wie das Programm (kompiliert), gestartet und bedient wird