

Modul Distributed Systems Advanced & Blockchain (DSA) Prüfung

Nachname: _____ Vorname: _____ HSR PID: _____

Nr.	Aufgabe	Max. Punkte	Erreichte Punkte
1	Blockchain Basics	17	
2	Distributed Systems	24	
3	UTXO	16	
4	Praktische Aufgaben	21	
5	Solidity/Ethereum	12	

Total 90

Hinweise allgemein:

- Zeit: 120 min.
- Zuerst diese Hinweise, dann die Prüfungsteile ganz durchlesen.
- Diese Prüfung ist **"Closed Book"**.
- Taschenrechner ist nicht erlaubt.
- Sonst sind keine Hilfsmittel zugelassen; auch kein Handy oder Smart Watch.
- Alle Antworten und Lösungen in diese Aufgabenblätter schreiben.
- Keinen Bleistift und keine rote Farbe verwenden.
- Die Antworten und Lösungen müssen nachvollziehbar sein. Bitte alles in leserlicher Schrift!
- Keine Fragen während der Prüfung. Bei Unklarheiten sind sinnvolle Annahmen zu treffen und diese klar zu dokumentieren.
- Die Bewertungspunkte der Aufgaben können nach der Korrektur noch etwas variieren.

Hinweise zum Prüfungsbeginn:

- Versehen Sie das Titelblatt mit Ihrem Namen.
- Den HSR-Badge sichtbar auf den Tisch legen.
- Allfällig bereitliegende Prüfung nicht umdrehen oder umblättern, bis die Aufsichtsperson explizit das OK dazu gegeben hat.

Hinweise zur Prüfungsabgabe:

- Kontrollieren Sie nochmals, ob die Prüfung mit Ihrem Namen versehen ist. Geben Sie alle Blätter – auch von nicht gelösten Aufgaben – geordnet und geheftet ab.
- Nach Prüfungsende: Drehen Sie Ihre Prüfung um und warten Sie, bis die Prüfung eingesammelt wird. Bleiben Sie dann sitzen bis die/eine Aufsichtsperson die Prüfungen gezählt hat und ein Zeichen zum Verlassen des Raums gegeben hat.

Aufgabe 1: Blockchain Basics (17 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

Bitte beachten Sie, dass keine Punkte für ein falsch gesetztes Kreuz abgezogen werden.

	Statement	True	False
1	Transaktionsdaten in Neo werden so verschlüsselt, dass nicht jeder die Transaktionen mitlesen kann.		
2	Transaktionsdaten in Ethereum werden nicht verschlüsselt, und jeder kann die Transaktionen mitlesen.		
3	Öffentliche Blockchains eignen sich gut um grosse Datenmengen zu speichern.		
4	Krypto-Mining kann mit jedem Rechner profitabel betrieben werden.		
5	Die Kostenfaktoren bei Krypto-Mining sind Anschaffungs- und Stromkosten.		

b) (6P) Aussage: Ethereum ist eine Blockchain welche «immutability» garantiert. Argumentieren Sie warum diese Aussage nicht stimmt.

d) (6P) Wie kann man sich gegen einen Fork bei einer Protokolländerung schützen? Beschreiben Sie einen möglichen Mechanismus.

Aufgabe 2: Distributed Systems (24 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

Bitte beachten Sie, dass keine Punkte für ein falsch gesetztes Kreuz abgezogen werden.

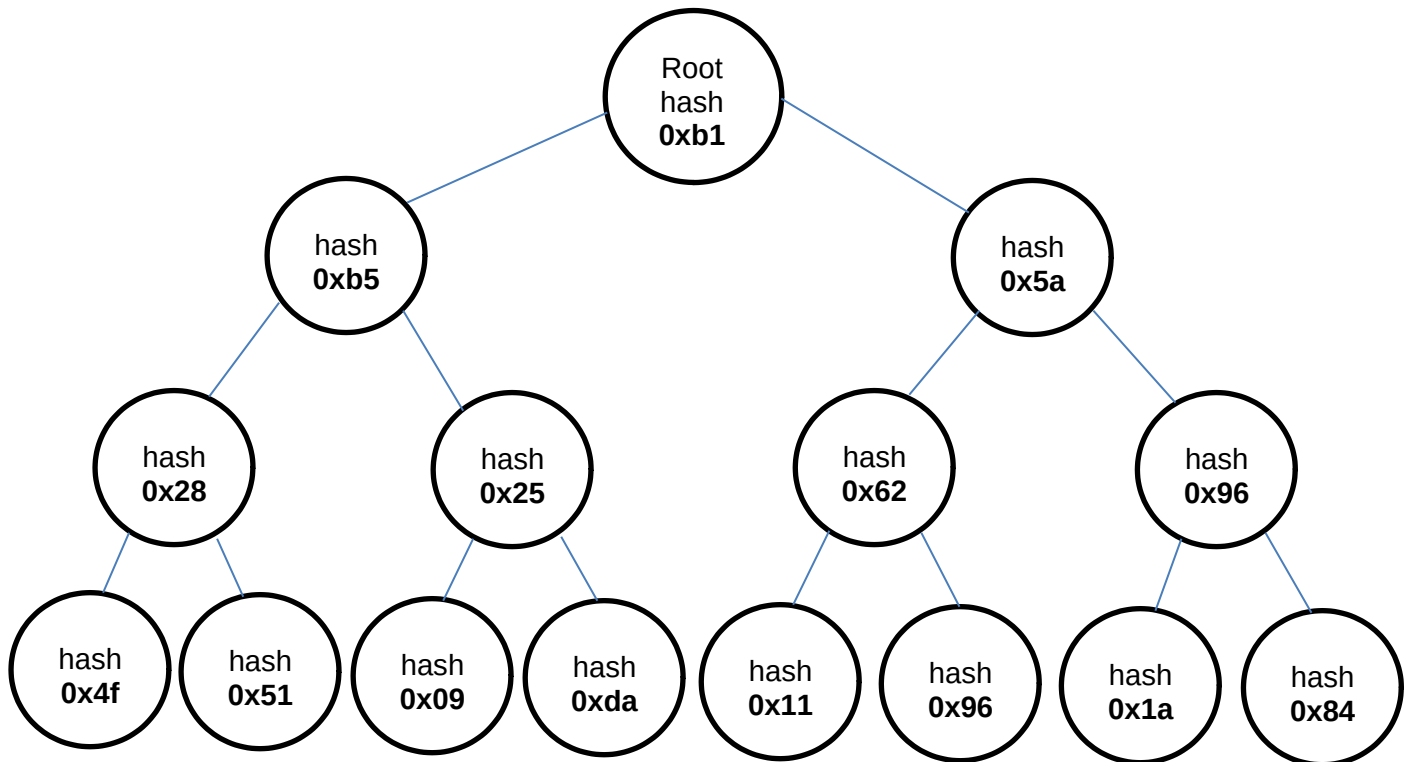
	Statement	True	False
1	In verteilten Systemen gibt es keine exakte globale Zeit.	☐	☐
2	Mit WebRTC können Browser untereinander kommunizieren.	☐	☐
3	In Paxos können 2 Phasen unterschieden werden: die «prepare/promise» und die «accept» Phase.	☐	☐
4	Bei Raft startet die Leader Election nachdem ein Leader nicht mehr erreichbar ist.	☐	☐
5	Bei einem byzantinische Fehler verhält sich ein System beliebig falsch.	☐	☐

b) (3P) Was ist das grundsätzliche Problem mit Replikas in einem verteilten System?

c) (3P) In P2P Systemen braucht es einen Bootstrap Peer/Node, um dem System beizutreten. Was braucht es bei WebRTC?

d) (4P) Was für Probleme sehen sie, wenn Sie eine eine Applikation mit Raft implementieren und es auf einem mobilen Gerät laufen lassen?

e) (5P) In dem folgenden Merkle tree befinden sich Hashes von 8 Dateien zum Zeitpunkt t in den leaf nodes. Sie besitzen bereits 7 Dateien und laden 1 Datei mit Hash **0xda** herunter. Was sind die minimalen Informationen um sicherzustellen, dass Sie alle Dateien mit dem richtigen Inhalt zum Zeitpunkt t haben.



Lösung:

g) (4P) Was sind Vor- und Nachteile von einer WebRTC Applikation, wie zum Beispiel WebTorrent gegenüber einer Applikation welche TCP/UDP direkt benutzt, wie zum Beispiel BitTorrent? Nennen Sie je 2 Vor- und 2 Nachteile.

Aufgabe 3: UTXO (16 Punkte)

Die folgende Tabelle zeigt Transaktionen in einem Block. Der Input einer Transaktion (TX) ist ein Pointer zu einer vorherigen Transaktion mit einem Index (z.B. der erste index wird mit idx:0 bezeichnet) zum Output. Es fallen keine Mining-Gebühren an. (Hinweis: es müssen nicht alle Inputs und Outputs verwendet werden).

TX	Input	Output	Betrag (Coins)
4	1, idx:0 / 1, idx:1	G, H	1, 1
5	2, idx:0	I	2
6	3, idx:0	J, E	3, 2
7	3, idx:1	K, P	4, 8
8	Coinbase	F	5

a) (4P) Alice besitzt die Schlüssel zu den folgenden Adressen F und E. Bob besitzt den Schlüssel zur Adresse M. Erstellen Sie eine Transaktion in welcher Alice 7 Coins an Bob schickt.

TX	Input	Output	Betrag (Coins)
9			

b) (4P) Transaktion 9 ist bestätigt. Bob besitzt den Schlüssel zur Adresse P. Konsolidieren sie Bobs UTXOs zum Output X (zu dem Bob ebenfalls den Schlüssel besitzt):

TX	Input	Output	Betrag (Coins)
10			

c) (4P) Transaktion 10 ist bestätigt und Charlie besitzt den Schlüssel zur Adresse N. Erstellen Sie eine Transaktion in der Bob 1 Coins an Charlie schickt:

TX	Input	Output	Betrag (Coins)
11			

d) (4P) Eine Transaktion von Dave beinhaltet 2 Outputs mit den Beträgen 1 und 0.2566221. Bei welchem von diesen Outputs handelt es sich um eine mögliche Bezahlung und bei welcher um einen Output an die Wallet von Dave. Argumentieren Sie warum Sie sich so entschieden haben.

Aufgabe 4: Praktische Aufgaben (21 Punkte)

Sie entwickeln eine verteilte Game Applikation mit Anbindung an eine Blockchain.

a) (4P) In Ihrer dezentralen Game Applikation müssen Sie sich zwischen einer Blockchain (z.B. Ethereum) und WebRTC für die Kommunikation entscheiden. Nennen Sie je 1 Vor- und Nachteil von Blockchain und WebRTC für die Kommunikation.

b) (4P) Sie wollen Ihren Smart Contract zuverlässig, schnell und automatisiert testen. Was müssen Sie beachten?

c) (4P) Sie verwenden für Ihre Applikation eine DHT. In einer DHT kann man entweder direkt oder indirekt replizieren. Welchen Replikationsmechanismus wählen Sie für den Online/Offline Status Ihrer Game Applikation und warum?

d) (9P) Für Ihre Applikation haben Sie die Wahl zwischen NEO, Ethereum oder Monero. Nennen Sie je 3 Eigenschaften pro Blockchain. Welche werden Sie in Ihrer Applikation verwenden und warum?

Aufgabe 5: Solidity/Ethereum (12 Punkte)

Dieser Smart Contract hält in der Variable ***balances*** fest, welche Adresse wie viele Ethers hat. Die Funktion ***transfer()*** überweist Ethers von diesem Contract auf eine Empfängeradresse.

```
pragma solidity ^0.4.25;

contract Exam {
    mapping (address => uint256) private balances;
    event Message(string msg);
    ...
    function transfer(address recipient, uint256 value) public {
        require(balances[msg.sender] - value >= 0);
        require(recipient.call.value(value)());
        balances[msg.sender] = balances[msg.sender] - value;
    }

    function payin() public {
        ...
        emit Message("done");
    }
}
```

a) (8P) Nennen Sie mindestens zwei Probleme in der Funktion ***transfer()***

b) (4P) In der Funktion ***payin()*** wird ein Event erzeugt. Wozu können diese Events verwendet werden? Können andere Smart Contracts direkt auf diese Events reagieren?