Modul Distributed Systems Advanced & Blockchain (ADSB) Prüfung

Nachname: _____ Vorname: _____ OST PID:_____

Nr.	Aufgabe	Max. Punkte	Erreichte Punkte
1	Blockchain Basics	9	
2	Distributed Systems	14	
3	Bloom Filter	10	
4	Praktische Aufgaben	8	
5	Solidity/Ethereum	4	
		Total 45	

Hinweise allgemein:

- Zeit: 60 min.
- Zuerst diese Hinweise, dann die Prüfungsteile ganz durchlesen.
- Diese Prüfung ist **"Closed Book".**
- Taschenrechner ist nicht erlaubt.
- Es 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 OST-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 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 (9 Punkte)

a) (3P) Kreuzen Sie Zutreffendes an.

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

	Statement	True	False
1	Transaktionsdaten in Bitcoin werden so verschlüsselt, dass nur der Sender und Empfänger sieht, was der Betrag ist.		
2	Transaktionsdaten in Ethereum werden so verschlüsselt, dass nur der Sender und Empfänger sieht, was der Betrag ist.		
3	Die grossen Kostenfaktoren bei Krypto-Mining sind Hardware-Anschaffungskosten und Stromkosten.		
4	Um Krypto-Mining profitabel zu betreiben braucht es zwingend ASICs.		
5	Daten in der Ethereum Blockchain kann man permanent löschen, sodass niemand mehr auf die Daten zugreifen kann.		
6	Die Datenspeicherung in öffentlichen Blockchains ist nicht teuer, deshalb können kostengünstig Daten gespeichert werden.		

b) (3P) Warum bereiten Zufallszahlen einer Blockchain Probleme?

d) (3P) Wozu dient ein Ethereum ERC-20 Smart Contract?

Aufgabe 2: Distributed Systems (14 Punkte)

a) (3P) 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	Mit WebRTC können vollständig verteilte Systeme erstellt werden und es braucht dazu keinen Server zur Koordination.		
4	WebRTC verwendet SCTP, welches im Internet gut unterstützt ist.		
5	Bei einem byzantinische Fehler verhält sich ein System beliebig falsch.		
6	Bei einem Sybil Attack werden für einen Angriff beliebig viele Nodes oder Identitäten erstellt.		

b) (2P) Was ist das grundsätzliche Problem mit Replikas in einem verteilen System?

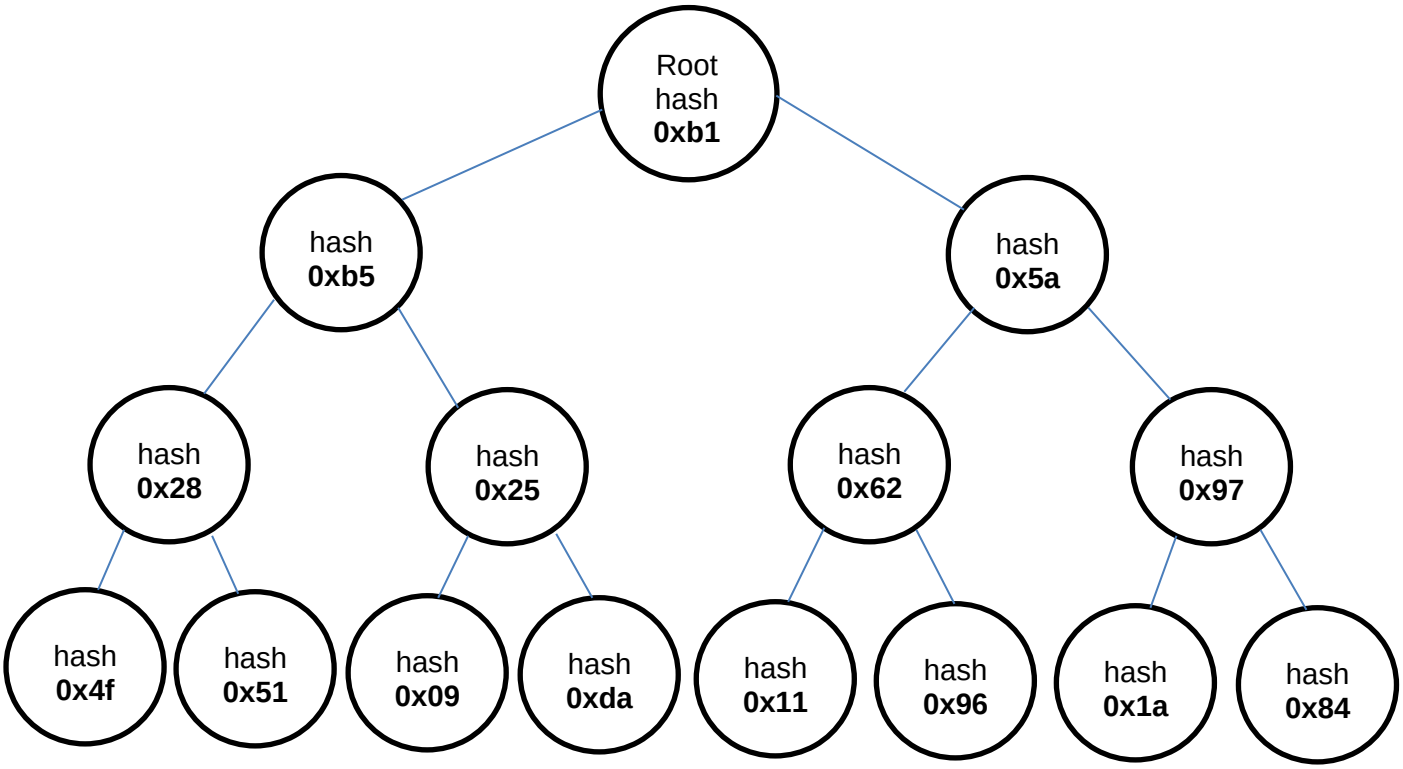
d) (2P) Was ist der Unterschied zwischen indirekter Replikation und direkter Replikation?

c) (2P) Beschreiben Sie wie man mit wenigen «put» Operationen eine DHT angreifen kann.

e) (3P) Gegeben ist der folgende Merkle-tree mit der Root-hash **0xb1** und 8 Transaktions-hashes. Sie erhalten die folgenden Merkle-proofs:

- Transaktions-hash 0x09, proof: 0x25, 0xb5, 0xb1
- Transaktions-hash 0x96, proof: 0x11, x97, 0xb5
- Transaktions-hash 0x4f, proof: 0x25, 0x5a, 0xb5

Sind das korrekte Merkle-proofs?



Lösung:

Transaktionshash 0x09, proof: 0x25, 0xb5, 0xb1 :
Transaktionshash 0x96, proof: 0x11, x97, 0xb5 :
Transaktionshash 0x4f, proof: 0x25, 0x5a, 0xb5 :

g) (2P) Erstellen Sie einen Merkle-proof für die Transaktion mit Hash 0x84.

Aufgabe 3: Bloom Filter (10 Punkte)

Die folgende Tabelle zeigt das Alphabet a-f mit den Werten 0-5. Ihr Bloom Filter hat eine Länge von 4 bit und Sie verwenden zwei Hashfunktionen, welche die ersten beiden Buchstabenwerte multipliziert, und die restlichen aufaddiert und umgekehrt. So hat z.B. der String «bca» mit der ersten Hashfunktion den Hash (1*2)+0=2, mit der zweiten Hashfunktion (1+2)*0=0.

a	b	c	d	e	f
0	1	2	3	4	5

a) (2P) Wie sieht Ihr Bloom Filter aus wenn Sie die folgenden Strings hinzugefügt haben: «cab», «ace», «bef»

Bit 0	Bit 1	Bit 2	Bit 3

b) (2P) Sie bekommen den folgenden Bloom Filter, sind die folgenden Strings enthalten: «caf», «bad»?

Bit 0	Bit 1	Bit 2	Bit 3
1	1	0	1

Antwort:

caf :

bad :

c) (2P) Da Sie auch Elemente löschen möchten, verwenden Sie einen Counting Bloom Filter. Fügen Sie die folgenden Strings zu ihrem Counting Bloom Filter hinzu: «bbe», «aed», «bba»

Counter 0	Counter 1	Counter 2	Counter 3

d) (4P) Bei einem Bloom Filter gibt es keine «false negatives» aber dafür «false positives». Beschreiben Sie wie es bei einem Counting Bloom Filter zu einem «false negative» kommen kann.

Aufgabe 4: Praktische Aufgaben (8 Punkte)

Sie entwickeln eine verteilte Finanz-Applikation mit Anbindung an eine Blockchain.

a) (2P) In Ihrer dezentralen Finanz-Applikation müssen Sie sich zwischen einer vollständig verteilter Architektur und einer skalierbaren Cloudarchitektur entscheiden. Nennen Sie je 1 Vor- und Nachteil von einer vollständig verteilter Architektur und einer Cloudarchitektur.

b) (3P) Sie haben sich für eine vollständig verteilte Architektur entschieden und möchten Coins zwischen zwei Blockchains dezentral tauschen per Cross-chain Atomic Swaps. Als Basis für diese Swaps dienen Hashed Time-Locked Contracts (HTLCs). Wie funktionieren HTLCs?

d) (3P) Für Ihre Applikation haben Sie die Wahl zwischen Bitcoin, Ethereum oder Tezos. Nennen Sie je 2 Eigenschaften pro Blockchain. Beantworten Sie zudem die Frage, welche Blockchain Sie in Ihrer Applikation verwenden und warum?

Aufgabe 5: Solidity/Ethereum (4 Punkte)

Dieser Smart Contract führt einen Batch Transfer durch. Der rot umrandete Teil weist einen schwerwiegenden Fehler auf.

```
255   function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
256     uint cnt = _receivers.length;
257     uint256 amount = uint256(cnt) * _value;
258     require(cnt > 0 && cnt <= 20);
259     require(_value > 0 && balances[msg.sender] >= amount);
260
261     balances[msg.sender] = balances[msg.sender].sub(amount);
262     for (uint i = 0; i < cnt; i++) {
263         balances[_receivers[i]] = balances[_receivers[i]].add(_value);
264         Transfer(msg.sender, _receivers[i], _value);
265     }
266     return true;
267   }
268 }
```

a) (4P) Was ist das für ein Fehler und wie wirkt sich dieser Fehler auf die Zeile 259 resp. 263 aus?

Fehler:

Auswirkung auf Zeile 259, resp. 263: