IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2019

MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
MSc in Computing Science (Specialist)
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C467

PRINCIPLES OF DISTRIBUTED LEDGERS

Wednesday 20th March 2019, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 The following questions address basic knowledge about decentralized blockchains (e.g. Bitcoin/Ethereum).

a At their core, a blockchain is a chain of blocks. There are different proposals on how to design such chains.

i) What is an *uncle block* in Ethereum? Explain what an uncle block is, why it has been introduced, and why Bitcoin does not have uncle blocks.

ii) What are the implications of uncle blocks on the security of a blockchain? What is different between the GHOST protocol as originally proposed and the version that is implemented in Ethereum?

iii) What is a Markov Decision Process, and why is it useful in the context of blockchain security?

b Different blockchains have different block generation intervals.

i) How long on average does it take to find a Bitcoin block and how long on average to find an Ethereum block?

ii) Please explain the benefits and drawbacks of having a faster or slower block generation interval. How would you motivate your choice of a block generation interval?

c We have seen multiple times in the lecture what an UTXO is.

i) Explain, with the aid of a diagram, what an UTXO is, and how it is used.

ii) How do UTXOs affect the scalability and the security of a blockchain?

The three parts carry, respectively, 40%, 25%, and 35% of the marks.

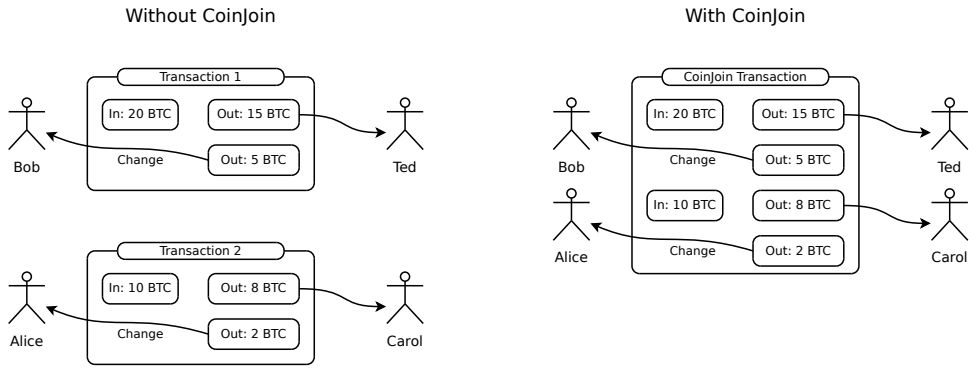

Fig. 1: CoinJoin Example where two transactions are merged into one.

2 a CoinJoin allows to merge different transactions (cf. Figure 1). The idea is to conduct transactions together with other persons. For peer discovery, a (typically centralized) service is used.

i) Based on your knowledge of Bitcoin, please explain the technical details of how a CoinJoin service could operate.

ii) Why does CoinJoin increase user privacy?

iii) What problems do you see with CoinJoin?

iv) How would you design a blockchain to protect the privacy of the users? Please provide ideas for full clients, as well as mobile/lightweight clients.

b Assume you want to implement the following protocol changes in Bitcoin. Read carefully and decide, whether the alterations can/must be implemented as a hard fork, soft fork or both.

i) Increase of the blocksize from 1MB to 8MB.

ii) Introduction of a chainID field to differentiate between two forked version of the same chain for replay protection, e.g., Ethereum and Ethereum Classic.

iii) Redefinition of the `OP_VERIFY` opcode to mark a transaction invalid if the top stack value is true.

The two parts carry, respectively, 85% and 15% of the marks.

3 Solidity is one of the most popular smart contract programming languages for the EVM (Ethereum Virtual Machine).

a Solidity offers different visibility settings such as *public* and *private*.

Given the contract below deployed on the public Ethereum network, who would be able to read "mySecret"? Elaborate your answer.

```
contract Example {

    address public owner;
    string private mySecret;

    constructor {
        owner = msg.sender;
    }

    function setSecret(string _secret) public {
        require(msg.sender == owner);
        mySecret = _secret;
    }

    function getSecret() public returns (string) {
        require(msg.sender == owner);
        return mySecret;
    }
}
```

b Smart contracts have suffered from quite severe security flaws and bugs. One of the most famous security flaws is the "The DAO" incident. There are major efforts to prevent future incidents.

i) What is the problem with the contract *Vulnerable* below?

```
contract Vulnerable {

    mapping(address => bool) authorized;
    mapping(address => uint) balances;

    function refund(uint amount) public {
        require(authorized[msg.sender]);
        require(amount <= balances[msg.sender]);

        msg.sender.call.value(amount)("");
        balances[msg.sender] -= amount;
    }
}
```

c i) What is a better alternative to the low-level *call* instructions? Motivate your answer.

ii) What is the result of the following line of code in Solidity? Why?

```
uint8(255) + uint8(1)
```

The three parts carry, respectively, 15%, 35%, and 50% of the marks.

4 a Existing permissionless blockchains, such as Proof-of-Work-based chains like Bitcoin and Ethereum currently only offer about 7–10 global transactions per second. Visa for example supports beyond 50 000 transactions per second.

i) What (simple) change(s) would significantly increase the throughput of PoW blockchains? Please give at least two options, and motivate your choice with advantages and disadvantages for each.

ii) Why can't Proof-of-Work blockchains scale significantly beyond 10–100 transactions per second?

b Payment channels have been introduced to scale blockchains. Lightning is one such example for Bitcoin.

i) Explain how payment channels are created, and how they work between two parties A and B.

ii) Why do payment channels increase the transaction throughput of blockchains?

iii) On what technical feature(s) are payment channels based?

iv) Please describe two different payment channel designs.

v) What security assumptions do payment channels rely on?

vi) Name at least four different drawbacks of payment channels.

c Besides payment channels, commit-chains have been introduced to scale blockchains on the second layer. Please explain what commit-chains are, and how they are fundamentally different to payment channels.

The three parts carry, respectively, 20%, 70%, and 10% of the marks.