IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

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

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

PAPER COMP70017=COMP97045=COMP97046

PRINCIPLES OF DISTRIBUTED LEDGERS

Tuesday 16 March 2021, 10:00
Duration: 120 minutes
Includes 0 minutes for access and submission

Answer ALL TWO questions
Open book assessment

This time-limited remote assessment has been designed to be open book. You may use resources which have been identified by the examiner to complete the assessment and are included in the instructions for the examination. You must not use any additional resources when completing this assessment.

The use of the work of another student, past or present, constitutes plagiarism. Giving your work to another student to use constitutes an offence. Collusion is a form of plagiarism and will be treated in a similar manner. This is an individual assessment and thus should be completed solely by you. The College will investigate all instances where an examination or assessment offence is reported or suspected, using plagiarism software, vivas and other tools, and apply appropriate penalties to students. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

Paper contains 2 questions

1 **Blockchain Privacy**

a Privacy is increasingly gaining popularity in our digital society.

i) Define blockchain privacy in terms of the types of information that might be kept confidential.

ii) If you read the network layer on Bitcoin/Ethereum by connecting to peers, what would you be able to capture?

iii) People often claim that blockchains are being used to pay for illicit services, or fraud. Why would it be a bad idea to use the blockchain for illegal activities?

iv) By observing only the network layer, how could you identify the true originator of a transaction?

b Clustering blockchain addresses is the process of associating multiple addresses to a real-world entity. Because real-world entities can create as many addresses as they please, such clusters can become large.

i) What heuristics could you use to cluster blockchain addresses in Bitcoin? What is the fundamental problem of heuristics?

ii) Could you still use the heuristics proposed in the previous question to cluster blockchain addresses on Ethereum? Please justify your answer.

c Blockchain privacy by design and add-on mixer.

i) Please list at least two blockchains with privacy by design. What different cryptographic primitives or protocols are used in these blockchains, compared to Bitcoin?

An add-on mixer is a decentralized application which allows different users to deposit their coins, and later withdraw them to a new address to break the link between the withdraw and the deposit.

ii) Why are add-on mixer technically weaker from a privacy perspective than blockchains with privacy by design?

iii) Tornado.cash is an add-mixer on Ethereum. In Figure 1, we show an interesting situation of money flow in Tornado.cash: Multiple depositors receive ETH from a single user account (EOA) before they deposit in Tornado.cash. Based on this plot, what heuristic can you propose to cluster Ethereum addresses?

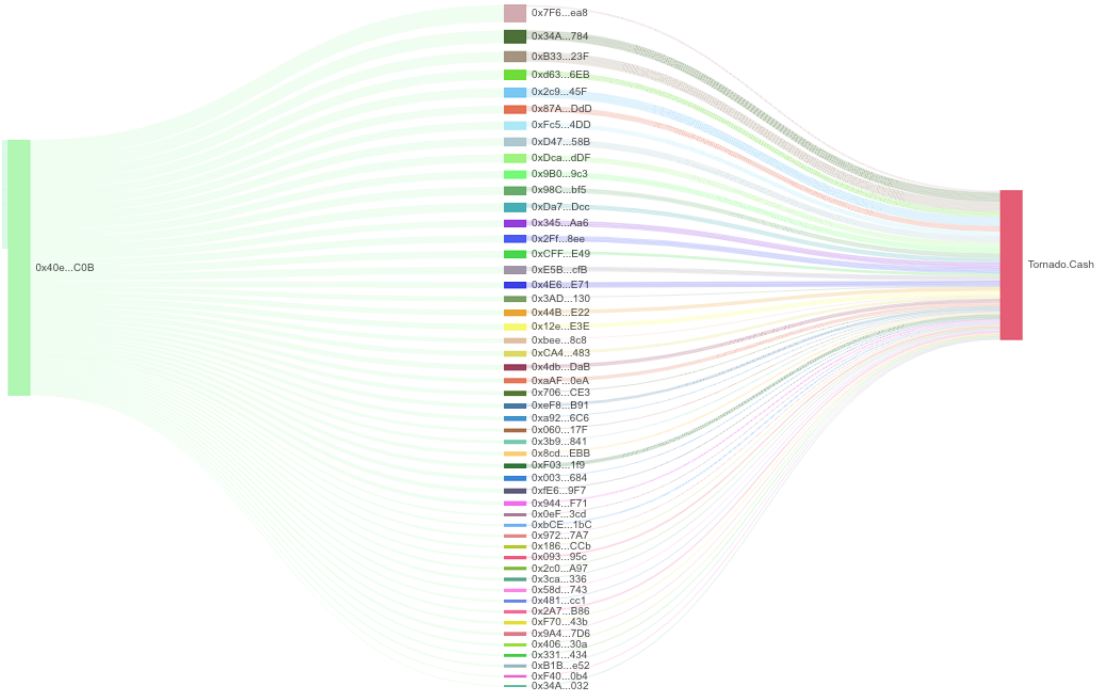

Fig. 1: Sample transaction flow into Tornado.cash

iv) Do you think current privacy-enhancing techniques in blockchain (privacy by design or add-on mixer) can sufficiently protect users' privacy? Please provide and motivate your opinion.

The three parts carry, respectively, 40%, 20%, and 40% of the marks.

2 **Smart contracts**

Smart contracts can be 40-lines of code and hold millions of dollars of assets. It is hard to write secure smart contracts which is evident as large-scale smart contract hacks are reported on a weekly basis.

a i) How would you describe a smart contract to a layman?

ii) A Solidity contract can have exactly one unnamed function called the fallback function. What is the usage of this function?

iii) Identify and explain three types of Ethereum transactions for interacting with smart contracts.

iv) What is a front-running attack for smart contracts and how can an average user do it without collaborating with a miner? Please explain with an example.

v) Name and explain three permission systems that can be built into smart contracts to control who can invoke a function?

b Re-entrancy.

i) Figure 2 presents a victim contract which is vulnerable to a re-entrancy attack and the attacker contract that can deploy the attack. Explain why the victim contract is vulnerable, how the attacker contract can withdraw more than 1 ether from the victim contract and how the contract can be changed to prevent the attack.

In your answer, please do not write code, but explain step-by-step how the re-entrancy attack is performed.

The two parts carry equal marks.

```
pragma solidity 0.4.24;

contract VictimContractInterface {
  function withdraw() public payable;
}

contract VictimContract {
  uint256 toTransfer = 1 ether;

  /* Only 1 ether can be sent by this contract */
  function withdraw() public payable {
    msg.sender.call.value(toTransfer)("");
    toTransfer = 0;
  }
  function deposit() public payable {}
}

contract AttackerContract {
  address victim;

  constructor(address _victim) public {
    victim = _victim;
  }
  function attack() public payable {
    VictimContractInterface(victim).withdraw();
  }
  function () external payable {
    attack();
  }
}
```

Fig. 2: Re-entrancy attack