In=16 Feb 8:45am; Out=06 Feb; Total Marks = 150

*You may partner with up to four others (**a group of max 5**) to submit a single write up in a single group. We encourage discussion with other students in class, even if they are not in your group. Each student must build a good understanding of all the questions even if they discuss and collaborate with others. Merely dividing the problems among group members and putting those together before submission will severely impact your learning, and we **strongly advice against it**. Furthermore, **blatant copying** from online or other resources is **forbidden**. If there are confusions or questions, post those on Piazza or see your TA or Instructor.*

Submission Instructions: *You must submit a PDF on LMS in the appropriate tab. This includes hand-written content (which you may scan and upload). There will be NO late days or extentions. Please write down the names and roll numbers on a front page. Make only one submission per group.*

Question 1 [2+4 marks]

a) Describe the workings of a 51% attack and list the conditions necessary for launching a 51% attack?

Solution: 51% attack is enabled when a miner gets access to 51% or more computing power in a blockchain network. Truthful miners would add their blocks to the blockchain using the longest chain protocol at their rate. On the other hand, a malicious miner would add blocks to his private chain faster owing to the 51% computing power, making it grow more quickly. The malicious miner then would broadcast his private chain once it gets longer. The old public chain is abandoned, including the miner's spending record. With this method, a miner can "double-spend" a coin he has already spent on the other (abandoned) chain.

b) The Tesla Company has agreed to sell Electric Vehicles in Exchange for Ethereum. Can a 51% attack be used to purchase this Electric Vehicle for free? If yes, describe how you would do this. If no, explain why this is not possible.

Solution: Yes, this attack is possible. Ethereum uses a Proof-of-Stake consensus mechanism. As such, in order to launch this attack, we will need 51% of the total stake in the network (instead of the computational power). If we have 51% of the total stake, we may launch the attack as follows:

- Buy an EV. This involves creating a publishing a transaction which transfers (say) 100 Eth from our account to Tesla's. Let's assume this is present in block # 1450.
- Once we have the EV, we create a block whose parent hash is block% 1449. We start to append to this new block instead.

- Eventually, fork will become the longest chain and, in this view, we never sent any Ethereum to Tesla's account.

Question 2 [3+3 marks]

a) Alice comes up with her own digital signature scheme. She writes a message and encrypts that message using her Private key. She then sends this message and the encrypted message both to Bob. Can Bob be sure that this message came from Alice?

Solution: No. As mentioned in class, Kevin may use Alice's *public* key to generate garbage data and send it to Bob.

b) Alice modifies this scheme. She writes a message and encrypts that message with her private key (as in the previous part). She then hashes the encrypted message and sends this hash along with the original message to Bob. Can Bob be sure that this message is from Alice?

Solution: No. Bob has no way of verifying whether the message was, indeed, originally from Alice or otherwise. This is because the hash function is only one way – and Bob can never know the encrypted intermediate message (since he doesn't have Alice's private key).

Question 3 [4+4 marks] In class we learnt that a symmetric key is often used for communication *in-bulk* i.e. when there is a lot of data that must be transferred. This is done because symmetric keys operate a lot faster than assymetric keys.

a) Assume that Alice and Bob share a secret, symmetric key. Devise a scheme with which Alice can send a plain-text message to Bob and ensure that Bob can verify the authenticity of this message.
Assume that Alice and Bob must share a LOT of data and asymmetric keys are not viable.

Solution: The scheme we are trying invent is very similar to that of Message Authentication codes. Since Bob and Alice share a single symmetric key and privacy is not a concern, only authenticity, we just need to make sure our message has not been tampered with.

Alice generates a message and then hashes it herself. Then she "signs" it by applying the symmetric key to the hash to create a signature. She sends the message and signature to Bob.

Bob can verify it is from Alice by hashing the message and applying the symmetric key to the signature again. If the result of these two operations match, the message is from Alice.

b) Consider a scenario in which Alice wishes to relay to Bob the message "Give me 3.50$". A malicious actor, Kevin, duplicates this message and sends it twice to Bob. According to the scheme you devised in the previous part, what will the outcome be? Further, devise a scheme in which duplicated messages are rejected.

Solution: Kevin can duplicate Alice's messages and Bob has no way of knowing.

There is an easy fix to this, add a nonce or a timestamp to the messages. If a message shows up with the same nonce or timestamp, Bob should reject it.

Question 4 [5 marks] There are multiple strategies to search for the 'nonce' value in Bitcoin mining. Suggest two strategies and explain which of the two you think is more strategic.

Solution: All strategies are equally valid. Since each new nonce value is equally likely to be a valid nonce, any strategy like increasing the nonce by one or by a factor of two is fine.

Question 5 [5 marks] Why is the nonce field included in the block header of Bitcoin when the number of possible rearrangements of transactions in a block (there are around 2500 transactions in a block) alone provides a monumentally huge space of $2500! \approx 10^{7411}$ while a nonce can have only 2^{32} values?

Solution: A bitcoin block may have very few transactions. (For example, a lot of the early blocks had only one transaction). In situations like this, it may be impossible to rearrange the transactions and get a valid hash for the block.

You would have to create a new public key (to lock the output to) just to go through the 2^{32} *nonce values again.*

Question 6 [3+2+2+2+3 marks] Consider a hash function h(x) that is publicly known and defined as:

$$h(x) = (7x^3 + 5x^2 + 11x + 13) \mod 19$$

Alice has a private key method

$$Pri(m) = (11m^3 + 7m^2 - 5m + 3) \mod 19$$

and the corresponding public key method

$$Pub(c) = (-11c^3 - 7c^2 + 5c - 3) \mod 19$$

.

(a) Given the message (47, 2), is it from Alice?
(b) What are the possible values the signature can take in this hash function?
(c) What is the minimum value that the signature can take in this hash function?
(d) Can you find two different messages that produce the same signature using the hash function? If so, provide an example.
(e) How secure is this hash function? Can you find a collision or preimage attack on this hash function?

Solution: (a) To determine if the message $(47, 2)$ is from Alice, you need to apply the public key method $Pub(c) = (-11c^3 - 7c^2 + 5c - 3)mod19$ to the signature 2 and compare the result with the message 47. If they match, the message is likely to be from Alice.
(b) The signature c in this hash function can take any value between 0 and 18 (inclusive) as the modulus used is 19. The signature c can be calculated using the private key method Pri(m) as:

$$c = Pri(m) = (11m^3 + 7m^2 - 5m + 3)mod19$$

$$c = Pri(47) = (11 * 47^3 + 7 * 47^2 - 5 * 47 + 3) mod 19$$

. Therefore, the possible values for the signature c in this hash function are $0, 1, 2, ..., 18$.
(c) The minimum value that the signature can take in this hash function is 0.
(d) Yes, it is possible to find two different messages that produce the same signature using this hash function. For example, (2, 14) and (16, 14) both produce the signature 14.
(e) The security of this hash function depends on the size of the modulus and the complexity of the calculation. However, it is generally considered a weak hash function and can be easily broken by attackers with sufficient computational resources. A collision attack involves finding two different messages that produce the same signature, and a preimage attack involves finding a message that produces a specific signature. Both types of attacks are possible on this hash function.

Question 7 [4 marks] What is the distinction between symmetric and asymmetric encryption that we studied in class?

Solution: Alice and Bob communicate via a single shared key. Each pair has its own key. There would be n(n-1)/2 symmetric keys in a network of n communicating nodes. You use the same keys to encrypt and decrypt. Asymmetry: Alice and Bob have opposing keys. There would be 2n asymmetric keys in a network of n communicating nodes. Two distinct keys are used for encryption and decryption.

Question 8 [4 marks] Why is it needed to create a fork? Who creates the fork? On what basis do they decide if a soft fork is created or a hard one?

Solution: Forks are created to change the rules of block creation. This can be useful for introducing new functionalities or can be a part of the natural functionality of the blockchain. Forks are created by various entities. In Bitcoin's case, for example, the soft fork (the halving of the bitcoin mining reward every 4 years) is written into the code of Bitcoin itself. In Ethereum's case, the Ethereum foundation may decide on forks. A soft fork is created when they want to upgrade functionality without creating a split. A hard fork is a calculated decision to introduce a split in the blockchain and create a new cryptocurrency. Ethereum vs Ethereum classic is a hard fork which was created as a result of a security breach.

Question 9 [5 marks] When a soft fork is invoked, why does it need majority of nodes to upgrade to ensure that a single chain emerges?

Solution: Initially a soft fork has two chains, one which is following the old rules but violating the new and one which is following both. If a majority is upgraded, they will append to the chain which is following both the upgraded rules. As a consequence, the older chain would grow stale because the newer one would grow in length and a majority of transactions would be present in the newer chain.

Question 10 [5 marks] In a generic blockchain, explain the five step process that allows a user transaction to become part of the ledger? Indicate the corresponding stages in data life cycle.

Solution: First, the user submits a transaction. The transaction is broadcasted by the user. It ends up in the log/mempool of other auditor nodes. After the transaction is verified, it ends up in a record along with other transactions and is broadcasted as a part of the record. Finally, it ends up in the private view (journal) of the blockchain of other nodes. If it ends up in the longest chain (ledger) then it is a confirmed, valid transaction.

Question 11 [5 marks] In Bitcoin, each user's balance is represented using the hash of a public key. A user's name or identity is not stored on the blockchain. An adversary may still use the transaction history associated with a public key in order to arrive at the identity of a user. Give an example of how this is possible and devise two strategies which may be used to protect the anonimity of a user.

Solution: A user's identity may be identified with correlating the activities of a user on the blockchain with people in real life. For example, I buy a Tesla for exactly 5.24234232 BTC. A third party observes that own a Tesla now and may look up the price of a Tesla (in BTC). Then they can check for recent transactions in which exactly 5.24234232 BTC were traded and guess which my identity on the blockchain. This is not a perfect approach but it can work with a high degree of success in most circumstances.

Many strategies may be used for this purpose. The most common (and recommended) strategy is to lock each UTXO to a *new* public/private key pair. (In fact, this is the most recommended approach if you are trying to future proof the BTC blockchain – the answer to question 12 will be relevant).

Another strategy is to involve multi-input and multi-output transactions. If multiple people are sending things individually, it is easier to track who is sending who. If a bunch of exchanges happen as a part of one transaction, this is a lot more difficult.

Further obfuscation is also possible, such as sending the same amount to public keys belonging to the same individual and so on.

Question 12 [5 marks] Read up on "Shor's algorithm". Describe how the existence of this algorithm (and similar algorithms) is harmful for the long-term security of Bitcoin-like blockchain.

Solution:

The Bitcoin blockchain relies on Elipctic Curve Cryptography (ECC)for its Public Key Infrastructure. The security provided by ECC is only possible because of the discrete logarithm problem. This makes it infeasible for someone to compute the private key from a public key.

Algorithms similar to Shor's algorithm, however, make it possible to derive the private key from the public key. If such a thing is possible, one could all the bitcoin locked to your public key. Luckily, we only store the hash of a public key and not the public key itself (except when unlocking an output). As such, Shor's algorithm (and similar algorithms) will not affect the security of the bitcoin blockchain substantially.

However, there are old transactions whose public key is, well, public. Satoshi's original transactions were locked to the public key (not its hash) and a party, utilizing these new algorithms, could steal Satoshi's Bitcoin.

Question 13 [5+5+5+5 marks] Consider as simple encryption procedure using a public/private key pair. You have the public key as the integer p=7 and the "secret" private key as the integer s=3.
The method to encrypt an integer m is: $c = m^p \mod 33$
And the method to decrypt an encrypted integer is: $m = c^s \mod 33$
Assume that the input characters have the following numerical values:
$a/A = 0, b/B = 1, c/C = 2, ..., z/Z = 25$.

a) Encrypt the word 'SurpriseQuiz' and write down the sequence of numbers.

Solution: If we convert it to numerical values, we get:
18 20 17 15 17 8 18 4 16 20 8 25 13 4 23 19 22 4 4 10.
If we apply the public key to this, we get:
6 26 8 27 8 2 6 16 25 26 2 31 22 16 16 10.

b) Decrypt the sequence "1 2 13 29 20 2 7 20 21 16 8 16 13 28". Write down the characters and what you think of them.

Solution:
This translates to "BitcoinOverEth"
Now lets consider a hashing function $h(x) = \lfloor e^x + 100 \rfloor \mod 32$ where $x \in [0, 50]$.
As $h(x)$ is defined only for one input, we will modify it to take in lists of numbers such as "31 23 4 2 13". Now, $h(31\ 23\ 4\ 2\ 13)$ will be $h((31 + 23 + 4 + 2 + 13) \mod 50)$. The idea is that every number in the list makes a contribution. Suppose Alice has the public key $p = 7$ and private key $s = 3$. Data is sent in the form "$m^1 m^2 m^3 ... m^n$ sig" where the m^i are numbers from the message and "sig" is the private key encryption of the hash of the message.

c) Suppose Alice wants to transmit the word "SaveMe". What is the final message Alice transmits?

Solution: The message is "18 0 21 4 12 4". We compute the hash as h((18+0+21+4+12+4) mod50)=h(59 mod 50) = h(9).
Then $h(9) = \lfloor e^9 + 100 \rfloor \mod 32$
this give h(9) = 11 Then the private key on 11 is $m = 11^3 \mod 33 == 11$
. So the final message is "18 0 21 4 12 4 11".

d) You receive a message "12 20 18 19 0 5 0 17 14 2 10 18 1 0 2 7 4 18 7 14 2 10 18 5".

Is this message from Alice? Why or why not? If it is a message from Alice then write the message.

Solution:
The hash of the message is: 12+20+18+19+0+5+0+17+14+2+10+18+1+0+2+7+4+18+7+14+2+10+
= 218 h(218 mod 50) = h(18) = 5. Now the public key on the signature is also 5 so this message is from Alice. message: “MustafaRocksBacheShocks”

Question 14 [2+2+2+2 marks] Say Rose is using the public and private keys to sign and encrypt a message to Jack. She considers two options:

1) Sign the message and then encrypt the result
2) Encrypt the message and then sign the result

a) Which is the preferable strategy (considering she doesn’t want to reveal her identity) and why?

Solution: Strategy 1 would be the preferable strategy. The result is encrypted so available only to Jack while she also authenticates herself.

b) Continuing from above part, which key will she use for each task (i.e. signing and encrypting the message)?

Solution: The encryption will be with Jack’s Public key and the signing with her private key.

c) Continuing from part **a**, if Caledon steals Rose’s private key. Will he able to decipher that this message was sent by Rose? Explain your answer.

Solution: No, the message is still encrypted with Jack’s public key. He still needs Jack’s private key to decipher the message.

d) Rose now wants to make a public announcement that could be authenticated by anyone on the internet. Which strategy should she use to accomplish this task. It can be from the above two strategies or one of your own. Explain your answer.

Solution: It is neither. She has a plaintext message which she signs and broadcasts. Anyone can use her public key to verify that the message was, indeed, from Rose.

Question 15 [2+4+2 marks] Alice wants to send a message to Sara and she wants to ensure that Sara can verify that this message was sent by her. So Alice takes a hash of the message and encrypts it with her private key to generate a digital signature (signature = PrivKey(hash(m))). She then sends the message along with the signature to Sara. Alice uses the following function to compute the hash:

$hash(m) = (m + 4) \mod 24$

Alice wants to send a message (integer m) to Sara. So, she computes the hash of the message and encrypts it with her private key and sends the message and signature to Sara.

a) Upon receiving this message, how will Sara verify that the message was sent by Alice and that the message has not been tampered with. Explain your answer.

Solution: She computes the hash of the message herself. Then she applies Alice's Public Key to the signature. If the result of these two operations match, the message has not been tampered with – assuming the hash function and private-public keys are cryptographically secure.

b) Jack is eavesdropping on the communication between Alice and Sara. Is it possible for him to modify the message in such a way that Sara cannot detect the tampering in the message? Explain your answer with an example.

Solution: Yes, this is because the hash function is *horrible*; it has a really small output space and a pre-image or collision can be easily found.

Jack can add or subtract any multiple of 24 to m and the hash would still be the same. As such, the digital signature would still match.

c) Ideally we would want that the signature is invalidated when the message contents change. How can we make our technique more robust to detect even the slightest of changes in the original message? Explain your answer

Solution: We use a cryptographically secure hash function. This requires it having a huge output space and collision and preimage resistance (amongst other things).

Question 16 [3+3+3 marks] Alice devises a hash function.

a) The only *two* outputs for this hash function are 0 and 1. The output of this hash function is random looking and has the avalanche effect. Can this function have pre-image and collision resistance? Is this function cryptographically secure?

Solution: Even if this function is random looking and has the avalanche effect – it lacks pre-image and collision resistance. This is because it would be trivial to find a pre-image for, say, y given $h(y) = 0$. We would have to try only two values (on average) before finding such a y. We would try two more values to get another y' which means that finding collisions is also trivial. This function is, of course, NOT cryptographically secure.

b) Alice modifies her hash function. The output of this function is still random looking and has the avalanche effect. However, it has two bits in the output, instead of one. Can this function have pre-image and collision resistance? Is this function cryptographically secure?

Solution: Even if this function is random looking and has the avalanche effect – it lacks pre-image and collision resistance. This is because it would be trivial to find a pre-image for, say, y given $h(y) = 00$. We would have to try only four values (on average) before finding such a y. We would try four more values to get another y' which means that finding collisions is also trivial.

c) Alice continues to modify her hash function while the function's output is still random looking and has the avalanche effect. She increases the number of bits in the output. After how many bits will you consider this hash function cryptographically secure?

Solution: Adding a bit decreases the probability of finding a pre-image by a factor of 2 and the collision probability by a factor of 2 as well. It is ultimately up to the user to decide how many bits should be in the output. MD5 has 128 bits, SHA-1 has 160 bits, SHA256 has 256 bits and SHA3 has 512 bits. The more bits the better. MD5 is broken and SHA1 collisions have already been found. It would generally best to keep at least 256 bits in your output.

Question 17 [5 marks] Read up on "Grover's algorithm". Will the existence of this algorithm (and similar algorithms) effect cryptographic hash functions such as SHA256? What about symmetric key encryption such as AES-128?

Solution: Grover's algorithm makes it easier to find a collision or pre-image for a hash function. However, the complexity goes from 2^{256} to 2^{128} for SHA256 which is infeasible for quantum computers in the forseeable future. As such, hash functions are secure against such algorithms.

The same applies for symmetric key encryption (which does not rely on prime numbers) such as AES-128. AES-128 would also go down to 2^{64} tries before being able to guess the key. This is out of the reach of quantum computers in the forseeable future.

Question 18 [20 marks] There are 10 people present in a room. Each of these people have 10 stones each. These people wish to trade these stones. HOWEVER, keeping actual, physical stones and trading them is cumbersome. Devise (in great detail) a scheme in which people can keep track of the number of stones each person has without the need of physical stones. Here are some questions you must ponder:

a. How will person A transfer an imaginary stone to person B?

b. How will person A determine how many stones person C has?

c. What are the capabilities necessary for each person in you scheme? For example, they should be able to make an announcement in the room or remember details about each person etc.

d. Can your scheme fail if a fraction of the users group together maliciously? If not, why not? If so, what is the size of this fraction and how will they destroy your scheme?

Solution: The simplest way is for person A to yell out that they are transferring something to person B. If a majority of the rest of the 8 people, who are listening, agree that this is possible then person B gets one more imaginary stone.

If person A wants to know how many stones person C has, they just yell out "How many stones does person C have?" and take the most common reply as the answer.

This requires a few things. First that each person can yell out to all the 10 people in the room (atomic broadcasting). Furthermore, a person listening can recognize a yeller by their voice (authentication). Lastly, each person has a perfect memory – they remember who has how many stones.

Our scheme can fail if more than 4 people band together. Let's suppose I ask for the amount of stones C has. The four people can say any value and it might become the most common reply. The fraction would be $\frac{T-2}{2}$ where T is the total number of people. We subtract 2 because ourselves and C reply should not be used to estimate the number of stones C has.

Question 19 [3+2 marks]

a) What is an "oracle" in the context of blockchain?

Solution: An oracle is an entity that allows smart contracts to interact with the outside world. This may involve acquiring information from outside or affecting something outside.

b) Recall the example of the Indian fertilizer in class. Which entity would be an oracle in that example?

Solution: The oracle would be the service that allows the smart contract to ping the weather website. This can be another external, with a public key that is hard coded in a smart contract.

Question 20 [5 marks] ChatGPT claims that digital signatures can be used for the purposes of non-repudiation. What is non-repudiation and is it necessary for a blockchain such as Bitcoin to function properly?

Solution: Non-repudiation is a property that prevents someone from claiming a lack of knowledge or awareness.

Let's suppose you Alice and Bob share an encrypted Key. Alice sends a message to Bob saying "give me 10$ in cash" and "signs" it using the encrypted key. Now, Bob gives her the 10$. Alice then claims she never sent this message and that Bob made it up (since he has the same key). Alice can deny having knowledge of the message in the first place.

Digital signatures provide non-repudiation. There is no way anyone other than Alice can sign any message in her name. Non-repudiation is necessary for Bitcoin, transactions are irreversible and it should be impossible for someone to claim they did not send it in the first place.