

Name - KUMAR NIKHIL

Roll no.: 1806082

Program Code : UG1-CSE

Branch: CSE

Course Title : Blockchain Technology

Course Code: CS6475

Exam Date: 19th May 2021

## ANSWERS

### Question-1

$H \rightarrow$  Hash function ( hiding + puzzle friendly)

#### 'Puzzle Friendliness'

A hash function is said to be puzzle friendly if for every possible  $n$ -bit output value  $y$ , if  $z$  is chosen from a distribution with high entropy.

Given:  $x$  and a highly-unlikely-and-randomly-chosen  $y$

It is difficult to find  $z$  such that  $h(x|z) = y$  (but it should exist).

Now in the given case:

$$G(z) = H(z) \parallel z_{last}$$

But it's already given  $H(z)$  is puzzle friendly, so for a given  $x$  and  $y$  it is difficult to find  $z$  in  $H(x|z) = y$ , so if we can't find  $z$  from  $H$  then it is difficult to find  $z_{last}$  as well and hence from  $G(z)$  we can't find  $z$  easily for the given  $x$  and  $y$  and it is also puzzle friendly.

Name - KUMAR NIKHIL  
Program Code - VG-CS

Roll no - 1806032

Course Title - Blockchain Technology  
Exam date - 19th May, 2021

Branch - CS1  
Course Code - CS6475

Hiding: A hash function is for hiding if secret value  $x$  is chosen from a probability distribution that has high entropy, then given  $H(x||z)$  it is infeasible to find  $x$ .

Given:  $H(x||z)$

Secret:  $z$  is a highly-unlikely and randomly chosen  $z$ .

Hard to find  $y$  such that  $H(y) = H(x||z)$

The hiding property should work for all plaintext spaces, even if your ~~one~~ plaintext space is {0, 1}.

$\Theta(z) = H(z)$  is not hiding for that plaintext space. that means if  $z=0 \text{ or } 1$  then for

$\Theta(z) = H(z)$   $H(z)$  last bit is  $z$ , so for given  $\Theta(z)$  last bit is  $z$  and hiding property doesn't hold for one bit numbers.

### SHA-256 (Secure Hash Algorithm)

It is one of the cryptographic hash function which has digest length of 256 bits. It's a

keyless hash function. It was developed by

National Institute of Science Standards & Technology.

Five requirements for SHA256 :-

- 1) It is one way - cannot restore data from hash value
- 2) It is deterministic
- 3) Fast computation
- 4) The Avalanche Effect
- 5) Must withstand Collisions

Name - Kumar Nikhil

Roll no - 1806032

Branch - CS

Program code - UG - CS

Course Title - Blockchain Technology Course Code - CSE475

Exam Date - 19<sup>th</sup> May '21

### Question-2

- a) No, the transactions that were created all require a digital signature. Creating a valid digital signature requires the private key. If the public key is fixed, the specific private key that was used required in order to modify the transaction.

An ISP doesn't have their own private keys. So they will be unable to produce a conflicting transaction as they will be unable to create a valid transaction. They cannot produce a valid signature, so a double spend cannot be made. The only thing ISP can do is censor a user's transactions.

If a block is found now then the next block will be found soon and there is a small probability that it will take a long time to find the next block.

10 min

Probability Density

time for next block  $\rightarrow$

Mean time to find a block = 10 minutes / fraction of hash power.

(1) Probability to find the next block in next 10 min = Mean power  $\times$  50%  $\approx$

Name - KUMAR NIRHIL

Roll no - 1806082

Course code - CS6475

b) It is a Poisson process

$$\text{Confidence process} = 99\% = 0.99$$
$$\lambda = 6$$

$$\text{Therefore } P(X=x) = \frac{(e^{1(-0.99)})^x (0.99)^6}{6!}$$

Evaluating we get,

$$x = 0.485\%$$

### Question-3

#### E-VOTING

a) I structure, variable and varish contract declaration

contract Election {

/ Model a candidate struct Candidate {

uint id; string name;

uint votecount; }

constructor() public { addCandidate("candidate 1");  
addCandidate("candidate 2"); }

contract Voter {

pragma solidity >=0.7.0 <0.9.0;

contract Ballot {

struct Voter {

Name - KUMAR NIKHIL

Roll no - 1806032

Course Code - CS6175

uint weight;

bool voted;

address delegate;

uint vote; }

struct Proposal {

bytes32 name;

uint voteCount; }

address public chairperson;

mapping ( address  $\Rightarrow$  voter ) public voters;

Proposal [] public proposals;

constructor ( bytes32[] memory proposalNames ) {

chairperson = msg.sender;

voters[chairperson].weight = 1;

}

function giveRightToVote ( address voter ) public {

require ( msg.sender == chairperson );

require ( !voters[voter].voted );

require ( voters[voter].weight == 0 );

voters[voter].weight = 1; }

function delegate ( address to ) public {

Voter storage sender = voters[msg.sender];

require ( sender.weight != 0, "Has no weight to vote" );

sender.voted = true;

sender.vote = proposal;

}

Name- KUMAR NIKHIL

Roll no- 1806032

Course Code- CS6475

```
function winnerName() public view  
    returns [bytes32 winnerName] {  
    winnerName = proposals[miningProposals()].name;  
}
```

- b) The main reason for expensive Bitcoin mining fees is supply and demand: the Bitcoin size is 1 MB which means miners can only confirm 1 MB worth of transactions for each block (one every 10 minutes). As a result, mining fees sky-rocketed.

- Transaction creation and signing are the most expensive in the Bitcoin mining process and Bitcoin transaction fees are directly proportional to size of bytes of your transaction.
- Similarly sending bitcoin transactions is a lot like sending mail through postal service. If you send a large package, shipment will take longer and be more expensive.
- Bitcoin fees are similar sensitive to size of transaction.
- 'Merkle Tree' are used in Bitcoin to summarize all transactions in a block, producing an overall digital fingerprint for verification of transaction. This data structure speed up verification.

STEPS:

STEP 1: Transaction creation and signing ✓

STEP 2: Broadcasting

STEP 3: Propagation and Verification

STEP 4: Validation

Name - KUMAR NIKHIL

Roll no - 1806032

Course Code - CS6475

#### Question-4

(i) Digital Signature Attacks :-

→ There are three types of digital signature attacks:-

(a) Chosen Message Attacks :-

It tricks the genuine users to digitally sign a message that user does not normally intend to sign.

(b) Known-message Attack :-

The attacker obtains some message that user sends and a key to create a new fault message and forge of user.

(c) Key-only attack :-

In this it is assumed that user ~~will~~ make some info public and attacker misuse the public information.

Cryptosystems Attack :-

←                          →  
Cryptanalytic attack      Implementation attack  
attacks the mathematical      attacks the specific  
weakness in the algorithm.      implementation of the cipher.

→ Cipher text only attack :-

The attacker knows only the ciphertext to be decoded.

Name - KUMAR NIKHIL

Roll no - 1806032

Course Code - CS6475

→ Known - plaintext attack:-

The attacker has a collection of plaintext ciphertext pairs and is trying to find the key to decrypt some other ciphertext.

→ Chosen Plaintext Attack:-

Attacker chooses the plaintext to be encrypted and has the ciphertext.

→ Chosen Ciphertext attack:-

Attacker can select any ciphertext and study the plaintext by decrypting them.

→ Chosen Text attack:-

The attacker has the ability required in the previous two attacks.

### (ii) RSA Scheme

Given,  $p = 809$ ,

$q = 751$

$n$  is  $p * q = 809 * 751 = 607559$

$\phi(n) = (p-1)*(q-1) = 808 * 750 = 606000$

As private key  $d = 23$

public key  $e$  will be

$$d * e \equiv 1 \pmod{\phi(n)}$$

$$23 * e \equiv 1 \pmod{606000}$$

$$\therefore e = 158087$$

Name- KUMAR NIKHIL

Roll no- 1806032

Course code - CS6475

Message will be signed with private key of sender  
and verified by public key of sender.

(a) Signing of  $M_1 = 100$  as

$$S_1 = M_1^d \bmod n = 100^{23} \bmod 607559 \\ = 223388$$

Verification of  $S_1 = 223388$  as:-

$$M_1 = S_1^e \bmod n = 223388^{158087} \bmod 607559 \\ = 100$$

(b) Signing of  $M_2 = 50$  as:-

$$S_2 = M_2^d \bmod n = 50^{23} \bmod 607559 \\ = 5627$$

Verification of  $S_2 = 5627$  as:-

$$M_2 = S_2^e \bmod n = 5627^{158087} \bmod 607559 \\ = 50$$

(c)  $M = M_1 * M_2 = 5000$

$$S = S_1 * S_2 = (223388 * 5627) \bmod 607559$$

$$\boxed{S = 572264}$$

$$S = M^d \bmod 607559$$

$$= 5000^{23} \bmod 607559$$

$$\boxed{S = 572264}$$

Hence Proved

END

9