Roll No: 20BCE204

Course Code and Course Name: 2CSDE93 Blockchain Technology

Practical No. 2

Aim: To create a blockchain and implement replay attacks on blockchain. Create a formal document showing the implementation steps and learning outcomes for the second practical.

Code:

*//imported hash algorithm from the crypto-js package*

const SHA256 = require("crypto-js/sha256");

*//created a class to represent a Block*

class Block{

constructor(index, timestamp, data, previousHash){

this.index = index;

this.timestamp = timestamp;

this.data = data;

this.previousHash = previousHash;

this.hash = this.generateHash();

}

generateHash(){

return SHA256(this.index + this.timestamp + this.previousHash + JSON.stringify(this.data)).toString()

}

}

class Blockchain{

constructor(){

this.blockchain = [this.createGenesisBlock()];

}

createGenesisBlock(){

return new Block(0, "11/04/2022", "This Is Genesis Block", "0");

}

getTheLatestBlock(){

return this.blockchain[this.blockchain.length - 1];

}

addNewBlock(newBlock){

newBlock.previousHash = this.getTheLatestBlock().hash;

newBlock.hash = newBlock.generateHash();

this.blockchain.push(newBlock);

}

*// testing the integrity of the chain*

validateChainIntegrity(){

for(let i = 1; i<this.blockchain.length; i++){

const currentBlock = this.blockchain[i];

const previousBlock = this.blockchain[i-1];

if(currentBlock.hash !== currentBlock.generateHash()){

return false;

}

if(currentBlock.previousHash !== previousBlock.hash){

return false;

}

}

return true;

}

}

*// Create an instance to test our blockchain*

let MyCoin = new Blockchain();

console.log("Mining MyCoin in Progress...");

MyCoin.addNewBlock(

new Block(1, "01/08/2023", {

sender: "ParthPatel",

recipient: "DhruvilPatel",

quantity: 25

})

);

MyCoin.addNewBlock(

new Block(2, "01/08/2023", {

sender: "DhruvilPatel",

recipient: "ParthPatel",

quantity: 34

})

);

MyCoin.addNewBlock(

new Block(3, "01/08/2023", {

sender: "ParthPatel",

recipient: "DhyanPatel",

quantity: 34

})

);

console.log(JSON.stringify(MyCoin, null, 5))

console.log("\n\n\nChecking Chain Integrity:-\n");

console.log((MyCoin.validateChainIntegrity())?"The Chain Is Valid":"The Chain is NOT-Valid");

console.log("Tempered Hash Of Latest Block, and Checking Chain Integrity:");

MyCoin.getTheLatestBlock().hash = MyCoin.getTheLatestBlock().hash.replace('a','b');

*// console.log(MyCoin.getTheLatestBlock().hash)*

console.log((MyCoin.validateChainIntegrity())?"The Chain Is Valid":"The Chain is NOT-Valid");

console.log("\n\n");

Output:

Mining MyCoin in Progress...

{

"blockchain": [

{

"index": 0,

"timestamp": "11/04/2022",

"data": "This Is Genesis Block",

"previousHash": "0",

"hash": "733edc4a434754ef537c6e8fdeceaca64ee2ef4951c3eb6cde8bdfbcca23d648"

},

{

"index": 1,

"timestamp": "01/08/2023",

"data": {

"sender": "ParthPatel",

"recipient": "DhruvilPatel",

"quantity": 25

},

"previousHash": "733edc4a434754ef537c6e8fdeceaca64ee2ef4951c3eb6cde8bdfbcca23d648",

"hash": "03cf0aa42cb2f6801898e0dbf71b8cd58835999775c07eab219695ddf9e18b92"

},

{

"index": 2,

"timestamp": "01/08/2023",

"data": {

"sender": "DhruvilPatel",

"recipient": "ParthPatel",

"quantity": 34

},

"previousHash": "03cf0aa42cb2f6801898e0dbf71b8cd58835999775c07eab219695ddf9e18b92",

"hash": "362bd80616c64610b2dfcbec5bcec75111b92df0e888e91bdd6b4f229d35d5b4"

},

{

"index": 3,

"timestamp": "01/08/2023",

"data": {

"sender": "ParthPatel",

"recipient": "DhyanPatel",

"quantity": 34

},

"previousHash": "362bd80616c64610b2dfcbec5bcec75111b92df0e888e91bdd6b4f229d35d5b4",

"hash": "ee5eff543dc501512adc3dc74842a9e719dcc9941917651c6679d4850907e624"

}

]

}

Checking Chain Integrity:-

The Chain Is Valid

Tempered Hash Of Latest Block, and Checking Chain Integrity:

The Chain is NOT-Valid