**Lab 2**

Npm init

npm install crypto

**Thuật toán des**

var crypto = require('crypto');

let key=Buffer.from('jhsgatt12sgsssqswhwqfsaxasxuasxs','base64');

let cipher = crypto.createCipheriv("des-ede3", key, null);

let encryptedData = cipher.update("The quick brown fox jumps over the lazy dog", "utf8", "base64");

encryptedData += cipher.final("base64");

console.log(encryptedData);

const decipher = crypto.createDecipheriv("des-ede3", key, null);

let decryptedData = decipher.update(encryptedData, "base64", "utf8");

decryptedData += decipher.final("utf8");

console.log("Decrypted message: " + decryptedData);

**Thuật toán AES**

// crypto module

const crypto = require("crypto");

const algorithm = "aes-256-cbc";

// generate 16 bytes of random data

const initVector = crypto.randomBytes(16);

// protected data

const message = "This is a secret message";

// secret key generate 32 bytes of random data

const Securitykey = crypto.randomBytes(32);

// the cipher function

const cipher = crypto.createCipheriv(algorithm, Securitykey, initVector);

// encrypt the message

let encryptedData = cipher.update(message, "utf-8", "hex");

encryptedData += cipher.final("hex");

console.log("Encrypted message: " + encryptedData);

// the decipher function

const decipher = crypto.createDecipheriv(algorithm, Securitykey, initVector);

let decryptedData = decipher.update(encryptedData, "hex", "utf-8");

decryptedData += decipher.final("utf8");

console.log("Decrypted message: " + decryptedData);

**Thuật toán RSA**

const crypto = require("crypto");

// The `generateKeyPairSync` method accepts two arguments:

// 1. The type ok keys we want, which in this case is "rsa"

// 2. An object with the properties of the key

const { publicKey, privateKey } = crypto.generateKeyPairSync("rsa", {

  // The standard secure default length for RSA keys is 2048 bits

  modulusLength: 2048,

});

// This is the data we want to encrypt

const data = "my secret data";

const encryptedData = crypto.publicEncrypt(

  {

    key: publicKey,

    padding: crypto.constants.RSA\_PKCS1\_OAEP\_PADDING,

    oaepHash: "sha256",

  },

  // We convert the data string to a buffer using `Buffer.from`

  Buffer.from(data)

);

console.log("encypted data: ", encryptedData.toString("base64"));

const decryptedData = crypto.privateDecrypt(

    {

      key: privateKey,

      padding: crypto.constants.RSA\_PKCS1\_OAEP\_PADDING,

      oaepHash: "sha256",

    },

    encryptedData

  );

console.log("decrypted data: ", decryptedData.toString());

Thuật toán SHA1

const crypto = require('crypto');

const data = 'xin chao';

const sh1Data = crypto.createHash('sha1').update(data).digest('hex');

console.log(sh1Data) // 8867d36fe297487f78951976c6c3bd0653ffb05a