**CRYPTOGRAPHY CASE STUDY REPORT**

Date: 17/11/2020

**Name: Anand Devarajan**

**Roll No: CB.EN.U4CSE18207**

**CODE**

'use strict';

const data = require('./data.json');

const crypto = require('crypto');

const { publicKey } = crypto.generateKeyPairSync('rsa', {

modulusLength: 2048,

});

const fs = require('fs');

const encryptionEncoding = 'base64';

const bufferEncryption = 'utf-8';

const bcrypt = require('bcrypt');

const json2csv = require('json2csv').Parser;

const startTime = new Date();

//RSA

const encryptionRSA = (message) => {

const encryptedData = crypto.publicEncrypt(

{

key: publicKey,

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

oaepHash: 'sha256',

},

Buffer.from(message)

);

return encryptedData.toString('base64');

};

//AES

const encryptionAES = (message) => {

const aesKey = Buffer.from(

'xNRxA48aNYd33PXaODSutRNFyCu4cAe/InKT/Rx+bw0=',

'base64'

);

const aesiv = Buffer.from('81dFxOpX7BPG1UpZQPcS6w', 'base64');

const cipher = crypto.createCipheriv('aes-256-cbc', aesKey, aesiv);

let encrypted =

cipher.update(message, bufferEncryption, encryptionEncoding) +

cipher.final('base64');

return encrypted;

};

//VIGENERE

const encryptionVigenere = (message, key) => {

let cipher = '';

message = message.toUpperCase();

for (let i = 0; i < message.length; i++) {

if (message[i] === ' ') {

cipher += message[i];

} else {

cipher += String.fromCharCode(

((message.charCodeAt(i) + key.charCodeAt(i)) % 26) + 65

);

}

}

return cipher;

};

//HASHING

const encryptionHash = (message) => {

const hash = bcrypt.hashSync(message, 10);

return hash;

};

(async () => {

let newData = [];

for (const item of data) {

const encry\_CVV = encryptionRSA(JSON.stringify(item.CVV));

let encry\_Account\_No = encryptionAES(

JSON.stringify(item['Account Number'])

);

let encty\_Phone\_no = encryptionVigenere(

JSON.stringify(item['Phone no']),

'CBENU4CSE18207'

);

let encry\_Name = encryptionHash(item.Name);

let userID = item.UserID;

newData.push({

userID,

encry\_Name,

encty\_Phone\_no,

encry\_Account\_No,

encry\_CVV,

});

}

const j2cp = new json2csv();

const csv = j2cp.parse(newData);

fs.writeFileSync('./out.csv', csv, 'utf-8');

const endTime = new Date();

console.log(

'Total time take = ' +

(endTime.getTime() - startTime.getTime()) / 1000 +

' seconds'

);

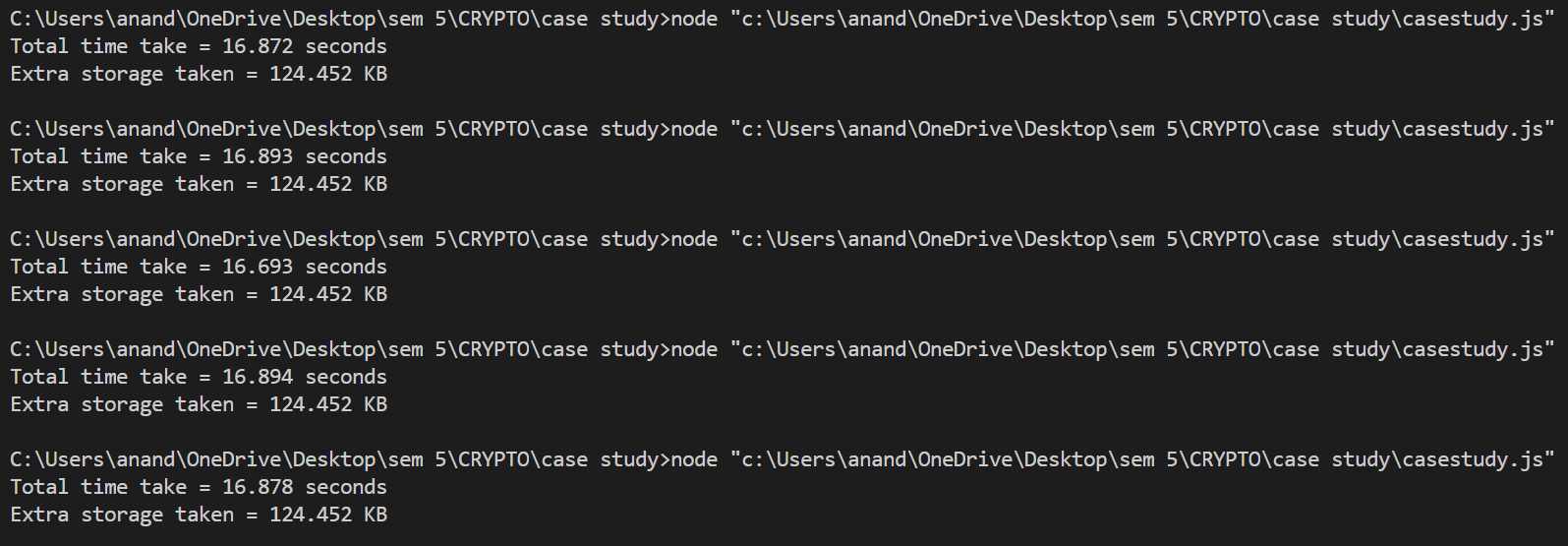
})();

var original = fs.statSync('./CaseStudy.csv').size;

var encrypted = fs.statSync('./Out.csv').size;

console.log('Extra storage taken = ' + (encrypted - original) / 1000 + ' KB');

**Output** (Checking total time taken for 5 different times)



1. Check the size difference between the **PT** file and **CT** file

Size difference = 124.452 KB

1. Check the time taken to totally encrypt all five columns

(finding average of total time obtained on 5 different time)

Total time taken =

(16.872 + 16.893 + 16.693 + 16.894 + 16.878)/5 = 16.846 seconds

1. Description of dataset and columns chosen

The dataset chosen for the case study is about banking details about a person.

The 5 columns chosen for the dataset are userID,name,phone no, Account no and CVV

1. Why each columns security is important