Homework #4
EE 547: Spring 2025

Name: Nissanth Neelakandan Abirami
USC ID: 2249203582
Instructor: Dr. Franzke

1)

If we start with store.json we can encrypt and check if the code creates decrypt.json by which we can confirm our code which encrypts store.json so we cannot view message and then after decryption we can check if message appears in decrypt.json

```
fs.writeFileSync('decrypt.json', JSON.stringify(JSON.parse(decryptedData.toString()), null, 2));
console.log('Decrypted data saved to decrypt.json');
```

Terminal:

```
nissanthneelakandanabirami@Nissanths-MacBook-Pro Cloud-EE-547 % cd Hw_4
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson

Enter the path to your private key: private.pem
Store successfully decrypted.

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "nissanth",
  "email": "nissanth@usc.edu",
  "message": "Hi"
}

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 1
Enter key: Hello
Enter value: Test
Added/Updated Hello: Test
```

```
        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "nissanth",
  "email": "nissanth@usc.edu",
  "message": "Hi",
  "Hello": "Test"
}

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 5
Exiting without encryption
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl pkeyutl -decrypt -in
store.encjson -inkey private.pem -out decrypted.json -pkeyopt rsa_padding_mode:oaep

Can't open "store.encjson" for reading, No such file or directory
4048490902000000:error:80000002:system library:BIO_new_file:No such file or
directory:crypto/bio/bss_file.c:67:calling fopen(store.encjson, rb)
4048490902000000:error:10000080:BIO routines:BIO_new_file:no such
file:crypto/bio/bss_file.c:75:
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson


        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 5
Exiting without encryption
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl pkeyutl -decrypt -in
store.encjson -inkey private.pem -out decrypted.json -pkeyopt rsa_padding_mode:oaep

Could not open file or uri for loading private key from private.pem: No such file or
directory
pkeyutl: Error initializing context
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl pkeyutl -decrypt -in
store.encjson -inkey private.pem -out decrypted.json -pkeyopt rsa_padding_mode:oaep

Could not open file or uri for loading private key from private.pem: No such file or
directory
```

```
pkeyutl: Error initializing context
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl genpkey -algorithm RSA
-out /keys/private.pem -pkeyopt rsa_keygen_bits:2048
genpkey: Can't open "/keys/private.pem" for writing, No such file or directory
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl genpkey -algorithm RSA
-out private.pem -pkeyopt rsa_keygen_bits:2048
.......+...+..........+..+.......+......+++++++++++++++++++++++++++++++++++++++*...+..
..........+...+..+...+............+.+..+....+++++++++++++++++++++++++++++++++++++++*..
.........+...+.......+..+......+....+..+....+...........+....+..+...+......+......+...
....+......+..+...+....+.....+.......+...+.....++++++
...........+....+......+.....+.+............+..............+..+....+......++++++++++
++++++++++++++++++++++++++*.........+..................+.............+..+...+.+...
.....+......+......+....+.....+....+...+..+.......+...+........+...+......++++++++++
++++++++++++++++++++++++++*......+..+...+...+.+...+...+.....+............+..+....+..
.+......++++++
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl rsa -pubout -in
/keys/private.pem -out public.pem
Could not open file or uri for loading private key from /keys/private.pem: No such
file or directory
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl rsa -pubout -in
private.pem -out public.pem
writing RSA key
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson


        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 1
Enter key: hi
Enter value: hello
Added/Updated hi: hello

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "hi": "hello"
}

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit
```

```
Choose an option (1-5): 5
Exiting without encryption
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl genpkey -algorithm RSA
-out private.pem -pkeyopt rsa_keygen_bits:2048
..+...+...++++++++++++++++++++++++++++++++++++++++*............++++++++++++++++++++++
+++++++++++++++*....+....+.....+......+......+.............+...+...+.......+......+.
...........+..+.+..+.........+...+..........+...+.+.........+..........+......+....
.....+......+......+.......+..+....+.....+.+...+........+............+..........+...
.......+...+...+..+.......+...+...+........+......+......+.......+.....+......+.......
+..+.+.....+.+......+.............+.+.........+........+.......+.........+.........+.
.....+..+....+...+..+...+..........+.....+...+...+....+.....+.+.....................
...+.+.....+.......+...........+....+...+.....+....+.....+....+.............+...+....
+.....+.............+.....+....+.....+......+...+.................+..+.+.........
...+..+....+.....+..........+..+...........+.......+...........+....+.....+.........
+.........+..+...+...........+.......+.....+.+.........+...+..+......+.............
......+.........+.............+.....+......+...+....+.....+...........+....+..+...
.......+.............................+.+..+...+.+........+......+......+.+.........
......+..+.+...........+...+...+.......+......+.....+....+..+.+...+................
..+.....+.........+.+......+...+..+.........+.+.....+....+.....+...+......+.+........+
......+.......+..+......+.+........+.+..+...........+...+.......+................+..
..+.............+.....+.......+...+..+...++++++
.+......++++++++++++++++++++++++++++++++++++++*....+......+...+.+.....+......+...+...
....++++++++++++++++++++++++++++++++++++*.........+....+...........+...........+..
...+......+.......+..+......+......+.+...+.....+..........+..+.......+..+...+......+..
..+..+......+...+....+........+........................+..+.+..+...+....+..+.+.......
..+.....+....+......+........+..........+...+................+.....................
..+...+..+......+....+..+.........+.+......+..+.......+...+.....+......+.+...+........
+...+....+.....+.+......+...........+.......+...+.....+.........+...+......+......+...
.+...........+.+.........+.....+......+.........+...+...+....+...+..++++++
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl rsa -pubout -in
private.pem -out public.pem
writing RSA key
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson
Enter the path to your private key: private,pem
Decryption failed.
Error details: ENOENT: no such file or directory, open 'private,pem'
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson
Enter the path to your private key: private.pem
Decryption failed.
Error details: error:0200006F:rsa routines::data too small
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl genpkey -algorithm RSA
-out private.pem -pkeyopt rsa_keygen_bits:2048
........+.....+.+...+.....+.........+..........+..+...+......+.......+++++++++++++++
++++++++++++++++++++*..+.+..+...+.+...+..+.........+.+...+..+.........+....+........
...............+...+.......+..+.+..+.......+...+..+.++++++++++++++++++++++++++++++++
+++++*.............+.........+.+................+...+.................+.+.....+....+
.....+....+...+.....+.......+.....+.+..+.......+...........+......+.+........+.......+
...+......+...+.................+......+.......+..+..........+...+.....++++++
..+.......+.............+.........+.....+.........+......+.......+.....+............
...+....+...+.........+...+.+......+..+.+.....+.+......+...+.....+.+........++++++++++
++++++++++++++++++++++++++*....+.....+.+.....+.......+.............................
...+..+....+...+...+..+......+.+...+...++++++++++++++++++++++++++++++++++++*......+
........+.+...........+...+...........+.........+...+......+.++++++
```

```
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % openssl rsa -pubout -in
private.pem -out public.pem
writing RSA key
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % touch store.encjson

nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson

Enter the path to your private key: private.pem
Decryption failed.
Error details: error:0200006F:rsa routines::data too small
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson

Store is not encrypted. Proceeding with the existing data.

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "nissanth",
  "email": "nissanth@usc.edu",
  "message": "Hi"
}

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 1
Enter key: name
Enter value: Hankm
Warning: Key 'name' already exists. Value was overwritten.
Added/Updated name: Hankm

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "Hankm",
  "email": "nissanth@usc.edu",
  "message": "Hi"
}
```

```
        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 2
Enter key to delete: message
Deleted key: message

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "Hankm",
  "email": "nissanth@usc.edu"
}

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 4
Enter the path to your public key: public.pem
Store encrypted and saved.
nissanthneelakandanabirami@Nissanths-MacBook-Pro Hw_4 % node app.js store.encjson

Enter the path to your private key: private.pem
Store successfully decrypted.

        Menu:
        (1) Add key-value
        (2) Delete key
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 3
Current store: {
  "name": "Hankm",
  "email": "nissanth@usc.edu"
}

        Menu:
        (1) Add key-value
        (2) Delete key
```

```
        (3) Show current key-value store
        (4) Encrypt key-value store
        (5) Exit

Choose an option (1-5): 5
Exiting without encryption
```

Files:

`Store.encjson`

```
{
   "name": "nissanth",
   "email": "nissanth@usc.edu",
   "message":"Hi"
}
```

Code:

```
const fs = require('fs');
const crypto = require('crypto');
const readline = require('readline');

// Get command-line arguments
const args = process.argv.slice(2);
const STORE_FILE = args[0];

// Display help
if (!STORE_FILE || args.includes('--help')) {
 console.log(`
       Usage: node app.js [store.encjson] [--help]

       Description:
       This application manages a secret key-value store with encryption and
       decryption using RSA public/private keys. On startup, if the store file
       exists, the application prompts for a private key to decrypt the file.
       After interacting with the key-value store, you can encrypt it with a
       public key before saving.

       Options:
       store.encjson Path to the key-value store file (required).
       --help Display this help message and exit.

       Examples:
```

```
      node app.js store.encjson       # Start with a store file.
      node app.js --help              # Display this help message.
 `);
 process.exit(0);
}

//https://nodejs.org/api/readline.html
let store = {}; // The key-value store
const rl = readline.createInterface({
 input: process.stdin,
 output: process.stdout
});

//https://stackoverflow.com/questions/66423321/node-crypto-publicencrypt-returns-diffe
rent-value-each-time-it-is-used
async function askQuestion(query) {
 return new Promise((resolve) => {
   rl.question(query, (answer) => {
     resolve(answer);
   });
 });
}

// Function to check if a file is encrypted or not
//https://piazza.com/class/m5wazi2qyon2u3/post/26
const isEncrypted = (filePath) => {
 try {
   const data = fs.readFileSync(filePath, 'utf8');
   JSON.parse(data);
   return false;
 } catch {
   return true;
 }
};

// Function to decrypt the store file using the provided private key
const decryptStore = (privateKeyPath) => {
 try {
   const encryptedData = fs.readFileSync(STORE_FILE);
   const privateKey = fs.readFileSync(privateKeyPath, 'utf8');
   const decryptedData = crypto.privateDecrypt(
     { key: privateKey,
```

```
      padding: crypto.constants.RSA_PKCS1_OAEP_PADDING },
    encryptedData
  );
  return JSON.parse(decryptedData.toString());
 } catch (error) {
  if (error instanceof SyntaxError) {
    console.error('Invalid store file.');
    process.exit(2);
  }
  console.error('Decryption failed.');
  console.error(`Error details: ${error.message}`);
  process.exit(3);
 }
};

// Function to encrypt and save the store file using the provided public key
const encryptStore = async () => {
 const publicKeyPath = await askQuestion('Enter the path to your public key: ');
 try {
  const publicKey = fs.readFileSync(publicKeyPath, 'utf8');
  const dataBuffer = Buffer.from(JSON.stringify(store));
  const encryptedData = crypto.publicEncrypt(
    { key: publicKey, padding: crypto.constants.RSA_PKCS1_OAEP_PADDING },
    dataBuffer
  );
  fs.writeFileSync(STORE_FILE, encryptedData);
  console.log('Store encrypted and saved.');
  process.exit(0);
 } catch (error) {
  if (error.message.includes('invalid key')) {
    console.error('Invalid public key file.');
  } else {
    console.error('Encryption failed.');
    console.error(`Error details: ${error.message}`);
  }
 }
};

// Function to validate the format of a key
const isValidKey = (key) => /^[a-zA-Z0-9 _-]+$/.test(key) && !key.startsWith(' ') &&
!key.endsWith(' ');
```

```
// Main function to handle the CLI menu
const main = async () => {
 if (fs.existsSync(STORE_FILE)) {
   if (isEncrypted(STORE_FILE)) {
     try {
       const privateKeyPath = await askQuestion('Enter the path to your private key:
');
       store = decryptStore(privateKeyPath);
       console.log('Store successfully decrypted.');
     } catch {
       console.error('Invalid store file.');
       process.exit(2);
     }
   } else {
     console.log('Store is not encrypted. Proceeding with the existing data.');
     try {
       store = JSON.parse(fs.readFileSync(STORE_FILE, 'utf8'));
     } catch {
       console.error('Invalid store file.');
       process.exit(2);
     }
   }
 }
 // CLI Menu
 while (true) {
   console.log(`
       Menu:
       (1) Add key-value
       (2) Delete key
       (3) Show current key-value store
       (4) Encrypt key-value store
       (5) Exit
   `);
   const choice = await askQuestion('Choose an option (1-5): ');
   switch (choice.trim()) {
     case '1':
       while (true) {
         const key = await askQuestion('Enter key: ');
         if (!key) break;
         if (!isValidKey(key)) {
           console.log('Invalid key. Please enter a valid key.');
           continue;
```

```
        }
        const value = await askQuestion('Enter value: ');
        if (store[key]) {
          console.log(`Warning: Key '${key}' already exists. Value was
overwritten.`);
        }
        store[key] = value;
        console.log(`Added/Updated ${key}: ${value}`);
        break;
      }
      break;

    case '2':
      const delKey = await askQuestion('Enter key to delete: ');
      if (!delKey) break;
      if (store[delKey]) {
        delete store[delKey];
        console.log(`Deleted key: ${delKey}`);
      } else {
        console.log('Key not found.');
      }
      break;

    case '3':
      console.log('Current store:', JSON.stringify(store, null, 2));
      break;

    case '4':
      await encryptStore();
      break;

    case '5':
      console.log('Exiting without encryption');
      rl.close();
      process.exit(0);

    default:
      console.log('Invalid option. Please choose a no between 1 and 5.');
  }
 }
};
main();
```