**UNIVERSITY OF SCIENCE – HCMVNU**

**FACILITY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**LAB**

**Subject: Application and Cryptography**

*RSA Encryption and Signature Lab*

Tp. Hồ Chí Minh – 03/2022

Mục lục

[Tổng quan: 3](#_Toc98231805)

[Đánh giá 3](#_Toc98231806)

[Bài 1 3](#_Toc98231807)

[1. Thiết lập mô hình mạng 3](#_Toc98231808)

[2. Yêu cầu 4](#_Toc98231809)

[Bài 2 14](#_Toc98231810)

# Overview:

|  |  |
| --- | --- |
| **Student ID** | **Full name** |
| 19127525 | Nguyễn Thanh Quân |
| 19127625 | Lâm Chí Văn |

# Task 1: Deriving the Private Key

Function **Task1(…)** in file **task1.c** will return *private key d*

Text

Description automatically generated

Folow [Wiki/RSA\_(cryptosystem)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))

A picture containing text, watch

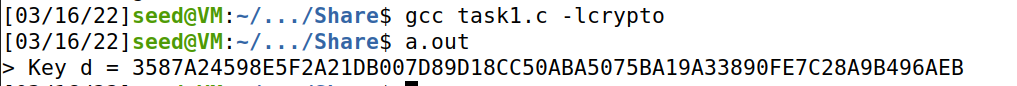
Description automatically generated

With

First, I *calculate* (line 19 to 28)

Finnally, use **BN\_mod\_inverse(…)** function of **openssl/bn.h** library (line 31) to *calculate* ***d***

Result: d = 3587A24598E5F2A21DB007D89D18CC50ABA5075BA19A33890FE7C28A9B496AEB



# Task 2: Encrypting a Message

Function **Task2\_Encrypt(…)** in file **task2.c** will return *encrypt message (ciphertext)* with Hex format

Use formula to calculate ciphertext with Hex format

Fisrt, *convert* ***message*** *from ASCII string format to Hex string* by **string2hexString(…)**(line 46)*.*

Then init its to BIGNUM variable (m) (line 47)

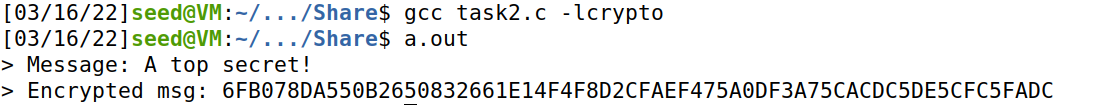
Finally, use formula (line 50) to *find* ***ciphertext***

Text

Description automatically generated

Result:

Encrypt message: 6FB078DA550B2650832661E14F4F8D2CFAEF475A0DF3A75CACDC5DE5CFC5FADC



# Task 3 : Decrypting a Message

Function **Task3\_Decrypt(…)** in file **task3.c** will print *Decrypted message (plaintext)* with ASCII string format

Use formula to *calculate* ***plaintext*** with Hex format

Fisrt, use formula to *find* ***Decrypt message*** *with Hex string* format by **BN\_mod\_exp(..)** (line 41)

Finally *convert* ***m*** *from Hex string to ASCII string* by **textFromHexString(…)** (line 47)

* Find origin message (plaintext)



Result:

Origin message: Password is dees

Text

Description automatically generated

# Task 4: Signing a Message

Function **Task4\_RSAsign(…)** in file **task4.c** will return *sign of message* in Hex format

With instruction in <https://cryptobook.nakov.com/digital-signatures/rsa-signatures>

Graphical user interface, text, application, email

Description automatically generated

We must hash msg first but follow instruction in **Task 4**, we do not need to hash this message.

So, we just *convert this* ***message*** *from string to Hex string* by **string2hexString(…)** (line 46) (consider this value is h(msg))

Finally, like Decrypt message, *use private key d to* ***sign this message*** by formula to by **BN\_mod\_exp(..)** (line 50)

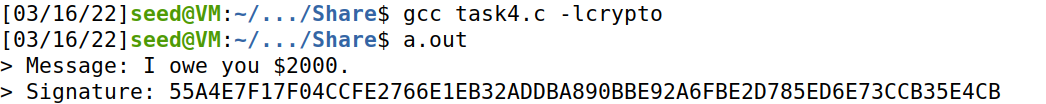
* We had *Sign this message*

Text

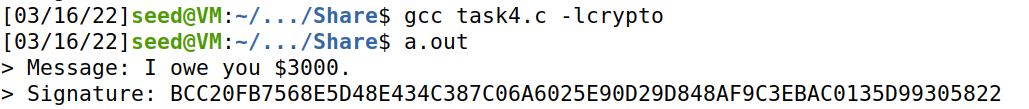
Description automatically generated

Result:

With Message: “I owe you $2000.”, we will have sign is 55A4E7F17F04CCFE2766E1EB32ADDBA890BBE92A6FBE2D785ED6E73CCB35E4CB



With Message: “I owe you $3000.”, we will have sign is BCC20FB7568E5D48E434C387C06A6025E90D29D848AF9C3EBAC0135D99305822



From 2 result above, we can see, these sign is **complete different**

55A4E7F17F04CCFE2766E1EB32ADDBA890BBE92A6FBE2D785ED6E73CCB35E4CB

and

BCC20FB7568E5D48E434C387C06A6025E90D29D848AF9C3EBAC0135D99305822

# Task 5: Verifying a Signature

Function **Task5\_VerifySignature(…)** in file **task5.c** will *return 0 or 1 with (0 is True and 1 is False)*

With instruction in <https://cryptobook.nakov.com/digital-signatures/rsa-signatures>

Graphical user interface, text, application, email

Description automatically generated

We must hash msg fisrt but follow instruction in **Task 5**, we do not need to hash this message.

First, we *calculate* ***h’*** by formula (line 45) (here, I assign h2 as h’)

Then, we *calc* ***h = h(M)*** by *convert* ***M*** *from ASCII string to hex string* by **string2hexString(…)** (line 48)

Finally, we *compare* ***h*** *and* ***h’*** (line 54).

* If return value is 0 this sign is True, else this sign is False

Text

Description automatically generated

With S = 643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB680**2F**

We have h and h’ value is 4C61756E63682061206D697373696C652E

h and h’ is equal Result: This signature is indeed Alice's

Text

Description automatically generated

With S = 643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB680**3F**

We have h value is: 4C61756E63682061206D697373696C652E

And h’ value is: 91471927C80DF1E42C154FB4638CE8BC726D3D66C83A4EB6B7BE0203B41AC294

h and h’ is different Result: This signature is NOT indeed Alice's

Text, application

Description automatically generated