0BÁO CÁO THỰC HÀNH

**Môn học: CRYPTOGRAPHY – Mật mã học**

**Tên chủ đề: PUBLIC-KEY CRYPTOGRAPHY**

*GVHD: Tô Trọng Nghĩa*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: **NT219.N21.ANTT.1**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email |
| 1 | Lê Đoàn Trà My | 21521149 | 21521149@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nội dung | Tình trạng | Trang |
| 1 | Number Theory | 100% | 02 – 04 |
| 2 | RSA Public-Key Encryption | 100% | 04 – 12 |
| 3 | RSA Application | 100% | 12 – 16 |
| 4 | Programming using the Crypto Library | 100% | 16 – 20 |
| Điểm tự đánh giá | | | **8.5-9/10** |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

**1. Number Theory**

**\* Thuật toán Miller-Rabin để kiểm tra các số nguyên tố lớn:**

- Là một phương pháp kiểm tra số nguyên tố theo thuật toán xác suất.

- Thuật toán Miller-Rabin hoạt động như sau:

*1. Chọn một số nguyên dương n cần kiểm tra tính nguyên tố.*

*2. Tìm số m là sao cho n-1 = m \* 2^k với k là một số nguyên dương.*

*3. Chọn một số nguyên dương a ngẫu nhiên trong khoảng [2, n-2].*

*4. Tính b = a^m mod n.*

*5. Nếu b = 1 hoặc b = n-1, thì n có thể là số nguyên tố, quay lại bước 3 và chọn một giá trị a khác.*

*6. Với i từ 1 đến k-1, tính b = b^2 mod n. Nếu b = n-1, thì n có thể là số nguyên tố, quay lại bước 3 và chọn một giá trị a khác.*

*7. Nếu b khác n-1 sau khi đã tính k lần, thì n không phải là số nguyên tố.*

**\* Thuật toán Euclidian để tìm ước chung lớn nhất (gcd):**

Cho a, b là hai số nguyên (giả sử a ≥ b), để tìm ước chung lớn nhất của hai số a và b ta cần thực hiện chia a cho b được thương q và số dư r (r ≥ 0) tức là a = b\*q + r, khi đó ta có:

Text

Description automatically generated with medium confidence

*Hình 1. Thuật toán Euclidian*

**\* Viết chương trình (C/C++/C#) thực hiện các yêu cầu:**

*Em sử dụng các hàm và thư viện hỗ trợ trong CryptoPP.*

***a, Số nguyên tố:***

- Tạo số nguyên tố ngẫu nhiên 2 bytes, 8 bytes, 32 bytes.

Text

Description automatically generated

*Hình 2. Hàm sinh số nguyên tố ngẫu nhiên n bit có sử dụng các hàm có sẵn của CryptoPP*

Text

Description automatically generated

*Hình 3. Đoạn code về phần sinh số nguyên tố trong hàm main()*

- Kiểm tra một số nguyên tuỳ ý nhỏ hơn 289 -1 có phải là số nguyên tố ?

Text

Description automatically generated

*Hình 4. Hàm kiểm tra số nguyên tố sử dụng các hàm có sẵn của CryptoPP*

A picture containing text

Description automatically generated

*Hình 5. Đoạn code kiểm tra số có là số nguyên tố trong hàm main()*

***b, Xác định ước chung lớn nhất (gcd) của 2 số “lớn” tuỳ ý.***

***Text

Description automatically generated***

*Hình 6. Hàm tính ƯCLN của 2 số sử dụng các hàm có sẵn của CryptoPP*

*Text

Description automatically generated with medium confidence*

*Hình 7. Đoạn code tính ƯCLN của 2 cặp số ví dụ trong hàm main()*

***c, Tính ax mod p (với x > 40).***

***Graphical user interface, text

Description automatically generated***

*Hình 8. Hàm tính ax mod p sử dụng các hàm có sẵn của CryptoPP*

***A picture containing diagram

Description automatically generated***

*Hình 9. Đoạn code tính ax mod p của 2 ví dụ trong hàm main()*

* Kết quả chạy code thu được:

***Text

Description automatically generated***

*Hình 10. Kết quả thu được sau khi chạy chương trình*

**2. RSA Public-Key Encryption**

**\* Các bước thuật toán RSA:**

*1. Chọn 2 số nguyên tố “lớn” p và q (p ≠ q) và tính n = p\*q.*

*2. Tính ɸ(n) = (p - 1) \* (q - 1).*

*3. Chọn số e sao cho: e là số nguyên tố cùng nhau với ɸ(n), e < ɸ(n).*

*4. Xác định d, với (e\*d) mod ɸ(n) = 1.*

*5. Khoá công khai PU = (e, n); khoá bí mật PR = (d, n).*

*6. Mã hoá văn bản gốc M:*

*- Để bảo mật: C = E(M, PU) = Me mod n;*

*- Để xác thực: C = E(M, PR) = Md mod n*

*7. Giải mã bản mã C:*

*- Để bảo mật: M = D(M, PR) = Cd mod n;*

*- Để xác thực: M = D(M, PU) = Ce mod n*

**\* Bài tập:**

***2.1. Xác định khoá công khai PU và khoá bí mật PR***

***Text

Description automatically generated***

*Hình 11. Đoạn code tính n và ɸ(n) hoặc dùng tool (*[*https://www.cryptool.org/en/cto/rsa-visual*](https://www.cryptool.org/en/cto/rsa-visual) *)*

(Em dùng tool <https://www.dcode.fr/modular-inverse> để tính toán d)

*a, p1 = 11, q1 = 17, e1 = 7 (decimal)*

- n1 = p1\* q1 = 11 \* 17 = 187

- ɸ(n1) = (p1 - 1) \* (q1 - 1) = 10 \* 16 = 160

- Chọn d = 23 (vì 23 \* 7 = 161 = 160 + 1)

→ PU = (7, 187); PR = (23, 187)

*b, p2 = 20079993872842322116151219, q2 = 676717145751736242170789, e2 = 17 (decimal)*

- n2 = 20079993872842322116151219\* 676717145751736242170789

= 13588476140342208394395166469647627226674348541791

- ɸ(n2) = (p2 - 1) \* (q2 - 1)

= 20079993872842322116151218 \* 676717145751736242170788

= 13588476140342208394395145712936608632615990219784

- Chọn d = 7993221259024828467291262184080358019185876599873 (dùng tool)

→ PU = (17, 13588476140342208394395166469647627226674348541791);

PR = (7993221259024828467291262184080358019185876599873,

13588476140342208394395166469647627226674348541791)

*c, p3 = F7E75FDC469067FFDC4E847C51F452DF,*

*q3 = E85CED54AF57E53E092113E62F436F4F, e3 = 0D88C3 (hexadecimal)*

- Đổi sang hệ thập phân:

p3 = 329520679814142392965336341297134588639,

q3 = 308863399973593539130925275387286220623, e3 = 886979

- n3 = p3 \* q3 =

101776877529005912638346811918779931246783058062684819617574643018368103302097

- ɸ(n3) = (p2 - 1) \* (q2 - 1) =

101776877529005912638346811918779931246144673982897083685478381401683682492836

- Chọn d =

24212225287904763939160097464943268930139828978795606022583874367720623008491 (dùng tool)

→ PU = (886979,

101776877529005912638346811918779931246783058062684819617574643018368103302097);

PR =

(24212225287904763939160097464943268930139828978795606022583874367720623008491, 101776877529005912638346811918779931246783058062684819617574643018368103302097)

***2. 2. Sử dụng khóa do p1, q1, e1 sinh ra để mã hóa và giải mã bản rõ M=5 trong cả hai trường hợp mã hóa để bảo mật và mã hóa để xác thực***

- Với kết quả từ 2.1.a, có: p1 = 11, q1 = 17, e1 = 7, d=23, PU = (7, 187); PR = (23, 187).

- Với M = 5:

+ Mã hoá để bảo mật: C = E(M, PU) = Me mod n = 57 mod 187 = 146

+ Mã hoá để xác thực: C = E(M, PR) = Md mod n = 523 mod 187 = 180

***2. 3. Sử dụng các khóa trên để mã hóa thông báo sau (bản mã trong thập lục phân)***

**The Faculty of Computer Networks and Communications**

- Đổi văn bản sang hệ 10 (dùng tool <https://www.rapidtables.org/vi/convert/number/ascii-hex-bin-dec-converter.html> )

**Text

Description automatically generated**

*Hình 12. Kết quả chuyển đổi của plaintext sang các hệ số*

- Thu được kết quả: 84 104 101 32 70 97 99 117 108 116 121 32 111 102 32 67 111 109 112 117 116 101 114 32 78 101 116 119 111 114 107 115 32 97 110 100 32 67 111 109 109 117 110 105 99 97 116 105 111 110 115

- Chia thành nhiều block nhỏ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 = 84 | P8 = 117 | P15 = 32 | P22 = 101 | P29 = 111 | P36 = 100 | P43 = 110 | P50 = 110 |
| P2 = 104 | P9 = 108 | P16 = 67 | P23 = 114 | P30 = 114 | P37 = 32 | P44 = 105 | P51 = 115 |
| P3 = 101 | P10 = 116 | P17 = 111 | P24 = 32 | P31 = 107 | P38 = 67 | P45 = 99 |  |
| P4 = 32 | P11 = 121 | P18 = 109 | P25 = 78 | P32 = 115 | P39 = 111 | P46 = 97 |  |
| P5 = 70 | P12 = 32 | P19 = 112 | P26 = 101 | P33 = 32 | P40 = 109 | P47 = 116 |  |
| P6 = 97 | P13 = 111 | P20 = 117 | P27 = 116 | P34 = 97 | P41 = 109 | P48 = 105 |  |
| P7 = 99 | P14 = 102 | P21 = 116 | P28 =119 | P35 = 110 | P42 = 117 | P49 = 111 |  |

*\* Với mã hoá cho bảo mật:*

- Mã hoá các block phía trên: (**Pn)e mod n**

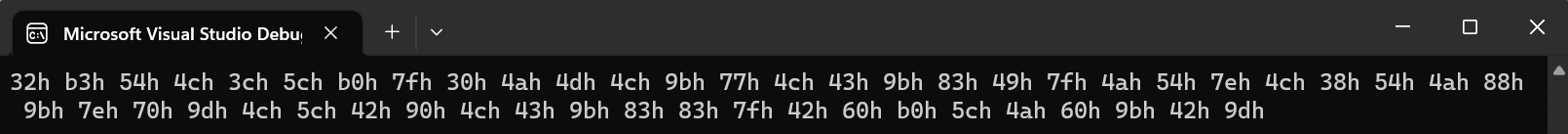
+ Sử dụng hàm dưới để mã hoá và in ra kết quả (vì sử dụng thư viện tính toán các số lớn của CryptoPP nên khi chuyển sang hệ hex, dấu “.” chuyển thành chữ “h”, kết quả Ciphertext cuối bỏ chữ “h”):

**Graphical user interface, text, application

Description automatically generated**

*Hình 13. Hàm tính ax mod p của một mảng các số a sử dụng các hàm có sẵn của CryptoPP*

*a, p1 = 11, q1 = 17, e1 = 7, d1 = 23, n1 = 187*



*Hình 14. Kết quả chạy của ví dụ 1*

→ **Ciphertext** (hex) (bỏ “h” và khoảng trắng “ ”):

32B3544C3C5CB07F304A4D4C9B774C439B83497F4A547E4C38544A889B7E709D4C5C42904C439B83837F4260B05C4A609B429D

*b, p2 = 20079993872842322116151219, q2 = 676717145751736242170789, e2 = 17 (decimal), n2 = 13588476140342208394395166469647627226674348541791, d2 = 7993221259024828467291262184080358019185876599873*

*A screenshot of a computer

Description automatically generated with medium confidence*

*Hình 15. Kết quả chạy của ví dụ 2*

→ **Ciphertext** (hex) (bỏ “h” và khoảng trắng “ ”):



*c, p3 = F7E75FDC469067FFDC4E847C51F452DF,*

*q3 = E85CED54AF57E53E092113E62F436F4F, e3 = 0D88C3 (hexadecimal)*

*Text

Description automatically generated*

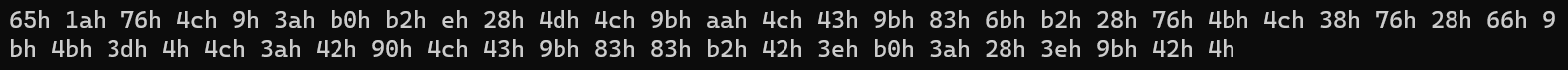
*Hình 16. Kết quả chạy của ví dụ 3*

→ **Ciphertext** (hex) (bỏ “h” và khoảng trắng “ ”):



*\* Với mã hoá cho chứng thực:* Mã hoá các block phía trên: (**Pn)d mod n**

*a, p1 = 11, q1 = 17, e1 = 7, d1 = 23, n1 = 187*

**

*Hình 17. Kết quả chạy của ví dụ 1*

*b, p2 = 20079993872842322116151219, q2 = 676717145751736242170789, e2 = 17 (decimal), n2 = 13588476140342208394395166469647627226674348541791, d2 = 7993221259024828467291262184080358019185876599873*

*Text

Description automatically generated*

*Hình 18. Kết quả chạy của ví dụ 2*

*c, p3 = F7E75FDC469067FFDC4E847C51F452DF,*

*q3 = E85CED54AF57E53E092113E62F436F4F, e3 = 0D88C3 (hexadecimal)*

hay:p3 = 329520679814142392965336341297134588639,

q3 = 308863399973593539130925275387286220623, e3 = 886979,

n3 =

101776877529005912638346811918779931246783058062684819617574643018368103302097

d =

24212225287904763939160097464943268930139828978795606022583874367720623008491

*Text

Description automatically generated*

*Hình 19. Kết quả chạy của ví dụ 3*

**3. RSA Application**

\* Nội dung 1 số hàm trong code (bản full được gửi đính kèm báo cáo):

**Text

Description automatically generated**

*Hình 20. Nội dung hàm tạo 1 tập số nguyên tố prime*

**Text, letter

Description automatically generated**

*Hình 21. Nội dung hàm chọn 1 số từ tập số nguyên tố prime*

**Text

Description automatically generated**

*Hình 22. Nội dung hàm tìm UCLN*

**Text

Description automatically generated**

*Hình 23. Nội dung hàm tính d*

**Text

Description automatically generated**

*Hình 24. Nội dung hàm chọn p, q, e ngẫu nhiên*

**Graphical user interface, text, application, email

Description automatically generated**

*Hình 25. Nội dung hàm nhập p, q, e từ bàn phím*

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

*Hình 26, 27. Nội dung hàm mã hoá/giải mã*

* Tiến hành thử:

**Text

Description automatically generated**

*Hình 28. Nội dung thử với p, q, e ngẫu nhiên*

**Text

Description automatically generated**

*Hình 29. Nội dung thử với p, q, e nhập từ bàn phím*

**4. Programing using the Crypto Library**

Viết ứng dụng mã hoá (sử dụng CryptoPP):

- Hỗ trợ: Thuật toán ECDSA.

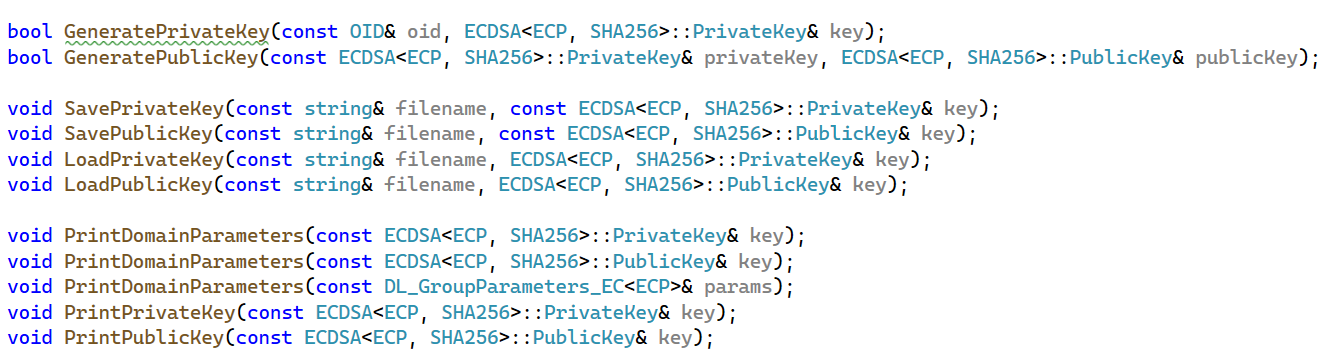
- Hỗ trợ: chức năng ký và chức năng xác minh.

- Đường cong ECC: nên chọn từ các đường cong tiêu chuẩn.

- Tin nhắn để ký/Chữ ký để xác minh: Nhập từ tệp (dùng tên tệp); Hỗ trợ tiếng Việt (dùng setmode, UTF-16) (điểm cộng)

- Khóa bí mật/khóa công khai: Các khoá tải từ tệp (cho cả 2 chức năng), Khóa công khai: >=256 bit

**\* Code hoàn chỉnh được gửi kèm báo cáo:**

****

****

*Hình 30, 31. Tạo khoá, in và lưu khoá dựa trên ECDSA*

****

*Hình 32. Nội dung hàm tính toán và ký*

****

*Hình 33. Nội dung hàm xác minh chữ ký*

**\* Thực hiện chạy code:**

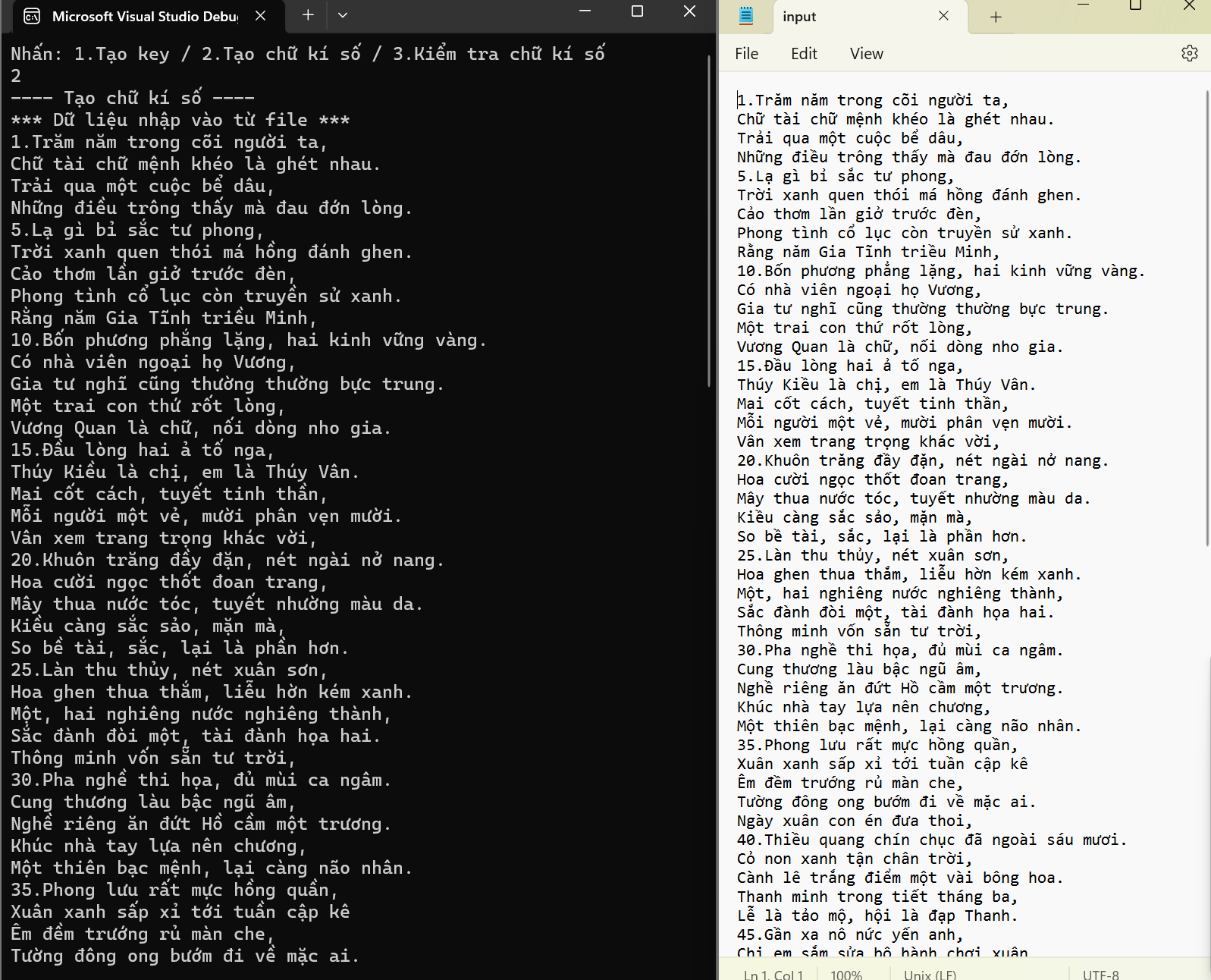
- Tạo key với ECDSA:

**Text

Description automatically generated**

*Hình 34. Tạo key với ECDSA*

- Tiến hành ký với file input.txt:

Text

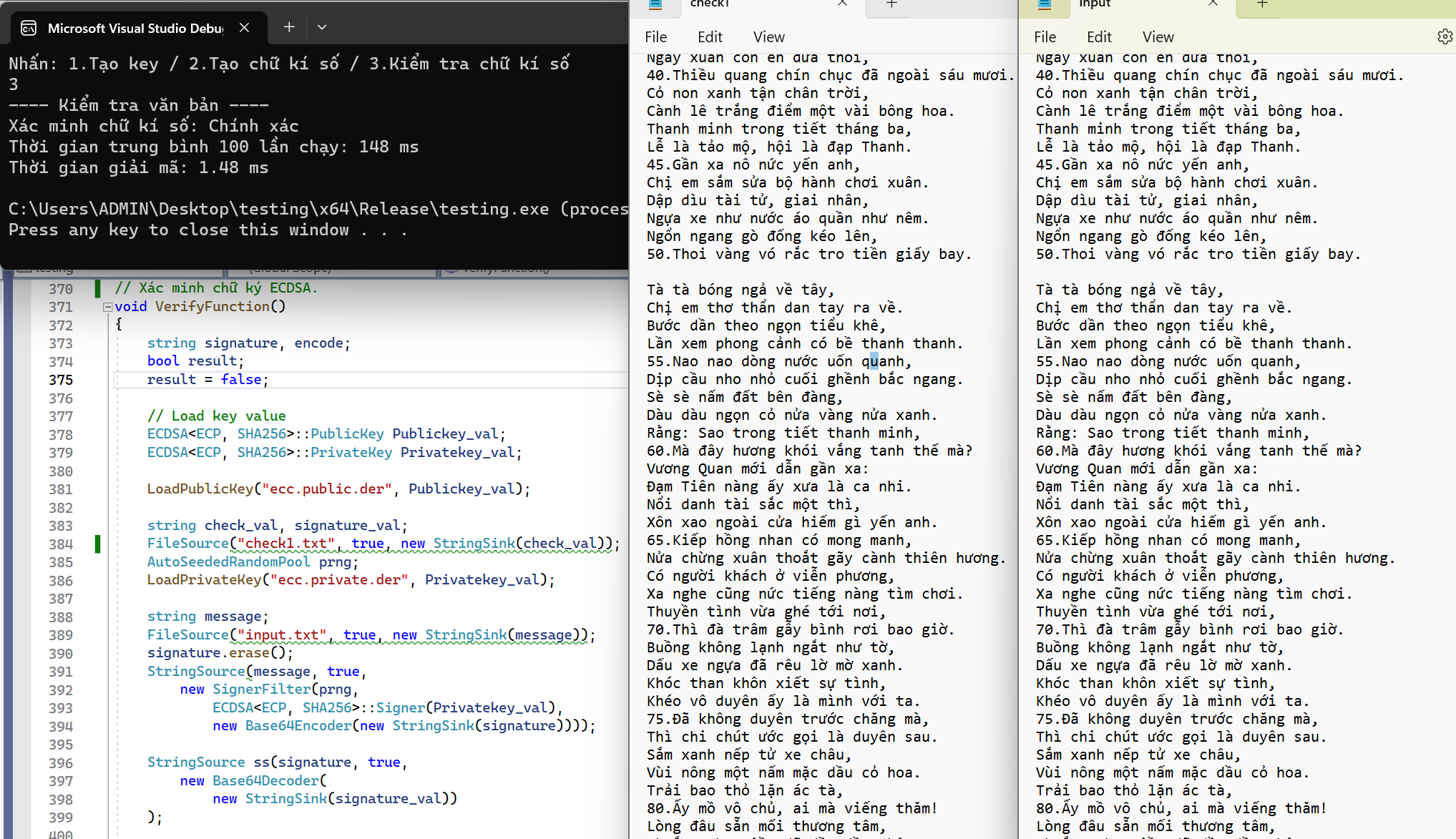
Description automatically generated

Text

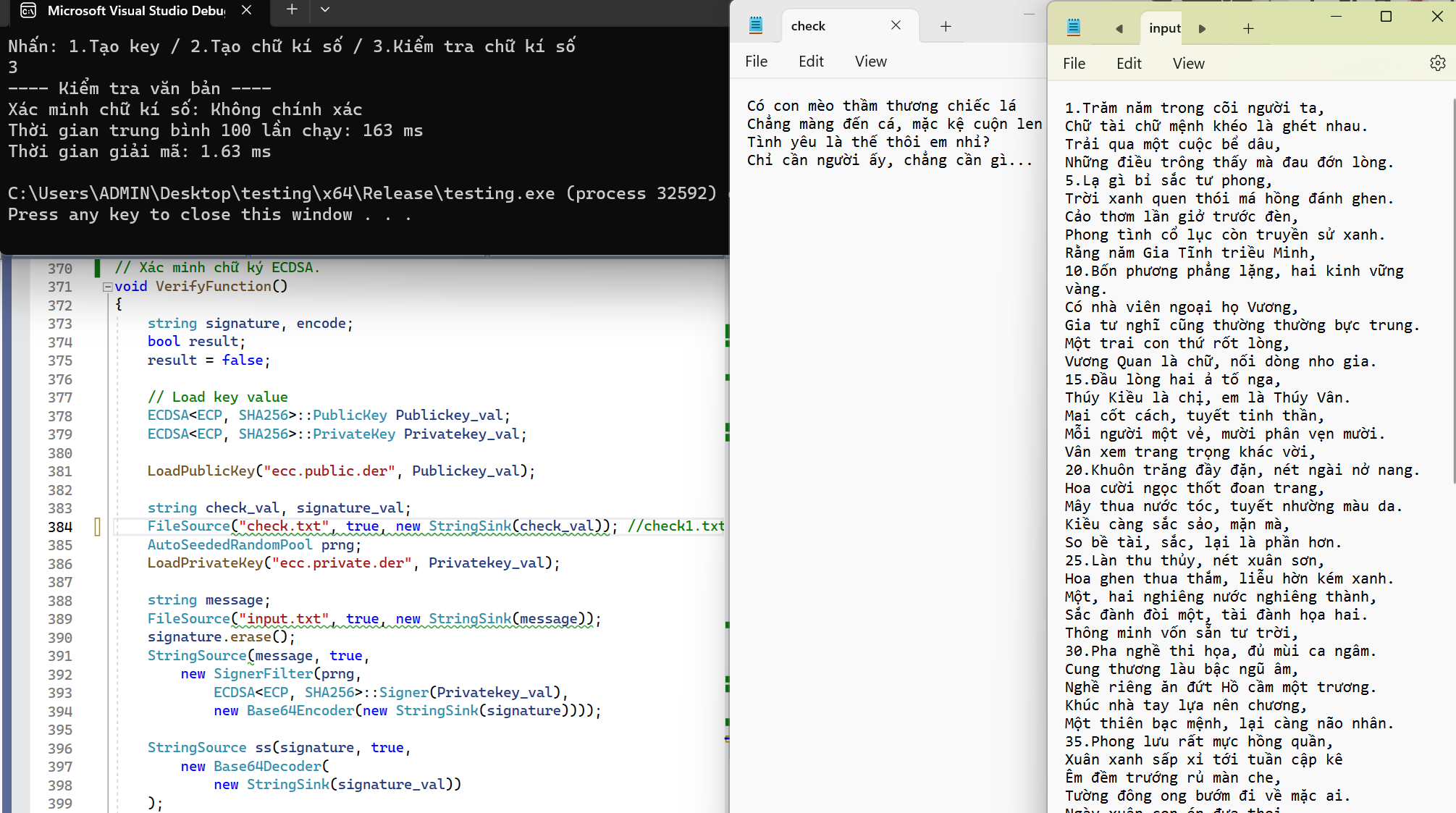
Description automatically generated

*Hình 35, 36, 37. Kết quả ký với file input.txt*

- Tiến hành xác minh chữ ký với file input.txt – check1.txt (nội dung check1.txt giống với input.txt) và input.txt – check.txt (nội dung check.txt khác input.txt):



*Hình 38. Kết quả xác minh chữ ký với file input.txt và file check1.txt*



*Hình 39. Kết quả xác minh chữ ký với file input.txt và file check1.txt*

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)