BÁO CÁO TRAO ĐỔI KHÓA ECDH

**Môn học: ABC**

**Tên chủ đề: ECDH**

*GVLT: Nguyễn Ngọc Tự*

1. **THÔNG TIN CHUNG:**

Lớp: NT219.N21.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email |
|  | Huỳnh Nguyễn Uyển Nhi | 21522424 |  |

**file source code: ECC\_CO1.cpp**

Giai đoạn khởi tạo:

* Chọn loại curve384: Định nghĩa curve qua các thông số nhập vào thủ công gồm p, a, và b. Trong đó p là số nguyên tố.

Tham khảo số liệu: <https://www.secg.org/sec2-v2.pdf>

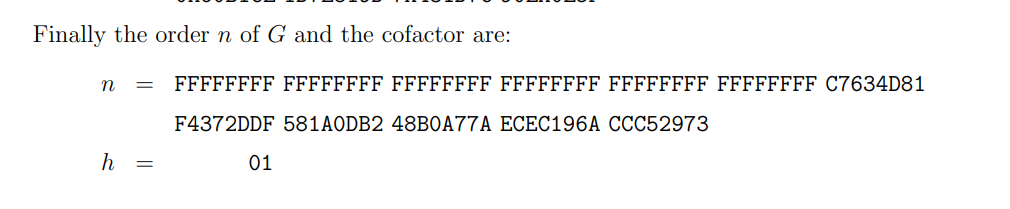
* Đường cong Elliptic được xác định bởi phương trình y^2 = x^3 + a\*x + b trên trường hữu hạn G(p): CryptoPP::ECP eqcurve384(p, a, b);
* Khởi tạo điểm sinh G: 04 biểu diễn cả x,y (384 bits đầu cho x, 384 bits sau cho y)

A picture containing text, font, screenshot, line

Description automatically generated



* N và h có thông tin như sau:



* Tiến hành tạo curve:



***Giai đoạn tính toán privatekey, publickey:***

Bên A:

* Chọn random một số privatekeyA trong khoảng từ 2 đến n-1:

Cụ thể chọn privatekeyA= 11517981818447497085967951605465729846919455156164689798014638985875583987829406262183357738592749044441967993136875

* Sau đó tính publickeyA bằng cách thực hiên phép cộng trên ECC: cộng n lần điểm sinh G (với n = privatekeyA) để tạo publickeyA.



Bên B:

Thực hiện tương tự như trên

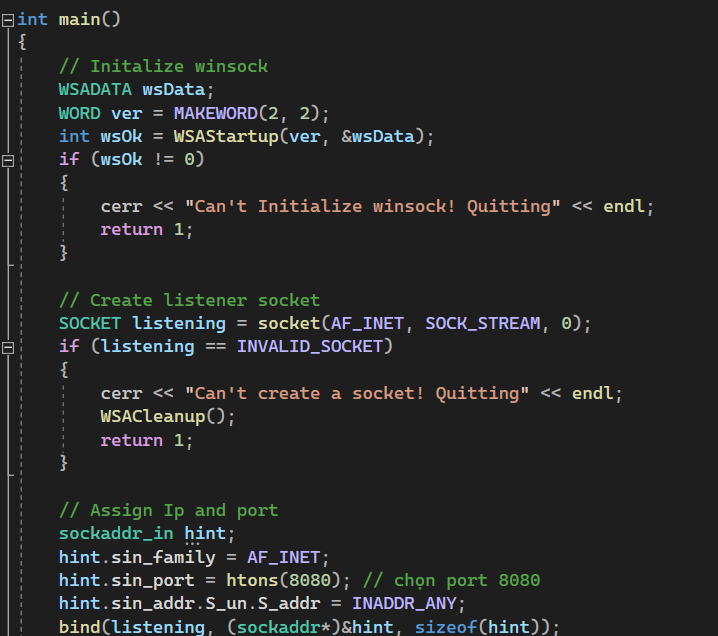
***Giai đoạn trao đổi khóa công khai:***

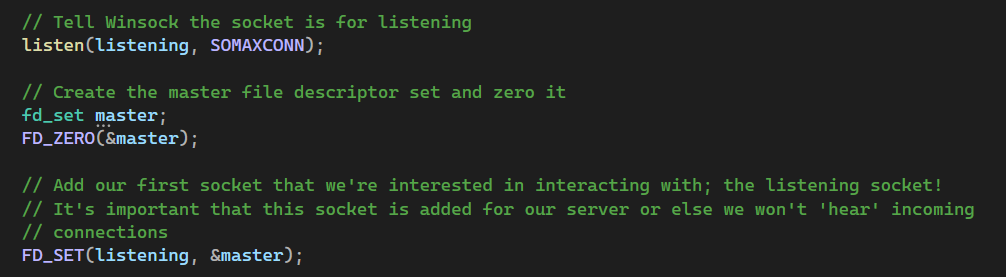
* Chọn partner:

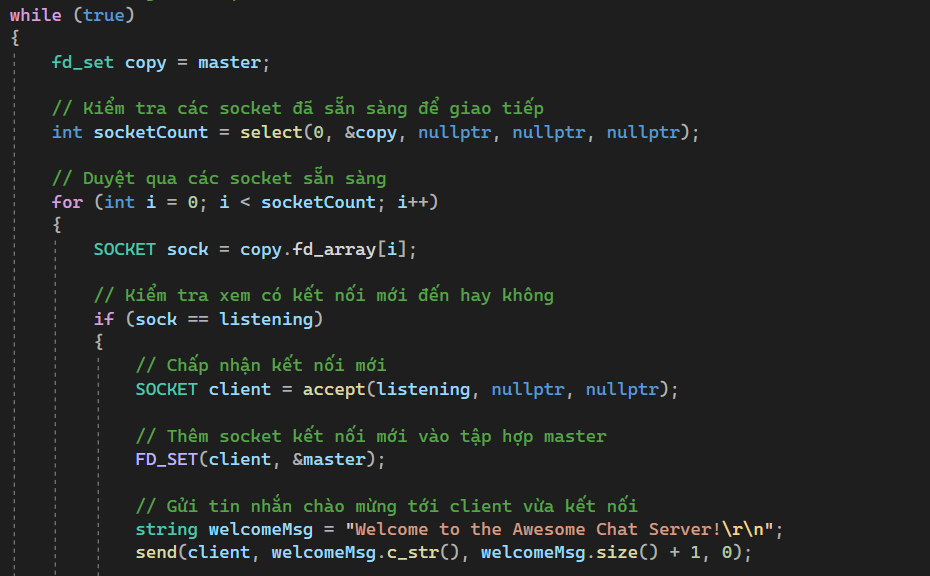
MSSV: 21520155

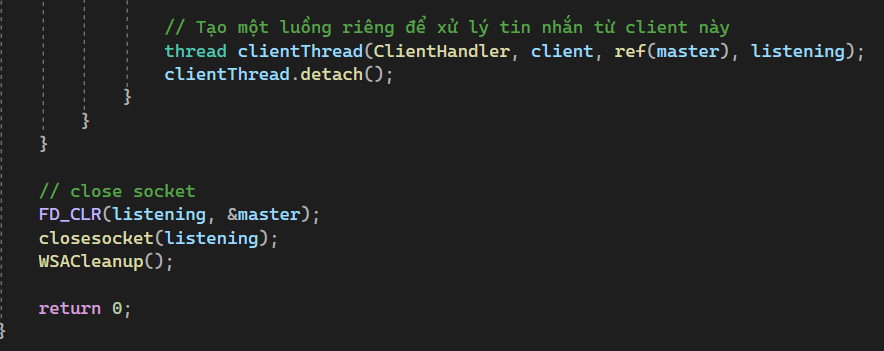
Họ tên: Nguyễn Triệu Thiên Bảo (B)

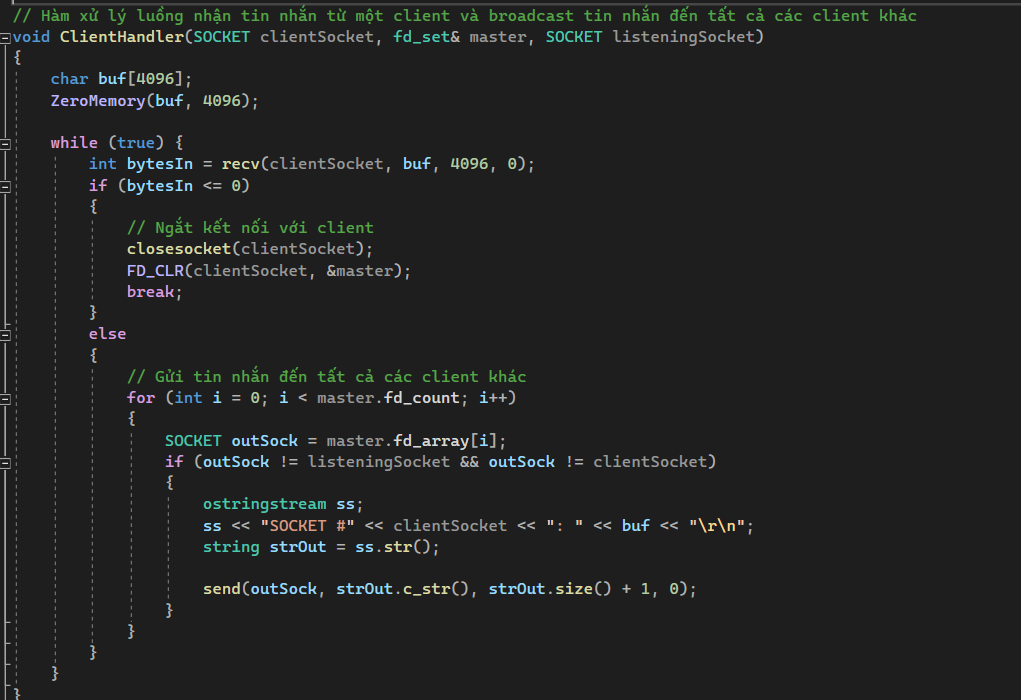
* Gửi public key qua tcp socket(vì tính reliability của TCP): Chọn triển khai trao đổi khóa công khai qua windows form C#
* TCP Server:



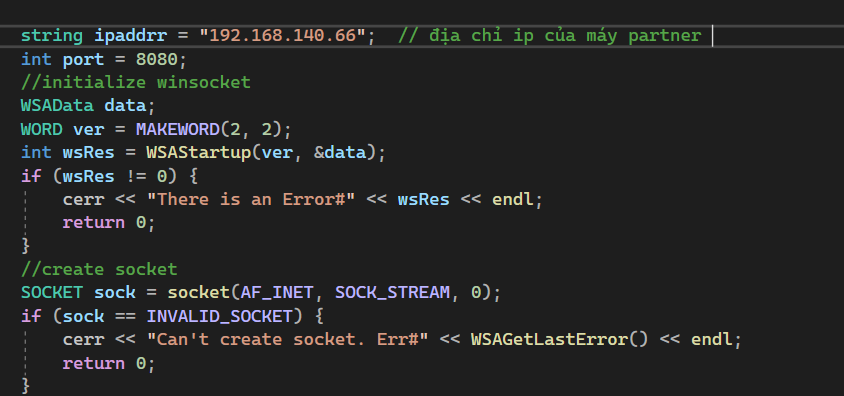


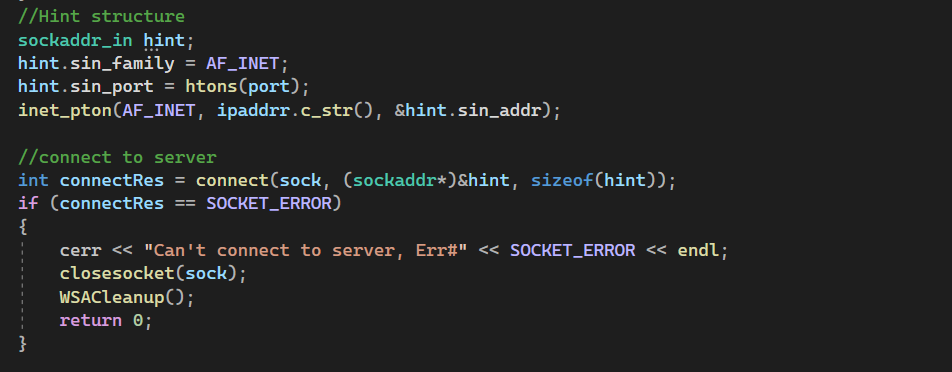




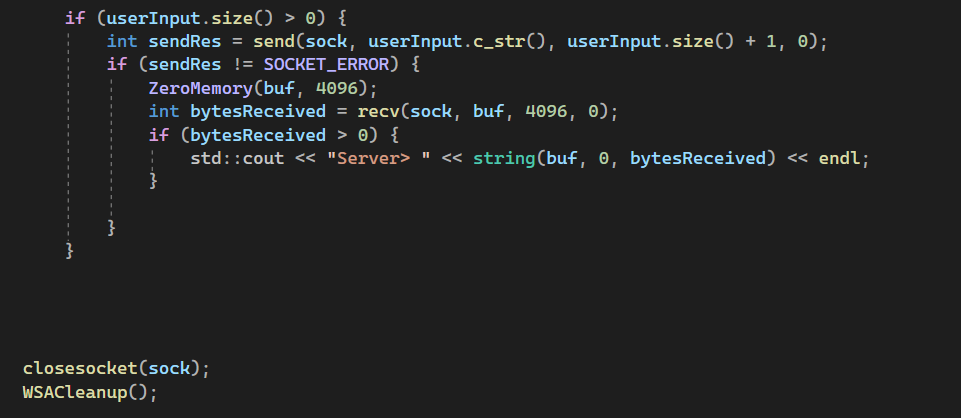


* TCP Client:









A picture containing text, screenshot, font

Description automatically generated

Fig : Giao diện khi trao đổi publickey

***Giai đoạn thỏa thuận khóa phiên:***

* Khóa phiên:

Bên A: tính bằng cách cộng publickeyB của B privatekeyA lần, tương đương:

publickeyB \* privatekeyA = G \* privatekeyB \* privatekeyA

Bên B: tính bằng cách cộng publickeyA của A privatekeyB lần, tương đương:

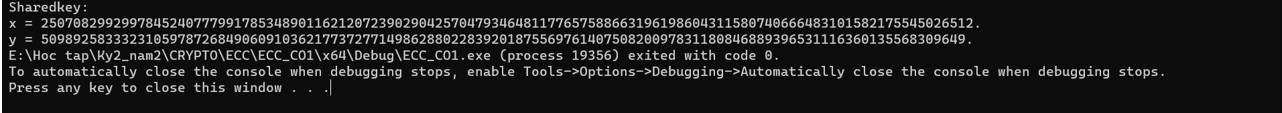
publickeyA \* privatekeyB = G \* privatekeyA \* privatekeyB

Mà privatekeyA\*privatekeyB = privatekeyB\*privatekeyA

Suy ra A và B cùng có chung một khóa phiên.

A screen shot of a computer

Description automatically generated with medium confidence



**file source code: ECC\_CO2.cpp. Đây là curve tương tự nên em chỉ thực hiện các tính toán trên curve**

Dùng curve chuẩn: ASN1::secp384r1()

A picture containing text, font, screenshot

Description automatically generated

***Giai đoạn tính toán privatekey, publickey:***

Bên A:

* Chọn random một số privatekeyA trong khoảng từ 2 đến n-1:

VD: privatekeyA= 28532526134272672038090746974273651943328603529713681629972460160474972683944309600686292867735723112090813839265681

* Sau đó tính publickeyA bằng cách thực hiên phép cộng trên ECC: cộng n lần điểm sinh G (với n = privatekeyA) để tạo publickeyA.



Bên B:

Thực hiện tương tự như trên.

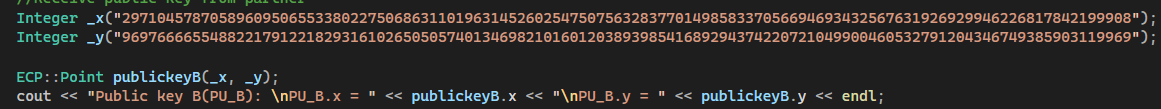
***Giai đoạn trao đổi khóa công khai:***

* Chọn partner:

MSSV: 21520155

Họ tên: Nguyễn Triệu Thiên Bảo (B)

* Gửi publickey qua tin nhắn.



***Giai đoạn tính khóa phiên:***

* Khóa phiên:

Bên A: tính bằng cách cộng publickeyB của B privatekeyA lần, tương đương:

publickeyB \* privatekeyA = G \* privatekeyB \* privatekeyA

Bên B: tính bằng cách cộng publickey của A nB lần, tương đương:

publickeyA \* privatekeyB = G \* privatekeyA \* privatekeyB

Mà privatekeyA \* privatekeyB = privatekeyB \* privatekeyA

Suy ra A và B cùng có chung một khóa phiên

A screen shot of a computer

Description automatically generated with medium confidence

Kết quả:

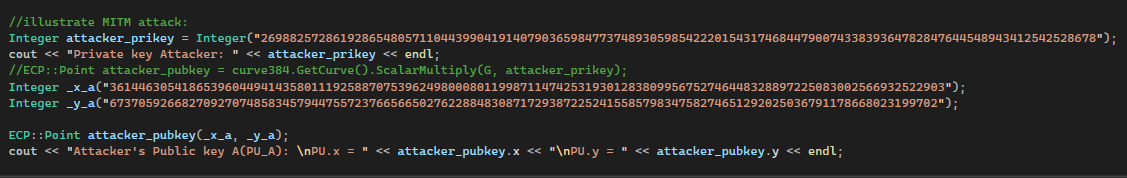
A picture containing text, screenshot, font, document

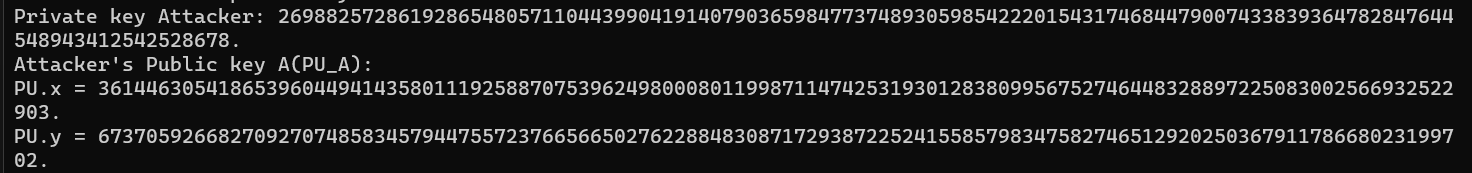
Description automatically generated

**Bonus point:**

Mô phỏng MITM:

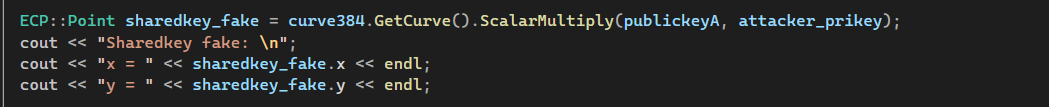
Tạo thêm khóa private key và public key cho attacker như sau:

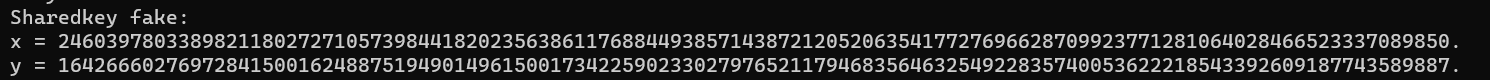




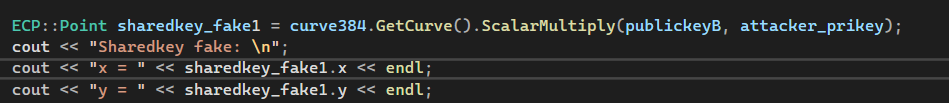
Tiếp đến, em và partner thực hiện trao đổi khóa như bình thường nhưng mà bị attacker bắt được và tính lại publickey mới (giả mạo). Từ đó tiến hành thỏa thuận khóa phiên như bên dưới.

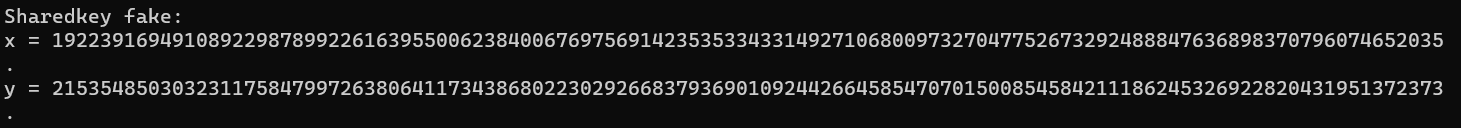
* Attacker vs bên A:



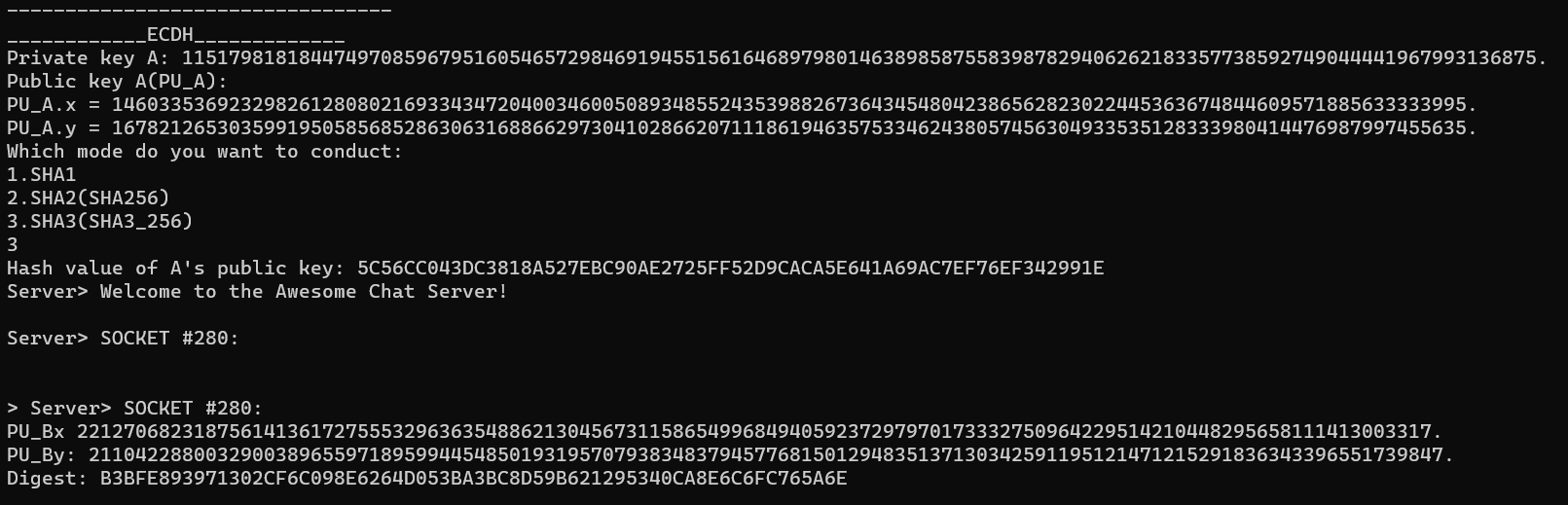


* Attacker vs bên B:



****

* Hacker sẽ dùng sharedkey\_fake để giao tiếp với A và sharedkey\_fake1 để giao tiếp với B mà A, B không biết được.
* Tuy nhiên để đảm bảo key không bị chỉnh sửa (vd: chống lại Man In the Middle attack) 🡪 **Ý tưởng**: Dùng hash function để check và public mã hash để cho partner có thể verify khi nhận publickey. Mỗi bên phải công khai hash func và mã hash.
* Chuyển từ Point sang string bằng cách nối hai tọa độ x, y lại sau đó tính hash của chuỗi string đó.
* Em chọn hàm sha3\_256 để tính hash và trao đổi key+digest như sau:



* Em biết được bạn partner dùng hàm hash sha256 và có mã hash = B3BFE893971302CF6C098E6264D053BA3BC8D59B621295340CA8E6C6FC765A6E



