

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY
FALCUTY OF COMPUTER NETWORKS AND COMMUNICATIONS



ASSIGNMENT

ELLIPTIC CURVE CRYPTOGRAPHY

ECDH

Instructor: Ph.D.Nguyễn Ngọc Tự

Course: NT219.N22.ATCL

Group members: Huỳnh Đình Khải Minh - 21521123 - 21521123@gm.uit.edu.vn
Trần Thành Lợi - 21522296 - 21522296@gm.uit.edu.vn
Nguyễn Nguyễn Duy An - 21520546 - 21520546@gm.uit.edu.vn





- Tài nguyên: Demo code trên MS TEAM.
- Thư viện: CryptoPP .
- Ngôn ngữ: C++.

- $a = 39402006196394479212279040100143613805079739270465446667948293404245721771496870329047266088258938001861606973112316$.
- $b = 27580193559959705877849011840389048093056905856361568521428707301988689241309860865136260764883745107765439761230575$.
- $G_x = 26247035095799689268623156744566981891852923491109213387815615900925518854738050089022388053975719786650872476732087$.
- $G_y = 8325710961489029985546751289520108179287853048861315594709205902480503199884419224438643760392947333078086511627871$.

Listing 1: ECDHmanualparamater.cpp

Page 1/3

```

30 // CryptoPP::DL_GroupParameters_EC<ECP> curve256(eqcurve256,G,n,h);
31 CryptoPP::DL_GroupParameters_EC<ECP> curve384;
32 curve384.Initialize(eqcurve384,G,n,h);
33
34 //Create PublicKey
35 ECP::Point PublicKeyA = curve384.GetCurve().ScalarMultiply(G,
    privateKeyA);
36 ECP::Point PublicKeyB = curve384.GetCurve().ScalarMultiply(G,
    privateKeyB);
37
38 cout << "PublicKeyAx= " << PublicKeyA.x << endl;
39 cout << "PublicKeyAy= " << PublicKeyA.y << endl;
40 cout << "PublicKeyBx= " << PublicKeyB.x << endl;
41 cout << "PublicKeyBy= " << PublicKeyB.y << endl;
42 //Create ShareKey ECDH
43 //ShareKey A side
44 ECP::Point ShareKeyA = curve384.GetCurve().Multiply(privateKeyA,
    PublicKeyB);
45 cout << "ShareKeyAx= " << ShareKeyA.x << endl;
46 cout << "ShareKeyAy= " << ShareKeyA.y << endl;
47 //ShareKey B side
48 ECP::Point ShareKeyB = curve384.GetCurve().Multiply(privateKeyB,
    PublicKeyA);
49 cout << "ShareKeyBx= " << ShareKeyB.x << endl;
50 cout << "ShareKeyBy= " << ShareKeyB.y << endl;
51 if(ShareKeyA == ShareKeyB)
52 {
53     cout<< "ShareKey hop le" << endl;
54 }
55 else
56 {
57     cout<< "ShareKey khong hop le" << endl;
58 }
59 }

```

Listing 2: ECDHstandardcurve.cpp

```

1 int main(int argc, char* argv[])
2 {
3     Integer privateKeyA("16");
4     Integer privateKeyB("32");
5     cout << "PrivateKeyA=" << privateKeyA << endl;
6     cout << "PrivateKeyB=" << privateKeyB << endl;
7     CryptoPP::OID oid=ASN1::secp384r1();
8     /* Create curve */
9     CryptoPP::DL_GroupParameters_EC<ECP> curve384;
10    curve384.Initialize(oid);
11    ECP::Point G=curve384.GetSubgroupGenerator();
12    ECP::Point PublicKeyA = curve384.GetCurve().ScalarMultiply(G,
        privateKeyA);
13    ECP::Point PublicKeyB = curve384.GetCurve().ScalarMultiply(G,
        privateKeyB);
14    cout << "PublicKeyBx=" << PublicKeyB.x << endl;
15    cout << "PublicKeyBy=" << PublicKeyB.y << endl;
16    cout << "PublicKeyAx=" << PublicKeyA.x << endl;
17    cout << "PublicKeyAy=" << PublicKeyA.y << endl;
18    cout << "PublicKeyBx=" << PublicKeyB.x << endl;
19    cout << "PublicKeyBy=" << PublicKeyB.y << endl;
20    //Create ShareKey ECDH
21    //ShareKey A side

```



ASSIGNMENT

```
22      ECP::Point ShareKeyA = curve384.GetCurve().Multiply(privateKeyA,
23                  PublicKeyB);
24      cout << "ShareKeyAx=" << ShareKeyA.x << endl;
25      cout << "ShareKeyAy=" << ShareKeyA.y << endl;
26      //ShareKey B side
27      ECP::Point ShareKeyB = curve384.GetCurve().Multiply(privateKeyB,
28                  PublicKeyA);
29      cout << "ShareKeyBx=" << ShareKeyB.x << endl;
30      cout << "ShareKeyBy=" << ShareKeyB.y << endl;
31  }
```

- Ta cho PrivateKey bên A là 16(Pa), bên B là 32(Pb). Đây là 2 số nằm trong khoảng $[1, p-1]$ nên thỏa mãn điều kiện.

- Từ privateKey ta thực hiện tạo Public Key bằng cách thực hiện phép nhân vô hướng giá trị PrivateKey với điểm G:

$$Qa = Pa * G$$

$$Qb = Pb * G$$

- Sau khi đã có Public Key của 2 bên ta sẽ tính Share Key bằng cách nhân 2 Private Key của A với Public Key của B và ngược lại, ta có:

$$\text{ShareA} = Pa * Qb$$

$$\text{ShareB} = Pb * Qa$$

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bedep\Desktop\CryptoPP MMH\cpptest> .\ECDH1.exe
PrivateKeyA=16.
PrivateKeyB=32.
PublicKeyAx= 32913932395616264895566414122765335444396157997049244607413007480094180059681244402128522915364132398797797
514391503.
PublicKeyAy= 33258149791966603262610080790276393461897267120335577178701134853798651660398480161316647903923629160436268
769425011.
PublicKeyBx= 19616433895130952913470644187706189275663672111625552851652659937224764286710743264709057308519168866467804
289199830.
PublicKeyBy= 3447897976581651812036185081490033478792105822460800047229805158637179175329583918997466925002252854250095
585101691.
ShareKeyAx= 262056738547320257649591320978875426944434172970684849520910558656633571870273456496279054742129800639538740
60926539.
ShareKeyAy= 187070631023868801326919143665224670558304286777862409548999626935623859666064222501992500748472078481250615
95418935.
ShareKeyBx= 262056738547320257649591320978875426944434172970684849520910558656633571870273456496279054742129800639538740
60926539.
ShareKeyBy= 187070631023868801326919143665224670558304286777862409548999626935623859666064222501992500748472078481250615
95418935.
ShareKey hop le
PS C:\Users\bedep\Desktop\CryptoPP MMH\cpptest> |
```

Figure 1: Chạy code và cho ra kết quả

Link drive code: [Link code ECDH](#)

END.