BÁO CÁO BÀI TẬP

**Môn học: NT209.N22.ATCL**

**Tên chủ đề: ECDH Trao đổi khóa**

**THÔNG TIN CHUNG:**

Lớp: NT209.N22.ATCL

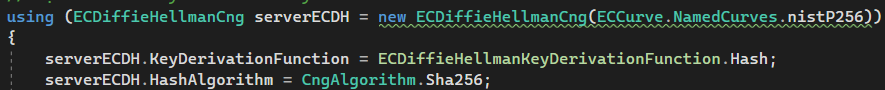
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Phan Thị Hồng Nhung | 21521250 | 21521250@gm.uit.edu.vn |
| 2 | Đoàn Hải Đăng | 21520679 | 21520679@gm.uit.edu.vn |

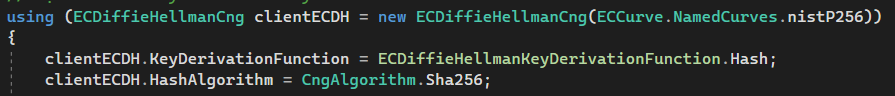
**BÁO CÁO CHI TIẾT**

Ý tưởng: Trao đổi khóa ECDH bằng ngôn ngữ C# sử dụng TCP

1. Đầu tiên Server chạy rồi lắng nghe kết nối, Client kết nối Server
2. Khi kết nối thành công, Server tạo khóa riêng tư và khóa công khai dựa trên Curve set sẵn, gửi khóa công khai cho Client.
3. Client nhận khóa công khai. Client tạo cặp public key, private key dựa trên Curve set sẵn và tính toán shared secret key. Client gửi public key cho Server.
4. Tương tự, Server tính shared secret key bằng khóa riêng tư của Server và public key của Client.
5. Shared secret key giống nhau và hoàn thành trao đổi khóa.

**Câu 1:**





Client – server sử dụng curve set sẵn (nistP256) để tạo cặp key.

**Câu 2:**

A picture containing text, screenshot, software

Description automatically generated

Server tạo cặp khóa, sau đó gửi public key đến client thông qua kênh truyền TCP

A picture containing text, screenshot, software

Description automatically generated

Client cũng tạo cặp khóa và gửi public key đến server

**Câu 3:**

A picture containing text, screenshot, font, line

Description automatically generated

Server nhận được public key của client, sau đó tính toán khóa phiên

A picture containing screenshot, text, font, line

Description automatically generated

Client cũng nhận được public key của server, sau đó tính toán khóa phiên

**Chạy thực nghiệm:**

A picture containing text, screenshot, font

Description automatically generated

Server đang lắng nghe khi chưa có client kết nối

A black screen with white text

Description automatically generated with low confidence

Kết quả phía server sau khi có client kết nối

A screenshot of a computer

Description automatically generated with medium confidence

Kết quả phía client

Code hoàn chỉnh:

**Phía server:**

using System;

using System.Net;

using System.Net.Sockets;

using System.Security.Cryptography;

using System.Text;

class Server

{

static void Main(string[] args)

{

try

{

// Lắng nghe kết nối đến port 1234

TcpListener listener = new TcpListener(IPAddress.Any, 1234);

listener.Start();

Console.WriteLine("Waiting for client connection...");

// Chấp nhận kết nối từ client

TcpClient client = listener.AcceptTcpClient();

Console.WriteLine("Client connected!");

// Tạo Private Key và Public Key

using (ECDiffieHellmanCng serverECDH = new ECDiffieHellmanCng(ECCurve.NamedCurves.nistP256))

{

serverECDH.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

serverECDH.HashAlgorithm = CngAlgorithm.Sha256;

// Lấy Public Key của server

byte[] serverPublicKey = serverECDH.PublicKey.ToByteArray();

// Gửi kích thước Public Key đến client

byte[] sizeBytes = BitConverter.GetBytes(serverPublicKey.Length);

client.GetStream().Write(sizeBytes, 0, sizeBytes.Length);

// Gửi Public Key đến client

client.GetStream().Write(serverPublicKey, 0, serverPublicKey.Length);

// Đọc kích thước Public Key từ client

byte[] clientPublicKeySizeBytes = new byte[4];

client.GetStream().Read(clientPublicKeySizeBytes, 0, 4);

int clientPublicKeySize = BitConverter.ToInt32(clientPublicKeySizeBytes, 0);

// Đọc Public Key từ client

byte[] clientPublicKeyBytes = new byte[clientPublicKeySize];

client.GetStream().Read(clientPublicKeyBytes, 0, clientPublicKeySize);

// Tính toán khóa chung

CngKey clientPublicKey = CngKey.Import(clientPublicKeyBytes, CngKeyBlobFormat.EccPublicBlob);

byte[] sharedSecret = serverECDH.DeriveKeyMaterial(clientPublicKey);

Console.WriteLine("Shared secret key: " + BitConverter.ToString(sharedSecret).Replace("-", ""));

}

client.Close();

listener.Stop();

}

catch (Exception ex)

{

Console.WriteLine("Error: " + ex.Message);

}

Console.ReadLine();

}

}

**Phía Client:**

using System;

using System.Net;

using System.Net.Sockets;

using System.Security.Cryptography;

class Client

{

static void Main(string[] args)

{

try

{

// Kết nối tới địa chỉ IP và cổng của server

TcpClient client = new TcpClient();

client.Connect(IPAddress.Parse("127.0.0.1"), 1234);

Console.WriteLine("Connected to server!");

// Tạo Private Key và Public Key

using (ECDiffieHellmanCng clientECDH = new ECDiffieHellmanCng(ECCurve.NamedCurves.nistP256))

{

clientECDH.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

clientECDH.HashAlgorithm = CngAlgorithm.Sha256;

// Lấy Public Key của client

byte[] clientPublicKey = clientECDH.PublicKey.ToByteArray();

// Gửi kích thước Public Key đến server

byte[] sizeBytes = BitConverter.GetBytes(clientPublicKey.Length);

client.GetStream().Write(sizeBytes, 0, sizeBytes.Length);

// Gửi Public Key đến server

client.GetStream().Write(clientPublicKey, 0, clientPublicKey.Length);

// Đọc kích thước Public Key từ server

byte[] serverPublicKeySizeBytes = new byte[4];

client.GetStream().Read(serverPublicKeySizeBytes, 0, 4);

int serverPublicKeySize = BitConverter.ToInt32(serverPublicKeySizeBytes, 0);

// Đọc Public Key từ server

byte[] serverPublicKeyBytes = new byte[serverPublicKeySize];

client.GetStream().Read(serverPublicKeyBytes, 0, serverPublicKeySize);

// Tính toán khóa chung

CngKey serverPublicKey = CngKey.Import(serverPublicKeyBytes, CngKeyBlobFormat.EccPublicBlob);

byte[] sharedSecret = clientECDH.DeriveKeyMaterial(serverPublicKey);

Console.WriteLine("Shared secret key: " + BitConverter.ToString(sharedSecret).Replace("-", ""));

}

client.Close();

}

catch (Exception ex)

{

Console.WriteLine("Error: " + ex.Message);

}

Console.ReadLine();

}

}