Nguyễn Văn Hưng – 23520569

using System;

using System.IO;

using System.Runtime.InteropServices;

class AESInterop

{

    [DllImport("kernel32", SetLastError = true)]

    private static extern bool SetDllDirectory(string lpPathName);

    const int AES\_KEY\_SIZE = 16;

    const int AES\_IV\_SIZE = 16;

    [DllImport(@"D:\CryptoLibrary\AES.dll", CallingConvention = CallingConvention.Cdecl)]

    public static extern void GenerateAESKeyIV(byte[] key, byte[] iv);

    [DllImport(@"D:\CryptoLibrary\AES.dll", CallingConvention = CallingConvention.Cdecl)]

    public static extern void SaveKeyToFile(string filename, byte[] key, byte[] iv);

    [DllImport(@"D:\CryptoLibrary\AES.dll", CallingConvention = CallingConvention.Cdecl)]

    public static extern void LoadKeyFromFile(string filename, byte[] key, byte[] iv);

    [DllImport(@"D:\CryptoLibrary\AES.dll", CallingConvention = CallingConvention.Cdecl)]

    public static extern void AESEncryptFile(byte[] key, byte[] iv, string infile, string outfile);

    [DllImport(@"D:\CryptoLibrary\AES.dll", CallingConvention = CallingConvention.Cdecl)]

    public static extern void AESDecryptFile(byte[] key, byte[] iv, string infile, string outfile);

    static void Main(string[] args)

    {

        SetDllDirectory(@"C:\\msys64\\mingw64\\bin");

        if (args.Length == 0)

        {

            Console.WriteLine("Usage:");

            Console.WriteLine("  generate         -> Generate and save AES key/IV");

            Console.WriteLine("  show             -> Show key/IV from saved file");

            Console.WriteLine("  encrypt <in> <out> -> Encrypt file");

            Console.WriteLine("  decrypt <in> <out> -> Decrypt file");

            return;

        }

        string command = args[0].ToLower();

        byte[] key = new byte[AES\_KEY\_SIZE];

        byte[] iv = new byte[AES\_IV\_SIZE];

        switch (command)

        {

            case "generate":

                if (args.Length != 2)

                {

                    Console.WriteLine("Wrong arguments: generate <keyfile>");

                    break;

                }

                GenerateAESKeyIV(key, iv);

                SaveKeyToFile(args[1].ToString(), key, iv);

                Console.WriteLine("Generated and saved AES key/IV to keyfile.bin.");

                break;

            case "show":

                if (args.Length != 2)

                {

                    Console.WriteLine("Wrong arguments: show <key>");

                    break;

                }

                if (!File.Exists(args[1].ToString()))

                {

                    Console.WriteLine("Key file not found.");

                    break;

                }

                LoadKeyFromFile(args[1].ToString(), key, iv);

                Console.WriteLine("Key: " + BitConverter.ToString(key).Replace("-", ""));

                Console.WriteLine("IV : " + BitConverter.ToString(iv).Replace("-", ""));

                break;

            case "encrypt":

                if (args.Length != 4)

                {

                    Console.WriteLine("Wrong arguments: encrypt <key> <input> <output>");

                    break;

                }

                LoadKeyFromFile(args[1].ToString(), key, iv);

                AESEncryptFile(key, iv, args[2], args[3]);

                Console.WriteLine("File encrypted.");

                break;

            case "decrypt":

                if (args.Length != 4)

                {

                    Console.WriteLine("Missing arguments: decrypt <input> <output>");

                    break;

                }

                LoadKeyFromFile(args[1].ToString(), key, iv);

                AESDecryptFile(key, iv, args[2], args[3]);

                Console.WriteLine("File decrypted.");

                break;

            default:

                Console.WriteLine("Unknown command.");

                break;

        }

    }

}

A screenshot of a computer screen

AI-generated content may be incorrect.