Carnegie Mellon University
Africa

18-631: INTRODUCTION TO INFORMATION SECURITY
HOMEWORK 2 – FALL 2024

SIMPLIFIED TLS HANDSHAKE SIMULATION

Names: Lucie Niyomutoni

AndrewID: lniyomut

Programming Language used: **Python**

INTRODUCTION

This report describes a TLS Handshake Simulation program that demonstrates the process of establishing a secure connection using TLS (Transport Layer Security). The simulation includes selecting TLS version and cipher suite, negotiating supported version with the server, and performing a key exchange using the Diffie-Hellman algorithm. Additionally, the program provides a recommendation to switch to the latest supported TLS version if the selected version is outdated.

DESCRIPTION OF THE APPROACH

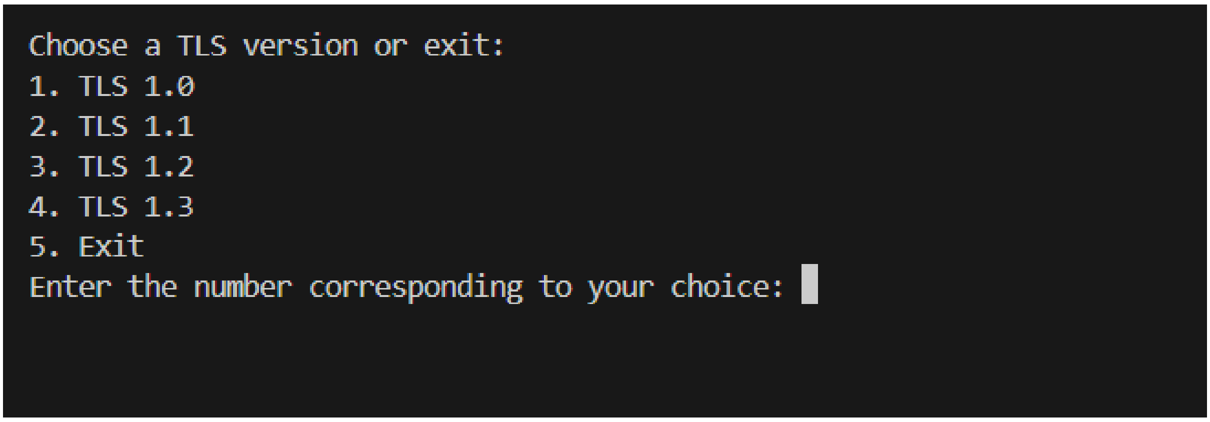
1. **TLS Version Negotiation**: The program allows the user to select a TLS version from a list. It checks if the selected version is supported by the server. If the version is not supported, the user is informed, and the program exits. If it is supported, the server negotiates that version.

   - **Recommendation to Switch to Latest TLS Version**:
     If the **selected** TLS version is **supported** but **not** the **latest** version, the program **suggests switching** to the **latest** supported TLS version.
     The user is given the choice to switch to this latest version or continue with their selected version.

2. **Cipher Suite Selection**: After selecting a TLS version, the user chooses a cipher suite from a list. The program checks if the chosen cipher suite is supported by the server. If it is supported, the server negotiates the cipher suite; otherwise, the program exits.
3. **Diffie-Hellman Key Exchange**: Once the TLS version and cipher suite are agreed upon, the program simulates the Diffie-Hellman key exchange to generate a shared secret key.

MENUS

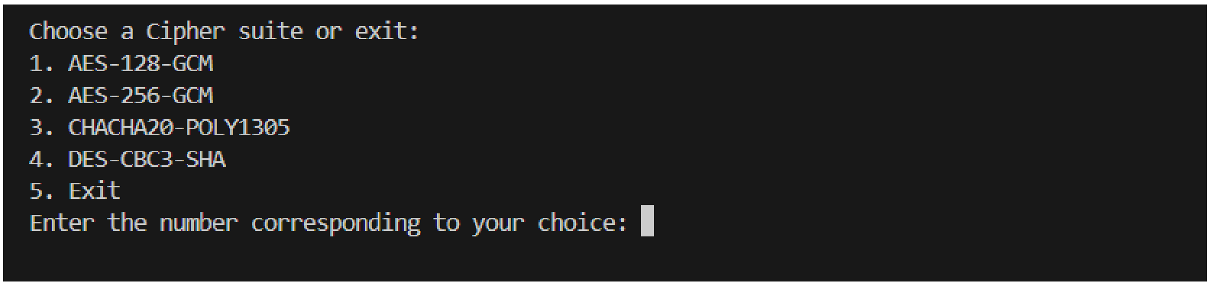
TLS VERSIONS MENU



```
Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice:
```

Figure 1 TLS Versions menu

CIPHER SUITES MENU

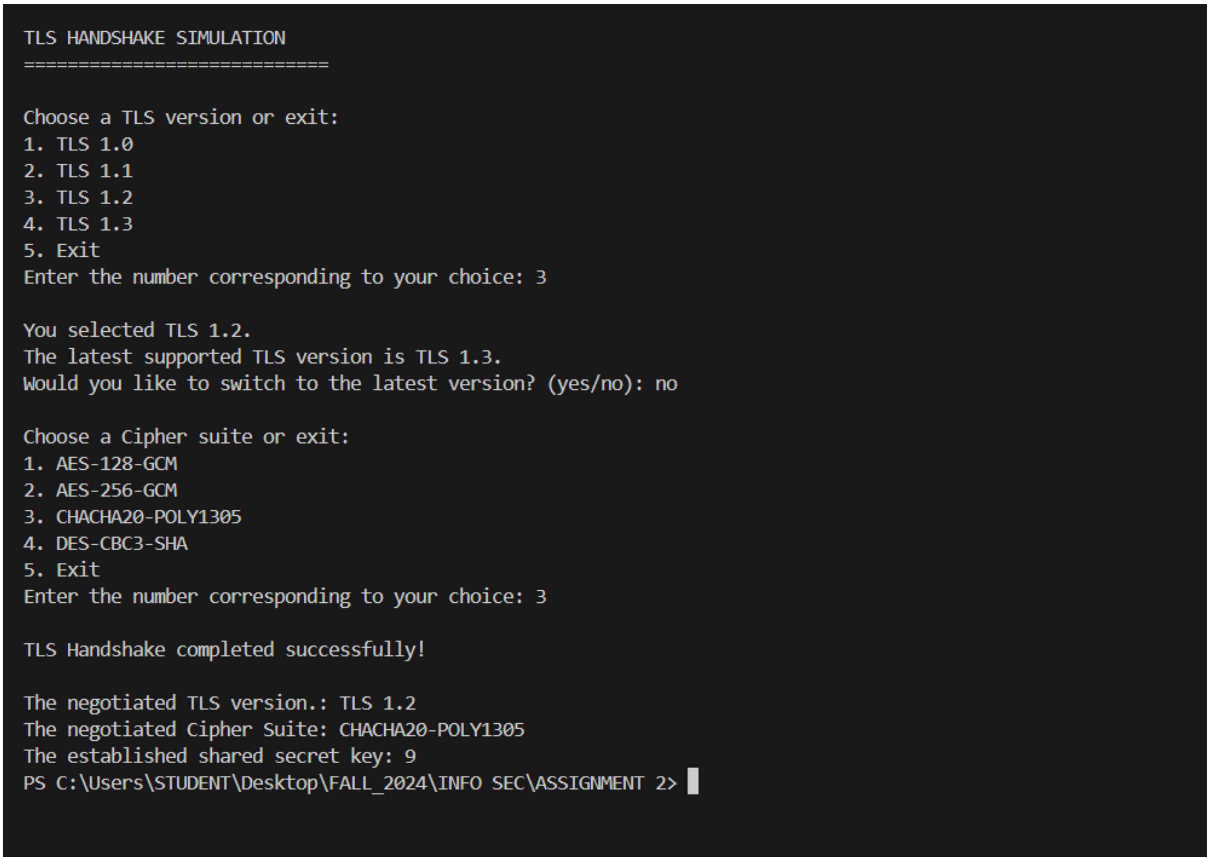


```
Choose a Cipher suite or exit:
1. AES-128-GCM
2. AES-256-GCM
3. CHACHA20-POLY1305
4. DES-CBC3-SHA
5. Exit
Enter the number corresponding to your choice:
```

Figure 2 Cipher Suites Menu

<u>RESULTS</u>

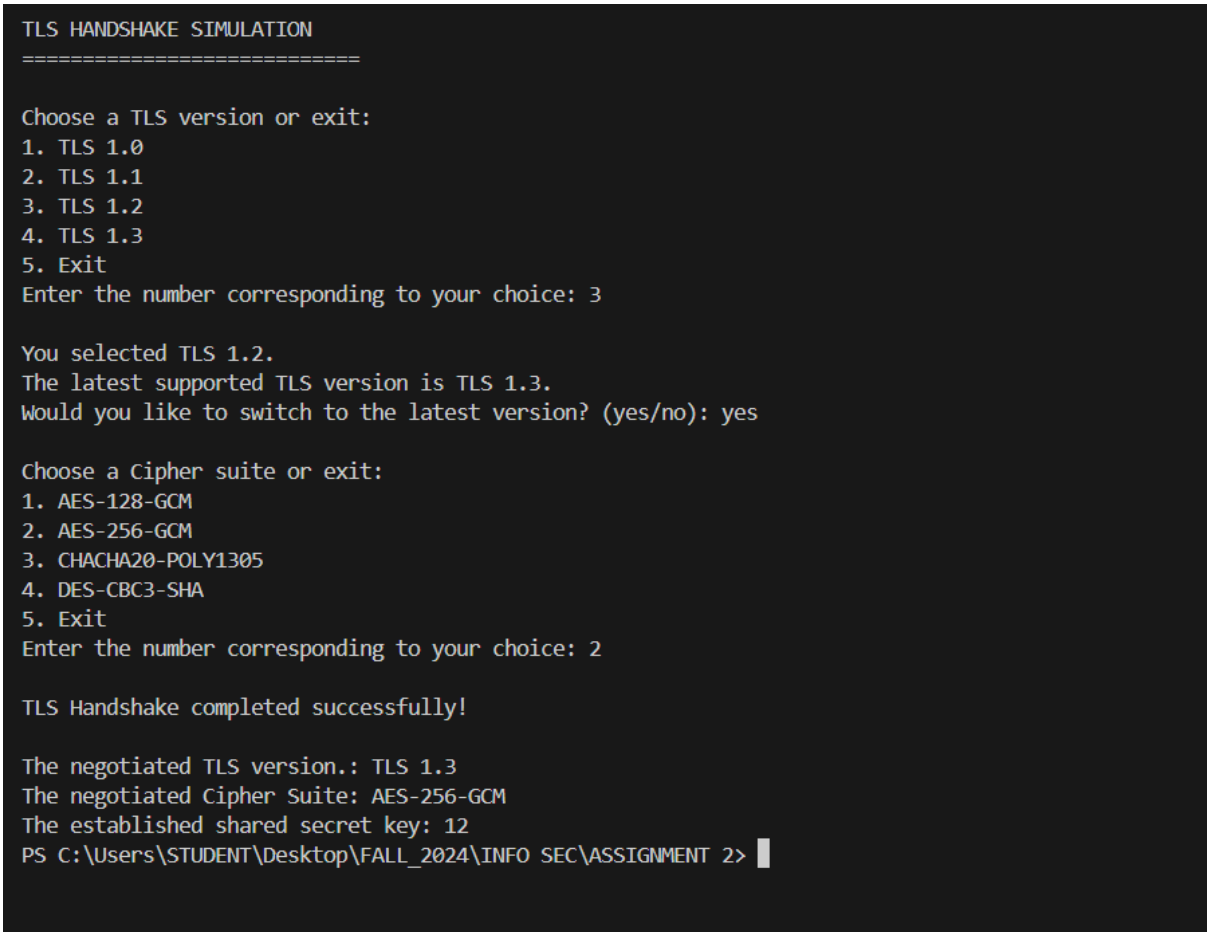
SUCCESSFUL HANDSHAKE



```
TLS HANDSHAKE SIMULATION
============================

Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice: 3

You selected TLS 1.2.
The latest supported TLS version is TLS 1.3.
Would you like to switch to the latest version? (yes/no): no

Choose a Cipher suite or exit:
1. AES-128-GCM
2. AES-256-GCM
3. CHACHA20-POLY1305
4. DES-CBC3-SHA
5. Exit
Enter the number corresponding to your choice: 3

TLS Handshake completed successfully!

The negotiated TLS version.: TLS 1.2
The negotiated Cipher Suite: CHACHA20-POLY1305
The established shared secret key: 9
PS C:\Users\STUDENT\Desktop\FALL_2024\INFO SEC\ASSIGNMENT 2>
```

Figure 3 Deny to switch to the latest version

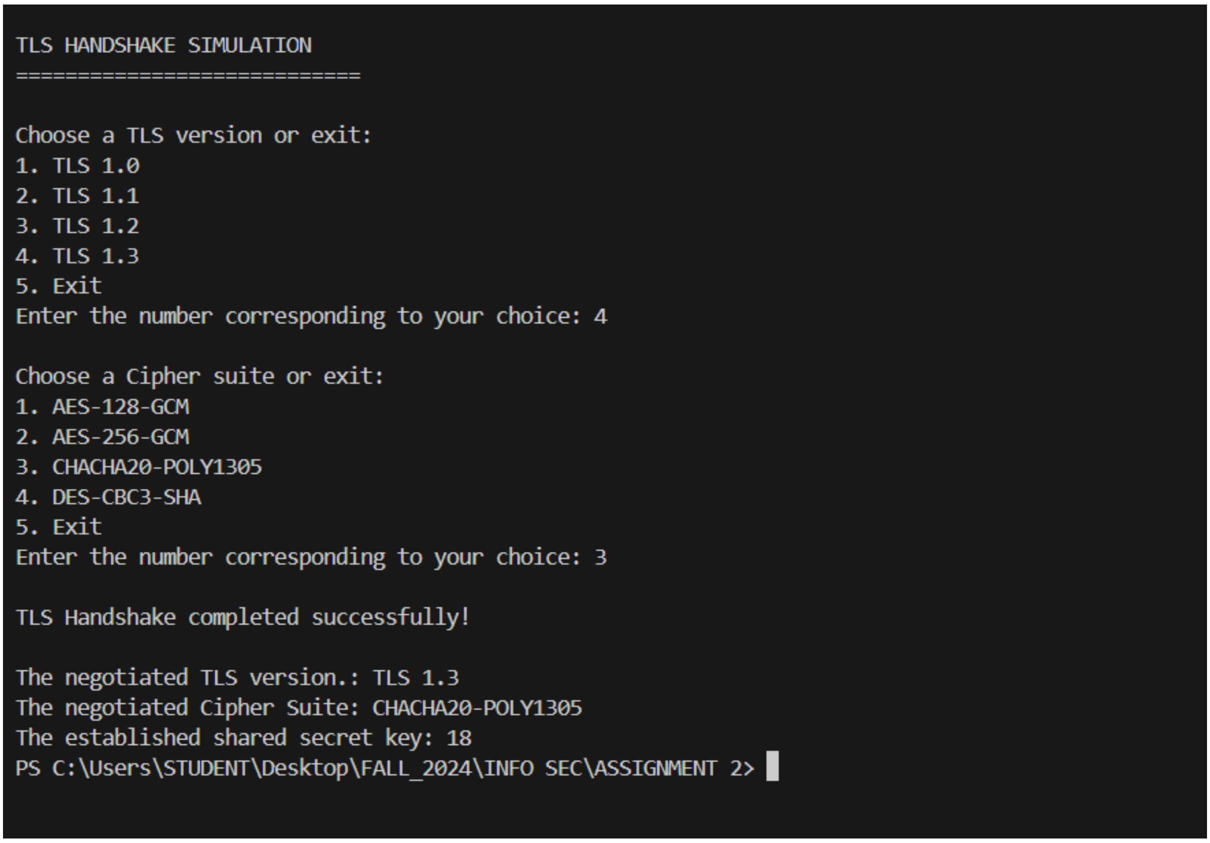
In this scenario, the user selects TLS 1.2, which is supported by the server. The program verifies this and recommends TLS 1.3, the latest or the highest version, but the user decides to continue with their initial choice by choosing ‘no’. The handshake proceeds with TLS 1.2 as negotiated. Next, the user selects AES-256-GCM for the cipher suite, which is supported by the server. The handshake successfully completes, establishing a secure connection with TLS 1.2 and AES-256-GCM, and the shared secret key is generated



```
TLS HANDSHAKE SIMULATION
============================

Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice: 3

You selected TLS 1.2.
The latest supported TLS version is TLS 1.3.
Would you like to switch to the latest version? (yes/no): yes

Choose a Cipher suite or exit:
1. AES-128-GCM
2. AES-256-GCM
3. CHACHA20-POLY1305
4. DES-CBC3-SHA
5. Exit
Enter the number corresponding to your choice: 2

TLS Handshake completed successfully!

The negotiated TLS version.: TLS 1.3
The negotiated Cipher Suite: AES-256-GCM
The established shared secret key: 12
PS C:\Users\STUDENT\Desktop\FALL_2024\INFO SEC\ASSIGNMENT 2>
```

Figure 4 Agrees to switch to the latest

In this case, the user initially selects **TLS 1.2**, which is supported by the server. The program suggests switching to **TLS 1.3**, the latest version because the selected one is not the latest/highest, and the user agrees by choosing ‘yes’. The handshake proceeds with TLS 1.3. The user then chooses **CHACHA20-POLY1305** for the cipher suite, which is also supported by the server. The handshake completes successfully with TLS 1.3 and CHACHA20-POLY1305, and a shared secret key is generated.

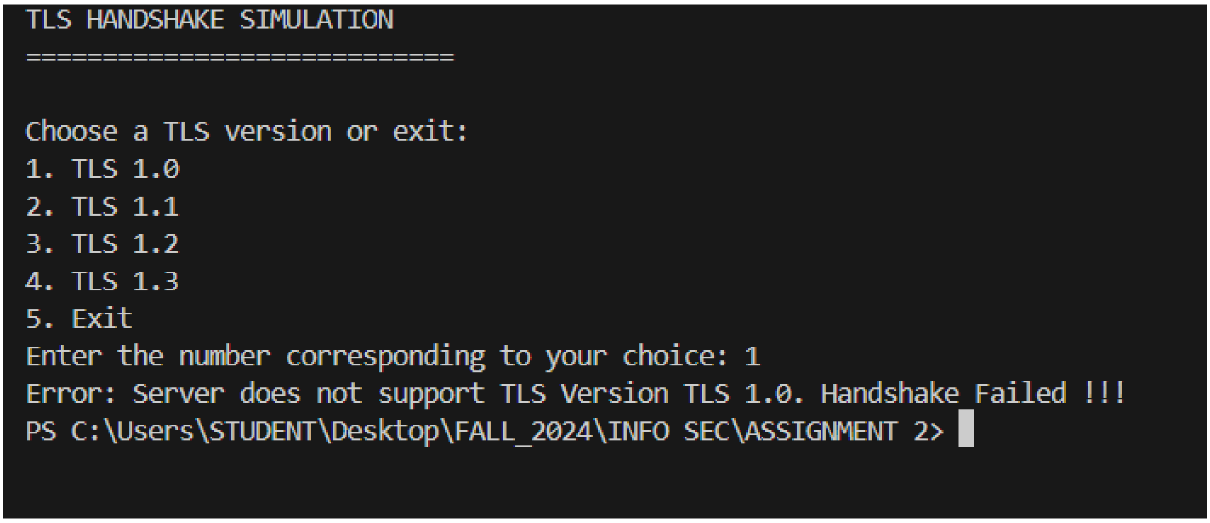


```
TLS HANDSHAKE SIMULATION
============================

Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice: 4

Choose a Cipher suite or exit:
1. AES-128-GCM
2. AES-256-GCM
3. CHACHA20-POLY1305
4. DES-CBC3-SHA
5. Exit
Enter the number corresponding to your choice: 3

TLS Handshake completed successfully!

The negotiated TLS version.: TLS 1.3
The negotiated Cipher Suite: CHACHA20-POLY1305
The established shared secret key: 18
PS C:\Users\STUDENT\Desktop\FALL_2024\INFO SEC\ASSIGNMENT 2> 
```

Figure 5 chosen the latest

Here, the user selects **TLS 1.3** right away, which is the latest supported version. Since it is supported by the server, no recommendation is needed. The user then chooses **CHACHA20-POLY1305** for the cipher suite, which is also supported. The handshake successfully completes with **TLS 1.3** and **CHACHA20-POLY1305**, establishing a secure connection and generating a shared secret key**.**

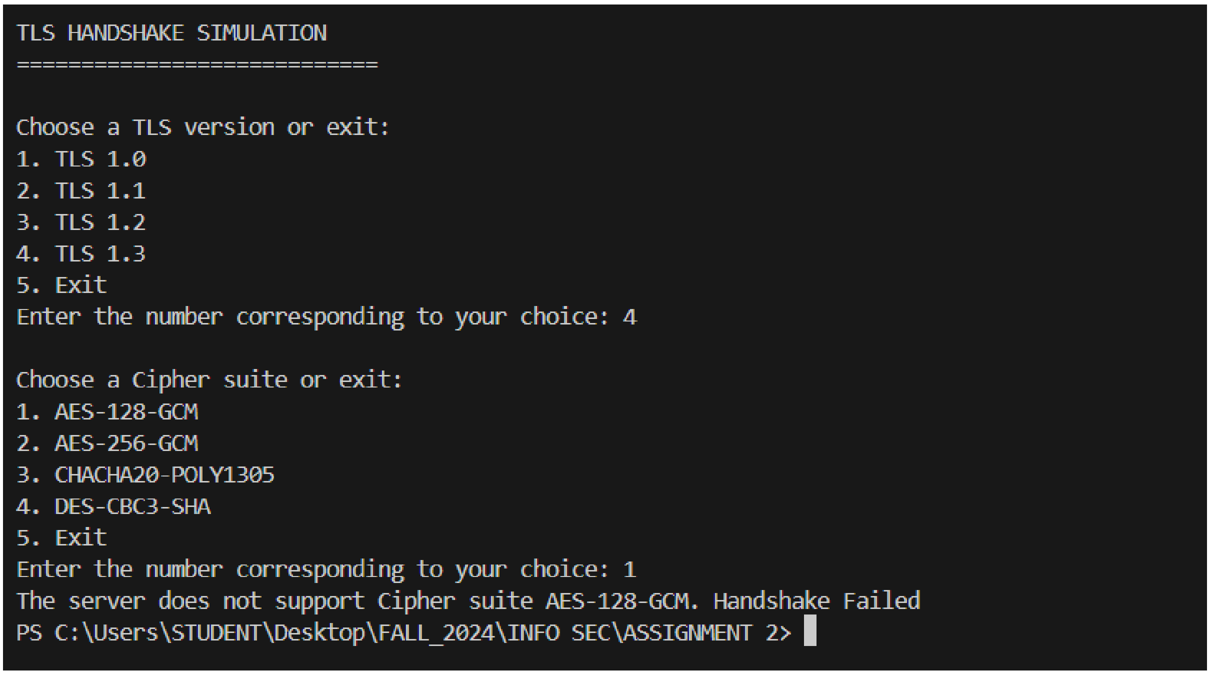
FAILED HANDSHAKE



```
TLS HANDSHAKE SIMULATION
============================

Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice: 1
Error: Server does not support TLS Version TLS 1.0. Handshake Failed !!!
PS C:\Users\STUDENT\Desktop\FALL_2024\INFO SEC\ASSIGNMENT 2>
```

Figure 6 Chosen unsupported version

In this scenario, the user selects **TLS 1.0**, which is not supported by the server. The program checks the chosen version against the supported versions and determines that TLS 1.0 is not supported. As a result, the handshake fails, and the user is notified that the selected version is not supported, ending the simulation without establishing a connection.



```
TLS HANDSHAKE SIMULATION
============================

Choose a TLS version or exit:
1. TLS 1.0
2. TLS 1.1
3. TLS 1.2
4. TLS 1.3
5. Exit
Enter the number corresponding to your choice: 4

Choose a Cipher suite or exit:
1. AES-128-GCM
2. AES-256-GCM
3. CHACHA20-POLY1305
4. DES-CBC3-SHA
5. Exit
Enter the number corresponding to your choice: 1
The server does not support Cipher suite AES-128-GCM. Handshake Failed
PS C:\Users\STUDENT\Desktop\FALL_2024\INFO SEC\ASSIGNMENT 2>
```

Figure 7 Chose unsupported cipher suite

Here, the user selects **TLS 1.2**, which is supported by the server, but chooses DES-CBC3-SHA as the cipher suite, which is not supported. The program verifies the cipher suite and finds that **DES-CBC3-SHA** is not among the supported options. so, the handshake fails, and the user is informed that the chosen cipher suite is not supported, ending the simulation without establishing a secure connection.

INTERPRETATION OF RESULTS

Successful Handshake: This indicates that the selected TLS version and cipher suite are supported by the server. The handshake process is completed successfully, and a shared secret key is generated, allowing secure communication.

Failed Handshake: This occurs when the selected TLS version or cipher suite is not supported by the server. In such cases, the program displays an error message and exits.

TLS Version Recommendation: If the user selects a TLS version that is supported but not the latest, the program will recommend switching to the latest version. This ensures the use of the most secure and updated protocol version, enhancing security.

DEMO INSTRUCTION

Running the Program:

1. Open a terminal
2. Navigate to the directory where the script is saved
3. Run the program with the command: python tls_handshake_simulation.py

Interaction:

- Follow the on-screen prompts to select a TLS version and cipher suite.
- Observe if the program recommends switching to the latest TLS version and decide whether to accept the recommendation.

4. **Exit**: The program provides an option to exit from both the TLS version and cipher suite menus. Select 'Exit' from the menu to terminate the program

REFERENCES

1. **https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/**
2. **https://www.ssl.com/article/ssl-tls-handshake-ensuring-secure-online-interactions/#:~:text=The%20SSL%2FTLS%20handshake%20is,and%2Dforth%20communication%20in%20milliseconds.**
3. **https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/**