CSCI 6708 NETWORK SECURITY

Assignment 4

1) Deriving secret key using Diffie Hellman Key Exchange

a) p= 11, g= 13

Alice and Bob exchange p & g values

selecting random value SA = 5

TA = g^{SA} mod p

= 13^{5} mod 11

= 10

selecting random value SB = 8

TB = g^{SB} mod p

= 13^{8} mod 11

= 3

Alice and Bob exchange TA & TB values

Secret key computation at Alice end

TB^{SA} mod p

= 3^{5} mod 11

= 243 mod 11

= 1 ----------------(a)

Secret key computation at Alice end

TA^{SB} mod p

= 10^{8} mod 11

= 100000000 mod 11

= 1 ------→ same as (a)

b) p= 7, g= 17

Alice and Bob exchange p & g values

selecting random value SA = 9

TA = g^{SA} mod p

= 17^{9} mod 7

= 6

selecting random value SB = 13

TB = g^{SB} mod p

= 17^{13} mod 7

= 3

Alice and Bob exchange TA & TB values

Secret key computation at Alice end

TB^{SA} mod p

= 3^9 mod 7

= 19683 mod 7

= 6 ----------------(a)

Secret key computation at Alice end

TA^{SB} mod p

= 6^{13} mod 7

= 1296 mod 7 *1296 mod 7 *1296 mod 7* 6 mod 7

= 1 *1* 1* 6

= 6 ------→ same as (a)

c) p= 17, g= 13

Alice and Bob exchange p & g values

selecting random value SA = 5

TA = g^{SA} mod p

= 13^5 mod 17

= 13

selecting random value SB = 11

TB = g^{SB} mod p

= 13^{11} mod 17

= 4

Alice and Bob exchange TA & TB values

Secret key computation at Alice end

TB^{SA} mod p

= 4^5 mod 17

= 1024 mod 17

= 4 ----------------(a)

Secret key computation at Alice end

TA^{SB} mod p

= 13^{11} mod 17

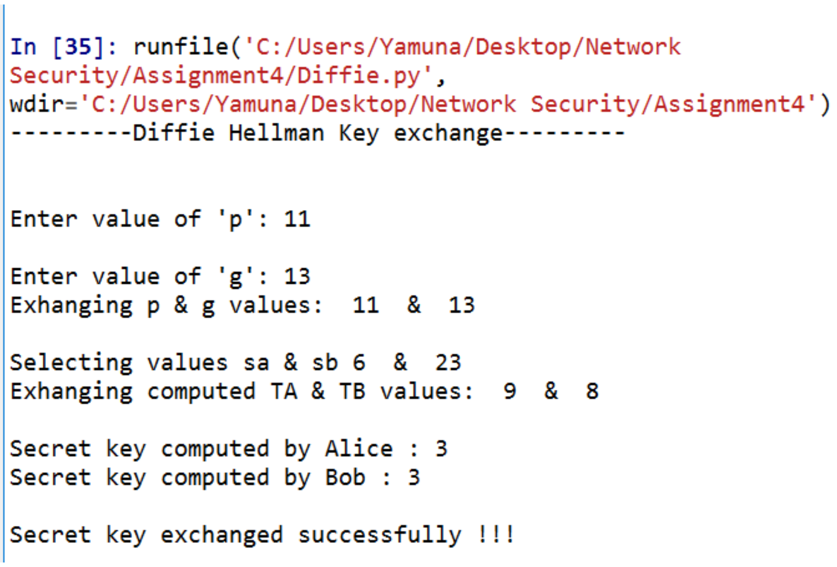
= 371293 mod 17 * 371293 mod 17 * 13 mod 17

= 4 ------→ same as (a)

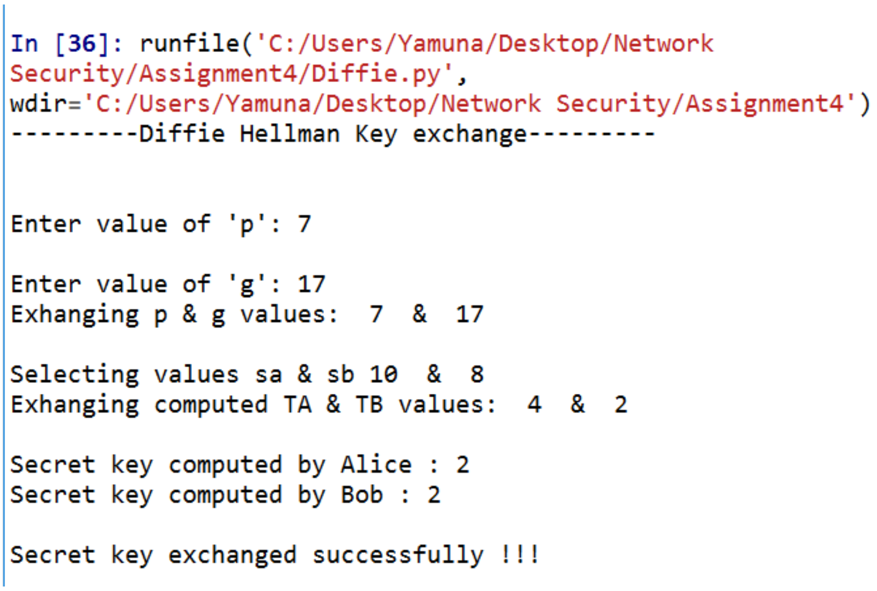
2) Python code illustrating Diffie Hellman Key exchange

Note: Python code file is attached with the report “Diffie.py”

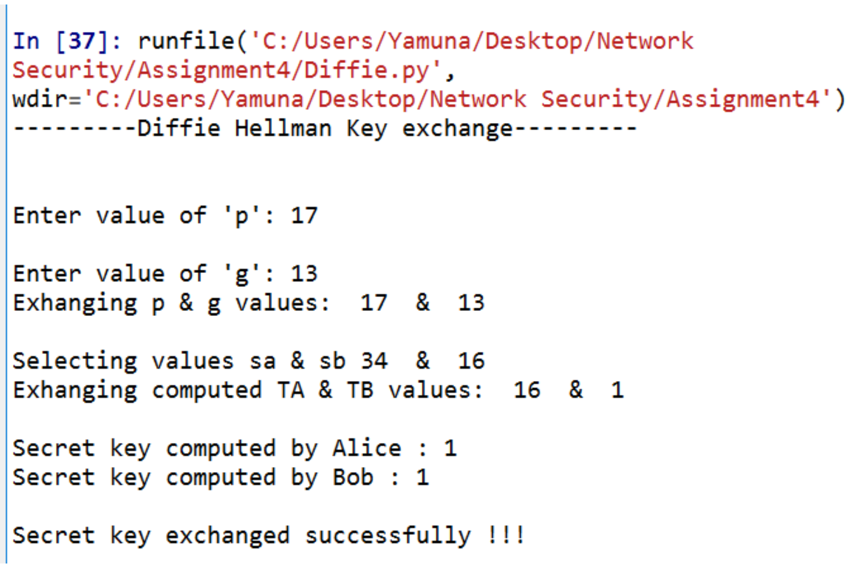
The code is run for the p & g values in Question 1 and results are captured below



```
In [35]: runfile('C:/Users/Yamuna/Desktop/Network
Security/Assignment4/Diffie.py',
wdir='C:/Users/Yamuna/Desktop/Network Security/Assignment4')
---------Diffie Hellman Key exchange---------


Enter value of 'p': 11

Enter value of 'g': 13
Exhanging p & g values:  11  &  13

Selecting values sa & sb 6  &  23
Exhanging computed TA & TB values:  9  &  8

Secret key computed by Alice : 3
Secret key computed by Bob : 3

Secret key exchanged successfully !!!
```



```
In [36]: runfile('C:/Users/Yamuna/Desktop/Network
Security/Assignment4/Diffie.py',
wdir='C:/Users/Yamuna/Desktop/Network Security/Assignment4')
---------Diffie Hellman Key exchange---------


Enter value of 'p': 7

Enter value of 'g': 17
Exhanging p & g values:  7  &  17

Selecting values sa & sb 10  &  8
Exhanging computed TA & TB values:  4  &  2

Secret key computed by Alice : 2
Secret key computed by Bob : 2

Secret key exchanged successfully !!!
```



```
In [37]: runfile('C:/Users/Yamuna/Desktop/Network
Security/Assignment4/Diffie.py',
wdir='C:/Users/Yamuna/Desktop/Network Security/Assignment4')
---------Diffie Hellman Key exchange---------


Enter value of 'p': 17

Enter value of 'g': 13
Exhanging p & g values:  17  &  13

Selecting values sa & sb 34  &  16
Exhanging computed TA & TB values:  16  &  1

Secret key computed by Alice : 1
Secret key computed by Bob : 1

Secret key exchanged successfully !!!
```

3) IP datagram encapsulation for given scenario

Authentication - ▨

Encryption - ░

Authentication & Encryption ▩

a. ESP transport mode only from end to end

(1) Original datagram

A, B	Payload

(2) A – G1

A, B	ESP Header	Payload	ESP Trailer

(3) G1 – G3

A, B	ESP Header	Payload	ESP Trailer

(4) G3 – G2

A, B	ESP Header	Payload	ESP Trailer

(5) G2 – B

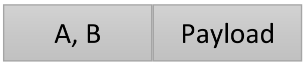
A, B	ESP Header	Payload	ESP Trailer

(6) At B

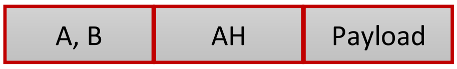
A, B	Payload

b. AH transport from A to B, ESP tunnel from firewall G1 to firewall G2

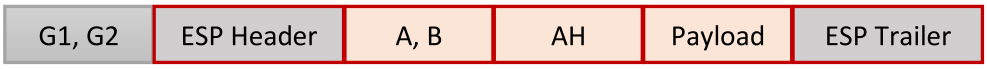
(1) Original datagram



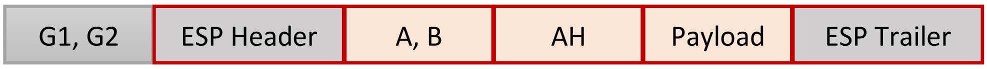
(2) A – G1



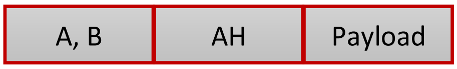
(3) G1 – G3



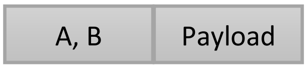
(4) G3 – G2



(5) G2 – B

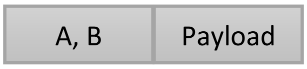


(6) At B

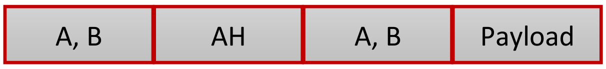


c. AH tunnel from A to B, ESP transport from firewall G1 to firewall G3

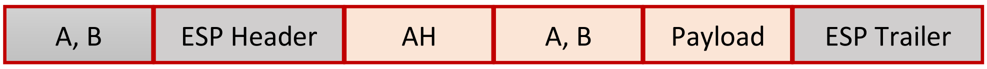
(1) Original datagram



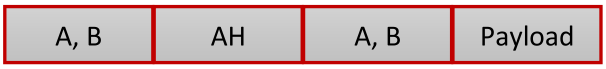
(2) A – G1



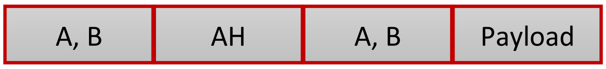
(3) G1 – G3



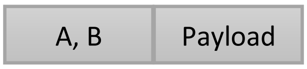
(4) G3 – G2



(5) G2 – B

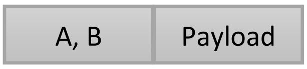


(6) At B

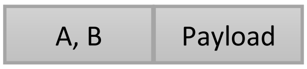


d. ESP tunnel from G3 to G2, and AH tunnel from G2 to B

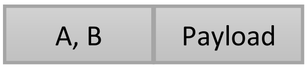
(1) Original datagram



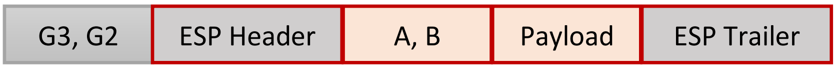
(2) A – G1



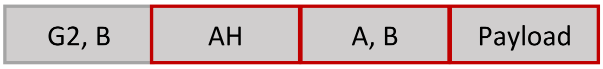
(3) G1 – G3



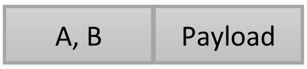
(4) G3 – G2



(5) G2 – B



(6) At B



4) Threats to VPNs overcome by particular feature of IPSec

a. Brute-Force Attack: An exhaustive search of the key space for a conventional encryption algorithm.

The ESP and AH header has the field named Security Parameters Index (SPI). It is a pointer to the encryption and the authentication algorithm from a list of negotiated algorithms. These algorithms are stored in the security association database. The encryption algorithm and the secret key used changes every few minutes the data is sent and it cannot be cracked by using this attack. All these parameters are negotiated in the Internet Key Exchange. Also, the packets are encrypted in the VPN.

b. Replay Attack: Earlier IPSec messages are replayed.

IPSec packets has a field called sequence number. This number is used to keep track of the packets received. If the packet is sent again to the host through replay attack the receiver will identify the packet to be illegitimate since the sequence number will be less than the last received sequence number. The packet will be dropped at the receiver end.

c. Man-in-the-middle attack: An attacker interposes during key exchange, acting as the client to the server and as the server to the client.

A VPN can be created using IPSec. This creates a tunnel between the sender and receiver. All the data is encrypted and the man-in-middle attack cannot be performed. Since the original IP header is also encrypted the attacker cannot know the exact source and destination address. Internet Key Exchange provides authentication in the first phase, hence this attack can be mitigated.

d. IP Spoofing: Uses forged IP addresses to fool a host into accepting bogus data.

Encapsulating Security Payload header is used in IPSec. This header tunnel mode encrypts the original IP header and mutual authentication takes place. Hence, the original IP cannot be obtained by the hacker. Hence the hacker cannot spoof the IP address of the user to fool the host for accepting bogus data.

e. SYN Flooding: An attacker sends TCP SYN messages to open half TCP connections.

In the initial steps of Internet key exchange that contains SYN cookies, cryptographic hashing of data and key exchange information is exchanged. After the initial communication, the server sends SYN-ACK responds with sequence number and other identifying information. When the client responds back, the hash is sent along with the ACK packet. The server verifies the ACK then only gives memory allocation for the connection.

REFERENCE

[1] Professor Handouts

[2] Imperva Incapsula, "TCP SYN Flood", https://www.incapsula.com/ddos/attack-glossary/syn-flood.html