EAST WEST UNIVERSITY

In-Course Assessment - 1

Course Code: CSE487

Course Title: Cyber Security, Ethics and Law

Section: 01

Submitted to:

Rashedul Amin Tuhin

Senior Lecturer,

Department of Computer Science and Engineering

Submitted by:

Md. Mahmud Alam

ID: 2018-3-60-014

Date of Submission: 11 March, 2022

ANSWER TO THE QESTION NO - 1

a.

```python
#Answer Q1.a:

x = '6da96dec2995ce9f2756f1ceb4f883b3e957f56fb5a649a6e3c02586207939be'

length = len(x)

print('\nLength of the hash value:',length)

unique = sorted(set(x))

print('\nSet of Unique char:',unique)

print('\nCount Unique char :',len(unique))
```

Console:

```
In [7]: runcell(0, 'D:/EWU/CSE487/CSE487.py')
Length of the hash value: 64

Set of Unique char: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f']

Count Unique char : 16
```

b.

Hence the given hash value is related to the CSE program, so 3 Uppercase letters will be ‘CSE’.

In Ubuntu ‘%’ represents numbers.

```
mahmud@mahmud:~/Desktop$ cd code/
mahmud@mahmud:~/Desktop/code$ crunch 6 6 -t CSE%%% -o string.txt
Crunch will now generate the following amount of data: 7000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000

crunch: 100% completed generating output
mahmud@mahmud:~/Desktop/code$
```

Size of the dictionary: 7000 bytes

```python
#Answer Q1.b:

import hashlib
import pandas as pd
import time

def hashConverter(string):
    hashValue = hashlib.sha256(string.encode()).hexdigest()
    return hashValue

series = pd.read_csv('string.txt', header=None)[0]

x = '6da96dec2995ce9f2756f1ceb4f883b3e957f56fb5a649a6e3c02586207939be'

start = time.time()

for i in series:
    if hashConverter(i) == x:
        string = i
        print('\nString Value:',string)
        break

end = time.time()

total_time = end - start
print("\nRequired Time:", str(total_time))
```

Console: **Required time to brute-force this dictionary.**

```
In [13]: runcell(3, 'D:/EWU/CSE487/CSE487.py')

String Value: CSE405

Required Time: 0.0009999275207519531

In [14]:
```

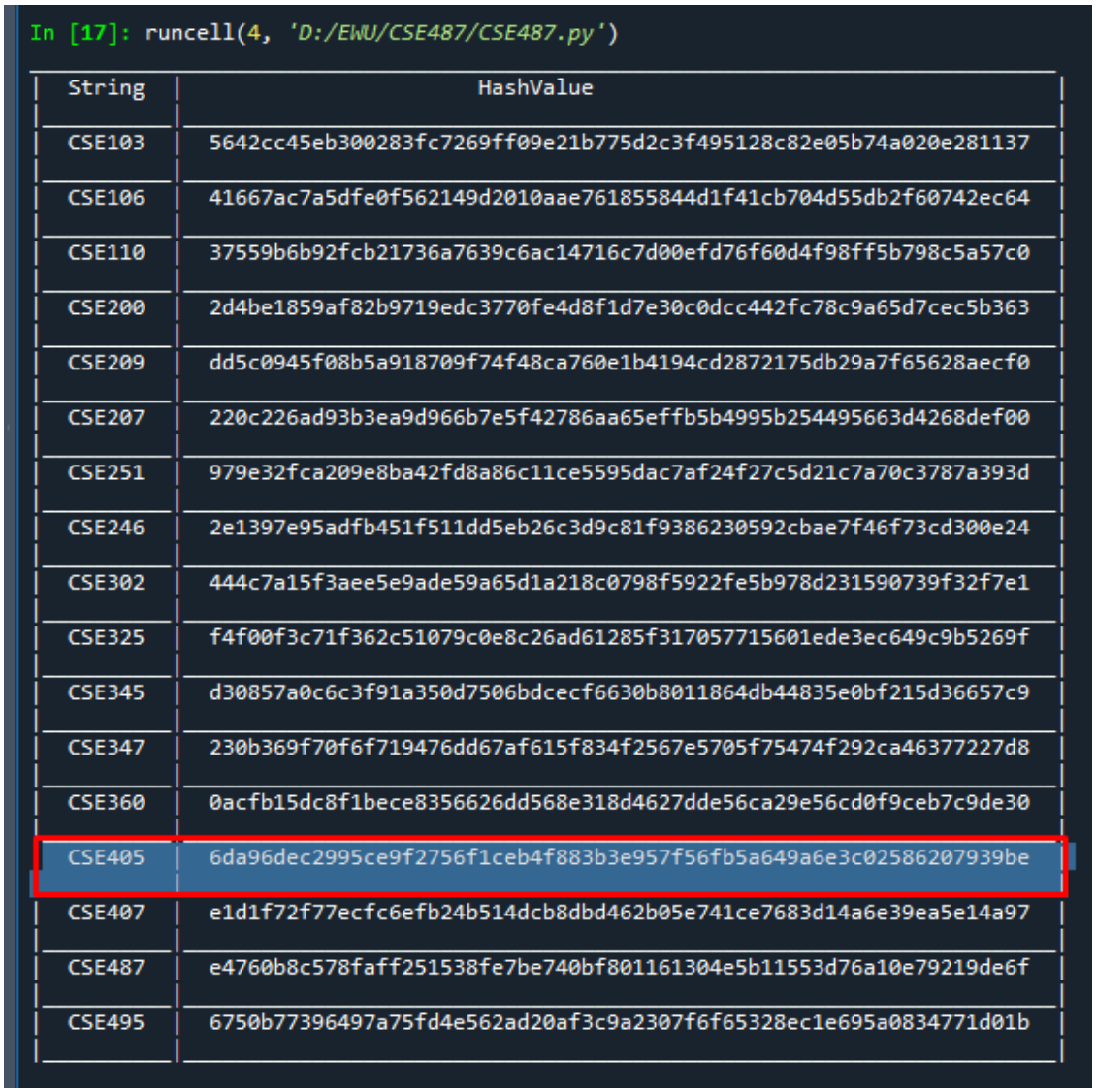
c.

```python
#Answer Q1.c:

courses = ['CSE103','CSE106','CSE110','CSE200','CSE209','CSE207',
           'CSE251','CSE246','CSE302','CSE325','CSE345','CSE347',
           'CSE360','CSE405','CSE407','CSE487','CSE495']

print('_____________________________________________________________')
print('|  String  |                    HashValue                     |')
print('|__________|__________________________________________________|')
for i in range(len(courses)):
    print('| ',courses[i],' | ',hashConverter(courses[i]),' |')
    print('|__________|__________________________________________________|')
```

CSE CORE COURSES

Console:

Rainbow Table with core CSE courses:



```
In [17]: runcell(4, 'D:/EWU/CSE487/CSE487.py')
```

String	HashValue
CSE103	5642cc45eb300283fc7269ff09e21b775d2c3f495128c82e05b74a020e281137
CSE106	41667ac7a5dfe0f562149d2010aae761855844d1f41cb704d55db2f60742ec64
CSE110	37559b6b92fcb21736a7639c6ac14716c7d00efd76f60d4f98ff5b798c5a57c0
CSE200	2d4be1859af82b9719edc3770fe4d8f1d7e30c0dcc442fc78c9a65d7cec5b363
CSE209	dd5c0945f08b5a918709f74f48ca760e1b4194cd2872175db29a7f65628aecf0
CSE207	220c226ad93b3ea9d966b7e5f42786aa65effb5b4995b254495663d4268def00
CSE251	979e32fca209e8ba42fd8a86c11ce5595dac7af24f27c5d21c7a70c3787a393d
CSE246	2e1397e95adfb451f511dd5eb26c3d9c81f9386230592cbae7f46f73cd300e24
CSE302	444c7a15f3aee5e9ade59a65d1a218c0798f5922fe5b978d231590739f32f7e1
CSE325	f4f00f3c71f362c51079c0e8c26ad61285f317057715601ede3ec649c9b5269f
CSE345	d30857a0c6c3f91a350d7506bdcecf6630b8011864db44835e0bf215d36657c9
CSE347	230b369f70f6f719476dd67af615f834f2567e5705f75474f292ca46377227d8
CSE360	0acfb15dc8f1bece8356626dd568e318d4627dde56ca29e56cd0f9ceb7c9de30
CSE405	6da96dec2995ce9f2756f1ceb4f883b3e957f56fb5a649a6e3c02586207939be
CSE407	e1d1f72f77ecfc6efb24b514dcb8dbd462b05e741ce7683d14a6e39ea5e14a97
CSE487	e4760b8c578faff251538fe7be740bf801161304e5b11553d76a10e79219de6f
CSE495	6750b77396497a75fd4e562ad20af3c9a2307f6f65328ec1e695a0834771d01b

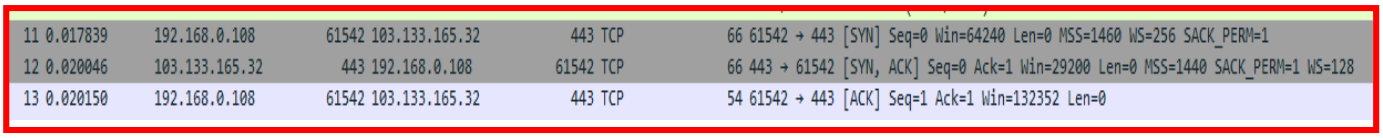
d.

Value of x = CSE405

ANSWER TO THE QESTION NO – 2

a.

Using appropriate filter ip.addr == 103.133.165.32, identify the TCP 3-way handshake packets



11	0.017839	192.168.0.108	61542	103.133.165.32	443	TCP 66 61542 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
12	0.020046	103.133.165.32	443	192.168.0.108	61542	TCP 66 443 → 61542 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440 SACK_PERM=1 WS=128
13	0.020150	192.168.0.108	61542	103.133.165.32	443	TCP 54 61542 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0

b.

NAT:

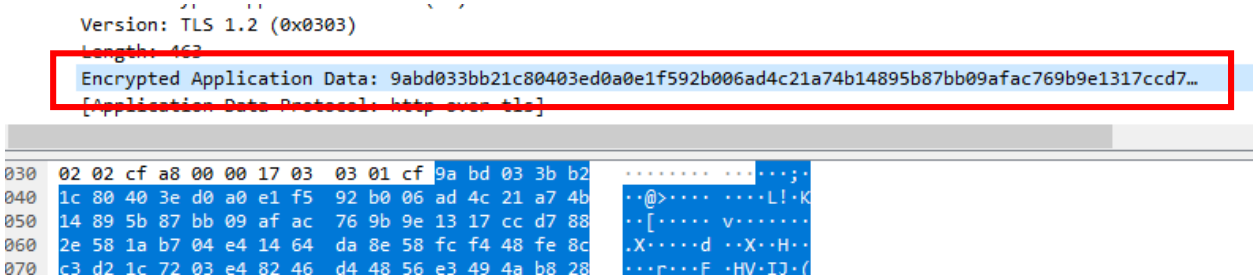
#	IP	Port
Source	192.168.0.108	61542
Destination	103.133.165.32	443

c.

First encrypted packet.

31	0.102211	192.168.0.108	61542	103.133.165.32	443	TLSv1.2 522 Application Data

#Inside the packet

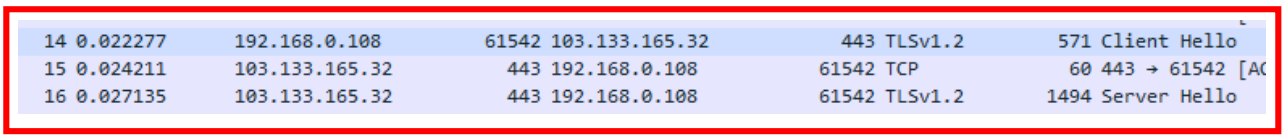


```
Version: TLS 1.2 (0x0303)
Length: 463
Encrypted Application Data: 9abd033bb21c80403ed0a0e1f592b006ad4c21a74b14895b87bb09afac769b9e1317ccd7…
[Application Data Protocol: http over tls]
```

```
030  02 02 cf a8 00 00 17 03  03 01 cf 9a bd 03 3b b2   ···· ···· ···;·
040  1c 80 40 3e d0 a0 e1 f5  92 b0 06 ad 4c 21 a7 4b   ··@>···· ····L!·K
050  14 89 5b 87 bb 09 af ac  76 9b 9e 13 17 cc d7 88   ··[···· v·····
060  2e 58 1a b7 04 e4 14 64  da 8e 58 fc f4 48 fe 8c   .X·····d ··X··H··
070  c3 d2 1c 72 03 e4 82 46  d4 48 56 e3 49 4a b8 28   ···r···F ·HV·IJ·(
```

d.

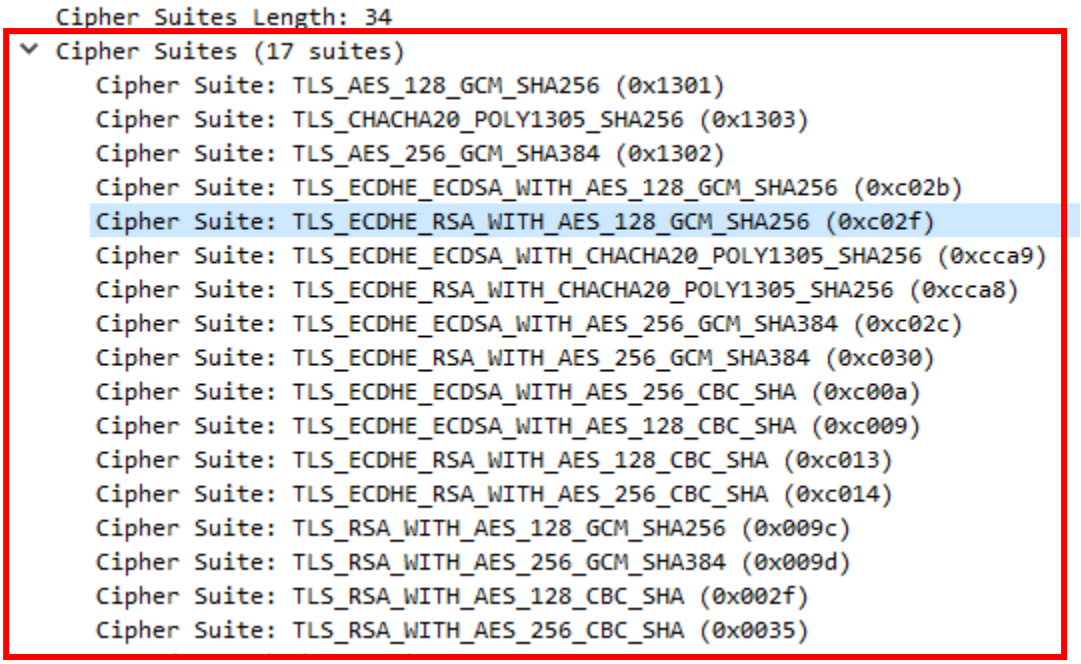
Client Hello packet no. 14 and *Server Hello* packet no. 16



14	0.022277	192.168.0.108	61542	103.133.165.32	443	TLSv1.2 571 Client Hello
15	0.024211	103.133.165.32	443	192.168.0.108	61542	TCP 60 443 → 61542 [AC
16	0.027135	103.133.165.32	443	192.168.0.108	61542	TLSv1.2 1494 Server Hello

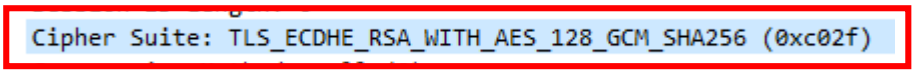
e.

The cipher suites offered by the client.



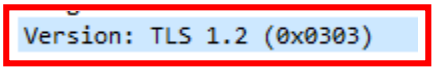
f.

The cipher suite selected by the server.



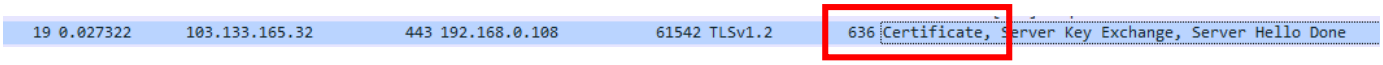
g.

TLS version: TLS 1.2



h.

The **packet no 19** is which EWUBD.EDU offered its certificate.



i.

HTTPS decryption demonstrating the release of message contents attack:

Generate my session key, **ssl-keys.log**

Input the key path: Wireshark>>Edit>>preferences>>protocols>>TLS>>ssl-keys.log

Fig1:

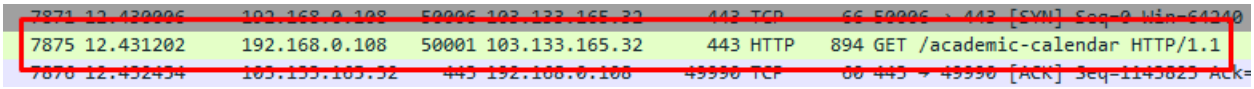
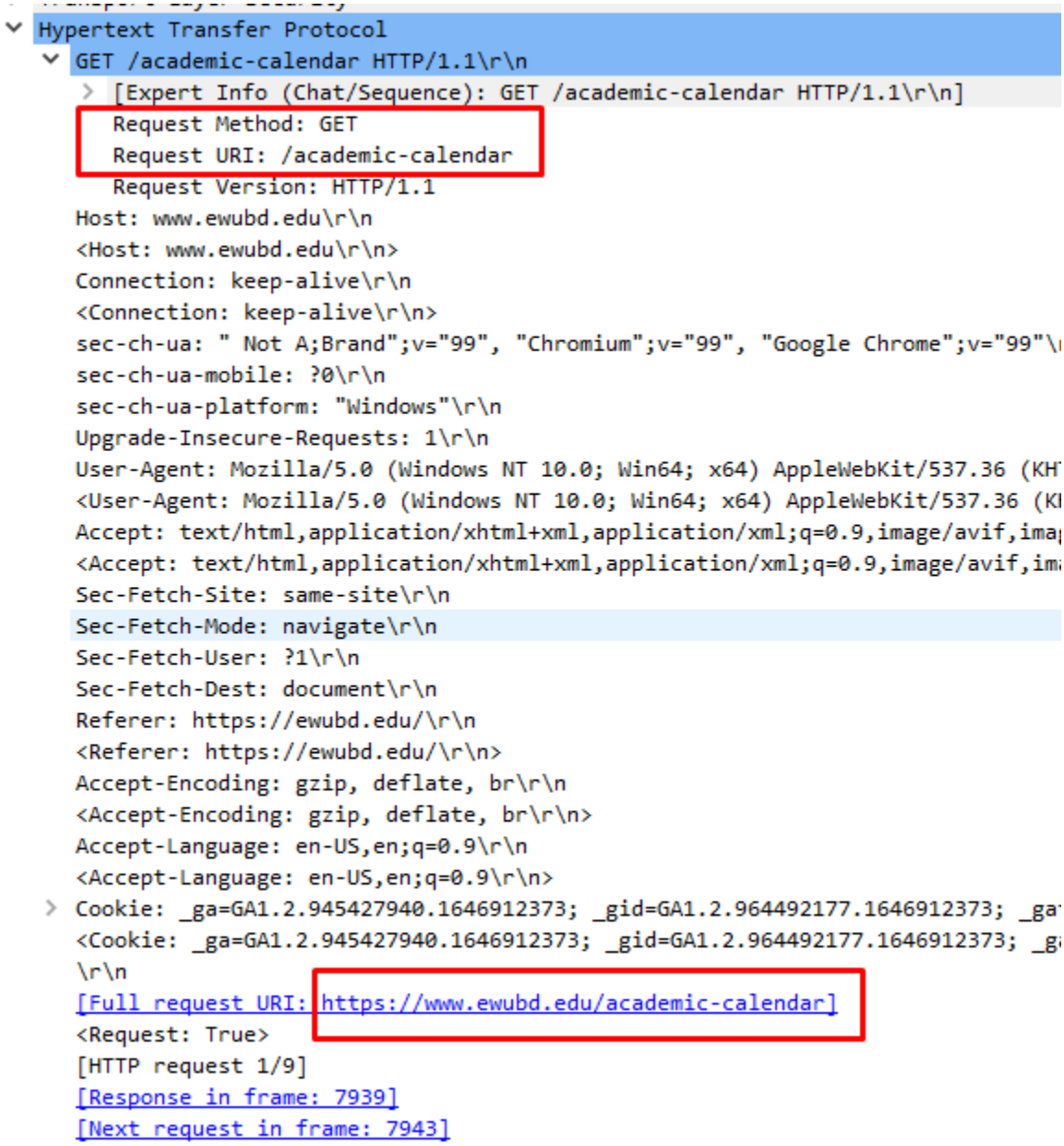


Fig2:



ANSWER TO THE QESTION NO – 3

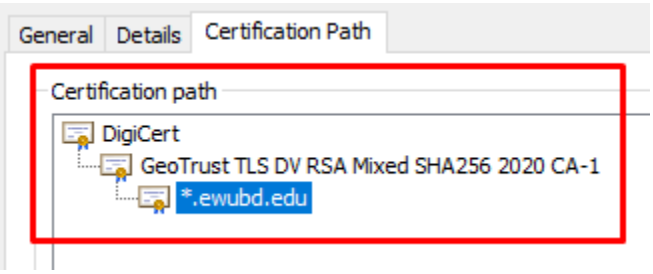
a.

The intended purpose of the certificate is:

1. Certificate provides Authentication. With this certificate, we can be sure that we are sending information to the right server and not to an imposter trying to steal our information.
2. This SSL certificate encrypt our sensitive information. This is very important because the information we send on the Internet is passed from computer to computer to get to the destination server. Any computer in between us and the server can see our credit card numbers, usernames and passwords, and other sensitive information if it is not encrypted with an SSL certificate.
3. It also proves website's identity to a remote computer.

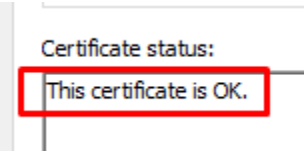
b.

The issuer of this certificate, with the CA trust chain of the SSL Certificate issued to EWUBD.EDU:



c.

The validity of the certificate:



No, this certificate is not valid for **www.ewubd.edu.bd**

This certificate will expire in **25 February, 2023**.

d.

The hashing algorithm used to create the certificate is **SHA256**.

e.

The algorithm used by the CA to sign the certificate is **SHA256RSA**.

f.

The public key of EWUBD.EDU:

```
30 82 01 0a 02 82 01 01 00 d2 a5 2c 35 d0 e6 0a e7 4e d0 de 83 80 94 48 59 2b 9b da 2c ec
83 3a 50 0f df 10 19 ab 37 41 0a 5e 3e 8c 36 e4 44 dc 78 54 d3 4f 8a 09 2a 52 2a 61 75 7d
97 6d 2c f7 6b 22 5a 76 21 b7 25 23 1d 5d 8a 0d a2 f9 66 75 7a e5 75 1e ab 8d 11 15 4e 00
ce 24 1a 5f dc bc df 99 9e b5 3a 95 55 b4 bf a0 a0 a0 3e 54 2d 46 ae 66 cc 19 12 a9 59 16
5d 46 a0 f8 be b4 6a 57 c0 d7 ed be 22 9e 51 5d 80 f2 e6 1c fe bb 40 ed 0d 73 a2 aa 2b c2
09 7a 14 d9 e8 81 98 7b d3 63 1a 4f 2f 8c 7f 64 3c 79 c9 be 2c da e8 ef a0 23 3e 68 36 2e
74 20 4d 3b 5d 3b 67 20 3b d6 b0 aa 15 dc 18 e4 3a 20 cb 9c f2 23 c1 13 d7 38 a4 fd 37 e3
e0 ec 77 62 4f e9 7f 2b 60 db b1 15 09 d8 68 d4 67 77 ff d4 cf fd be 08 b0 4c 1b a1 e2 c6
11 0c 7b 33 81 69 09 ef fc 62 6b 35 b5 ea ff 14 b1 c7 59 d3 d2 f3 01 b1 85 02 03 01 00 01
```

The length of this public key is 2048 Bits.

g.

The public key algorithm that was used to generate public-private key pair is **RSA**.

h.

Certificate revocation is the act of invalidating a TLS/SSL before its scheduled expiration date. Certificates that are revoked are stored on a list by the CA, called the Certificate Revocation List(CRL).

[1] CRL Distribution Point
Distribution Point Name:
Full Name:
URL=http://crl3.digicert.com/GeoTrustTLSDVRSAMixedSHA2562020CA-1-1.crl
[2] CRL Distribution Point
Distribution Point Name:
Full Name:
URL=http://crl4.digicert.com/GeoTrustTLSDVRSAMixedSHA2562020CA-1-1.crl

i.

CAs use a private key to cryptographically sign all issued certificates. Such signatures can irrevocably prove that a certificate was issued by a specific CA and that it was not modified after it was signed. CAs establish ownership of their signing key by holding a self-issued certificate (called the *root*) for the corresponding public key. CAs have to observe tightly controlled and audited procedures to create, manage and utilize a root, and to minimize exposure will normally use a root to issue *intermediate* certificates. These intermediates can then be used to issue their customers' certificates.

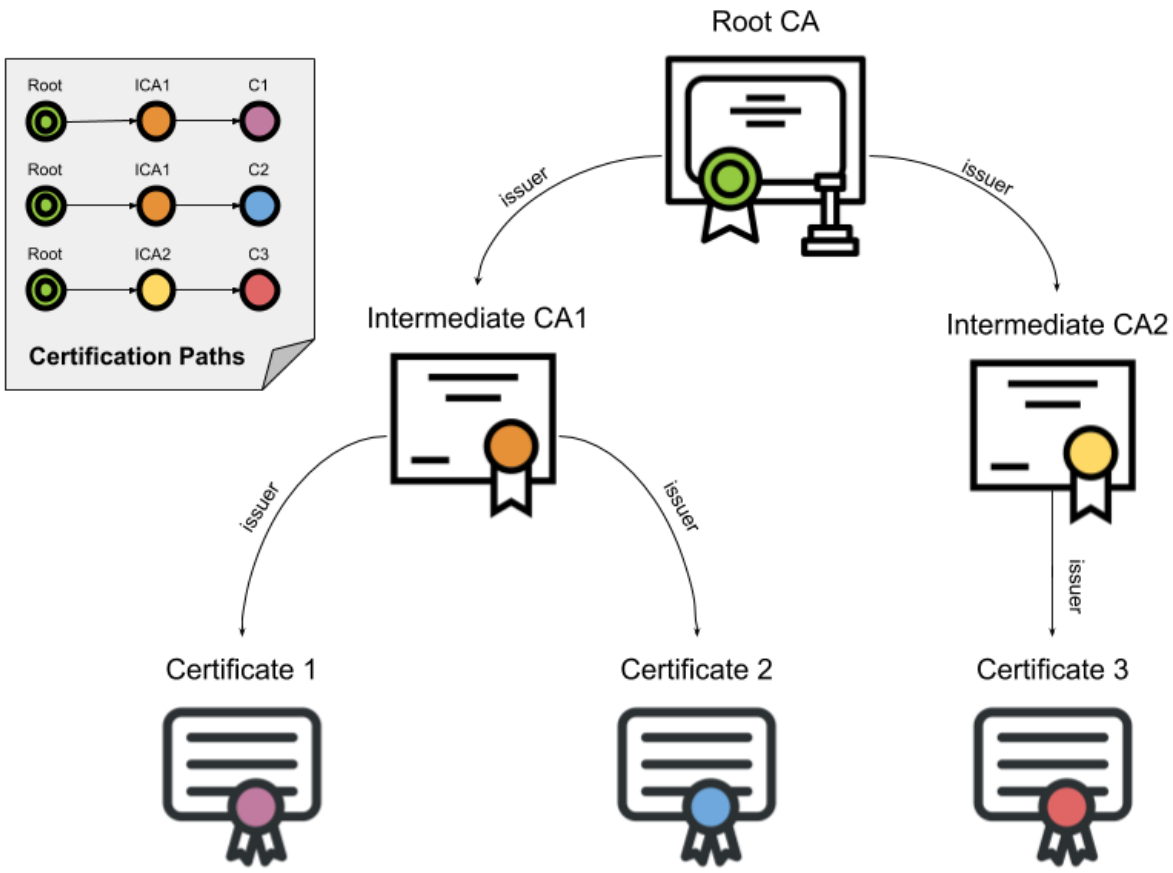


Figure: Certification authority tree

Browsers are shipped with a built-in list of trusted roots. (These are roots from CAs who have passed the browser's stringent criteria for inclusion.) To verify a certificate, a browser will obtain a sequence of certificates, each one having signed the next certificate in the sequence, connecting the signing CA's root to the server's certificate. This sequence of certificates is called a *certification path*. The path's root is called a *trust anchor* and the server's certificate is called the *leaf* or *end entity* certificate.

Oftentimes browsers have to consider multiple certification paths until they can find a valid one for a given certificate. Even though a path may contain certificates that "chain" together properly to a known anchor, the path itself may be rejected due to restrictions on path length, domain name, certificate usage, or policy. Constructing and evaluating all possible paths is an expensive process performed for every new certificate a browser encounters. Browsers have implemented various optimizations to minimize the number of rejected candidate paths.

After a candidate certification path is constructed, browsers validate it using information contained in the certificates. A path is valid if browsers can cryptographically prove that, starting from a certificate directly signed by a trust anchor, each certificate's corresponding private key was used to issue the next one in the path, all the way down to the leaf certificate.

ANSWER TO THE QESTION NO – 4

a.

Alice and Bob can establish a secure connection using the RSA Algorithm. Alice generates a public/private key pair and using Alice's public key, Bob can encrypt a message M and sent to Alice through internet. When the encrypted message M surfing through internet, everyone who sits on **Wireshark** gets the encrypted message, so our well-known hacker Eve also gets it but he cannot decrypt it because of the RSA algorithm. **In RSA Algorithm**, a message which encrypted by receiver's public key, only decrypt by receiver's private key which means only Alice can decrypt the message using her private key. So, even if Eve got their encrypted message, he cannot decrypt it and this is how Alice and Bob can establish a secure connection using the RSA Algorithm.

For calculating Alice's public/private key pair in RSA Algorithm, here are the steps:

1. Select two prime numbers, p = 23 and q = 7

2. Calculate n = p X q = 23 X 7 = 161

3. Calculate Φ(n) = (p - 1) (q - 1) = 22 X 6 = 132

4. Select *e* such that *e* is relatively prime to Φ(n) = 132 and less than Φ(n); we choose *e* = 19

5. Determine *d* such that $de \equiv 1 \pmod{132}$ and $d < 132$. The correct value is *d* = 7,

because 7 X 19 = 133 = (1 X 132) + 1

Alice's key pair:

public key PubA = {19, 161}

private key PvtA = {7, 161}

Now, for calculating Bob's public/private key pair in RSA Algorithm:

1. Select two prime numbers, p = 17 and q = 11

2. Calculate n = p X q = 17 X 11 = 187

3. Calculate Φ(n) = (p - 1) (q - 1) = 16 X 10 = 160

4. Select *e* such that *e* is relatively prime to Φ(n) = 160 and less than Φ(n); we choose *e* = 7

5. The correct value is *d* = 23, because 23 X 7 = 161 = (1 X 160) + 1

Bob's key pair:

public key PubB = {7, 187}

private key PvtB = {23, 187}

Now, Bob wants to send a message M=9 to Alice. So Bob encrypt this message M=9 with Alice's public key PubA = {19, 161}

Here is the calculation:

Main message, M = 9 and public key of Alice is, PubA = {19, 161}

So, 9^{19} mod 161 = 128

Now Bob send this encrypted message, C = 128

Alice got the encrypted message and now she decrypts it through her private key PvtA = {7, 161}

Here is the calculation:

Encrypted message, C = 128 and private key of Alice is, PvtA = {7, 161}

So, 128^{7} mod 161 = 9

Finally, Alice got the actual message, M = 9

b.

Confidentiality is one of the CIA triad concepts, which means when someone sends a message to a particular person, no one else cannot see that message. If other people can see that message, then we say message content confidentiality has been breached. So, confidentiality is a security services which ensures sender and receiver that no one else did not see their message content.

Non-Repudiation is another security services, which provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. It carries proves that who sent the message and who received the message. So, after sending something to someone, sender cannot deny that he did not send the massage and also receiver cannot deny that he did not receive the message.

Both Confidentiality and Non-Repudiation terms can be achieved by the RSA Algorithm.

RSA algorithm is an asymmetric encryption algorithm, and it has a pair of keys, one is private key, and the another one is public key. In this cryptography algorithm both public and private key can encrypt a message. If the public key is used for encrypted the message, then the private key is used for decrypt it. So, when Bob sends any message to Alice, the message should be encrypted by Alice's public key, and this message is only decrypt by Alice private key. So, even if Eve got the encrypted message and tries to decrypt it, he can't do it for not having Alice private key. So only Alice can decrypt the message with her private key. In this way, RSA algorithm achieved message confidentiality and also non-repudiation which ensured to prevent **Man In The Middle (MITM) attacks**.

c.

The values of *e* (encryptor/exponent) and n (modulus) for my case is: *e* = 19 and n = 161

For my case, the public key is: {19, 161}

The public key of EWUBD.EDU is:

```
30 82 01 0a 02 82 01 01 00 d2 a5 2c 35 d0 e6 0a e7 4e d0 de 83 80 94 48 59 2b 9b da 2c ec
83 3a 50 0f df 10 19 ab 37 41 0a 5e 3e 8c 36 e4 44 dc 78 54 d3 4f 8a 09 2a 52 2a 61 75 7d
97 6d 2c f7 6b 22 5a 76 21 b7 25 23 1d 5d 8a 0d a2 f9 66 75 7a e5 75 1e ab 8d 11 15 4e 00
ce 24 1a 5f dc bc df 99 9e b5 3a 95 55 b4 bf a0 a0 a0 3e 54 2d 46 ae 66 cc 19 12 a9 59 16
5d 46 a0 f8 be b4 6a 57 c0 d7 ed be 22 9e 51 5d 80 f2 e6 1c fe bb 40 ed 0d 73 a2 aa 2b c2
09 7a 14 d9 e8 81 98 7b d3 63 1a 4f 2f 8c 7f 64 3c 79 c9 be 2c da e8 ef a0 23 3e 68 36 2e
74 20 4d 3b 5d 3b 67 20 3b d6 b0 aa 15 dc 18 e4 3a 20 cb 9c f2 23 c1 13 d7 38 a4 fd 37 e3
e0 ec 77 62 4f e9 7f 2b 60 db b1 15 09 d8 68 d4 67 77 ff d4 cf fd be 08 b0 4c 1b a1 e2 c6
11 0c 7b 33 81 69 09 ef fc 62 6b 35 b5 ea ff 14 b1 c7 59 d3 d2 f3 01 b1 85 02 03 01 00 01
```

Comparison:

For my case, I consider p=23 and q=7 two prime numbers which are smaller than 100 and the n (modulus) = 161 which is 8 Bits long

But for EWUBD.EDU, their public key is 2048 Bits long. So, obviously their two prime numbers, p and q must be very large numbers.

ANSWER TO THE QESTION NO – 5

With the help of diagrams, Security services and Security attacks are explaining below:

Security service: A service that enhances the security of the data processing systems and the information transfers of an organization or any individuals. **RFC4949 is an Internet Security Glossary** which provides definitions, abbreviations, and explanations of terminology for information system security.

Nowadays there are many types of security services available. I discussed some of them below:

1. **Authentication:** This services provides assurances that the people who are communicating each other, are the real one, not pretender. There are two kinds of authentication services:
    i. **Peer Entity Authentication:** The people who are communicating are real, this ensures peer entity authentication.
    ii. **Data Origin Authentication:** In a connectionless transfer, the data origin authentication provides assurance that the source of received data is real.
2. **Access Control:** This security service provides who can access what in a system. It prevents user to use something who are unauthorized. Only authorized users can access their data.
3. **Data Confidentiality:** This service protect data from unauthorized users. There are some kinds of data confidentiality services:
    i. **Connection Confidentiality:** This is the protection of all user data on a connection channel.
    ii. **Connectionless Confidentiality:** This is the protection of all user data in a single data block.
    iii. **Selective Field Confidentiality:** The confidentiality of selected fields within the user data on a connection or in a single data block.
    iv. **Traffic Flow Confidentiality:** The protection of the information that might be derived from observation of traffic flows.
4. **Data Integrity:** This service assures users that what message they received are sent by authorized users. No changes or modifications did not happen by any hacker. There are two types of data integrity:
    i. **Connection Integrity with Recovery:** If any king of change, modification, deletion happened in the message contain then system can detect the change and recover the actual message contain with this service.
    ii. **Connection Integrity without Recovery:** If any king of change, modification, deletion happened in the message contain then system can only detect it that a modification has been occur but system cannot recover the actual message contain with this service.

Security attack: Security attack is an action that compromises the security of information owned by an organization or any individuals.

There are two types of security attacks:

1. **Passive Attacks:** Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The purpose of the hacker is to collect information that is being transmitted.

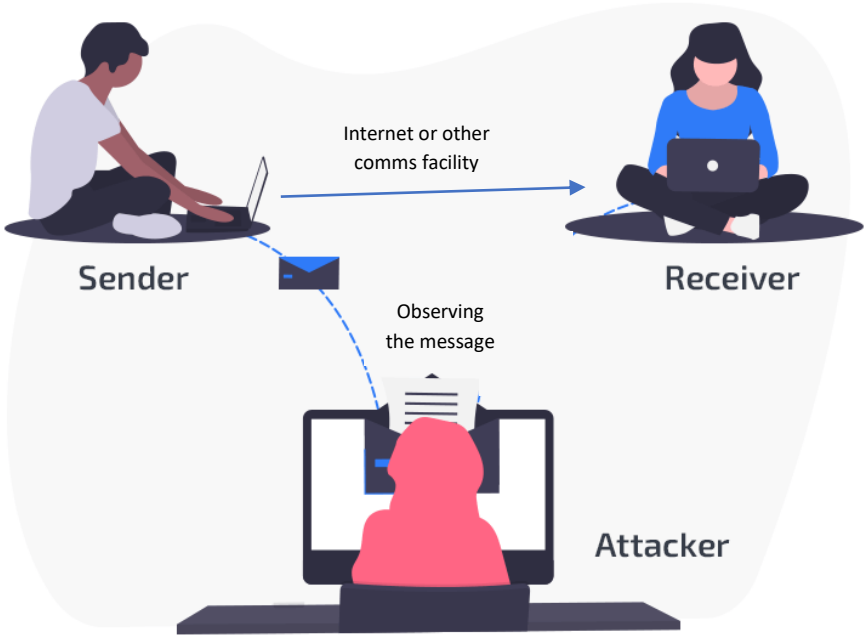


Figure 1: Passive attacks scenario

There are two types of passive attacks:

i. **Release of message contents:** This is easily understood from the above image. A hacker can be seated in our communication channel and observe our every content such as, our telephone conversation, our electronic mail message, and our transferred file which may contain sensitive or confidential information.

ii. **Traffic analysis:** If somehow we got a way to encrypt the messages, so the hacker, even if he captured our encrypted messages, he could not decrypt the information from the message. But still the hacker might be able to observe the pattern of these messages. The hacker could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

2. **Active attacks:** In an active attack, the hacker modify the sender's message and sends the modified message to the receiver. The purpose of the hacker is to manipulate information that is being transmitted.

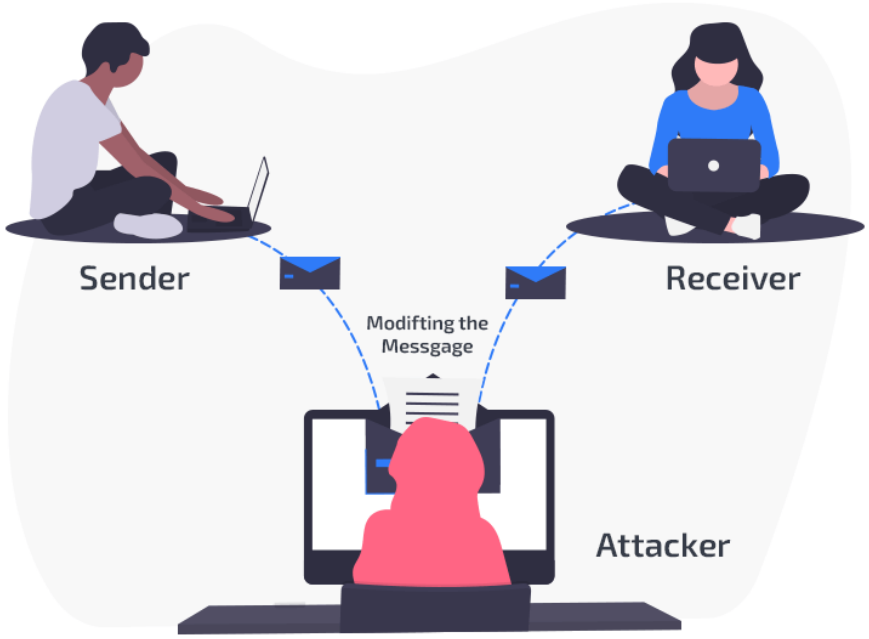


Figure 2: Active attacks scenario

There are four types of active attacks:

i. **Masquerade:** Masquerade takes place when hacker pretends to be an another person. For instance, hacker sends a message to Alice that appear to be from Bob. Here hacker pretend that he is Bob who sends messages to Alice.

ii. **Replay:** One example of replay is, after captured the message from sender, hacker intentionally delayed the message and send to the receiver, which showed a long distance to receiver and he thinks he is way far from where he is supposed to be.

iii. **Modification of messages:** It means, hacker will change the sender message and then send the modified message to the receiver. For instance, sender send “hi” to the reviser, hacker gets the message and modify the message to “bye” and send it to receiver. So receiver get “bye” message.

iv. **Denial of service (DOS):** When a hacker generates millions of half open connections to a server just for making the server unavailable to their legitimate users, that type of attack called DOS attack.

ANSWER TO THE QESTION NO – 6

a.

After established a secure connection with asymmetric encryption, they would still want to establish another connection with a symmetric key because symmetric key is a session key which is faster encryption than asymmetric encryption. Hence, asymmetric encryption need public/ private key pair for encryption/decryption a message, so that asymmetric encryption is more secure but for complex calculation, this encryption becomes computationally expensive and slow. So for making their encryption faster, they still want to establish a symmetric encryption.

b.

Diffie-Hellman Key Exchange calculation demonstrate below:

Suppose,

Alice's private key, x = 19 [Only Alice know]

Bob's private key, y = 23 [Only Bob know]

Generator, g = 5 and n = 11 [Both g and n are publicly shared]

Now Alice will generate a public key, $A = g^x \bmod n$

$= 5^{19} \bmod 11$

$= 9$

Now Bob will generate a public key, $B = g^y \bmod n$

$= 5^{23} \bmod 11$

$= 4$

After generating the public keys, now they exchange them to each other.

With the help of Bob's public key, B = 4, Alice will generate a secret key, $K1 = B^x \bmod n$

$= 4^{19} \bmod 11$

$= 3$

With the help of Alice's public key, A = 9, Bob will generate a secret key, $K2 = A^y \bmod n$

$= 9^{23} \bmod 11$

$= 3$

Here, K1 and K2 is equal. So, now Alice and Bob can encrypt their message with their private key and eavesdropper Eve cannot decrypt it.

c.

In Diiffie-Hellman Key Exchange, generator g and n are two large numbers which are publicly shared so anyone in the internet can get the numbers. Now for safe communication, Alice and Bob will share their public key that has been generated using the value of generator g, n and their own private keys. Eve might know their public key, but he cannot get their private keys from their public key because of the discrete logarithm formula. So it is so difficult for Eve to find out their private keys. So no one can decrypt their message content. This is how security requirements are achieved by Diffie-Hellman Key Exchange.

d.

The main problem of this Diffie-Hellman Key Exchange is **Man In The Middle (MITM)** attack. In this symmetric key encryption system, if Eve impersonates as Alice to Bob and impersonate as Bob to Alice then Alice and Bob have no idea that what is going on inside the network and they are sending messages to someone else. This is called Man In the Middle attack. If they are only using Diffie-Hellman Key Exchange, then this MITM attack could happen and compromises the confidentiality and integrity of message.

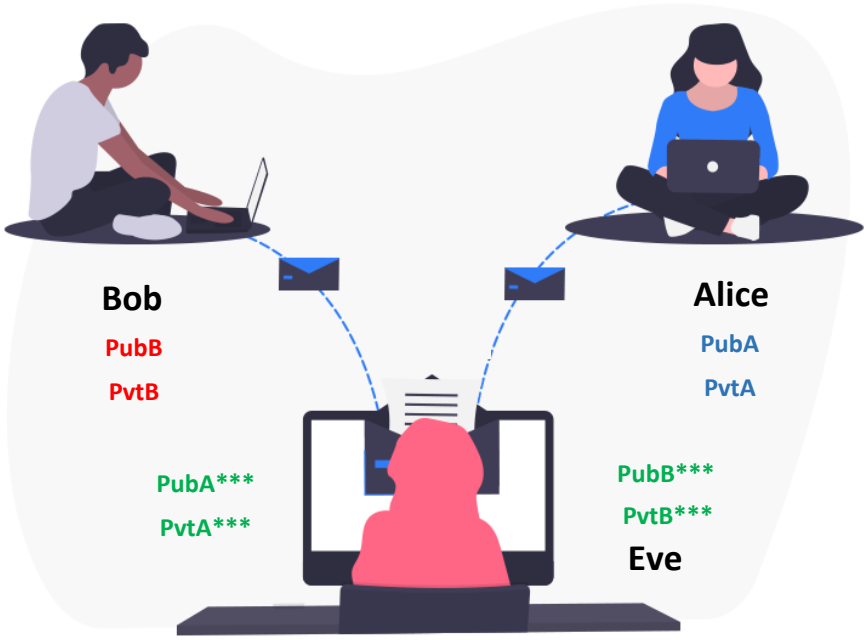


Figure 2: Man In The Middle (MITM) attack scenario

For instance, when Alice and Bob try to share their public keys, Eve eavesdrops in the network and takes their public key and generate two secret keys. Then he creates two private and two public keys and public keys share with Alice and Bob. Then Alice and Bob take the public keys and generate their secret keys and both of them start to communicate with Eve without knowing that they are communicating with a wrong person.

They start sharing their secret information, files, emails through the secret key and thinking these messages are safe because secret keys are only getting by them but in reality, all the messages are getting by Eve but Bob and Alice have no idea about that.

So, this type of MITM attack could be happened and message content confidentiality and integrity can be compromised if only Diffie-Hellman Key Exchange protocol is used alone.