Elijah Andrushenko

12/07/18

CPTS 427

Computer Security

Project 5

1.

A) p = 3, q=11, e=7, M=5

n = p\*q= 3 \* 11 -> **n = 33**

ϕ(n)=(p-1)(q-1)=2\*10->**ϕ(n)=20**

GCD(ϕ(n),e) = GCD(20, 7) = **1**

d=e^-1 % ϕ(n) = 1

7\*d % 20 = 1

**d=3**

public key -> (e,n) = (7,33)

private key->(d,n)=(3,33)

Encryption:

C = m^e % n=5^7 % 33->**C=14**

Decryption:

M = C^d % n = 14^3 % 33->**M=5**

B) p=5 q=11, e=3, M=9

n =p\*q->**n=55**

ϕ(n)==(p-1)(q-1)-> **ϕ(n)=40**

GCD(ϕ(n),e) = 1

d=e^-1 % ϕ(n)-> **d=27**

Encryption:

C=M^e % n->**C=14**

Decryption:

M = C^d % n->**M=9**

C) p=7, q=11, e=17, M=8

**n=77**

**ϕ(n)=60**

GCD(60, 17)=1

d \* 17 % 60 = 1

17d % 60 = 1

**d = 53**

Encryption:

**C = 57**

Decryption:

**M = 8**

2.

q = 11, a = 2

A) user A Ya = 9

Ya = a^Xa

9 = 2^Xa

**Xa = 3**

B) Yb = 3

Shared key = Yb ^ Xa = 3 ^ 3

**Shared key = 9**

3.

No it is not secure, an attacker could calculate m^e % n, for all 26 values of m. Then the attacker can make a table where the decryption of m^e % n = m.

4.

A) No, proof would be if we supposed C1 is corrupted. The output block p# is only dependent on the input blocks C2 and C3.

B) errors in P1 would affect C1, being that C1 is the input of C2, now C2 is also affected. The pattern would continue making all cipher text blocks affected. Once the decryption algorithm receives it, then the plaintext is corrected for all blocks except the original one that had the error. As a result the effect of the error is in the decrypted plaintext block.

5.

A) If an attacker knows that Ci = Cj, then the attacker can figure out that, Ci-1 ⊕ Cj-1 = Pi ⊕ Pj. Therefore, the attacker can solve Pi⊕Pj and get information about the two plaintext blocks.

B) 10110111

C) 128 bits = 16 bytes

255 \* 16 = **4080 attempts**

6.

A) Common Name(CN): wsu.edu

B) 2048 bits, this would be considered the bare minimum for security in today’s standards.

C) TLSv1, so version 1

D)Cipher: DHE-RSA-AES256-SHA

DHE provides the basis for authenticated protocols and provides forward secrecy to in Transport Layer Security’s ephemeral nodes.

RSA provides the basis for the cryptosystem and enables public key encryption to secure sensitive data when it is being sent over the internet.

AES256 provides a symmetric key algorithm, allowing encryption and decryption data via the same key.

SHA provides a cryptographic has function, it is used for storing passwords or other sensitive data

E) Master-Key Value: E2F2B13463151CDF68BC3CF397BB1216956FADD6C0C21944D0F85799134EACD29DC950C54CB9946&6124D944377992D5

7.

B) Subject is the Common Name which is wsu.edu

C) Issuer is Let’s Encrypt Authority X3, Let’s Encrypt Authority was no found in /etc/ssl/certs therefore it must not be trusted

D)                 Public-Key: (2048 bit)  
                Modulus:  
                    00:ba:f4:86:ac:55:6f:10:44:e3:41:4a:b8:5b:d1:  
                    a8:d9:49:5a:8e:7d:ab:da:ab:56:8e:4d:8f:82:a4:  
                    9c:d4:13:a4:0e:56:b0:d8:27:9e:3c:97:c1:d7:91:  
                    f7:17:61:e0:fc:9b:cb:38:7b:3d:13:2c:f9:49:1a:  
                    bb:ce:44:de:d0:bf:11:fd:0a:39:8f:30:4d:f2:99:  
                    1a:c7:aa:85:77:c7:5a:e6:59:97:78:ac:2c:30:72:  
                    fc:6a:b6:38:a9:6f:3b:98:34:5f:a5:ee:24:c7:73:  
                    e1:39:11:f8:9e:19:d3:ac:3e:38:42:11:ad:4c:e5:  
                    dd:6d:3c:a1:cc:98:a9:72:0b:82:d6:69:24:28:b5:  
                    3f:a8:7b:2c:4c:09:e9:b7:4b:06:bc:12:a8:0e:80:  
                    eb:b9:01:8b:0f:81:1f:86:ed:f9:16:63:3a:cc:ea:  
                    bf:9b:d7:95:3e:1b:a0:e8:ce:24:e8:4d:42:63:06:  
                    40:60:bd:a8:96:1e:98:95:3e:56:0d:7d:c9:a2:87:  
                    dc:37:48:b9:d7:81:0b:c6:c6:14:8f:a5:e7:3b:13:  
                    30:14:e5:f7:a2:2b:6a:a4:48:05:69:dc:c7:d5:22:  
                    6e:e0:d5:9a:cc:e9:5b:b3:42:47:30:e6:ce:ec:a5:  
                    d4:45:0e:8c:cb:81:6b:6c:ab:01:26:05:2e:48:d1:  
                    8e:cf  
                Exponent: 65537 (0x10001)

E) October 16 2018, no it is no longer valid

8.

A) SSLv2 SSLv3 and TLSv1 were all found on the man page but when running

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –tls1 WORKED

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –ssl2 COMMAND DOESN’T EXIST

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –ssl3 ERROR

In conclusion, TLSv1 is supported

SSLv2 is not supported

SSLv3 is not supported

B)

ECDHE-RSA-NULL-SHA

Openssl ciphers | grep ECDHE-RSA-NULL-SHA wasn’t found

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –cipher -ECDHE-RSA-NULL-SHA

1-Not supported

2-NULL cipher says to not encrypt data, which cancels the protection of SSL/TLS

ECDHE-RSA-RC4-SHA

Openssl ciphers | grep ECDHE-RSA-RC4-SHA was found

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –cipher -ECDHE-RSA-RC4-SHA

1-Not supported

2-RC4 is weak to several attacks many famous ones being, “Bar Mitzvah attack” or RC4 NOMORE attack which can decrypt RC4 relatively quickly

ECDHE-RSA-AES256-SHA

Openssl ciphers | grep ECDHE-RSA-AES256-SHA was found

Openssl s\_client –connect [www.wsu.edu:443](http://www.wsu.edu:443) –cipher - ECDHE-RSA-AES256-SHA

1-Supported

2- Hard to say if there is a weakness? Maybe when Quantum Computers come out AES256 will be useless. Otherwise no complaints.