Homework 2:

Your Name:**Pramod kumar, pjk5502@psu.edu**

Chapter 3: (38pts)

Problem 3 (8pts),

Part A

```
Planin text (P):                              10110110
Let cyper be(C):                              01100101
----------------------------------------------------------------------------------
Key stream will be(XOr of P&C)                 11010011
```

Since trudy knows Plain text and he can capture C between alice and bob. Using XOR he can get keystream.

Part B

Since trudy had keystream now, he can have a another text p′ which can be increpted using and key stream and Bob will be able to decrypt it sucessfully.

```
P′ = 11110000
K =  11010011
------------------------
C =   00100011
```

Which bob can decript

```
C =  00100011
K =  11010011
-------------------------
P’ =  11110000
```

Problem 5

SNO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
x	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1				
y	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
z	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

1st iteration:

Maj(1,0,1)
X steps & Z steps

SNO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
x	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0				
y	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
z	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0

Key = XOR(0,1,0)
=> 1

2nd iteration:

Maj(0,0,1)

SNO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
x	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1				
y	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	
z	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0

Keybit = XOR(1,1,0)
=> 0

3rd iteration:

X: 0001010101010101010
Y: 1011001100110011001100
Z: 01111000011110000111100

Keybit = XOR (0 , 0 , 0)
=> 0

4th iteration
X: 0001010101010101010
Y: 0101100110011001100110
Z: 10111100001111000011110
Keybit = XOR(0 , 0 , 0)
=> 0

5th
X: 0000101010101010101
Y: 1010110011001100110011
Z: 10111100001111000011110
Keybit = XOR(1 , 1 , 0)

=> 0

6th

X: 0000010101010101010

Y: 1010110011001100110011

Z: 01011110000111100001111

Keybit = XOR(0 , 1 , 1)

=> 0

7th

X: 0000001010101010101

Y: 0101011001100110011001

Z: 10101111000011110000111

Keybit = XOR(1 , 1 , 1)

=> 1

8th

X: 0000000101010101010

Y: 1010101100110011001100

Z: 10101111000011110000111

Keybit = XOR(0 , 0 , 1)

=> 1

X & Y & Z after 8 iteration

X: 0000000010101010101

Y: 1010101100110011001100

Z: 01010111100001111000011

Problem 14:

a) 64 bit

b) 64 bit

c) Effective length is 56, other wise 64(8 bits used for parity)

d) 48bit

e) 16

f) 8 Sbox

g) 48
h) 32

Problem 27(8pts),

CBC support random read access, as we need previous block cyper text to decrypt the next block. Disadvantage of CBC over CTR is writing a block will require to rewriting all the following blocks but in CTR we just need to know the offset from the base pointer and IV.

Problem 39.a(assuming k1!=k2) (6pts).

Yes, bob will detect the tampering, When Bob decrypts using K_2 , he gets: IV, P 0 , X p , P 2 , P 3 , MAC. Then, Bob uses K 1 to verify the integrity and calculate "MAC" ≠ MAC Hence, Bob knows that the integrity of the message is broken.

Chapter 4: (39pts)

Problem 2(9pts),

a) Before bob verfies the certificate, he doesn't know anything about the sender as certificate will be encrypted and only with CA public key it can be opened
b) Bob will verify the signate by using the CA public vertificate installed in the system/browser.
c) We know following details:
   1) Country
   2) Name
   3) City,state
   4) FQDN – FULL DOMAIN NAME
   5) EMAIL ADDRESS
   6) CA authurizer

Problem 6(10pts),

a) N = 33, e =3
   M = 19
   19^3 mod 33 = 6859 mod 33
   = 28
   Decrypt = 28^7 mod 33
   = 19
b) Allice will digitally sign using private key:

a. 25^7 mod 33 => 31

Now bob received it , he will decrypt it using public key

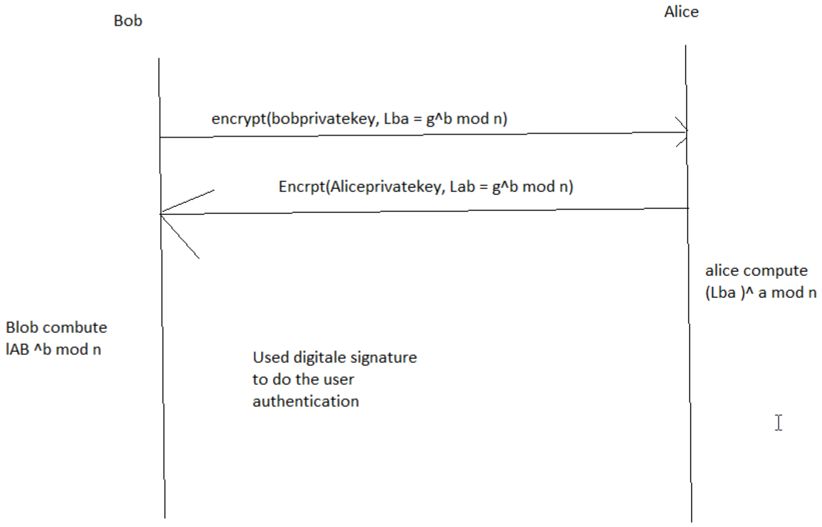
31^3 mode 33 =>25

So bob will match M with the result above, both will match

Problem 12(10pts),

Diffie-Hellman is key exchage algorithm, it doesn't provide user authentication. We can use various other crypto(Digital signature) provide user authentication which can be used while doing diffi-hellmen exchanges.

If Bob and alice share a symmetric



Problem 15 (5pts),

MAC is used to provide data integrity but to do so we don't need publi-private key concept, we just calculate hash, which anyone can do. While in digital signature, each key is bind to specific user. As non-one other the key holder has the private key so it is non-repudiated while hash is not binded to user.

Problem 25(5pts)

As Alice will compute as = g^(abt) mod P and bob will compute same. This way they can have a same pair but trudy only knows g^a mod p and P, as trudy don't know "a", since diffie hellman security lies on difficulty of factorigation, find "a" will be very difficult. So attack will not succed.

g^a + g^t != g^(at) , it will be NP complete tp find "a" and then multiple with "t".

Additional Questions:

1. (6) Consider a Diffie-Hellman scheme with a common prime *p=11* and a primitive root *g=2*.
    a) If user A has a private key of 3, what is A's public key?
    b) If user B has a private key of 5, what is its public key?
    c) What is their shared secret key K?

    Solution:

    a) 2^3 mod 11 = 8
    b) 2^5 mod 11 = 10
    c) A = 10^3 mod 11 => 10
    B = 8^5 mod 11 => 10
    Secret key K is 10

2. (5 pts) Perform both encryption and decryption using the RSA algorithm for the following: p=3, q=13, e=5, and M=3. (you may guess d, you may use the repeated squaring based power reduction technique we introduced in the class)

Solution:

N = p*q = 3*13 = 39
O = 3-1 * 13-1 => 2*12 = 24
e = 5

ed = 1 mod 24
5*5
Let d be 5

Message is 3

Encryption : 3^ 5 mod 39 = 3^2. 3^3 mod 39
((3^2)mod 39 . 3^3 mod 39) mod 39
6. 9 mod 39
= 9

Decryption: 9^5 mod 39
((9^2 mod 39)*(9^2 mod 39)* 9mod 39) mod 39
= 3

3. (6pts) Let h1 and h2 be two hash functions. Show that if either h1 or h2 is collision resistant, then the hash function h(x) = h1(x) ||h2(x), is collision resistant. (here ``||" means concatenation)

Solution:
As H be concatenation of both has function and concatenation has property that any collision on H leads to collision on both H1 and H2 so if both are collision resistant then H is also collision resistant.
Eg SHA-512 || whirlpool

4. (6 pts) What is a PKI? What are the three trust models for PKI (explain their meanings and example in some details)?

Solution
PKI trust model:

1) Monoply model: One universily trusted organization which manage all these certificates called CA. if CA is compromised then whole pKI fails. Public key of CA embedded in all principal hardware/software
Example : verisign, godady ranked 3
2) Oligarchy : multiple CA , any certificate issued by authorized CA can be trusted. Currently all software comes with these CA public certificate for verification of https Less secure than monopoly model since total security compromised if any configured trust anchor is compromise. Another problem is rough trust achor. You may choose whichone to trust. Example swisssgn and 50 more
3) Anarchy model: where everyone is CA, then user can decide which CA to trust. We need to search a lot to find a trust worth path if we need to

trust a user. Trust chain is less relaiable and more vulnerable to fraud and Its not scalable.
Eg pretty good privacy