Computer Science COMP4670/8640

Assignment 2

Due: End of Sunday, November 3, 2019

Note

Some questions might be open-ended and therefore require your own opinions and/or creativity, as the answers can vary considerably.

Question 1 (12%)

- How many tests are necessary to break a DES encryption by brute force attack?

  Assuming a 56-bit key, 2^{56} keys should be tested for a brute force attack

- What is the expected number of tests to break a DES encryption by brute force attack?

  At least half of the keys are expected to be tested i.e. 2^{55}

- How many tests are necessary to break a 128-bit key AES encryption by brute force attack?

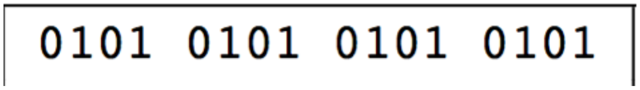
  Assuming a 128-bit key, 2^{128} keys should be tested for a brute force attack

- What is the expected number of tests to break a 128-bit key AES encryption by brute force attack?

  At least half of the keys are expected to be tested i.e. 2^{127}

Question 2 (12%)

As we saw in Chapter 2, there are four weak keys in DES. First explain why there are weak keys, and then explain one of them, all zeros, in complete detail. As you can see on slide 61 in this chapter, the first weak key is:



0101 0101 0101 0101

Thoroughly explain why this key is weak.

In every round we use a subkey, weak keys cause the encryption to act identically to the decryption, because they will produce sixteen identical subkeys.

The reason that the above key is considered as all-zero key is because we have one parity bit in every byte, and the above 64-bit key will be converted to a 56-bit key that all the bits are zero.

Question 3 (15%)

Using the RSA cryptosystem perform encryption and decryption for the following cases:

- ***p*** **= 5 ; *q* = 11 ; *e* = 3 ; *M* = 8 ; *d* = ? ; *C* = ?**

  n = p*q = 5 * 11= 55

  ϕ(n) = (p-1)*(q-1) = 4*10= 40 So, inverse of 3 mod 40 is d= 27

  Now, C= M^e mod n = 8^3 mod 55 C =17

  Verify: M= C^d mod n = 17^27 mod 55 = 8

- ***p*** **= 17 ; *q* = 31 ; *e* = 7 ; *M* = 32 ; *d* = ? ; *C* = ?**

  n = p*q = 17 * 31= 527

  ϕ(n) = (p-1)*(q-1) = 16*30= 480 So, inverse of 7 mod 480 is d= 343

  Now, C= M^e mod n = 32^7 mod 527 C = 280

  Verify: M= C^d mod n = 280^343 mod 527 = 32

- ***n*** **= 35 ; *e* = 5 ; *C* = 16 ; *d* = ? ; *M* = ?**

  n = 35, so let p= 5 and q= 7 ϕ(n) = (p-1)*(q-1) = 4*6= 24

  So, inverse of 5 mod 24 is d= 5

  Now, M= C^d mod n = 16^5 mod 35 So, M = 11

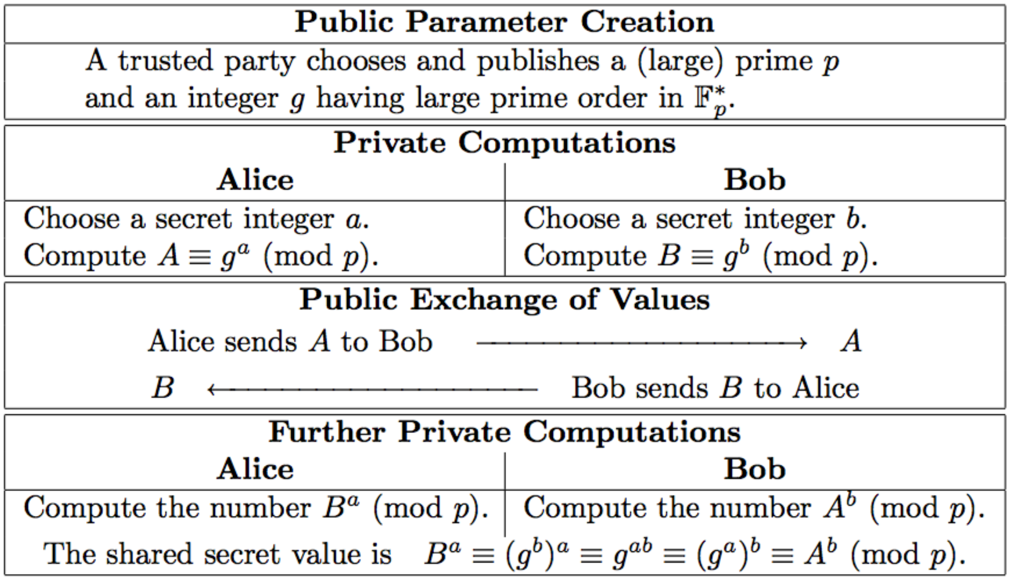
  Verify: C= M^e mod n = 11^5 mod 35 = 16

***M*: *message*; *e*: public key; *d*: private key; *C*: cipher**

Question 4 (14%)

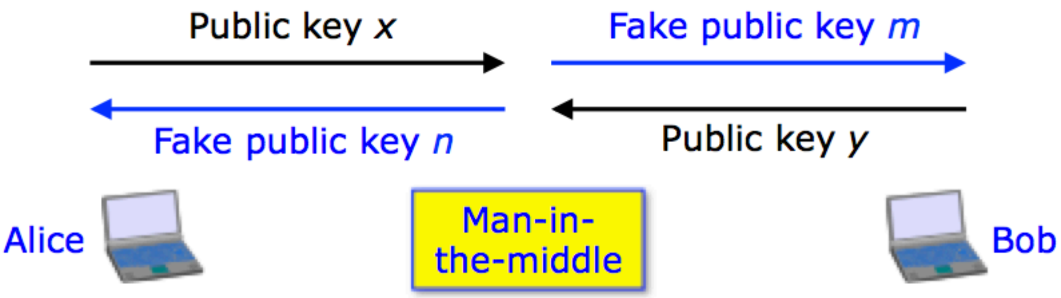
One of the initial tasks in establishing a secure channel between two parties is generating a secure key exchange. Diffie-Hellman key exchange is a solution for this task. However, it is vulnerable to a man-in-the-middle attack.

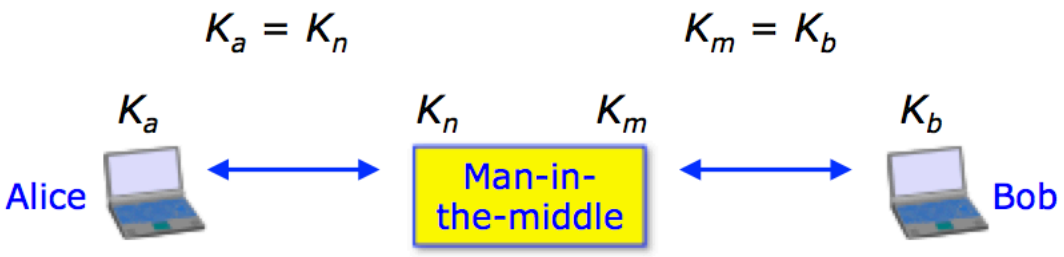
- Explain Diffie-Hellman key exchange, and show how it is vulnerable to man-in-the-middle attack?



Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private Computations	
Alice	**Bob**
Choose a secret integer a. Compute $A \equiv g^a \pmod{p}$.	Choose a secret integer b. Compute $B \equiv g^b \pmod{p}$.
Public Exchange of Values	
Alice sends A to Bob $\longrightarrow$	A
B $\longleftarrow$	Bob sends B to Alice
Further Private Computations	
Alice	**Bob**
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.	

An opponent Carol intercepts Alice's public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice. Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key. After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party. This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants.





- Provide an efficient solution for secure key exchange, which is secure against man-in-the-middle attack.

Station-to-Station (STS) protocol

STS protocol proceeds as follows. If a step cannot be completed, the protocol immediately stops. All exponentials are in the group specified by*p*.

1. Alice generates a random number *x* and computes and sends the exponential g^x to Bob.
2. Bob generates a random number *y* and computes the exponential g^y.
3. Bob computes the shared secret key $K = (g^x)^y$.
4. Bob concatenates the exponentials (g^y, g^x) (order is important), signs them using his asymmetric (secret) key *B*, and then encrypts the signature with *K*. He sends the ciphertext along with his own exponential g^y to Alice.
5. Alice computes the shared secret key $K = (g^y)^x$.
6. Alice decrypts and verifies Bob's signature using his asymmetric public key.
7. Alice concatenates the exponentials (g^x, g^y) (order is important), signs them using her asymmetric (secret) key *A*, and then encrypts the signature with *K*. She sends the ciphertext to Bob.
8. Bob decrypts and verifies Alice's signature using her asymmetric public key.

Alice and Bob are now mutually authenticated and have a shared secret. This secret, *K*, can then be used to encrypt further communication.

Question 5 (12%)

Using El-Gamal cryptosystem solve the following problems:

1. **Encrypt** the message $M = 18$ (i.e. find $C=(C_1, C_2)$) using the following parameters:
   $p = 89\ ;\ a = 11\ ;\ x = 8\ ;\ k = 7\ ;\ y = ?\ ;\ C = (?,?)$

$y=67$, $c_1=87$, $c_2=69$

Encryption : $r = a^k \bmod p$

$s = m \cdot y^k \bmod p$

Decryption : $m = s\,(r^x)^{-1} \bmod p$

y is public
x is private

Dig. Signature : $r = a^k \bmod p$

$s = k^{-1}(m - x \cdot r) \bmod p\text{-}1$

Verification: $(y^r \cdot r^s \bmod p) \overset{?}{=} (a^m \bmod p)$

***M*: *message*; *y*: public key; *x*: private key;**

***p*: prime; *k*: random; *C*: cipher**

2. Suppose, in the above scenario, you want to **sign** the message $M = 25$. What would be the **signature** $C=(C_1, C_2)$ for the message M ?

$k^{-1} \bmod p-1 = 63$ $c_1=87$, $c_2=55$

3. **Verify** that the above signature you have found is correct.

(Compute ($y^r * r^s \bmod p$) and ($a^m \bmod p$).)

$y^r \bmod p = 4$, $r^s \bmod p = 88$, $a^m \bmod p = 85$

=> $(y^r * r^s \bmod p) = (a^m \bmod p)$

Question 6 (5%)

Five parties have received secret shares, using **Shamir Secret Sharing** scheme, such that at least three of them are required to collaborate and reconstruct the secret. Using the following shares belong to three participants find the secret value.

(2, 29) , (4, 51) , (5, 32)

Please refer to the related slides in Chapter 2 as a sample solution with different values.

Question 7 (10%)

Suppose you want to encrypt some numbers **not greater than 63** (6 binary digits), using **Merkle–Hellman Knapsack** cryptosystem. Answer the following questions, by showing your work.

- Create a **private key** for this cryptosystem (a **super-increasing knapsack**). Call it ***S***.

$$S = [1,2,5,9,18,37] \qquad \sum_{i=1}^{6} S_i = 72$$

- Select a number, ***n***, as a modulus, and a multiplier, ***w***, such that ***GCD* (*n, w*) = 1**. (You need to compute $\boldsymbol{w^{-1}}$ **mod *n***)

$$n = 79 \qquad w = 5 \qquad w^{-1} = ?$$

5 * 16 = 80 = 1 mod 79 => w^{-1} mod 79 = 16

- Create a **public key** for this cryptosystem. Call it ***H***.

H = w . S mod n

H = [5,10,25,45,11,27]

- Using the public key, ***H***, **encrypt** the massage ***M*=41 (101001)** to generate cipher, ***c***.

C = M . H

C = [1,0,1,0,0,1] . [5,10,25,45,11,27] = 5+25+27 = 57

- Using the private key, ***S***, **decrypt** the cipher, ***c***, to get the original message, ***M***.

w^{-1} . C mod n = 16 . 57 mod 79 = 912 mod 79 = 43

Using S : 43 = 1 + 5 + 37 => M = [1,0,1,0,0,1] = 41

Bonus Question (5%)

The **basic RSA cryptosystem** is vulnerable to **Chosen-Ciphertext Attack**. This means if we have a ciphertext in hand, say c_1, corresponding to an **unknown** message, M_1, and then submit a **chosen** ciphertext, say c_2, and receive its corresponding message, M_2, then we are able to recover the unknown message, M_1. Clearly show how we can do this attack. Then, briefly explain how we can prevent this type of attack.

> **Note**: Assumption is that the attacker knows c_1, and is able to choose c_2, and receive M_2, such that $E(M_2) = c_2$. Attacker also knows the public key (e,n), but has no information about the private key d. Now, we would like to know how the attacker, having this info, can find M_1.

Attacker will choose $c_2 = c_1 * 2^e$, and will submit to receive its corresponding message, M_2. Now, based on RSA, we have:

$$M_2 = {c_2}^d \; mod\; n = (c_1 * 2^e)^d \; mod\; n$$

$$= \; ({c_1}^d * 2^{e*d}) \; mod\; n = ({c_1}^d * 2^1) \; mod\; n$$

$$= (2 * {c_1}^d) \; mod\; n = (2 * M_1) \; mod\; n$$

$$\Rightarrow \; M_1 = 2^{-1} * M_2$$