CODING TECHNIQUES

Exam & Exercises

MARIO MUÑOZ ORGANERO
FRANCISCO VALERA PINTOR
IVÁN VIDAL FERNÁNDEZ

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática

EXAM 1

Question 1

In 1993, T. AND. C. Woo and S. S. Lam presented the following protocol:

1. ***A→B:*** A
2. ***B→A:*** N_B
3. ***A→B:*** $K_{AS}[N_B]$
4. ***B→S:*** $K_{BS}[A, K_{AS}[N_B]]$
5. ***S→B:*** $K_{BS}[N_B]$

a) Describe the operation of the protocol step by step, explain in detail the purpose of each step and what is the purpose of all parameters exchanged at each step.

Unidirectional authentication protocol based on a trusted third party. The use of the parameters has been seen in the theory classes.

b) ***In step number 4, B sends to S '$K_{AS}[N_B]$'. Explain why this is done and why it would not suffice sending only 'N_B'.***

Not enough to send Nb – S would have nothing to verify.

c) One of the weaknesses of this protocol is that an adversary M (suppose that M can do interception, fabrication, modification and interruption of messages and this is known by S) is able to impersonate A so that A doesn't get to know this. Show how this is possible.

1. ***M→B:*** A
2. ***M→B:*** M
3. ***B→A:*** N_B
4. ***B→M:*** N_B'
5. ***M→B:*** $K_{MS}[N_B]$
6. ***M→B:*** $K_{MS}[N_B]$
7. ***B→S:*** $K_{BS}[A, K_{MS}[N_B]]$
8. ***B→S:*** $K_{BS}[M, K_{MS}[N_B]]$
9. ***S→B:*** $K_{BS}[N_B"]$
10. ***S→B:*** $K_{BS}[N_B]$

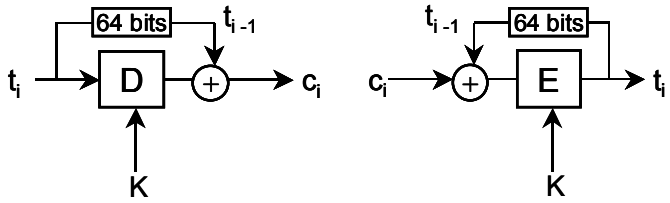
d) Explain how the problem can be solved in the simplest possible way.

It would be enough for the return message from S (4) to also include the identity of the person whose key was used.

Question 2

In some protocols like Kerberos, DES modes of operation are utilized with slight variations (e.g., PCBC). In this problem we are going to evaluate another possible variation of CBC.

a) Draw the cryptosystem that is utilized in CBC, but exchange the two different schemes on the sender and the receiver, so that the sender uses the decryption scheme and the receiver the encryption scheme. Explain if this cryptosystem is valid. If not, propose a mechanism that is valid.



It is valid.

b) Suppose a bit error has occurred during the transmission of a block using in the cryptosystem of part a) that you have validated:

b-1. Evaluate the integrity that the system is capable to offer.

b-2. If an alphabet with symbols of 64 bits is utilized, how many letters are affected in the output of the decryption?

After the erroneous block, every subsequent block will be deciphered incorrectly (C1 wrong implies T1 mangled since C1 goes thro E to get T1. T1 XOR'd with C2 so this input to E to get T2 is completely mangled etc.)

c) In case of the PCBC mode of operation of Kerberos, it is possible to interchange two consecutive blocks so that this change does not affect the remainder of the decryption process (i.e., there is not propagation of changes). Explain in detail if the same holds for the cryptosystem that you have validated before.

In this case, unlike PCBC, swapping two blocks will mangle all subsequent blocks – never recover synchronization.

EXAM 2

Question 1

A computer M has captured from the network a sequence of bits (C) from a sender A to a receiver B and knows that they have been coded using RSA. The decimal number that represents this sequence of bits is 10. M doesn´t have access to the directory of public keys, but knows that the key K={d,n}={5, 95} is the public key of one of the parties. You are asked to:

a) Explain what the possible objectives of the sender might have been and, in each case, to whom the public key would belong (A or B).

(Authentication, K_A), (Confidentiality,K_B), (Authentication, K_B)=(Confidentiality,K_A)

a) In each case above derive the original message, showing what M would need to calculate to be able to do this.

(Authentication, K_A)
trivial case
Use this public key to decrypt, i.e. m = c^d mod n = 10^5 mod 95 = 60
M=60

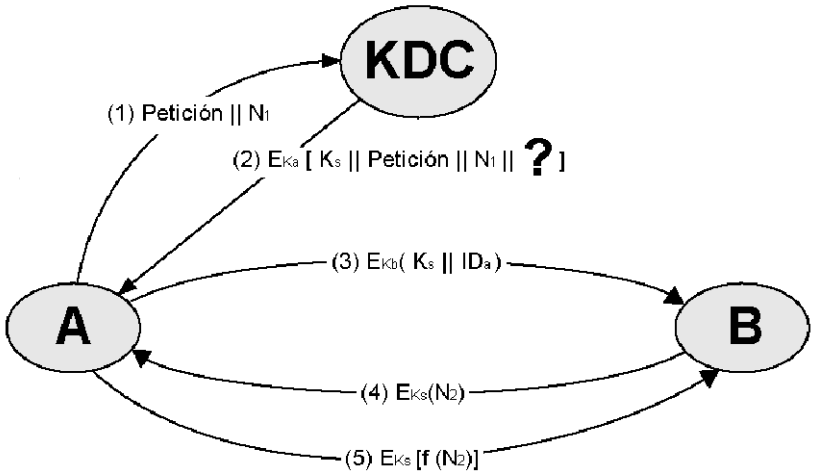
(Confidentiality,K_B)
Now B´s public key was used to encrypt so we must calculate his private key to decrypt.
Need {5^-1 mod phi(n), n}
Spot that 95 = 5*19 = p*q so phi = (p-1)(q-1)=4*18=72.
Now use Euclid to get 5^-1 mod 72.

72	0	1
5	1	0
2	-14	1
1	29	-2

KB = {29,95} and M=10^29 mod 95 =90
K_R={5,29} y M=90

Question 2

The following figure shows a well-known security mechanism:

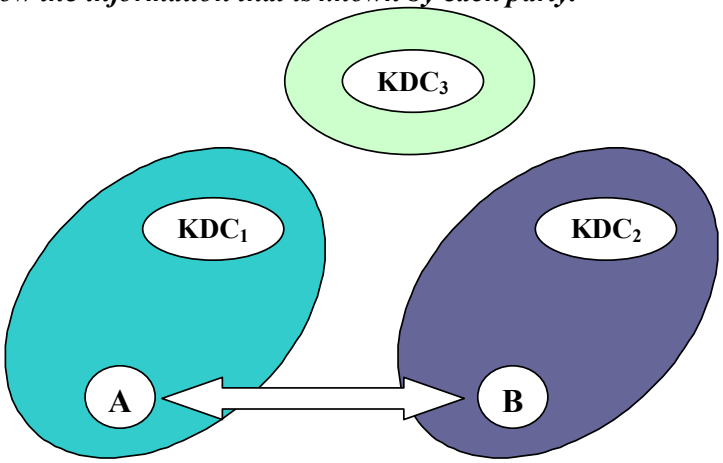


a) Replace the '?' in the second message with its value, indicate what purpose the mechanism serves and comment on the purpose of each of the messages exchanged

It´s a protocol for authentication arbitrated by a KDC (central key distribution). The '?' needs to be $E_{KB}[K_S||ID_A]$.

b) Generalise the previous case to adapt it to the security architecture shown in the following figure.

1. ***Indicate the messages exchanged between the different parties to complete the mechanism assuming that A and B want to establish communication***
2. ***Also show the information that is known by each party.***



1. $A \rightarrow KDC_1$: Pet_1, N_1
2. $KDC_1 \rightarrow KDC_3$: Pet_2, N_2
3. $KDC_3 \rightarrow KDC_1$: $E_{Kkdc1}[K_X, Pet_2, N_2, E_{Kkdc2}[K_X, ID_{KDC1}]]$
4. $KDC_1 \rightarrow KDC_2$: $E_{Kkdc2}[K_X, ID_{KDC1}]$
5. $KDC_2 \rightarrow KDC_1$: $E_{KX}[N_3]$
6. $KDC_1 \rightarrow KDC_2$: $E_{KX}[f(N_3)]$
7. $KDC_1 \rightarrow KDC_2$: $E_{KX}[K_S, ID_A, ID_B]$
8. $KDC_2 \rightarrow KDC_1$: $E_{KX}[E_{KB}[K_S, ID_A]]$
9. $KDC_1 \rightarrow A$: $E_{KA}[K_S, Pet_1, N_1, E_{KB}[K_S, ID_A]]$
10. $A \rightarrow B$: $E_{KB}[K_S, ID_A]$
11. $B \rightarrow A$: $E_{KS}[N_4]$
12. $A \rightarrow B$: $E_{KS}[f(N_4)]$

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática

EXAM 3

Question 1

Suppose that A and B share a key 'k'. To generate a MAC they decide to use a scheme in which 32 parity bits are computed from the message M (M is divided, assuming it doesn´t need padding, into 32 pieces and one parity bit is obtained from each of the pieces) and the number 'k' is the concatenation of the resulting bits.

- **a)** ***Say whether the procedure seems secure to you or not.***
- **b)** ***Explain with reasons if there is any difference between a MAC and a digital signature. Define both concepts.***
- **c)** ***Is it possible to use a digital signature as a MAC? What about the reverse?***
- **d)** ***Is it possible to use DES to generate digital signatures? If so, give an example and if not explain with reasons why not.***

The mechanism is not secure: for a given MAC it is very easy to construct another message with the same MAC and so it is possible to replace the given message with another without the exchange being detected. The opponent doesn´t have to calculate the MAC but each of the 32 parity bits from the message. The only thing he has to do to replace M is to construct a message that respects the calculated parity, something that is trivial to do.

Question 2

In all the machines in an organisation, intrusions have been detected and it is suspected that the attacks have been made by means of a "precomputed dictionary" on the system password file: they store in a file the result of processing all the words in the dictionary (as if they were user keys) and compare the file with the file of keys.

Knowing that the access mechanism is based on that of Unix (DES coding+salt):

- **a)** ***Draw schematically how the following attacks would be made:***
  1. ***Dictionary attack without precomputation (like the one done in the practicals)***
  2. ***Precomputed dictionary attack.***

  See Lecture Notes

- **b)** ***Compare both attacks discussing advantages and disadvantages of both***

  See Lecture Notes

- **c)** ***Assuming that the user keys are constructed from an alphabet of 'n' symbols, evaluate which of the following mechanisms is best suited to defend against a precomputed dictionary attack:***
  1. ***Keys of 6 characters with salts of 6 bits***
  2. ***Keys of 4 characters with salts of 10 bits***

  This answer seems to come from using a dictionary of all possible words, thus:
  6 chars and a 6 bit salt gives a total possibility of n^6*2^6
  4 chars and a 10 bit salt gives total possibility of
  n^4*2^10
  The former is greater when n>4 (easy)

Question 3

The following protocol (known as the SPLICE/AS protocol), will be the subject of study in this problem:

1. $A \rightarrow C: A,B,N_1$
2. $C \rightarrow A: C,K_{RC}[C,A,N_1,K_{UB}]$
3. $A \rightarrow B: A,B,K_{RA}[A,T,K_{UB}[N_2]]$
4. $B \rightarrow C: B,A,N_3$
5. $C \rightarrow B: C,K_{RC}[C,B,N_3,K_{UA}]$
6. $B \rightarrow A: B,A, K_{UA}[B,N_2+1]$

a) ***Describe the objective of the protocol, giving the purpose of each message. Describe what the utility of T is in message 3 and if it could be eliminated from this message without compromising the security of the protocol.***

b) ***In 1995 two simple attacks on the above protocol were proposed. The first was an impersonation of A to B and the second an impersonation of B to A (each one with the identities in the protocol). Show how it is possible to carry these out.***

c) ***Later a small alteration of the protocol was proposed, varying the messages that C sends:***

2. $C \rightarrow A:$ $C,K_{RC}[C,A,N_1,B,K_{UB}]$

5. $C \rightarrow B:$ $C,K_{RC}[C,B,N_3,A,K_{UA}]$

Show how it is still possible to impersonate B to A in spite of the suggested modification.

Mutual authentication arbitrated by a trusted third party.

Suplantación de A:

1.	M→C:	M,B,N_1	
2.	C→M:	$C,K_{RC}[C,M,N_1,K_{UB}]$	(obtención de K_{UB})
3.	M(A)→B:	$A,B,K_{RM}[A,T,K_{UB}[N_2]]$	(spoofing the A)
4.	B→M(C):	B,A,N_3	(intercepción)
5.	M(B)→C:	B,M,N_3	(spoofing the B)
6.	C→B:	$C,K_{RC}[C,B,N_3,K_{UM}]$	
7.	B→M(A):	$B,A, K_{UM}[B,N_2+1]$	(intercepción)

Suplantación de B:

1.	A→M(C):	A,B,N_1	(intercepción)
2.	M(A)→C:	A,M,N_1	(spoofing the A)
3.	C→A:	$C,K_{RC}[C,A,N_1,K_{UM}]$	
4.	A→M(B):	$A,B,K_{RA}[A,T,K_{UM}[N_2]]$	(intercepción)
5.	M→C:	M,A,N_3	
6.	C→M:	$C,K_{RC}[C,M,N_3,K_{UA}]$	(obtención de K_{UB})
7.	M(B)→A:	$B,A, K_{UA}[B,N_2+1]$	(spoofing the B)

Suplantación tras modificación:

3.	A→M(B):	$A,B,K_{RA}[A,T,K_{UB}[N_2]]$
4.	M→B:	$M,B,K_{RM}[M,T,K_{UB}[N_2]]$
6.	B→M:	$B,M, K_{UM}[B,N_2+1]$
7.	M(B)→A:	$B,A, K_{UA}[B,N_2+1]$

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática

EXAM 4

Question 1

The following messages, ordered n a suitable way, form a well-known security protocol between three parties X, Y and Z.

1. $K_S[N_2]$
2. X,Y,N_1
3. $K_{ZY}[K_S,X]$
4. $K_{ZX}[K_S, X, Y, N_1, K_{ZY}[K_S,X]]$
5. $K_S[f(N_2)]$

a) ***Order the messages indicating from whom it was sent and to whom and explain the objective of the protocol and the purpose of each message.***

Needham-Schroeder.
X t oZ 1. X,Y,N_1
Z to X 2. K_{ZX} [K_S, X, Y, N_1, K_{ZY}[K_S,X]]
X to Y 3. K_{ZY} [K_S,X]
Y to X 4. $K_S[N_2]$
X to Y 5. K_S[f(N_2)]

b) ***From the point of view of authentication, explain with reasons what entities are authenticated with respect to whom***

↓ authenticated w.r.t. →	**X**	**Y**	**Z**
X	-	Yes	NO
Y	NO	-	NO
Z	Yes	NO	-

c) ***In message number 4:***
- ***1) Explain with reasons why it sent and if it would be possible to do without X.***
- ***2) Would it be possible to do without Y?***
- ***3) Would it be possible to do without N_1?***

NOTE:If any parameter is considered essential, give an example of an attack that could be made against the protocol in the absence of the said parameter.

1) It´s possible to omit X (only X can decipher this message anyway)
2) It´s not possible to omit Y (needed to distinguish multiple requests).
3) It´s not possible to omit N_1 (to ensure talking to KDC, not a replay with an old key of Bob´s – far fetched scenario)

d) ***The Kerberos protocol uses a modified version of this protocol, in which among other things, a timestamp is added at the end of message 3 ($K_{ZY}[K_S,X,T]$). Explain what attack is prevented by this change.***

It tries to avoid a replay attack against Y.

e) ***In message 5,***

1) ***Explain whether it is necessary to apply the function f (typically to add 1 to N2) or whether the number N2 would be sufficient.***
2) ***Explain with reasons whether this message is really necessary or whether message 1 alone would be sufficient to show that Ks had been correctly extracted.***

1) It is essential
2) It is essential

EXAM 5

Question 1

A secret key cryptosystem is being installed and evaluated that tries to use an algorithm similar to Rijndael coding (with key of length 256 bits and a block size o f128 bits).

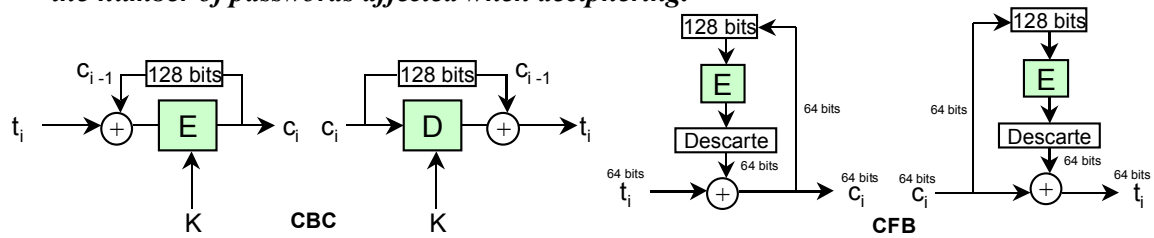
As regards the modes of operation, different options are being evaluated:

- ***Using CBC.***
- ***Using CFB with a block size of 64 bits.***

A string of bits representing 16 passwords of 8 symbols each is going to be transmitted, where each symbol is coded using 7 bit ASCII.

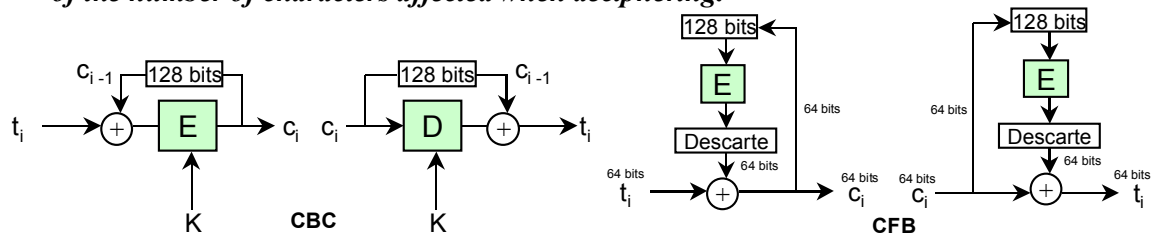
If a 1 bit error in the transmission of the text is introduced:

a) ***Indicate, giving reasons, which mode of operation is most suitable from the point of view of the number of passwords affected when deciphering.***



CFB is the most suitable

b) Indicate, giving reasons, which mode of operation is the most suitable from the point of view of the number of characters affected when deciphering.



CFB is the most suitable.

c) Indicate, giving reasons, the number of bits that will be deciphered incorrectly in each of the modes.

CBC: 64 + 1= 65 bits
CFB: 1 + 32 + 32= 65 bits

If it is known that during tranmission, strings of bits can be lost whose size can vary between 32, 63 or 128 bits,

d) ***Show with reasons which mode of operation is now the most convenient from the point of view of the number of letters affected when deciphering.***

CFB is most suitable.

It is desired to also use the algorithm to offer the possibility of digitally signing the messages that are sent.

e) If it is possible, indicate with reasons which of the two modes seem most suitable. If it is not possible explain with reasons why not.

AES is a symmetric coding algorithm. It can not be used for digital signatures.

Question 2

The members of a work group in Universidad Carlos III decide to create an environment that lets them exchange information with known integrity and that has provably been sent by a member of the group. For this they decide to use an HMAC in the following way:

- ***They send the message in the clear and add the result of applying the hash function to the message.***
- ***To calculate the hash they divide the message into 3 bit pieces.***
- ***They decide to work with the field GF(3^2) with the reduction rule t^2+1.***
- ***They calculate the hashing function according to Merkle in the form:***
  $H(i)=[H(i-1)\cdot m(i)]+m^1(i)+ m^2(i)$ where m(i) are the three bits at position i of the message (whose hash we are caculating in the field GF(3^2)), $m^1(i)$ are the three bits m(i) rotated right one positin and $m^2(i)$ rotated 2 positions.
- ***In order to map a list of bits into an element of GF(3^2)the following table is used:***

0	**0**
1	**1**
2	**2**
3	**t**
4	**2t**
5	**t+1**
6	**t+2**
7	**2t+1**

- ***They agree on H(0) which will be the shared system key for all users of the group.***
- ***The result of the hash will map to 4 bits taken from the inverse of the above table.***

a) Calculate the hash of the message that a user wants to send: m=AE7 if H(0)=t+1.

The message hash will be: 0101

b) Estimate the robustness of the method against collisions.

The output table for all inputs is:

0 -> 0
1 -> 0
2 -> 3
3 -> 2
4 -> 7
5 -> 6
6 -> 8
7 -> 5

Now the users of our group want to add confidentiality to their messages. For this they are going to use asymmetric cryptography based on public keys (following McEliece). They agree among themselves to generate the public keys in the following way:

- ***They want to send meesages m of length k.***
- ***A matrix G is chosen such that the code that results from the multiplication mG has n bits and is capable of correcting t errors. This means that if a user sends mG and someone adds a "1"to this in up to t random bit positions then we can still recover m without problems.***
- ***Each user invents a permutation matrix P of size nxn. This matrix satisfies the rule that if a vector v has t "1s", v*P will also have t "1s".***
- ***Each user invents a random matrix S of size kxk.***
- ***The public key of a user will be the matrix, G'=SGP and the number t.***

When a user wants to send a coded message to a receiver he uses the public key of the latter in the following manner:

- ***c=mG' + v where v will have a random number of "1s" between 0 and t in random positions.***

c) Demonstrate how the receiver will be able to recover the message.

Postmultiply by P^-1 to get
c -> $cP^{-1} = (mG' + v)\ P^{-1} = (mSGP + v)\ P^{-1} = (mS)G + vP^{-1}$
Now, P and hence P^-1 are simple permutations so vP^-1 will have the same number of "1s" as v, not more than t of them, so (mS)G can be recovered by the property of G. Hence we can get mS and then m (S and G are known by the receiver)

EXAM 6

Question 1

A coding application uses a policy of transparent padding to align the plain text to 24 octet words before proceeding to code the resulting text using DES (in any one of the modes of operation). The padding functions as follows:

If the length of the text is 'n' octets, the padding is p=24-(n mod 24) octets and each of those octets takes the value 'p'.

a) Explain with reasons the differences between the option of using this policy of padding and that of having no stuffing at all (suppose ECB is used as mode of operation)

Seen in theory classes and practicals.

b) Compare the modes CBC and PCBC from the point of view of the integrity of the message and indicate with reasons if there is any difference made by using the padding or if, on the contrary, the padding makes no real difference.

The generic comparison between CBC and PCBC has been seen in the theory classes and in the practicals.
The use of padding makes a decisive difference in the detection of the integrity of the plaintext.

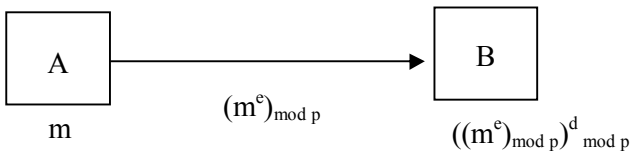
A string of 320 bits is going to be sent using the above padding scheme and it is known that the transmission will produce a change of one bit in the coded text.

c) Explain with reasons which mode of operation will give the receiver more possibility to detect that the message has lost integrity: with CBC or with CFB (with size 64 bits)

CBC will give more possibility of detecting the loss of integrity of the plain text.

Question 2

Pohlig and Hellman, shortly before the appearance of RSA, proposed a cryptographic system as follows:



where m is the message, p is a prime number and the system is such that B recovers the message sent by B with the operation shown in the figure.

Answer the following questions with reasons:

a) What conditions do d and e have to satisfy? Is is a public key system? Is non-repudiation possible? Justify your answers.

The product must be 1 mod Euler´s function. Also d and e must be relatively prime w.r.t Euler´s function (and must be less than that value). In this case Euler´s function has the value 16 (since p is prime, 17, all number less than it are relatively prime). It is not a public key system (in fact it is the main improvement of RSA over Pohlig Hellman). It is not possible to publish d nor y and p simultaneously because the calculation of one from the other is easy. Non-repudiation is not attainable because the sender and the receiver both know all the information of the crypto system.

b) Si p=17 y e=5 calcule d.

Use any of the methods shown in the class to get d=13.

c) What happens if we take e=2? Justify the answer by means of appropriate calculations.

If e=2 then it is not relativley prime to 16 and so the multiplicative inverse, d, would not exists.

d) ***Propose an implementation of the algorithm based on elliptic curves.***

where a*b is the order of the chosen elliptic curve. Re denotes the real part. M will be the point on the curve whose real part is the message.

EXAM 7

Question 1

Assume that an organization uses an access control mechanism based on the UNIX scheme (DES algorithm + SALT) and the passwords are stored using the typical format in the file `/etc/passwd`.

The first two lines of the password file are the following:

```
user1:P1:500:100:GECOS 1:/home/user1:/bin/bash
user2:P2:501:100:GECOS 2:/home/user2:/bin/bash
```

P_i are words composed of 13 characters. The first two characters are the SALT (random number used to modify the DES expansion phase) that is applied to the corresponding user and the other eleven characters are the result of processing the password with the typical authentication mechanism (DES+SALT).

a) Provide a detailed description about the password verification process for a user that wants to access the system.

See Lecture Notes

Let's have the following definitions:

- ***t_1, is the time that a password cracker is normally spending (using a brute force attack) to get the password of user 1, supposing the password file is only containing information about user 1.***
- ***t_2, is the time that a password cracker is normally spending (using a brute force attack) to get the password of user 2, supposing the password file is only containing information about user 2.***
- ***t_{12}, is the time that a password cracker is normally spending (using a brute force attack) to get the password of user 1 and 2, supposing the password file contains information about user 1 and 2.***

b) For each of the following cases, indicate with reasons if $t_1 + t_2$ is greater than, less than, or equal to t_{12}:
1) User 1 and user 2 have different SALTs.
2) User 1 and user 2 have the same SALT.

1) $t_{12} = t_1 + t_2$
2) $t_{12} < t_1 + t_2$

Assume that an opponent wants to use the 'pre-computation' technique in order to generate a word list so that it can be directly compared with every P_i to retrieve a password.

For this purpose, an opponent is using the typical P_i generation mechanism and applying it to all the possible user passwords that may be obtained using the characters of the alphabet under consideration, and storing the result into a file. That way, once the opponent has obtained the password file he or she just has to compare each P_i with the P_i obtained from his precomputed file.

Supposing that user passwords are built using an alphabet of 'n' symbols and supposing that in the authentication mechanism it is possible to change the size of the SALT:

c) Indicate with reasons which of the following options should be used in the system in order to protect it against a attack of precomputed dictionaries:

1) 5 character passwords with 8 bits SALT.
2) 6 character passwords with 6 bits SALT.

n<4: SALT 8 and password 5 is better, n>4: SALT 6 and password 6 is better, n=4: it is the same

Question 2

We want to design a digital signature system based on elliptic curves. For this purpose, we need to use asymmetric keys in the following way:

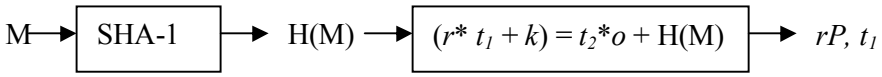
$$Kpub = (C, P, kP)$$
$$Kpriv = (k)$$

C: is an elliptic curve of order o, P a point on this curve, and k an integer of 256 bits.

In order to obtain the digital signature a hashing function shall be used first.

a) If SHA-1 is used, what are the implications with regard to the order of the curve?
Ans: the order of the curve must be more than 2^{160}.

The signature for a message M is calculated as follows:



With r being a random 256-bit integer. t_1, t_2 are computed such that the equation in the last box from the previous picture is true.

b) ***If the following information is sent (M, rP, t_1), demonstrate that every receiver having the public key from the sender is able to validate the signature.***

Ans: $H(M)*P = t_1*rP + kP$

In order to simplify our system we propose working with exponentiation algorithms instead of working with E.C.

c) ***Work out how the previous digital signature system should be implemented using El Gamal based on exponentiation with the following keys*:**

$$Kpub = (g, p, g^k \, modp)$$
$$Kpriv = (k)$$

Make: $(r* t_1 + k) = t_2*(p-1) + H(M)$

and send: $(M, g^r \, mod \, p, t_1)$

In order to validate this signature the receiver will need to raise the second parameter to the power t_1 and multiply the result by the public key $g^k \, modp$ verifying that it is equal to $g^{H(M)} \, mod \, p$.

Finally, use the finite field GF(2^3). The reduction rule shall be $t^3+t+1=0$.

d) Assume the following conversion table is used:

0	0
1	1
2	t
3	t+1
4	t^2
5	t^2+1
6	t^2+t
7	t^2+t+1

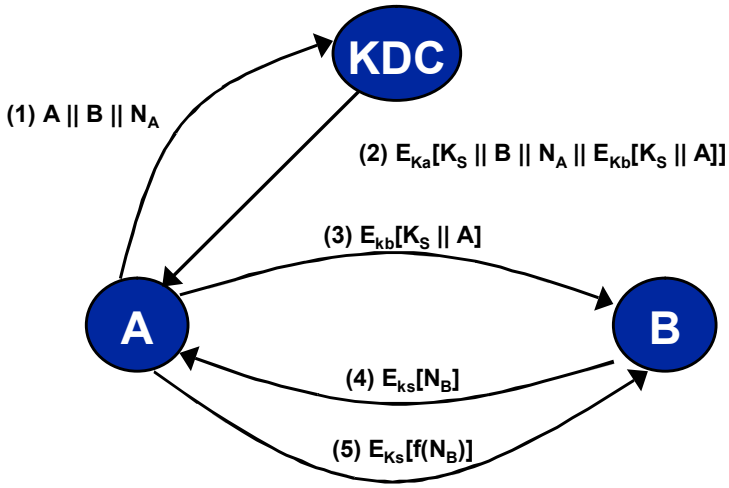
and instead of using El Gamal based on operation congruent to modulo p, we use the above finite field GF(2^3). Calculate the public key associated with k=2 and g=6.

Public key $g^k = 2$

EXAM 8

Question 1

The following picture shows a well-known security mechanism:



a) Describe the different parts and the purpose that they serve

b) In message (2), show the purpose of the parameters B and N_A. Show also, if it would be reasonable to leave them out.

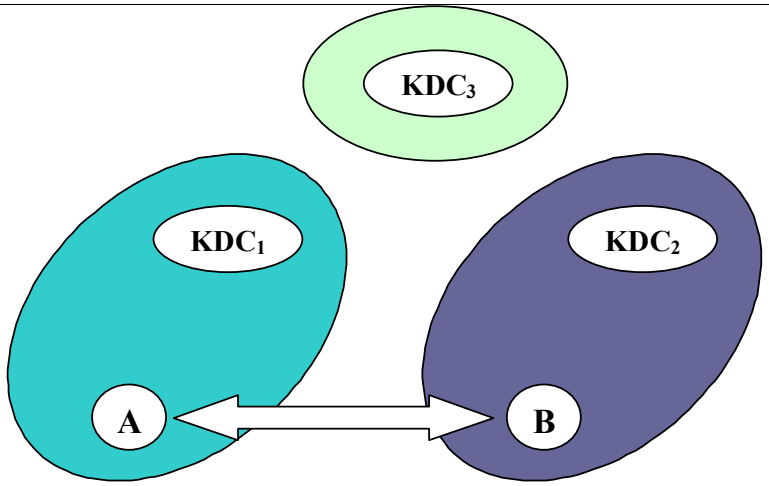
B is sent so that an attacker M cannot execute a man-in-the-middle attack between A and the KDC. This way, a message from the KDC could arrive at A with a ticket for M whereas A assumes the ticket is for B. N_A is necessary to avoid that an opponent obtains an old session key K_s and replays this message towards B to impersonate A.

c) Discuss with reasons the validity of the following statements:

1. ***After receiving the message (2), A knows that the KDC is who it claims to be.***
   A knows this because the KDC responds to the challenge N_A. Moreover, it knows that there is no man-in-the-middle attack since the message (2) contains the identifier B
2. ***After receiving the message (3), B knows that A is who it claims to be.***
   False since the message could be a replay
3. ***After receiving the message (4), A knows that B is who it claims to be.***
   False (only true if N_B contains some amount of redundancy)

d) Generalize the previous case and adapt it to the security architecture as shown in the following figure

1. ***Show the messages which the different parts exchange to complete the mechanism under the assumption that A and B want to establish a communication session.***
2. ***Show also what information each node is supposed to know.***



1. ***Show the messages which the different parts exchange to complete the mechanism under the assumption that A and B want to establish a communication session***.

1. $A \rightarrow KDC_1$: A, B, N_1
2. $KDC_1 \rightarrow KDC_3$: KDC_1, KDC_2, N_2
3. $KDC_3 \rightarrow KDC_1$: $E_{Kkdc1}[K_X, KDC_2, N_2, E_{Kkdc2}[K_X, KDC_1]]$
4. $KDC_1 \rightarrow KDC_2$: $E_{Kkdc2}[K_X, KDC_1]$
5. $KDC_2 \rightarrow KDC_1$: $E_{KX}[N_3]$
6. $KDC_1 \rightarrow KDC_2$: $E_{KX}[f(N_3)]$
7. $KDC_1 \rightarrow KDC_2$: $E_{KX}[K_S, A, B]$
8. $KDC_2 \rightarrow KDC_1$: $E_{KX}[B, E_{KB}[K_S, A]]$
9. $KDC_1 \rightarrow A$: $E_{KA}[K_S, B, N_1, E_{KB}[K_S, ID_A]]$
10. $A \rightarrow B$: $E_{KB}[K_S, A]$
11. $B \rightarrow A$: $E_{KS}[N_4]$
12. $A \rightarrow B$: $E_{KS}[f(N_4)]$

2. ***Show also what information each node is supposed to know.***

KDC_1: Shared secrets with the users in its domain and with KDC_3. Moreover, must know that B is in the domain of KDC_2
KDC_2: Shared secrets with the users in its domain and with KDC_3.
KDC_3: Shared secrets with all KDCs.
A, B: Shared secrets with their domain KDCs, respectively.

Question 2

a) ***We want to use RSA as mechanism to sign messages. A certain user of the system utilizes the public key n=55 and e=3. Compute the private key.***

The private key is d=27.

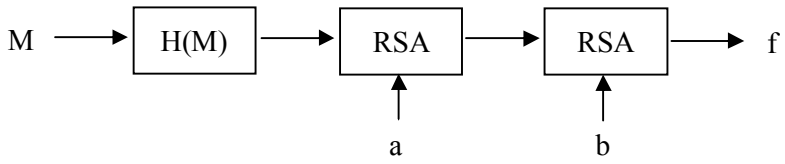
b) ***If we use the congruence operation modulo n as hash function, validate the following signature:***

(M, f) = (607, 18)

where M is the message and f the signature.

To calculate the signature: H(M) = 2 and 2^{27} mod 55 = 18

c) ***The previous system allows that each user can sign message with his/her private key. If an attacker compromises the machine of the user, where the keys are stored, the attacker could generate valid signatures and impersonate the identity of the original key owner. Such an attack is particularly harmful if the compromised machine is the certification authority of the system. In order to avoid this attack toward the CA, a mechanism called "threshold encryption" has been designed. This mechanism consists of decomposing the private key into two subkeys, so that two RSA operations are necessary (one for each subkey) if the CA signs a message. This signature scheme is, therefore, as shown in the following figure***:



This way, if an attacker can get one of the two subkeys, he/she cannot generate signature but must obtain also the other subkey.
Calculate b for the system of the first part if you know that a = 7.

The equation a * b * e = 1 mod 40 must be fulfilled. We know a and e in this case, b can be computed as b=21.

d) ***Propose a scheme equivalent to the previous one based on elliptic curves.***

It will be necessary to substitute the boxes for RSA by equivalent ones for elliptic curves. The hash value of the message will be associated with a point on the elliptic curve. The exponentiation is transformed into a product. Now a*b*e must be a multiple of the order of the elliptic curve.

EXAM 10

Question 1

The following messages are the modified version (by Gavin Lowe) of a well-known cryptographic protocol where the entities A, B and C are involved (being K_{UA} and K_{RA} the public and private key pair of A, etc.). However, it is obvious that the messages of the protocol are not ordered properly:

(1) A,B
(2) B,A
(3) $K_{UA}(N_A,N_B,B)$
(4) $K_{RC}(K_{UB},B)$
(5) $K_{UB}(N_B)$
(6) $K_{RC}(K_{UA},A)$
(7) $K_{UB}(N_A,A)$

1. Put all the messages in the right order, clearly explaining (a) the origin and the destination entity of each message (b) which is the purpose of the protocol (c) who is authenticated against whom once the protocol has been completed.

(A->C) A,B
(C->A) $K_{RC}(K_{UB},B)$
(A->B) $K_{UB}(N_A,A)$
(B->C) B,A
(C->B) $K_{RC}(K_{UA},A)$
(B->A) $K_{UA}(N_A,N_B,B)$
(A->B) $K_{UB}(N_B)$

The modification introduced by Lowe was the addition of parameter B in message (3) as it is shown in the previous list of messages.

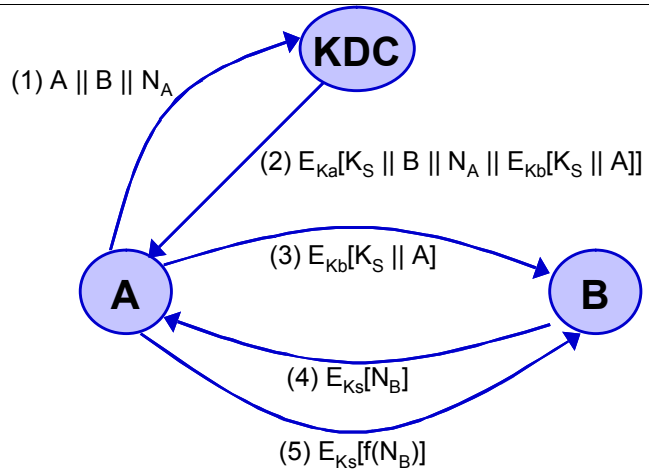
Although it is just a slight modification, this parameter B is mandatory in order to avoid an impersonation attack (A can be impersonated against B by 'M', provided that 'M' has a public key recognized by the system so that it is, therefore, able to talk with A and with B).

2. Explain how this attack can be performed and why introducing B can help to avoid it.

(A->M) $K_{UM}(N_A,A)$
(M->B) $K_{UB}(N_A,A)$
(B->M) $K_{UA}(N_A,N_B)$
(M->A) $K_{UA}(N_A,N_B)$
(A->M) $K_{UM}(N_B)$
(M->B) $K_{UB}(N_B)$

This protocol is the public-key version of another well-known protocol used for session key distribution using symmetric encryption based on a trusted third party.

3. Describe the message interchange that takes place in this protocol briefly explaining its most important flaws.



Question 2

Trying to optimize the number of messages in the IKE negotiation in IPsec, we propose using a single message for both the authentication and the hashing algorithm proposal. In order to do this, the first message sent in the new IKE negotiation is calculated by applying the proposed hashing algorithms (one after the other if there is more than one) to a shared secret. The other end also knows the shared secret so it will have to try one by one every possible combination of the hashing algorithms to this secret. In this problem we are going to assume that the number of hashing algorithms known to both the sender and the receiver is 3.

1. ***How many trials has the receiver to perform in the worst case in order to find the proposed hashing algorithms? Assume that all the hashing algorithms have the same output length.***

Hash functions in general do not fulfill the condition that $H_1(H_2) = H_2(H_1)$ which is why the order which the HASH functions are realizes is important. This way, the possible combinations for 3 hash algorithms are:

a. 3 with one algorithm.
b. 6 with 2 algorithms.
c. 6 with 3 algorithms.

which is why there are 15 possibilities to be tried in the receiver in total.
In case that not all hash functions would have the same output length, the latest hash function applied could be identified by looking at the output size, which would reduce the number of possible cases at the receiver.

In order to optimize the number of messages further, we add the D-H negotiation to the first message (and its reply) in the following way: The sender sends:

$$g^A \bmod p$$

where A is the concatenation of the results of the hashing algorithms applied to the shared secret. The receiver works out the proposed hashing algorithms, selects one and calculates B as the hash value of the shared secret with the selected algorithm. The receiver sends the following message back to the sender:

$$g^B \bmod p$$

The session key will be then:

$$g^{AB} \bmod p$$

where we assume that g and p are well-known to both ends in the communication.

2. ***Argue if the session key could have been calculated as A*B and if this would have any advantage over g^{AB} mod p in the particular case in which the emitter proposes only one hashing algorithm.***

In this case, the session key can be calculated as AB because the sender and the receiver have this information (which does not occur in a conventional D-H). Moreover, the key is secure since an attacker, who does not know the shared secret, can only try to access g^A and g^B with which he/she can calculate AB and which requires to calculate two inverses. This is an advantage of this method compared to calculate g^{AB} in which the complexity would be only half as high.
One of the possible problems we could find in case that only a single hash algorithm is used in the proposal of the sender because in this case A and B are the same. In this case, the complexity to break AB would be the same as for g^{AB}.

In the previous proposal there is still something missing: the proposal and selection of the encryption method. In our optimization of the IKE negotiation we propose that there is only one possibility for encryption so there is no need of negotiating this. We propose to use a secret cipher only known to the implementers of the modified IPsec stack (like A5 in GSM). We propose to use a modified version of ElGamal in order to make it symmetric. Each message m is encrypted like this:

$$m * g^{AB} g^{K} \bmod p = c$$

where K is an integer that is calculated so that the algorithm uses the same key for encryption and decryption and that AB is the session key. In other words, the method for decrypting the ciphertext c is:

$$m = c * g^{AB} \bmod p$$

3. ***If A=5, B=2, and p=7, calculate K and use it to encrypt the message m=2 if g=3. Verify that the algorithm works by decrypting the obtained ciphertext.***

The condition must be fulfilled that 2AB+K mod (p-1) = 0 from the Euler-Fermat theorem, since in this case any number g which is chosen as 2AB+K will give us unity and in this case to decrypt c we will finally obtain m.
In our case, we obtain that 2AB+K = n(p-1) generates the equation 20 + K = n*6. To find a K between 0 and (p-1), we set n=4 and obtain K=4.
To encrypt m=2 with g=3 we must calculate first 3^4 mod 7 and 3^{10} mod 7. Therefore, we can use the method for fast exponentiation shown in the class:

$$3^1 \bmod 7 = 3$$
$$3^2 \bmod 7 = 2$$
$$3^4 \bmod 7 = 4$$
$$3^8 \bmod 7 = 2$$
$$3^{10} \bmod 7 = (3^4 \bmod 7 * 3^4 \bmod 7) \bmod 7 = 4$$

to calculate c = 2 * 4 * 4 mod 7 = 4 and to finally calculate m = c * 4 mod 7 = 2.

4. ***Propose a modification of the former encryption and decryption algorithms in order to use elliptic curves.***

As shown in the theoretical session, products must be replaced by sums of points on the curve and the exponentiation must be replaced by the product of a scalar with a point of the curve. This way, the encryption will be M + AB*P + K*P with P a point on the curve and decryption will be C+ AB*P. To obtain the original message, the following condition must be fulfilled: 2AB+K = a multiple of the order of P.

EXAM 11

Question 1

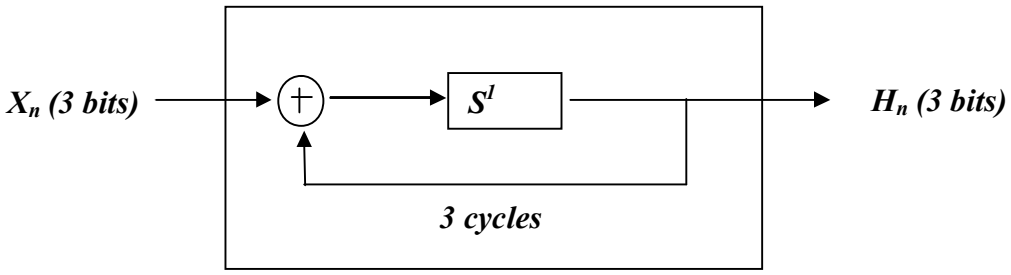
Answer the following questions about the modes of operation CBC, CFB and OFB used with the DES algorithm (the size of block which can be varied for the modes is 32 bits)

- ***a) A random sequence of bits is sent and noise on the line causes one of the "0"s to become a "1". When decoding the text, how many bits in total will be altered?***
- ***b) A sequence of bits is sent consisting of 32 symbols of 16 bits each and noise on the line causes two bits that were "1" to become "0": bit number 27 and bit number 470. en la línea hay dos bits que pasan de valer '1' a valer '0': el bit número 27 y el número 470. When decoding the text, how many characters in total will be altered?***
- ***c) A sequence of bits is sent consisting of 32 symbols of 16 bits each and noise on the line cuases symbols 5, 6 and 30 not to arrive. When decoding the text, how many characters in total will be altered?***

a) CBC: 33 bits (if it is in the last block, 32), CFB: 33 bits (if it is in the last block, 1 bit, if in the penultimate 17), OFB: 1 bit
b) CBC: 9 symbols, CFB: 8 symbols, OFB: 2 symbols
c) CBC: 28 symbols, CFB: 9 symbols, OFB: 28 symbols.

Question 2

Consider the following message digest function:



Where X is a chunk of 3 bits of the message and H the hash of the block resulting from hashing all the previous (following Merkle).

To calculate the hash we iterate 3 times.Firstly with X and the hash of the hashing of all the previous chunks of the message, then secondly and thirdly the input X and the output from the previous iterations. S^1 is a right shift of 1 bit. Explain why this is not a good digest function.

It can be shown not to meet the requirements of a hash function:

- Output not correlated with the input
- Varying one bit in the input gives the probability of a given bit in the output varying of 0.5 Good behaviour with respect to collision and
- Coverage of the output space

Question 3

Justify firstly for symmetric key cryptosystems and then for asymmetric key cryptosystems, how they are capable of providing data confidentiality, integrity, authentification of the parties and non-repudiation of the sender.

	CONFIDENC.	INTEGRITY	AUTHENTIF.	NON-REPUD.
SYMMETRIC	Encrypting the message with the secret key	Message authentification/integrity codes	Encrypting the message with the secret key	It is not possible
ASYMMETRIC	Encrypting the message with the K_{PUB} of the receiver	Digital sign of T by the sender	Digital sign of T by the sender	Digital sign of T by the sender

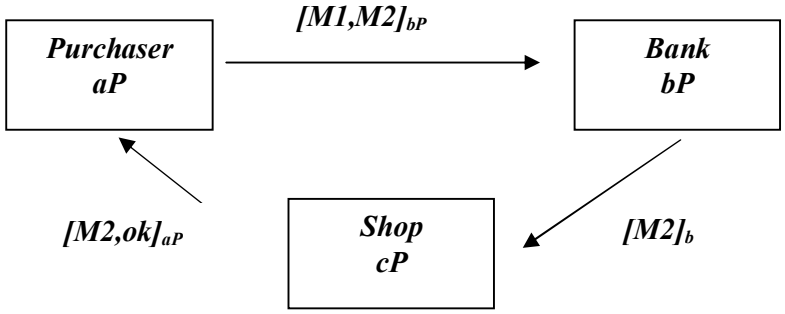
Question 4

Explain the operation of the self-certifying keys of Girault. Comment on the implications of signing an electronic message with the private key attaching as a certificate the public key to allow the verification of the signature.

First part in notes.
Verification of the signature requires that the signer possesses the "a" of the public key $(g^a * I)^d$ which implies that the signer must be on-line

Question 5

A system of secure payments uses asymmetric cryptography based on elliptical curves. The figure shows the sequence of messages in a payment.



aP, bP and cP are the public keys of the system. The purchaser sends to the bank the concatenation of the banking data and the purchase order, all coded with the pubilc key of the bank. The bank verifies the buyer´s data, charges him the purchase amount and sends the signed purchase order to the shop. This gives the ok to the client.

- ***a. Assuming a system based on ElGamal explain how the coded message is sent from the user to the bank.***
- ***b. In order to sign, it is necessary to translate to elliptic curves the algorithm of ElGamal based on exponentiation. If the said method based on exponentiation calculatse r as***

$$r = g^h \bmod_p$$

present the expression that would be utilised using elliptic curves and explain each of the terms of the expression

a. Seen in theory: [M1,M2]+k(bP),kP
b. r=hG where h is a random integer less than the order of the point G and G a point on the elliptic curve.

EXAM 12

EXERCISE 1

A symmetric key cryptosystem is using a certain operation mode together with DES algorithm. The structure of the encrypting operation is defined by the following expression (where 'h' is a non-specified operation that maintains input block size in its output).

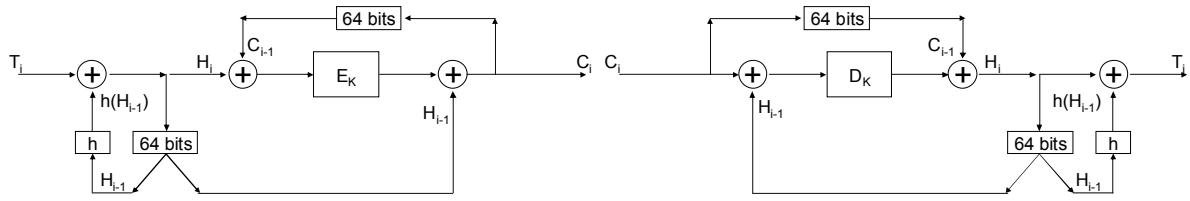
$H_i=T_i \oplus h(H_{i-1})$
$Ci=E_K[H_i \oplus C_{i-1}] \oplus H_{i-1}$

a) Write the expression corresponding to the decryption operation and probe analytically that after the whole process it is possible to obtain the original plaintext.

$H_i= D_K[C_i \oplus H_{i-1}] \oplus C_{i-1}$
$T_i=H_i \oplus h(H_{i-1})$

$D_K[C_i \oplus H_{i-1}] \oplus C_{i-1} \oplus h(H_{i-1})= D_K[[E_K[H_i \oplus C_{i-1}] \oplus H_{i-1}] \oplus H_{i-1}] \oplus C_{i-1} \oplus h(H_{i-1})=$
$\cancel{D_K}[[\cancel{E_K}[H_i \oplus C_{i-1}] \oplus \cancel{H_{i-1}}] \oplus \cancel{H_{i-1}}] \oplus C_{i-1} \oplus h(H_{i-1})= H_i \oplus \cancel{C_{i-1}} \oplus \cancel{C_{i-1}} \oplus h(H_{i-1})= H_i \oplus h(H_{i-1})=T_i$

b) Depict both the encryption and the decryption structure indicating what information must be shared between the sender and the receiver entity so that this last one can decrypt all the data.



The information shared will be C_0, H_0 y K.

Assume that a 6420 bits message is going to be send to a destination and that h(X)=X.

c) Evaluate whether it is possible for this operation mode to provide an appropriate integrity service over the data that has been sent and compare it with the service offered by CBC operation mode

This mode propagates errors to the block where they have been produced and to all the following ones. It is different to CBC in the sense that in case there is an error, the padding would also be corrupted and the modification detected.

EXERCISE 2

We plan to use a system based on ElGamal both for encryption and signing. Let's call *x* to the private key, *p* to the prime number and *g* to the generator.

a) If we used p=13, g=3 and x=5, prove that we would not be able to sign using ElGamal (*).

g is not a body generator, i.e., there is an exponent q so that $g^q \bmod p = 1$ with q being a divider of Euler function. We can see that q=3 for this situation.

b) Let's assume that p=13, g=6, and x=5. Find the associated public key.

If g = 6 and = $6^5 \bmod 13 = 2$

c) Find k so that the encrypted message coincides with the clear text (making use of the keys in 2).

It must be that $g^{xk} \bmod p = 1$. Since g is a generator it must be that xk is Euler function factor. In this case k = 12

d) If k=2 and the encrypted message is c=3, find the message sent in clear text.

if k=2 and c=3 we have that m = 4

(*) Hint: remember that the signature with ElGamal validates the expression $g^{H(M)} = g^{H'(M)}$ and we want to be sure that H(M)=H'(M)

EXAM 13

EXERCISE 1

We have the following unordered four messages that belong to a security protocol:

1) $A \rightarrow B$: $A, E_{Kab}[N_A]$

2) $B \rightarrow A$: $E_{Kab}[N_A, N_B]$

3) $A \rightarrow B$: $E_{Kab}[N_B]$

4) $B \rightarrow A$: $E_{Kab}[K_S, N'_B]$

a) Order these messages, indicating what is the purpose of the protocol and detailing the utility of every exchange message and parameter.

This is a session key Exchange protocol with mutual authentication based on challenges (the "Andrew Secure RPC Protocol"). The correct order would be 1, 2, 3, 4. The first message identifies the origino f the request and shows B the master key it has to use to decrypt the message (K_{ab}). Next message answers the challenge so that B authenticated and another challenge is performed. Next message answers this challenge and authenticates A. The last message exchanges the session key and sends a new challenge to use it later.

Supposing that an opponent M is able to interrupt, intercept, modify or fabricate messages, this protocol could be easily used to perform an impersonation attack.

b) Explain how can this attack be done, detailing all the messages that will be exchanged between the different entities.

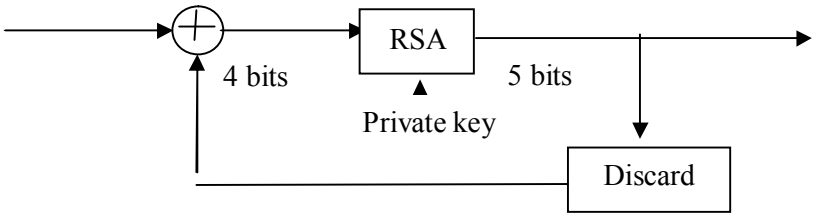
Typical reply attack on the direct key exchange. Since there is no timestamp or challenge on A in the last message, if the opponent obtains K_S (he has unlimited time to do it), he has just to resend the previously stored message $E_{Kab}[K_S, N'_B]$ to impersonate B.

c) Indicate how could this attack be easily prevented with just a minor modification on the protocol and without increasing the number of exchanged messages.

It could be easily avoided including N_A in the last message: $B \rightarrow A$: $E_{Kab}[K_S, N_A, N'_B]$

EXERCISE 2

Let's design a hash algorithm based on RSA using the following schema:



The message m is divided into 4 bit blocks that are used one by one starting from the least significant one. The discard box discards the most significant bit in the output. This box is initialized to 0s.

a) If n=21 (for the RSA box), make comments about why to use or not to use the following public keys: e=3, e=5 and e=13.

n=7*3; Euler function remains 6*2=12
e=3 cannot be selected because it is not prime with Euler function
e=5 can be selected
e=13 cannot be selected because it is greater that Euler function

b) If e=5, find the hash value of the message m=31.

If e=5 it is easy to calculate that d=5 as well.
M=31 in binary is expressed as: 0001 | 1111
We have to start by the least significant ones (1111=15). If we operate with the discard box initialized to 0 we have: $15^5 \bmod 21 = 15$ → XOR 1 → $14^5 \bmod 21 =$ **14**

c) Find a preimage for H(m)=17.

In this case it is easy to invert RSA → d= 5 → $17^5 \bmod 21 =$ **5**

EXAM 14

QUESTION 1

An asymmetric encryption algorithm takes 56 bits plain text blocks as input, and generates 56 bits of encrypted text in the output.

A user wants to confidentially send 68 times the letter X (coding them using 7 bits ASCII).

a) Explain what operation mode would you choose taking into account the four standard modes (detailing why do you consider that the three not chosen modes have to be discarded).

The best mode is CBC.

The selected operation mode is going to be compared with PCBC mode (a non standard mode used in Kerberos). In case PCBC were used with a symmetric encryption algorithm it would be defined according to the following equation:

$$C_i=E_K[T_i \oplus T_{i-1} \oplus C_{i-1}] \text{ y } T_i=D_K[C_i] \oplus C_{i-1} \oplus T_{i-1}$$

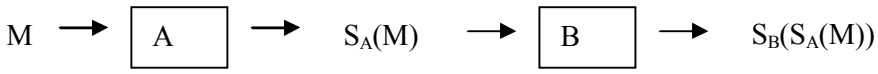
In order to perform the comparison the previous asymmetric algorithm is used to send the 68 letters.

b) Compare both operation modes in terms of number of affected bits after decryption (1), number of incorrectly decrypted passwords (2) and integrity provided in the reception (3), supposing that an error occurs in the transmission of the bit number 200.

An error in the bit 200 -> error in the bit number 4 of the symbol number 29
In CBC: criterium (1), 29 bits;criterium (2), 9 letters;criterium (3), unless the receiver knows what is he going to receive no changes would be noticed.
In PCBC: criterium (1),154 bits ;criterium (2), 44 letters; criterium (3), changes would be noticed.

QUESTION 2

We want to design a double signature system. We pretend to build it on top of RSA. Each message to be signed in this system is first signed by A and then signed by B as illustrated below:



a) What information has to be sent as the double signature of the message M? Can you propose a way to validate this double signature?

Both $S_A(M)$ and $S_B(S_A(M))$ are needed because the RSA scheme for digital signature includes a hash function. If we do not send both we can not validate the signature. The corresponding public keys are needed in the validation process. A's public key is used for $S_A(M)$ and B's is used for $S_B(S_A(M))$

Cryptoanalyzing an RSA based encryption system a hacker finds a message M that accomplishes the following expression: $M^t \bmod n = 1$
where t is a much smaller number than p and q (of course not known by the hacker).

b)

1. **How could the hacker break the system?**
2. **Use the following numbers to concrete a certain attack: n= 39, M=2 and t=12 (t is not smaller than p and q because we have chosen a simple example in order to make calculations simpler)**

*We know that t divides the Euler's function of n and so the following expression is met: d*e = 1 + k*t and if we know the value of e we can calculate d.*

In this case we have: $2^{12} \bmod 39 = 1$
*If we take e = 5 for instance, we will have: d*5 = 1 + k*12 and d=5 and so the system is broken*

When the international operations manager (IOM) hears that the RSA based system has been broken, first he panics, then he decides to change RSA and use ElGamal instead. Let's assume that the system uses the following numbers:

$$\text{Public key} = (p, g, g^x) = (11, 2 \text{ or } 3, 8 \text{ or } 5)$$
$$\text{Private key} = x = 3$$

The IOM does not know which number to use as g.

c) Could you help him or her? Is any of them valid? Why?

g=3 is not valid since 3 is not a generator of the finite field ($3^5 \bmod 11 = 1$). We should take then g=2.

Let's assume that the IOM choses as public key (11, 2, 8). We want to sign a message that generates H(M)= 5. In order to do that, we generate a random h= 3. We know that ElGamal signature requires the following calculations to be made:

$$r = g^h \bmod p$$
$$s = (H(M) - x*r) * inv[h, \phi(p)] \bmod \phi(p)$$

d) Calculate [r,s]. Validate the signature.

r = 8, s = 7
*The validation process should be: $y^r r^s \bmod p = g^{H(M)} \bmod p$??? and giving values: $8^8 8^7 \bmod 11 = 2^5 \bmod 11$, 5*2=10*

EXAM 15

EXERCISE 1

We plan to use a system based on RSA both for encryption and signing.

a) if we used n=55 and e=2, prove that we would not be able to encrypt nor decrypt.

n=55 therefore p=5 and q=11. Euler phi function (55)=4*10
e=2 is not coprime to 40.

b) if we used n=55 and e=3 what would be the public and private keys?

private key {27, 55}, public key {3, 55}

c) If the hash value of the plaintext is 49, verify if 34 is a valid signature. If it isn't give the proper signature value

14 is the proper signature. 34 therefore not a valid signature

d) If the ciphertext intended for the private key owner is 3, find the cleartext.

private key owner should do
3^27mod55=42 (*the* answer :)

e) Suppose that Alice and Bob exchange their public keys this way:

A -> B A,E_{PA}
B -> A B,E_{PB}

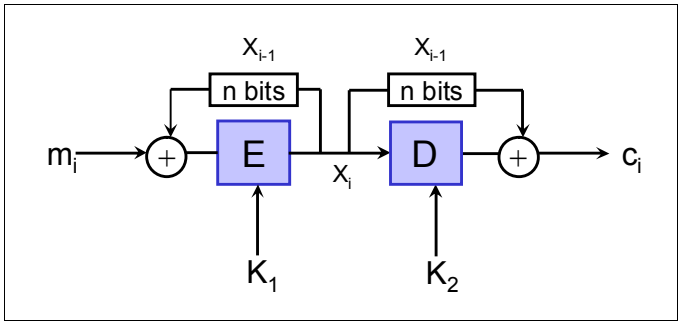
Describe how Mallory could perform fabrication, interception and modification attacks on all future communication between A and B thought to be confidential.

Man-in-the-middle attack. Mallory intercepts the public keys and sends it's own instead. both believe to be in possesion of each others keys while actually only of Mallory's key.

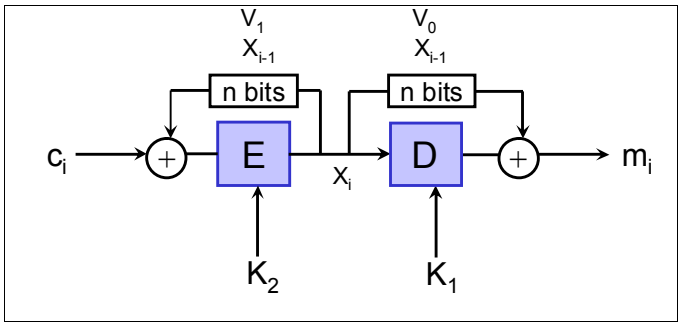
EXAM 16

EXERCISE 1

A certain enterprise has decided to use several operation modes combined at the same time together with the AES algorithm in order to encrypt data. The whole box that would be located in the source entity is shown below:



a) Complete the whole schema with the details about the box that will be used by the receiver to decrypt the message.



b) Demonstrate analytically the all the cryptosystem is valid, verifying that from the cipher text C_i generated by the origin it is possible to recover M_i in the destination box (remember that every cryptosystem must generically accomplish $D_K[E_K(M)]=M$).

$$C_1 = D_{k_2}\left[E_{k_1}\left[V_0 \oplus M_1\right]\right] \oplus V_1$$
$$M_1 = D_{k_1}\left[E_{k_2}\left[V_1 \oplus C_1\right]\right] \oplus V_0$$
$$C_2 = E_{k_1}\left[V_0 \oplus M_1\right] \oplus D_{k_2}\left[E_{k_1}\left[E_{k_1}\left[V_0 \oplus M_1\right] \oplus M_2\right]\right]$$
$$M_2 = E_{k_2}\left[V_1 \oplus C_1\right] \oplus D_{k_1}\left[E_{k_2}\left[E_{k_2}\left[V_1 \oplus C_1\right] \oplus C_2\right]\right]$$
$$C_i = E_{k_1}...\left[E_{k_1}\left[E_{k_1}\left[V_0 \oplus M_1\right] \oplus M_2\right] \oplus M_{i-1}\right] \oplus D_{k_2}\left[E_{k_1}...\left[E_{k_1}\left[E_{k_1}\left[V_0 \oplus M_1\right] \oplus M_2\right] \oplus ...M_i\right]\right]$$
$$M_i = E_{k_2}...\left[E_{k_2}\left[E_{k_2}\left[V_1 \oplus C_1\right] \oplus C_2\right] \oplus C_{i-1}\right] \oplus D_{k_1}\left[E_{k_2}...\left[E_{k_2}\left[E_{k_2}\left[V_1 \oplus C_1\right] \oplus C_2\right] \oplus ...C_i\right]\right]$$

Changing in the last equation C_i by its corresponding value it can be verified that M_i=M_i (it can be easily seen when C_1, C_2,, etc. are progressively introduced in the equation and cancellation the different expressions)

c) Suppose that a 1024 bits text is going to be transmitted and that a padding is always added (transparent padding). If an error occurs on bit number 400 of the cipher text, how many bits (average) will be wrongly decrypted by the receiver?

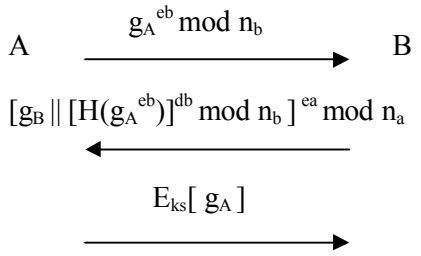
1024/128= 8 blocks (error in the 4th block) and an additional block with the padding
*Error in 5 blocks 128*5/2= 320 error bits in the plaintext*

d) What can you say regarding the integrity provided by the mechanism described in this exercise? and how is it related with the usage of padding?

Since all the decrypted plaintext is modified from the error bit until the end, as soon as some redundancy is added at the end of the plaintext (padding for instance) it would be possible to detect any intentioned or non intentioned modification.

EXERCISE 2

Let's consider a system in which all the users have RSA keys. A CA is responsible of the generation of certificates for all the public keys of the users in the system. These certificates are distributed to all the users before any secured communication is established among these users. Let's also assume that in order to create a secured connection, A and B (users of the system) exchange the following messages:



where g_A is a number chosen by A, e_b and n_b represent the public key of B, g_B is a number chosen by B, e_a and n_a represent the public key of A, H is a hash well designed function, d_b is the private key of B, the sign || means concatenation, Ks is the session key for a given symmetric cipher E and is calculated using the following expression:

$$Ks = g_A * g_B$$

a) Justify if the authentication can o can not be repudiated.

The second message contains a digital signature from B of the first message so B can not repudiate the authentication process. However, A uses the shared symmetric key for its authentication and this can be repudiated.

b) If Ks had been simply chosen to be the value of g_A, indicate how to supplant A in the system

In this case A can be supplanted as follows. A user C generates g_C and encrypts it for B. B calculates the digital signature of the message, concatenates it to g_B and sends it back to C but encrypted for A. However, in this case, the symmetric key is simply g_C so C can generate the third message without decrypting the second.

c) If a man in the middle (M) captures the first message and has the capacity of making B sign whichever message, indicate how M could supplant B in the system represented in the figure (assume that RSA is used to sign and that no hash function is used in the signature process).

If we take out the hash function from the signature in RSA, M can make B sign the first message to obtain g_A, then M will generate a g_M and generate the second message concatenating both numbers encrypted for A.

d) If we know that the public key of A is 29 and that the Euler function of n_A is 40, calculate A's private key using the fastest algorithm you know

Using the Euclides extended algorithm we obtain that the private key is also 29.

EXAM 17

Exercise

Alice (A) wants to establish a safe communication with Bob (B). For that purpose they decide to use IPSec. In the communication establishment they decide to use AES and SHA-1 and exchange a Diffie-Hellman key.

1) Outline the message exchange for the establishment of the IKE SA.

First message exchange, Alice does propose some encryption and hash function algorithms. One of the available options is AES and SHA-1. Bob response indicates that these are the chosen algorithms.

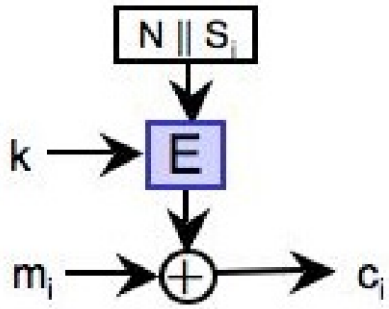
In the second message exchange, Alice sends a Diffie-Hellman public key and a Nonce, Bob answers with his DH public key and another nonce. Both then compute the shared key to be used for the IKE SA.

In the third message exchange, Alice encrypts with the pre-shared key and AES as algorithm his identity and the fingerprint of the previous messages using HMAC-SHA-1.

2) What is the shared key if p = 11, primitive root of p = 2 and the public key sent by A is 8 and the private key used by B is 5?

p=11 g=2
the public key sent by B is $g^x \bmod p = 2^5 \bmod 11 = 10$.
The private key of A is such that $2^x \bmod 11 = 8$, so we chose x=3
A receives 10, B receives 8. The shared key as computed by A is $10^3 \bmod 11 = 10$
The shared key as computed by B is $8^5 \bmod 11 = 64*64*8 \bmod 11 = 81*8 \bmod 11 = 10$

3) In the IPSec SA negotiation. They decide to use a simplification of CCM, an algorithm that provides encryption and authentication using the same key. CCM stands for CTR + CBC-MAC, where CBC-MAC is used to generate a MAC and CTR is a counter mode of operation for a symmetric cipher, which works as follows:



where N is a Nonce and S_i is a counter that starts being 0 and rises by 1 on every pass of the algorithm. So, to cipher m_1, we would concatenate N and '00000000'; for m_2 we have N and '00000001', etc…

a) **To achieve authentication all the c_i blocks generated with CTR are passed through CBC-MAC. Draw a box diagram explaining the CBC-MAC algorithm.**

Diagram in lecture notes

b) What data does the recipient of the cipher text require and what operations are necessary in order to decrypt and verify authentication of a received cipher text?

The recipient requires the Nonce as well as the key (both have to be sent in a secure authenticated manner), the cipher text and the CBC-MAC. In order to decrypt the cipher text we will use the Nonce and the counter initialized to 0 to pass through the cipher algorithm and the result will be xor'ed with the cipher block to obtain the plaintext. Before actually showing the plaintext, the integrity of the cipher text should be verified; generating the CBC-MAC of the received cipher blocks will do this. If the result is identical to the CBC-MAC received, the integrity of the cipher text and its authenticity is verified and the plaintext can be accepted and shown.

c) Detail the similarities and differences between CTR and OFB modes of operation. Point out any weakness of CTR if found.

Similarities:

- *both coder and decoder are identical in both cases. We use the Encryption algorithm to code and decode in the same way, encrypting.*
- *There is no error propagation, in case of transmission errors, the erroneous bits will be affected in the cleartext, but no more.*
- *Both require and IV (or Nonce)*
- *Both have no auto synchronization on block loss*

Differences

- *In CTR a predictable string is encrypted on each round. It is therefore important that the Nonce is fresh for every encoding!*

d) Imagine Mallory (M) could intercept the cipher text transmitted from A to B. What consequences would it have if M disorders 2 cipher text blocks? Justify your answer.

On reorder of two blocks both will be xor'ed with the output of the encryption of N + a different counter S and therefore will not generate the same plaintext back. The rest of the plain text won't be affected. As in this case we do have integrity checks through the CBC-MAC, before any wrong plain text is shown to the receiver the CBC-MAC will be verified. As 2 blocks come disordered, the MAC won't be the same and the integrity check will fail warning the receiver and avoiding M doing undetectable changes.

EXAM 18

EXERCISE

The following messages belong to a security protocol:
1. B, A, $E_?[K_2]$
2. B, $E_{K1}[K_2]$
3. A
4. A, B, $E_?[K_1]$

a) Put the messages of the protocol in order explaining (1) the origin and the destination of the messages (2) the purpose of the protocol, (3) the value of '?' ('?' represents a key), (4) the purpose here of the trusted third party.

This is a session key exchange protocol
1. A→S: A, B, $E_{KuS}[K_1]$, 2. S→B: A, 3. B→S: B, A, $E_{KuS}[K_2]$, 4. S→A: B, $E_{K1}[K_2]$
? is S public key. S provides authenticity to the message exchange

b) Describe two easy attacks that M can perform, one to impersonate A and another one to impersonate B (this last one is obviously requiring A to start sending a message fist).

1. M(A)→S: A, B, $E_{KuS}[K_M]$, 2. S→B: A, 3. B→S: B, A, $E_{KuS}[K_2]$, 4. S→M(A): B, $E_{KM}[K_2]$
1. A→S: A, B, $E_{KuS}[K_1]$, 2. S→M(B): A, 3. M(B)→S: B, A, $E_{KuS}[K_M]$, 4. S→A: B, $E_{K1}[K_M]$

c) Combining the two previous attacks it is possible to create a situation where A and B think they are having a secure communication while M knows the session key they are using (confidentiality attack).

1. M(A)→S: A, B, E_{KuS} $[K_M]$, 2. S→B: A, 3. B→S: B, A, E_{KuS} $[K_2]$, 4. S→M(A): B, $E_{KM}[K_2]$, 5. A→S: A, B, $E_{KuS}[K_1]$, 6. S→M(B): A, 7. M(B)→S: B, A, $E_{KuS}[K_2]$, 8. S→A: B, $E_{K1}[K_2]$

EXERCISE 2

We want to deploy a security service in order to provide confidentiality to our network communications. As the length of the messages is small (less than 2048 bits) we think that asymmetric encryption is a good alternative. Despite this, the IT manager of our company wants more and adds a shared secret based mechanism to RSA. The shared secret required is known by every user in our company. The proposed schema for encryption is:

$$C = (k_c * m)^e \bmod n$$

Where kc is the shared key by all users and e is the appropriate RSA key.

a) Explain the algorithm to get back the original message from C.
First calcúlate C^d (d is the RSA prívate key) and then multiply the result per k_c^{-1} modulo n.

b) If the keys for a particular receiver are n=73*89 and e=5, calculate d.
The value of the Euler's function is 6336 and after using the Euclide's extended algorithm we get d= 5069

c) In order to decrypt messages we think that the Chinese reminder theorem can be of help (we can avoid making exponentiations of lengthy exponents). Explain the operations to be done.
We want to solve $N = C^d \bmod n$ as follows:

$$N = \{A_p[C_p^{dp} \pmod p] + A_q[C_q^{dq} \pmod q]\} \bmod n$$

with: $A_p = q\,[inv\,(q,p)] = q^{p-1} \bmod n$

$A_q = p\,[inv\,(p,q)] = p^{q-1} \bmod n$

$d_p = d \bmod (p-1)$ $\quad$ $d_q = d \bmod (q-1)$

$C_p = C \bmod p$ $\quad$ $C_q = C \bmod q$

d) If a particular system uses kc=5 and we receive C=4921 calculate m (in order to simplify calculations we know that if we apply d to C we get the value of 6111).
M=3821

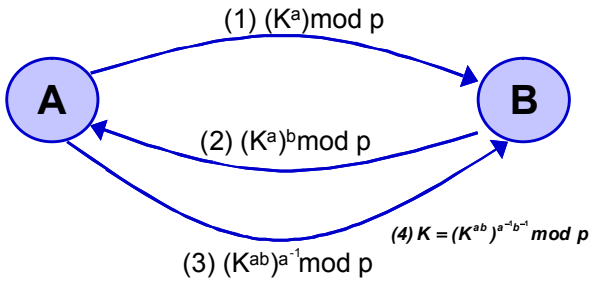
e) If now we were interested in doing calculations using elliptic curves, explain the mathematical expression to be used for encryption.
Only applied to the asymmetric part:

P is a point in the curve with its x component $\rightarrow x=[(k_c * m) \bmod n]$
*C= is the x component of the point in the curve e*P*

EXAM 19

QUESTION 1

Adi Shamir proposed a session key exchange mechanism that implied the usage of a commutative encryption algorithm (the order chosen to apply the different encryption operations does not affect the final result). When the chosen algorithm is exponentiation, the resulting protocol is shown in this picture:



a) Now suppose that the chosen algorithm is the XOR operation. Detail how would the complete message exchange is.

1. K XOR A, 2. (K XOR A) XOR B, 3. ((K XOR A) XOR B) XOR A, 4. (((K XOR A) XOR B) XOR XOR B

Ex.: K=1111, A=1010, B=0101

1. 1111 XOR 1010=0101, 2. 0101 XOR 0101=0000, 3. 0000 XOR 1010=1010, 4. 1010 XOR 0101=1111

b) Suppose that an opponent can sniff, generate and replay messages when the XOR based protocol is being used. What serious problem presents the protocol in this situation?

The sum of the captured messages equals K

QUESTION 2

1. Explain what are the paired keys in RSA

Theory

2. We want to establish an IPSEC connection using AH in order to transfer a 128 bits message encrypted with DES. Draw (using boxes) the different stages in the process both in the sender and in the receiver (assume a value of 1024 bytes for the MTU).

Theory

3. Explain what the attribute certificates are inside a PMI infrastructure.

Theory

4.Justify the improvements of DSS if compared to El Gamal for signing.

Theory

EXAM 20

EXERCISE

You have been selected to design the implementation of RSA as cryptographic algorithm to prevent plaintext communications on the wireless interface that communicates a team of devices with their controllers. Initially, because of the technical restrictions associated with the minuscule dimensions of the devices, the size of the numbers that they can process is limited to 7 bits.

a) When you are about to choose the parameter n, your boss (driven by misuse concerns) instructs you to use the biggest feasible value for n. Taking into consideration the condition find out p, q and n.

The 7-bit constraint means that $n \leq 127$.
Testing for several values of n:

- $n = 127$ *Unfeasible (only one prime factor).*
- $n = 126 = 2 \cdot 3^2 \cdot 7$ *Unfeasible (more than two prime factors).*
- $n = 125 = 5^3$ *Unfeasible (one prime factor).*
- $n = 124 = 2^2 \cdot 31$ *Unfeasible (more than two prime factors).*
- $n = 123 = 3 \cdot 41$ *Feasible (two prime factors exactly).*
  - *Therefore $p = 3$ and $q = 41$*
  - *$\Phi(n) = (p - 1)(q - 1) = 2 \cdot 40 = 80 = 2^4 \cdot 5$*

To implement confidentiality and using your value for n, another coworker defined the different pairs of cryptographic keys (one pair for each device). Now that you need to communicate with device A you realize that your colleague gave you the list of private keys (instead of giving you the list of the public keys).

b) Explain why you cannot communicate with the device using the related private key of the list. Justify satisfactorily whether it is generally possible to learn the public key from the private key or not. Finally, find out the public key of the device A (its private key is $d = 13$ and n).

As everything sent to the device is going to be decrypted with its private key, first you need the public key to send data.
Normally, deducing the public key from the private key is a computationally costly process, which involves factoring n (usually a big number). In this case, n is not a huge number; additionally we know the factors of n in advance.
$e = 13^{-1} \bmod \Phi(n) = 13^{-1} \bmod [(p - 1)(q - 1)] = 13^{-1} \bmod 80 = 37$

c) For the plaintext $M = 11$ and using the public key of the device B ($e = 31$ and n) find out the corresponding ciphertext C.

The ciphertext can be calculated from e and n.
$C = M^e \bmod n = 11^{31} \bmod 123 = 65$
A recently discovered technology allows the handling of 8-bit numbers in the devices and your boss decides to port the cryptographic system to the new platform.

d) Within this new architecture and again, using the biggest feasible value for n, find out p, q, and n.

The 8-bit constraint means that $n \leq 255$.
Testing for several values of n:

- $n = 255 = 3 \cdot 5 \cdot 17$ *Unfeasible (three prime factors).*
- $n = 254 = 2 \cdot 127$ *Feasible (two prime factors).*
  - *Therefore $p = 2$ and $q = 127$*
  - *$\Phi(n) = (p - 1)(q - 1) = 1 \cdot 126 = 126 = 2 \cdot 3^2 \cdot 7$*

When you are just going to define the new pairs of cryptographic keys (one different pair for each device) you note that the abovementioned coworker wrote the d value of the associated private key (for the 7-bit implementation) in the expensive read-only memory integrated with each device. As a result, now you cannot change the d values of the private keys. Therefore in each device you must

use for the new 8-bit implementation the same value of d for the private key that you were using for the old 7-bit implementation.

e) Find out the new public key for the device A (remember, its private key is $d = 13$ and the new value for n)

The new key will be:

$e = 13^{-1} \bmod \Phi(n) = 13^{-1} \bmod 126 = 97$

f) Communications are not possible with the device with private key $d = 7$, but all tests indicate that everything in the device is working OK. Justify adequately in the 8-bit scenario whether the impossibility of changing the private keys will cause problems or not. In the affirmative, which private keys of the 7-bit case are no longer valid for the 8-bit case?

For RSA, just like e, d must be also odd, $d < \Phi(n)$ and d and $\Phi(n)$ must be coprimes. While for the 7-bit case $d = 7$, $n = 123$ is a valid private key, [d and $\Phi(123) = 80 = 2^4 \cdot 5$ are coprimes] for the 8-bit case $d = 7$, $n = 254$ are not because $d = 7$ and $\Phi(254) = 126 = 2 \cdot 3^2 \cdot 7$ are obviously not coprimes (both have 7 as common factor). Normally any given e coprime of $\Phi(n_1)$ is not going to be always coprime of $\Phi(n_2)$, therefore one needs to check the coprimality condition.

As d must be odd, we can forget about the factor 2, and conclude that the 7-bit keys that are no longer valid for the 8-bit case are those that were multiple of 3 or 7 (the two new factors introduced by $\Phi(127) = 126 = 2 \cdot 3^2 \cdot 7$).

EXAM 21

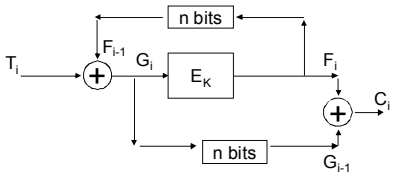
EXERCISE 1

The following equations define the encryption box of the operation mode called PES-PCBC (*Privacy Enhanced Sockets Propagating CBC*):

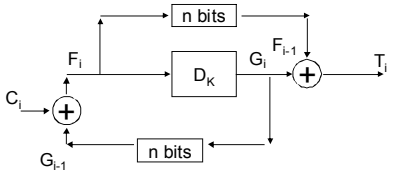
$$C_i = F_i \oplus G_{i-1}$$
$$F_i = E_k(G_i)$$
$$G_i = T_i \oplus F_{i-1}$$

You are asked to:

a) Show graphically the sender encryption box using a block diagram that represent these equations.



b) Show graphically (block diagram) what the receiver should have in order to decrypt a message encoded using the previous encryption box including in addition the corresponding equations.



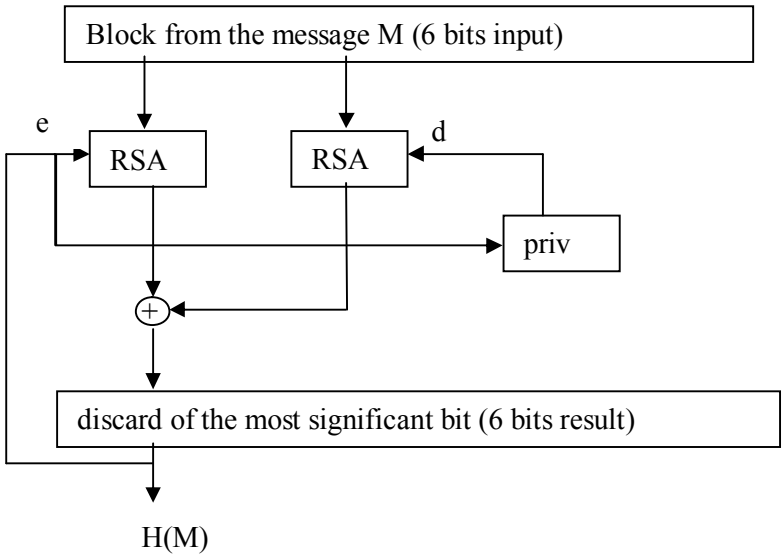
$$T_i = F_{i-1} \oplus G_i$$
$$G_i = D_k(F_i)$$
$$F_i = C_i \oplus G_{i-1}$$

Suppose that a 800 bits text is transmitted and that an error occurs at bit 300.

c) Evaluate, studying the error propagation properties, how this operation mode is capable of providing an adequate integrity service over transmitted data, comparing it with the service that CFB offers with a block size of 32 bits.
Error propagation: PES-PCBC does not recover while CFB does (33 erroneous bit, average).
Integrity (error detection): if the text has some structure, both of them offer an integrity service (better with PES-PCBC). Otherwise only PES-PCBC does since the padding becomes corrupted.

EXERCISE 2

We are lucky enough to be the president of a prestigious cryptographic firm (Crackingcrackedcrackers). One of our engineers knocks the door of our office one morning and tells us about his new MDC function:



where e is the RSA public key, obtained from the hash value of the previous block (initialized to H(0)), the "priv" function computes the RSA private key from the correspondent public key, the two RSA branches are added bit by bit using an XOR function (at a bit level) and the discard function at the end makes the result 6 bits long (possible values from 0 to 63). The MDC function divides the message into 6 bit blocks and iterates block by block using the result from block i as the public key in the computation of block i+1.

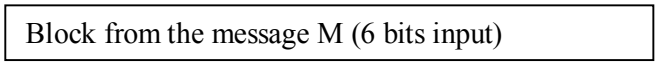
a) Comment the flaws you see in the design of such MDC function.
The algorithm has some important flaws. We highlight 2 of them:
   a. *The keys depend on a hash value which can generate weak keys. The detection is very simple since both branches are XOR added at the end producing a null result for such keys.*
   b. *The discard function makes the different outputs not equally likely.*

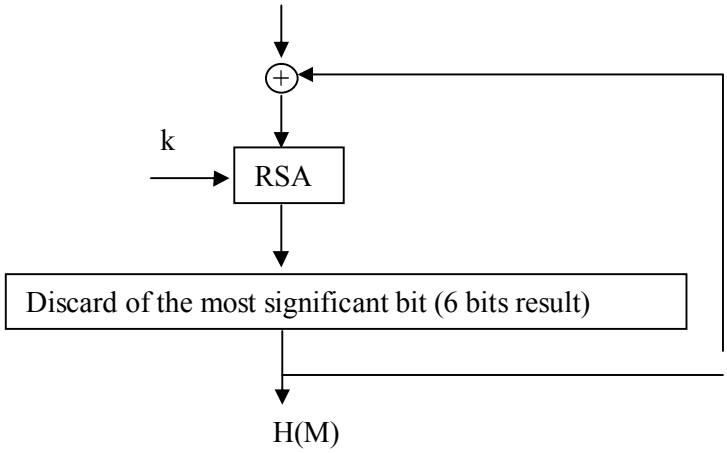
b) If M=10 and we select n=77 (n is the number used for the RSA modular operations) and H(0)=7, calculate the hash value obtained and comment the result.
The extended Euclides algorithm gives us d = 43. This calculation is not needed in this case since the RSA encrypted message is the same as the original message and the computed hash value will be 0. This is because H(0)-1 is 6 which divides the Euler function of 77 (60) and 10 is a generator of order 6.

Hint: the values for 10^x mod 77 are captured in the following table for the different values of x:

x	10^x mod 77
4	67
8	23
16	67

After the failure in the design of the previous MDC function, our engineer does not claudicates. On the contrary, he uses the lessons learnt to build the following MAC function:





where k is the key for the MAC function (which is directly used as the RSA public key in the RSA box). Each block from the input is "XOR-ed" bit by bit with the result from hash value of the previous block. Again, a discard function is used to get a 6 bit output.

When our engineer presents his new design to you, a smile appears on your face. To make him think about it you tell him to calculate a collision for the message M=13 in case he use n=77, k=7 and H(0)=7.

c) What collision can our engineer easily find?
If M=13 and H(0)=7, after the first XOR we get M'=10 which is the input for the RSA cipher. Then H(1)=10. How can we find a collision? Very easily. If we add 64 to 10 we get a second output from the RSA cipher which gives the same output after the discard function (giving the same H(1)). In this case, the output for the RSA box should be 74 (therefore the hint). If we decrypt 74 with the private key (d = 43) we get an input M' = 39. Therefore, the message M should be M' xor 7 = 32 which is a collision with message M=13.

Hint: the values for 74^x mod 77 are captured in the following table for different values of x:

x	**74^x mod 77**
2	9
4	4
16	25
32	9