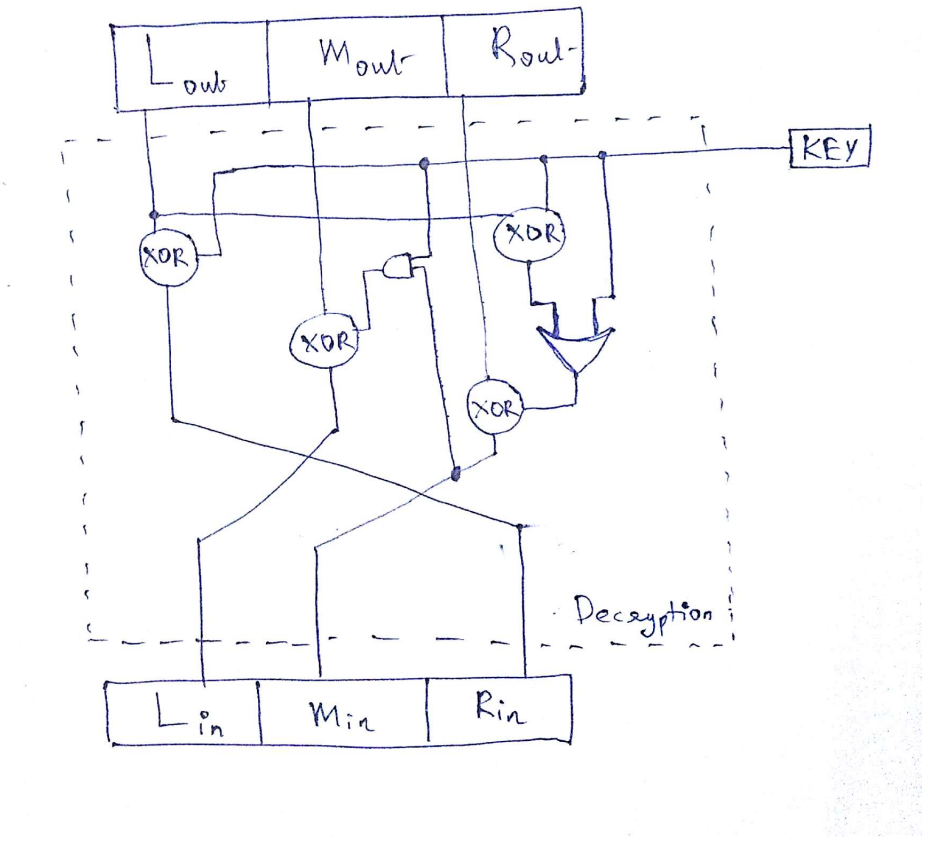
Computer Science and Engineering
VI semester

PRINCIPLES OF CRYPTOGRAPHY ASSIGNMENT-2

Done By:

A.SATHISH KUMAR	**106115002**
DHANANJAY.P	**106115024**
PRADEEP.R	**106115062**
ROHITH BALAJI.S	**106115070**
SIVANANDHAM.S	**106115074**

1) Consider the following block structure encryption. The input blocks is divided into 3 sub-blocks: L_{in} (Left sub-block), M_{in} (Middle sub-block), R_{in} (Right sub-block). The encrypted output is composed $L_{out}, M_{out}, R_{out}$ as in the input block. Design the decryption structure.



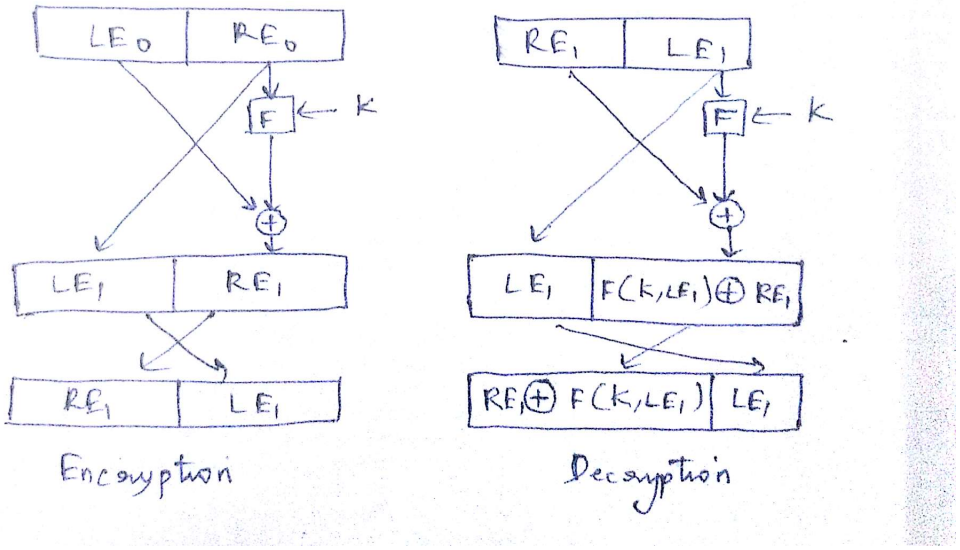
---

2)

- Show that DES Decryption is similar to Encryption.
- Draw the diagram for single round of DES Algorithm (with diagram of F-function). Let X' be the bitwise complement of an encryption key is taken, then the result of DES encryption with these values is the complement of the original ciphertext. That is, If $Y = E(K, X)$, then $Y' = E(K', X')$. You can use the results: $(A \oplus B)' = A' \oplus BandA \oplus B = A' \oplus B'$
- How many key space searching is required to make brute-force attact on DES. Does the result of part(a) change that under chosen plaintext attack? How?

Proof that Encryption is similar to Decryption in DES using Mathematical Induction:-

Step 1:
For $n = 1$, That is for one round,

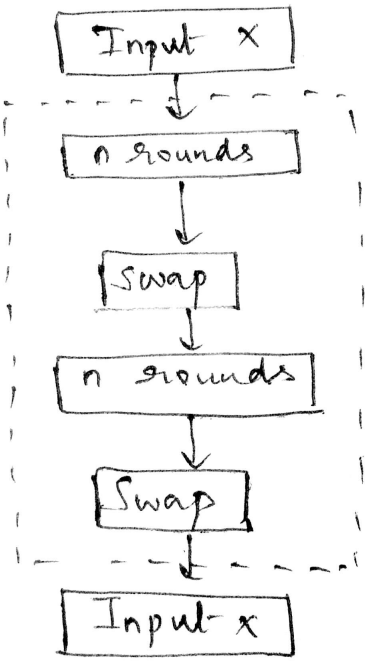


Now to prove: $LE_0 = RE_1 \oplus F(K_1, LE_1)$ - - 1
Proof:
$RE_1 = LE_0 \oplus F(K_1, RE_0)$ - - 2
$RE_0 = LE_1$ - - 3

Apply 2 and 3 in 1 $LE_0 = LE_0 \oplus F(K_1, RE_0) \oplus F(K_1, RE_0) = LE_0$
Hence proved for one round.

Step 2:

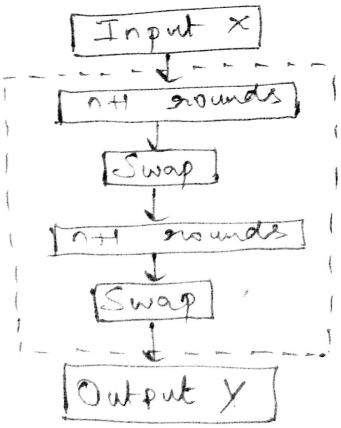
Let us assume it is true for n rounds,

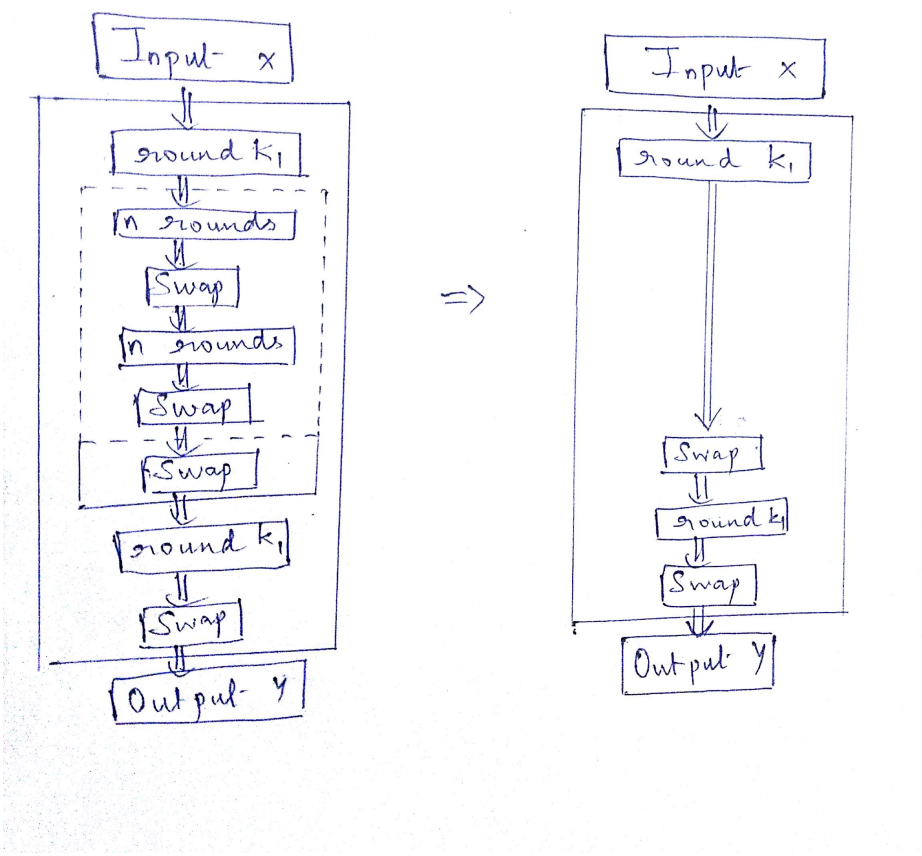


Step 3:

Now to prove for n+1 rounds,

Since, it is true for one round, outputY = inputX, hence it is true for n+1 rounds





Hence proved.

b) To prove: $Y' = E(K', X')$

Given: $Y = E(K, X)$

$X = L_i || R_i$

$Y = L_{i+1} || R_i + 1$

After DES encryption, we obtain the following results:-

1. $L_i = R_{i-1}$ - - 1
2. $R_i = L_{i-1} \oplus F(R_{i-1}, K)$ - - 2

From 1 we get $==> L'_i = R'_{i-1}$ - - 3

Take,

$$\begin{aligned} L_{i-1} \oplus F(R_{i-1}, K) =& L'_{i-1} \oplus F'(R_{i-1}, K)(using A \oplus B = A' \oplus B') \\ =& L'_{i-1} \oplus F(R'_{i-1}, K') \\ =& L'_{i-1} \oplus F(R_{i-1}, K)(using F(R', K') = F(R, K)) \\ =& [L_{i-1} \oplus F(R_{i-1}, K)]'(using (A \oplus B) = A \oplus B' = A' \oplus B)) \\ =& R'_i \\ R'_i =& R_i \\ R'_i =& R'_i = L'_{i-1} \oplus F(R'_{i-1}, K') \end{aligned}$$

From 3 and 4, we can conclude that $Y' = E(K', X')$. Hence proved.

c) When brute force attack is done on the cipher, the key space searching can be calculated using the key size which is 56, hence total no of keys is 256.
Hence complexity if the attack is $O(256)$
Under chosen plaintext attack the complexity changes,
$M_1 ==> C_1 = DES_{K1}(M_1)$
$M'_1 ==> C_2 = DES_{K1}(M'_1)$
This information can be used to reduce the exhaustive search by half,
$C'_2 = DES_K 1'(M_1)$
Hence now we have
$M_1 ==> C_1(K1)$
$M_1 ==> C'_2(K1')$
Now we will try permutations of key,
If M_1 gives C_1 with K $==>$ the key is K
if M_1 gives C'_2 with K $==>$ the key is K'
else K and K' are not keys
$==>$ Therefore, in worst case scenario, both the keys are wrong. So the search key space of K is reduced by half, so the complexity is $O(256/2) = O(255)$

---

3) Explain the meet in the middle attack on 3-DES. Find the memory and time complexity of this attack.

In $3 - DES$, the size of the key is 56. Hence when brute force is applied, the time complexity of cryptanalysis is $O(2^{168})$ But using "Meet in the Middle" attack, it can be reduced.

$$M ==> E_{K1}[M] ==> E_{K2}[E[M]] ==> E_{K3}[E[E[M]] = C$$

Here the middle place happens after the plaintext is encrypted twice and where the ciphertext is decrypted once. Now let us find the time complexity of each event that occurs in the cryptanalysis individually.

1. The time complexity for encrypting the plaintext twice is $O(2^{112})$ (2 sets of keys $=K1, K2$)
2. The time complexity for decrypting the ciphertext once is $O(2^{56})$ (key $K3$)
3. The time complexity for sorting the set of ciphertexts is $O(2^{56} * log(2^{56}))$
4. The time complexity for comparing $K1, K2$ and $K3$ is $O(2^{112} * log(2^{56})$

Hence the total time complexity is sum of all 4 is

$$\begin{aligned}
&=O(2^{112}) + O(2^{56}) + O(2^{56} * log(2^{56})) + O(2^{112} * log(2^{56})) \\
&=O(2^{112} * log(2^{56}))(All\ the\ other\ terms\ are\ negligible) \\
&=O(57 * 2^{112}) \\
&=O(2^6 * 2^{112}) \\
&=O(2^{118})
\end{aligned}$$

---

4) Briefly explain an ideal block cipher. In this ideal block cipher, what is the probability that two keys (k_1, k_2) will give the same pair of plaintext and ciphertext.

A block cipher works by replacing a block of N bits from the plaintext with a block of N bits from the ciphertext, if we consider $N = 4$, then there are 16 different possible 4-bit patterns. We can represent each pattern by an integer between 0 and 15. So the bit pattern 0000 could be represented by the integer 0, the bit pattern 0001 by integer 1, and so on. The bit pattern 1111 would be represented by the integer 15.

In an ideal block cipher, the relationship between the input blocks and the output block is completely random. But it must be invertible to decrypt. Therefore, it must

be one-to-one, meaning that each input block is mapped to a unique output block. The encryption key for the ideal block cipher is the table that shows the relationship between the input blocks and the output blocks.

Key $k1, k2$ map two different input blocks to two different output blocks, these mappings will give same results iff $k1$ and $k2$ are equal. Hence if $k1$ and $k2$ are different, then the probability that two keys $(k1, k2)$ will give the same pair of plaintext and ciphertext is 0.

---

5) $Z_7 = (1, 2, 3, 4, 5, 6)$, $mod 7$ is a group. Write all cyclic subgroups (different order) of Z_7. Is Z_7 cyclic group.

$$\begin{aligned} Z_7 =& (1, 2, 3, 4, 5, 6) * mod 7 \\ =& (3^0, 3^2, 3^1, 3^4, 3^5, 3^3) \end{aligned}$$

So, Z_7 can be written as powers of one of its elements which is 3.
Hence, Z_7 is a cyclic group.
Identity element is 1. And $2^{-1} = 4, 3^{-1} = 5$

$$\begin{aligned} &< 1 >= 1 \\ &< 2 >= 2, 4, 1 \\ &< 3 >= 3, 2, 6, 4, 5, 1 \\ &< 4 >= < 2 >= 2, 4, 1 (because 2^{-1} = 4) \\ &< 5 >= < 3 >= 3, 2, 6, 4, 5, 1 (because 3^{-1} = 5) \\ &< 6 >= 6, 1 \end{aligned}$$

So, the required cyclic subgroups of Z_7 are : $1, 2, 4, 1, 3, 2, 6, 4, 5, 1$ and $6, 1$.

---

6) $H = (0, 1, 2, 3, 4, 5, 6, 7)$, $+mod 8$ is a group. Write all cyclic subgroups(different order) of H. Is H cyclic group.

$H = (0, 1, 2, 3, 4, 5, 6, 7), +mod 8$

$< 0 >=0$
$< 1 >=1$
$< 2 >=2, 4, 0$
$< 3 >=3, 1$
$< 4 >=4, 0$
$< 5 >=5, 1$
$< 6 >=6, 4, 0$
$< 7 >=7, 1$

So, the required cyclic subgroups are $0, 1, 2, 4, 0, 3, 1, 4, 0, 5, 1, 6, 4, 0$ and $7, 1$.

Since no cyclic subgroup contains all the elements of H, (H cannot be represented as powers of any of its elements)
H is not a cyclic group.

---

7) Find generator of Schnorr group for following prime $p = 11$. Also find other elements of the Schnorr group. $p = 11 = 2 * 5 + 1$

Given $p = 11$
$p - 1 = 10 = 2 * 5 = r * q$ where q is prime.
So, q can be either 2 or 5
Schnorr groups are cyclic subgroups of $Z_p * modp$ of size q
$Z_{11} = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$
Let 'g' be the generator For

$g =1, G = 1$
$g =2, G = 2, 4, 8, 5, 10, 9, 7, 3, 6, 1$
$g =3, G = 3, 9, 5, 4, 1$
$g =4, G = 4, 5, 9, 3, 1$
$g =5, G = 5, 3, 4, 9, 1$
$g =6, G = 6, 3, 7, 9, 10, 5, 2, 4, 1$
$g =7, G = 7, 5, 2, 3, 10, 4, 6, 9, 8, 1$
$g =8, G = 8, 9, 6, 4, 10, 3, 2, 5, 7, 1$
$g =9, G = 9, 4, 3, 5, 1$
$g =10, G = 10, 1$

So, Schnorr group $= 1, 3, 4, 5, 9$ Generators of this Schnorr group are $3, 4, 5, 9$

---

8) Consider $R = Z_{18}[x]/(x^4 + 1)$, the ring of polynomials with coefficients from Z_{18}, with operations defined modulo $x^4 + 1$. It is known that R is a commutative ring with unity. Is it a field.

The properties of a field are as follows: -
1. The first operation should have closure, associative, commutative, identity and inverse properties
2. The Second operation should have closure, associative, commutative, identity and inverse (except identity for first operation) properties.
3. The second operator should distribute over the first operator.

Proofs: 1. Condition 1 It is given in the question that R forms a commutative ring, hence the first condition of field is satisfied.

2. Condition 2 For this we need to prove that inverse exists, as ring solves the other part, we know that we can find an inverse for every polynomial in R, because a prime polynomial exists, such that $x * x^{-1} \equiv 1 \mathrm{mod\ prime\ polynomial}$. But we cannot find inverse for 0.

3. Condition 3 Distributive property holds on polynomial multiplication

As all the conditions are satisfied, it is proved that R is a field.

---

9) Consider $R = Z_{11}[x]/(x^4 + 1)$, the ring of polynomials with coefficients from Z_{11}, with operations defined modulo $x^4 + 1$. It is known that R is a commutative ring with unity. Is it a field.

The properties of a field are as follows: -
1. The first operation should have closure, associative, commutative, identity and inverse properties
2. The Second operation should have closure, associative, commutative, identity and inverse (except identity for first operation) properties.
3. The second operator should distribute over the first operator.

Proofs: 1. Condition 1 It is given in the question that R forms a commutative ring, hence the first condition of field is satisfied.

2. Condition 2 For this we need to prove that inverse exists, as ring solves the other part, we know that we can find an inverse for every polynomial in R, because a prime polynomial exists, such that $x * x^{-1} \equiv 1 \mathrm{mod\ prime\ polynomial}$. But we cannot find inverse for 0.

3. Condition 3 Distributive property holds on polynomial multiplication

As all the conditions are satisfied, it is proved that R is a field.

---

10) How to create field of size (3^2). Write all the elements of field of size (3^2). Assume that prime (irreducible) polynomial exist in every degree.

Method:
$F[p^n]$ = poly of degree $<= (n-1)$ with cefficients from F_p Then choose a prime polynomial in $F_p(x)$ of degree 'n' which is $\Pi(x)$ so that the operations of the field are defined as
$->$ Addition same as with F_p
$->$ Multiplication same as with $F_p mod \Pi(x)$

Calculation for $F[3^2]$:
$F[3^2]$ = poly of degree $<= 1$ with coefficients from F_3
Here, $p = 3, n = 2$ and $\Pi(x) = x^2 + 1$
$F[3^2] = a_0 + a_1 * x$; a_i belongs to F_3

Possible values:

a_0	a_1
0	0
0	1
0	2
1	0
1	1
1	2
2	0
2	1
2	2

$F[3^2] = 0, x, 2x, 1, 1+x, 1+2x, 2, 2+x, 2+2x$

Addition verification:

+	0	x	2x	1	1+x	1+2x	2	2+x	2+2x
0	0	x	2x	1	1+x	1+2x	2	2+x	2+2x
X	x	2x	0	1+x	1+2x	1	2+x	2+2x	2
2x	2x	0	x	2x+1	1	1+x	2+2x	2	2+x
1	1	1+x	1+2x	2	2+x	2+2x	0	x	2x
1+x	1+x	1+2x	1	2+x	2+2x	2	x	2x	0
1+2x	1+2x	1	1+x	2+2x	2	2+x	2x	0	X
2	2	2+x	2+2x	0	x	2x	1	1+x	1+2x
2+x	2+x	2+2x	2	x	2x	0	1+x	1+2x	1
2+2x	2+2x	2	2+x	2x	0	x	1+2x	1	1+x

Multiplication mod ∏(x) verification :

*	0	x	2x	1	1+x	1+2x	2	2+x	2+2x
0	0	0	0	0	0	0	0	0	0
X	0	2	1	x	2+x	1+x	2x	1	1+2x
2x	0	1	2	2x	2	0	2	0	1
1	0	x	2x	1	1+x	1+2x	2	2+x	2+2x
1+x	0	2+x	1+2x	1+x	2x	0	2+2x	1	x
1+2x	0	1+x	2+2x	1+2x	0	x	2+x	2x	1
2	0	2x	x	2	2+2x	2+x	1	1+2x	1+x
2+x	0	2+2x	1+x	2+x	1	2x	1+2x	x	2
2+2x	0	1+2x	2+x	2+2x	x	1	1+x	2	2x

Hence, the field elements are found and the operations are also verified.

---

11) Find the additive and the multiplicative inverse of $x^3 + x + 1$ in $GF(2^4)$, with prime polynomial $= x^4 + x + 1$.

$GF(2^4) => 0, 1^4 + mod 2^4 * mod 2^4$

Additive inverse of $x^3 + x + 1$:

Let the additive inverse be 'a'

$$\begin{aligned} x^3 + x + 1 + a &= 0 \\ a &= -(x^3 + x + 1) \\ &= x^3 + x + 1 \end{aligned}$$

	q		
$x^4 + x + 1$		1	0
$x^3 + x + 1$	x	0	1
$x^2 + 1$	x	1	$-x = x$
1		$-x = x$	$1 – x * x = 1 + x^2$

So, additive inverse $= x^3 + x + 1$

---

12. Suppose that using commodity hardware it is possible to build a computer for about 200 that can brute force about 1 billion AES keys per second. Suppose an

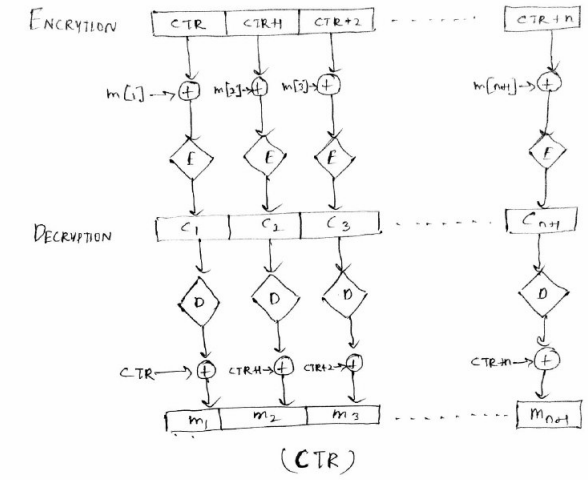
organization wants to run an exhaustive search for a single 128-bit AES key and was willing to spend 4 trillion dollars to buy these machines (this is more than the annual US federal budget). How long would it take the organization to brute force this single 128-bit AES key with these machines? Ignore additional costs such as power and maintenance.
Soln:
No of machines that can be bought = Amount allocated / cost of a single machine
$= 400trillion\$/200\$ = 2 * 10^{10}$

Total time taken to brute force the 128 bit AES key = Total no of keys / total no of keys that can be brute forced by a machine $= 2^{128}/10^9 persecond$
$= 3.4 * 10^{10}/10^9 (As 2^{128} \approx 3.4 * 10^{10})$
$= 3.4 * 10^{29} Secs$

Hence total time taken to brute force the 128 bit AES key with $2*10^{10}$ machines = Total time taken to brute force the 128 bit AES key / No of machines that can be bought
$= 3.4 * 10^{29} Secs/2 * 10^{10}$
$= 1.7 * 10^{19} Secs$

---

13. Let m be a message consisting of l AES blocks (say l = 100). Alice encrypts using randomized counter mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number l=2 is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted? Soln:



From the above pic it can be easily seen that the general equation for the cipher is,

$M_{n+1} = D(C_{n+1}) \oplus [CTR + n)]$

Using the general equation, we can say,

$M_{l/2-1} = D(C_{l/2-1}) \oplus [CTR + (l/2 - 2)]$

$M_{l/2} = D(C_{l/2}) \oplus [CTR + (l/2 - 1)]$

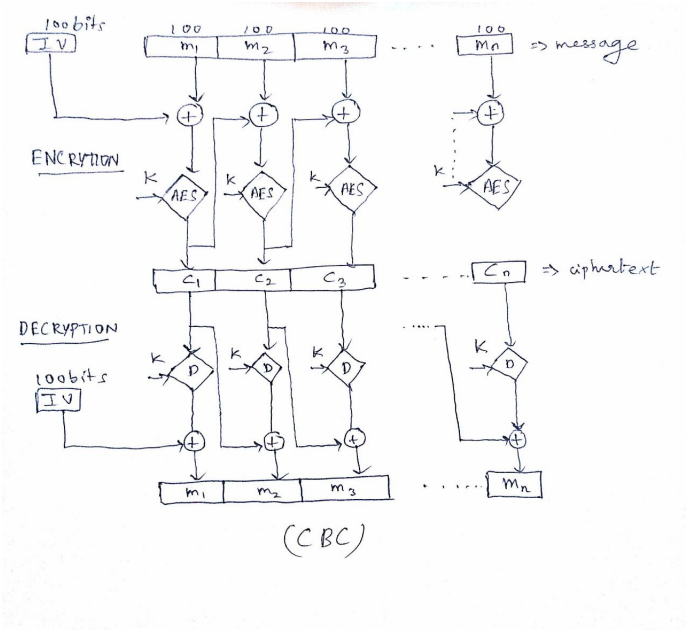
$M_{l/2+1} = D(C_{l/2+1}) \oplus [CTR + (l/2)]$

From the above 3 equations, if $C_{l/2}$ gets corrupted then only $M_{l/2}$ gets corrupted, so the no of plaintext blocks that gets corrupted is 1.

---

14. Let m be a message consisting of l AES blocks (say l = 100). Alice encrypts using CBC mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?

Soln:



From the above pic it can be easily seen that the general equation for the cipher is

$M_n = D(C_n) \oplus C_{n-1}$

Using the general equation, we can say,

$$M_{l/2} = D(C_{l/2}) \oplus C_{l/2-1} \quad (1)$$

$$M_{l/2+1} = D(C_{l/2+1}) \oplus C_{l/2} \quad (2)$$

From (1) and (2), when $C_{l/2}$ gets corrupted, $Ml/2$ and $M_{l/2+1}$ get corrupted, so the no of plaintext blocks that gets corrupted is 2.

---

15. Prove that block cipher with ECB mode is not semantically secure.
Soln:

To Prove:
Block cipher with Electronic code boot(ECB) is not secure
Proof:
let the message size be 256bits(32 characters)

If, $C_0 = AAAAAAAAAAAAAAAA \mid AAAAAAAAAAAAAAAA$
and $C_1 = AAAAAAAAAAAAAAAA \mid BBBBBBBBBBBBBBBB$

from the above two ciphertexts it can easily be seen that the 2 ECB encrypted messages have a common prefix, because ECB encrypts identical plaintext blocks into identical ciphertext blocks, it does not hide data patterns well.
Hence the attacker can gain infromation about the cipher and it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all.
Hence it is semantically insecure.

---

16. Find the minimum value of k (minimum no. of students) such that probability is greater than 0.5 that at least two people in a group of k-people have the same birthday? How it improves the attack on collision resistant property of the hash function.
Soln:

Let P(different) be the probability that all of them have different birthday and Ei be the event that ith person's birthday is different from all others.

Let M = 365
$P(different)$
$= P(E1 \wedge E2 \wedge ...Ek)$
$= P(EK/E1 \wedge E2... \wedge EK-1) * P(E1 \wedge E2...EK-1)$
$= P(EK/E1 \wedge E2... \wedge EK-1) * P(EK-1/E1 \wedge E2... \wedge EK-2) * ... * P(E2/E1) * P(E1)$
$= ((365-K+1)/365) * ... * ((365-2)/365) * ((365-1)/365) * (365/365)$
$= (1-((K-1)/365)) * ..... * (1-(2/365)) * (1-(1/365)) * (1-(0/365))$

For $x << 1, e^x \approx 1+x = exp(-((1+2+...+k-1)/365))$
$= exp(-(k*(k-1)/(2*365)))$

Probablity that atleast 2 out of k people will have same birthday
$= 1 - exp(-(k*(k-1)/(2*365))) \geq 1/2$

$\Rightarrow exp(-(k*(k-1)/(2*365))) \leq 1/2$
$\Rightarrow -k*(k-1)/730 \leq -1$
$\Rightarrow K \approx 1.2*\sqrt{365}$
$K \approx 22.92 \approx 23$
So required value of k is 23.

Hence complexity to find collision with probability more than $1/2$ is $O(M^{1/2})$ (where M = 365)

To achieve (probability = 1) for finding a pair of student having same birthdate, the time complexity is O(M) as we need 367 students to do that.

But to find collision with probability more than $1/2$ is $O(M^{1/2}) <= O(M)$, hence the hash is easier to attack with less key space. So, the attack gets improved on collision resistant property of the hash function.

---

17. Alice and Bob share a secret key of some private key system. Bob has a message he claims came from Alice and to prove this he produces a plaintext message and a ciphertext. The ciphertext decrypts to the plaintext under the secret key which Alice and Bob share. Please explain why this does not satisfy the requirements of non-repudiation of origin.
Soln:

Non-repudiation refers to a state of affairs where the author of a statement will not be able to successfully challenge the authorship of the statement or validity of an associated contract. The term is often seen in a legal setting wherein the authenticity of a signature is being challenged. In such an instance, the authenticity is being "repudiated".

Here the authenticity cannot be verified, because both Alice and Bob share the same private key. So, Bob could encrypt a plaintext and say that he has received it from Alice (and vice versa). Hence it does not satisfy the condition for authentication.

---

18. Show that 64-bit message digest is vulnerable to collision attack. Assuming that adversary can perform 2^{20} tests(hash values) per second.
Soln:

A hash function is said to be collision resistant when from given x, x' it is hard to find H(x) = H(x').

It is given that it is a 64-bit message, hence the total no of distinct hash values that can be produced is 2^{64}. Hence only for 2^{64} plaintexts, different hash values can be assigned. Once after that we must assign hash values that have been assigned already. Hence the $(2^{64}+1)^{th}$ plaintext gets a redundant hash value which results in collision.

The attacker will be able to find collision at $((2^{64}/2^{20})+1)^{th}$ second. Hence 64-bit message digest is vulnerable to collision attack.

---

19. What is the minimum and maximum number of padding bits that can be added to a message in SHA-512.
Soln:

a) The minimum length of padding is 0 and it happens when (-—M—-128) mod 1024 is 0. This means that —M— = -128 mod 1024 = 896 mod 1024 bits. i.e. the last block in the original message is 896 bits. We add a 128-bit length field to make the block complete.

b) The maximum length of padding is 1023 and it happens when (-—M—-128) = 1023 mod 1024. This means that the length of the original message is —M— = (-128-1023) mod 1024 or the length is —M— = 897 mod 1024. In this case, we cannot just add the length field because the length of the last block exceeds one bit more than 1024. So we need to add 897 bits to complete this block and create a second block of 896 bits. Now the length can be added to make this block complete.

---

20. Let $E : \{0\}^k \otimes B^n -> B^n$ be a block cipher, where $B = \{0,1\}^n$. View a message $M \rightarrow B^*$ as a sequence of l-bit blocks, $M = M[1]...M[m]$. Consider $MAC : \{0,1\}^k \otimes B^* \rightarrow B$. Show that following MAC are forgeable under chosen message attack.
a) Function MAC is defined by $MAC_k(M[1]...M[m]) = E_k(M[1]) \oplus ... \oplus E_k(M[m])$.
b) Here $l = n - 32$. Function MAC is defined by $MAC_k(M[1]...M[m]) = E_k(< 1 > ||M[1]) \oplus ... \oplus E_k(< m > ||M[m])$. $< i >$: is the 32-bit binary representation of the block index i.

a) Soln:

$MAC_k(M[1]...M[m]) = E_k(M[1]) \oplus ... \oplus E_k(M[m])$
Authentication rule : Verify(M,k,t)
Let, t' = MAC(m,k)
if (t == t') ⇒ accept it
else ⇒ reject it
For $M = M[1]||M[2]||M[3]||M[4]...||M[m]$, Let the corresponding tag be t
For $M' = M[2]||M[1]||M[3]||M[4]...||M[m]$, Let the corresponding tag be t'
Substituting M and M' in given MAC equation, we get,

$$MAC(M,k) = MAC_k(M[1]...M[m]) = E_k(M[1]) \oplus E_k(M[2]) \oplus ... \oplus (E_k(M[m]) \quad (1)$$

$$MAC(M',k) = MAC_k(M[2]...M[m]) = E_k(M[2]) \oplus E_k(M[1]) \oplus ... \oplus (E_k(M[m]) \quad (2)$$

It can clearly be seen that (1) and (2) are equal.

$\Rightarrow MAC(M,k) == MAC(M',k) \Rightarrow (t == t')$, which means that for 2 different plaintexts we get same ciphertexts and hence the MAC is forgeable(Selective forgery) under chosen message attack.

b) Soln:

$MAC_k(M[1]...M[m]) = E_k(<1>||M[1]) \oplus ... \oplus E_k(<m>||M[m])$.

Here selective forgery is not possible, but other attacks are possible Let us divide the message into 2 blocks,

Hence, $MAC_k(M[1]||M[2]) = E_k(<1>||M[1]) \oplus E_k(<2>||M[2])$.

Using this we can say that,

$$MAC_k(M[1]||M[2]) = E_k(<1>||M[1]) \oplus E_k(<2>||M[2]) \quad (1)$$
$$MAC_k(M[3]||M[2]) = E_k(<1>||M[3]) \oplus E_k(<2>||M[2]) \quad (2)$$
$$MAC_k(M[1]||M[3]) = E_k(<1>||M[1]) \oplus E_k(<2>||M[3]) \quad (3)$$

XORing (1), (2) and (3) results in,

$= E_k(<1>||M[3]) \oplus E_k(<2>||M[3])$

Which is the tag of (M[3],M[3]), a new message. Hence this MAC is forgeable under chosen message attack.

---

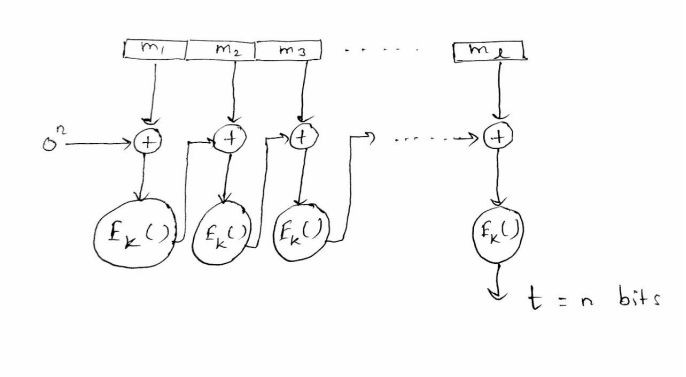
21. To make the message multiple of block length n, padding is required. If padding is 00...00 then show that it may lead to some kind of forgery under CMA.
Soln:

Given padding is 0^* i.e. 00....00,
$\Rightarrow H(m) = H(m||0^*)$
Let us take the block size to be 512, so $(m||padding) = x * 512$
If $|m| = 509$ then,
$H(m) = H(m||000)$
$H(m) = H(m||00)$
$H(m) = H(m||0)$
This implies that the hash function (Tag here) for different messages is the same when this padding is used. Hence collision exists and so 0^* should not be used. This comes under selective forgery under CMA as adversary is able to produce tag for a selected message.

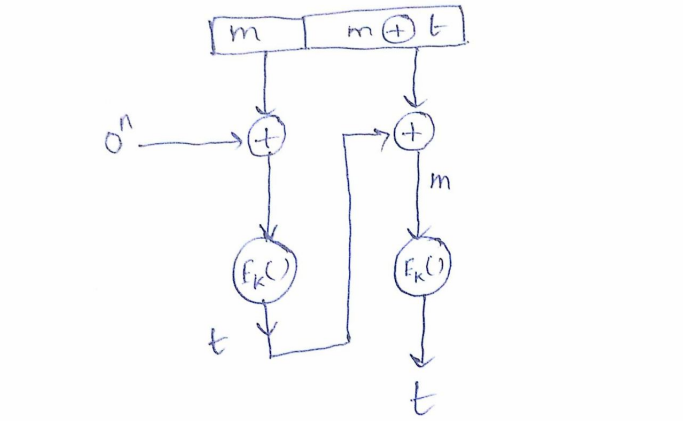
---

22. If basic CBC-MAC is used for variable number of blocks then show that basic CBC-MAC is vulnerable for some kind of attack.
Soln:

Let m be split into l blocks of size n $\quad (m = ln)$
$\Rightarrow m = m_1||m_2||m_3||.....||m_l$



Under chosen message attack(CMA) i.e. adversary knows m and corresponding t(Tag). And let us assume that the first block in m and the second block in $m \oplus t$



From the diagram above, it can clearly be seen that the tag for block $(m \oplus t)$ is also t, which means that both m and $(m \oplus t)$ have the same tag which makes basic CMA-MAC very vulnerable.
