ANNEXURE I

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Rajiv Gandhi Salai (OMR), Kalavakkam - 603 110

Register Number	1	9	5	0	0	1	0	7	1
Degree and Branch	B E Computer Science						Semester	V	
Subject Code and Name	UCS1505 - Introduction to Cryptographic Techniques								
Date	15/02/2022	Session: FN / ~~AN~~				No. of Pages Used		18	
Chief Superintendent's Signature / Facsimile	All Particulars given are verified Signature of the Hall Superintendent with date Name of the Hall Superintendent						For Office Use Only		

Date	15/02/2022	Session: FN / ~~AN~~
Subject Code and Name	UCS1505 - Introduction to Cryptographic techniques	
No. of Pages Used	18	

For Office Use Only

Date	15/02/2022	Session: FN / ~~AN~~
Subject Code and Name	UCS1505 - Introduction to Cryptographic Techniques	
No. of Pages Used	18	

For Office Use Only

Details of Marks Obtained										Grand Total (In Words)
Part A (2 Marks)		Part B (6 Marks)				Part C (10 Marks)				
Question No.	Marks	Question No.	[a] Marks	[b] Marks	Total Marks	Question No.	[a] Marks	[b] Marks	Total Marks	
1		11				16				
2						17				
3		12				18				
4						19				
5		13				20				
6						21				
7		14				22				
8						23				Grand Total
9		15				24				
10						25				
Total (A)		Total (B)				Total (C)				

Declaration by the Examiner: Verified that all the questions attended by the student are valued and the total is found to be correct.

Date	Name of the Examiner	Signature of the Examiner

Part-C

24) RSA: i) This is an assymmetric Cryptographic Scheme.

ii) RSA uses 2 algebraic structures.

For encryption and decryption it uses a commutative ring $R = \langle Z_n, +, \times \rangle$

For key-generation it uses a group. $G = \langle Z_{\phi(n)}^*, \times \rangle$

RSA Key generation:

i) Select 2 prime numbers 'p' and 'q' such that $p \neq q$.

ii) Multiply p and q to get n. $n = p \times q$.

iii) Calculate $\phi(n)$ using Euler Totient Function.

$$\phi(n) = (p-1)*(q-1)$$

iv) Select e (Public key) such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$

v) Calculate the private key d which is $e^{-1} \bmod \phi(n)$ by using Extended Euclid Algorithm.

vi) Public key = (e, n)

Private key = (d)

Given: $N = 33$, $e = 7$

N can be prime factorized into 3×11

$\therefore P = 3, q = 11$

$\therefore \phi(N) = (P-1)(q-1)$

$= (3-1)(11-1) = 2 \times 10 = 20$

195001071
B.E CSE

Finding Private/Secret key.

$d \leftarrow e^{-1} \bmod \phi(n)$

$\Rightarrow d \leftarrow 7^{-1} \bmod 20$

Finding $7^{-1} \bmod 20$ using extended Euclid's Algorithm.

GCD for 7 and 20

$20 = 7 \times 2 + 6$

$7 = 6 \times 1 + 1$

$1 = 7 - 6$

$1 = 7 - (20 - 7 \times 2)$

$1 = 7 \times 3 - 20$

$\therefore$ Inverse of 7 mod 20 is 3

$\therefore d = 3$

Public key = $(7, 33)$

Private key = 3

23) Diffie Hellman key exchange:

This is a key exchange method for securely exchanging cryptographic keys over a public channel.

This is an asymmetric key exchange algorithm for exchanging a shared symmetric secret key.

Working:

q is a prime number

$\alpha < q$ is a primitive root of q

that is $\alpha^i \bmod q = \{1, 2, \ldots, q-1\}$, $i = 0, 1, 2, \ldots, q-1$

α and q are public.

Alice

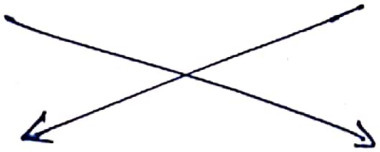
$X_A \leftarrow$ Private key of Alice

$Y_A = \alpha^{X_A} \bmod q$

Bob

$X_B \leftarrow$ Private key of Bob

$Y_B = \alpha^{X_B} \bmod q$



$K_1 = (Y_B)^{X_A} \bmod q$

$K_2 = (Y_A)^{X_B} \bmod q$

$$K_1 = K_2$$

Now Alice and Bob have a shared secret key.

Proof that $K_1 = K_2$

$$K_1 = (Y_B)^{X_A} \bmod q = (\alpha^{X_B})^{X_A} \bmod q = (\alpha^{X_A})^{X_B} \bmod q$$
$$= (Y_A)^{X_B} \bmod q = K_2.$$

S.Nivi 3

$q = 17$, $g = 4$ (primitive root)

$X_A = 3$, $X_B = 6$

Alice

$X_A = 3$

$Y_A = g^{X_A} \bmod q$

$= 4^3 \bmod q$

$= 64 \bmod 17$

$= 13$

$\therefore \boxed{Y_A = 13}$

Bob

$X_B = 6$

$Y_B = g^{X_B} \bmod q$

$= 4^6 \bmod q$

$= (4^3)^2 \bmod 17$

$= (13)^2 \bmod 17$

$= 169 \bmod 17$

$\boxed{Y_B = 16}$

$K_1 = (Y_B)^{X_A} \bmod q$

$= 16^3 \bmod 17$

$= 16^2 . 16 \bmod 17$

$= [(256 \bmod 17)(16 \bmod 17)] \bmod 17$

$= [1 . 16] \bmod 17$

$\boxed{K_1 = 16}$

$K_2 = (Y_A)^{X_B} \bmod q$

$= 13^6 \bmod 17$

$= (13^2)^3 \bmod 17$

$= (169)^3 \bmod 17$

$= 16^3 \bmod 17$

$= 16^2 . 16 \bmod 17$

$= [(256 \bmod 17) . (16 \bmod 17)] \bmod 17$

$= (1 . 16) \bmod 17$

$\boxed{K_2 = 16}$

$K_1 = K_2$

The shared secret key is exchanged successfully.

17) To Prove/Refute: An encryption scheme with message space M is perfectly secret if and only if for every probability distribution over M and every $c_0, c_1 \in C$,

$$P[C = c_0] = P[C = c_1]$$

The statement given above is wrong.

Proof:

Perfect Secrecy:

An encryption scheme (Gen, Enc, Dec) with message space M and ciphertext space C is perfectly secret if for every distribution over M, every $m \in M$ and every $c \in C$ with $P[C = c] > 0$, it holds that

$$P[M = m \mid C = c] = P[M = m]$$

i) This means that by observing the ciphertext, the adversary does not gain any information about the plaintext.

ii) This also implies that the message distribution and ciphertext distribution over their respective domains are independent of each other. But the statement given in the question implies that probability of all the ciphertexts that are generated due to encryption is equal. That is they are all uniformly distributed.

iii) Ciphertext uniformly distributed $\longrightarrow$ Perfect secrecy is true where as the converse is not.

That is

Ciphertext uniformly distributed $\longleftrightarrow$ Perfect secrecy is wrong

$\therefore$ The given statement is wrong.

Example:

Let the encryption scheme $\Pi = (Gen, Enc, Dec)$ be a perfectly secure encryption scheme.

Let $\Pi' = (Gen', Enc', Dec')$ defined as

$Enc'_k(m)$:

$c \leftarrow^{\Pi} Enc_k(m)$

B = Randomly choose number in the set $\{000, 001, 010, 011, 100, 101, 110, 111\}$

If $B = 000$:

return 011c

return 1||c

The $Dec'_k(m)$ uses $Dec_k(m)$ after ignoring the most significant bit in the ciphertext.

The ciphertext distribution of Π doesn't depend on the plaintext. It ~~may~~ seems that, also in case of Π' the ciphertext distribution doesn't depend of the plaintext as only a random bit is appended, which is independent of the message. Thus it is seen that Π' is perfectly secret.

However the ciphertext distribution in Π' is not uniform as the ciphertext is 6 times more likely to start with ~~an~~ a 1 than 0. This contradicts the given statement.

20) Substitution - Permutation Network

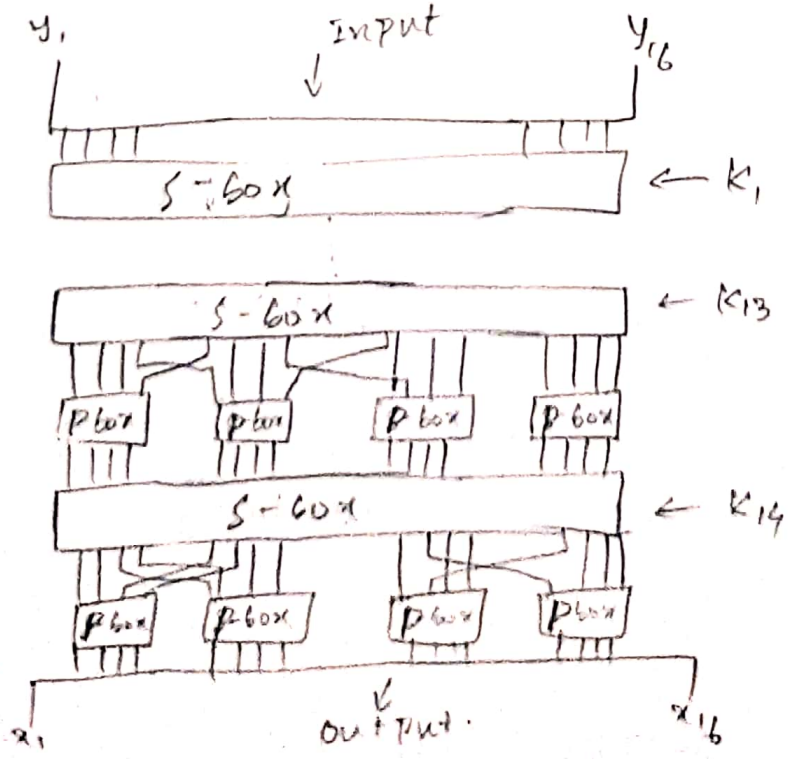
i) This is analogous to the SP boxes in DES algorithm
ii) used for creating the required confusion - diffusion paradigm

As the output of the SPN and the key is given, it is indeed possible to retrieve the input.

i) The retrieval process uses the key that has been used for encryption which is shifted multiple times to produce respective sub keys which can be used to retrieve the input.

ii) Every sub key is XORed in its corresponding layer which cascade to the initial layer which returns the input given to it.

iii) Permutations created in the S-box is reordered to reposition bits to their initial position.



195001071
B.E CSE

UCS1505 - Introduction to Cryptographic Techniques

18) Let A be an adversary that queries it oracle with 2 messages.

$m = m_1 || m_2$ and $m' = m_1' || m_2'$, where $m_1 \neq m_1'$ and $m_2 \neq m_2'$

Let $t = t_1 || t_2$ and $t' = t_1' || t_2'$ be the respective responses from its oracle.

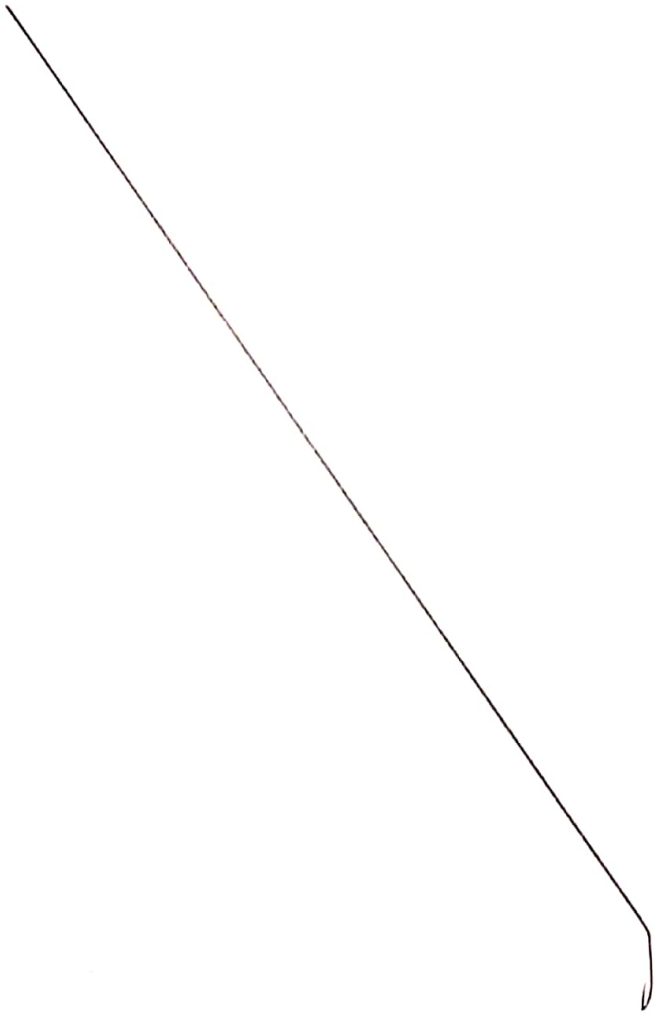
A then outputs the message $\bar{m} = m_1 || m_2'$ and tag $\bar{t} = t_1 || t_2'$

By the definition of MAC, $\bar{t}$ is a correct tag for $\bar{m}$ and thus $< F_k(m_1), F_k(F_k(m_2)) > = 1$ always.

Since $m_1 \neq m_1'$ and $m_2 \neq m_2'$ we have that $\bar{m} \notin Q$.

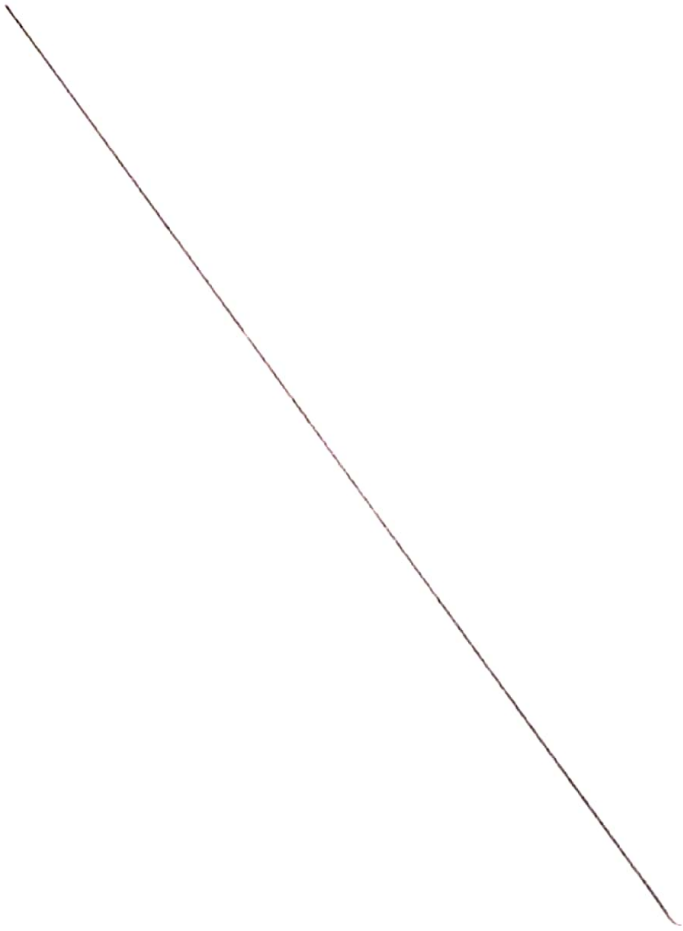
Thus A succeeds with probability 1 and the scheme is not secure.

UCS1505 – Introduction to Cryptographic Techniques



195001071
B.E CSE

UCS1505 - Introduction to Cryptographic Techniques



Sanjith

195001071
B.E CSE

UCS1505 - Introduction to Cryptographic Techniques

Part-A

1) Guarantees provided for security protocols

i) Confidentiality: Confidential information must be protected against malicious actions.

ii) Integrity: Information needs to be changed constantly, only by authorised entities and through authorised mechanisms

iii) Availability: Information created and stored by an organisation must be available to all authorised entities.

2) If an adversary has unbounded computation power, then it is easier for them to break a cipher even using a brute-force attack. For a bounded adversary it would take a longer time to execute bruteforce attacks.

3) Properties of Hash function:

a) Strong collision resistance

b) Fixed Digest length

c) Easy to compute.

d) Pre-Image resistance: Hard to reverse a hash function.

4) Bob should construct the tag from the message sent to him to verify the ~~authen~~ integrity of the message. If he tries to choose a tag, then it might not match the message that is sent to him.

S.Nithya 11

5) Avalanche effect: A small change in either the plaintext or the key should produce a significant change in the ciphertext.

6) The one-way functions (OWFS) is hard to invert for the adversary as he has to invert the intercepted text to get the plaintext. One-way functions are hard to invert without the key used for encryption.

7) Attacks that are possible in DHKE:

i) Man-in-the-middle Attack.

ii) Timing attack.

iii) Brute force attack.

8) If Alice and Bob use a shared key, the public keys of Alice and Bob are exchanged and then the shared secret key is generated. This leaves no room for the adversary to form the shared secret key due to the discrete log property.

9) Let the (public key, private key) pair of Bob be (P_K, S_K).
For Alice to send a message to Bob, she encrypts the message using Bob's public key and when Bob receives this, he decrypts it using his secret key/private key to get the plaintext back.

10)

Private key	Public key.
i) Kept secret and is known only to the receiver of the message.	i) Available publicly and is known to everyone.
ii) Used to decrypt a cipher text	ii) Used to encrypt a plain text.
iii) Used to generate signature	iii) Used to verify signatures.

Part - B

11) A simple crypto system is defined by a tuple $(\mathcal{C}, \mathcal{P}, K, \mathcal{E}, \mathcal{D})$

$\mathcal{C}$ - Cipher text

$\mathcal{P}$ - plain text

K - key space

E_K - encryption rule. $E_K \in \mathcal{E}$, $E_K : \mathcal{P} \to \mathcal{C}$

D_K - Decryption rule $D_K \in \mathcal{D}$, $D_K : \mathcal{C} \to \mathcal{P}$

$$D_K(E_K(x)) = x \quad \forall x \in \mathcal{P}.$$

Private key encryption:

A private key encryption scheme is defined by a message space $\mathcal{M}$ and 3 algorithms (Gen, Enc, Dec)

i) Gen (key-generation algorithm): outputs $k \in K$.

ii) Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$. as input and outputs ciphertext c.

$c \leftarrow Enc_k(m)$

iii) Dec (Decryption algorithm) : Takes key k and ciphertext c as inputs and outputs m or "error"

$$m \leftarrow Dec_k(c)$$

For all $m \in M$ and k output by Gen

$$Dec_k(Enc_k(m)) = m.$$

12) In MAC 2 users have a shared key.

i) When a message needs to be sent, the ~~used~~ user sends the MAC tag(t) along with it.

ii) The MAC is generated from key k and message m.

iii) After receiving (t, m) the receiver verifies by generating the MAC tag(t') using the received message. If $t' = t$, B verifies that the message sent was valid and non-corrupted.

Given MAC is uniformly distributed, all messages in the message space will have MAC which has equal distribution of 0's and 1's.

Let n be the length of the message and using XOR to find MAC.

Let key k = 100

M = 1 XOR k = 1 XOR 100 = 101

M = 0 XOR k = 0 XOR 100 = 100

The key size is taken as 3 since anything more than max(len(m), len(t)) will not affect the tag.

For $n=2$, $k=100$

M		k		t	
00	⊕	100	=	100	} uniformly distributed.
01	⊕	100	=	101	
10	⊕	100	=	110	
11	⊕	100	=	111	

For $n=3$, $k=100$

M		k		t	
000	⊕	100	=	100	} uniformly distributed.
001	⊕	100	=	101	
010	⊕	100	=	110	
011	⊕	100	=	111	
100	⊕	100	=	000	
101	⊕	100	=	001	
110	⊕	100	=	010	
111	⊕	100	=	011	

Tag space is entirely used up for $n=3$. Hence increase in no. of message will cause collision in the given tag space.

13) A pseudorandom string is a string that looks like a uniformly distributed string, as long as the entity that is looking (adversary) runs in polynomial time. Pseudorandomness is a computational relaxation of true randomness.

NO fixed string is said to be pseudorandom, rather Pseudorandomness refers to a distribution on strings.

Let the distribution D over strings of length l is pseudo-random, which means D is indistinguishable from the uniform distribution over strings of length l.

It is infeasible for any PPT algorithm to tell whether it is given a pseudorandom ~~random~~ string or an l-bit string chosen uniformly at random

If cipher text looks random, ~~then it is clear~~ the adversary can't learn any information from it about the plain text.

This is similar to perfect secrecy where the cipher text is uniformly distributed and reveals no information about the plain text.

A pseudorandom generator is a deterministic algorithm that receives a short truely random seed and stretches it into a long string that is pseudo random.

14) DHKE is used for generating a secret shared key. It is an asymmetric key exchange algorithm for exchanging symmetric keys

~~Let~~ If Alice and Bob wants to send and receive messages they must first establish a shared secret key.

Alice and Bob have they own set of private keys.

q = Prime modulous.

g = primitive root of q. $g < q$ and

$g^i \bmod q = \{1, 2, \ldots, q-1\}$

$i = 0, 1, 2 \ldots, q-1$

g and q are Public.

Alice

X_A – Alice's private key

Y_A = Alice's Public key

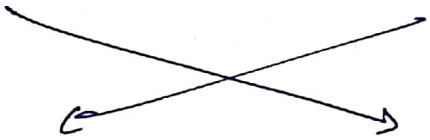
$Y_A = g^{X_A} \bmod q$

Bob

X_B – Bob's private key.

Y_B – Bob's Public key.

$Y_B = g^{X_B} \bmod q$



$K_1 = Y_B^{X_A} \bmod q$

$K_2 = Y_A^{X_B} \bmod q.$

$K_1 = K_2$

Proof:

$$K_1 = Y_B^{X_A} \bmod q$$
$$= (g^{X_B})^{X_A} \bmod q$$
$$= (g^{X_A})^{X_B} \bmod q$$
$$= (Y_A)^{X_B} \bmod q$$
$$= K_2$$

Now Alice and Bob have a shared secret key which they can use for encryption and decryption.

UCS1505 - Introduction to Cryptographic Techniques

15) Public key encryption :

A public key encryption scheme is a triple of probabilistic polynomial time algorithms (Gen, Enc, Dec) such that.

i) The key-generation algorithm Gen takes as input the security parameter 1^n and outputs a pair of keys (P_K, S_K)

$P_K \leftarrow$ Public key

$S_K \leftarrow$ Private key

length of P_K and S_K is same $= n$.

ii) The encryption algorithm (Enc) : Takes as input a public key P_K and a message m for message space. It outputs a ciphertext C, $C \leftarrow Enc_{P_K}(m)$.

iii) Decryption algorithm (DEC) : The deterministic algorithm that takes as input a private key and the ciphertext C and outputs a message m or a special symbol $\perp$ in case of failure.

$m := Dec_{S_K}(c)$

$Dec_{S_K}(Enc_{P_K}(m)) = m \quad \forall m \in$ message space.