SL.NO:CAA176805

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

76.5

THEORY EXAMINATIONS

Register Number	205001085		
Name of the Student	V. Sabarivasan		
Degree and Branch	BE CSE	Semester	V
Subject code and Name	UCS1505 Introduction to Cryptographic Techniques.		
Assessment Test No.	I	Date	21/9/2022

Details of Marks Obtained									
Part A		Part B				Part C			
Question No.	Marks	Question No.	(a) Marks	(b) Marks	Total Marks	Question No.	(a) Marks	(b) Marks	Total Marks
1	2	7	3		3	10			
2	2					11	8		8
3	0	8	5		5				
4	2					12	8		8
5	2	9	4		4	13			
6	2								
Total (A)	10	Total (B)			12	Total (C)			16
Grand Total (A+B+C)	38				Marks (in words)	Three	eight		
Signature of Faculty									

SSN

PART-C

(14)

$P(x=a) = 1/2$; $P(x=b) = 1/3$, $P(x=c) = 1/6$

$P(K=k_1) = P(K=k_2) = P(K=k_3) = 1/3$

$P = \{a, b, c\}$; $C = \{1, 2, 3, 4\}$.

⇒ To prove perfectly secrecy:

$$P(Enc_K(m_0)) = C = P(Enc_K(m_1))$$

⇒ $P(Enc_K(a)) = P(C=c / M=a)$

$= P(M=a) \cdot P(K=k_1) + P(M=a) \cdot P(k_2)$

$+ P(M=a) \cdot P(k_3)$

$= 1/2(1/3) + 1/2(1/3) + 1/2(1/3)$

$= 1/2 [1/3 + 1/3 + 1/3] = 1/2$ //

⇒ $P(Enc_K(b)) = P(C=c / M=b)$

$= P(M=b) \cdot P(K=k_1) + P(M=b) \cdot P(k_2)$

$+ P(M=c) \cdot P(K=k_3)$

$= 1/3(1/3) + 1/3(1/3) + 1/3(1/3)$

$= 1/3 [1/3 + 1/3 + 1/3] = 1/3$ //

$\Rightarrow P[Enc_k(c)] = P(C=c/M=c)$

$= P(M=c) \cdot P(K=k1) + P(M=c) \cdot P(K=k2)$
$+ P(M=c) \cdot P(K=k3)$

$= 1/6\,(1/3) + 1/6\,(1/3) + 1/6\,(1/3)$

$= 1/6\,(1/3 + 1/3 + 1/3) = 1/6$ //

$\Rightarrow$ By checking with the values provided in encryption matrix and the probability results which we obtained, it is proved that the cryptosystem built is perfectly secure.

(12)

$P[M=kim] = 0.5$; $P[M=ann] = 0.2$;

$P[M=boo] = 0.3$; $C = DGAQ$?

Sol:

We can infer that $C = DGAQ$ can be obtained from messages $M=ann$ with $K=3$ or from $M=boo$ with $K=2$.

$$P(C=DGAQ / M=m)$$

$$= P(M=ann) \cdot P(K=3) + P(M=boo) \cdot P(K=2)$$

$$= 0.2(1/26) + (0.3)(1/26)$$

$$= (0.2 * 0.3)(1/26) = 0.0192 \text{ //}$$

⇒ Applying Bayes' Theorem,

$$P(M=m / C=DGAQ)$$

$$= \frac{\sum P(C=DGAQ / M=m) \cdot P(M=m)}{P(C=c)}$$

$$= \frac{0.0192(0.2) + 0.0192(0.3)}{0.2 + 0.3}$$

$$= 0.4 \;//$$

To prove: for every encryption scheme that is perfectly secret, it holds that for every distribution over message space $\mathcal{M}$, every $m, m' \in \mathcal{M}$ and every $c \in C$: $P(M=m / C=c) = P(M=m' / C=c)$

$\Rightarrow P(C=c / M=m)$

$= P(Enc(m)=c / M=m)$

$= P(Enc(m)=c) \;//$

$\Rightarrow$ We know that,

$$\frac{P(C=c / M=m)}{P(M=m / C=c)} = \frac{P(C=c)}{P(M=m)}$$

$\Rightarrow P(Enc(m) = c)$

$= P(C = c / M = m)$

$= P(C = c)$ //

Similarly,

$\Rightarrow P(M = m) (C = c) = P(M = m)$ //

$\Rightarrow P(Enc(m) = c)$

$= P(C = c / M = m)$

$= P(C = c)$

$= P(C = c / M = m')$

$= P(Enc(m') = c)$ //

∴ For every $c \in C$,

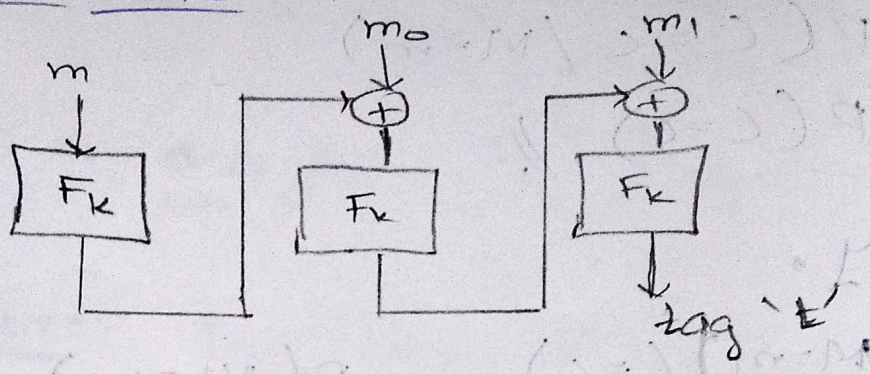
$\Rightarrow P(Enc(M = m)) = c = P(Enc(M = m'))$

$\Rightarrow P(M = m / C = c) = P(M = m' / C = c)$

∴ Hence, proved.

PART-B

(9) CBC MAC:



⇒ CBC MAC is defined as Cipher Block Chaining Message Authentication Code.

⇒ It is a technique in which message authentication codes are generated from block ciphers.

⇒ CBC MAC does not use Random IV.

⇒ The tag which is received from the final cipher block is considered to be the final tag.

⇒ It consists of three algorithms of polynomial time bound.

(i) Gen : It accepts (1^n) as input and outputs a key $k \in \{0,1\}^*$

(ii) Mac(k,m) : It accepts the key k and message 'm' as inputs, and generates a tag 't' for that message

(iii) Vrfy (m, t) : It accepts tag 't' and message 'm' and key 'k' as inputs and verifies where the tag is valid or not for that message. If valid it returns 1, otherwise 0.

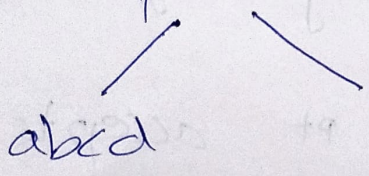
⇒ $Pr\left[Vrfy\left(Mac(k,m)\right)\right] = 1$

⇒ In MAC algorithm the message is passed as $m_0, m_1, \ldots m_n$ of length n.

⇒ The messages $m_0, m_1, \ldots m_n$ are passed to next cipher block with XOR operation.

---

(8) Users password



⇒ The user has used shift cipher to encrypt his passwords.

⇒ Shift Cipher is nothing but shifting the alphabets of a string for k places forward or backward.

⇒ There are only 26 alphabets available and also the length of the both passwords are too small.

⇒ So the attacker can use brute force method, to decrypt the cipher by ~~moving~~ shifting the alphabets of the ~~en~~cipher in reverse order.

⇒ Shift cipher can be decrypted easily and is not safe, since it has a fixed algorithm which is known to everyone.

⇒ Shift cipher algorithms:

i) Gen : generate a key $\in$ {1, 2, 3, ... 26} ~~a, b, c, ... z~~

ii) $Enc_k(M=m)$: gets key $k \in K$ and message 'm' as input.

$Enc_k(M=m) = \{c \mid c := m + k \bmod 26\}$

iii) $Dec_k(C=c)$: gets key and ciphertext as inputs and provides the message as output.

$$Dec_k(C=c) = \{m \mid m := c - k \bmod 26\}.$$

⇒ Hence, the attacker can easily determine the user's password.

(7)

Given cipher text : EVIRE

⇒ Shift cipher decryption :

EVIRE → (instead)

DUHRD → (key = 1)

CTGQC → (key = 2)

BSFPB → (key = 3)

ARENA → (key = 4)

→ At k=4, the message is decrypted as 'ARENA'. Hence Antony will go to Coliseum (which is the arena)

→ $Dec_k(C=C) = \{m | m = c - k \bmod 26\}$ //

PART-A

(1) Mono-alphabetic substitution cipher

i) Gen:

~~accepts~~ generates key 'k' from $\{A, B, C, D, \ldots Z\}$.

ii) Enc(M=m):

accepts message 'm' and key 'k' $\in K$ and provides a single alphabet as ciphertext output.

iii) Dec(C=c):

accepts ciphertext 'c' and key 'k' $\in K$ and provides the message as output.

Eg:

M =	A	B	C	D	E	F	G	H	I	J	K	L		
C =	L	K	J	I	A	H	B	G	C	F	D	E		
M =	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
C =	V	O	Z	M	N	T	U	P	Q	R	S	W	X	Y

---

(2)

Symmetric ciphers:

No. of keys = 1.

Assymmetric ciphers:

No. of keys = 2

---

(4) Kerckhoff's principle:

The principle states that even if the encryption scheme is known by the adversary, the main secret element is the key. The key should be made secure in such a way that adversary cannot read the message even though he has unlimited computation power.

(5) Encryption:

This algorithm accepts key and message and inputs and provides a cipher text as an output. It also has a decryption algorithm.

MAC:

In this technique, messages are sent via message authentication codes which means messages are sent along with a tag. Verify(t,m) algorithm is used to verify the tag.

Hash functions:

In this algorithm, accepts message and key as input and generates a hashcode via a hash function. Each hashcode for a message will be unique.

(6) Properties of hash function:

⇒ Messages are passed as inputs to the hash function and it generates a hash code equivalent to PT.

⇒ Same messages will not get the same hashcode again.

⇒ Hash functions can be changed according to the message and keys which leads to unique hash codes.

---

(3) Key = cafe.

Ciphertext = YEGPJIREDOZXOE
OIUTLNVHIRYWZH
PJREOENFIKACTF
TRISHKFORUMCAT

2) In symmetric cipher, only one key is needed for encryption at the sender side and decryption at receiver side.

In asymmetric cipher, 2 keys are required for 2 people to communicate. The sender and receiver both have a public and a private key.

3)

V	E	Q	P	J	I	R	E	D	O	Z	X	O	E
C	A	F	E	C	A	F	E	C	A	F	E	C	A
T	E	L	L	H	I	M	A	B	O	U	T	M	E

Rough

L M N O P Q R
A B C D E
T U V W X Y Z

Assume that each letter is given a no, in the form A = 0, B = 1, ... Z = 25.

In order to decrypt the cipher, we must subtract the key from the ciphertext. For ex: C = 2, V − C = T.

Here, plaintext = TELLHIMABOUTME.

4) Kerchoff's principle states that:

> A cipher method must not be required to kept a secret, and must be able to fall into the hands of the enemy without inconvenience.

According to kerchoff's principle, the encryption method need not be kept secret. However, the key used must be kept secret.