# Assignment 2

## problem 1

When message m1 is not equal-length with message m2, if we output the ciphertext c1 and c2 which their length is not equal, we can easily know the mapping relation of ciphertext and plaintext.

So we can define the length of ciphertext which the algorithm Enck output is *l(n) ,* it need the length of key which the algorithm Gen output is *l(n), So we need a Pseudorandom Generator to get a key which length is l(n).*

*For expanding the message m, we can choose the first block of m to show the length of m, and fill the rest part with 0. According to this way, we can know the length.*

*this is the formal description of scheme:*

*Gen: on input 1 n ,choose uniform and output it as the key*

*Enc: on input a key and a message output the m` and ciphertext*

*Dec:* *on input a key and a ciphertext, output the message*

*According to Definition 3.16, if G is a PRG, this scheme is coa-secure, and it satisfies Definition 3.8*

## problem 2

### (a)

G`(s) is a pseudorandom generator. Because we change the length of s in polynomial, and this change is neglect for this PRG algorithm.

**PROOF:** Because G(s) is PRG, so there is an N, for any number c and all *n > N, f(n) <* **.** We can set *p`(n) = 2c\* nc is a polynomial , let N` =N, when n > N`:*

so is neglect for PPT.Let D is an PPT distinguisher, so:

|

= |

*Because negl`(n) is neglect, so G(s)` is a PRG.*

### (b)

G`(s) is not PRG. For , the output , so we just check whether r = for to judge the scheme is PRG or TRG. The success probability is

### (c)

G`(s) is not PRG. To simplify the problem，we consider the , so ,

In this way, will not change in the procedure. So

so for the output r, assume the length of r is 2l(n), so we just check:

the success probability is

## problem 3

### (a)

I think *F`k(x)* is not PRF. when k is fix, we can see *F*k as a PRG:

So

for output r, we can check

If this holds, *F`k(x)* is not PRF.

### （b）

## problem 4

F is a PRF if for all probabilistic polynomial-time distinguisher D, there is a negligible function *negl* such that:

so

So:

According to the above proof, we can know that this PRF definition is equivalent to Definition 3.5.

## problem 5

## problem 6

### (a)

a is not CPA-secure but COA-secure. (a) if a standard private-key encryption scheme based on PRG. So according to Theorem 3.18, it is COA-secure for polynomial-time distinguisher.

(a) is not CPA-secure because when key is fix, we will get the same ciphertext for the same plaintext. So in training phase, we can send m0 to challenge to get the c0, and then we send the m0 , m0 to get c. just check whether c equals to c0 we can output b = 1 or b = 0.

so

## (b)

(b) is not CPA-secure but COA-secure. Because the r is fixed and r = 0n , so when we get the key , we get the result of Fk(0n) . So F is same as G. According to (a), we know it is COA-secure.

(b) is not CPA-secure. Because the r is fixed and r = 0n so we use the same key to encrypt message. The reason is same as (a). So in training phase, we can send m0 to challenge to get the c0, and then we send the m0 , m0 to get c. just check whether c equals to c0 we can output b = 1 or b = 0.

### (c)

## problem 7

We can construct an adversary A to distinguish the CBC:

a) Query phase: A send message m = 0n-1 1 to challenge and get ciphertext <IV,c >

b) if IV is odds then output b = 1 or 0 randomly.

c) If IV is even then send m0 = 0n and arbitrary bits m1 and receive the answer <IV+1,>

d) if c = c` ,output 1 else output 0

At first, we know that if IV is even, ,So:

Accord to the deduction process, we know that if IV is odds, the success probability is , if IV is even, we can check the whether c equal to c1 to know which message is encrypted. The success probability is . So the total probability is .Obviously, this scheme is not CPA-secure.

## problem 8

For simplify, assume that we transmitted ciphertext and receiver receive

1) CBC mode. If dropped ciphertext block, the receiver will not decrypt the ciphertext correctly. For example, we want to decrypt c3 .So:

Because the has lost, we will replace with wrongly. Obviously, we get the wrong message .

2) OFB and CTR mode. These two mode is different with CBC ,because computation of pseudorandom stream is independently of the actual message encrypted. So even some ciphertext is lost, we also can calculate the mapping pseudorandom stream of ciphertext. So we can decrypt the ciphertext correctly.

## problem 9

a) I think there is not strict for .According to the theorem 3.29, we know that if F is PRF, CTR mode is a CPA-secure. Because F is a PRF, CTR mode is CPA-secure without the strict of

### b)

for CTR mode, we can not use a ctr more than once. because:

So we need the block length .

## problem 10

### (a) & (b)

Actually, there is not essentially different for the scheme in P11 and P14 of 2019A\_−L6-BlockOP\_v1.pptx. So we can use the same attack to break them.

We assume that the IV is increased by 1 each time a message is encrypted. We have proof that it is not CPA -secure in **problem 7.** Just sending the message m =