Cryptography Engineering Midterm

110550126 曾家祐

# Question 1 :

# 

# The output of my python program is

# “The secuet message is: WhZn using a stream cipher, never use the key more than once”

# The true plaintext answer may be

# “The secret message is: When using a stream cipher, never use the key more than once”

# Question 2 :

# First we use the plaintext and ciphertext by xor to calculate the key.

# From the key we can get the ciphertext of pur plaintext.

# (we need to convert the string into ascii code form)

# Ans: 9e1c5f70a65ac519458e7f13b33

# 一張含有 文字 的圖片 自動產生的描述

# 一張含有 文字 的圖片 自動產生的描述

# Question 3 :

# Ans: (C) 11, (E) 6, (G) 26, (H) 1

# Since key 25 is right of the key 0 🡪 we can safely include all elements under left of key 0 (key 1)

# Since key 25 is left of the key 2 🡪 we can safely include all elements under right of key 2 (key 6)

# Since key 25 is right of the key 5 🡪 we can safely include all elements under left of key 5 (key 11)

# Since key 25 is left of the key 12 🡪 we can safely include all elements under right of key 12 (key 26)

# Ans is 1, 6, 11, 26

# 一張含有 圖表 的圖片 自動產生的描述

# Question 4 :

# Ans: (C) n

# Since the key need to be encrypted under the node on the path from node to the target, and the length of the path is n.

# Question 5 :

# First, start from c = mk mod p we get mk-c = pX

# 🡪mk = pX+c -> k = (px+c)/m

# We have Pr[E(k,m) = c] = 1

# From m0 m1

# We have Pr[E(k, m0) = c] = Pr[E(k, m1)=c]

# This cipher provides perfect secrecy

# The definition of perfect secrecy:

# 一張含有 文字, 人員, 室內, 螢幕擷取畫面 的圖片 自動產生的描述

# The definition of semantic security

# For one-time pad:

# E(k, m) = k XOR m = c => c XOR m = k XOR m XOR m = k , k=1.

# We can know that it is one-to-one. So we can prove that OTP is perfect secrecy

# For Semantic Secure:

# Since we can use random generate key and we can not get the plaintext information from ciphertext.

# The key is generated by random, and the key will only be used once.

# So there is no relation to statistical analysis

# No, if we have a public key, the attacker can try to encrypt the message by the public key. The attacker can try from 1-bit,2-bit … and looking for a matching ciphertext to encrypt the message.

# Question 6 :

# From Xi = aX(i-1) +b mod p we can get

# X2-X1 = a(X1-X0) mod p

# a = (X2-X1)/(X1-X0 mod p

# b = X1 – aX0 mod p

# so we found the definition formula and may predict the remaining sequence

# It is not secure to use the congruential generator as the keystream generator for a stream cipher, since if the attacker know about some sequential part, the attacker may calculate the part after it.

# Since attacker already know a ,b and p from Xi = aX(i-1) +b, we only need one bit we can calculate the rest of all

# The attacker need three bit to form

# X1 = aX0 – b mod p

# X2 = aX1 – b mod p

# And use this two to calculate a and b, and can use a,b ,p to calculate the rest.

# Question 7 :

**Ans: (D)**

# If N = p\_1 \* p\_2 \*…\* p\_k, = (p\_1 -1) \* (p\_2 -1)\*…\*(p\_k -1)

# Question 8 :

# N = 105 = 2\*5\*7 🡪 φ(N) = (3-1) x (5-1) x (7-1) = 2 x 4 x 6 = 48

# To find d, we can start from 13 since 13 is coprime to 48.

# We can use Extended Euclidean Algorithm to find the key.

# 48 = 3\*13+9

# 13 = 1\*9+4

# 9 = 2\*4 +1

# 🡪

# 1 = 9-2\*4

# 1 = 9-2\*(13-1\*9)

# 1 = 3\*9 – 2\*13

# 1 = 3\*(48-3\*13) – 2\*13

# 1 = 3\*48 – 11\*13

# 🡪

# d = -11 mod(48)

# -11 +48 =37(add 48 to make it positive)

# Ans : the key is 37

# (I also use Extend Euclidean Algorithm to write a program to solve this problem)

# Below is the code.

# 一張含有 文字 的圖片 自動產生的描述

# Question 9 :

# Ans:

# 20814804c1767293bd9f1d9cab3bc3e7ac1e37bfb15599e5f40eef805488281d

# 一張含有 文字 的圖片 自動產生的描述

# 

# Question 10 :

**Ans: (A)**

**(C)**

**These two is hard to compute**

**(B)**

**(D)**

**B and D can be easy to solve.**