Problem Set 1

Andrew Kaufman
SID: 998048873

April 22, 2016

PROBLEM 5

(a)

(b)

(c)

PROBLEM 6

The following probabilities were obtained after running the RC4 algorithm for 100000 iterations:

0: 0.00815
1: 0.00386
2: 0.00393
3: 0.0036
4: 0.00416
5: 0.00393
6: 0.00417
7: 0.00374
8: 0.00368
9: 0.00384

A simple adversary can attempt to distinguish RC4 output from truly random bits by looking at the second byte of the output and outputting a 1 if the second byte is 0, or outputting a 0 otherwise. If the adversary returns a 1 the output is interpreted as true RC4 output The reason we have the adversary doing this is because the probability of the second byte being 0 is much larger than the probabilities of the other possible

values. If the second byte is a 0, it is more probable to be real RC4 output.

The advantage of the adversary can then be defined as follows:

$$Adv(A) = \left[\Pr[A(F) = 1] - \Pr[A(G) = 1]\right]$$

where F denotes the algorithm for RC4 output and G denotes the algorithm for random bits.

The advantage of the adversary that we described would then be:

$$Adv(A) = \frac{2}{256} - \frac{1}{156} = \frac{1}{256}$$

```
# Andrew Kaufman
# 998048873

# run using Python3

import math
import random
import string


def keygen():
    key = []
    for i in range(16):
        byte = random.randint(0, 255)
        key.append(byte)

    return key

def init(key):
    length = len(key)
    S = list(range(256))

    j = 0
    for i in range(256):
        j = (j + S[i] + key[i % length]) % 256
        S[i], S[j] = S[j], S[i]

    return S

def stream(S):
    i = 0
```

```
    j = 0

    K = []

    for _ in range(2):
        i = (i + 1) % 256
        j = (j + S[i]) % 256

        S[i], S[j] = S[j], S[i]
        K.append(S[(S[i] + S[j]) % 256])

    return K


def main():
    table = [0 for i in range(10)]

    for i in range(100000):
        key = keygen()
        S = init(key)
        K = stream(S)

        if 0 <= K[1] <= 9:
            table[K[1]] += 1

    sum = 0
    for j in range(10):
        table[j] /= 100000
        print(str(j) + ": " + str(table[j]))

if __name__ == '__main__' : main()
```

PROBLEM 7