Exercise - Symmetric Key Cryptography

Summer School on Post-Quantum Cryptography 2017

June 20, 2017

1 Classic Ciphers

Consider a cipher where we substitute each letter of the alphabet with a different letter of the same alphabet. We call such a cipher a *substitution cipher*. In this case the secret key K is now a list of theses substitutions, for instance $(A \to T, B \to R, C \to D, \ldots)$. You can also see this a block cipher with a very small block size.

Exercise 1.
How many possible secret keys K are there for this encryption scheme?

Exercise 2.
The following text was encrypted with a random substitution cipher. The text is in English and only contains the letters from A to Z. Can you recover the original text?

```
ANXYIWNHBPGNWPHURNWYDEFNYTPGGNAUHIZGPAYXCBUHQGRWYELNDPIUNWAR
TAHAYZZRDWRWPGNQHWPBYQHIWGHAARPRDGHAARPHDGNRWYTWHXHDWRENUNEY
ARPWPUYDZNGHFNLNUPGNBYXPPGYPGNUNXNRLNWLRWRPWYPGRWGHIWNAYZNDE
BUHQNTLNWEFYULNWYDEPGNFRVYUEZYDEYTBQYCNGRQPGNHAMNXPHBWHQNWTR
ZGPWIWORXRHDRDYEERPRHDNLNUWRDXNARTAHXYQNAYXCPHPGNWGRUNFRPGPG
NURDZFGRXGGNGYWCNOPWNXUNPBUHQDNYUTSNLNUSHDNGNGYWDHPWNNQNEPHY
                              ZNYPYTT
```

2 Block Ciphers

The *Data Encryption Standard* (DES) and *Advanced Encryption Standard* (AES) are two of the most important block ciphers. The DES has a key size of 56-bit and block size of 64 bits while the AES has a key size of 128-, 192- or 256 bits and a block size of 128 bits. In the following we will look at how expensive it is to recover the secret key using exhaustive search.

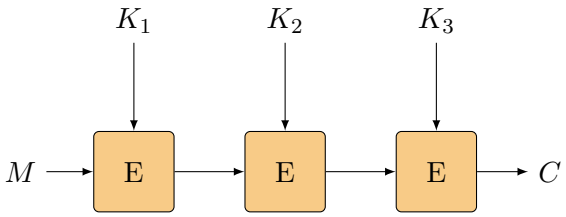

Figure 1: Triple-DES applies three times the encryption of DES using the 56-bit keys k_1, k_2, k_3.

Exercise 3.
The probability of guessing the correct key for AES-128 in a single try is 2^{-128}. Assume you play in a lottery where you have to pick 7 different numbers ranging from 1 to 36 and win the main price if you guessed all of them correctly. How often do you have to consecutively win the lottery such that it is more likely that you successfully guessed the correct key for AES-128?

Exercise 4.
In the lecture we have seen that DES only has a key size of 56-bit. As a solution to increase the key size we could use double or triple encryption (see Figure 1), this means we encrypt first with a key k_1, then k_2 and finally k_3 leading to an effective key size of 112 bits respectively 168 bits. Can you find a way to recover the full key (k_1, k_2) (or (k_1, k_2, k_3)) with less than 2^{112} (or 2^{168}) evaluations of the DES block cipher?

3 Stream Ciphers

RC4 is a stream cipher which has been widely used due to its efficiency in software and was a popular choice in the TLS protocol. However, several weaknesses have been found with this algorithm and it should not be used anymore. RC4 consists of an initial key scheduling algorithm which initializes the 256 byte state and then the pseudo-random generation produces a single output byte in each iteration.

Exercise 5.
For an ideal stream cipher we would want that the possible values of each output *digit* to be uniformly distributed and independent. This means in the case of RC4 each value should appear with a probability of 1/256 over random choices of the key. Compute the first 100 output bytes of the RC4 stream cipher with random choices of keys. Is the output stream unbiased?
As a starting point you can use the Python implementation below or implement your own version.

Example Python code for RC4

```
def keyschedule(key):
    keylength = len(key)
    S = range(256)
    j = 0
    for i in range(256):
        j = (j + S[i] + key[i % keylength]) % 256
        S[i], S[j] = S[j], S[i]
    return S

def prg(S):
    i = 0
    j = 0
    while True:
        i = (i + 1) % 256
        j = (j + S[i]) % 256
        S[i], S[j] = S[j], S[i]
        K = S[(S[i] + S[j]) % 256]
        yield K

if __name__ == '__main__':

    # Setup key
    key = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    S = keyschedule(key)

    # Generate
    for outputbyte in prg(S):
        print(outputbyte)
```