Comp90043-Assignment 1

Feedback and Solutions:

Model Solutions:

Q2:
(a) Determine the complexity of brute force attack to break the cipher?

You can mount a simple brute force attack by decrypting the ciphertext by trying all possible keys. The complexity is proportional to number of keys. There are exactly 12*26 = 312 keys and one of the keys will result in meaningful English text. Decryption of the ciphertext with a key involves computational proportional to the length of the given ciphertext, which is a fixed number in our case.

(b) Given that the plain text “THE" is mapped to the cipher text "SWX", can you
break the above cipher text? If you can, decrypt the cipher text and describe
how you break the code.

“T” is mapped to “S”
“H” is mapped to “W”
“E” is mapped to “X”

You will get three equations:

a * map(T) + b = map (S)-----(1)
a * map(H) + b = map (W)-----(2)
a * map(E) + b = map (X)-----(3)

In the above equations map is the function from English alphabet to integers modulo 26.

Solving (1) and (3) gets you a and b.
Then you can find inverse function as explained in Workshop2.

Decrypted text is:
THEQUICKBROWNFOXJUMPEDOVERTHELAZYDOG

Q3:
(a) The definition of risks and attacks were discussed in lectures. Refer to RFC2828 and the text for details. A standard answer would require an example each to illustrate risk and attack.

An example involving Microsoft security patches: These patches are released because of risks in the operating system. If someone is lucky, an attack may not occur even if the patches were not applied to the system. So the attack refers to actual misusing of the vulnerability. Definitely installing patches reduces or eliminates the risks.

(b) There are many answers, a few examples are:
Similarities:
1. A strong encryption algorithm or function.
2. Two keys are used in both modes.

Differences:

1. Symmetric cryptography operations are usually faster.
2. Asymmetric uses different keys for encryption and decryption.
3. Symmetric key must be secret to all but sender/receiver, for asymmetric typically one can be made public.
4. Symmetric encryption and decryption are usually faster than asymmetric.

Q4:
(a) A model solution is given below. Please refer to the textbook for a detailed explanation.

CFB
Encryption fn:
$I_1 = IV$
$I_j = LSB_{b-s}(I_j-1)||C_j-1$ j=2,…,N
$O_j = E(K,I_j)$ j=1,…,N
$C_j – P_j + MSB_s(O_j)$ j=1,…,N

Decryption fn:
$I_1 = IV$
$I_j = LSB_{b-s}(I_j-1)||C_j-1$ j=2,…,N
$O_j = E(K,I_j)$ j=1,…,N
$P_j – C_j + MSBs(O_j)$ j=1,…,N

OFB:
Encryption fn:
$I_1 = Nonce$
$I_j = O_j-1$ j=2,…,N
$O_j = E(K,I_j)$ j=1,…,N
$C_j = P_j + O_j$ j=1,…,N-1
$C_N = P_N + MSB_u(O_N)$

Decryption fn:
$I_1 = Nonce$
$I_j = LSB_{b-s}(I_j-1)||C_j-1$ j=2,…,N
$O_j = E(K,I_j)$ j=1,…,N
$P_j = C_j + O_j$ j=1,…,N-1
$P_N = C_N + MSB_u(O_N)$

CTR
Encryption fn:
$C_j = P_j + E(K, T_j)$ j=1,…,N-1
$C_N = P_N = MSB_u[E(K,T_N)]$
Decryption fn:
$P_j = C_j + E(K, T_j)$ j=1,…,N-1
$P_N = C_N = MSB_u[E(K,T_N)]$

b: size of a block
u: remaining size of block such that $u < b$
T: counter

(b) In ECB, encryption and decryption can be made parallel, as each block is independent of other blocks. Similarly for CTR, it can be made encrypted and decrypted concurrently.
ECB is simple, but repetition in plaintext could show up in ciphertext. CTR mitigates this weakness by having each block encrypt a different number, while still retaining the speed advantage.

Encryption is not parallelizable in CBC and CFB, but decryption is. In both modes, a ciphertext block depends on all preceding blocks, so they are very good at preventing cryptanalysis.

In OFB mode, parallel operations are possible for both encryption and decryption, given the output blocks are pre-computed beforehand (and so IV is known). A bit error or missing does not propagate. A disadvantage is IV must not be reused, and both parties must remain in sync.

(c) The question assumes that there was an error in block C4 of the transmitted ciphertext.
ECB mode: In this mode, ciphertext block Ci is used only as input for the direct dencryption of plaintext block Pi. Therefore, a transmission error in block C4 will only corrupt block P4 of the decrypted plaintext.
CBC mode: In this mode, ciphertext block Ci is used as input to the XOR function when obtaining plaintext blocks Pi and Pi+1. Therefore, a transmission error in block C4 will corrupt blocks P4 and P5 of the decrypted plaintext, but will not propagate to any of the other blocks.
CTR mode: In this mode, ciphertext block Ci, as well as the encrypted counter ti, are used only as input for the direct decryption of plaintext block Pi. Therefore, a transmission error in block C4 will only corrupt block P4 of the decrypted plaintext.

Q5a

For a to have an inverse, the gcd of a and n must be 1, meaning they are coprime with each other. Given two integers a and n, their gcd can be rewritten as $ax + ny$. So, $gcd(a, n) = (ax) \mod n$, since $n \times y$ is a multiple of n. Thus, we get $1 = (ax) \mod n$, which is equivalent to $(ax) = 1 \mod n$. Then, we can solve a's inverse by running the extended Euclidean algorithm.

Suppose we have a function *gcd(a,n)*, when given two inputs a and n, it outputs their greatest common divisor, x and y.

```
d,x,y = gcd(a,n)
if (d == 1):
    return x
else:
    there is no inverse
```

Q5b

There are 624 different combinations of (a,b,c) which we can use to encrypt p. However, there are only 312 distinct permutations which can be formed, meaning several combinations may produce identical substitutions. For instance, consider $k_1 = (0, 1, 0)$ and $k_2 = (13, 14, 0)$. For k_1, it is easy to see that $E_{k_1}(n) = n$. For k_2, we can see:

$$\begin{aligned}
E_{k_2}(1) &= (13 + 14) \mod 26 = 1 \\
E_{k_2}(2) &= (13 \times 4 + 14 \times 2) \mod 26 = 80 \mod 26 = 2 \\
E_{k_2}(3) &= (13 \times 9 + 14 \times 3) \mod 26 = 159 \mod 26 = 3 \\
&\vdots \\
E_{k_2}(25) &= (13 \times 625 + 14 \times 25) \mod 26 = 8475 \mod 26 = 25
\end{aligned}$$

This is obviously a trivial key, so the number of non-trivial keys is 311.

Q5c

1. For key of length n, each can be one of 26 possible characters, there are 26^n possible keys in total.
2. “yahkqpt”.
3. “yahkqpt” - “unimelb” = “enzymes”; “enzymes” + “rmituni” = “vzhrgra”