COMP90043 Cryptography and Security
Semester 2, 2020, Workshop Week 4 Solutions

Part A

1. Let C_1 and C_2 be two n-bit ciphertexts obtained by encrypting using one-time pad key K on plaintexts M_1 and M_2 respectively. Show that $M_1 \oplus M_2 = C_1 \oplus C_2$. What is the consequence of Known Plaintext attack on the one-time pad encryption?

$$\begin{aligned} C_1 &= M_1 \oplus K \\ C_2 &= M_2 \oplus K \\ C_1 \oplus C_2 &= (M_1 \oplus K) \oplus (M_2 \oplus K) \\ &= M_1 \oplus M_2 \oplus K \oplus K \\ &= M_1 \oplus M_2 \end{aligned}$$

   Ciphertext can be decrypted by using another pair of plaintext and ciphertext encrypted using the same key.

2. The Vernam cipher can be considered as a one-time pad where message and cipher space are English text treated as sequences of integers between 0 and 25 and the $\oplus$ operation is replaced by sum modulo 26. Let $M[i], K[i] \in \{0, 1, \cdots, 25\}, 0 \leq i < n$, then the encryption function can be implemented as:

```
for i = 0 to n-1 do
    C[i] = M[i] + K[i] mod 26
```

   (a) What's the decryption function?
   The decryption of C with the key K can be given by

```
for i = 0 to n-1 do
    M[i] = C[i] - K[i] mod 26
```

   (b) If the length of the key is n, how many different possible keys are there in Vernam cipher?
   26^n

   (c) Encrypt "unimelb" with the key "tuesday".
   u(20) + t(19) = n(13)
   n(13) + u(20) = h(7)
   i(8) + e(4) = m(12)
   m(12) + s(18) = e(4)
   e(4) + d(3) = h(7)
   l(11) + a(0) = l(11)
   b(1) + y(24) = z(25)

(d) What should be the key that decrypts the ciphertext in (c) to “rmituni”?

n(13) - r(17) = w(22)
h(7) - m(12) = v(21)
m(12) - i(8) = e(4)
e(4) - t(19) = l(11)
h(7) - u(20) = n(13)
l(11) - n(13) = y(24)
z(25) - i(8) = r(17)

3. State the condition for perfect secrecy.

$$\mathbf{Pr}[\mathbf{M} = \mathbf{x} | \mathbf{C} = \mathbf{y}] = \mathbf{Pr}[\mathbf{M} = \mathbf{x}]$$

Part B: Block Cipher Modes

(see next page)

(1) Only the plaintext unit corresponding to the ciphertext character is affected. In OFB method, the bit errors in transmission do not propagate. For example, if a bit error occurs in C_1, only the recovered value of P_1 is affected; subsequent plaintext units are not corrupted.

(2) In some modes, the plaintext does not pass through the encryption function, but is XORed with the output of the encryption function. The math works out that for decryption in these cases, the encryption function must also be used.

(3)(a) If the IVs are kept secret, the 3-loop case has more bits to be determined and is therefore more secure than 1-loop for brute force attacks.

(3)(b) For software implementations, the performance is equivalent for most measurements. One-loop has two fewer XORs per block. Three-loop might benefit from the ability to do a large set of blocks with a single key before switching. The performance difference from choice of mode can be expected to be smaller than the differences induced by normal variation in programming style.
For hardware implementations, three-loop is three times faster than one-loop, because of pipelining. That is: Let P_i be the stream of input plaintext blocks, X_i the output of the first DES, Y_i the output of the second DES and C_i the output of the final DES and therefore the whole system's ciphertext.
In the 1-loop case, we have:

$$
\begin{aligned}
X_i &= DES(XOR(P_i, C_{i-1})) \\
Y_i &= DES(X_i) \\
C_i &= DES(Y_i)
\end{aligned}
$$

where C_0 is the single IV.

If P_1 is presented at $t = 0$ (where time is measured in units of DES operations), X_1 will be available at $t = 1$, Y_1 at $t = 2$ and C_1 at $t = 3$. At $t = 1$, the first DES is free to do more work, but that work will be: $X_2 = DES(XOR(P_2, C_1))$ but C_1 is not available until $t = 3$, therefore X_2 can not be available until $t = 4$, Y_2 at $t = 5$ and C_2 at $t = 6$.

In the 3-loop case, we have:

$$
\begin{aligned}
X_i &= DES(XOR(P_i, X_{i-1})) \\
Y_i &= DES(XOR(X_i, Y_{i-1})) \\
C_i &= DES(XOR(Y_i, C_{i-1}))
\end{aligned}
$$

where X_0, Y_0 and C_0 are three independent IVs.

If P_1 is presented at $t = 0$, X_1 is available at $t = 1$. Both X_2 and Y_1 are available at $t = 4$. X_3, Y_2 and C_1 are available at $t = 3$. X_4, Y_3 and C_2 are available at $t = 4$. Therefore, a new ciphertext block is produced every 1 tick, as opposed to every 3 ticks in the single-loop case. This gives the three-loop construct a throughput three times greater than one-loop construct.