Intro to Security HW 1

Taylor Foxhall

September 18, 2016

1 Problem 1

An affine cipher is a type of simple substitution where each letter is encrypted according to the following rule $c = (ap+b) \mod 26$. Here, p, c, a, and b are each numbers in the range of 0 to 25, where p represents the plaintext letter, c the ciphertext letter, and a and b are constants. For the plaintext and ciphertext, 0 corresponds to "a," 1 corresponds to "b," and so on. Consider the ciphertext QJKES REOGH GXXRE OXEO, which was generated using an affine cipher. Determine the constants a and b and decipher the message. Hint: Plaintext "t" encrypts to ciphertext "H" and plaintext "o" encrypts to ciphertext "E."

1.1 Answer

Given that "t" encrypts to "H", and "o" encrypts to "E".

$$7 = 19a + b \mod 26$$
$$4 = 14a + b \mod 26$$

Subtracting them gives us

$$
\begin{aligned}
3 &= 5a \mod 26 \\
5^{-1} &= 21 \mod 26 \\
a &= 21 * 3 \mod 26 \\
a &= 11 \\
\\
7 &= (11)(19) + b \mod 26 \\
7 &= 209 + b \mod 26 \\
b &= -202 \mod 26 \\
b &= 6
\end{aligned}
$$

So with a little python script we can decipher the text.

```
def encrypt_letter(p, a, b):
    pp = ord(p) - ord('a')
    c = (pp*a + b) % 26
    return chr(c + ord('A'))

A = 11
B = 6
CIPHERTEXT = "QJKESREOGHGXXREOXEO"
ALPHABET = "abcdefghijklmnopqrstuvwxyz"

# Create a mapping from ciphertext letters to plaintext letters
decrypt_letter = {encrypt_letter(p, A, B): p for p in ALPHABET}

# Lookup every ciphertext's letter and join the whole result
return "".join(decrypt_letter[c] for c in CIPHERTEXT)
```

Which yields us $P = ifyoubowatallbowlow$, or with appropriate spacing and punctuation: "If you bow at all, bow low."

2 Problem 2

Consider a Feistel cipher with four rounds. Then the plaintext is denoted as P = (L0, R0) and the corresponding ciphertext is C = (L4, R4). What is the ciphertext C, in terms of L0, R0, and the subkey, for each of the following round functions? (You should get the most concise solution.)

- F(Ri-1, Ki) = 0
- F(Ri-1, Ki) = Ri-1
- F(Ri-1, Ki) = Ki
- F(Ri-1, Ki) = Ri-1 Ki

(Note that for each of cases A – D, the cipher uses four rounds.)

2.1 Answer

2.1.1 F($\mathbf{R_{i-1}}$, $\mathbf{K_i}$) = 0

$$
\begin{aligned}
L_1 &= R_0 \\
R_1 &= R_0 \oplus 0 = L_2 \\
R_2 &= R1 \oplus 0 = L_3 = R_0 \oplus 0 \\
R_3 &= R2 \oplus 0 = L_4 = R_0 \oplus 0 \\
R_4 &= R_3 \oplus 0 = R_0 \oplus 0
\end{aligned}
$$

2.1.2 F($\mathbf{R_{i-1}}$, $\mathbf{K_i}$) = $\mathbf{R_{i-1}}$

$$
\begin{aligned}
L_1 &= R_0 \\
R_1 &= R_0 \oplus R_0 = L_2 = 0 \\
R_2 &= R_1 \oplus R_1 = L_3 = 0 \\
R_3 &= R_2 \oplus R_2 = L_4 = 0 \\
R_4 &= R_3 \oplus R_3 = 0
\end{aligned}
$$

2.1.3 F($\mathbf{R_{i-1}}$, $\mathbf{K_i}$) = $\mathbf{K_i}$

$$
\begin{aligned}
L_1 &= R_0 \\
R_1 &= R_0 \oplus K_1 = L_2 \\
R_2 &= R_1 \oplus K_2 = L_3 = R_0 \oplus K_1 \oplus K_2 \\
R_3 &= R_2 \oplus K_3 = L_4 = R_0 \oplus K_1 \oplus K_2 \oplus K_3 \\
R_4 &= R_3 \oplus K_4 = R_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4
\end{aligned}
$$

2.1.4 $\mathbf{F(R_{i\text{-}1}, K_i) = R_{i\text{-}1} \quad K_i}$

$$
\begin{aligned}
L_1 &= R_0 \\
R_1 &= R_0 \oplus R_0 \oplus K_1 = L_2 \\
R_2 &= R_1 \oplus R_1 \oplus K_2 = L_3 = K_2 \\
R_3 &= R_2 \oplus R_2 \oplus K_3 = L_4 = K_3 \\
R_4 &= R_3 \oplus R_3 \oplus K_4 = K_4
\end{aligned}
$$

3 Problem 3

Suppose that we use a block cipher to encrypt according to the rule:

$$
\begin{aligned}
C0 &= IV E(P0, K), \\
C1 &= C0E(P1, K), \\
C2 &= C1E(P2, K), \\
&\dots
\end{aligned}
$$

3.1 Answer

3.1.1 What is the corresponding decryption rule?

$$
\begin{aligned}
P_0 &= D(IV \oplus C_0, K) \\
P_i &= D(C_{i-1} \oplus C_i, K) \qquad\qquad\qquad i > 0
\end{aligned}
$$

3.1.2 Give two security disadvantages of this mode as compared to CBC mode.

Say $P = P_0P_1P_2P_3$ and in particular $P_1 = P2$. Then,

$$C_0 = IV \oplus E(P_0, K)$$

$$C_2 = IV \oplus E(P_0, K) \oplus E(P_1, K) \oplus E(P_2, K) = IV \oplus E(P_0, K)$$

This is a compromise of confidentiality, because the same plaintext blocks can yield the same ciphertext blocks, similar to the issues with ECB. At the same time, say Trudy switches C_0 and C_2. Then Bob, who receives the message, will get $P_{trudy} = P_1P_2P_1P_3$. This is a compromise of integrity, as the message can be rearranged and altered given certain matching plaintexts.