Name:- Hrithik Salunke
Roll No:- 181105049

classmate
Date
Page

CCT- Assignment - 3

1)

A a) Given the ciphertext $[M_n][M_{n+1}]$ switch sides to obtain $[M_{n+1}][M_n]$. We know that for

$$M_{n+1} = M_{n-1} \oplus f(k, M_n)$$

So we can obtain M_{n-1} from M_{n+1} by

$$M_{n-1} \oplus f(k, M_n) \oplus f(k, M_n)$$

since $f(k, M_n) \oplus f(k, M_n)$ will have a net result of 0,
By repeating this process we can deduce M_j until we reach $[M_1][M_0]$, which then by switching these halves gives the plaintext $[M_0]$ $[M_1]$

A b) Round 1: $[M_0][M_1] = [M_1][M_0 \oplus f(k, M_1)]$
$= [M_1][M_0 \oplus k \oplus M_1]$
$= [M_1][M_2]$

Round 2: $[M_1][M_2] = [M_2][M_1 \oplus f(K, M_2)]$
$= [M_2][M_1 \oplus k \oplus M_2]$
$= [M_0 \oplus k \oplus M_1][M_1 \oplus M_1 \oplus k \oplus k \oplus M_0]$
$= [M_0 \oplus k \oplus M_1][M_0]$

So knowing the ciphertext, we could combine 2 halves to obtain $M_0 \oplus k \oplus M_1 \oplus M_0 = k \oplus M_1$
After 2 rounds the ciphertext also lets you determine M_0 and therefore $M_1 \oplus k$ but not M_1 or k individually

A c) Round 3: $[M_0][M_0 \oplus k \oplus M_1 \oplus f(k, M_0)]$
$= [M_0][M_0 \oplus k \oplus M_1 \oplus k \oplus M_0]$
$= [M_0][M_1]$

Round 3 produces the plaintext, making the method non-secure

A 2) If someone discovers the fixed key and obtains the encrypted password file, this person can easily decrypt by the usual decryption procedure. However knowing the ciphertext and the plaintext does not readily allow one to deduce the key.

A3) CBC We have $D_k(C_j) \oplus C_{j-1}$

$$= D_k(E_k(P_j \oplus C_{j-1})) \oplus C_{j-1}$$

$$= P_j \oplus C_{j-1} \oplus C_{j-1}$$

$$= P_j$$

CFB We have $C_j \oplus L_8(E_k(X_j))$

$$= (P_j \oplus L_8(E_k(X_j))) \oplus L_8(E_k(X_j))$$

$$= P_j$$

A4) Split P into halves L_0 and R_0, then

$$[L_0][R_0] \xrightarrow{k} [R_0][L_0 \oplus f(k, R_0)]$$

Now consider the complement of k, C and P.

Split $\overline{P}$ into 2 halves $\overline{L_0}$ and $\overline{R_0}$, then

$$[\overline{L_0}][\overline{R_0}] \xrightarrow{\overline{k}} [\overline{R_0}][\overline{L_0} \oplus f(\overline{R_0}, \overline{k})]$$

Here.

$$\overline{L_0} \oplus f(\overline{R_0}, \overline{k})$$

$$= \overline{L_0} \oplus \overline{R_0} \oplus 111\ldots \oplus k \oplus 111\ldots$$

$$= \overline{L_0} \oplus R_0 \oplus k$$

$$= \overline{L_0} \oplus f(R_0, k) \rightarrow (1)$$

and $\overline{L_0} \oplus f(\overline{R_0}, \overline{k}) = L_0 \oplus 111\ldots \oplus R_0 \oplus k$

$$= L_0 \oplus R_0 \oplus k \oplus 111\ldots$$

$$= L_0 \oplus f(R_0, k) \oplus 111\ldots$$

$$= \overline{R_1} \rightarrow (2)$$

From (1) and (2)

$$[\overline{L_0}][\overline{R_0}] \xrightarrow{\overline{k}} [\overline{R_0}][\overline{L_0} \oplus f(\overline{R_0}, \overline{k})]$$

$$= [\overline{L_1}][\overline{R_1}]$$

$$= \overline{C}$$

Scanned by CamScanner

A5)

a) The keys $k_1 \ldots k_{16}$ are all the same (all 1's). Decryption is accomplished by reversing the order of the keys to $k_{16} \ldots k_1$. Since the k_i are all the same, this is the same as encryption, so encrypting twice gives back the plaintext.

b) The key of all 0's.

A6) Let (m, c) be a plaintext-ciphertext pair. Make one list of $E_k(E_k(m))$, where k runs through all possible keys. Make another list of $D_{k'}(c)$, where k' runs through all possible keys. A match between the two list is a pair k, k' of keys with $E_{k'}(E_k(E_k(m))) = c$. There should be a small number of such pairs. For each such pair try it on another plaintext and see if it produces the corresponding ciphertext. This should eliminate most of the incorrect pairs. Repeating a few more times should yeild the pair k_1, k_2.

A7) a) To perform the meet in the middle attack, you need a plaintext m and ciphertext c pair. So, make 2 lists. The left lists consist of encryptions using the second encryption E^2 with different choices for k_2. Similarly, the right side contains decryptions using different keys for the first encryption algorithm. Then the list looks like

$E_1^2(m) = y_1$ $\qquad$ $z_1 = D_1'(c)$

$E_2^2(m) = y_2$ $\qquad$ $z_2 = D_2'(c)$

$\vdots$ $\qquad$ $\vdots$

$E_{788}^2(m) = y_{788}$ $\qquad$ $z_{788} = D_{788}'(c)$

$\vdots$ $\qquad$ $\vdots$

Scanned by CamScanner

Look for matches between y_i and z_ℓ. A match using k_i^1 for E^2 and k_i' for D' indicates

$$E^2_{k_i'}(m) = y = D'_{k_i'}(c)$$

and hence $E^1_{k_i}(E^2_{k_i}(m)) = c$

b) There are 26 possibilities for β and 12 possibilities for α. Let $E_\alpha^2(x) = \alpha x \pmod{26}$ and $E_\beta^1(x) = x + \beta \pmod{26}$. The composition of these two gives the affine cipher. The total computation needed involves producing 26 encryptions for E^2 and 12 decryptions for E^1. The total is 38.

Ans 8.) Suppose we modify the feistel setup as follows. Divide the plaintext into 3 equal blocks L_0, M_0, R_0. Let the key for i^{th} round be k_i and let f be some function that produces the appropriate size output. The i^{th} round of encryption is given by,

$$L_i = R_{i-1}, \quad M_i = L_{i-1}, \quad R_i = f(k_i, R_{i-1}) \oplus M_{i-1}$$

This continues for n rounds. Consider the decryption algorithm that starts with the ciphertext A_n, B_n, C_n and uses the algorithm.

$$A_{i-1} = B_i, \quad B_{i-1} = f(k_i, A_i) \oplus C_i, \quad C_{i-1} = A_i$$

Continue this for n rounds, down to A_0, B_0, C_0. Show that $A_i = L_i, B_i = M_i, C_i, R_i$ for all i and that the decryption algorithm returns the plaintext.

Here, the i^{th} encryption step is the same as the $n-i^{th}$ decryption step.

Consider the n^{th} round of encryption:

$$[L_{n-1}][M_{n-1}][R_{n-1}] \xrightarrow{k} [R_{n-1}][L_{n-1}][f(k_n, R_{n-1}) \oplus M_{n-1}]$$

Scanned by CamScanner

$$= [L_n][M_n][R_n]$$
$$= [A_n][B_n][C_n]$$

Decryption

$$[A_n][B_n][C_n] \xrightarrow{k} [B_n][f(K_n, A_n) \oplus C_n][A_n]$$
$$= [L_{n+1}][f(K_n, R_{n-1}) \oplus f(K_n, R_{n-1}) \oplus M_{n-1}][R_{n-1}]$$
$$= [L_{n-1}][M_{n-1}][R_{n-1}]$$

So each round of decryption gives the prior round after encryption. Continuing gives $A_i = L_i$, $B_i = M_i$ and $C_i = R_i$

Ans 9)

a) At the decryption side, the decryptor has $\{C_1, C_2 \dots\}$ and the initial ~~x~~ X_1. To decrypt, the decryptor starts with $j = 1$ and ~~cobut~~ calculates

$$P_j = C_j \oplus L_{32}(E_k(X_j))$$
$$X_{j+1} = R_{32}(X_j) \| C_j$$

b) Start with X_1 and a sequence of ciphertext $\tilde{C}_1, C_2, C_3 \dots$
To decrypt the first block, we calculate

$$\tilde{P}_1 = \tilde{C}_1 \oplus L_{32}(E_k(X_1))$$
$$\tilde{X}_2 = R_{32}(X_1) \| \tilde{C}_1$$

Observe that the decrypted plaintext $\tilde{P}_1$ is corrupted because it has the corrupted $\tilde{C}_1$ as part of it, and also that $\tilde{X}_2 \neq X_2$ since it has been corrupted. $\tilde{C}_1$ is as part of it. The next couple steps of decryption proceed as

$$\tilde{P}_2 = C_2 \oplus L_{32}(E_k(\tilde{X}_2))$$
$$\tilde{X}_3 = R_{32}(\tilde{X}_2) \| C_2 = \tilde{C}_1 \| C_2$$
$$\tilde{P}_3 = C_3 \oplus L_{32}(E_k(\tilde{X}_3))$$
$$X_4 = R_{32}(\tilde{X}_3) \| C_3 = C_2 \| C_3$$

X_4 is no longer corrupted. The subsequent decryption step is

$$P_4 = C_4 \oplus L_{32}(E_k(X_4))$$
$$X_5 = R_{32}(X_4) \| C_4$$

All subsequent decryption steps will be free of errors

Scanned by CamScanner

classmate

Date ______
Page ______

A10) In CBC, suppose that an error occurs in block C_j to produce the corrupted $\tilde{C}_j$ and that the subsequent blocks C_{j+1} and C_{j+2} are c

$$\tilde{P}_j = D_k(\tilde{C}_j) \oplus C_{j-1}$$

is corrupted.

Next, $\tilde{P}_{j+1} = D_k(C_{j+1}) \oplus \tilde{C}_j$

although C_{j+1} is correct, when we add the corrupted $\tilde{C}_j$ we get a corrupted answer

Now for $P_{j+2} + D_k(C_{j+2}) \oplus C_{j+1}$

It is uncorrupted since $D_k(C_{j+2})$ and C_{j+1} are uncorrupted.

A11.) Let k be the key we wish to find. Using the limit we have

$C_1 = E_k(M_1)$ and $C_2 = E_k(\overline{M_1})$

Suppose we start brute force attack by encrypting M_1 with different keys. If when we use k_j we get $E_{k_j}(M_1) = C_1$ then we are done and key we desire is $k = k_j$.

If $E_{k_j}(M_1) = \overline{C_2}$ then we know that $E_{\overline{k_j}}(\overline{M_1}) = C_2$. If this happens, we know the key is $\overline{k_j}$ since $\overline{k_j}$ would decrypt C_2 to get $\overline{M_1}$. We are effectively testing two keys for the price of one. Hence the key space is cut in half and we only have to ~~exa~~ search out an average of 2^{54}.

Scanned by CamScanner