Homework 1 - Xavier Gitiaux

February 2019

1 Exercise 1

Done!

2 Exercise 2

2.1

Denote $c = c_1c_2c_3c_4$ the cyphertext. The adversary compute $k = a - c_1\%26$. If $c_2 - k = b\%26$, it concludes that the password was "abcd"; otherwise, he concludes that the password was "kdmf".

2.2

Denote $c = c_1c_2c_3c_4$ the cyphertext. The adversary compute $k = a - c_1\%26$. If $c_3 - k = c\%26$, it concludes that the password was "abcd"; otherwise, he concludes that the password was "kdmf".

2.3

Denote $c = c_1c_2c_3c_4$ the cyphertext. The adversary compute $k = a - c_1\%26$. If $c_4 - k = d\%26$, it concludes that the password was "abcd"; otherwise, he concludes that the password was "kdmf".

2.4

If the key length is equal to the message length, perfect security is achieved: an adversary cannot distinguish the two passwords (i.e., cannot do better than flipping a coin to distinguish "abcd" and "kdmf").

3 Exercise 3

3.1

Plaintexts and cyphertexts are of size n bits.

3.2

Given a pair (m, c), the brute force attack consists in searching the $2^l \times 2^l$ key space for the pair of keys (k_1, k_2) such that $Enc_{k_1,k_2}(m) = c$. For each k_1, the intermediate encrypted value $En_{k_1}(m)$ is stored in the $n-$bit available memory s, which is then accessed to encrypt with key k_2. In the worst-case, the total number of loops is 2^{2l} and each loop requires 2 encryptions, so a $O\left(n2^{2l}\right)$ running time if each of the 2×2^{2l} encryptions takes $O\left(n\right)$ time.

Algorithm 1 Exercise 3: Brute-Force Attack - Question 2

Input: m, n, a $n-$bit storage s, $\{E_k(.)\}_{k\in\{0,1\}^l}$, $\{D_k(.)\}_{k\in\{0,1\}^l}$
for i=1..l **do**
    for j=1..l **do**
        $s \leftarrow E_{k[i]}(m)$
        $s \leftarrow E_{k[j]}(s)$
        if $s == c$ **then**
            return $k[i], k[j]$.

3.3

Given a pair (m, c), an attack can use the $n2^l$ memory space in the following way:

- For each key k_1 in the 2^l key space, encrypt the plaintext m and store $Enc_{k_1}(m)$ along with the key k_1 in memory (assuming that storing a key is memory free) for example in a hash table.
- For each key k_2 in the 2^l key space, decrypt the cyphertext c using $Dec_{k_2}(c)$ and look for a match in the memory space.
- If a match is found, return the corresponding k_1 and k_2.

The first loop takes 2^l encryptions, each at a cost $O(n)$. The second loop takes at most 2^l decryptions, each at a cost $O(n)$. Therefore the total running time is $O(n2^l)$.

Algorithm 2 Exercise 3: Brute-Force Attack - Question 3

$ATTACK2DES(m, c)$
Input: m, n, a $n2^l$bit storage s, $\{E_k(.)\}_{k\in\{0,1\}^l}$, $\{D_k(.)\}_{k\in\{0,1\}^l}$
$storage \leftarrow \{\}$
for i=1..l **do**
    $s[E_{k[i]}(m)] = k[i]$
for j=1..l **do**
    $c1 \leftarrow D_{k[j]}(c)$
    if $storage[c1]$ **then**
        return $storage[c1], k[j]$
.

3.4

Given a pair (m, c), an attack can use the $n2^l$ memory space in the following way: search through all the keys k_1 and for each k_1, compute $c_1 = En_{k_1}(m)$ and apply the procedure $ATTACK2DES$ of the previous question to $c1$ and c. There are at most 2^l for loops and each iteration requires one encryption and one call to $ATTACK2DES$, so $O(2^l)$ encryptions and $O(2^l)$ decryptions. Therefore the total running time is $O(n2^{2l})$.

Algorithm 3 Exercise 3: Brute-Force Attack - Question 4

```
ATTACK3DES(m, c)
Input: m, n, a n2^l bit storage s, {E_k(.)}_{k∈{0,1}^l}, {D_k(.)}_{k∈{0,1}^l}
for i=1..l do
    c1 ← E_{k[i]}(m).
    k ← ATTACK2DES(c1, c)
    if k is not None then
        return k[i], k[1], k[2]
```

.

4 Exercise 4

4.1

The adversary first sends a message m with bit 0 only and receives a cyphertext c. Then he sends two messages $m_0 = m$ and $m_1 = 11$. The challenger returns c_b. If $c_b = c$, the adversary returns $b = 0$; otherwise he returns $b = 1$. The adversary distinguishes $b = 0$ from $b = 1$ with probability 1, since m and m_0 will be encrypted to the same cyphertext $m \oplus F_k(m)$, while m_1 encrypts to $m_1 + F_k(m_1) \neq m \oplus F_k(m)$.

4.2

The adversary first sends one message m with bit zero only. He receives a cyphertext $c = r||(H(r) \oplus m \oplus k)$. He takes the last n bits of c and compute $h = (H(r) \oplus m \oplus k) \oplus m = H(r) \oplus k$ and then, deduce the key k as $k = h \oplus H(r)$. Then, the adversary sends two messages $m_0 \neq m_1$. When receiving $c_b = r^{'}||c_{2b}$, if $c_{2b} \oplus H(r^{'}) \oplus k = m_o$, he returns $b = 0$; otherwise, he returns $b = 1$. The adversary distinguishes $b = 0$ from $b = 1$ with probability 1, since he has obtained the key k in the first phase.

5 Exercise 5

5.1

This is a valid block cypher: for each $k\{0, 1\}^2$ (i.e. each row), there is no repeat along the row and there are exactly 2^3 columns, so the row represents a permutation from $\{0, 1\}^3$ to $\{0, 1\}^3$.

5.2

This is a valid block cypher since for each k the identity function $E_k(x) = x$ is a permutation.

5.3

For each $k \in \{0,1\}^n$, $E_k^{''}$ is a one-to-one function: first, for $x \neq y$, $E_k^{''}(x) = E_k^{''}(y)$ implies that $\overline{x} = \overline{y}$ and thus that $x = y$. Moreover, for $y \in \{0,1\}$, $y = E_k^{''}(k \oplus \overline{y})$, so $E_k^{''}$ is onto. Therefore, for $k \in \{0,1\}^n$ $E_k^{''}$ is a permutation and $E^{''}$ is valid block cypher.

5.4

If a distinguisher queries x_1 and $x_2 \neq x_1$ from an oracle $E_k^{'}$ for some secret key k , he gets x_1 and x_2 respectively which would happen with negligible probability if E_k was randomly drawn from the set of all possible permutations of $\{0,1\}^n$. Therefore, $E_k^{'}$ is not a secure block cypher.

Similarly, by querying x_1 and $x_2 \neq x_1$ from the challenger, , a distinguisher would obtain y_1 and y_2 and compute $y = y_1 \oplus y_2$. If $y = x_1 \oplus x_2$, the distinguisher returns 1 (i.e. the challenger uses $E_k^{''}$ for some secret key k); otherwise, the distinguisher returns 0 (the challenger uses a permutation randomly drawn from all permutations of $\{0,1\}^n$). Since $y = x_1 \oplus x_2$ happens with negligible probability with a random permutation, the distinguisher will win with non-negligible probability.