CS641
Modern Cryptology
Indian Institute of Technology, Kanpur

Mid Semester Examination

Group Number: NaVi
Aaryen Milan Mehta (190495), Naivedya Amarnani (190522), Himanshu Kishor Choubey (190376)

Date of Submission:
March 10, 2021

Question 1

Consider a variant of DES algorithm in which the S-box S1 is changed as follows:

> For every six bit input α, the following property holds: $\text{S1}(\alpha) = \text{S1}(\alpha \oplus 001100) \oplus 1111$.

All other S-boxes and operations remain the same. Design an algorithm to break four rounds of this variant. In order to get any credit, your algorithm must make use of the changed behavior of S1.

Solution

We are given,

$$S1(\alpha) = S1(\alpha \oplus 001100) \oplus 1111$$

Proof. Taking XOR on both sides with $S1(\alpha \oplus 001100)$,

$$S1(\alpha) \oplus S1(\alpha \oplus 001100) = 0000 \oplus 1111 \tag{1.1}$$

$$S1(\alpha) \oplus S1(\alpha \oplus 001100) = 15 \tag{1.2}$$

with probability 1. □

As in the case of breaking 4 or more rounds of DES by differential analysis, we consider two pairs of inputs with the same right half (i.e. if the two inputs are $L_o R_o$ and $L_o^{'} R_o^{'}$, then $R_o = R_o^{'}$). We then examine how the XOR of these inputs travels through the rounds much like how we checked in the lectures.

From the mathematical analysis done above, we can conclude that if we consider input pairs that result in the XOR of inputs at second round S1 to be 001100, then we expect that with probability 1, the XOR of the output will be 1111. Ensuring that input XOR to remaining S-boxes in the second round is all zeroes, we can predict the XOR of the second round output of S-boxes with certainty (i.e. with probability 1).

Following a similar analysis of the pair of plaintext and ciphertext and their XOR as in the last slide of lecture 6, we see that in the last round, we know the XOR of the inputs to the S box as well as the output of the S1 box with certainty.

Repeating the mathematical analysis in the first few slides of lecture 7, we define α_i, $\alpha_i^{'}$, β_i and $\beta_i^{'}$ and $k_{4,i}$ in an identical fashion.

Let $y_i = S_i(\beta_i)$ and $y_i^{'} = S_i(\beta_i^{'})$. Clearly, $|y_i| = 4 = |y_i^{'}|$. We know α_i, $\alpha_i^{'}$ and the fact that $\beta_i \oplus \beta_i^{'} = \alpha_i \oplus \alpha_i^{'}$. We also know a value y (1111 in this case) such that $y_i \oplus y_i^{'} = y$, which happens with certainty. Define $X_i = \{(\beta, \beta^{'}) \mid \beta \oplus \beta^{'} = \beta_i \oplus \beta_i^{'}$ and $S_i(\beta) \oplus S_i(\beta^{'}) = y\}$ (Here we make use of the assumption that the attacker has the information about the updated $S1$ box)

We see that a pair $(\beta_i, \beta_i^{'}) \in X_i$ whenever $y_i \oplus y_i^{'} = y$, which happens with certainty. Define $K_i = \{k \mid \alpha_i \oplus k = \beta$ and $(\beta, \beta^{'}) \in X_i$ for some $\beta^{'}\}$. Since, $(\beta_i, \beta_i^{'}) \in X_i$ with certainty, we have $k_{4,i} \in K_i$ with certainty.

Now this scenario bears resemblance to the method of breaking of three round DES as described in the lectures. We have $|K_i| = |X_i|$ since α_i and $\beta \oplus \beta^{'}$ is fixed for $(\beta, \beta^{'}) \in X_i$. If $|X_i| < 64$ then we have eliminated some possibilities of $k_{4,i}$. As done earlier, we can then repeat for multiple pairs of plaintexts with the same constraints as described above to reduce the possibilities further until $k_{4,i}$ is uniquely identified. (The case where $|X_i| = 64$ has been dealt with in the lecture and can be handled here in a similar fashion.)

Note: This is a chosen plaintext attack since the two inputs are chosen so that:

1. They have the same right half.
2. The input XOR to all S-boxes except $S1$ in the second round is 0.
3. The input XOR to S1 in the second round is 001100.

Question 2

The SUBSET-SUM problem is defined as follows:

> Given $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ and $m \in \mathbb{Z}$, find $(b_1, \ldots, b_n) \in \{0,1\}^n$ such that $\sum_{i=1}^{n} a_i b_i = m$ if it exists.

This problem is believed to be a hard-to-solve problem in general. Consider a hypothetical scenario where Anubha and Braj have access to a fast method of solving SUBSET-SUM problem. They use the following method to exchange a secret key of AES:

> Anubha generates an $n = 128$ bit secret key k. She then chooses n positive integers $a_1, \ldots, a_n$ such that $a_i > \sum_{1 \leq j < i} a_j$. She computes $m = \sum_{i=1}^{n} a_i k_i$ and sends $(a_1, a_2, \ldots, a_n, m)$ to Braj, where k_i is ith bit of k. Upon receiving numbers $(a_1, a_2, \ldots, a_n, m)$, Braj solves the SUBSET-SUM problem to extract the key k.

Show that an attacker Ela does not need to solve SUBSET-SUM problem to retrieve the key k from $(a_1, a_2, \ldots, a_n, m)$.

Solution

We are given,

$$a_i > \sum_{1 \leq j < i} a_j \tag{2.1}$$

and,

$$m = \sum_{i=1}^{n} a_i k_i \tag{2.2}$$

Now, We assume that there is always a solution to the subset sum problem, since the key will always exist.
We find the key recursively as follows:
Consider l such that the following equations hold:

$$a_l \leq m \tag{2.3}$$

$$a_{l+1} > m \tag{2.4}$$

Note that such an l will exist whenever m is non-zero (From Equation 2.1).
Using Equation 2.1 we know that

$$a_l > \sum_{1 \leq j < l} a_j \tag{2.5}$$

so, using Equation 2.3, Equation 2.4, Equation 2.5 and the fact that an a_l can only be added once in the summation (since the coefficients $k_i's$ belong to {0,1}), the key's l^{th} element must be 1.

Proof. Proof for the above statement:

Now, using Equation 2.1, write,

$$a_l > \sum_{1 \leq j < l-1} a_j$$

$$m \geq a_l$$

$$\therefore m > \sum_{1 \leq j < l-1} a_j \geq \sum_{1 \leq j < l-1} k_j a_j$$

$$\therefore m > \sum_{1 \leq j < l-1} k_j a_j$$

$$\because m < a_{l+1}$$

$$k_l = 1$$

. □

Now, we write a recursive function with the new sum $m' = m$ - a_l to find new l', so that

$$k_{l'} = 1$$

until $m = 0$.
At the end we have the positions in the key where $k_i = 1$. Put $k_i = 0$ in all other places.

This gives us the complete key.
Equation 2.1 guarantees that the integers a_i increase as i increases. Hence the recursion would start looking for suitable $l's$ from n to 1 and would always terminate because of how m is defined i.e. from the definition of m, after the least non-zero k_i is encountered (we are approaching from the right side), the recursive running sum would become zero, rendering all of the later $k_i's$ to become zero, which is consistent with the definition.

So, Ela doesn't need to solve the subset-sum problem to decipher the key.
The code in python is as follows:

```
import sys
import math
input = sys.stdin.readline

#define a recursive function which returns the largest no. a[i] smaller than m
#then, run it again with sum m-a[i], until the sum is 0. Update the key
#during the process.

def find(a,key,s):
  if s==0:
    print(*key)
    return
  else:
    for i in range(n):
      if a[i]>s:
        key[i-1]=1
        find(a,key,s-a[i-1])
        break

      #special case handled here when m is a number in a[i]

      elif a[i]==s:
        key[i]=1
        print(*key)
        return
        break
```

```
n = int(input())+1
a = list(map(int, input().split()))
#add a large number at the end of the list to make sure our logic works,
#since we have to find the largest no. smaller than m
a = a+[10**9+7]
key=[0]*(n-1)
s = int(input())
find(a,key,s)
```

Question 3

Having falied to arrive at a secret key as above, Anubha and Braj try another method. Let G be the group of $n \times n$ invertible matrices over field F, $n = 128$. Let $a, b, g \in G$ such that $ab \neq ba$. The group G and the elements a, b, g are publicly known. Anubha and Braj wish to create a shared secret key as follows:

> Anubha chooses integers ℓ, m randomly with $1 < \ell, m \leq 2^n$, and sends $u = a^\ell g b^m$ to Braj. Braj chooses integers r, s randomly with $1 < r, s \leq 2^n$, and sends $v = a^r g b^s$ to Anubha. Anubha computes $k_a = a^\ell v b^m = a^{\ell+r} g b^{m+s}$. Braj computes $k_b = a^r u b^s = a^{\ell+r} g b^{m+s}$. The secret key is thus $k = k_a = k_b$.

Show that even this attempt fails as Ela can find k using u and v.

Hint: Show that Ela can

1. find elements x and y such that $xa = ax$, $yb = by$, and $u = xgy$,
2. use x, y, and v to compute k.

Solution

Theorem 3.1. Given the elements $x, y \in G$ such that $xa = ax$ and $yb = by$,

1. $x^{-1}a = ax^{-1}$ and $y^{-1}b = by^{-1}$
2. For some integers r and s, $x^{-1}a^r = a^r x^{-1}$ and $y^{-1}b^s = b^s y^{-1}$.

Proof. 1. As $x \in G$, the inverse of x exists.

$$ax = xa$$

$$x^{-1}ax = x^{-1}xa$$

$$x^{-1}axx^{-1} = ax^{-1}$$

$$x^{-1}a = ax^{-1}$$

Similarly, $y^{-1}b = by^{-1}$ can also be proved.

2.

$$x^{-1}a = ax^{-1}$$

$$\therefore (ax^{-1})a = aax^{-1}$$

$$\therefore (x^{-1}a)a = aax^{-1} (\because \text{ associativity of groups and 1st part of Theorem 3.1})$$

$$\therefore x^{-1}a^2 = a^2x^{-1}$$

By following the same procedure, we can arrive at the general result for r,

$$x^{-1}a^r = a^rx^{-1}$$

and similarly, $y^{-1}b^s = b^sy^{-1}$ can also be proved.

□

Theorem 3.2. There exist elements $x, y \in G$ such that $xa = ax$, $yb = by$, and $u = xgy$.

Proof.

$$xa = ax \implies x^{-1}a = ax^{-1}$$

Now,

$$u = xgy$$

$$\therefore x^{-1}u = gy \quad (\because x \in G)$$

There are a total of 3 linear equations now.

$$x^{-1}a = ax^{-1} \tag{3.3}$$

$$yb = by \tag{3.4}$$

$$x^{-1}u = gy \tag{3.5}$$

As there are 3 linear equations, there are $3n^2$ equations and $2n^2$ variables (number of equations are more than the number of variables). So the system of equations must have at least one non-trivial solution. [Shp08] Now,

$$x^{-1}u = gy$$

$$\therefore x^{-1}uu^{-1} = gyu^{-1}$$

$$\therefore x^{-1} = gyu^{-1}$$

$$\therefore x^{-1}a = gyu^{-1}a = ax^{-1}$$

$$\therefore gyu^{-1}a = agyu^{-1} \text{ and } yb = by$$

Which further simplifies this system of equations to have $2n^2$ equations and n^2 variables. So again, by the same logic, the system of equations must have at least one non-trivial solution. [Shp08] □

Theorem 3.6. x, y, and v can be used to compute the key k.

Proof.

$$u = xgy \tag{3.7}$$

$$v = a^r g b^s \tag{3.8}$$

Now, as $x, y \in G$, their inverse exists. So,

$$u = xgy \implies g = x^{-1}uy^{-1}$$

By substituting the value of g to Equation 3.8 we get,

$$v = a^r(x^{-1}gy^{-1})b^s$$

$$\therefore v = x^{-1}(a^r u b^s)y^{-1} \quad (\because \textit{Theorem } 3.1 \text{ and associativity of groups})$$

$$\therefore xvy = a^r u b^s = a^r(a^l g b^m)b^s = a^{r+l} g b^{m+s} = k$$

□

So this attempt will also fail as Ela can easily determine the key k.

References

[Shp08] Vladimir Shpilrain. Cryptanalysis of Stickel's Key Exchange Scheme. volume 5010, pages 283–288, 06 2008.

Lecture Slides-CS641 Spring 2021, Dr. Manindra Agrawal