CS641

Modern Cryptology
Indian Institute of Technology, Kanpur

Group Name: PolkaDots
Yatharth Goswami (191178), Rohan Baijal (190714), Sarvesh Chandra (190773)

Mid Semester Examination

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: $\mathrm{S1}(\alpha) = \mathrm{S1}(\alpha \oplus 001100) \oplus 1111$.

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

Consider input XOR to S1 as **001100** and the input pair as (β_1, β_2),

$$S1(\beta_1) = S1(\beta_1 \oplus 001100) \oplus 1111 \tag{1.1}$$

$$S1(\beta_1) = S1(\beta_2) \oplus 1111 \tag{1.2}$$

$$S1(\beta_1) \oplus S1(\beta_2) = 1111 \tag{1.3}$$

The output XOR is thus **1111** with probability 1.

We would use Differential Cryptanalysis using 2 round characteristic to break the DES. Following the same procedure as discussed in lecture 6 slide 15, The 2-round characteristic is calculated as, $(6000\bar{0}, \bar{0}\bar{0}, 1, \bar{0}\bar{0}, 6000\bar{0}, 1, 6000\bar{0}, 00808202)$, as input to S1 for round two would be '001100' (discussed in the same lecture) and Permutation ('1111'+'0'*28) can be calculated as 00808202.

The probability of the characteristic is thus 1, implying $L_3R_3 = R_2L_4$ is a fixed constant for each plaintext block pair $L_0R_0, L'_0R'_0$. This further implies that $k_{4,i}$ must necessarily be present in K_i for $1 <= i <= 8$, because K_i represents the set of all possible keys and the values of γ and α_i are fixed constants due to them being dependent only on $L_3 \oplus L'_3$ and R_3 respectively, which are fixed known constants for a given plaintext block pair $L_0R_0, L'_0R'_0$. (here K_i is defined in the same manner as given lecture 7 slide 4)
For a set of l plaintext pairs, the value that is present in every $K_{ij} \quad 1 <= j <= l$ is $k_{4,i}$ as $probability = 1$. For every other key k in K_i, there exists a set of keys K'_i corresponding to another plaintext pair in which k is not present.

We can now formulate the above idea in an algorithm as follows

```
Given : The two round characteristic is: $(6000\bar{0}, \bar{0}\bar{0}, 1, \bar{0}\bar{0}, 6000\bar{0}, 1, 6000\bar{0}, 00808202)$,
 $L_0R_0, L_4R_4, L'_0R'_0, L'_4R'_4, E(), S(), R_3 = L_4, R'_3 = L'_4$.
Want : $k_4$
1. Calculate $\alpha_i$, $\alpha'_i$, $\beta_i \oplus \beta'_i$, $\gamma_i \oplus \gamma'_i$ as given in lecture 7 page 3
2. Calculate $X_i, K_i$ as defined in Lecture 7 page 4
3. After performing steps 1 and 2 on the first plaintext block, continue to perform
 1, 2 for $l$ pairs of plaintext blocks $L_0R_0, L'_0R'_0$ as below
4. for $i = 1$ till $i = 8$ do
    $P = K_{i0}$
    $j = 0$    //j denotes the number of plaintext pairs currently generated
    While $n(P) > 1$ {    // n(P) denotes cardinality of P
        Steps 1 and 2 with a new plaintext block pair
        $P = P \cap K_{ij}$    $//K_{ij}$ denotes the set $K_i$ for $j$th pair
        j+=1
    }
    $k_{4,i} = P$
end
5. $k_4 = k_{4,1} \dots k_{4,8}$
6. End
```

Algorithm 1: Find Key

The attack discussed above is pretty efficient in the sense that we only require a handful of plaintexts pairs in order to derive key for the last round, as the loop terminates as soon as the cardinality of P becomes 1. According to the analysis done in class, we would require around $l = 20/p$ plaintext pairs, since $p = 1$ here, taking around 20 pairs would identify key uniquely for most cases. Also, once we obtain the key for round 4, we will end up with 48 out of 56 bits of the master key. We can either run brute force attack on the rest of the 8 bits or if keys are independent we can convert the DES to 3 round DES, and after getting k_3 using a similar approach as discussed above, convert this DES to 2-round DES, which can be solved easily as discussed in lecture 5. We thus get the keys to all 4 rounds, which breaks the DES completely.

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 want to find an n-bit key k. We will provide a method that will find the right key only for the case when m is actually a possible subset-sum.

Theorem 2.1. $k_n = 1$ iff $m > \sum_{i=1}^{i=n-1} a_i$.

Proof. $\Rightarrow$ Suppose $k_n = 1$.
Then, $m = \sum_{i=1}^{i=n-1} a_i k_i + a_n k_n$.
Now, since all a_is are positive numbers, therefore $m \geq a_n k_n = a_n$. But $a_n > \sum_{i=1}^{i=n-1} a_i$.
This implies that : $m > \sum_{i=1}^{i=n-1} a_i$ □

Proof. $\Leftarrow$ We will do proof by Contradiction.
Suppose $m > \sum_{i=1}^{i=n-1} a_i$ but $k_n = 0$.
We know that $m = \sum_{i=1}^{i=n} a_i k_i = \sum_{i=1}^{i=n-1} a_i k_i$. [Because $k_n = 0$] .
Now, the maximum value of m we can now get is $m_{max} = \sum_{i=1}^{i=n-1} a_i$. [By setting $k_i = 1 \forall i$]
But, we know that $m > m_{max}$ from our hypothesis, which provides a fallacy hence we derive a contradiction as a result, which implies that our assumption was wrong.
This implies $k_n = 1$. □

Let's make an important observation. This result is valid only for figuring out the last bit of k, but we can easily find all the bits using recursion on $(a_1, a_2, ...a_{n-1}, m - k_n a_n)$. We can now extend this result into an algorithmic solution.

```
Given : $m, n, a_1, a_2, ..., a_n$.
Want : $k$
for $i = n$ down to 2 do
    $k_i = 0$
    if $m > \sum_{j=1}^{j=i-1} a_j$ then
        $k_i = 1$
    end
    $m = m - k_i a_i$
end
if $m == a_1$ then
    $k_1 = 1$
end
else
    $k_1 = 0$
end
```

Algorithm 2: Find Key

Proof of Correctness

We prove the correctness by **induction on** n.

Base Case: if $n == 1$, we will have only a_1 to compare to. Hence, trivially the algorithm outputs correct 1 bit key for this case.

Proposition : Suppose the algorithm correctly finds a key k of length $n - 1$.

The loop starts by finding the n^{th} bit of k. [**Theorem 2.1**].

Now m is updated to give $m' = m - k_n a_n$. We can make this update since we know that solution exists for m and hence a solution will exist for $m - k_n a_n$ as well.

This m' is the same SUBSET SUM problem for a key of length $n - 1$. And by the assumption, the algorithm will find the key.

Hence, the algorithm finds a key of length n correctly.

Hence, Proved.

As we can see, we get an $O(n)$ algorithm (summations of a_i can be stored in a pre computed array) to find the key k. We don't need to solve a hard problem :)

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

In the solution ahead, we will be using a, b and g as described in the problem to be available publicly. All other variables like u and v also remain same. Let us denote the random integers chosen by Anubha as ℓ and m and Braj's as r and s for generation of private key.

Claim 3.1. There exists non-trivial elements x, y in G such that $xa = ax$, $yb = by$ and $u = xgy$ or the system of equations has atleast one non-trivial solution.

Proof. This claim can be easily proven, seeing the fact that $x = a^\ell$ and $y = a^m$ provides a solution to the system of equations. Therefore, the above system of equations is consistent. □

Claim 3.2. If $pq = qp$ for some $p, q \in G$, then $pq^\alpha = q^\alpha p$ for all $\alpha \geq 1$.

Proof. We will prove this claim using induction on α. For base case, we take $\alpha = 1$ which lead to our hypothesis $pq = qp$ only. Hence, base condition is satisfied. Now, we assume that it is true for some $i > 1$ and we will prove that it holds for $i + 1$ as well.

$$pq^i = q^i p$$

$$
\begin{aligned}
pq^i q &= q^i pq \\
pq^{i+1} &= q^i qp \\
pq^{i+1} &= q^{i+1} p
\end{aligned}
$$

Hence, proved using induction that the claim is true for all $\alpha \geq 1$ □

Claim 3.3. If we have a solution for the the set of equations $xa = ax$, $yb = by$ and $u = xgy$ then we can obtain the private key k as just xvy.

Proof. We notice that $v = a^r gb^s$. Now, plugging in this value, we get $xvy = xa^r gb^s y$. Now, using **claim 3.2**, this rearranges to $xvy = a^r xgyb^s$. Now, substituting u, we get $xvy = a^r ub^s$ which is equal to k_b which is in turn equal to k. Hence proved. □

Now, the only task is to find the values of such x and y efficiently. We notice that out of the three equations, only the last one is non-linear in nature. We use the fact that x belongs to the group of invertible matrices and hence, the third equation transforms to $x^{-1}u = gy$. But now the variables have changed and we have introduced a new variable x^{-1}, therefore we will rearrange the first equation as well to it's equivalent form by pre-multiplying and post-multiplying with x^{-1} to $x^{-1}a = ax^{-1}$. Let $x^{-1} = x'$, which transforms the three equations as

$$
\begin{aligned}
x'a &= ax' \\
yb &= by \\
x'u &= gy
\end{aligned}
$$

The third equation gives us $x' = gyu^{-1}$. Substituting this in first equation leaves $gyu^{-1}a = agyu^{-1}$. Now, we are left with two linear equations and just one unknown which is y. The existence of a solution can be easily seen using **claim 3.1**. We have $2n^2$ linear equations for n^2 variables in matrix y and we can make an augmented matrix of size $2n^2 \times (n^2 + 1)$. Hence, existence of a non-trivial solution guarantees the presence of at least one free variable in the reduced echelon form of the matrix for solving the system of equations. Considering the fact that we have an overwhelming excess of equations over the variables, the number of free variables will also not be much probabilistically. An intuititve way of seeing the previous fact, relies on noticing that since we have to make final rank of echelon matrix to be less than n^2, therefore we would have to eliminate more than n^2 rows atleast and therefore we will generate more pivots while doing so since we would have to use large number of row operations and therefore since pivots would be more, the number of free variables would be less in general. Hence, we can go over values of these free variables to generate an invertible matrix y. Also, notice the fact that checking for invertibility of a matrix can be easily done by converting it to echelon form which takes roughly cubic operations using Gauss-Jordan Elimination. This analysis allows us to break the enryption scheme completely.

Complexity of operations: Considering the fact that the probability of finding singular matrices of order $n \times n$ in $\mathbb{R}^{n^2}$ is negligible for large n [1] which can be extended to any field, we can be sure to find an invertible matrix solution to the above system in finitely many attempts. Also, the matrix multiplication can be completed in cubic time complexity and Gauss-Jordan elimination also takes cubic time, which are pretty much within modern computational limits.

References

[1] Probability of random matrix being singular. https://math.stackexchange.com/questions/226128/probability-of-having-zero-determinant.