CS641
Modern Cryptology
Indian Institute of Technology, Kanpur

Mid Semester Examination

Group Number: Codagami
Chinmaya Singal (180207), Dipesh Khandelwal (180249), Rythm Agarwal (180636)

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 notice that if we pick a pair of plaintexts such that the XOR of the input (α, β) to S-box S1 is 001100 then:

$$\begin{aligned} S1(\alpha) &= S1(\alpha \oplus 001100) \oplus 1111 \\ S1(\alpha) &= S1(\beta) \oplus 1111 \\ \implies S1(\alpha) \oplus S1(\beta) &= 1111 \end{aligned}$$

Hence the XOR of output of S-box S1 will be 1111 if the XOR of input to it is 001100. Using this fact we can break four round of this variant using an algorithm similar to the one discussed in class. It works as follows:

1. Pick a pair of plaintexts such that their right halves are the same (hence the XOR is 0) and the left halves are such that the XOR of their expansion would have first six

bits (Input to S1) as 001100. Therefore, the XOR of the left halves should be 60000000 in hexadecimal.

2. Let (L_0, R_0) and (L'_0, R'_0) be the input plaintext pairs and (L_i, R_i) and (L'_i, R'_i) be the output pairs after i^{th} round of DES.

3. Now, we can know the values or the XOR of the values in the two encoding at various parts of the algorithm as discussed in class but not at all the locations.

4. Since, the XOR of input to S-box S1 in the second round of DES is 001100 (due to the nature of the pair of plaintexts chosen), we know that the XOR of output of the S-box S1 must be 1111 as shown above.

5. Hence, we can calculate $R_2 \oplus R'_2$ which comes out to be 008080202 (Permutation taken from [Sch96]). Using this we know $L_3 \oplus L'_3$ (Because $L_3 = R_2$) and since we know the ciphertexts (L_4, R_4) and (L'_4, R'_4) we can find the XOR of output of S-box in fourth round of DES.

6. Also, using L_4 and L'_4 we can find the XOR of input to S-boxes in fourth round of DES since $R_3 = L_4$.

7. Thus, we have reached a situation similar to what we had for breaking three round DES as discussed in lectures (Lecture 6 Slides 6, 7, 8, 9, 15). So by using a similar method of making sets X_i and K_i and then taking set intersection of K_i over various plaintext pairs we can find $k_{4,i}$. Note that there is one difference which is when the XOR of output of S-box S1 in fourth round is 1111 then $|K_1| = 64$ which is not helpful so we will have to pick such that the XOR of output of S1 in fourth round is not 1111 for finding $k_{4,1}$.

8. Using this we can find the key k_4. After this, for finding the remaining keys we can just use the algorithm as discussed in class. Only difference would be while finding k_3 we will have to take care to choose plaintexts such that XOR of output of S-box S1 is not 1111. The rest of the steps will remain exactly the same.

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

Ela can retrieve the key k from $(a_1, a_2, \ldots, a_n, m)$ using the following algorithm:

1. Initialize the key bits ($k[1 \text{ to } n]$) to 0.
2. Initialize sum to m. ($sum = m$)
3. for $i = n$ to 1 do:
   (a) If $sum \geq a_i$ do:
      i. $k[i] = 1$;
      ii. $sum = sum - a_i$
4. return k

Using this algorithm, Ela can retrieve the key k without solving the SUBSET-SUM problem. The proof why this is true is as follows:
Proof: After each iteration of the for loop, the value of sum will necessarily be less than a_i. This is because $\forall i, a_i > \sum_{1 \leq j < i} a_j$. Hence:

1. If before the iteration, $sum < a_i$ then we do nothing and it remains as is.

2. If before the iteration, $sum \geq a_i$ then we subtract a_i from sum. Hence, the maximum possible value of sum after iteration would be $\sum_{1 \leq j < i} a_j < a_i$. This is because if our claim is true then we have already subtracted $a_k, k > i$ wherever possible and hence $sum < a_k \forall k > i$.
   Since this is trivially true before first iteration, hence by induction it is true for all iterations.

Also, at any iteration, if $sum \geq a_i$, then it must be that $k_i = 1$. This is because if that is not the case (that is, if $k_i = 0$), then the maximum value of sum is $sum \leq \sum_{1 \leq j < i} a_j < a_i \leq sum$ (By the claim proved above, $sum < a_k \forall k > i$) which is a contradiction. Hence, $k_i = 1$ if $sum \geq a_i$.
Therefore, by iterating over all the bits, Ela can retrieve the key without solving the SUBSET-SUM 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

Going by the same lines as given in the hint,
First we try to find elements $x, y \in G$ s.t.

1. $xa = ax$
2. $yb = by$
3. $u = xgy$

As, we know that $x \in G$, hence x is invertible.
$\therefore$ We can rearrange the third equation as $x^{-1}u = gy$.
Also, the first equation can be written as $ax^{-1} = x^{-1}a$ by pre and post multiplying both sides by x^{-1}.
Now eliminating x^{-1}:
As, u is formed by multiplying matrices from G, therefore $u \in G$. Now, using equation $x^{-1}u = gy$, we have $x^{-1} = gyu^{-1}$.

Putting the value of x^{-1} in $ax^{-1} = x^{-1}a$, we get

$$agyu^{-1} = gyu^{-1}a$$

Now, we have two equations $agyu^{-1} = gyu^{-1}a$ and $yb = by$. Each of these matrix equations will give us n^2 linear equations. This gives us $2n^2$ linear equations in n^2 variables (treating every element of matrix y as variable we get n^2 variables). Now, this system of linear equations can be solved easily to find the matrix y as it is a overdetermined system of linear equations. This system of linear equations will have at least one non-trivial solution, namely, $x = a^l$ and $y = b^m$. Hence, the system of linear equations is solvable and we can use any one of the valid solutions to find x and subsequently the key as shown below.

Thus, we can compute y. Using y and equation $u = xgy$, we can compute x. So, our first step of hint is complete. We have found elements x and y such that $xa = ax$, $yb = by$, and $u = xgy$.

Going on to step 2 of the hint.

We can find the key after calculating the matrix x and y by computing xvy

$$\begin{aligned}
xvy &= xa^r gb^s y \\
&= axa^{r-1} gb^{s-1} yb \qquad (\because xa = ax \text{ and } yb = by) \\
&= a^r xgyb^s \qquad (\text{By successively using } xa = ax \text{ and } yb = by) \\
&= a^r ub^s \\
&= k
\end{aligned}$$

As matrices x, v, y are known to us, hence we can compute the key $k = xvy$.

Hence, even this attempt fails since Ela can find k using u and v by finding matrices x and y satisfying the equations as given above and then calculating $k = xvy$.

References

[Sch96] Bruce Schneier. *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth)*. John Wiley and Sons, Inc., 1996.