CS641

Modern Cryptology

Indian Institute of Technology, Kanpur

Group Name: The Kryptonians

Maulik Singhal (190489), Maryam Raza Khan (190488), Pranshu Gaur (200703)

Mid Semester Examination

Submission Deadline:
March 3, 2022, 23:55hrs

Question 1

Consider a variant of DES algorithm in which all the S-boxes are replaced. The new S-boxes are all identical and defined as follows.

> Let $b_1, b_2, \cdots, b_6$ represent the six input bits to an S-box. Its output is $b_1 \oplus (b_2 \cdot b_3 \cdot b_4), (b_3 \cdot b_4 \cdot b_5) \oplus b_6, b_1 \oplus (b_4 \cdot b_5 \cdot b_2), (b_5 \cdot b_2 \cdot b_3) \oplus b_6$.

Here '$\oplus$' is bitwise XOR operation, and '$\cdot$' is bitwise multiplication. Design an algorithm to break 16-round DES with new S-boxes as efficiently as possible.

Solution

We will use chosen-plaintext attack to break 16-round DES.We will use differential cryptanalysis to find the key. Consider the differential 000010 going into the S-box S2 after passing through the expansion block. As we are taking xor values they can pass through permutation block without any changes. Input to other S-boxes is 000000.
Let first input with the given differential be $b_0b_1b_2b_3b_4b_5$ and the corresponding output be $c_0c_1c_2c_3$. For differential 000010 second input would be $b_0(b_1 \oplus 1)b_2b_3b_4b_5$ and let its corresponding output be $c'_0c'_1c'_2c'_3$. Then we can say

$$c_0 \oplus c'_0 = 0$$

$$c_1 \oplus c'_1 = b_3b_4$$

$$c_2 \oplus c'_2 = b_2b_4$$

$$c_3 \oplus c'_3 = b_2b_3$$

Therefore the differential output is 0000 with probability $\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{1}{2}$
Cases :

$$i.\, b_3 = b_4 = 0 \; and \; b_2 = 1$$

$$ii.\, b_3 = b_2 = 0 \; and \; b_4 = 1$$

$$iii.\, b_2 = b_4 = 0 \; and \; b_3 = 1$$

$$iv.\, b_2 = b_3 = b_4 = 0$$

As differential input to other boxes is 000000, the differential output is 0000 after going through the boxes.
Let Z be zero differential for 32 bits - 0000 0000 0000 0000 0000 0000 0000 0000
Let P differential be for 32 bits - 0000 0001 0000 0000 0000 0000 0000 0000
For first round we take left half round input L_0 as P and right half round input R_0 as Z. We get the output $L_1 = R_0 = Z$ and $R_1 = Z \oplus P = P$ with probability 1
For second round we take left half round input L_1 as Z and right half round input R_1 as P. We get the output $L_2 = R_1 = P$ and $R_2 = Z$ with probability $\frac{1}{2}$

Therefore 2-round characteristic equation can be written as :

$$[P, Z] \xrightarrow{p=1} [Z, P] \xrightarrow{p=\frac{1}{2}} [P, Z]$$

Using this high probability 2-round characteristic equation and extending it to 16 round characteristic equation. The total probability of 16 round characteristic equation we get is p = $(1 * \frac{1}{2})^8$ = $\frac{1}{256}$. As per the analysis done in lecture 7, number of input pairs(l) required would be l = $\frac{20}{p}$ = 20*256 = 5120 pairs. Therefore using few thousand pairs key can be recovered and 16-round DES can be broken.

Question 2

Suppose Anubha and Braj decide to do key-exchange using Diffie-Hellman scheme except for the choice of group used. Instead of using F_p^* as in Diffie-Hellman, they use S_n, the group of permutations of numbers in the range $[1, n]$. It is well-known that $|S| = n!$ and therefore, even for $n = 100$, the group has very large size. The key-exchange happens as follows:

> An element $g \in S_n$ is chosen such that g has large order, say l. Anubha randomly chooses a random number $c \in [1, l-1]$, and sends g^c to Braj. Braj choses another random number $d \in [1, l-1]$ and sends g^d to Anubha. Anubha computes $k = (g^d)^c$ and Braj computes $k = (g^c)^d$.

Show that an attacker Ela can compute the key k efficiently.

Solution

We show that it is possible to calculate either of **c** or **d** efficiently and hence find the key.

Assumptions : Ela has access to n, g, g^c, g^d

As finding either of **c** or **d** is equivalent, without loss of generality we find the value of **c**.

First we break the groups into disjoint cycles including the unchanged elements in the form of single cycles as follows:
$g = \sqcup_{i=1}^{p} A_i$
$g^c = \sqcup_{i=1}^{q} B_i$

Next, we create two functions X_1, X_2 for 'g' such that
$X_1(i) = x$, index of cycle A_x containing the element **i**
$X_2(i)$ = position of the element **i** in the corresponding cycle

Similarly we create two functions Y_1, Y_2 for 'g^c' such that
$Y_1(i) = a$, first element of the cycle B_i
$Y_2(i) = b$, second element of the cycle B_i

Note: If the cycle B_i contains only a single element, then $Y_2(i) = Y_1(i) = a$

Claim 1: All the elements of B_i occur in the same cycle A_j for some $j \in \{1, 2, ...p\}$

Proof: Suppose not, then there exists an $x \in \{1, 2, ..., n\}$ such that $x \in B_i$ but $x \notin A_j$ and ${A_j}^c = B_i$
Now, $x \in B_i \Rightarrow x \in {A_j}^c \Rightarrow x \in A_j$
Contradiction !
Hence the claim stays.

Now, $Y_1(i)$ and $Y_2(i)$ belong to the same cycle B_i and hence by Claim 1 they belong to the same cycle A_j for some $j \in \{1, 2, ..., p\}$

Then, for each $i \in \{1, 2, ..., q\}$ we find the corresponding j such that $Y_1(i) \in A_j$ and $Y_2(i) \in A_j$ This can be done in time complexity $O(n)$.

We create another function $Len(i)$ such that $Len(i)$ = length of the cycle containing the element **i**.

Now, we form a sequence M_i such that M_i = position of $Y_2(i)$ in A_j — position of $Y_1(i)$ in A_j

Now, we have got $|M_i|$ linear equations in the form of
$c \equiv M_i$ (mod $Len(i)$) This can be solved by the Chinese Remainder Theorem.
We got **c** through the above steps and now we can calculate the key.

Calculation of Key : As we know the value of c, we can calculate the value of the key as $k = (g^d)^c$. This can be done by simple modular arithmetic in time complexity of $O(n)$.

Hence, the key **k** can be calculated efficiently by Ela.
The total time complexity of computing the key is O($n^2 log^2 n$).

References

CS641A lecture slides by Dr. Manindra Agrawal