# CS641
Modern Cryptology
Indian Institute of Technology, Kanpur

# Mid Semester Examination

Group Name: Goldfish
Jeet Sarangi(21111032), Akshay Kumar Chittora (21111007), Alok Kumar Trivedi(21111008)

Submission Deadline:
March 1, 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

According to the given question we have a 16 round modified DES algorithm to crack the key, now to do so we are using differential cryptanalysis.
So, we will predict the output of 14th round for which we will need a 14round characteristics with as high probability as possible to precisely predict the key in the last round.

Assuming the initials values for the round 1 of DES $L_0 = \bar{0}\bar{0}$ $R_0 = 8000\bar{0}$

Assuming the two inputs to S-box S1 be $\alpha_1$ and $\alpha_2$ after expansion of the right half of the plaintext and $a_i$ where $0 < i < 5$ and $a_i'$ where $0 < i < 5$ be their corresponding outputs from S-box. Taking the two inputs to the S boxes be: $\alpha_1 = b_1b_2b_3b_4b_5b_6$ , $\alpha_2 = b_1(b_2 \oplus 1)b_3b_4b_5b_6$ And hence, $\alpha_1 \oplus \alpha_2 = 010000$

Now the input to S-box is 010000, as we can see $\alpha_1$ and $\alpha_2$ satisfy the above condition $\alpha_1 \oplus \alpha_2 = 010000$ , we can see the XOR of the output of the S-boxes as 0000 with probability,

$$\frac{1}{4} + \frac{1}{4} + \frac{1}{4} = \frac{3}{4}$$

Now we know that, $a_1 \oplus a_1' = b_3 b_4, \quad a_2 \oplus a_2' = 0, \quad a_3 \oplus a_3' = b_4 b_5, \quad a_4 \oplus a_4' = b_5 b_3$

Now 000001 goes to S-box S2. And $\alpha_1$ and $\alpha_2$ be the inputs to the S-box and $a_1 a_2 a_3 a_4$ and $a_1' a_2' a_3' a_4'$ be the corresponding outputs. Then:

$$\alpha_1 = b_1 b_2 b_3 b_4 b_5 b_6$$

$$\alpha_2 = b_1 b_2 b_3 b_4 b_5 (b6 \oplus 1)$$

$$\alpha_1 \oplus \alpha_2 = 000001$$

And, $a_1 \oplus a_1' = 0 \quad a_2 \oplus a_2' = 1 \quad a_3 \oplus a_3' = 0 \quad a_4 \oplus a_4' = 1$. Now, the output of the S-box can be predicted as 0101 with probability 1. After performing permutation on the values obtained from S-box, all the 1's go to different blocks and we got $R_1$=00000820 and $L_1 = 80000\bar{0}$ with probability=$\frac{3}{4}$

Next Round:

Now only three s-boxes will have non-zero inputs. Let $\alpha_1$ and $\alpha_2$ be the input value to S-box S1.

$\alpha_1 = b_1 b_2 b_3 b_4 b_5 b_6 \quad \alpha_2 = b_1 b_2 (b_3 \oplus 1) b_5 b_6$

Let $a_1 a_2 a_3 a_4$ and $a_1' a_2' a_3' a_4'$ be the outputs of S-box for $\alpha_1$ and $\alpha_2$ respectively.

$$a_1 \oplus a_1' = b_2 b_4$$

$$a_2 \oplus a_2' = b_4 b_5$$

$$a_3 \oplus a_3' = 0$$

$$a_4 \oplus a_4' = b_2 b_5$$

The output can be predicted as 0000 with probability $\frac{3}{4}$ i.e. $(\frac{3}{4})$

Similarly, if the input is 010000, then the output can be predicted as 0000 with probability = $\frac{3}{4}$

Similarly, if the input is 000001, as above the output can be predicted as 0101 with probability 1.

After performing permutation on the values obtained from the S-boxes, we get $L_2$ as

00000820 and perform xor between the permuted output and $L_1$ to obtain $R_2$ as A0004000 with probability =

$$\frac{3}{4} * \frac{3}{4} * 1 = \frac{9}{16}$$

*NextRound*

After performing expansion on the $R_2$, 3 of the S-boxes will get non-zero differentials, considering each of those and then permuting the same we get $L_3$ as A0004000 and $R_3$ as $\bar{0}\bar{0}$ with probability $\frac{3}{16}$. Again only 3 s-boxes will have non-zero differential,By modifying each of those

We repeat the above steps for 5 more rounds and obtain the following characteristics :

$R_4$ ( A0004000 ,$\bar{0}\bar{0}$ ,1)
$R_5$ ( 00000820 , A0004000,$\frac{9}{16}$)
$R_6$ (80000000 ,00000820 ,$\frac{3}{4}$)
$R_7$ ( $\bar{0}\bar{0}$, 80000000 ,$\frac{3}{16}$)
$R_8$( 80000000 , $\bar{0}\bar{0}$ , $\frac{3}{4}$)

The 9th round differential is same as 1st round differential and using the same repetation till the 14th round we can break the modified DES.

The probability of the characteristic till round 8 is multiplication of all the probabilities of the all the rounds, which is:

$$(\frac{3}{4})(\frac{9}{16})(\frac{3}{16})(1)(\frac{9}{16})(\frac{3}{4})(\frac{3}{16})(\frac{3}{4}) = (\frac{3^9}{2^{22}})$$

And we know about the 9th round too,as it will be same as first round, we iterate the same process for 6 more rounds to get the 14th round characteristics with a certain probability , p= $\frac{3^{15}}{2^{36}}$

Now using the above obtained results, we can break the DES by following process.

last round key analysis: ($k_r$)

Define

$$X_i = \{(\beta, \beta') | \beta \oplus \beta' = \beta_i \oplus \beta_i' and S_i(\beta) \oplus S_i(\beta') = \gamma\}$$

pair $(\beta_i, \beta_i') \in X_i$ when $\gamma_i \oplus \gamma_i' = \gamma$ is correct (which happens to be with the above probability).

$$K_i = \{k | \alpha_i \oplus k = \beta and (\gamma, \gamma') \in X_i for some \beta'\}$$

$(\beta_i, \beta_i') \in X_i$ with probabilty $\geq \frac{3^{15}}{2^{36}}$ , we have $k_{16}) \in K_i$ with above probability.

Then $K_r$ can be calculated using the $X_i$ set as: $K_i = \beta_i \oplus \alpha_i$ and $\beta_i$ and $\alpha_i$ are known with as high probability as possible .

# 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

## Overview

Diffie-Hellman scheme of key exchange works on the basis of the principle of Discrite logarithm problem which works as given a symmetric group $S_n$.And given that an element $g$ is publicly known and we do composition on $g$ , $\alpha$ times such that $h = g^\alpha$ and the elements $g$ and $h$ are known discrete log problem is based on task to calculate $\alpha$.

In the above both Anubha and Braj has there own private variables $c$ and $d$ using which they calculate the compositions of $g^c$ and $g^d$ respectively. We need to devise an algorithm which can calculate the values c and d using h and g(publicly known values).And the constraint is the algorithm shall be computationally feasible to calculate the private values in real time. So, the key can then be easily be calculated by $g^{c \cdot d}$

## Implementation

As we know that $g$ and $h$ are publicly available and goal is to compute $\alpha$. Any elements $e \in S_n$ can represented via cycle notation or a list of images $[e(1), e(2), \cdots, e(n)]$. Writing $g$ and $h$ in cycle-notation

$$g = g_1 \circ g_2 \circ \cdots \circ g_r$$

$$h = h_1 \circ h_2 \circ \cdots \circ h_s$$

where the the composition operation is denoted by $\circ$. Here we have included length one cycles so that every $i \in \{1, 2, \cdots, n\}$ lies in exactly one cycle.

Now Maintaining arrays $G$ and $H$ , where $G[y]$ stores the index of $y$ within the cycle and the position of the $y$ in the cycle.
The $i$-th index of $G[i]$ contains:

- the index $j$ of the cycle $g_j$ having $i$
- the position of $i$ within this cycle $(1 \leq i \leq n)$

Similarly $H[i]$ will also be maintained. :

- the index $k$ of the cycle $h_k$ having $i$
- the location of $i$ within this cycle $(1 \leq i \leq n)$

$G[i]$ can be considered as a tuple $(j, p(i))$ which stats that element $i$ appears in cycle $g_j$ at a position $p(i)$.
Like above $H[i] = (k, p(i))$ can also be considered as a tuple which states the element $i$ appears in cycle $h_k$ at location $p(i)$.

We now need two new arrays $X[k], Y[k]$ . X and Y stores first and second elements of each cycles $h_k$ of h.If we get X[k] = Y[k] that represents cycles of length = 1.Using array G we can find the cycle of g containing X[k] and Y[k] for $k \in n$.Then for each element we calculate the length of the cycle containing it and put it in array W.

Now To obtain the value of $\alpha$, we need to use Chinese Remainder Theorem(if one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime). And we have equations formed as

$$\alpha \equiv Z[i] \mod W[i] \text{ for } 1 \leq i \leq |Z|.$$

So we have around $|Z$-linear equations, for $i, j$ $\gcd(W_i, W_j) \neq 1$ which we need to solve and get $\alpha$.

**Time Complexity:**
To decompose $g$ and $h$ in cycle notation, we take one by one $i$ and find the image of $i$ under $g$. This step requires $O(n)$-time.
Then to compute the values of $G$ and $H$ arrays having $2n$ integers requires $O(n)$-time.

To compute arrays $X$ and $Y$ we need $O(n)$ lookups and also we need $O(n)$ subtraction operation.Now to compute the values of alpha , we need to solve order n equation. Then we need to perform n-1 times extended euclidean algorithms,thus the time complexity will be

$$O(\sum_{k=1}^{n-1} k \cdot \log^2 n) = O(n^2 \log^2 n).$$