CS641
Modern Cryptology
Indian Institute of Technology, Kanpur

Mid Semester Examination

Group Name: Lazarus
Aditya Loth (21111004), Harsh Agarwal (21111030), Varun Vankudre (21111064)

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 referred Professor's lecture slides[Agg] and paper [BS91] for solution guidance and key extraction procedure
We know that to break 16 round DES efficiently we need to find 14 round characteristic with as high probability as possible.
To find the 14 round characteristic we use differential cryptanalysis.

$$a_1 = b_1 \oplus (b_2 \cdot b_3 \cdot b_4)$$

$$a_2 = (b_3 \cdot b_4 \cdot b_5) \oplus b_6$$

$$a_3 = b_1 \oplus (b_4 \cdot b_5 \cdot b_2)$$

$$a_4 = (b_5 \cdot b_2 \cdot b_3) \oplus b_6$$

Let input differential to S-box S2 be 000100. Let one of the inputs satisfying given differential be $b_1b_2b_3b_4b_5b_6$ with output as $a_1a_2a_3a_4$. Let second input satisfying differential mentioned earlier be $b_1b_2b_3(b_4 \oplus 1)b_5b_6$ with output $a'_1a'_2a'_3a'_4$.

Therefore,

$$a_1 \oplus a'_1 = b_2 \cdot b_3$$
$$a_2 \oplus a'_2 = b_3 \cdot b_5$$
$$a_3 \oplus a'_3 = b_2 \cdot b_5$$
$$a_4 \oplus a'_4 = 0$$

So output differential is 0000 with respect to given input differential with probability $\frac{1}{2}$ (at least two out of b_2, b_3, b_5 is 0)

Let input differential to S-box S1 be 000000. Let one of the inputs satisfying given differential be $b_1b_2b_3b_4b_5b_6$ with output as $a_1a_2a_3a_4$. Let second input satisfying differential mentioned earlier be $b_1b_2b_3b_4b_5b_6$ with output $a'_1a'_2a'_3a'_4$.

Therefore,

$$a_1 \oplus a'_1 = 0$$
$$a_2 \oplus a'_2 = 0$$
$$a_3 \oplus a'_3 = 0$$
$$a_4 \oplus a'_4 = 0$$

So output differential is 0000 with respect to given input differential with probability 1.
Similarly for all other S-boxes from 3 to 8, input differential is 000000 and output differential respect to this is 0000 with probability 1.
After combining output differential of all 8 S-boxes we can get below output with probability $\frac{1}{2}$

0000 0000 0000 0000 0000 0000 0000 0000

And as left half of block is considered to be all 0, right half of the next block is also 32 bits of 0.

Let,

$$D_1 = 00000000$$

$$D_2 = 04000000$$

where each 0 represents 4 bits of 0 i.e. 0000

Therefore 2 round characteristic we get is mentioned below. The probability associated with it is 2^{-1} i.e $\frac{1}{2}$

$$(D_2, D_1, 1, D_1, D_2, 0.5, D_2, D_1)$$

To get 14 round characteristic from above 2 round characteristic we iterate it 7 times. So we get 14 round characteristic with probability of 2^{-7} i.e $\frac{1}{128}$

Extracting the key

Let

$$E(R_{15}) = \alpha_1\alpha_2...\alpha_8$$

$$E(R'_{15})) = \alpha'_1\alpha'_2...\alpha'_8$$

$$\text{such that } |\alpha_i| = 6 = |\alpha'_i|$$

Let

$$k_{16} = k_{(16,1)}k_{(16,2)}...k_{(16,8)}$$

Let

$$\beta_i = \alpha_i \oplus k_{(16,i)}$$

$$\beta'_i = \alpha'_i \oplus k_{(16,i)}$$

$$\text{so } |\beta_i| = 6 = |\beta'_i|$$

Let

$$\gamma_i = S_i(\beta_i)$$

$$\gamma'_i = S_i(\beta'_i)$$

$$\text{so } |\gamma_i| = 4 = |\gamma'_i|$$

We have determined a γ value such that $\gamma_i \oplus \gamma_i' = \gamma$ occurs with probability $\frac{1}{2}$ for certain input differential(s)

Define

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

Pair $(\beta_i, \beta_i') \in X_i$ whenever our guess for $\gamma_i \oplus \gamma_i' = \gamma$ is correct . which happens with probability $\frac{1}{2}$

Define

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

Since, $(\beta_i, \beta_i') \in X_i$ with probability $\geq \frac{1}{2}$, we have $k_{(16,i)} \in K_i$ with probability $\geq \frac{1}{2}$.

We know $|K_i| = |X_i|$ since α_i and $\beta \oplus \beta'$ is fixed for (β,β') $\in X_i$.

If the probability of characteristics is p and we use l plain text block pairs, $k_{16,i}$ will be present in $(pl + \frac{l}{4}(1-p))\ K_i's$.

Any other a $\neq$k$_{16,i}$ will be present in about $\frac{l}{4}$ pairs.

So, $k_{16,i}$ is present in roughly $\frac{3}{4}pl$ additional pairs.

We need $l \approx \frac{20}{p}$ in order to ensure that $k_{16,i}$ is most frequently occurring value.

So, the number of plain text block pairs required is

$$l \approx \frac{20}{\frac{1}{128}} \approx 2^{12}$$

which is less than brute force.

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 have referred to this article [CC21] to figure out the solution.
A finite symmetric group S_n defined over set $\{1, 2, ..., n\}$ is the group in which all elements are bijections from the set to itself and the group operation is composition of functions.
Composition of function is an operation that takes two functions f and g and produces a function $h = g \circ f$ such that $h(x) = g(f(x))$

Cycle Notation

Cycle notation describes the effect of repeatedly applying the permutation on the elements of the set. It expresses the permutation as a product of cycles; since distinct cycles are disjoint, this is referred to as "decomposition into disjoint cycles". e.g.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 2 & 1 & 4 \end{pmatrix}$$

What this means is

$$1 \to 5 \to 4 \to 1$$
$$2 \to 3 \to 2$$

This is in Two line notation and this can be written in cycle notation as

$$\sigma = (154)(23)$$

Generally Cycle of length 1 are not represented in this notation. Cycle Notation is the notation we will be using through out the solution for question.

Let there be 2 permutations

$$\sigma_1 = (154)(23) \text{ and } \sigma_2 = (12345)$$

$$\begin{aligned}
\sigma_1 \circ \sigma_2 &= (154)(23) \circ (12345) \\
&= (13) \quad \text{(Note 2,4,5 form 1 length cycle as are not mentioned)}
\end{aligned}$$

$$\begin{aligned}
\sigma_1 \circ \sigma_1 &= (154)(23) \circ (154)(23) \\
\sigma_1^2 &= (145) \quad \text{(Note 2,3 form 1 length cycle as are not mentioned)}
\end{aligned}$$

$$\begin{aligned}
\sigma_1^3 &= (154)(23) \circ (154)(23) \circ (154)(23) \\
\sigma_1^3 &= (23) \quad \text{(Note 1,5,4 form 1 length cycle as are not mentioned)}
\end{aligned}$$

After taking a small example for g we observed following
1. For any power of g, elements of a disjoint cycle can only be mapped to themselves and other elements of the same disjoint cycle.
2. Each disjoint cycle is periodic with period equal to its cycle length.

The attacker ELA can easily access g, g^c, g^d and to calculate the key she need to know the value of either c or d.
Here, without loss of generality, we will solve for value of c. Same process can be followed for d

Step 1. Decompose g and g^c into disjoint cycles. We also include cycle of length 1.
Step 2. Compute array G such that for every index i from $[1, .., n]$ we store a tuple (a, b) where a is index of the disjoint cycle in which i is present and b is i's index in that disjoint cycle. Similarly we create an array Gc.
Step 3. Compute array $X[i]$ and $Y[i]$ where $X[i]$ has the first element of the each disjoint cycle in Gc and $Y[i]$ has the second element of the each disjoint cycle in Gc

Step 4. For each element in X and Y find their position in g using array G let y be the position of element of Y and x be the position of element of X the store $y - x$ in array Z and also store cycle length of disjoint cycle of g to which these elements belong in L.
Step 5. So c is the solution to each remainder $Z[j]$ modulo $L[j]$ for $1 \leq j \leq |Z|$. Solution can be efficiently computed using CRT i.e Chinese Remainder Theorem

$$\begin{aligned} c &\equiv Z[1] \pmod{L[1]} \\ &\vdots \\ c &\equiv Z[i] \pmod{L[i]} \end{aligned}$$

After obtaining the value of c, ELA can compute key $k = (g^d)^c$ efficiently.

Time Complexity

1. Complexity of step 1 is $\mathbf{O(n)}$ as there are n look ups and n stored integers
2. Complexity of step 2 is $\mathbf{O(n)}$ - requires at most 2n integers
3. Complexity of step 3 is $\mathbf{O(n)}$ - at most n look ups
4. Complexity of step 4 is $\mathbf{O(n)}$ - n computations and n look ups
5. Complexity of step 5 is $\mathbf{O(n^2 log^2(n))}$ - to solve linear system of congruences which are upper bounded by $\lceil \frac{\mathbf{n}}{\mathbf{2}} \rceil$ we use at most $(\mathbf{n-1})$ times extended euclidean algorithm which has cost of $\mathbf{O}\left(\sum_{\mathbf{k=1}}^{\mathbf{n-1}} \mathbf{k} \cdot \log^{\mathbf{2}} \mathbf{n}\right) = \mathbf{O}\left(\mathbf{n^2} \log^{\mathbf{2}} \mathbf{n}\right)$. From reference [CC21] we know that $\log |\mathbf{g}| = \mathbf{O}(\sqrt{\mathbf{n} \log \mathbf{n}})$

Overall Complexity of Algorithm is $\mathbf{O(n^2 log^2(n))}$ or $\mathbf{O}(\log^{\mathbf{4}} |\mathbf{g}|)$

References

[Agg] Prof. Manindra Aggarwal. Lecture notes on DES.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

[CC21] Jorge Martínez Carracedo and Adriana Suárez Corona. Cryptanalysis of a Group Key Establishment Protocol. *Symmetry*, 13(2):332, 2021.