CS641

Modern Cryptology
Indian Institute of Technology, Kanpur

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

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

Your solution goes here.

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

$$1 \rightarrow 5 \rightarrow 4 \rightarrow 1$$

$$2 \rightarrow 3 \rightarrow 2$$

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

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

$$\begin{aligned} \sigma_1 \circ \sigma_2 &= (154)(23) \circ (12345) \\ &= (13) \quad \text{(Note 2,4,5 form 1 length cycle as are not mentioned)} \\ \sigma_2 \circ \sigma_1 &= (12345) \circ (154)(23) \\ &= (24) \quad \text{(Note 1,3,5 form 1 length cycle as are not mentioned)} \end{aligned}$$

Therefore function composition is not commutative

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.

Complexity
1. Complexity of step 1 is $\mathcal{O}(n)$ as there are n look ups and n stored integers
2. Complexity of step 2 is $\mathcal{O}(n)$
3. Complexity of step 3 is $\mathcal{O}(n)$
4. Complexity of step 4 is $\mathcal{O}(n)$
5. Complexity of step 5 is $\mathcal{O}(n^2 log^2(n))$
6. Overall Complexity of Algorithm is $\mathcal{O}(n^2 log^2(n))$

References

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