Ministerul Educației, Culturii și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

Raport

pentru lucrare de laborator Nr. 4
la cursul "*Cifruri bloc. Algoritmul DES*"

A efectuat: Alexei Ciumac, FAF-212
A verificat: Cătălin Mîțu

Chișinău - 2023

Subject: Cryptanalysis of polyalphabetic ciphers

Tasks:

Studiați materiale didactice plasate pe ELSE. De elaborat un program în unul din limbajele de programare preferate pentru implementarea unui element al algoritmului DES. Sarcina se va alege în conformitate cu numărul n de ordine al studentului din lista grupei, în conformitate cu formula: nr_sarcina = n mod 11 (7 mod 11 = 7). Pentru fiecare sarcină să fie afișate la ecran tabelele utilizate și toți pașii intermediari. Datele de intrare să fie posibil de introdus de utilizator sau de generat în mod aleatoriu. Atenție! La susținerea lucrării vor fi puse întrebări despre lucrul întregului algoritm!!!

Theoretical notes:

The DES (Data Encryption Standard) algorithm is a widely used symmetric-key block cipher designed for securing electronic data. It operates on 64-bit blocks of plaintext, employing a 56-bit secret key for encryption and decryption. The algorithm consists of key generation, initial and final permutations, and 16 rounds of complex operations, including substitution, permutation, and XOR operations. DES provides a balance between security and efficiency, though its key length is now considered inadequate for certain applications, and more advanced encryption standards have been developed.

In the DES algorithm, the substitution operation involves the use of S-boxes, which play a crucial role in enhancing the security of the cipher. These S-boxes are tables containing specific 4-bit output values corresponding to different 6-bit input values. During each round, 48 bits of data undergo substitution through eight S-boxes. The process involves dividing the 6-bit input into two parts: the first and last bits determine the row, and the middle four bits determine the column. The S-boxes use these coordinates to locate a specific 4-bit output value, effectively replacing the original 6 bits. This non-linear substitution adds complexity and introduces a key-dependent aspect to the algorithm, contributing to the overall strength of DES against various cryptographic attacks.

Implementation:

I started the implementation from setting the tables defining the functions S1,...,S8:

```
s_boxes = {}
s_boxes["1"] = [
  [14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7],
  [0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8],
  [4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0],
  [15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13]
]
s_boxes["2"] = [
  [15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10],
  [3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5],
  [0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15],
  [13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9]
]
s_boxes["3"] = ...
s_boxes["4"] = ...
s_boxes["5"] = ...

s_boxes["6"] = [
  [12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11],
  [10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8],
  [9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6],
  [4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13]
]
s_boxes["7"] = [
  [4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1],
  [13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6],
  [1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2],
  [6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12]
]
s_boxes["8"] = [
  [13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7],
  [1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2],
  [7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8],
  [2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11]
]
```

The next step is the main functionality of the program:

```
def applySBox(input_bits):
  if len(input_bits) != 48:
    raise ValueError("Input bits must have a length of 48.")

  # Apply S-boxes to B1, B2, B3, B4, B5, B6, B7, B8
  sboxes = []
  s_box_index = 0
  for i in range(0, len(input_bits), 6):
    group = input_bits[i:i+6]
    i_index = int(str(group[0]) + str(group[5]), 2)
    j_index = int(str(group[1]) + str(group[2]) + str(group[3]) +
str(group[4]), 2)
    bin_string = format(s_boxes["1"][i_index][j_index], '04b')
    s_box_index += 1
    for bit in bin_string:
      sboxes.append(int(bit))

  return sboxes
```

In this piece of code is provided for applying S-box substitution within the DES encryption process.

The function `applySBox` takes a 48-bit input and ensures its validity.
It then proceeds to apply the S-boxes to eight groups of 6 bits each (B1 to B8).

The loop iterates over these groups, interpreting the first and last bits to determine the row index (i_index) and the middle four bits to determine the column index (j_index) for the S-box lookup.

The function retrieves the corresponding 4-bit output value from the S-box table and converts it to a binary string.

The resulting bits are appended to the list 'sboxes,' forming the output of the S-box substitution for the entire 48-bit input.

Here are the example input
and output printing function:

```
B_bits = [
  1, 0, 1, 0, 0, 0, 1, 1,
  0, 0, 1, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 1, 1, 0, 0,
  0, 0, 0, 1, 1, 0, 0, 0,
  1, 1, 0, 1, 0, 0, 0, 1,
  0, 1, 0, 0, 1, 0, 1, 0,
]



result = applySBox(B_bits)
print(result)
```

Conclusion:

In conclusion, this laboratory work has provided a practical exploration of the DES (Data Encryption Standard) algorithm, focusing on the crucial S-box substitution operation. The implementation successfully applied S-boxes to eight groups of 6 bits, following the specific rules for row and column index determination. The utilization of S-boxes within the DES algorithm introduces non-linearity and key-dependent transformations, enhancing the overall security of the encryption process. Through the code implementation and algorithmic understanding, this laboratory work has afforded valuable insights into the fundamental mechanisms of DES, a foundational symmetric-key block cipher, and its vital components.