MINISTRY OF EDUCATION, CULTURE AND RESEARCH OF REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Criptography and Security

Laboratory work 4: Block ciphers. The DES algorithm

Elaborated:
st.gr. FAF-211 Grama Alexandru
Verified:
asist.univ. Catalin Mitu

Chișinău, 2023

Content

Block ciphers. The DES algorithm

A block cipher is an encryption algorithm that encrypts a fixed size of data (block) at a time. The data or plaintext is transformed into an encrypted output of the same block size, known as ciphertext, using a specific encryption key. Block ciphers are symmetric key algorithms, meaning the same key is used for both encryption and decryption processes.

Features:

1. Block Size: DES encrypts and decrypts data in 64-bit blocks.
2. Key Size: It uses a 56-bit key for encryption and decryption (though technically it's a 64-bit key with 8 bits used for parity).
3. Round: DES utilizes 16 rounds of complex transformations to convert plaintext into ciphertext.

Working Mechanism:

1. Key Generation: Initially, a 56-bit key is generated from the given 64-bit key by discarding every 8th bit.
2. Initial Permutation: The 64-bit plaintext block is subjected to an initial permutation that rearranges the bits.
3. Rounds: The permuted block is divided into two 32-bit halves, which are then processed through 16 rounds of encryption. Each round includes expansion, substitution, permutation, and mixing with the key.
4. Key Generation: Initially, a 56-bit key is generated from the given 64-bit key by discarding every 8th bit.
5. Final Permutation: After the 16 rounds are completed, the two halves are combined and subjected to a final permutation to get the 64-bit ciphertext.
6. Decryption: The decryption process is the reverse of the encryption process, applying the round keys in the reverse order.

Implementation

Task 2.3

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$_s$*arcina* = *nmod*11.*Pentrufiecaresarcinăsăfieafișatelaecrantabeleleu*

```
import random
def apply_permutation(original, permutation):
    return ''.join(original[i-1] for i in permutation)

def left_shift(data, shifts):
    return data[shifts:] + data[:shifts]

PC1 = [
    57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18,
    10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36,
    63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22,
    14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4
]

PC2 = [
    14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10, 23, 19, 12, 4,
    26, 8, 16, 7, 27, 20, 13, 2, 41, 52, 31, 37, 47, 55, 30, 40,
    51, 45, 33, 48, 44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32
]

initial_key = bin(random.getrandbits(64))[2:].zfill(64)
print("Cheia inițială de 64 de biți: ", initial_key)

key_after_pc1 = apply_permutation(initial_key, PC1)
print("Cheia după aplicarea permutării PC-1 (56 de biți): ", key_after_pc1,"\n")

print("Împărțim cheia de 56 de biți în două jumătăți de câte 28 de biți fiecare:")
```

```
C0 = key_after_pc1[:28]
D0 = key_after_pc1[28:]
print("C0: ", C0)
print("D0: ", D0)

i = 1
rotations = 1
print(f"\nRunda {i} cu {rotations} rotații la stânga:")

Ci = left_shift(C0, rotations)
Di = left_shift(D0, rotations)
print("Ci: ", Ci)
print("Di: ", Di)

combined_key = Ci + Di
print("\nCheia combinată de 56 de biți după rotație: ", combined_key)

key_after_pc2 = apply_permutation(combined_key, PC2)
print("Cheia de rundă Ki pentru runda", i, "este:", key_after_pc2)
```

Screenshot:

Task 2.3

```
Cheia inițială de 64 de biți:  1011010011010010001010000011110111101001001010110000111001011010
Cheia după aplicarea permutării PC-1 (56 de biți):  00010011100100100011110111000111000100100100111111100101 1

Împărțim cheia de 56 de biți în două jumătăți de câte 28 de biți fiecare:
C0:  00010011100100100011110111000
D0:  11100010010010011111111001011

Runda 1 cu 1 rotații la stânga:
Ci:  00100111001001000111101100000
Di:  11000100100100111111100101111

Cheia combinată de 56 de biți după rotație:  00100111001001000111101100001100010010010011111111001011 1
Cheia de rundă Ki pentru runda 1 este: 101100100110110001010100010111110101110110100010
```

Conclusion

In the domain of cryptography, block ciphers hold a significant position. These encryption methods, which include the notable DES (Data Encryption Standard) algorithm, have been instrumental in securing data transmission and storage for decades. Block ciphers work by encrypting fixed-size blocks of plaintext into equal-sized blocks of ciphertext using specific encryption keys. These algorithms are symmetrical, meaning the same key is used for both the encryption and decryption processes. Historically, DES was a linchpin in the cryptography landscape. Adopted as a federal standard in the United States in the 1970s, DES found extensive application in various sectors, including finance and communications. The algorithm encrypts 64-bit blocks of data with a 56-bit key, though it's practically a 64-bit key with 8 bits used for parity.

The mechanism of DES includes an initial permutation, 16 rounds of complex transformations, and a final permutation. Each round comprises processes of expansion, substitution, permutation, and mixing with the round-specific key derived from the original key. The meticulous design of the S-boxes (substitution boxes) and permutations in DES was intended to protect against differential and linear cryptanalysis attacks.

However, with the advent of more powerful computing technologies, DES's vulnerabilities, primarily rooted in its relatively small key size, were exposed. By the late 1990s, it was evident that DES could be cracked in a matter of days or even hours, prompting a shift towards more secure alternatives. 3DES emerged as a stopgap measure, applying the DES algorithm thrice with two or three distinct keys, enhancing the encryption's complexity and security.

In the contemporary digital era, both DES and 3DES are considered obsolete for safeguarding sensitive information. They have paved the way for more advanced algorithms like AES (Advanced Encryption Standard), which boasts improved security attributes, operational efficiency, and resistance to an array of cryptanalytic attacks. AES's versatility and robustness have earned it widespread adoption globally, marking it as the de facto standard for a plethora of applications.

While the legacy of DES as a pioneering cryptographic algorithm is undeniable, its utility in today's world is limited. It serves as a reminder of the perpetual evolution in the field of cryptography. As computational capabilities advance, so do the methods to compromise cryptographic algorithms, necessitating continuous innovation to stay ahead of threats.