

Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 4

**Discipline**: Cryptography and Security

Elaborated: FAF-221, Cusnir Grigorii

Checked: asist. univ., Dumitru Nirca

Chișinău 2023

**Introduction:**

The Data Encryption Standard (DES) is a symmetric-key algorithm used for data encryption. It works by encrypting data in blocks using a 56-bit key and a series of transformations that include substitutions and permutations. The DES algorithm processes data through 16 rounds of encryption, each involving a series of operations including XOR, substitution using S-blocks, and permutation.

In this task, we are focusing on one element of the DES algorithm: computing the new value of Ri​ in round i, given the value of Li−1 and the result of the S-block substitution. Specifically, Riis computed by performing an XOR operation between Li−1​ and the output of the S-box substitution for the current round.

The program implementation includes the ability to input data manually or generate it randomly. It computes Ri based on the given inputs, displaying the steps involved in the process.

**Functional Description**

The program performs the following steps:

**Input Handling:**

The user is prompted to choose between two methods of data input:

* Input from the keyboard, where the user manually provides a 32-bit 𝐿𝑖−1 and a 32-bit S-block result.
* Randomly generated data, where both 𝐿𝑖−1​ and the S-block result are generated randomly as 32-bit binary strings.

**Validation:**

The program ensures that both the 𝐿𝑖−1 and S-block result are valid 32-bit binary strings. If invalid input is provided, an error message is displayed.

**Computation of 𝑅𝑖:**

Once the inputs are validated, the program computes 𝑅𝑖 by performing a bitwise XOR operation between 𝐿i−1 and the S-block result. This operation is performed bit by bit, where the XOR of two bits results in 1 if the bits are different, and 0 if they are the same.

**Output:**

The program displays the value of 𝑅𝑖 , which is the result of the XOR operation.

**Code Walkthrough**

**Input Method Selection:**

The program prompts the user to choose between entering data manually (keyboard input) or generating random binary strings for both 𝐿𝑖−1 and the S-block result. If random generation is selected, the binary strings are produced using the Random class.

**Validation of Binary Strings:**

The method **isValidBinary** checks whether a string consists solely of '0' and '1' characters and has a length of exactly 32 bits.

**Random Binary Generation:**

The **generateRandomBinary** method generates a random binary string of the specified length by appending random bits (0 or 1) to a StringBuilder.

**XOR Calculation:**

The **xorBinaryStrings** method performs the bitwise XOR operation between two binary strings of equal length. The result is stored in a StringBuilder, which is then converted to a string and returned.

**Output**:

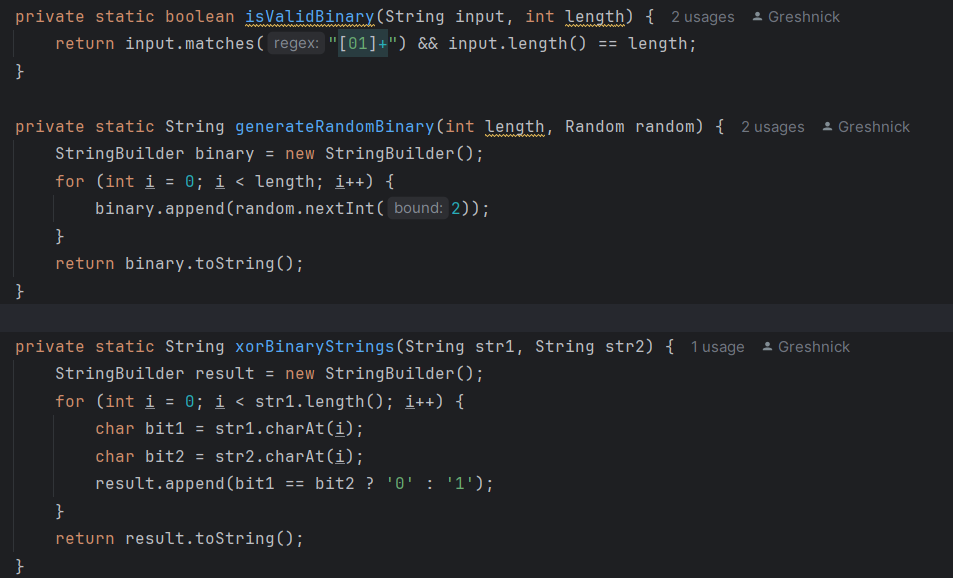
After calculating 𝑅𝑖, the program prints the result.

**Error Handling:**

If the user inputs a binary string that is not exactly 32 bits long or contains invalid characters (anything other than '0' or '1'), the program will display an error message and terminate.

The program also handles invalid input by requiring the user to choose the correct method of input (keyboard or random generation).

**Code:**

****

**Conclusion**

This implementation computes 𝑅𝑖 in the context of the DES algorithm by applying the XOR operation to two 32-bit binary strings. The program allows for both manual input and random generation of input data, making it flexible for various use cases. The algorithm correctly handles validation and ensures that the operation is performed as intended, outputting the final result in binary format.