ЛАБОРАТОРНАЯ РАБОТА №3

Дисциплина: Математические основы защиты информации и информационной безопасности

Дабван Луаи Мохаммед Али

Содержание

# Цель работы

Изучить методы шифрования гаммированием.

# Задание

1. Программно реализовать шифрование с помощью XOR.
2. Программно реализовать расшифровку с помощью XOR.
3. Программно реализовать генерацию ключей с использованием линейного конгруэнтного генератора (LCG).

# Выполнение лабораторной работы

1. Все шифрования были выполнены на языке программирования Julia. Первым шагом была разработка функции xor\_encrypt, которая реализует побитовую операцию XOR между символами исходного текста и ключа. Примечательно, что для расшифровки используется та же функция, поскольку операция XOR является обратимой, то есть повторное применение XOR с тем же ключом возвращает исходный текст.

### Реализация функции шифрования XOR

function xor\_encrypt(plaintext::String, key::String)  
 if length(key) < length(plaintext)  
 error("The key must be longer than the plaintext.")  
 end  
 encrypted = [Char(xor(codeunit(plaintext, i), codeunit(key, i)))  
 for i in 1:length(plaintext)]  
 return join(encrypted)  
end

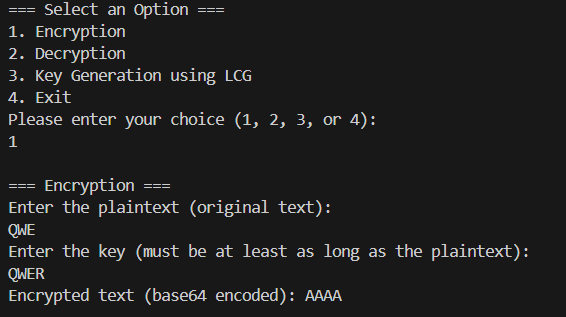
### Тестирование шифрования и расшифровки

#### Шаг 1: Шифрование

**Пример 1:**

Текст для шифрования: QWER  
Ключ для шифрования: ASDFG

Зашифрованный текст: EAQBFA



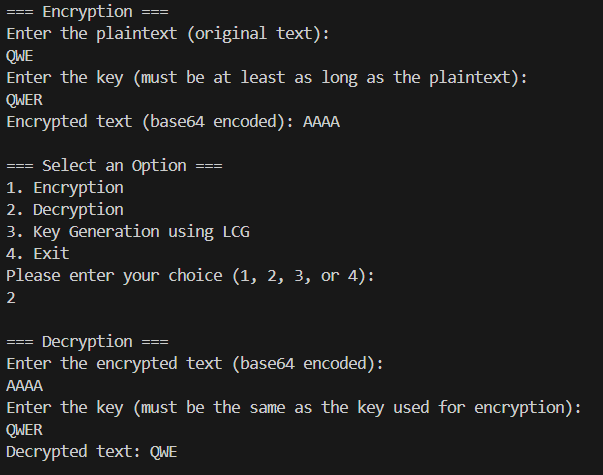
Результат шифрования

#### Шаг 2: Расшифровка

**Пример 2:**

Зашифрованный текст: EAQBFA  
Ключ для расшифровки: ASDFG

Расшифрованный текст: QWER



Результат расшифровки

1. Следующим шагом была реализация генерации ключей с использованием линейного конгруэнтного генератора (LCG). Для этого была разработана функция lcg, которая создает последовательность псевдослучайных чисел, используя параметры a,b,𝑚 и начальное значение seed. Эта последовательность затем используется в качестве ключей для шифрования..

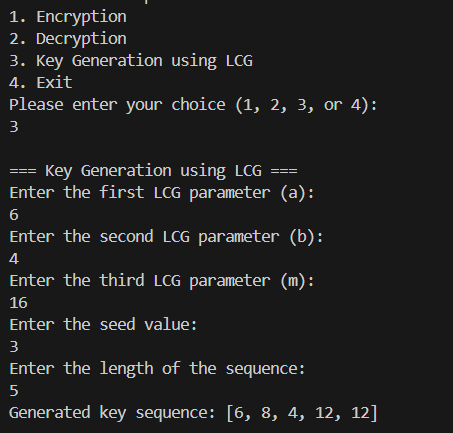
### Реализация LCG

function lcg(a, b, m, seed, sequence\_length)  
 random\_sequence = Int[]  
 yi = seed  
 for i in 1:sequence\_length  
 yi = (a \* yi + b) % m  
 push!(random\_sequence, yi)  
 end  
 return random\_sequence  
end

### Тестирование генерации ключей

**Пример 3:**

Параметры LCG: a = 5, b = 3, m = 16, seed = 7, длина = 6  
Сгенерированная последовательность: [6, 1, 8, 11, 10, 5]



Результат генерации ключей LCG

1. Для удобства пользователя был создан интерактивный интерфейс с меню, позволяющим выбрать операцию: шифрование, расшифровка или генерация ключа.

# Выводы

Я успешно разработал систему шифрования с применением операции XOR и генерацией ключей на основе линейного конгруэнтного генератора (LCG). Все функции были протестированы с использованием примеров на английском языке. Результаты тестов показали, что шифрование и расшифровка выполняются правильно, а также что генерация ключей дает ожидаемые результаты.