Лабораторная работа №8

Информационная безопасность

Николаев Д. И.

14 октября 2023

Российский университет дружбы народов, Москва, Россия

Цели

Цели

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Задачи

Задачи

1. Реализовать режим однократного гаммирования;
2. Найти текст второго сообщения по известным шифротекстам и первому сообщению.

Выполнение работы

Вместо

$$C_1 \bigoplus C_2 \bigoplus P_1 = P_1 \bigoplus P_2 \bigoplus P_1 = P_2. \tag{1}$$

имеем следующие выражения

$$C_1 + C_2 \equiv P_1 + K + P_2 + K \equiv P_1 + P_2 + 2K \pmod{N}, \tag{2}$$

$$C_i \equiv P_i + K \pmod{N}, \quad i = 1, 2 \tag{3}$$

$$C_1 + C_2 \equiv P_1 + (C_2 - K) + 2K \equiv P_1 + C_2 + K \pmod{N}, \tag{4}$$

$$P_2 \equiv C_2 - K \equiv C_2 - (C_1 + C_2 - P_1 - C_2) \equiv C_2 - (C_1 - P_1) \pmod{N}. \tag{5}$$

В итоге имеем выражение (6)

$$P_2 \equiv C_2 - C_1 + P_1 \pmod{N}. \tag{6}$$

Получение текста второго сообщения по шифротекстам и первому сообщению 1

```
include("C:/Users/User/Documents/work/study/2022-2023/
Информационная безопасность/infosec/labs/lab07/report/report/gamma.jl")
function Gamma_Hijack_Message(Source_Message_1::String,
Encrypted_Message_1::String, Encrypted_Message_2::String)::String
    n1 = length(Source_Message_1)  # Длина исходного сообщения 1
    n2 = length(Encrypted_Message_1)
    n3 = length(Encrypted_Message_2)
    println("Первое исходное сообщение - ", Source_Message_1)
    println("Первое зашифрованное сообщение - ", Encrypted_Message_1)
    println("Второе зашифрованное сообщение - ", Encrypted_Message_2)
    n1 != n2 != n3  ? println("Несоответсвие размерности исходного и зашифров
    Source_Code_1 = []
    Encrypted_Code_1 = []
    Encrypted_Code_2 = []
```

Получение текста второго сообщения по шифротекстам и первому сообщению 2

```
for i in Source_Message_1
    push!(Source_Code_1, Dictionary[i])
end
for i in Encrypted_Message_1
    push!(Encrypted_Code_1, Dictionary[i])
end
for i in Encrypted_Message_2
    push!(Encrypted_Code_2, Dictionary[i])
end
println("Код первого исходного сообщения - ", Source_Code_1)
println("Код первого зашифрованного сообщения - ", Encrypted_Code_1)
println("Код второго зашифрованного сообщения - ", Encrypted_Code_2)
Source_Code_2 = []   # Код второго исходного сообщения
```

```
    for i in range(1, n1)
        a = Encrypted_Code_2[i] - Encrypted_Code_1[i] + Source_Code_1[i]
        a <= 0 ? a += N : skip
        a > N ? a %= N : skip
        push!(Source_Code_2, a)
    end
    println("Код второго исходного сообщения - ", Source_Code_2)
    Source_Message_2 = ""
    for i in Source_Code_2
        Source_Message_2 *= Dictionary2[i]
    end
    println("Второе исходное сообщение - ", Source_Message_2)
    return Source_Message_2
end
```

```
P1 = "На Ваш исходящий от 1204"  # 24 символа
P2 = " в Северный филиал Банка"
Initial_Key = "АБВГДЕжзийклмнопрстуфхЦЧ"
println("Находим первое зашифрованное сообщение")
C1 = Gamma_Find_Encrypted_Text(P1, Initial_Key) # тексты зашифрованных сообще
println("Находим второе зашифрованное сообщение")
C2 = Gamma_Find_Encrypted_Text(P2, Initial_Key)
println("Находим второе сообщение по известным шифротекстам и первому сообщен
Hijacked_P2 = Gamma_Hijack_Message(P1, C1, C2)
if P2 == Hijacked_P2
    println("Взлом второго сообщения прошел успешно!")
else
    println("Неудача")
end
```

Результат

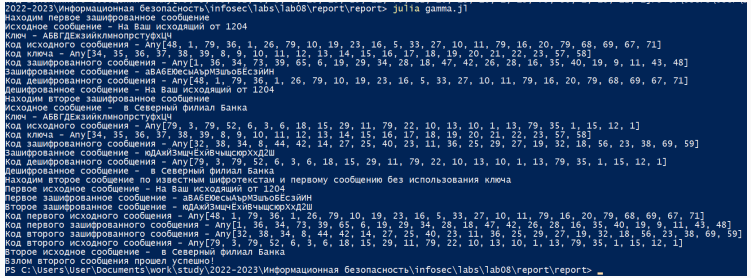

Рис. 1: Реализация взлома однократного гаммирования

Результаты

Результаты

По результатам работы, я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.