Отчет по лабораторной работе №3

Шифрование гаммированием

Арам Грачьяевич Саргсян

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить метод шифрования гаммированием

2 Задание

Реализовать алгоритм шифрования конечной гаммой.

3 Теоретическое введение

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком $\oplus$) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами:

$$0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0.$$

Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой[1].

4 Выполнение лабораторной работы

1. Я реализовал необходимый программный комплекс.

```
alphabet = 'а':'я'
function Text_to_Numbers(Text::String, Alphabet::StepRange{Char, Int64} = alphabet)::V
    numbers = []
    for char in lowercase(Text)
        push!(numbers, findfirst(c -> c == char, Alphabet))
    end
    return numbers
end

function Numbers_to_Text(Numbers::Vector{Any}, Alphabet::StepRange{Char, Int64} = alph
    text = ""
    for number in Numbers
        text *= alphabet[number]
    end
    return lowercase(text)
end

function Cipher_Gamma(Message::String, Gamma::String, Alphabet::StepRange{Char, Int64}
    Message_Numbers = Text_to_Numbers(Message, Alphabet)
    Gamma_Numbers = Text_to_Numbers(Gamma, Alphabet)
    length_alphabet = length(Alphabet)
```

```
    Encrypted_Numbers = []
    for i in 1:length(Message_Numbers)
        encrypted_number = (Message_Numbers[i] + Gamma_Numbers[(i-1) % length(Gamma_Nu
        push!(Encrypted_Numbers, encrypted_number == 0 ? length_alphabet : encrypted_n
    end
    return Numbers_to_Text(Encrypted_Numbers, Alphabet)
end

function Decipher_Gamma(Encrypted_Message::String, Gamma::String, Alphabet::StepRange{
    Encrypted_Numbers = Text_to_Numbers(Encrypted_Message, Alphabet)
    Gamma_Numbers = Text_to_Numbers(Gamma, Alphabet)
    length_alphabet = length(Alphabet)
    Message_Numbers = []
    for i in 1:length(Encrypted_Numbers)
        message_number = (Encrypted_Numbers[i] - Gamma_Numbers[(i-1) % length(Gamma_Nu
        push!(Message_Numbers, message_number == 0 ? length_alphabet : message_number)
    end
    return Numbers_to_Text(Message_Numbers, Alphabet)
end

message = "приказ"
gamma = "гамма"
println("Исходное сообщение: ", message, ";\nКонечная гамма шифрования: ", gamma)

ciphertext = Cipher_Gamma(message, gamma)
println("Результат шифрования: ", ciphertext)

decrypted_message = Decipher_Gamma(ciphertext, gamma)
println("Результат дешифрования: ", decrypted_message)
```

2. Получил результаты, аналогичные примеру (рис. 4.1).

```
message = "приказ"
gamma = "гамма"
println("Исходное сообщение: ", message, ";\nКонечная гамма шифрования: ", gamma)

ciphertext = Cipher_Gamma(message, gamma)
println("Результат шифрования: ", ciphertext)

decrypted_message = Decipher_Gamma(ciphertext, gamma)
println("Результат дешифрования: ", decrypted_message)
```

```
Исходное сообщение: приказ;
Конечная гамма шифрования: гамма
Результат шифрования: усхчбл
Результат дешифрования: приказ
```

Рис. 4.1: Результат работы алгоритма

5 Выводы

Я реализовал алгоритм шифрование конечной гаммой.

Список литературы

1. Abdullaev T.R., Juraev G.U. Application three-valued logic in symmetric block encryption algorithms // Journal of Physics: Conference Series. IOP Publishing, 2021. Т. 2131, № 2. С. 022082.