Отчет по лабораторной работе №3

Шифрование гаммированием

Арам Грачьяевич Саргсян

Содержание

# 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 ) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами:

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)::Vector{Any}  
 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} = alphabet)::String  
 text = ""  
 for number in Numbers  
 text \*= alphabet[number]  
 end  
 return lowercase(text)  
end  
  
function Cipher\_Gamma(Message::String, Gamma::String, Alphabet::StepRange{Char, Int64} = alphabet)::String  
 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\_Numbers) + 1]) % length\_alphabet  
 push!(Encrypted\_Numbers, encrypted\_number == 0 ? length\_alphabet : encrypted\_number)  
 end  
 return Numbers\_to\_Text(Encrypted\_Numbers, Alphabet)  
end  
  
function Decipher\_Gamma(Encrypted\_Message::String, Gamma::String, Alphabet::StepRange{Char, Int64} = alphabet)::String  
 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\_Numbers) + 1]) % length\_alphabet  
 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)

1. Получил результаты, аналогичные примеру (рис. 1).

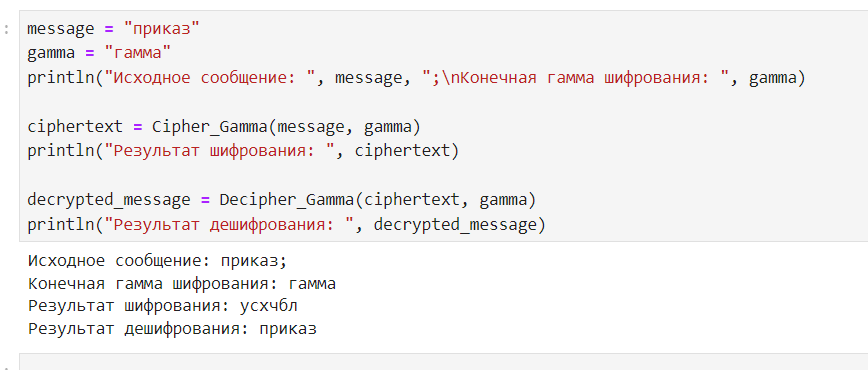


Рис. 1: Результат работы алгоритма

# 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](https://doi.org/10.1088/1742-6596/2131/2/022082) // Journal of Physics: Conference Series. IOP Publishing, 2021. Т. 2131, № 2. С. 022082.