Лабораторная работа №8

Информационная безопасность

Николаев Дмитрий Иванович

Содержание

# 1 Цель работы

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

# 2 Теоретическое введение

Шифротексты двух телеграмм (исходные сообщения) можно получить по формулам режима однократного гаммирования: где — шифротексты, — открытые (исходные) тексты, , — единый ключ шифрования.

Открытый текст можно найти в соответствии с (), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства () складываются по модулю 2. Тогда с учётом свойства операции XOR получаем:

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированного формата, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар (известен вид обеих шифровок). Тогда зная и учитывая (), имеем:

Таким образом, злоумышленник получает возможность определить те символы сообщения , которые находятся на позициях известного шаблона сообщения . В соответствии с логикой сообщения , злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения . Затем вновь используется () с подстановкой вместо полученных на предыдущем шаге новых символов сообщения . И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска [1].

# 3 Выполнение лабораторной работы

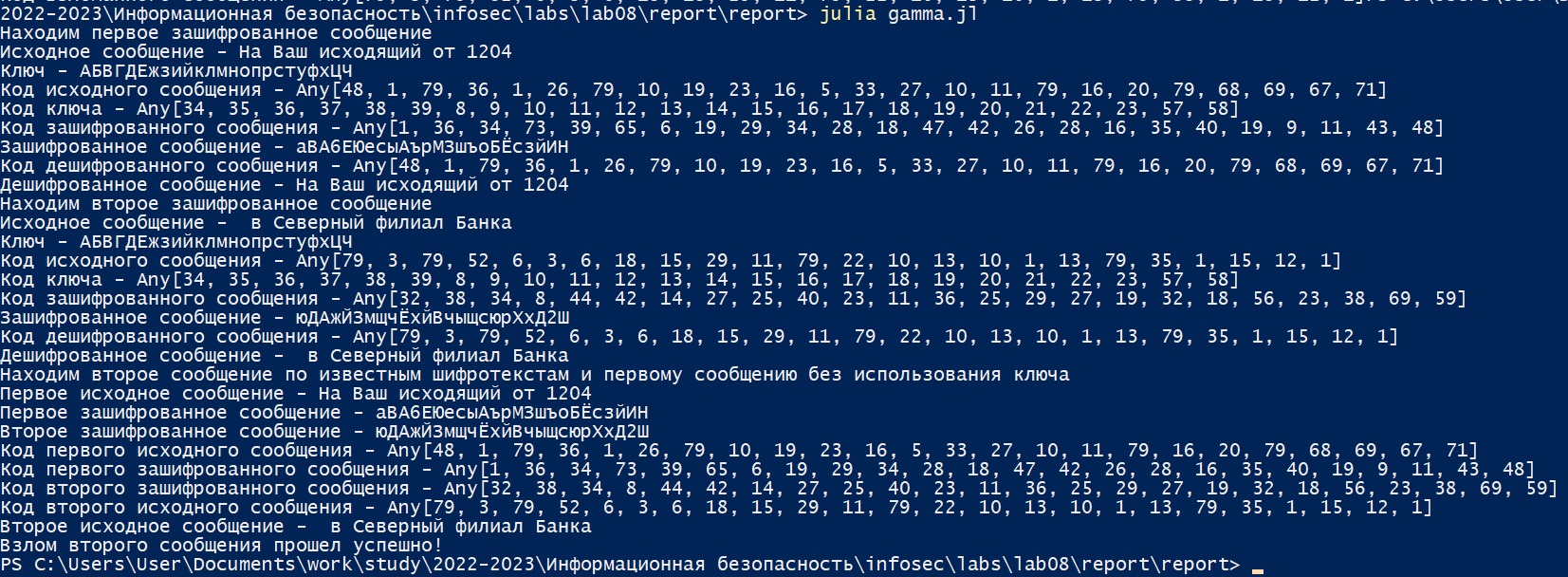
Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты и в режиме однократного гаммирования. Приложение должно определить вид шифротекстов и обоих текстов и при известном ключе. Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Имеем две телеграммы: = “На Ваш исходящий от 1204” и = ” в Северный филиал Банка”. Используя функции из предыдущей лабораторной и выбрав произвольный ключ, найдём значения обоих шифротекстов. После этого реализуем функцию, которая по обоим известным шифротекстам и одному из сообщений находит второе сообщение. Результат работы программы представлен на ([??]).

Так как в программе реализован собственный словарь (длины 81), то рассматривается не операция исключающего ИЛИ, а модульная арифметика по основанию длины словаря (). Так, для реализации описанной выше функции, вместо имеем следующие выражения В итоге имеем выражение () где — длина словаря, — код ключа, — код первого исходного сообщения, — код второго исходного сообщения, — код первого зашифрованного сообщения, — код второго зашифрованного сообщения; остаток 0 означает последний элемент словаря.

Ниже представлен код реализации на Julia:

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   
 зашифрованных сообщений") : skip  
 Source\_Code\_1 = []  
 Encrypted\_Code\_1 = []  
 Encrypted\_Code\_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 сообщений  
println("Находим второе зашифрованное сообщение")  
C2 = Gamma\_Find\_Encrypted\_Text(P2, Initial\_Key)  
  
  
println("Находим второе сообщение по известным шифротекстам и первому   
сообщению без использования ключа")  
Hijacked\_P2 = Gamma\_Hijack\_Message(P1, C1, C2)  
  
if P2 == Hijacked\_P2  
 println("Взлом второго сообщения прошел успешно!")  
else  
 println("Неудача")  
end



Реализация взлома однократного гаммирования

# 4 Ответы на вопросы

1. Как, зная один из текстов ( или ), определить другой, не зная при этом ключа?

* С помощью формулы
* где и — шифротексты двух исходных текстов.

1. Что будет при повторном использовании ключа при шифровании текста?

* Мы получим исходное сообщение.

1. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

* С помощью формул
* где — шифротексты, — открытые (исходные) тексты, , — единый ключ шифрования.

1. Перечислите недостатки шифрования одним ключом двух открытых текстов.
   1. Имея на руках одно из сообщений в открытом виде и оба шифротекста, злоумышленник способен расшифровать каждое сообщение, не зная ключа.
   2. Зная шаблон сообщений, злоумышленник получает возможность определить те символы сообщения , которые находятся на позициях известного шаблона сообщения , то есть сильно сокращает возможные варианты для перебора.
   3. Зная ключ, злоумышленник сможет расшифровать все сообщения, которые были закодированы при его помощи.
2. Перечислите преимущества шифрования одним ключом двух открытых текстов.
   1. Данный способ помогает упростить процесс шифрования и дешифровки.
   2. При отправке сообщений между двумя компьютерами, удобнее пользоваться одним общим ключом для передаваемых данных.

# 5 Выводы

В ходе выполнения лабораторной работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

# Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №8 [Электронный ресурс]. RUDN, 2023. URL: <https://esystem.rudn.ru/pluginfile.php/2090214/mod_resource/content/2/008-lab_crypto-key.pdf>.