Лабораторная работа № 7

Элементы криптографии. Однократное гаммирование

Абакумов Егор Александрович

Содержание

# Цель работы

Освоить на практике применение режима однократного гаммирования.

# Теоретическое описание

Простейшей и в то же время наиболее надёжной из всех схем шифрования является так называемая схема однократного использования, изобретение, которое чаще всего связывают с именем Г.С. Вернама.

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. С точки зрения теории криптоанализа, метод шифрования случайной однократной равновероятной гаммой той же длины, что и открытый текст, является невскрываемым. Кроме того, даже раскрыв часть сообщения, дешифровщик не сможет хоть сколько-нибудь поправить положение – информация о вскрытом участке гаммы не даёт информации об остальных её частях [1].

“Наложение” гаммы – не что иное, как выполнение операции сложения по модулю 2 (xor) её элементов с элементами открытого текста. Эта операция в в математике обозначается знаком .

Гаммирование является симметричным алгоритмом. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и дешифрование выполняется одной и той же программой.

К. Шенноном было доказано, что если ключ является фрагментом истинно случайной двоичной последовательности с равномерным законом распределением, причём его длина равна длине исходного сообщения, и используется этот ключ только один раз, после чего уничтожается, то такой шифр является абсолютно стойким, даже если криптоаналитик располагает неограниченным ресурсом времени и неограниченным набором вычислительных ресурсов. Действительно, противнику известно только зашифрованное сообщение , при этом все различные ключевые последовательности возможны и равновероятны, а значит, возможны и любые сообщения , т.е. криптоалгоритм не даёт никакой информации об открытом тексте [2].

# Ход работы

1. В первую очередь оговоримся, что использовать будем среду Jupyter Notebook и язык программирования Питон. Для выполнения задания нам необходимо будет подключить библиотеки random и string. Пишем блок необходимых функций, которые и реализуют всю логику программы: функция generate\_new\_key принимает на вход длину требуемого ключа и возвращает случайную строку символов, что и будет являться ключом; функция hexadecimal\_form возвращает шестнадцатиричный вид подаваемой на вход строки; функции single\_gamming, unencrypt и compute\_initial\_key принимают на вход две строки и выполняют их посимвольное сложение по модулю 2 (иллюстр. 1).

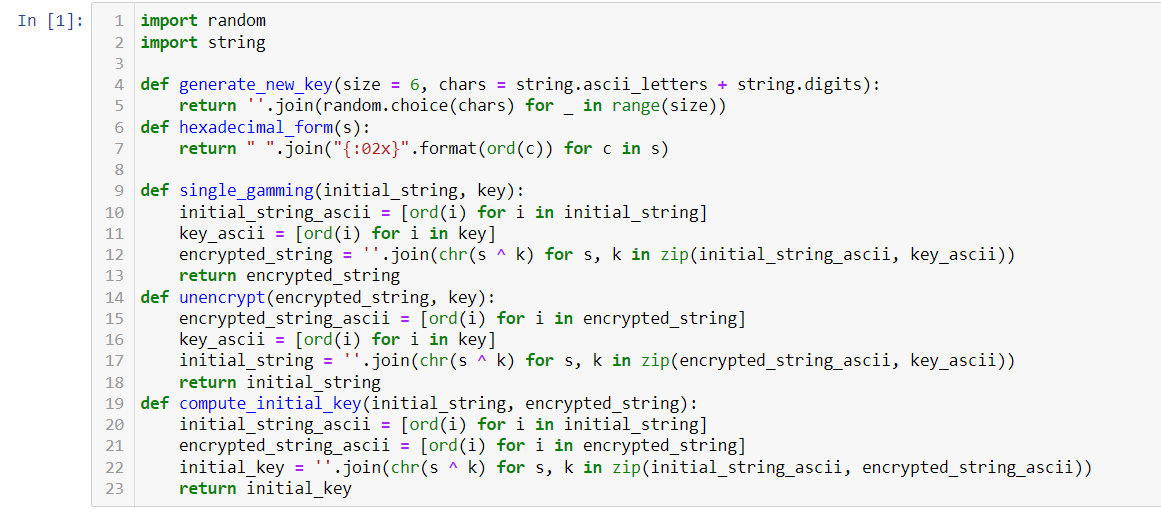


Figure 1: Блок функций для расчетов

1. Далее пишем блок расчетов всех необходимых параметров: initial\_string получает с клавиатуры входную строку (“С Новым Годом, друзья!” в нашем случае); key - начальный используемый ключ; encrypted\_string - первоначальный шифротекст; new\_key - случайный ключ, используемый для получения варианта шифротекста; unencrypted\_new\_key - вариант текста, получаемый с помощью случайного ключа и изначального шифротекста; initial\_key - начальный ключ, получаемый гаммированием открытого начального текста имеющимся шифротекстом; unencrypted\_initial\_key - расшифрованный вычисленным исходным ключом шифротекст (иллюстр. 2).

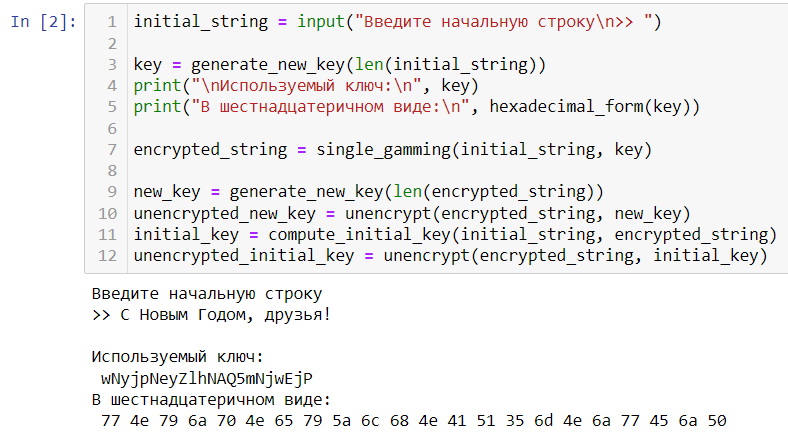


Figure 2: Блок расчетов переменных

1. Пишем блок вывода данных для первого задания. Выводим на экран шифротекст, полученный однократным гаммированием исходной строки сгенерированным ключом (иллюстр. 3)

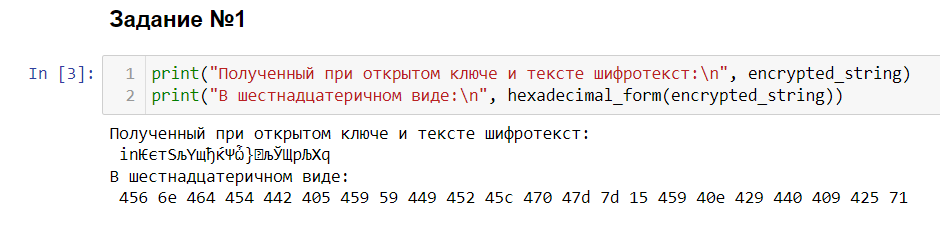


Figure 3: Задание №1

1. Пишем блок вывода данных для второго задания. Выводим на экран случайный сгенерированный ключ, позволяющий вычислить один из вариантов расфишровки шифротекста, непосредственно сам вариант расшифровки, далее из шифротекста и исходной строки вычисляем и выводим исходный ключ и с его помощью выводим расшифрованный шифротекст (иллюстр. 4)

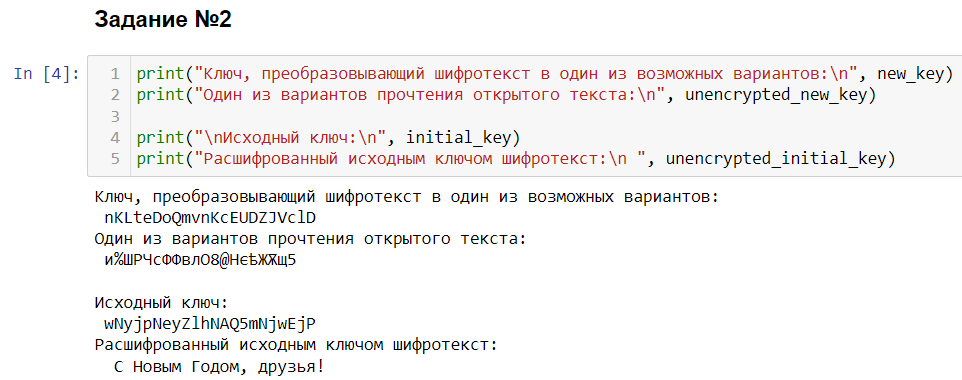


Figure 4: Задание №2

# Выводы

В ходе работы мы успешно на практике освоили применение режима однократного гаммирования.

# Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования.

* Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Однократное гаммирование – это когда каждый символ попарно с символом ключа складываются по модулю 2 (XOR) (обозначается знаком ).

1. Перечислите недостатки однократного гаммирования.

* Ключ должен быть того же размера, что и сообщение, нельзя кодировать два сообщения одним ключом

1. Перечислите преимущества однократного гаммирования.

* Невскрываемость при соблюдении требований и простота реализации.

1. Почему длина открытого текста должна совпадать с длиной ключа?

* Потому что каждый символ сообщения должен быть закодирован попарным сложением по модулю два.

1. Какая операция используется в режиме однократного гаммирования, назовите её особенности?

* XOR или “исключающее ИЛИ”. Сложение по модулю 2. Особенность в симметричности - операция при повторном применении дает исходный результат.

1. Как по открытому тексту и ключу получить шифротекст?

* Сложить по модулю 2 каждый символ открытого текста и ключа.

1. Как по открытому тексту и шифротексту получить ключ?

* Сложить по модулю 2 каждый символ открытого текста и шифротекста.

1. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?
2. случайность ключа;
3. совпадение длины ключа и сообщения;
4. однократное использование ключа.

# Список литературы

1. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
2. Харин, Ю.С. Математические и компьютерные основы криптологии : учебное пособие / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич. – Мн. : Новое знание, 2003. – 382 с.
3. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. M.: РУДН, 2015. 64 с..