Отчет по лабораторной работе №8

Дисциплина: Информационная безопасность

Выполнила Дяченко Злата Константиновна, НФИбд-03-18

Содержание

# Цель работы

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

# Задание

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 P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

# Теоретические вводные данные

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом.

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар C1 XOR C2 (известен вид обеих шифровок). Тогда зная P1, имеем:

Таким образом, злоумышленник получает возможность определить те символы сообщения P2, которые находятся на позициях известного шаблона сообщения P1. В соответствии с логикой сообщения P2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P2. Затем вновь используется формула с подстановкой вместо P1 полученных на предыдущем шаге новых символов сообщения P2. И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

# Выполнение лабораторной работы

## Шаг 1

Писать программу решила на языке C++. Были написаны функции, показанные на Рисунке 1 (рис. 1). Функция convert преобразует символы открытого текста в их коды в ASCCI. Функция printM выводит массив данных (в нашем случае коды символов открытого текста, зашифрованный текст, ключ). Функция gammir осуществляет однократное гаммирование. Функция poisk находит открытый текст P2, если известны шифротексты C1, C2 и открытый текст P1.

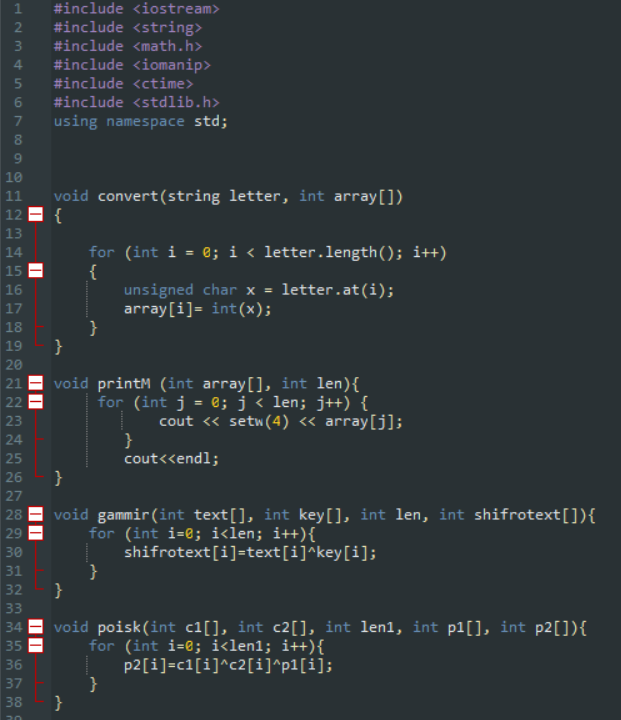


Figure 1: Функции

## Шаг 2

В функции main создаем переменные p1 и p2, содержащие открытый текст (рис. 2). Применяя описанную ранее функцию convert, переводим символы в их коды. С помощью массива K задаем ключ K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54. С помощью однократного гаммирования шифруем сообщения.

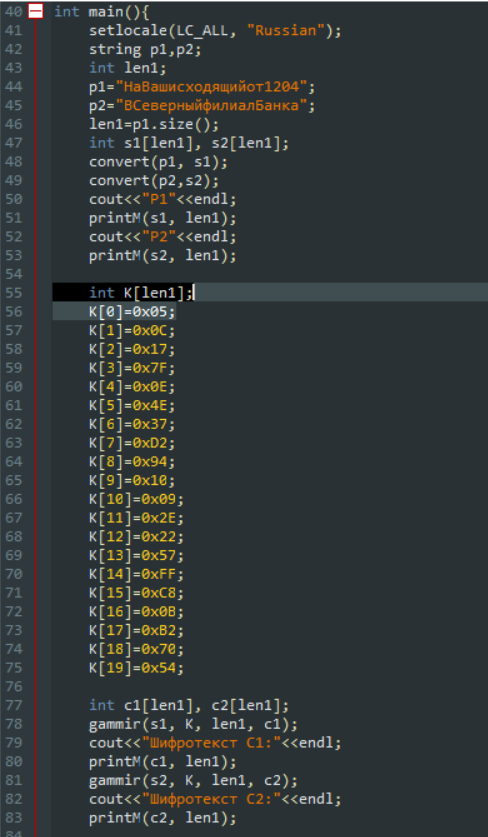


Figure 2: Осуществление однократного гаммирования

## Шаг 3

С помощью функции gammir осуществляем также дешифрование сообщений (рис. 3).

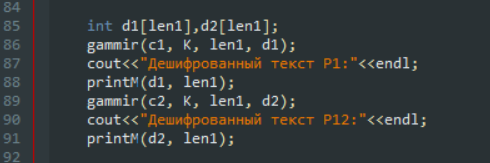


Figure 3: Дешифровка

## Шаг 3

Результаты выполнения функций представлены на Рисунке 4 (рис. 4).

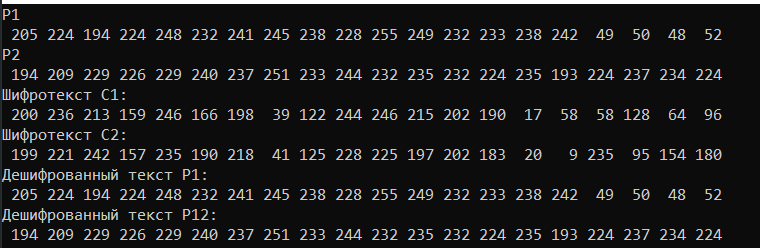


Figure 4: Результаты шифровки и дешифровки

## Шаг 4

Для того, чтобы прочитать один из текстов, зная другой, применим функцию poisk (рис. 5).

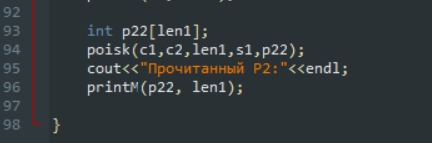


Figure 5: Чтение текста P2

## Шаг 4

Мы получили верный текст P2, что видно на Рисунке 6 (рис. 6).

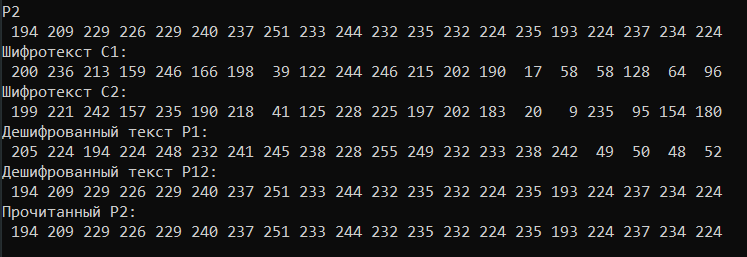


Figure 6: Прочитанный P2

# Выводы

В результате работы я освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом. Результаты работы находятся в [репозитории на GitHub](https://github.com/ZlataDyachenko/workD), а также есть [скринкаст выполнения лабораторной работы](https://www.youtube.com/watch?v=1qykNYRhPjY).