Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Уральский федеральный университет имени первого Президента России Б.Н.Ельцина»

**Кафедра технической физики**

**ОТЧЕТ**

**по лабораторной работе №1**

АЛГОРИТМ КРИПТОПРЕОБРАЗОВАНИЯ – ГОСТ 28147-89

|  |  |
| --- | --- |
| Студент: | Борисенко Д.С. |
| Группа: | Фт-430203 |
|  |  |
| Преподаватель: | Александров О. Е. |

Екатеринбург

2016

## ВВЕДЕНИЕ

Для защиты данных от нежелательного доступа используют различные криптографические методы. Они опираются на свойства самой информации и не используют свойства ее материальных носителей, особенности узлов ее обработки, передачи и хранения. Можно сказать, что криптографические методы строят барьер между защищаемой информацией и реальным или потенциальным злоумышленником из самой информации. Для разъяснения принципов построения шифров предлагаются данные методические материалы к лабораторной работе по курсу «Теория информационных систем».

Под криптографической защитой подразумевается шифрование данных. Раньше, когда эта операция выполнялось человеком вручную или с использованием различных приспособлений, и при посольствах содержались многолюдные отделы шифровальщиков, развитие криптографии сдерживалось проблемой применения шифров. Появление цифровых электронно-вычислительных машин изменило все коренным образом и в этой сфере. С одной стороны, взломщики шифров получили в свои руки чрезвычайно мощное орудие, с другой стороны, барьер сложности реализации исчез, и для создателей шифров открылись практически безграничные перспективы. Все это определило стремительный прогресс криптографии в последние десятилетия.

Российская Федерация имеет свой стандарт шифрования. Этот стандарт закреплен ГОСТом №28147-89, принятом, как явствует из его обозначения, еще в 1989 году в СССР.

## ЦЕЛЬ РАБОТЫ

При выполнение данной работы необходимо ознакомиться с теорией алгоритма криптопреобразования ГОСТ 28147-89, получить практические навыки, решив следующие задачи:

* Научиться компилировать и запускать программы GOST и GOST89.
* Также необходимо дать ответы на вопросы:

1. Что такое блок данных?
2. Что такое таблица замен и для чего она нужна?
3. Что такое ключ и для чего он нужен?
4. В каком случае (чему должны быть равны ключ и таблица замен) шифрованные данные будут тождественны исходным?
5. Что такое гаммирование?
6. Что такое имитовставка? И для чего она нужна?
7. Чем гаммирование с обратной связью отличается от простого гаммирования?
8. Какой алгоритм простое шифрование, гаммирование или гаммирование с обратной связью быстрее и почему?
9. Какой алгоритм простое шифрование, гаммирование или гаммирование с обратной связью не может выполняться параллельно, т.е. на нескольких процессорах? Почему?
10. Почему ГОСТ рекомендует использование гаммирования для шифрования, чем гаммирование лучше простого шифрования?

**ХОД РАБОТЫ**

**Ответы на вопросы:**

**1.** Блок данных – единица обработки информации по ГОСТу. В блоке данных определены алгоритмы для криптографических преобразований: шифрования и дешифрования. Алгоритмы шифромания и дешифрования – называются базовыми циклами.

**2.** Таблица замен – это матрица 8×16, она содержит 4-битовые элементы: их можно представить в виде целых чисел от 0 до 15. Также таблица замен является одной их структур данных, составляющих ключевую информацию.

Узлы замен – это строки таблицы замен. Они должны содержать значения от 0 до 15 в произвольном порядке.

Таблица замен является долговременным ключевым элементом, то есть действует в течение гораздо более длительного срока, чем отдельный ключ. Предполагается, что она является общей для всех узлов шифрования в рамках одной системы криптографической защиты. Даже при нарушении конфиденциальности таблицы замен стойкость шифра остается чрезвычайно высокой и не снижается ниже допустимого предела.

Таблица замен используется для зашифровывания, а затем расшифровывания блока данных: при помощи таблицы замен осуществляется поблочная замена 32-битного значения полученного в результате сложения младшей половины преобразуемого блока с используемым элементом ключа.

**3.** Ключ – конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор одного преобразования из совокупности возможных для данного алгоритма преобразований. Ключ является массивом из восьми 32-битных элементов.

**4.** Шифрованные данные будут тождественны исходным, если таблица замен будет равна единице в каждом ее элементе, при симметричном ключе

**5.** В режиме «гаммирование» происходит наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть

последовательности элементов данных, вырабатываемых с помощью

некоторого криптографического алгоритма, для получения зашифрованных

(открытых) данных. В режиме «гаммирования» гамма получается следующим

образом: с помощью некоторого алгоритмического рекуррентного генератора

последовательности чисел вырабатываются 64-битовые блоки данных,

которые далее подвергаются преобразованию по циклу 32-З, то есть

шифрованию в режиме простой замены, в результате получаются блоки

гаммы. Благодаря тому, что наложение и снятие гаммы осуществляется при

помощи одной и той же операции побитового исключающего или, алгоритмы

шифрования и дешифрования в режиме «гаммирования» идентичны.

Шифрование в режиме «Простая замена» заключается в применении цикла 32

З к блокам открытых данных, дешифрование – цикла 32-Р к блокам

зашифрованных данных. Это наиболее простой из режимов, 64-битовые блоки

данных обрабатываются в нем независимо друг от друга.

**6.** Имитовставка – это контрольная комбинация, зависящая от открытых данных и секретной ключевой информации. Целью использования имитовставки является обнаружение всех случайных или преднамеренных изменений в массиве информации.

**7.** Чем гаммирование с обратной связью отличается от простого гаммирования?

Гаммирование с обратной связью отличается от простого способом выработки элементов гаммы – очередной элемент гаммы вырабатывается как результат преобразования по циклу 32-З предыдущего блока зашифрованных данных, а для зашифровки первого блока массива данных элемент гаммы вырабатывается как результат преобразования по тому же циклу синхропосылки. Этим достигается зацепление блоков – каждый блок шифртекста в этом режиме зависит от соответствующего и всех предыдущих блоков открытого текста.

**8.** Какой алгоритм простое шифрование, гаммирование или гаммирование с обратной связью быстрее и почему?

Самым быстрым алгоритмом шифрования является простая замена, так как он прост в реализации.

Режим простой замены принимает на вход данные, размер которых кратен 64-м битам. Результатом шифрования является входной текст, преобразованный блоками по 64 бита в случае зашифрования циклом «32-З», а в случае расшифрования — циклом «32-Р».

Режим гаммирования принимает на вход данные любого размера, а также дополнительный 64-битовый параметр — синхропосылку. В ходе работы синхропосылка преобразуется в цикле «32-З», результат делится на две части. Полученное объединением обеих преобразованных частей значение, называемое гаммой шифра, поступает в цикл «32-З», его результат порязрядно складывается по модулю 2 с 64-разрядным блоком входных данных. Если последний меньше 64-х разрядов, то лишние разряды полученного значения отбрасываются. Полученное значение подаётся на выход. Если ещё имеются входящие данные, то действие повторяется: составленный из 32-разрядных частей блок преобразуется по частям и так далее. Таким образом, видно, что гаммирование по своей реализации сложнее просто замены, и ему необходимо больше времени на выполнение.

Режим гаммирования с обратной связью также принимает на вход данные любого размера и синхропосылку. Блок входных данных поразрядно складывается по модулю 2 с результатом преобразования в цикле «32-З» синхропосылки. Полученное значение подаётся на выход. Значение синхропосылки заменяется в случае зашифрования выходным блоком, а в случае расшифрования — входным, то есть зашифрованным. Если последний блок входящих данных меньше 64 разрядов, то лишние разряды гаммы (выхода цикла «32-З») отбрасываются. Если ещё имеются входящие данные, то действие повторяется: из результата зашифрования заменённого значения образуется гамма шифра и т.д. Данный режим также требует больших временных затрат.

**9.** Какой алгоритм простое шифрование, гаммирование или гаммирование с обратной связью не может выполняться параллельно, т.е. на нескольких процессорах? Почему?

При дешифровании блока данных в режиме «гаммирования с обратной связью», блок открытых данных зависит от соответствующего и предыдущего блоков зашифрованных данных. Поэтому, если внести искажения в зашифрованный блок, то после дешифрования искаженными окажутся два блока открытых данных – соответствующий и следующий за ним, причем искажения в первом случае будут носить тот же характер, что и в режиме «гаммирования», а во втором случае – как в режиме простой замены. Другими словами, в соответствующем блоке открытых данных искаженными окажутся те же самые биты, что и в блоке шифрованных данных, а в следующем блоке открытых данных все биты независимо друг от друга с вероятностью 1/2 изменят свои значения.

В режиме простой замены отсутствуют зависимости между операциями шифрования блоков открытого текста, поэтому распараллеливание возможно.

В режиме гаммирования гамма накладывается независимо на различные блоки открытого текста. Сам процесс выработки гаммы представляет собой суммирование по модулю 232 и 2 32 -1 содержимого соответствующих полублоков с константами. Данная рекурсия легко преобразуется к параллельному виду. Таким образом, режим гаммирования пригоден для реализации на параллельном вычислителе.

В режиме гаммирования с обратной связью гамма формируется на основе предыдущего блока зашифрованных данных. Этот режим не распараллеливается.

**10.** Почему ГОСТ рекомендует использование гаммирования для шифрования, чем гаммирование лучше простого шифрования?

Проблемы простого шифрования: в простом шифровании при шифровании 2 одинаковых блоков текста мы получим одинаковые шифр-тексты. Из-за этого криптоаналитик придет к выводу об идентичности блоков, что является недопустимым для серьезного шифра. Также, если длина массива не кратна 64 битам, появится проблема с дополнением последнего неполного блока памяти (увеличение числа шифртекста нежелательна, а заполнение пустых блоков нулями или фиксированными 0-1, приведет к возможности перебором определить содержимое, что делает алгоритм некриптостойким). Т.о., очень сильно ограничивается возможность применения шифра. Потому ГОСТ рекомендует использование такого метода для шифрования ключевых данных. Эти проблемы решаются при шифровании гаммированием. Потому оно и рекомендуется ГОСТом.

**ВТОРАЯ ЧАСТЬ**

Предложить и реализовать алгоритм прямого перебора ключей для взлома

ГОСТ 28147-89.

**АЛГОРИТМ НАХОЖДЕНИЯ НУЖНОГО КЛЮЧА**

1. Имея перехваченное сообщение, мы генерируем все возможные ключи (32 бита) которые, возможно, использовались для шифровки каждого блока (64 бита) сообщения.
2. Генерируем все возможные варианты одного блока данных.
3. Используем каждый ключ, сгенерированный нами для кодирования, для кодирования каждого варианта блока данных, так же сгенерированных нами ранее.
4. Сравниваем каждый полученный зашифрованный блок с тем блоком, который мы имеем после перехвата (образец)
5. Если зашифрованные блоки совпадают, то ключ сохраняем, записывая в массив, например.
6. Получив более узкий круг поиска истинного ключа (т.к. множество ключей могут совпасть, при кодировке одного блока) мы проделываем аналогичные шаги, но уже для другого блока в перехваченном сообщении, используя только сохраненные ключи.
7. Повторяем этап 6 до тех пор, пока мы не получим один ключ, который будет истинным ключом, с помощью которого мы извлечем необходимую информацию.

**ВЫВОД**

В ходе данной лабораторной работы ознакомились с алгоритмом криптопреобразования – ГОСТ 28147-89 и получили практические навыки его использования.