Отчет по лабораторной работе “Реализация блочного шифра Магма”

Работу выполнила Смирнова Екатерина Андреевна, ККСО-03-17,1 курс

Задача: реализовать шифрование информации с помощью блочного шифра “Магма” на языке C и Python

Теоретическая часть:ГОСТ 28147-89 (Магма) — российский стандарт [симметричного](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [блочного](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D0%BE%D0%B5_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) шифрования, принятый в [1989 году](https://ru.wikipedia.org/wiki/1990_%D0%B3%D0%BE%D0%B4).

Полное название — «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Является примером [DES](https://ru.wikipedia.org/wiki/DES)-подобных [криптосистем](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), созданных по классической итерационной [схеме Фейстеля](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D1%8C_%D0%A4%D0%B5%D0%B9%D1%81%D1%82%D0%B5%D0%BB%D1%8F).

ГОСТ 28147-89 — блочный шифр с 256-[битным](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) [ключом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)) и 32 циклами (называемыми раундами) преобразования, оперирующий 64-битными блоками. Основа [алгоритма](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) шифра — [сеть Фейстеля](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D1%8C_%D0%A4%D0%B5%D0%B9%D1%81%D1%82%D0%B5%D0%BB%D1%8F). Выделяют четыре режима работы ГОСТ 28147-89:

* [простой замены](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BD%D0%B8%D0%B3%D0%B8)
* [гаммирование](https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)
* [гаммирование с обратной связью](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B9_%D1%81%D0%B2%D1%8F%D0%B7%D0%B8_%D0%BF%D0%BE_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83)
* режим выработки [имитовставки](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%BE%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0" \o "Имитовставка).

В своей работе я использовала режим работы простой замены

Для зашифровывания в этом режиме 64-битный блок [открытого текста](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82) сначала разбивается на две половины (младшие биты — A, старшие биты — B). На i-ом цикле используется подключ Ki.{\displaystyle B\_{i+1}=A\_{i}}

Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: K1…K8.

Ключи K9…K24 являются циклическим повторением ключей K1…K8 (нумеруются от младших битов к старшим). Ключи K25…K32 являются ключами K8…K1.

После выполнения всех 32 раундов алгоритма, блоки A33 и B33 склеиваются (обратите внимание, что старшим блоком становится A33, а младшим — B33) — результат есть результат работы алгоритма.

Расшифровывание выполняется так же, как и зашифровывание, но инвертируется порядок подключей Ki.

**Функция {\displaystyle f(A\_{i},K\_{i})}** вычисляется следующим образом:

Ai и Ki складываются по модулю 232.

Результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего *узла таблицы замен* (в порядке возрастания старшинства битов), называемого ниже *S-блоком*. Общее количество S-блоков стандарта — восемь, то есть столько же, сколько и подпоследовательностей. Каждый *S-блок* представляет собой перестановку чисел от 0 до 15 (конкретный вид S-блоков в стандарте не определен). Первая 4-битная подпоследовательность попадает на вход первого S-блока, вторая — на вход второго и т. д.

Если узел *S-блока* выглядит так:

1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12

и на входе S-блока 0, то на выходе будет 1, если 4, то на выходе будет 5, если на входе 12, то на выходе 6 и т. д.

Выходы всех восьми S-блоков объединяются в 32-битное слово, затем всё слово циклически сдвигается влево (к старшим разрядам) на 11 битов.

Режим простой замены имеет следующие недостатки:

* Может применяться только для шифрования открытых текстов с длиной, кратной 64 бит
* При шифровании одинаковых блоков открытого текста получаются одинаковые блоки шифротекста, что может дать определенную информацию криптоаналитику.

Таким образом, применение ГОСТ 28147-89 в режиме простой замены желательно лишь для шифрования ключевых данных.

Практическая часть: в моем коде использовано 3 функции. (Примеры будут рассмотрены на языке С)

Первая функция-fillkey (). С ее помощью пользователь может ввести свой ключ, а функция выполнит разбиения ключа на итерационные ключи Ki, i = 1, 2, …, 32

void fillkey(){

long int key,k0,k1,k2,k3,k4,k5,k6;

fscanf(stdin, "%08x %08x %08x %08x %08x %08x %08x %08x",&key,&k0,&k1,&k2,&k3,&k4,&k5,&k6);

K[0]=key;K[1]=k0;K[2]=k1;K[3]=k2;K[4]=k3;K[5]=k4;K[6]=k5;K[7]=k6;

K[8]=key;K[9]=k0;K[10]=k1;K[11]=k2;K[12]=k3;K[13]=k4;K[14]=k5;K[15]=k6;

K[16]=key;K[17]=k0;K[18]=k1;K[19]=k2;K[20]=k3;K[21]=k4;K[22]=k5;K[23]=k6;

K[24]=k6;K[25]=k5;K[26]=k4;K[27]=k3;K[28]=k2;K[29]=k1;K[30]=k0;K[31]=key;

}

Вторая функция-fest(&a,&b,i); Принимает номер раунда и исходные данные. Далее выполняет алгоритм Фейстеля и выводит зашифрованные данные.

Третья функция-getkey(i). Принимает номер раунда и выбирает подходящий итерационный ключ.

int getkey(int i){

if (mode=='1')

x=K[(i)];

if (mode=='2')

x=K[(31-i)];

}

Заключение: поставленная успешно выполнена задача, разработан программный комплекс, с помощью которого можно выполнить шифрование информации, используя блочный шифр “Магма”