Правительство Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н.Тихонова

ОТЧЕТ
О ПРАКТИЧЕСКОЙ РАБОТЕ № 1
по дисциплине «Криптографические методы защиты информации»
ТЕМА РАБОТЫ

Студент гр. МКБ231

_____

«___» _____ 2024 г.

Руководитель
Заведующий кафедрой информационной безопасности киберфизических систем
канд. техн. наук, доцент

_____

«___» _____ 2024 г.

Москва 2024

СОДЕРЖАНИЕ

1. Задание на практическую работу

Целью работы является создание программной реализации алгоритма блочного шифрования (расшифрования) Магма ГОСТ 34.12-2015

В рамках практической работы необходимо выполнить следующее:

1. Изучить ГОСТ 34.12-2015
2. Составить описание работы алгоритма Магма ГОСТ 34.12-2015
3. Создать программу на удобном языке программирования, которая принимает на вход:
   a. Файл исходного сообщения и файл ключа для процедуры зашифрования
   b. Файл шифр текста и файл ключа для процедуры расшифрования

   и формирует на выходе файл за(рас)шифрованного сообщения.
4. Подготовить отчет о проделанной работе.

2. Описание алгоритма Магма ГОСТ 34.12-2015

2.1 Описание процедуры шифрования

Алгоритм блочного шифрования Магма используется для зашифрования и расшифрования сообщений с использованием закрытого ключа размером 256 бит. Кроме того, ГОСТ 34.12-2015 определены значения полей блока замены.

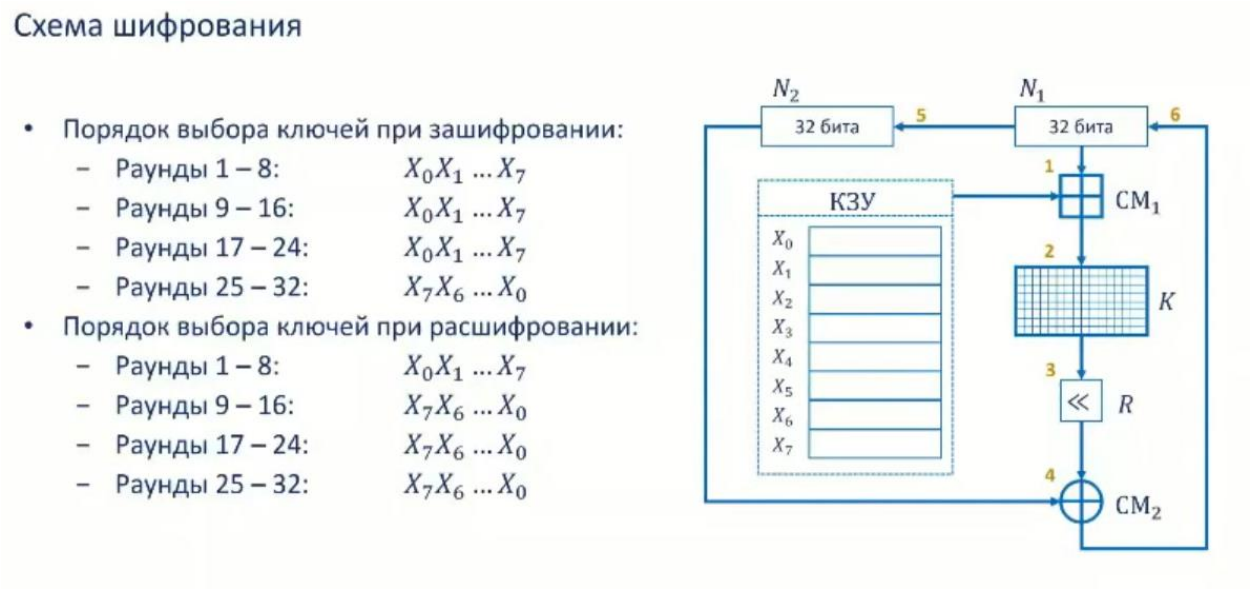

Рисунок 1.1. – Схема шифрования

На рисунке 1.1 представлена схема шифрования с использованием алгоритма Магма. Шифрования происходит следующим образом:

1. Ключ шифрования 256 бит разбивается на 8 32-битных сеансовых ключей - значения x0..x8.
2. Входящее сообщение разбивается на 64-битные блоки, последний неполный блок дополняется нулевыми значениями до размера 64 бит.
3. 64-битный блок из входящего сообщения разделяется на два 32-битных блока и заносится в регистры N1(младшая часть) и N2(старшая часть).
4. Теперь начинается раунд шифрования:
   4.1.значение регистра N1 суммируется (СМ1) по модулю 2 в степени 32 с определенным сеансовым ключом шифрования X (выбирается в зависимости от раунда).
   4.2.Полученная сумма поступает в блок замены K, где в зависимости от входящего 32-битного значения формируется исходящее 32-битное значение, получаемое путем замены исходных 4-битных блоков на предопределенные ГОСТ.
   4.3.Далее 32-битное значение поступает в блок R, где осуществляется циклический сдвиг влево на 11 разрядов.
   4.4.Полученное значение складывается по модулю 2 (СМ2) со значением из регистра N2.
   4.5. Наконец в регистр N2 записывается содержимое регистра N1, а в регистр N1 записывается полученное значение (из СМ2).

5. На этом раунд шифрования завершен, берется следующий сеансовый ключ (выбирается в зависимости от операции (зашифрование или расшифрование) и порядкового номера раунда согласно Схемы шифрования (рис. 1.1)
6. После завершения 32 раундов шифрования значения регистров N1 и N2 записываются в 64-битный блок с перестановкой (N2 младшая часть, N1 старшая часть), который добавляется к шифр сообщению.
7. Из входящего сообщения берется следующий 64-битный блок и заносится в регистры N1 и N2 и выполняются очередные 32 раунда преобразований.

2.2 Описание процедуры расшифрования

Алгоритм блочного шифрования Магма является симметричным, поэтому процедура расшифрования отличается от процедуры зашифрования обратным порядком сеансовых ключей шифрования согласно Схеме шифрования (рис. 1.1)

2.3 Значения подстановок для нелинейного биективного преобразования согласно ГОСТ 34.12-2015 для 64-битных блоков

π0 ' = (12, 4, 6, 2, 10, 5, 11, 9, 14, 8, 13, 7, 0, 3, 15, 1);
π1 ' = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15);
π2 ' = (11, 3, 5, 8, 2, 15, 10, 13, 14, 1, 7, 4, 12, 9, 6, 0);
π3 ' = (12, 8, 2, 1, 13, 4, 15, 6, 7, 0, 10, 5, 3, 14, 9, 11);
π4 ' = (7, 15, 5, 10, 8, 1, 6, 13, 0, 9, 3, 14, 11, 4, 2, 12);
π5 ' = (5, 13, 15, 6, 9, 2, 12, 10, 11, 7, 8, 1, 4, 3, 14, 0);
π6 ' = (8, 14, 2, 5, 6, 9, 1, 12, 15, 4, 11, 0, 13, 10, 3, 7);
π7 ' = (1, 7, 14, 13, 0, 5, 8, 3, 4, 15, 10, 6, 9, 12, 11, 2).

3. Реализация алгоритма Магма ГОСТ 34.12-2015

3.1 Общие сведения

Программа за(расшифрования) реализована на языке программирования C# в виде консольного приложения с использованием IDE Visual Studio 2022 и включает в себя два файла Program.cs и Methods.cs

3.2 Описание работы программы

1. Файл сообщения или шифр текста загружается в массив байт, который дополняется для кратности 64 битам нулевыми значениями и переносится в массив 64-битных значений (входящее сообщение).
2. Файл ключа загружается в массив байт, который переносится в массив из 8 32-битных сессионных ключей. Затем формируются два массива по 32 32-битных сеансовых ключей (для зашифрования и расшифрования).
3. Производится процедура за(рас)шифрования входящего сообщения, в результате которого получается исходящее сообщение (массив 64-битных значений), которое преобразуется в массив байт и записывается в файл.
4. Список методов, реализованных в программе:
    - 4.1 **ulong crypt(ulong in_data, uint[] session_keys)** – функция, осуществляющая шифрование 64-битного блока, возвращает 64-битный блок
    - 4.2 **uint add_mod2_32(uint a, uint b)** – функция, осуществляющая сложение двух 32-разрядных чисел в кольце вычетов по модулю 2 в степени 32
    - 4.3 **uint replace_K(uint in_data)** – функция, осуществляющая подстановку значений операции биективного преобразования 32-битного значения
    - 4.4 **uint uint_from_bytes(byte[] input, int index)** – функция формирует 32-битное число из массива байт с определенной позиции
    - 4.5 **ulong ulong_from_bytes(byte[] input, int index)** – функция формирует 64-битное число из массива байт с определенной позиции
    - 4.6 **byte[] bytes_from_uint(uint input)** – функция формирует массив байт из 32-битного значения
    - 4.7 **byte[] bytes_from_ulong(ulong input)** – функция формирует массив байт из 64-битного значения
    - 4.8 **ulong[] read_from_file(string filename)** – функция читает файл сообщения в массив 64-битных значений
    - 4.9 **byte[] abyte_from_aulong(ulong[] aulong)** – функция формирует массив байт из массива 64-битных блоков

4. Результаты работы программы

4.1 Входные данные

1. Файл исходного сообщения – message.txt
2. Файл ключей – magma.key
3. Файл зашифрованного сообщения – cipher.dat

4.2 Выходные данные

1. Файлы зашифрованного сообщения – cipher.dat, control.dat
2. Файл расшифрованного сообщения – control.txt

4.3 Использование программы

1. Зашифрование – **magma.exe -e message.txt -k magma.key > encres.txt**
2. Расшифрование - **magma.exe -d cipher.dat -k magma.key > decres.txt**

4.4 Результат работы программы(консольный вывод)

1. Зашифрование:

Исходное 64 разрядное сообщение	Зашифрованное 64 разрядное сообщение
	7572429346300757947
15036905090638029009	2422411951271356200
9426233931517231824	11930594693709831105
13749686271986414880	9501281920132526340
15038312478406529204	18062001058642844390
15100518465438863541	10248874258305507929
2364814012900753104	2699866033154437383
12813173944342330832	4313121396218282557
13317343250679648443	15054947309170657772
15100324159811532241	17702498142134178244
9354401733990777040	1425094775709865767
13749686271751536080	9899992826148327023
13317582859290116304	11278128447631334428
12740878855943848657	6335749201606774878
10025242292798935230	7212068509157453656
15038593931907485884	17905164913307339241
15041127250991876304	5676857402969038482
13317571608455462608	3185072195428049040
12884992944491247825	342055060126703519
10313473515045572793	15881586821504147525
15040844835607072952	14504447386543080178
15096576948182110395	7350124007286090089
15099673138579100046	18012945685375238504
15098828705045744906	9570900605450616503
15037748660868993214	16148347729019698748
15096295443153473724	9753713740797912626
2364562224503108560	5437733531316999081
13605568129286656190	10181891749963535805
15038593085798928769	13428066215444969001
15041407845500228560	983027142368842143
11443824570593719092	11504803822207922368

3328496769300443445	11564143847863569650
3318319811039556384	17920302022320132711
7881694542353686892	16576416879334915896
7453301705571593504	7760083729328909203
8102661169684245806	

2. Расшифрование

Исходный 64 разрядный шифртекст	Расшифрованное 64 разрядное сообщение
7572429346300757947	15036905090638029009
2422411951271356200	9426233931517231824
11930594693709831105	13749686271986414880
9501281920132526340	15038312478406529204
18062001058642844390	15100518465438863541
10248874258305507929	2364814012900753104
2699866033154437383	12813173944342330832
4313121396218282557	13317343250679648443
15054947309170657772	15100324159811532241
17702498142134178244	9354401733990777040
1425094775709865767	13749686271751536080
9899992826148327023	13317582859290116304
11278128447631334428	12740878855943848657
6335749201606774878	10025242292798935230
7212068509157453656	15038593931907485884
17905164913307339241	15041127250991876304
5676857402969038482	13317571608455462608
3185072195428049040	12884992944491247825
342055060126703519	10313473515045572793
15881586821504147525	15040844835607072952
14504447386543080178	15096576948182110395
7350124007286090089	15099673138579100046
18012945685375238504	15098828705045744906
9570900605450616503	15037748660868993214
16148347729019698748	15096295443153473724
9753713740797912626	2364562224503108560
5437733531316999081	13605568129286656190
10181891749963535805	15038593085798928769
13428066215444969001	15041407845500228560
983027142368842143	11443824570593719092
11504803822207922368	3328496769300443445
11564143847863569650	3318319811039556384
17920302022320132711	7881694542353686892
16576416879334915896	7453301705571593504
7760083729328909203	8102661169684245806

5 Выводы о проделанной работе

Был изучен и реализован алгоритм симметричного блочного шифрования Магма ГОСТ 34.12-2015 в виде консольного приложения, запускаемого с параметрами. Программа производит зашифрование и расшифрование файлов с использованием подготовленного

файла ключа. Для возможности последующего использования в других проектах необходимо скомпоновать программу в виде класса. Использование блочных промежуточных 64-битных массивов возможно исключить, операции производить сразу с массивами байт.

6 Список использованных источников

1. Технический комитет по стандартизации «криптографическая защита информации», ГОСТ 34.12-2015 URL: https://tc26.ru/standard/gost/GOST_R_3412-2015.pdf (Дата обращения 06.04.2024)