## Практическая работа №3 - Симметричный алгоритм шифрования

**Цель работы:**

Исследование структуры симметричных алгоритмов шифрования: *AES, ГОСТ 28147-89, DES, RС6, IDEA, SEED.*

**Задание:**

- Реализовать один из существующих симметричных алгоритмов шифрования: *AES,.ГОСТ 28147-89, DES, RС6, IDEA, SEED.*

- Разработать форму содержащую: 3 текстовых поля, кнопку «Шифровать», кнопку «Расшифровать», текстовое поле ключ для шифрации, текстовое поле ключ для дешифрации.

- Организовать дополнительный интерфейс настройки алгоритма по необходимости.

- Реализовать разработанный алгоритм на любом языке программирования.

**Методика выполнения Практической работы**

Студентами рассматриваются один из алгоритмов симметричного шифрования. Изучается алгоритм работы выбранного алгоритма шифрования и после чего, реализуется на каком либо языке программирования.

**Основные теоретические положения**

Симметричные криптосистемы - способ шифрования, в котором для [шифрования](http://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [расшифровывания](http://ru.wikipedia.org/w/index.php?title=%D0%A0%D0%B0%D1%81%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) применяется один и тот же криптографический [ключ](http://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_%28%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F%29). До изобретения [схемы асимметричного шифрования](http://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC) единственным существовавшим способом являлось симметричное шифрование. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Ключ алгоритма выбирается сторонами до начала обмена сообщениями. В настоящее время симметричные шифры - это:

* [блочные шифры](http://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80). Обрабатывают информацию блоками определённой длины (обычно 64, 128 бит), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми [раундами](http://ru.wikipedia.org/w/index.php?title=%D0%A0%D0%B0%D1%83%D0%BD%D0%B4&action=edit&redlink=1). Результатом повторения раундов является [лавинный эффект](http://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B2%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8D%D1%84%D1%84%D0%B5%D0%BA%D1%82) - нарастающая потеря соответствия [битов](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) между блоками открытых и зашифрованных данных.
* [поточные шифры](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80), в которых шифрование проводится над каждым [битом](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) либо [байтом](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82) исходного (открытого) текста с использованием [гаммирования](http://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). Поточный шифр может быть легко создан на основе блочного (например, [ГОСТ 28147-89](http://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147-89) в режиме гаммирования), запущенного в специальном режиме.

Основными параметрами алгоритмов симметричного шифрования являются:

* [стойкость](http://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%BE%D0%B9%D0%BA%D0%BE%D1%81%D1%82%D1%8C);
* длина ключа;
* число [раундов](http://ru.wikipedia.org/w/index.php?title=%D0%A0%D0%B0%D1%83%D0%BD%D0%B4&action=edit&redlink=1);
* длина обрабатываемого блока;
* сложность аппаратной/программной реализации;
* сложность преобразования.

В качестве примера алгоритма симметричного шифрования можно рассмотреть отечественный алгоритм шифрования ГОСТ 28147-89, который определен в стандарте[3]. Алгоритм шифрует данные 64-битными блоками с использованием 256-битного ключа шифрования. Базовым режимом шифрования по ГОСТ 28147—89 является режим [простой замены](http://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BC%D0%B5%D0%BD%D0%B0&action=edit&redlink=1). Для зашифрования в этом режиме [открытый текст](http://ru.wikipedia.org/w/index.php?title=%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&action=edit&redlink=1) сначала разбивается на две половины (младшие биты - A, старшие биты - B). На i-ом цикле используется подключ Ki:

( = двоичное «исключающее или»)



Выполняется 32 раунда преобразований, в каждом из которых предусмотрены следующие операции (рис. 2):

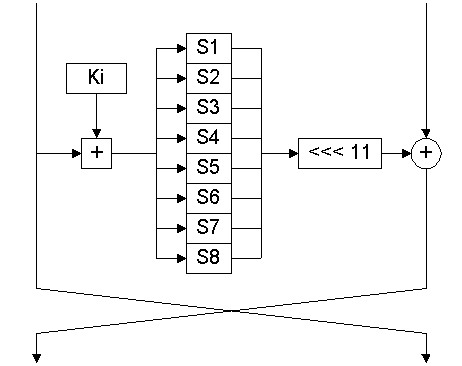


Рис. 2. Раунд алгоритма ГОСТ 28147-89

Шаг 0. Один из 32-битных субблоков данных складывается с 32-битным значением ключа раунда **Ki** по модулю 232.

Шаг 1. Результат предыдущей операции разбивается на 8 фрагментов по 4 бита, которые параллельно «прогоняются» через 8 таблиц замен **S1…S8**. Таблицы замен в стандарте [3] не определены. В качестве примера реализации можно привести следующий исходный код;

for (int i = 7; i > -1; i--)

{

S[i] = (byte)((\*N1) & 0x0F);

S[i] = H[i,S[i]];

(\*N1) = (\*N1) >> 4;

}

Шаг 2. 4-битные фрагменты (после замен) объединяются обратно в 32-битный субблок, значение которого циклически сдвигается влево на 11 бит.

for (int i = 0; i < 8; i++)

{

(\*N1) = (\*N1) | S[i];

(\*N1) = (\*N1) << 4;

}

rot Rot = new rot();

(\*N1) = Rot.left((\*N1),11);

Где класс rot имеет описан следующим образом вида:

public class rot

{

private const uint intSize = sizeof(int);

public uint left(uint N, int n)

{

uint a;

uint k = (uint)(1 << (int)(intSize - 1));

for (int i = 1; i <= (n % intSize); i++)

{

a = N & k;

N = N << 1;

a = (uint)((int)a >> (int)(intSize - 1));

N = N | a;

}

return N;

}

public uint right(uint N, int n)

{

uint a;

for (int i = 1; i <= (n % intSize); i++)

{

a = N & 1;

N = N >> 1;

a = (uint)((int)a << (int)(intSize - 1));

N = N | a;

}

return N;

}

}

Шаг 3. Обработанный предыдущими операциями субблок накладывается на необработанный с помощью побитовой логической операции «исключающее или» (XOR).

*(\*N1) = (\*N1) ^ (\*N2)*;

Шаг 4. Субблоки меняются местами.

Процедура расширения ключа в алгоритме ГОСТ 28147-89, фактически, отсутствует: в раундах шифрования последовательно используются 32-битные фрагменты K1…K8 исходного 256-битного ключа шифрования в следующем порядке: **K1, K2, K3, K4, K5, K6, K7, K8** – за исключением последних 8 раундов – в раундах с 25-го по 31-й фрагменты используются в обратном порядке.

Расшифрование полностью аналогично зашифрованию, но с другим порядком использования фрагментов ключа:

- в прямом порядке – в первых 8 раундах;

- в остальных раундах – в обратном порядке.

Стандарт [3] также предусматривает и описывает различные режимы применения алгоритма:

- описанный выше режим простой замены;

- режимы гаммирования и гаммирования с обратной связью, предусматривающие вычисление с помощью описанных выше преобразований псевдослучайной последовательности – гаммы шифра – и ее наложение на шифруемый текст;

- режим вычисления имитовставки – криптографической контрольной суммы, используемой для подтверждения целостности данных; в данном режиме выполняется 16 раундов преобразований вместо 32-х.

Как видно из описания, алгоритм ГОСТ 28147-89 является весьма простым в реализации, что является его несомненным достоинством.

К достоинствам ГОСТа можно отнести:

- бесперспективность [силовой атаки](http://ru.wikipedia.org/wiki/Brute_force) ([XSL-атаки](http://ru.wikipedia.org/wiki/XSL_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0) в учёт не берутся, т.к. их эффективность на данный момент полностью не доказана);

- эффективность реализации и соответственно высокое быстродействие на современных [компьютерах](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80);

- наличие защиты от навязывания ложных данных (выработка имитовставки) и одинаковый цикл шифрования во всех четырех алгоритмах ГОСТа.[4]

На рис. 3 представлен результат работы алгоритма шифрования ГОСТ 34.10-2001. В левом текстовом поле приведен текс подвергаемый шифрованию. В правом текстовом поле приведен уже зашифрованный текст отраженный в китайской раскладке.

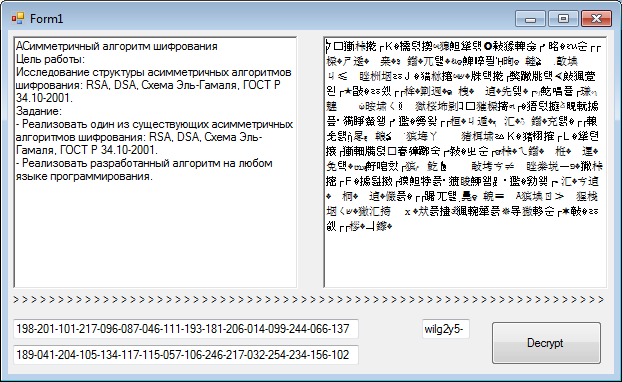


Рис. 3. Форма Алгоритм шифрования ГОСТ 34.10-2001

**Требования к предъявляемой работе:**

- Программа шифрования должны быть наделена понятным и удобным пользовательским интерфейсом.

- Полная реализация выбранного алгоритма.

- Запрещено использование любых сторонних библиотек.