ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Информационная безопасность

Отчет по лабораторной работе №4  
«Демонстрационная версия криптостойкого блочного алгоритма Rijndael»

Выполнил:  
Студент группы №P34101

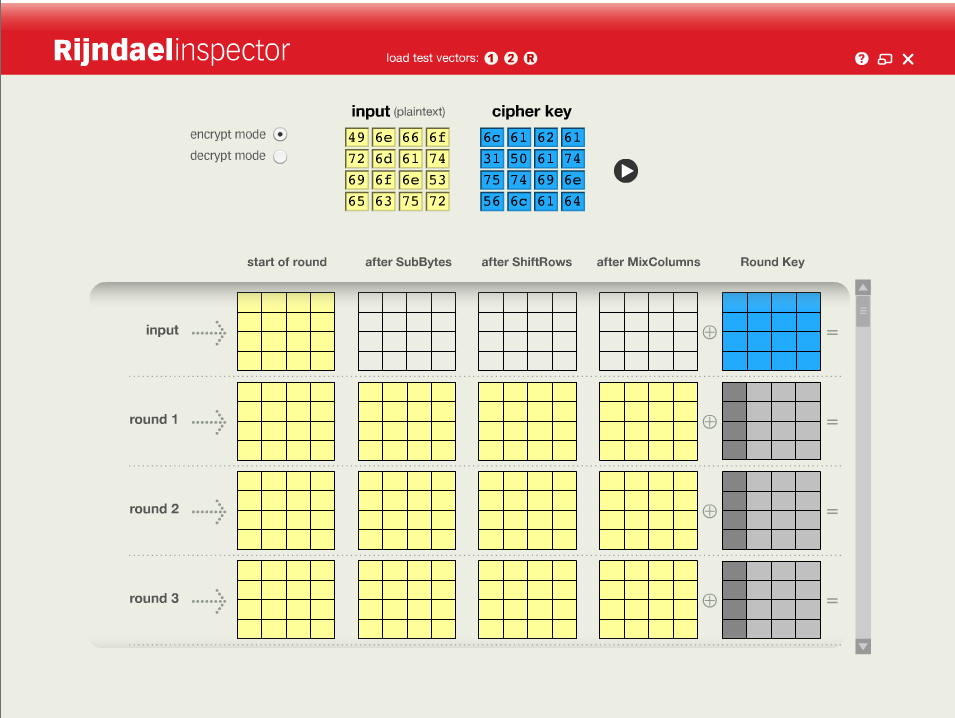
*Патутин В.М.*Преподаватель:  
*Маркина Т. А.*

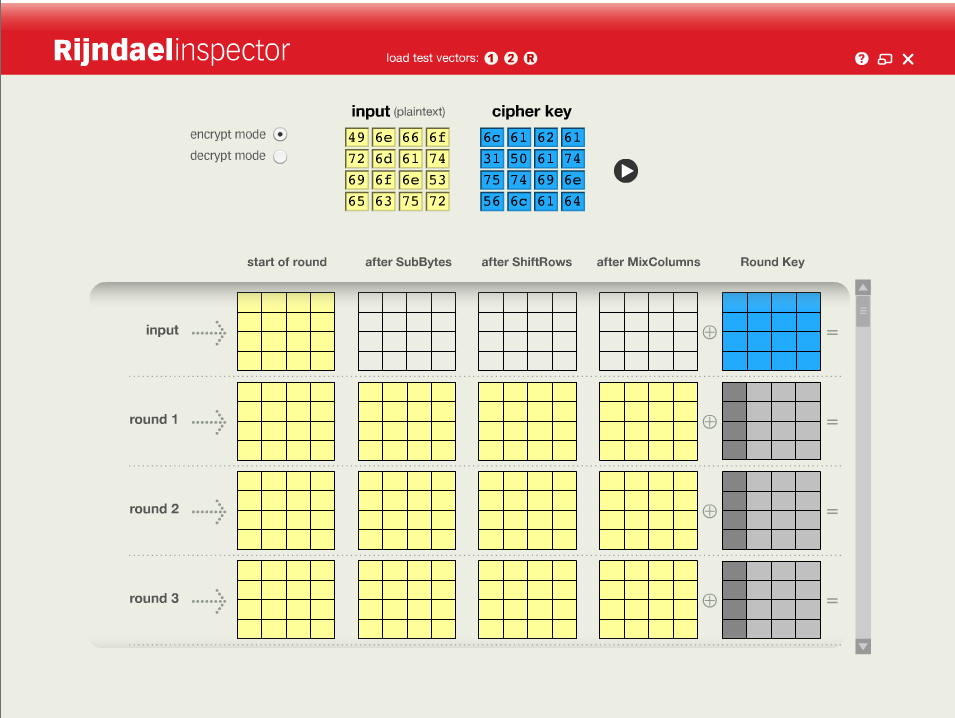
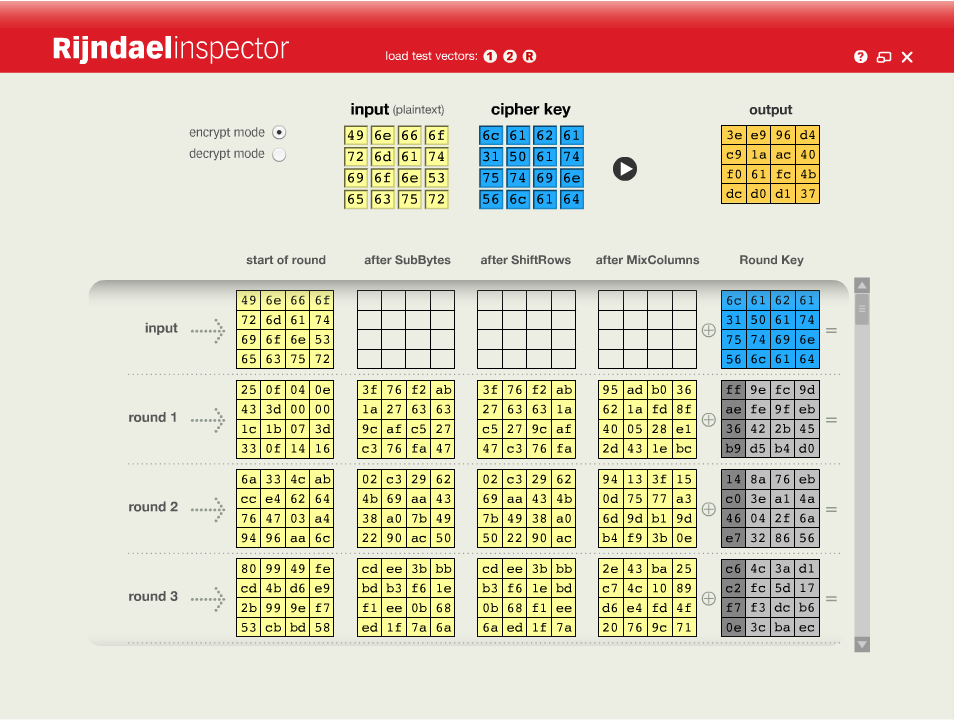
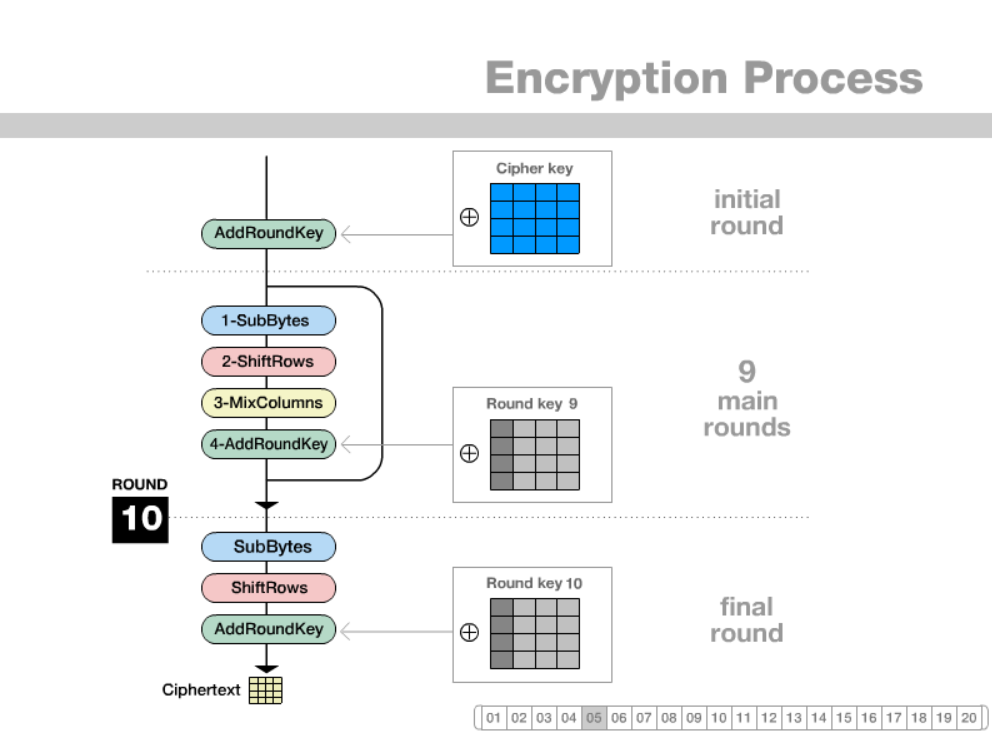
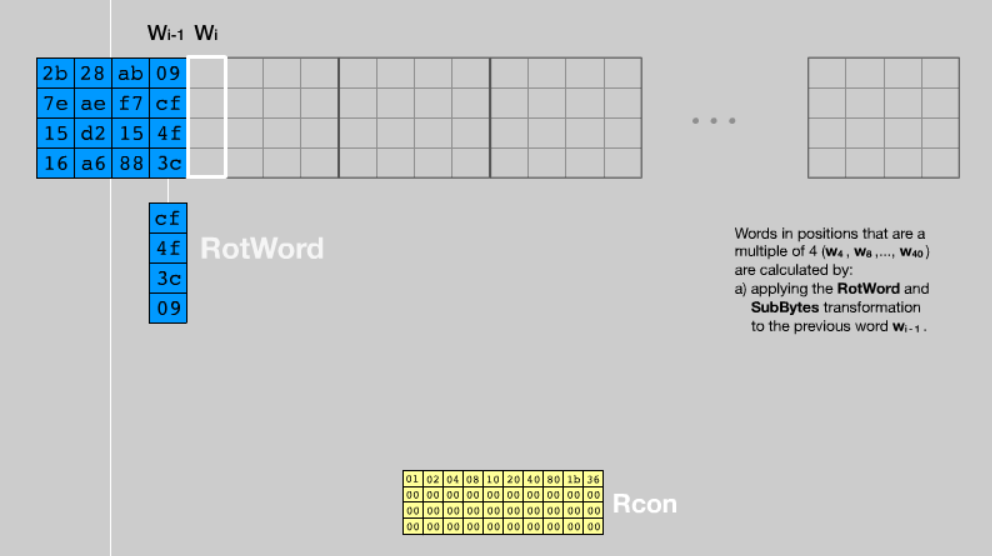
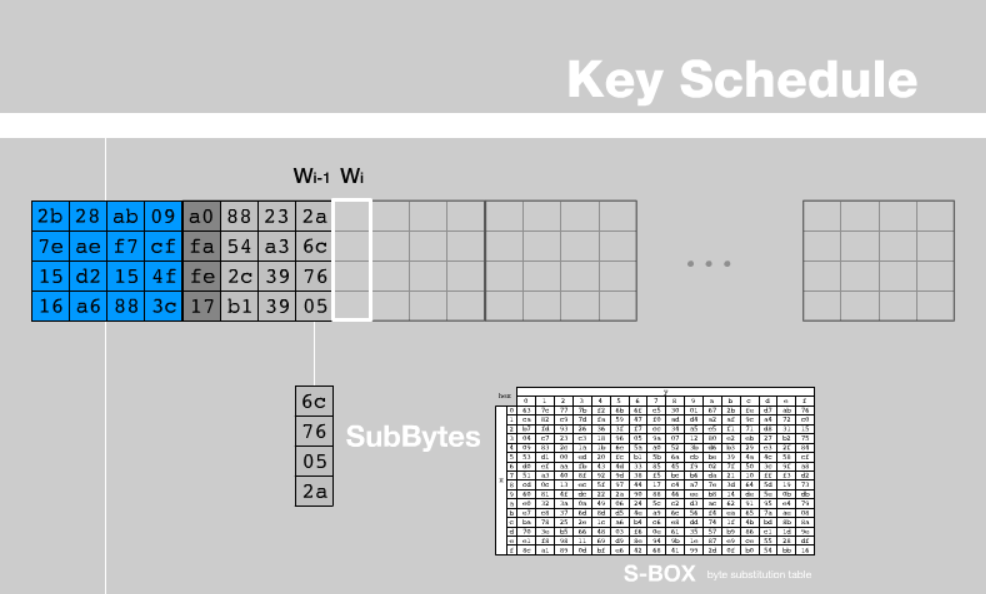
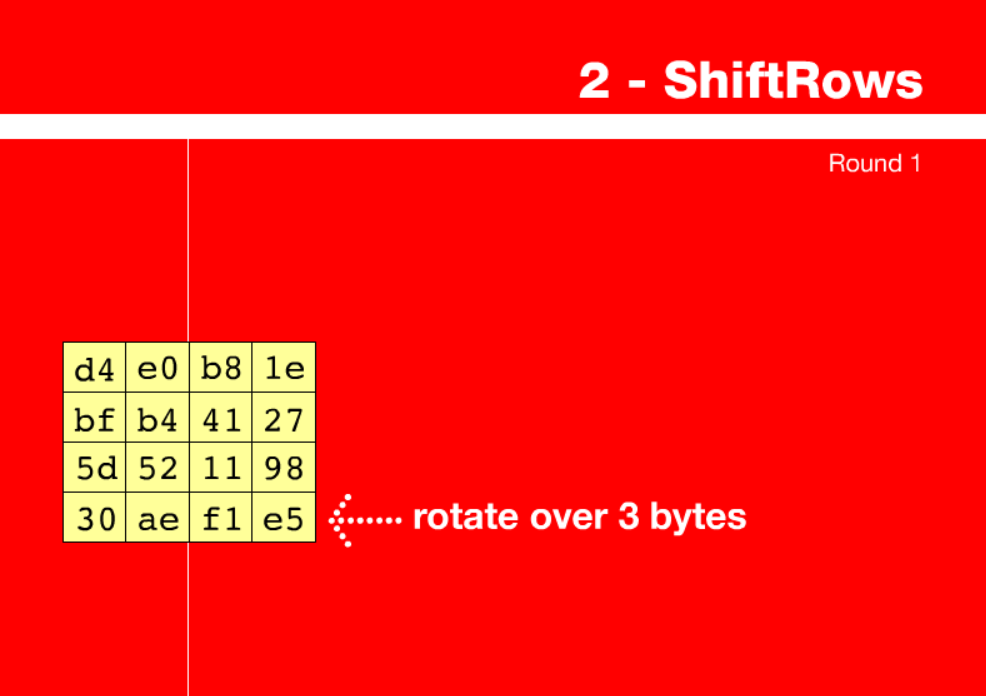
Санкт-Петербург

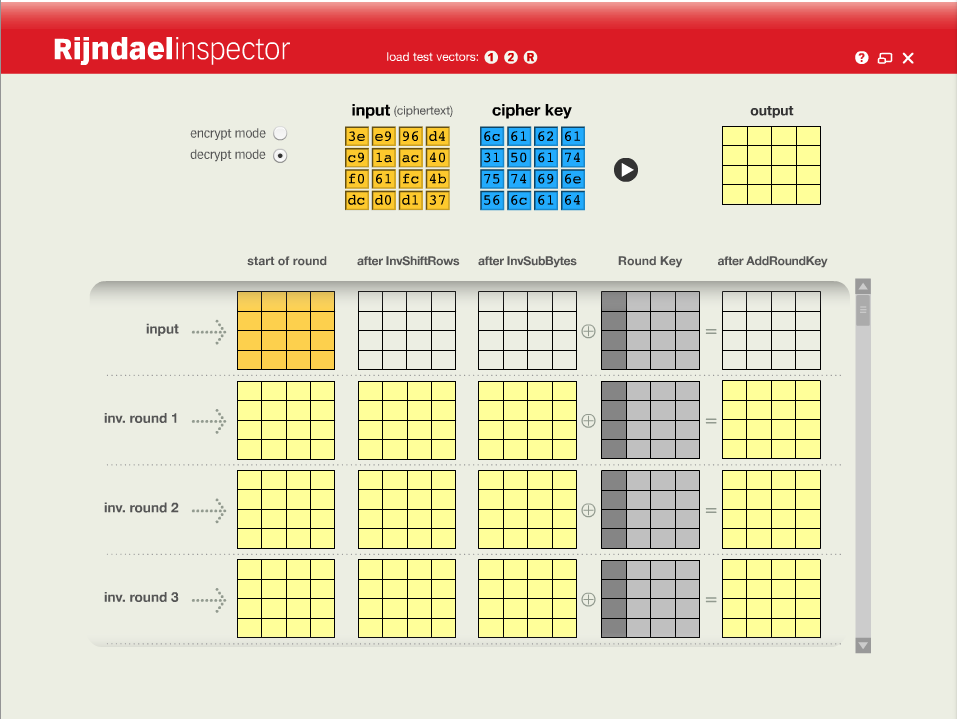
2022

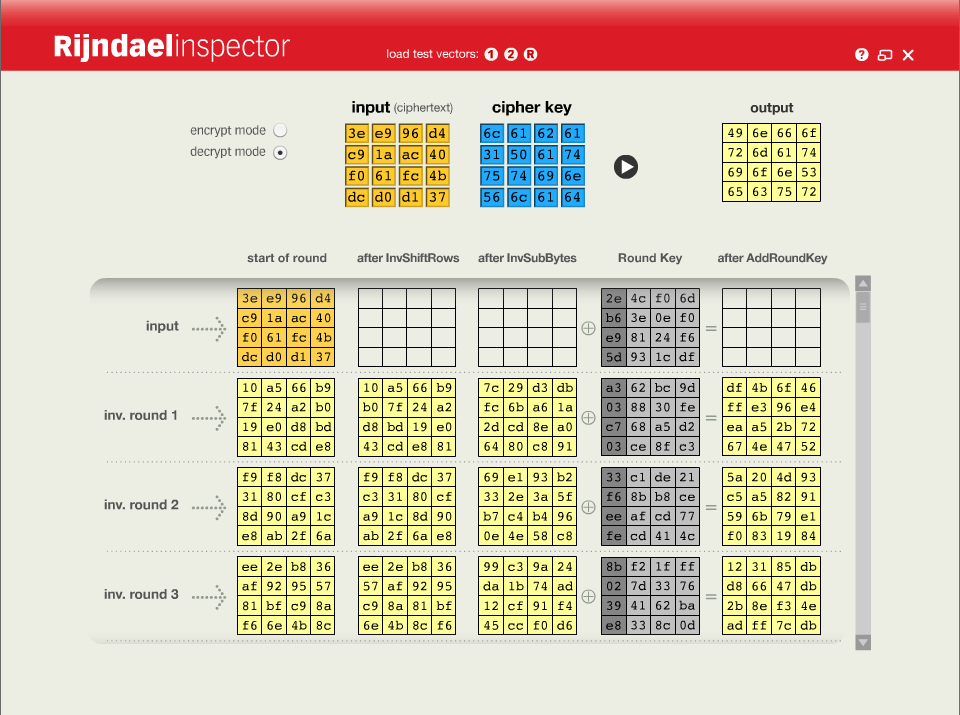
1. **Цель работы**Ознакомление с принципами шифрования, используемыми в алгоритме симметричного шифрования **AES RIJNDAEL**
2. Вариант **задания**   
   Задание по варианту 22  
   Сравните эквивалентность прямого и обратного преобразований в алгоритмах Rijndael и ГОСТ 28147-89.
3. **Листинг программы с комментариями**
   1. **Входные данные для работы алгоритма**  
      В качестве текста для шифрования я взял строку вида:  
      InformationSecurityInAction  
      И её представление в 16-ричном виде согласно таблице ASCII  
      49 6E 66 6F 72 6D 61 74 69 6F 6E 53 65 63 75 72 69 74 79 49 6E 41 63 74 69 6F 6E

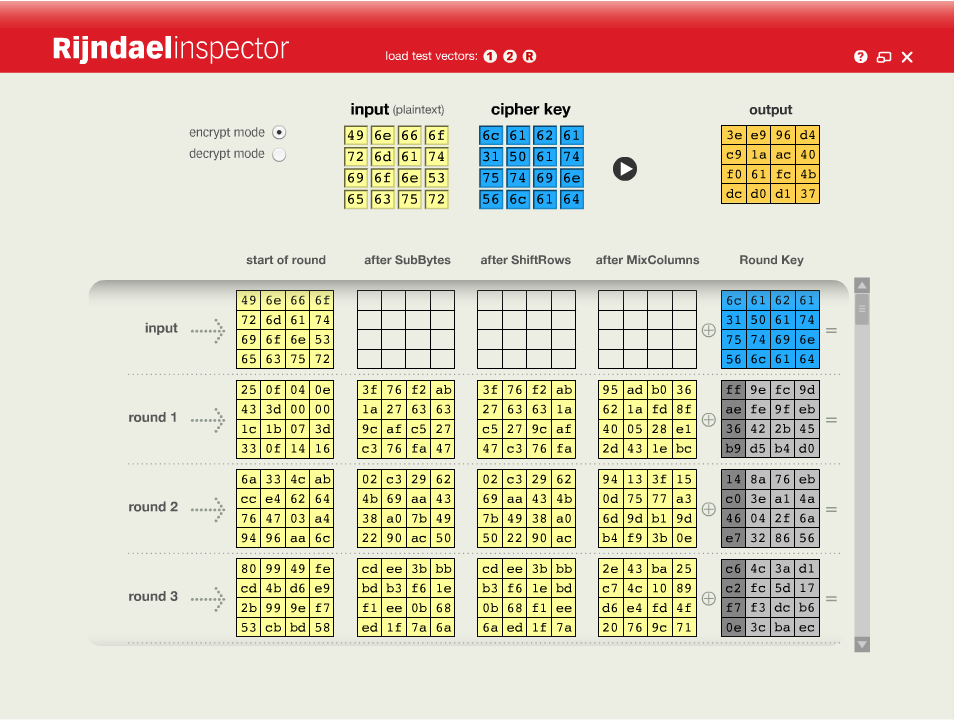
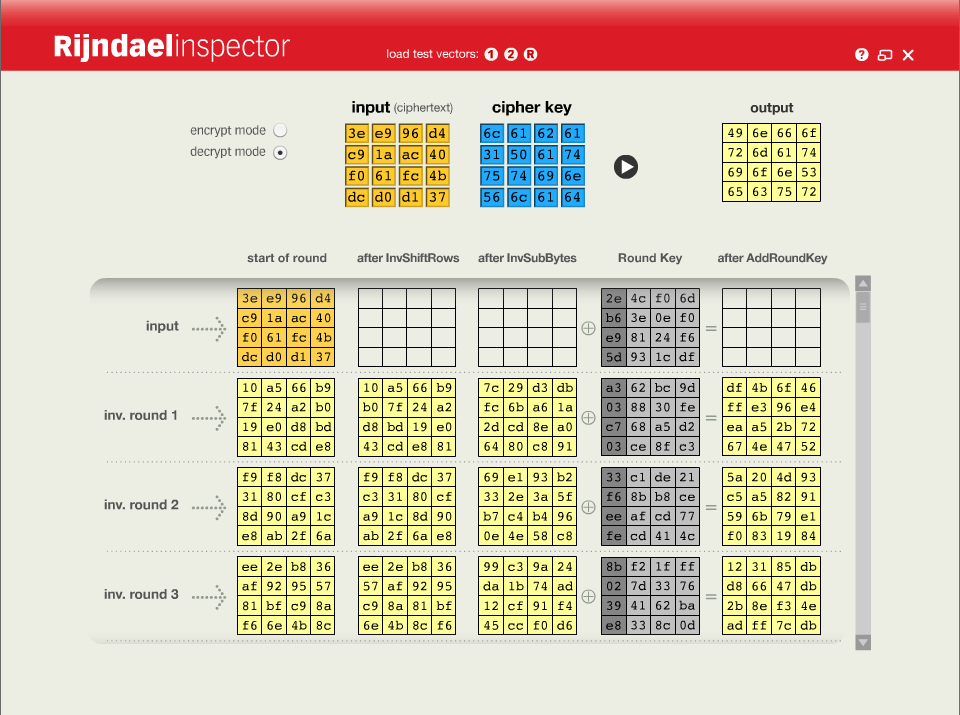
Ограничиваем строку до 16 символов, чтобы можно было воспользоваться программой:  
49 6E 66 6F 72 6D 61 74 69 6F 6E 53 65 63 75 72  
  
Также нам необходим ключ, который будет в текстовом формате иметь вид: laba1PatutinVladimir2022

6C 61 62 61 31 50 61 74 75 74 69 6E 56 6C 61 64 69 6D 69 72 32 30 32 32  
И её представление с учетом ограничений будет:  
6C 61 62 61 31 50 61 74 75 74 69 6E 56 6C 61 64

* 1. **Пошаговое выполнение алгоритма шифрования**Загоним данные в программу Rijndael Inspector  
     Исходное положение программы:  
       
       
     Запускаем алгоритм шифрования  
       
       
     На примере входного значения и первого раунда разберем работу алгоритма.   
       
     Общий процесс:  
       
       
     Первый этап сразу после начала работы программы **addRoundKey –** на этом этапе производится матричный XOR с **roundKey**, roundKey вычисляется путем применения процедуры keyExpansion к ключу шифрования  
       
     keyExpansion  
       
     Алгоритм берет wi-1 слово делает RotWord и SubBytes и производит XOR с wi-4 (разница в индексах зависит от размера матрицы) и XOR cо столбцом Rcon (константные значения на каждый раунд, по столбцу на каждый раунд)  
       
     **RotWord** - функция, использующаяся в процедуре Key Expansion, которая берёт 4-байтовое слово и производит над ним циклическую перестановку  
       
     **SubBytes** - трансформации при шифровании, которые обрабатывают State, используя нелинейную таблицу замещения байтов (S-box), применяя её независимо к каждому байту State  
       
       
       
     После этого начинается несколько раундов, каждый из которых обозначен четырьмя операциями, несколько из них мы разобрали выше  
       
     **ShiftRows** – поэлементный сдвиг влево на +1 каждой строки матрицы  
       
       
     **MixColumns -** трансформация при шифровании, которая берёт все столбцы State и смешивает их данные (независимо друг от друга), чтобы получить новые столбцы
  2. **Пошаговое выполнение дешифрования**Основные операции выполняются по инвертированному механизму от основных операций



  
  
На каждом раунде выполняются операции:  
**InvShiftRows** – обратная к операции shift rows, следовательно поэлементный сдвиг вправо  
**InvSubBytes** – обратная операция к subBytes, дешифрование через таблицу замещения  
**AddRoundKey** – описанная в шифровании, не изменяется  
**InvMixColumns** – обратная операция к mixColumns  
  
Перед и после начала раундов выполняется AddRoundKey, чтобы сохранить последовательность, после раундов выполняются по одной операции InvShiftRows, InvSubBytes, AddRoundKey

1. **Результат работы программы**
   1. **Шифрование  
      Дешифрование**
   2. ****
2. **Ответ на контрольные вопросы  
   Вариант 22**Сравните эквивалентность прямого и обратного преобразований в алгоритмах Rijndael и ГОСТ 28147-89.

**Гост:**

В алгоритме ГОСТ28147-89 эквивалентность структуры прямого и обратного криптографического преобразования не обеспечивается специально, а является простым следствием использованного архитектурного решения. В любой однородной сбалансированной сети Файстеля оба эти преобразования алгоритмически идентичны и различаются только порядком использования ключевых элементов: при расшифровании элементы используются в порядке, обратном тому, в котором они используются при зашифровании.

**Rijndael:**  
Шифр Rijndael построен на базе прямых преобразований. Как и для всех подобных алгоритмов, обратное преобразование строится из обращений шагов прямого преобразования, применяемых в обратном порядке. В силу сказанного обеспечить такую же степень идентичности прямого и обратного преобразования, которая достигается в сетях Файстеля, не представляется возможным. Однако специальными конструкторскими решениями достигается близкая степень соответствия: прямое и обратное преобразование получаются идентичными с точностью до используемых в преобразованиях констант.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
<https://ecm-journal.ru/files/150076131.pdf>