1. **Формулировка задания**

**Задание:**

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:

1. Шифровать данные с использованием заданного в варианте режима шифрования, применённого для того симметричного алгоритма, который был реализован в предыдущей лабораторной работе:

1) шифруемый текст должен храниться в одном файле, ключ шифрования – в другом, а вектор инициализации – в третьем;

2) зашифрованный текст должен сохраняться в файл;

3) в процессе шифрования предусмотреть возможность просмотра и изменения ключа, вектора инициализации, шифруемого и зашифрованного текстов в шестнадцатеричном и символьном виде.

2. Шифровать данные по заданной в варианте схеме кратного шифрования.

3. Исследовать лавинный эффект:

1) приложение может самостоятельно строить необходимые графики либо графики можно строить в стороннем ПО, но тогда приложение для шифрования должно сохранять в файл необходимую для построения графиков информацию.

II. Реализовать приложение для дешифрования, позволяющее выполнять следующие действия:

1. Дешифровать данные с использованием заданного в варианте режима шифрования, применённого для того симметричного алгоритма, который был реализован в предыдущей лабораторной работе:

1) зашифрованный текст должен храниться в одном файле, ключ – в другом, а вектор инициализации – в третьем;

2) расшифрованный текст должен сохраняться в файл;

3) в процессе дешифрования предусмотреть возможность просмотра и изменения ключа, вектора инициализации, зашифрованного и расшифрованного текстов в шестнадцатеричном и символьном виде.

2. Дешифровать данные по заданной в варианте схеме кратного шифрования.

III. С помощью реализованных приложений выполнить следующие задания:

1. Протестировать правильность работы разработанных приложений.

2. Исследовать лавинный эффект для реализованного режима шифрования (рассматривать текст из трёх блоков):

1) построить графики зависимости числа изменённых бит в блоках, от позиции изменившегося бита в открытом тексте (3 отдельных графика или 3 зависимости на 1 графике);

2) построить графики зависимости числа изменённых бит в блоках, от позиции изменившегося бита в ключе (3 отдельных графика или 3 зависимости на 1 графике);

3) построить графики зависимости числа изменённых бит в блоках, от позиции изменившегося бита в векторе инициализации (3 отдельных графика или 3 зависимости на 1 графике);

4) построить графики зависимости числа изменённых бит в блоках, от позиции изменившегося бита в зашифрованном тексте (3 отдельных графика или 3 зависимости на 1 графике).

3. Исследовать лавинный эффект для реализованной схемы кратного шифрования (рассматривать текст из 1 блока).

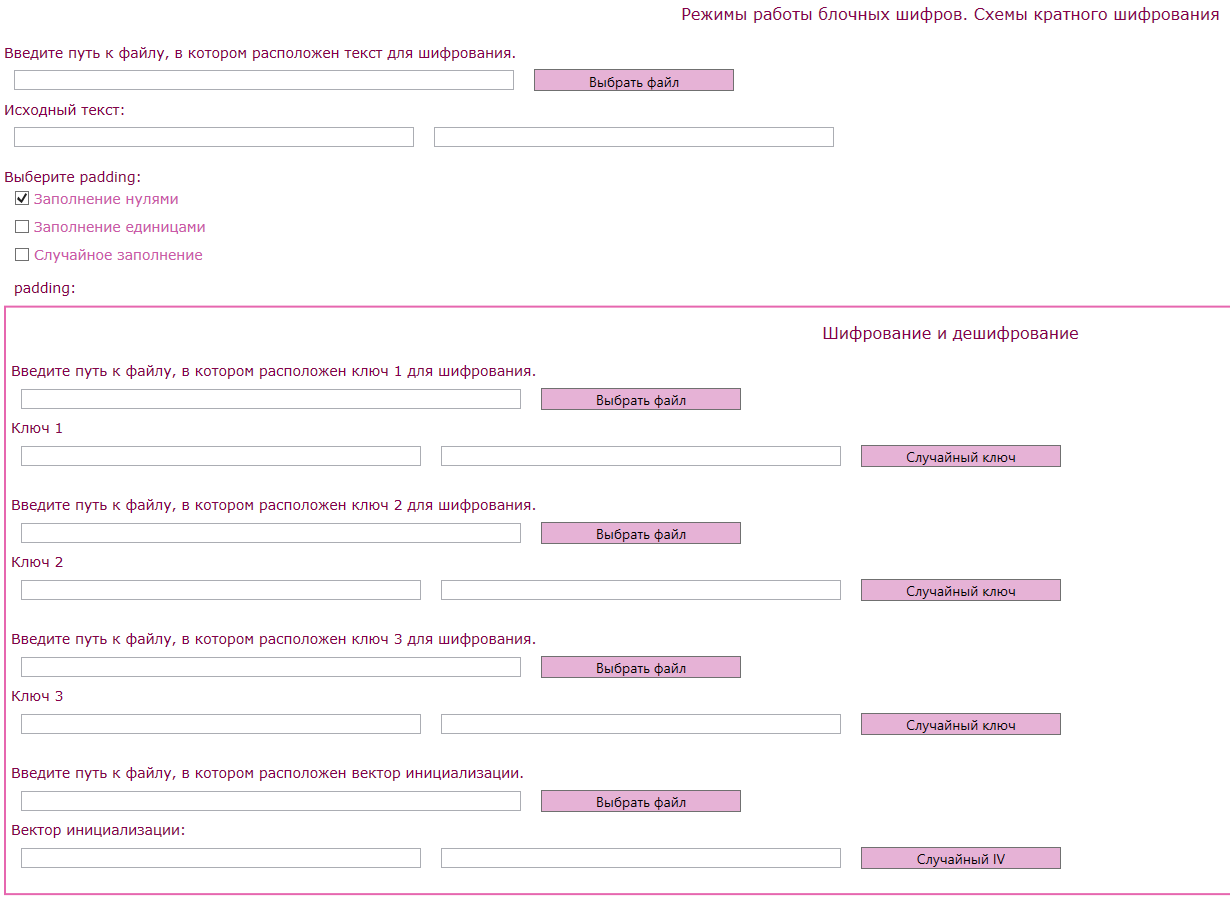
4. Сделать выводы о проделанной работе.

**Вариант 3:**

|  |
| --- |
| Режим шифрования: PCBC |
| Схема кратного шифрования: EDE |

1. Описание разработанного программного средства

Решение задачи реализовано в виде программы, написанной на языке C#. Интерфейс, через который осуществляется работа с ней, имеет следующий вид:



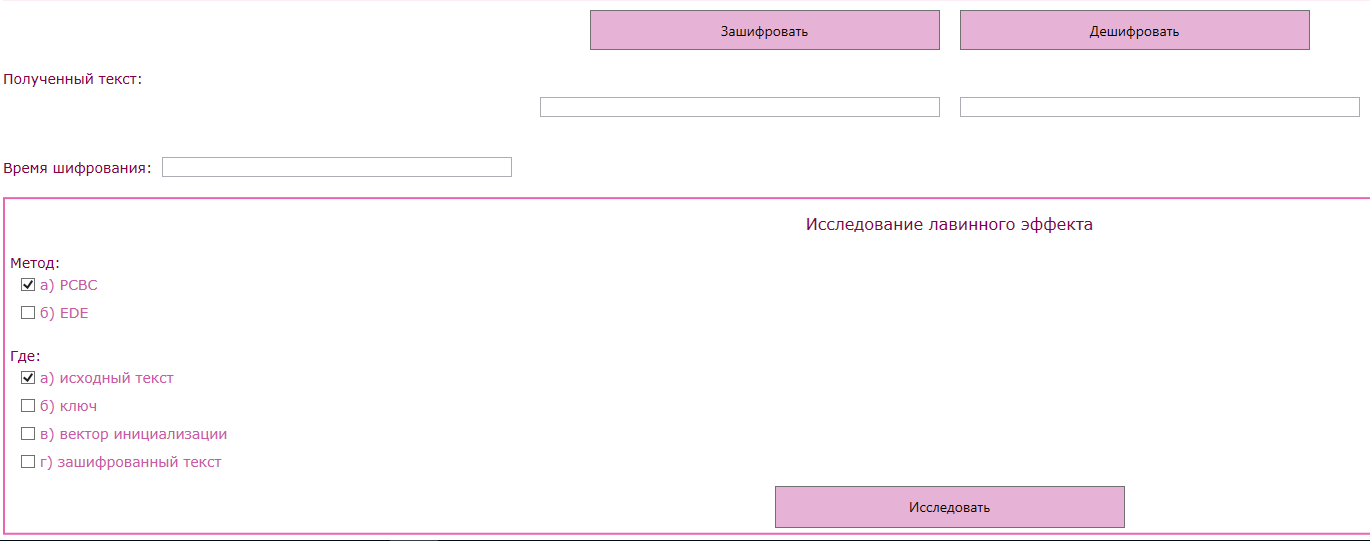


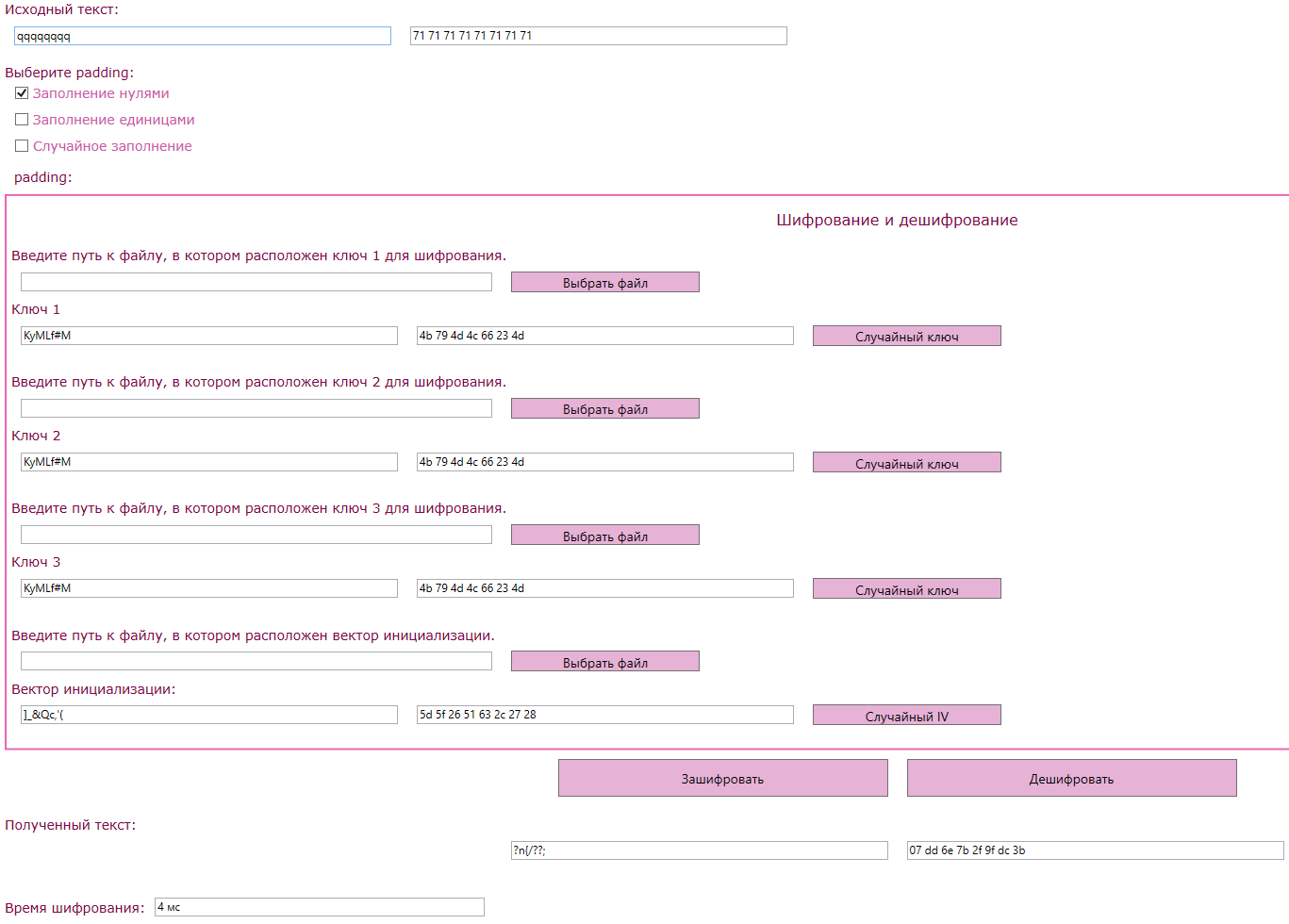
Рис. 1 – Интерфейс реализованного приложения

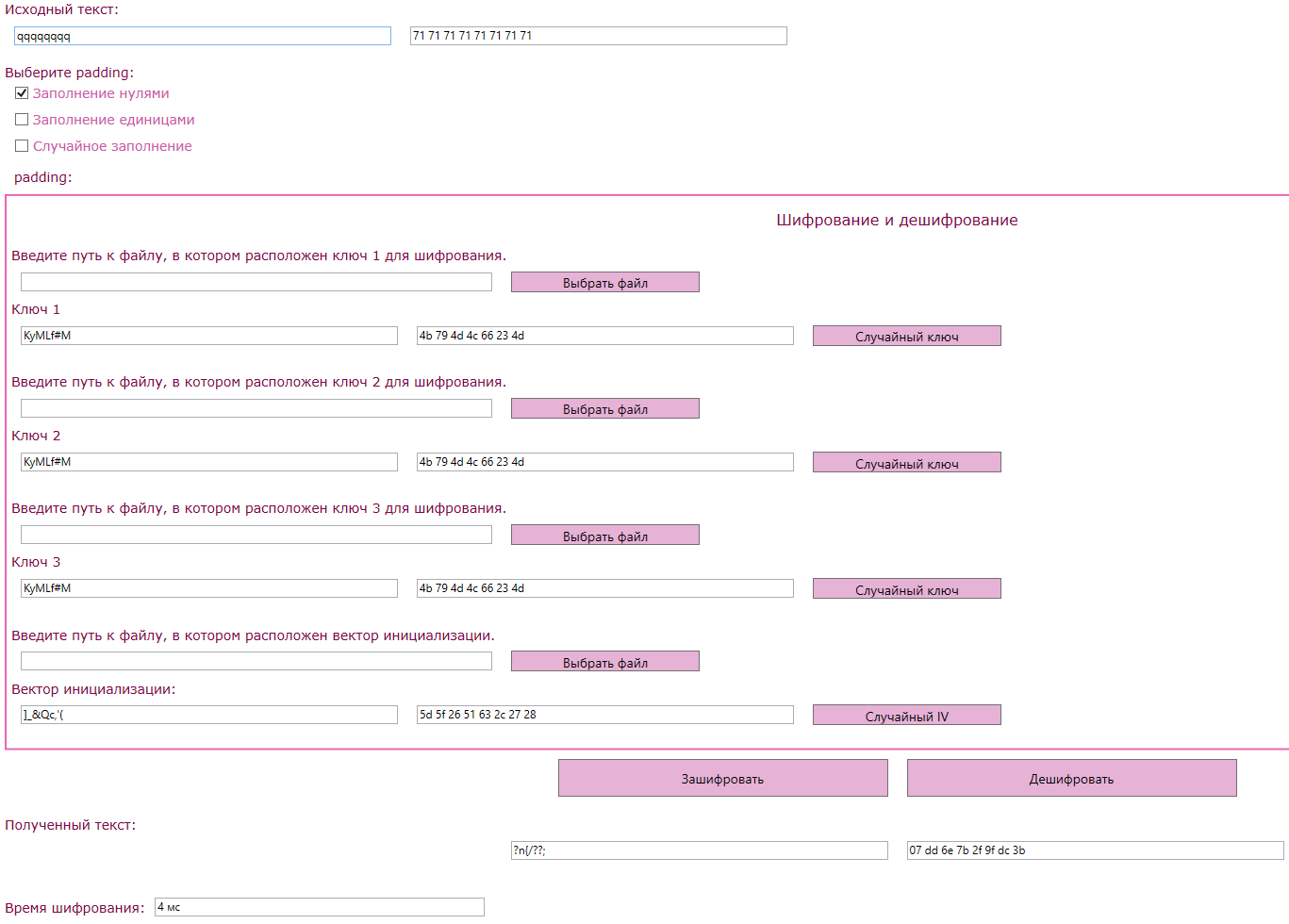
1. **Тестирование**

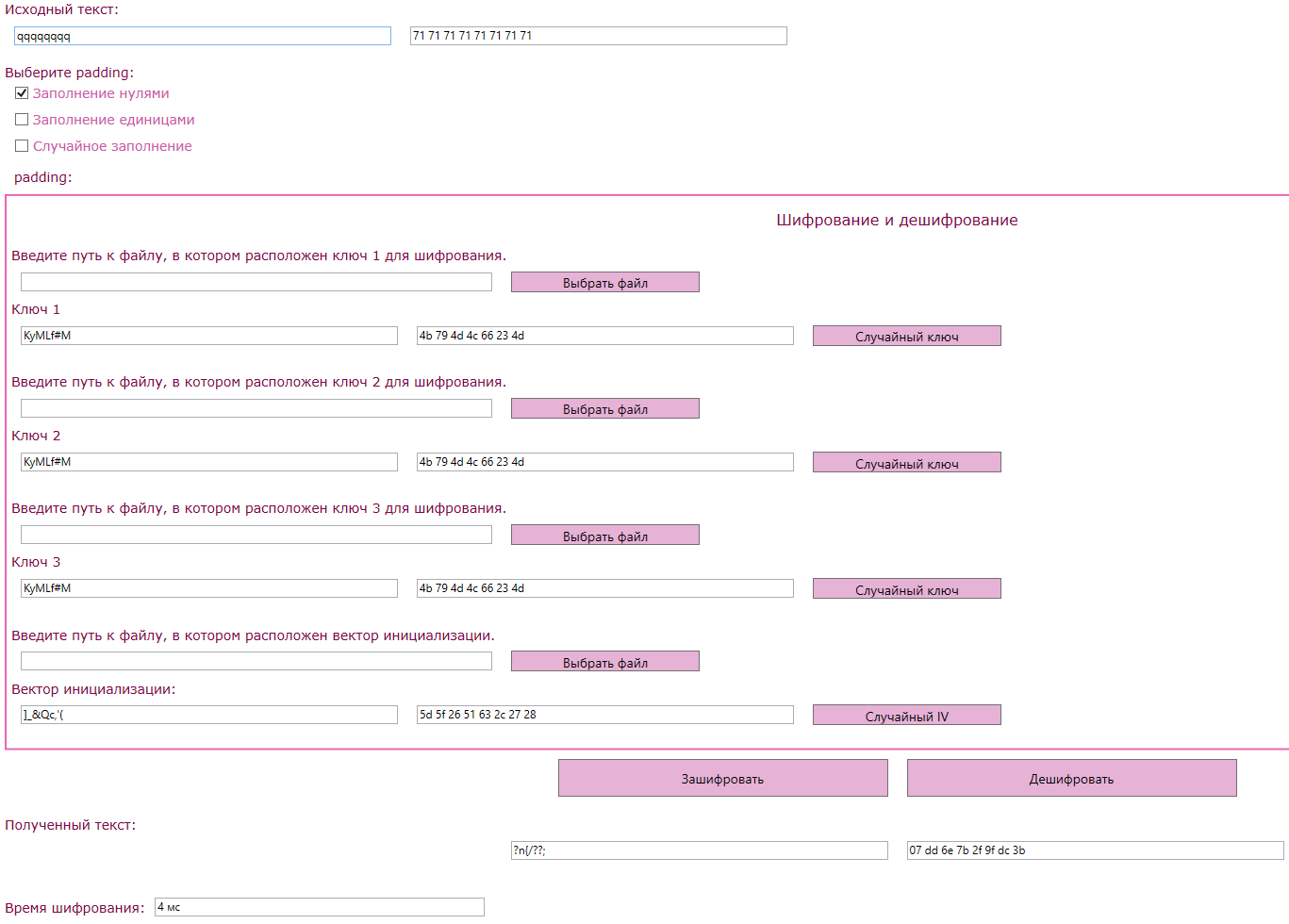
* Тест 1. Шифрование и дешифрование с одним ключом

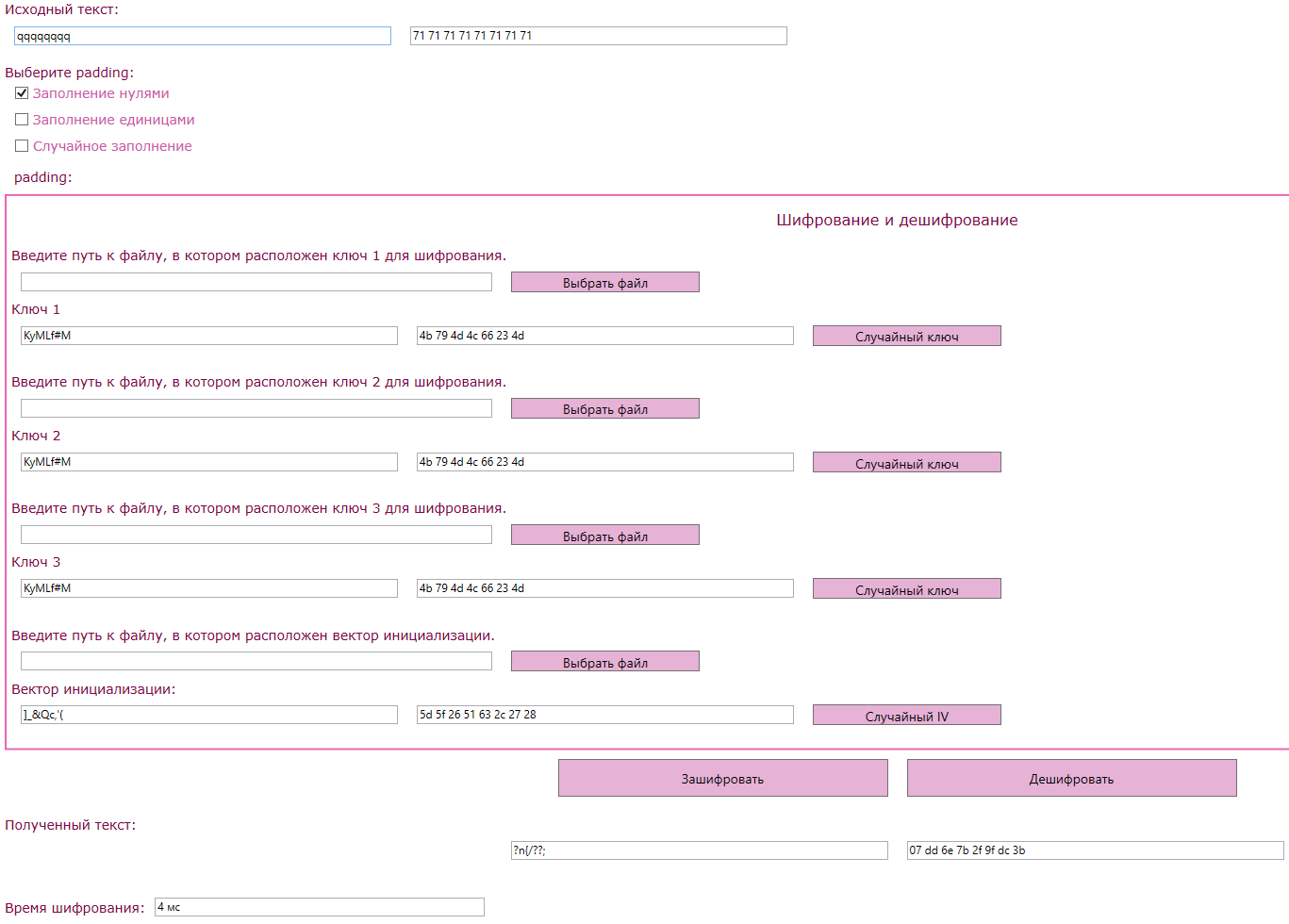
Выберем текст из файла – 8 символов ‘q’. Сгенерируем случайный ключ 1 и скопируем его значение в остальные ключи. Случайным образом сгенерируем вектор инициализации.

Полученная схема шифрования:









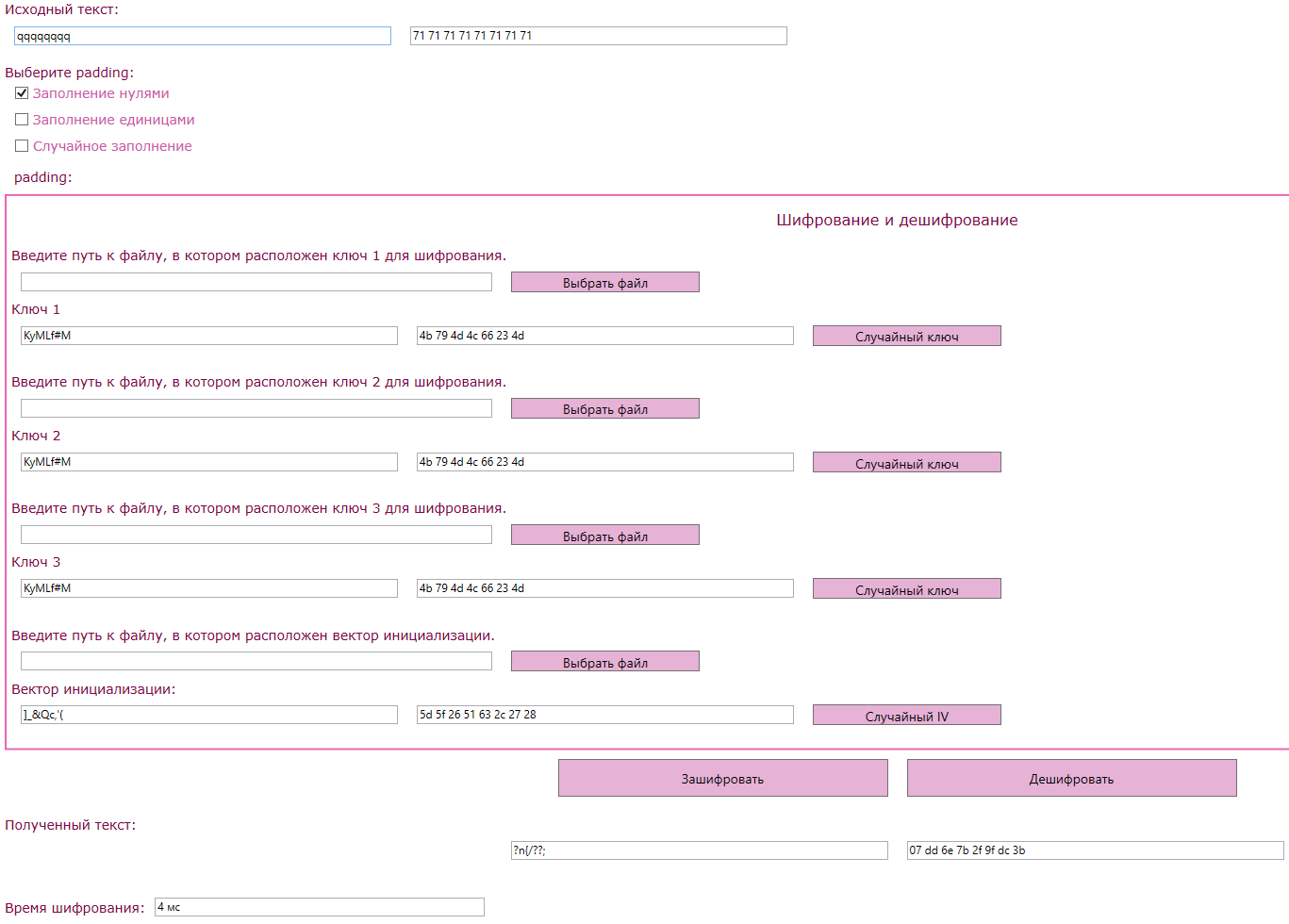


Рис. 2 – Шифрование текста с использованием одного ключа

Выберем в качестве исходного текста полученные зашифрованный текст и дешифруем его.







Рис. 3 – Дешифрование текста с использованием одного ключа

* Тест 2. Шифрование и дешифрование с двумя ключами

Случайным образом сгенерируем второй ключ. Перейдем к следующей схеме шифрования:

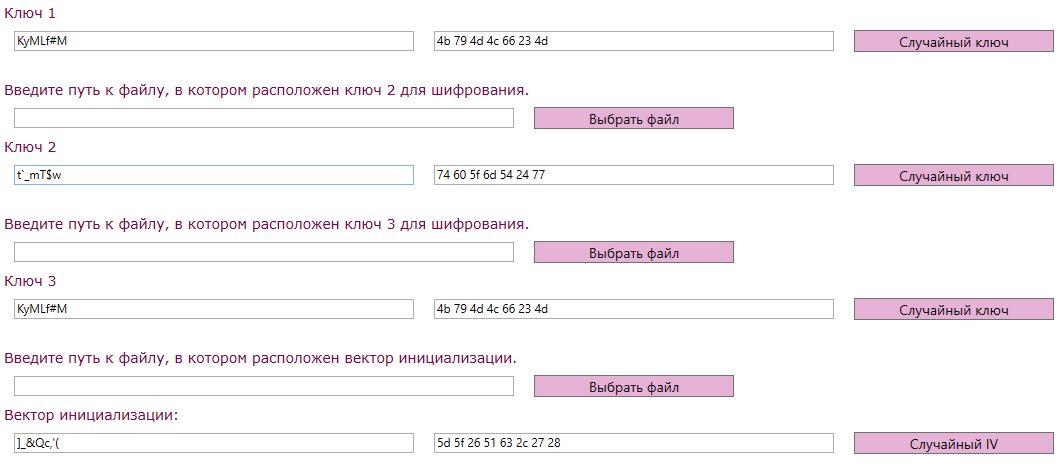








Рис. 4 –Шифрование текста с использованием двух ключей

При дешифровании успешно получен открытый текст.

* Тест 3. Шифрование и дешифрование с тремя ключами

Случайным образом сгенерируем третий ключ. Перейдем к следующей схеме шифрования:







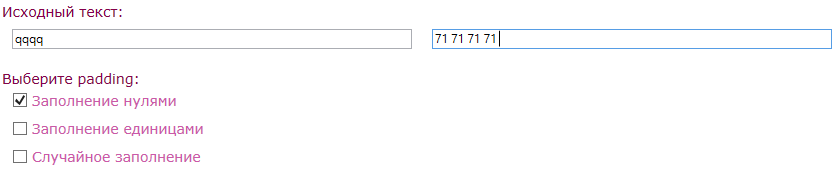


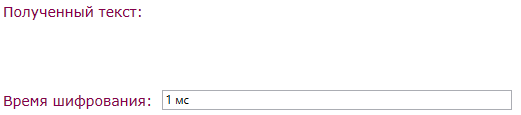
Рис. 5 –Шифрование текста с использованием трех ключей

При дешифровании успешно получен открытый текст.

* Тест 4. Шифрование и дешифрование с нулевым padding

Уменьшим количество символов в открытом тексте. Выберем заполнение padding нулями.







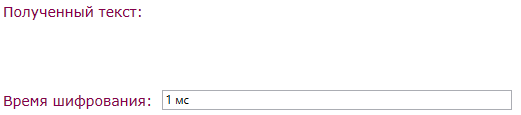


Рис. 6 –Шифрование текста с использованием нулевого padding

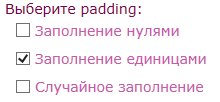
При дешифровании получаем следующий результат:



Рис. 7 –Дешифрование текста с использованием нулевого padding

* Тест 5. Шифрование и дешифрование с единичным padding

Выберем заполнение padding единицами.



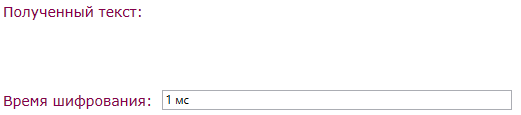




Рис. 8 –Шифрование текста с использованием единичного padding

При дешифровании получаем следующий результат:



Рис. 9 – Дешифрование текста с использованием единичного padding

* Тест 6. Шифрование и дешифрование со случайным padding

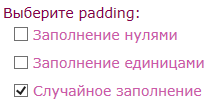






Рис. 10 –Шифрование текста с использованием случайного padding



Рис. 11 – Дешифрование текста с использованием случайного padding

* Тест 7. Шифрование и дешифрование с нулевым вектором инициализации



Рис. 12 –Шифрование текста с использованием нулевого вектора инициализации



Рис. 13 –Дешифрование текста с использованием нулевого вектора инициализации

* Тест 9. Исследование лавинного эффекта для PCBC

Возьмем исходный текст, состоящий из трех блоков:



* + Тест 9.1 для исходного текста

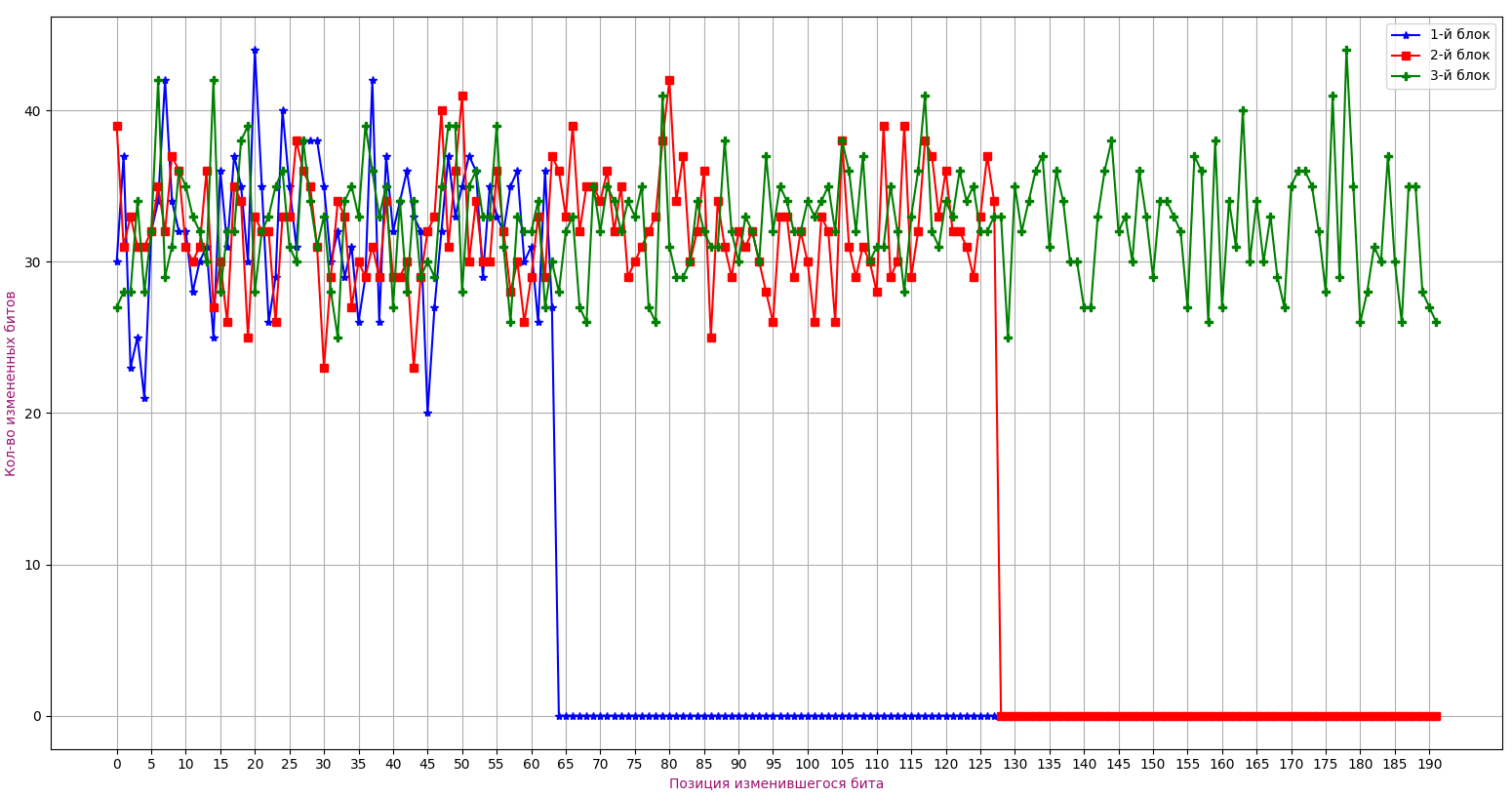


Рис. 14 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в открытом тексте (PCBC)

* + Тест 9.2 для ключа

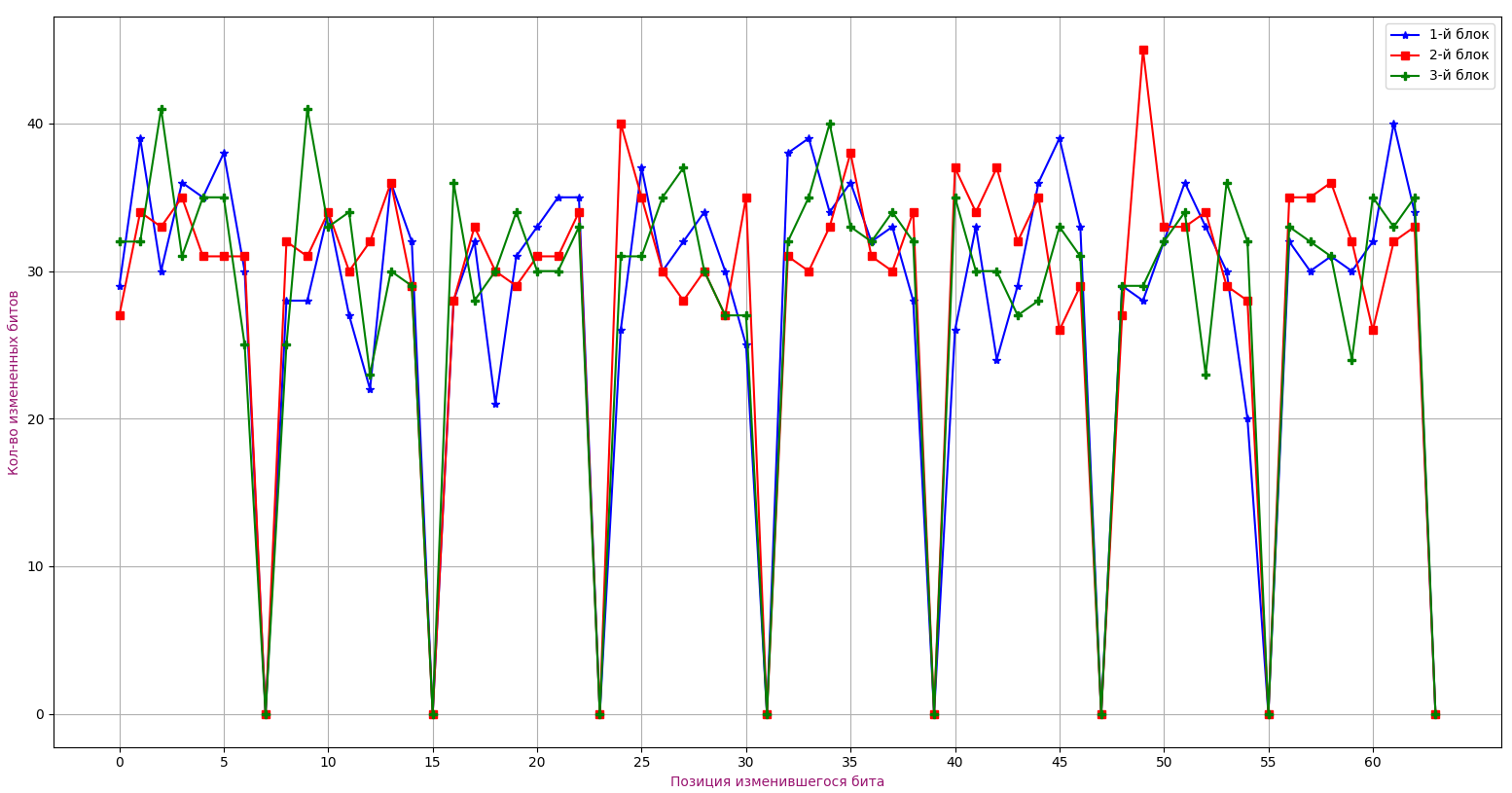


Рис. 15 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в ключе (PCBC)

* + Тест 9.3 для вектора инициализации

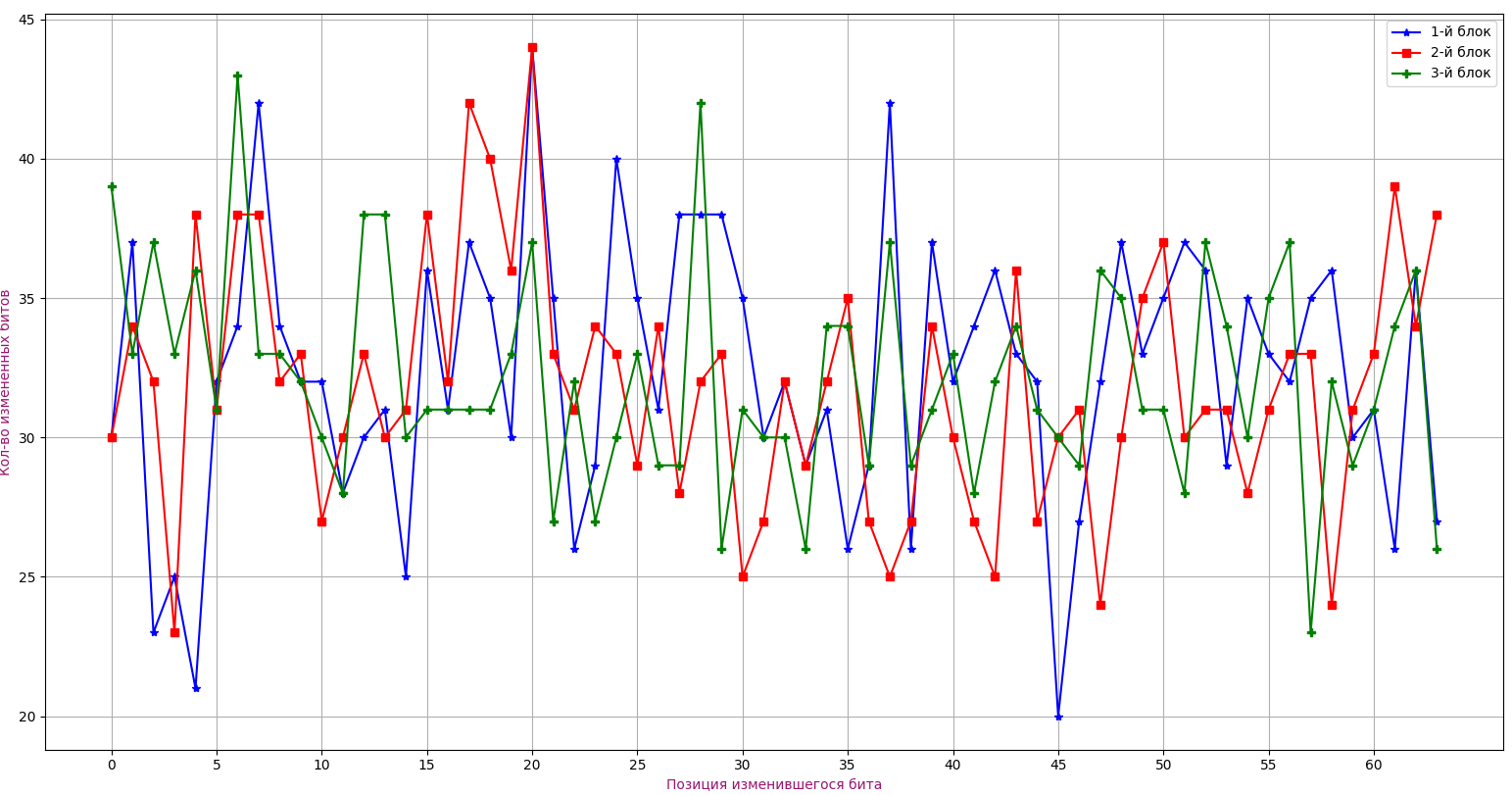


Рис. 16 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в векторе инициализации (PCBC)

* + Тест 9.4 для зашифрованного текста

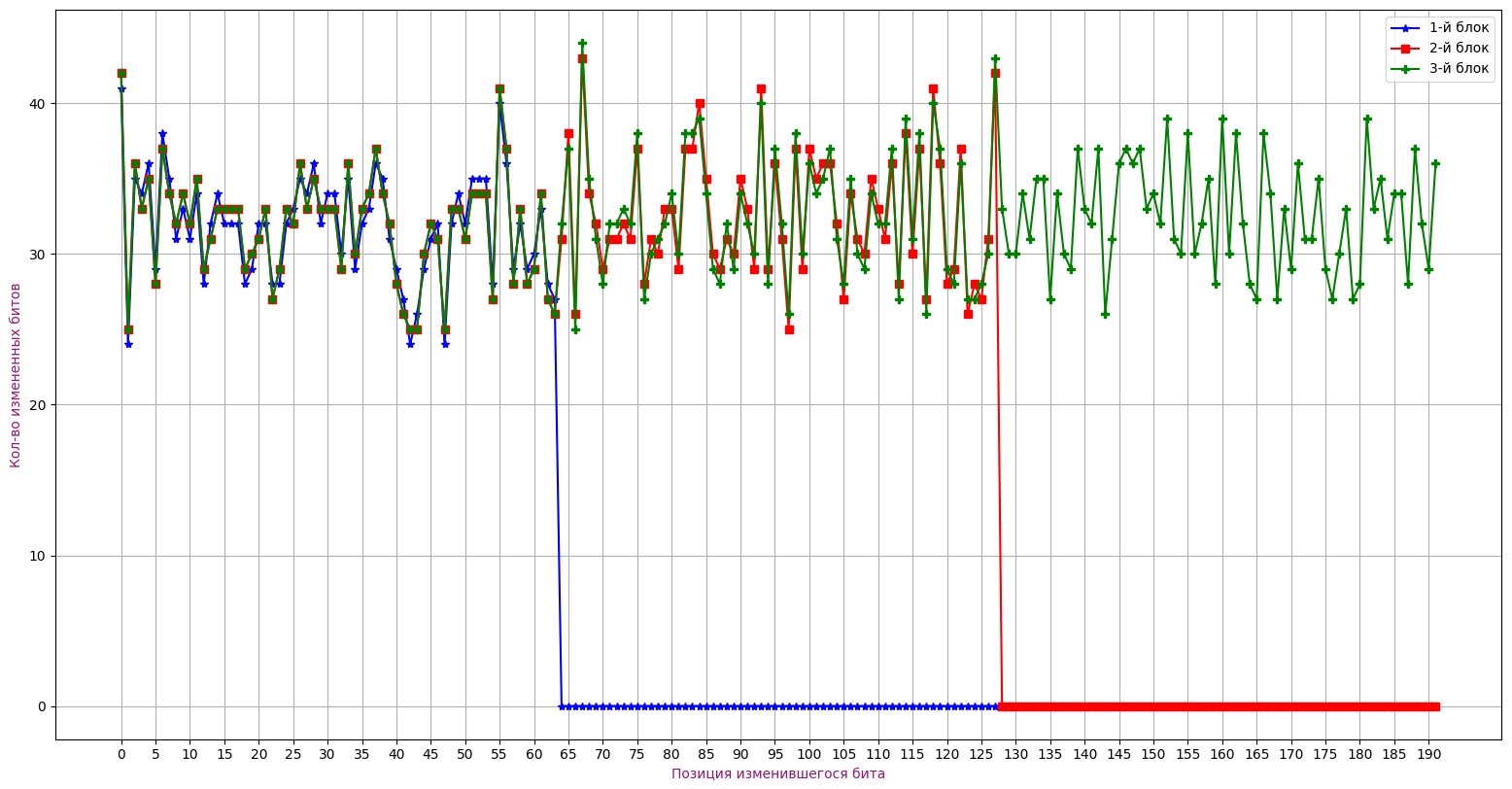
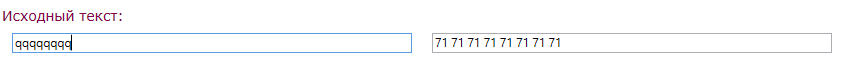


Рис. 17 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в зашифрованном тексте(PCBC)

* Тест 10. Исследование лавинного эффекта для EDE

Возьмем исходный текст, состоящий из одного блока:



* + Тест 9.1 для исходного текста

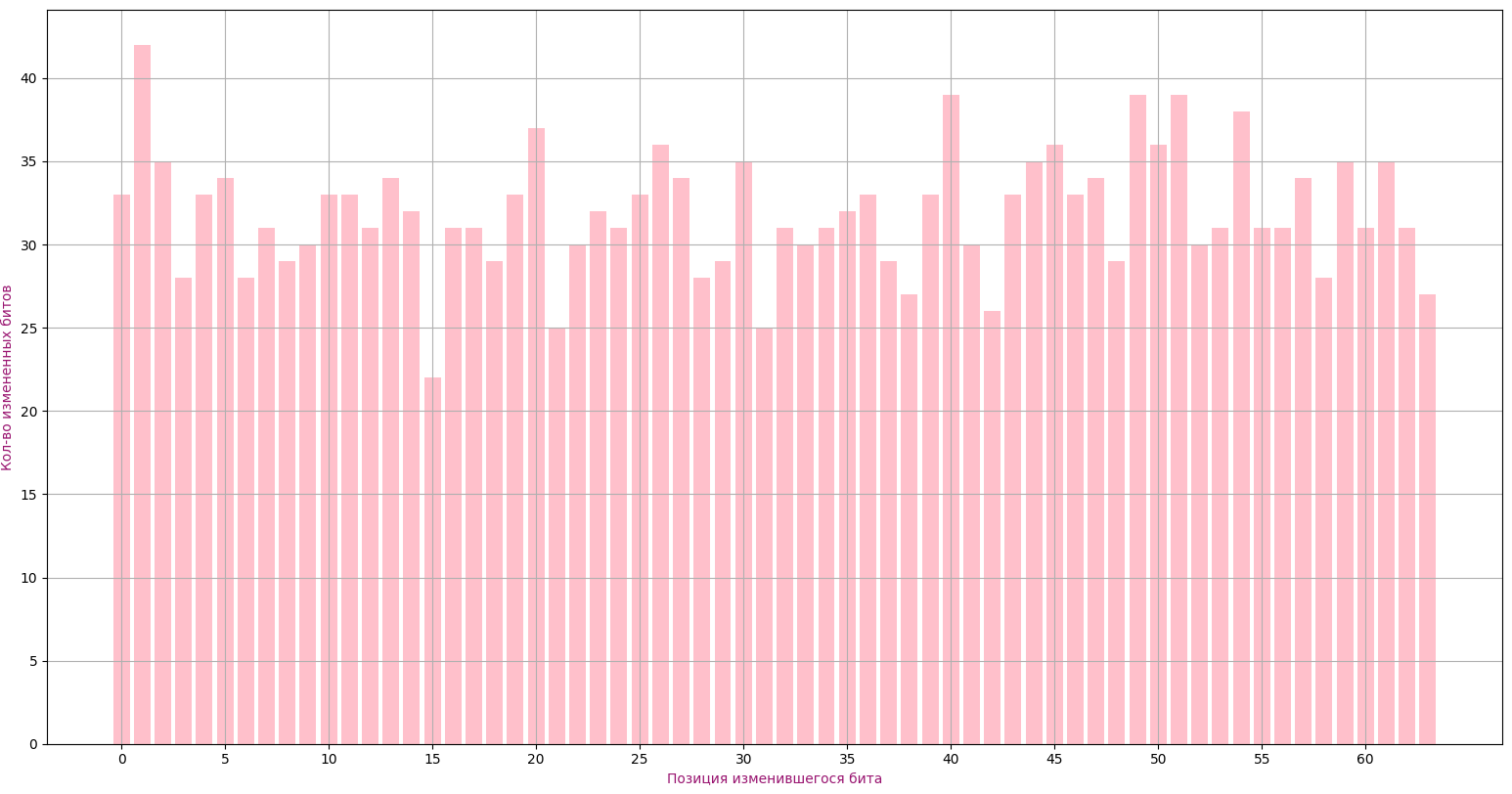


Рис. 18 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в открытом тексте(EDE)

* + Тест 9.2 для ключа

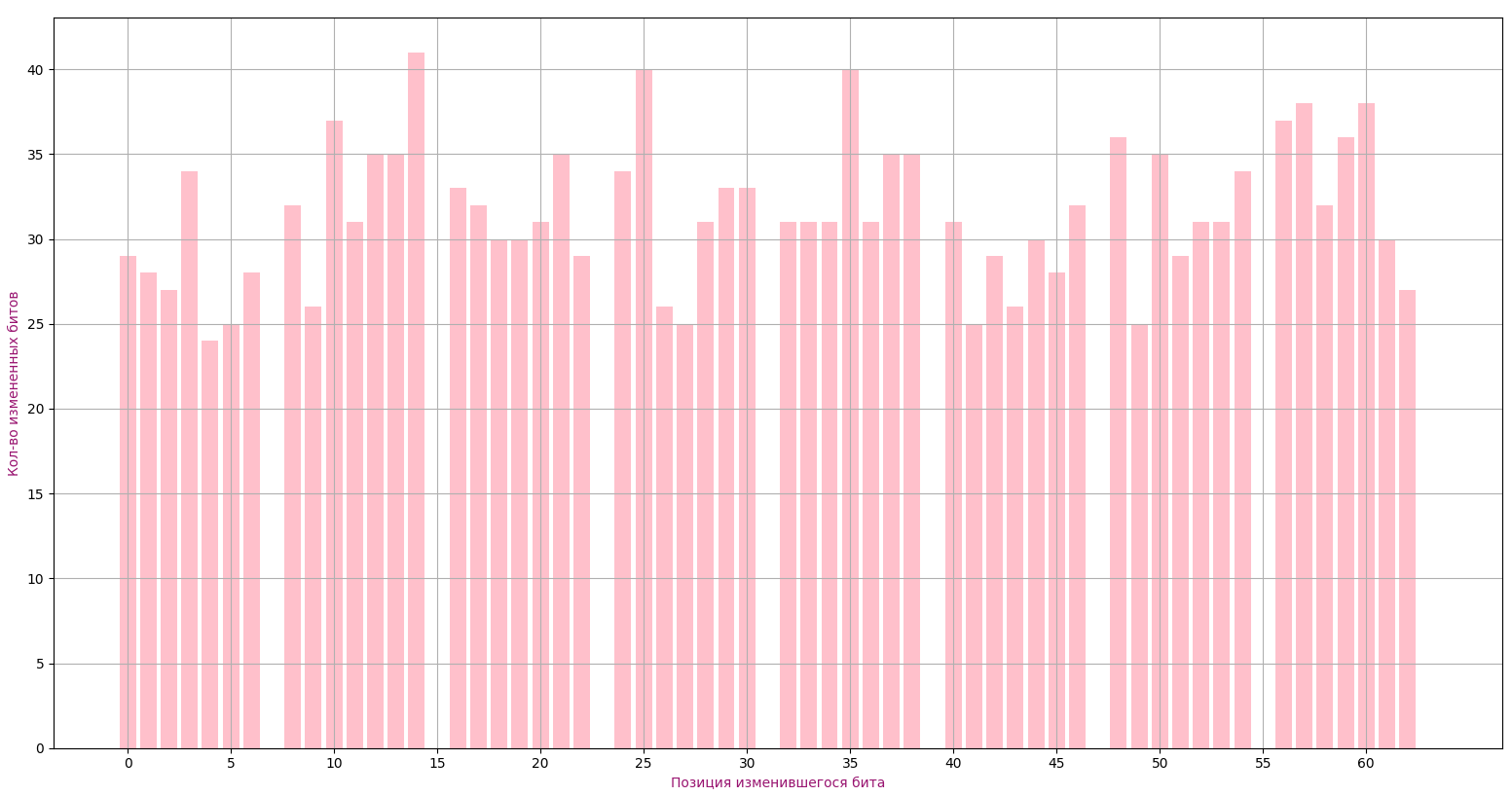


Рис. 19 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в ключе(EDE)

* + Тест 9.3 для зашифрованного текста

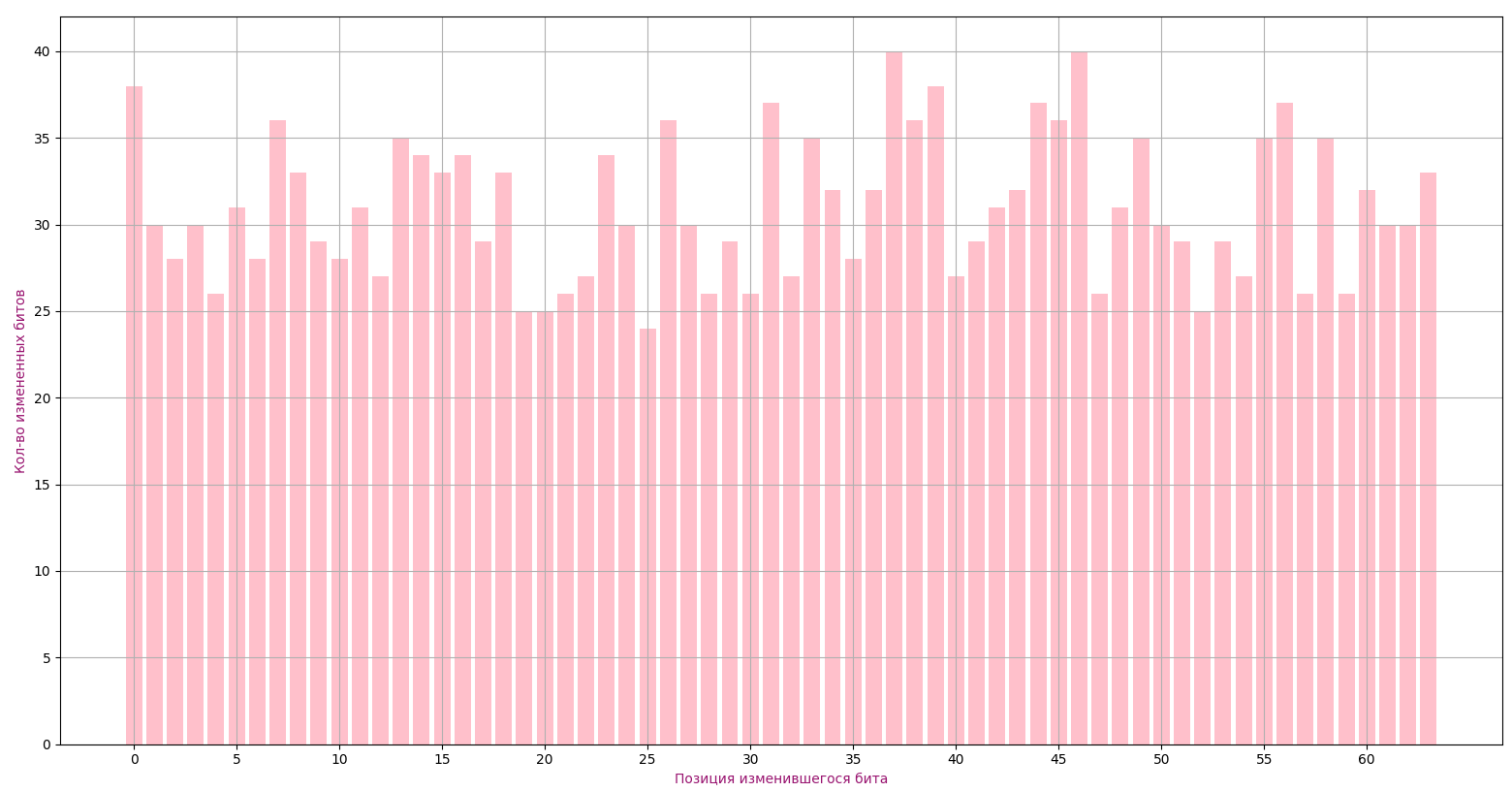


Рис. 20 – Зависимость количества изменившихся бит в зашифрованном тексте при изменении бита в зашифрованном тексте(EDE)

1. **Вывод**

В лабораторной работе мы реализовали приложение, позволяющее шифровать и дешифровать информацию алгоритмом DES с использованием схемы тройного шифрования EDE и режимом шифрования PCBC. Был реализован и проверен лавинный эффект.