ЛАБОРАТОРНАЯ РАБОТА №7

СТАНДАРТ СИММЕТРИЧНОГО ШИФРОВАНИЯ AES (ADVANCED ENCRYPTION STANDARD)

ЦЕЛЬ РАБОТЫ

Изучение принципов и механизмов шифрования, используемых в алгоритме симметричного блочного шифрования AES RIJNDAEL. Практическое освоение пошагового процесса шифрования через интерактивную визуализацию. Исследование различных режимов работы блочных шифров (ECB, CBC, CTR, GCM) и анализ их криптографических свойств.

1. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1.1. Подготовка к работе

Для выполнения лабораторной работы необходимо:

1. Скачать и запустить программу **MIREA AES Visualizer**
2. Убедиться, что программа корректно запустилась и отображается главное окно
3. Ознакомиться с тремя основными вкладками:
    - **Шифрование** — пошаговая визуализация процесса
    - **Теория AES** — теоретические основы алгоритма
    - **Режимы AES** — описание режимов работы

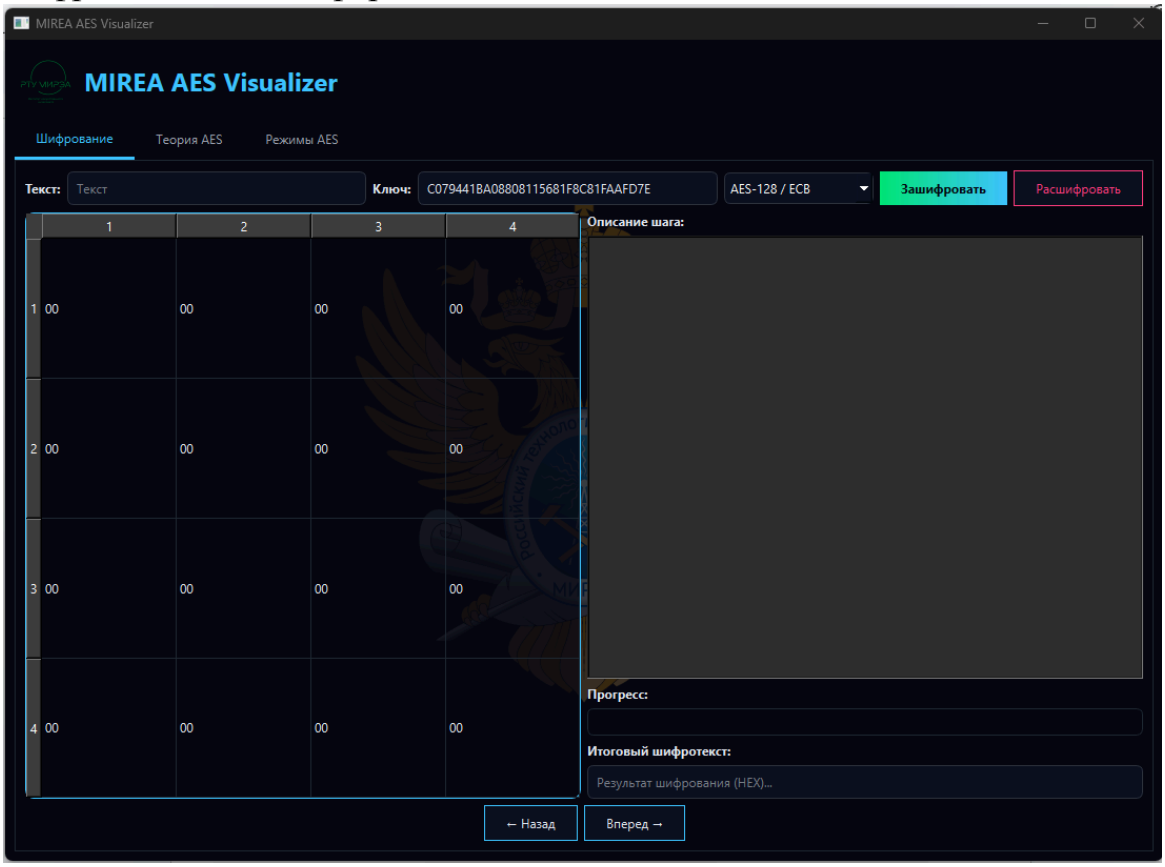
Важно! Программа автоматически не сохраняет данные при закрытии. Рекомендуется заранее подготовить текстовый документ для записи промежуточных результатов.

1.2. Описание интерфейса программы

Главное окно программы содержит следующие элементы (см. Рис. 1.1):

- **Поле ввода текста** — для ввода открытого текста (plaintext), подлежащего шифрованию
- **Поле ввода ключа** — для ввода 128-битного ключа в шестнадцатеричном формате (32 символа)

- **Выпадающее меню выбора режима** — для выбора режима шифрования (AES-128 / ECB)
- **Кнопка "Зашифровать"** (зелёная) — запуск процесса шифрования
- **Кнопка "Расшифровать"** (красная) — запуск процесса расшифрования
- **Матрица состояния 4×4** — визуализация текущего состояния данных в виде матрицы байтов
- **Кнопки навигации "← Назад" и "Вперёд →"** — пошаговый просмотр трансформаций
- **Прогресс-бар** — отображение текущего шага процесса
- **Текстовая область "Описание шага"** — пояснение текущей выполняемой операции
- **Текстовая область "Итоговый шифротекст"** — результат шифрования в HEX-формате



1.1: Главное окно программы MIREA AES Visualizer

2. ПРАКТИЧЕСКАЯ ЧАСТЬ: ВИЗУАЛИЗАЦИЯ ПРОЦЕССА ШИФРОВАНИЯ

2.1. Подготовка исходных данных

Шаг 1: Ввод текста

1. Нажмите на поле **"Текст"** и введите произвольный текст (например, слово **"Текст"** или любое 16-байтовое значение)
2. Текст должен быть кодируемым в ASCII или содержать только символы, поддерживаемые кодировкой UTF-8

Примеры для экспериментов:

- Простой текст: "MIREA2024"
- Техническое слово: "CRYPTOGRAPHY"
- Русский текст: "ШИФРОВАНИЕ" (будет преобразовано в HEX)

Шаг 2: Ввод ключа шифрования

1. Нажмите на поле **"Ключ"** и введите 128-битный ключ в шестнадцатеричном формате
2. Ключ должен содержать ровно **32 шестнадцатеричных символа** (128 бит)

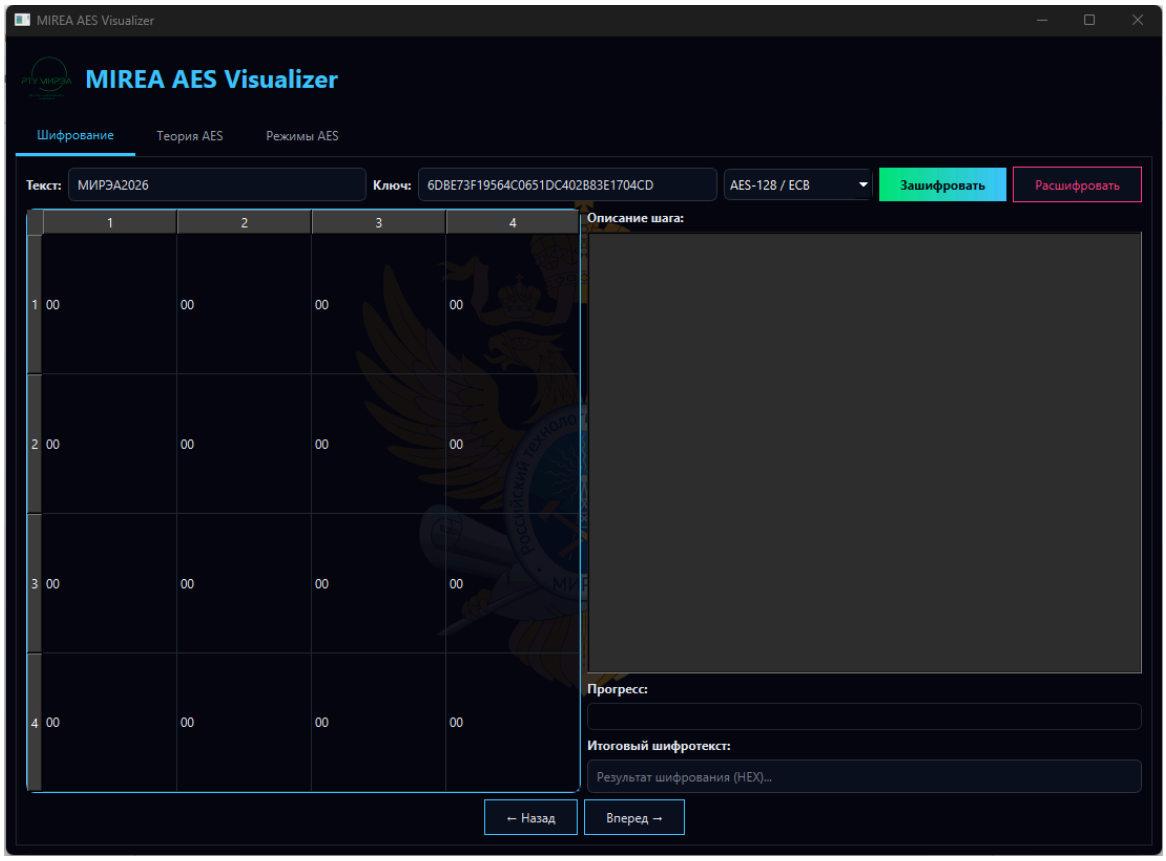
Примеры валидных ключей:

- A18D4984C83A3D50696C855209A77A00
- 01EB27CD9C76D4F0E95F85C8BC0A6895
- C079441BA08808115681F8C81FAAFD7E

Если вы не уверены, используйте предложенный ключ.

Шаг 3: Выбор режима

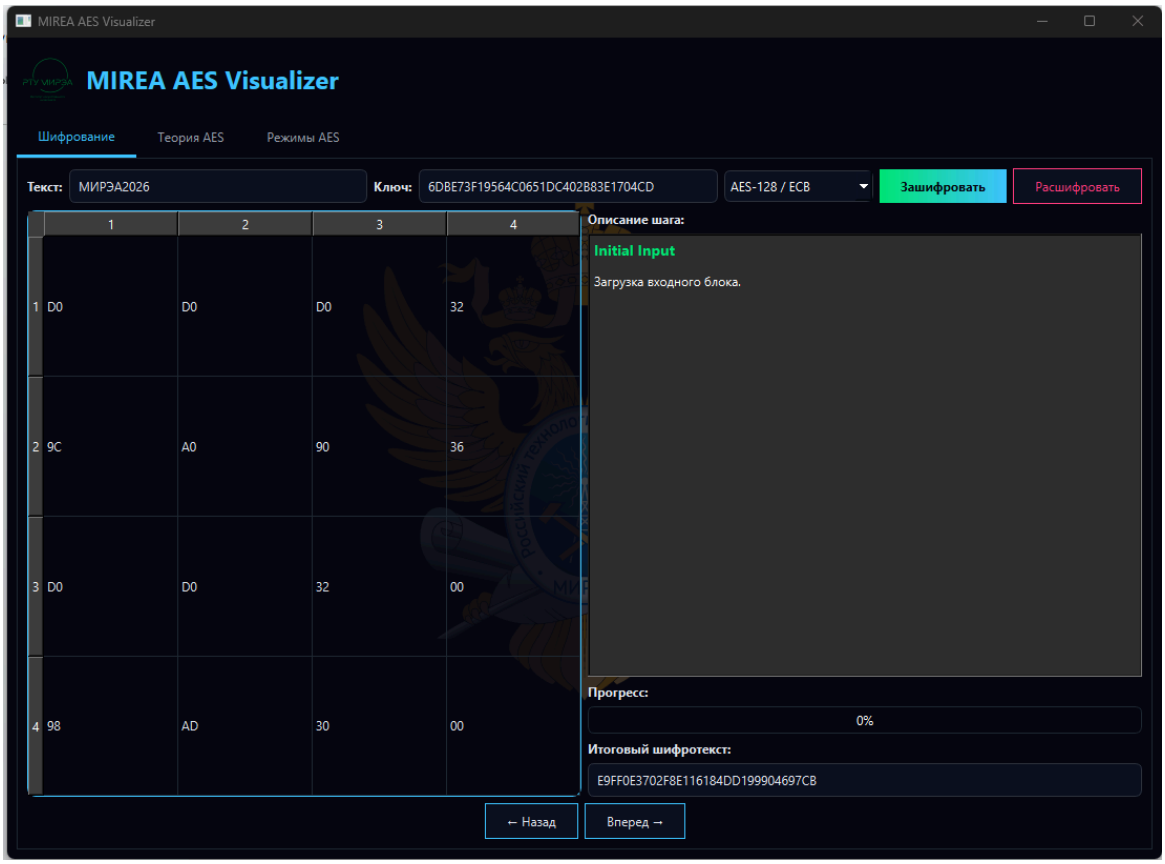
1. Нажмите на выпадающее меню выбора режима
2. Выберите **"AES-128 / ECB"** для базового режима (электронная кодовая книга)



2.1: Окно с заполненными полями ввода текста и ключа

2.2. Запуск процесса шифрования

1. Убедитесь, что оба поля (текст и ключ) заполнены корректно
2. Нажмите зелёную кнопку **"Зашифровать"**
3. Программа разобьёт процесс на пошаговые операции и отобразит начальное состояние



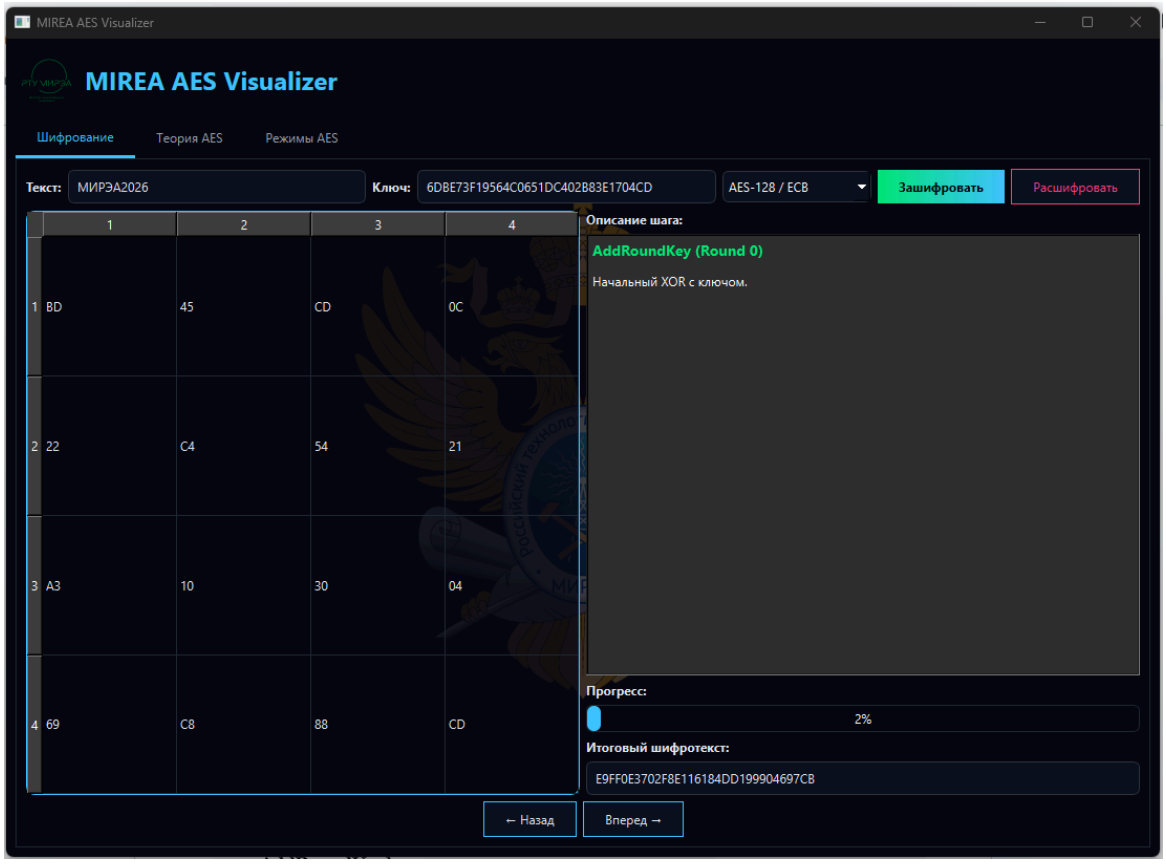
2.2: Начальное состояние матрицы после запуска

2.3. Пошаговый анализ шифрования

Используя кнопки **"← Назад"** и **"Вперёд →"**, последовательно изучите каждый этап. Ниже описаны ключевые этапы для AES-128:

Этап 0: Начальное AddRoundKey

- **Описание**: Текст преобразуется в матрицу 4×4 и выполняется операция XOR с начальным ключом
- **Математика**: $\text{State XOR Key}_0 = \text{InitialState}$
- **Значение**: Эта операция предварительно смешивает текст с ключом



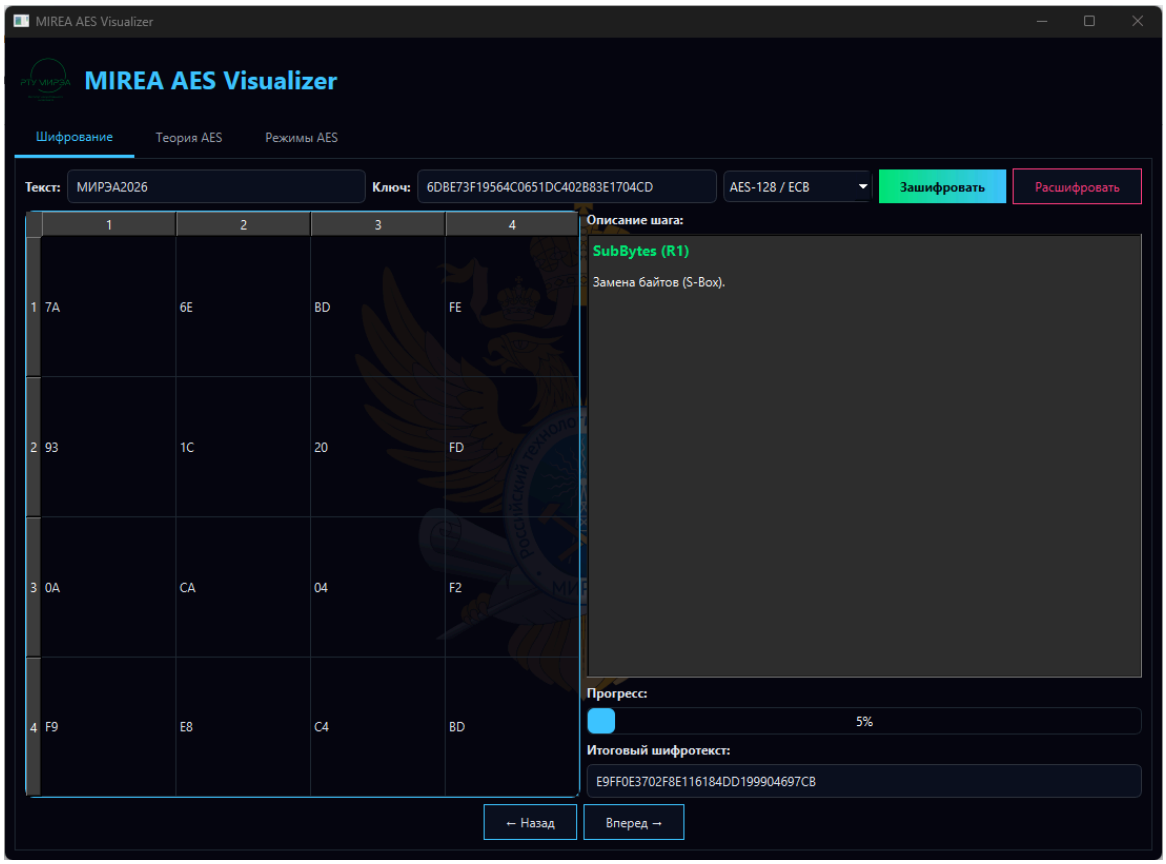
2.3: Матрица после начального AddRoundKey

Раунды 1–9: Полные раунды

Каждый раунд состоит из четырёх операций:

1. SubBytes (Побайтовая подстановка)

- **Описание**: Каждый байт матрицы заменяется соответствующим значением из S-Box таблицы
- **Назначение**: Нелинейное преобразование, обеспечивающее "конфузию" (запутанность)
- **Свойство**: Одинаковые входные байты → одинаковые выходные байты

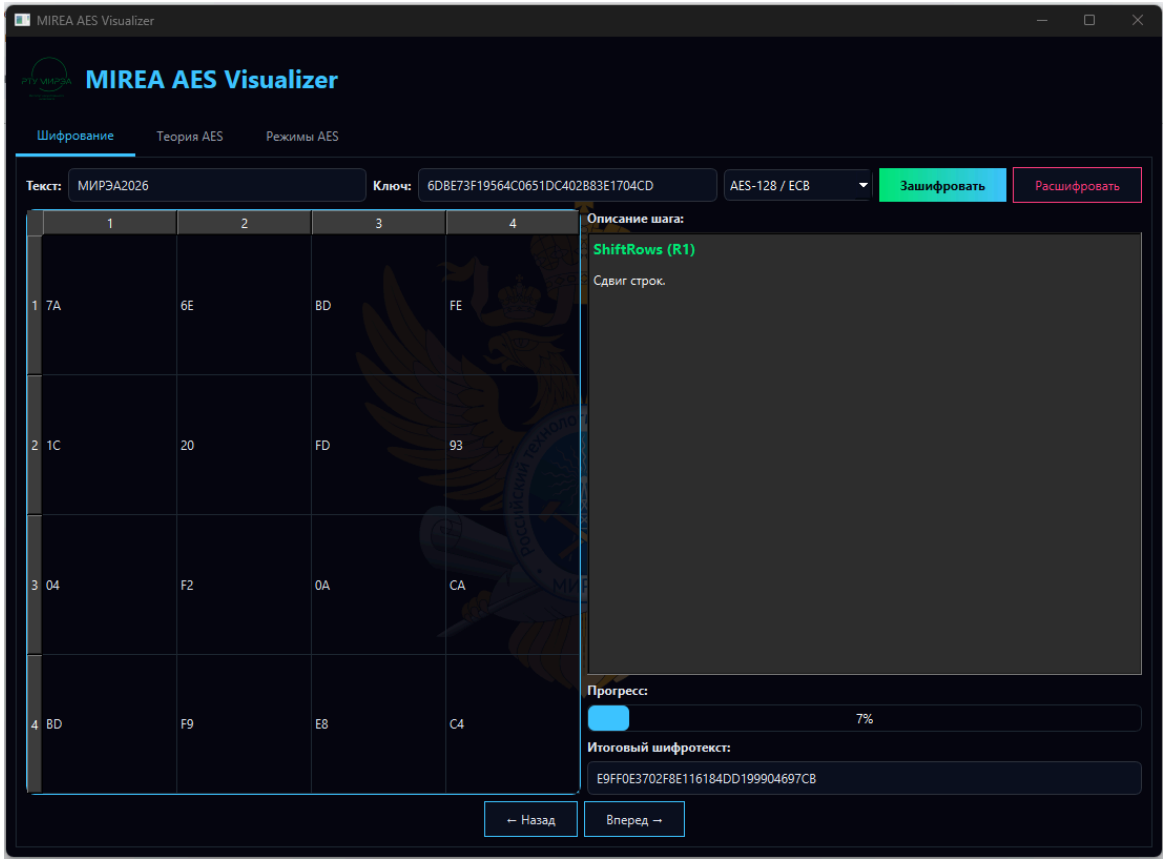


2.4: Матрица после SubBytes первого раунда

2. ShiftRows (Циклический сдвиг строк)

- **Описание**: Строки матрицы циклически сдвигаются:
  - Строка 0: сдвиг на 0 позиций
  - Строка 1: сдвиг на 1 позицию
  - Строка 2: сдвиг на 2 позиции
  - Строка 3: сдвиг на 3 позиции

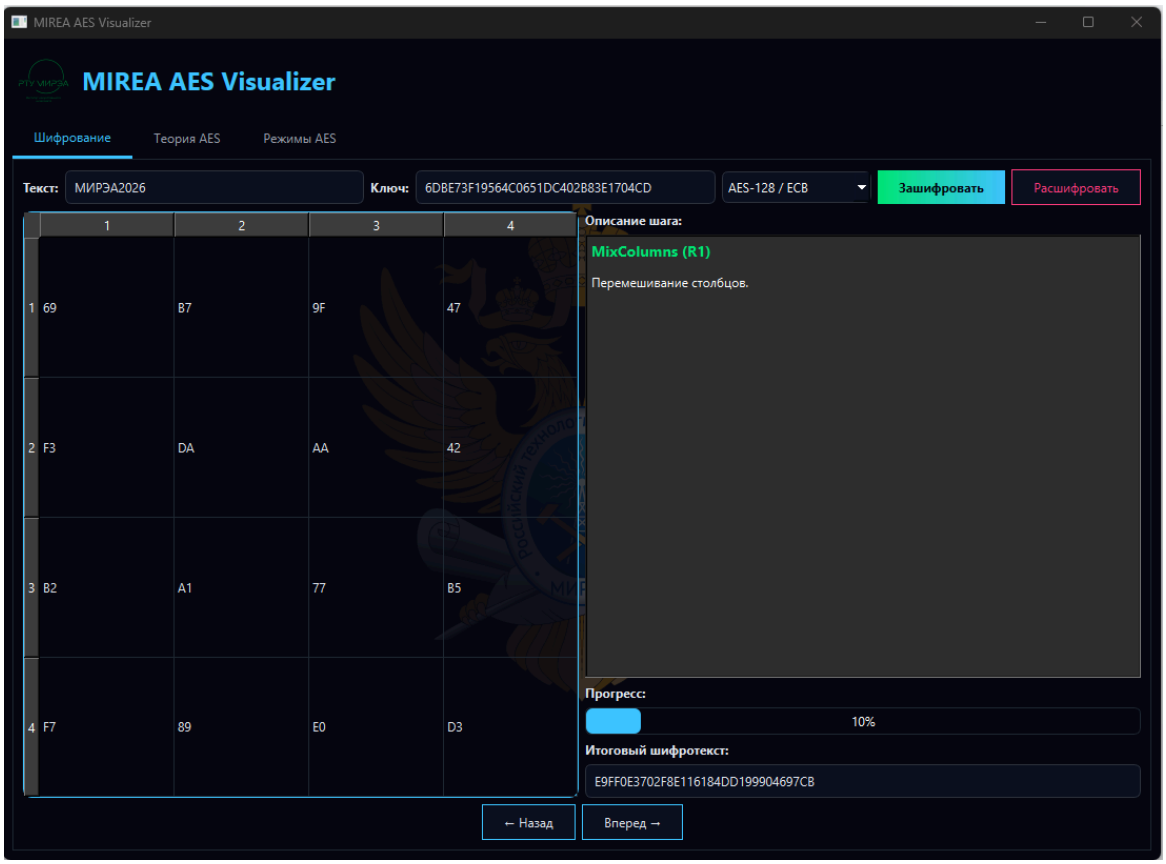
- **Назначение**: Обеспечивает "диффузию" (распространение изменений)



2.5: Матрица после ShiftRows первого раунда

3. MixColumns (Перемешивание столбцов)

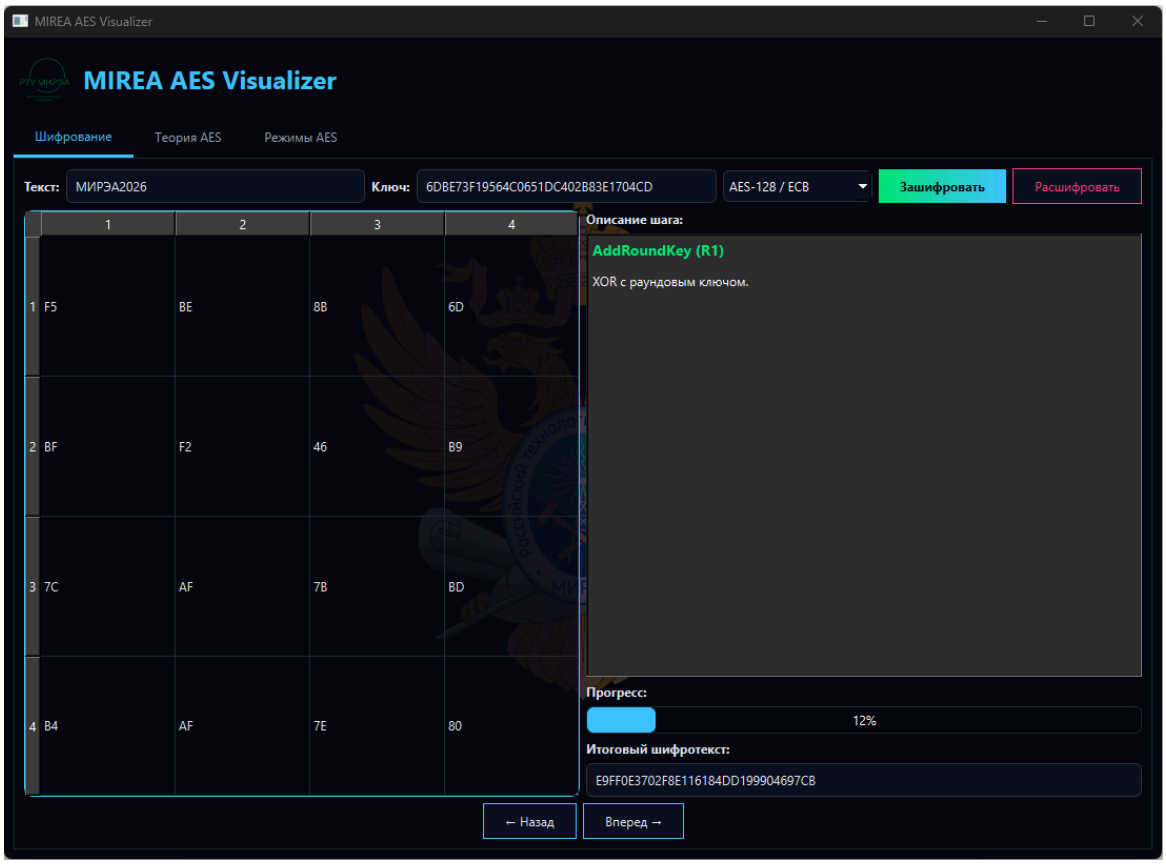
- **Описание**: Каждый столбец матрицы умножается на фиксированную полиномиальную матрицу в поле Галуа GF(2^8)
- **Формула**: Каждый столбец преобразуется независимо через линейное преобразование
- **Эффект**: Каждый выходной байт зависит от всех четырёх входных байтов столбца



2.6: Матрица после MixColumns первого раунда

4. AddRoundKey (Добавление раундового ключа)

- **Описание**: Текущее состояние XOR-ится с i-й частью расширенного ключа
- **Математика**: $\text{State XOR Key}_i = \text{NewState}$
- **Количество**: Выполняется один раз в конце каждого раунда



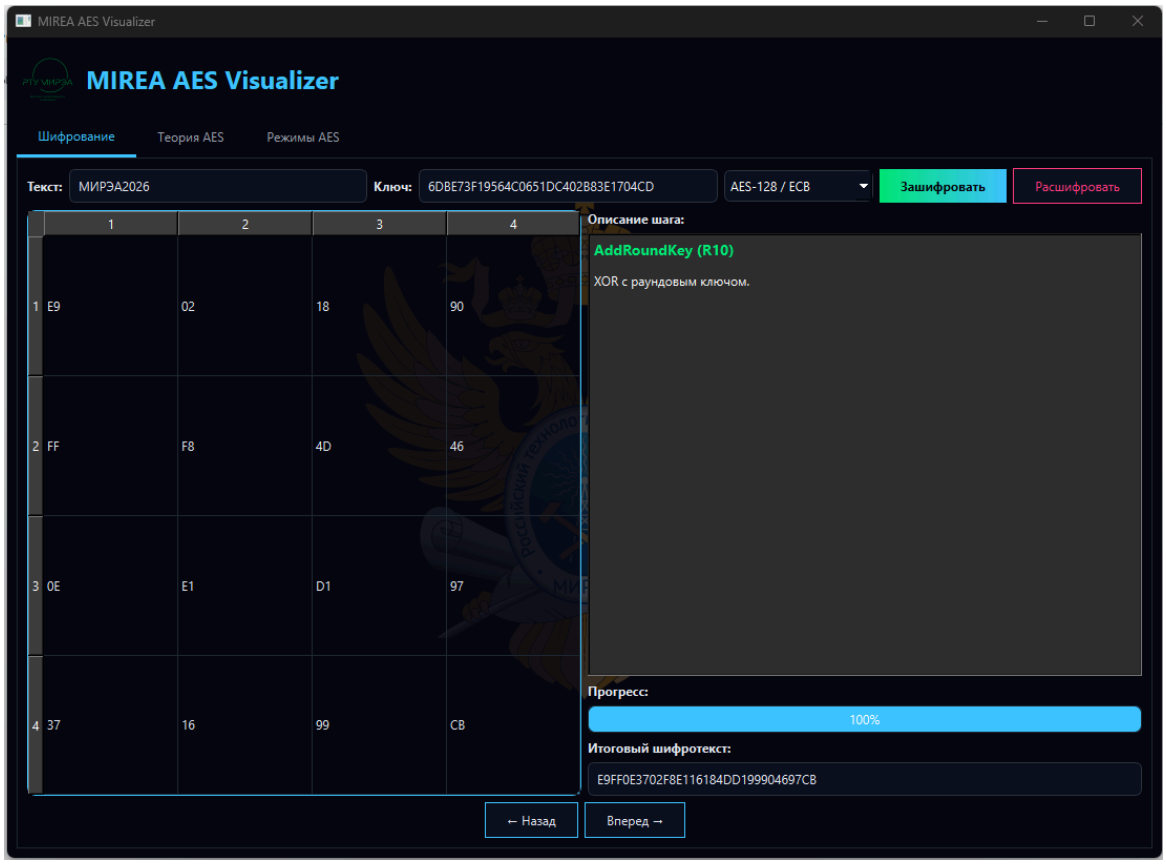
2.7: Матрица после AddRoundKey первого раунда

Раунд 10 (Финальный раунд)

Финальный раунд отличается тем, что **не содержит операцию MixColumns**:

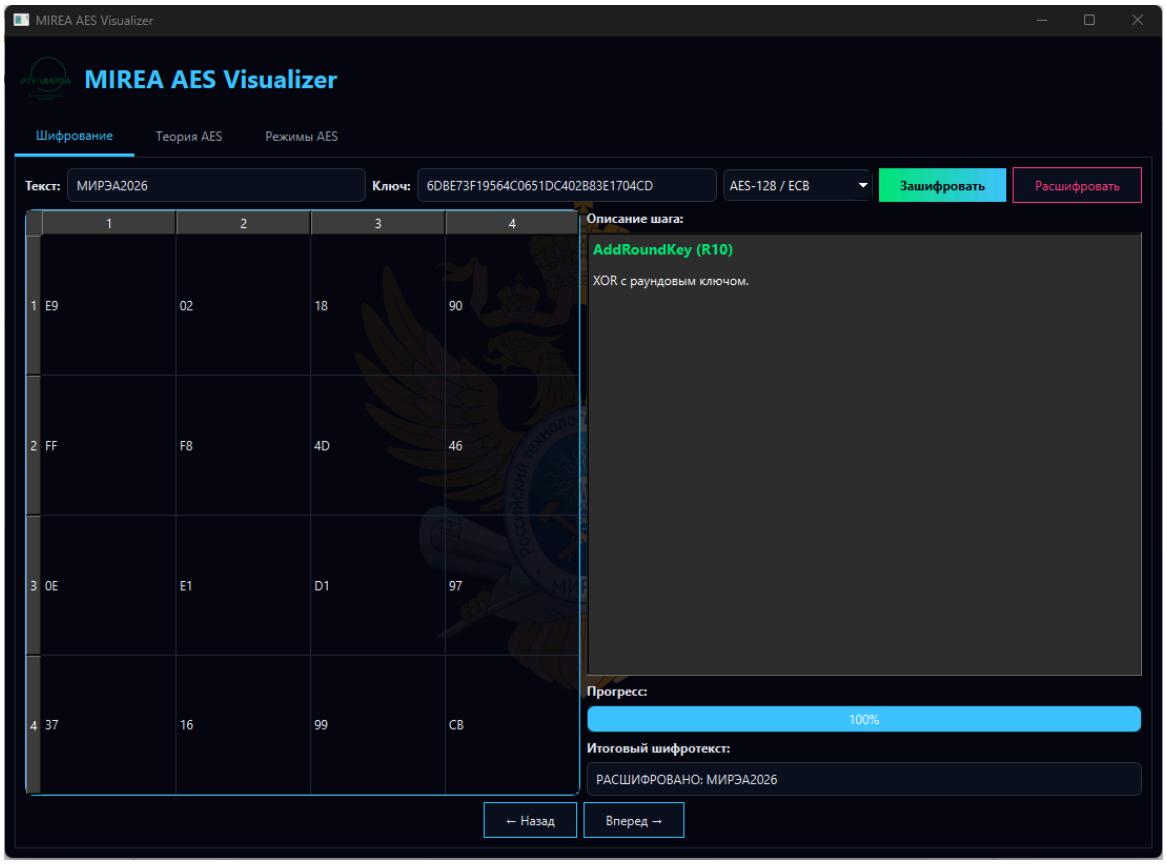
1. **SubBytes**
2. **ShiftRows**
3. **AddRoundKey** (без MixColumns!)

Вопрос для размышления: Почему в финальном раунде отсутствует MixColumns?



2.8: Финальное зашифрованное состояние
Итоговый результат

- **Шифротекст** отображается в текстовой области "Итоговый шифротекст" в шестнадцатеричном формате



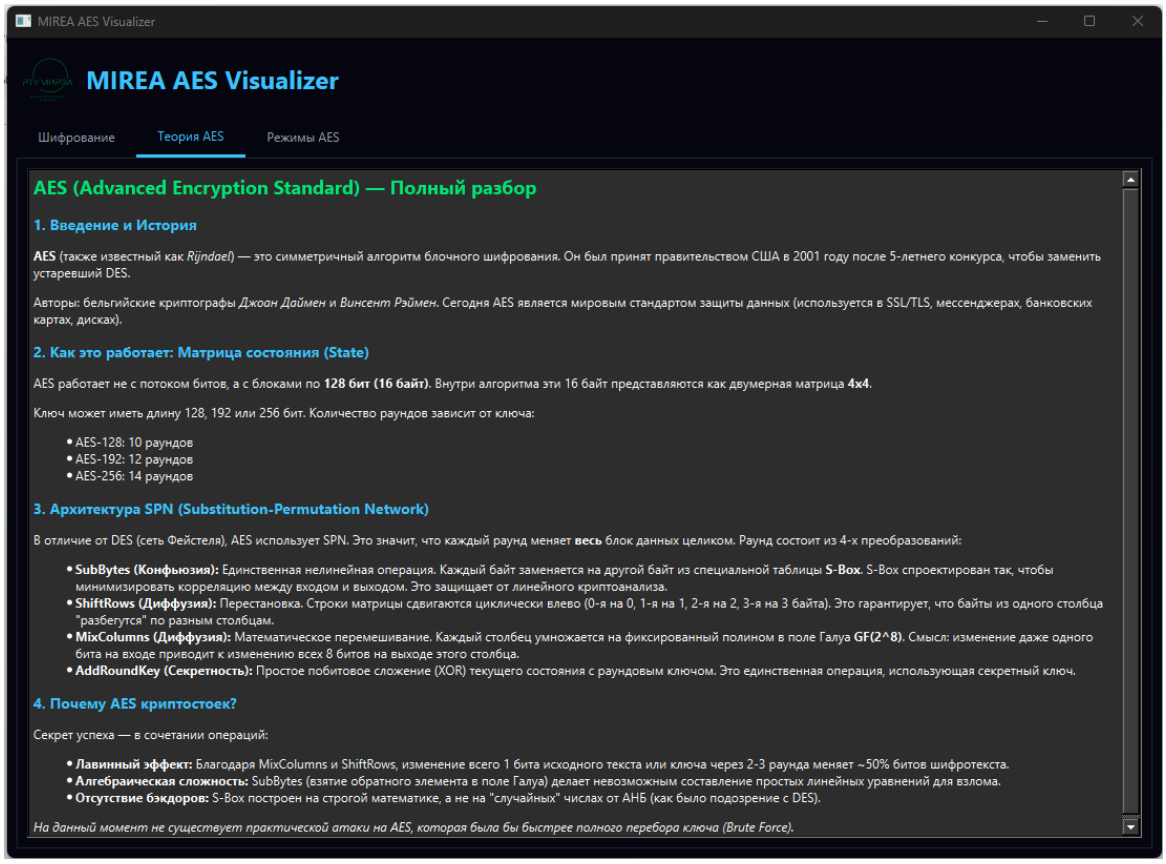
2.9: Расшифровка зашифрованного текста по кнопке

- "Итоговый шифротекст" в шестнадцатеричном формате необходимо расшифровать нажав на кнопку Расшифровать и обязательно прикрепить скриншот.

3. ИЗУЧЕНИЕ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

3.1. Переход на вкладку "Теория AES"

1. Нажмите на вкладку **"Теория AES"** в верхней части окна программы
2. Изучите следующие разделы:



3.1: Вкладка "Теория AES"

3.2. Основные концепции

Введение и история

- **Название**: AES (Advanced Encryption Standard), также известен как Rijndael
- **Авторы**: Бельгийские криптографы Джоан Даймен и Винсент Рэймен
- **Принятие**: Одобрен правительством США в 2001 году после 5-летнего конкурса

- **Стандартизация**: Используется как стандарт в SSL/TLS, VPN, банковских системах

Архитектура SPN

- **Отличие от DES**: AES использует архитектуру SPN (Substitution-Permutation Network), а не сеть Фейстеля
- **Ключевое свойство**: Каждый раунд преобразует весь 128-битный блок целиком
- **Результат**: Более высокая стойкость и параллелизм операций

Параметры AES

Режим	Длина ключа	Количество раундов
AES-128	128 бит	10
AES-192	192 бит	12
AES-256	256 бит	14

3.3. Математические основы

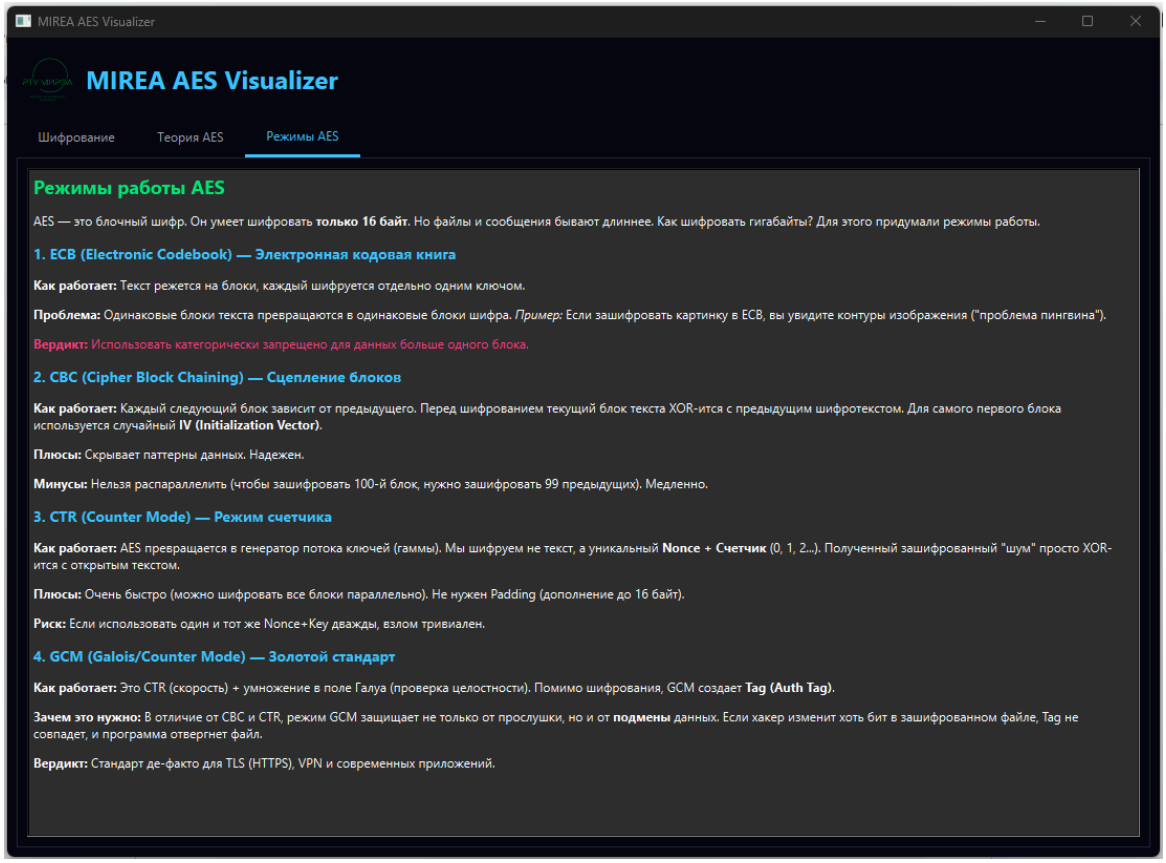
Поле Галуа $GF(2^8)$

- Все операции в MixColumns выполняются в конечном поле
- Каждый байт рассматривается как полином над GF(2)
- Умножение: $(x+1) * (x^2+1) = x^3+x^2+x+1$
- Операции выполняются по модулю неприводимого полинома

4. ИССЛЕДОВАНИЕ РЕЖИМОВ РАБОТЫ

4.1. Переход на вкладку "Режимы AES"

1. Нажмите на вкладку **"Режимы AES"** в верхней части окна программы
2. Изучите описание четырёх основных режимов



4.1: Вкладка "Режимы AES"

4.2. ECB (Electronic Codebook) — Режим электронной кодовой книги

Как работает:

- Текст разбивается на блоки по 16 байт
- Каждый блок шифруется **независимо** одним и тем же ключом

Проблема:

- Одинаковые открытые блоки → одинаковые зашифрованные блоки

- Образуются видимые паттерны в шифротексте

Пример — "проблема пингвина":

- Если зашифровать изображение в режиме ECB, контуры исходного изображения остаются видны в шифротексте

Вердикт: Категорически запрещён для шифрования данных больше одного блока

4.3. CBC (Cipher Block Chaining) — Режим сцепления блоков

Как работает:

- Каждый блок открытого текста XOR-ится с **предыдущим шифрованным блоком**
- Для первого блока используется случайный **IV (Initialization Vector)**

Преимущества:

- Одинаковые открытые блоки → различные зашифрованные блоки
- Паттерны замаскированы

Недостатки:

- Не допускает параллелизацию (нужно зашифровать блок N-1, чтобы зашифровать блок N)
- Медленнее других режимов

Применение: Ранее использовался широко, сейчас вытеснен GCM

4.4. CTR (Counter Mode) — Режим счётчика

Как работает:

- AES превращается в генератор потока ключей
- Шифруется не текст, а уникальное значение **Nonce + Counter** (0, 1, 2, ...)
- Результат XOR-ится с открытым текстом

Преимущества:

- Очень быстро — можно шифровать все блоки **параллельно**
- Не требуется padding (дополнение до 16 байт)

Критическая уязвимость:

- Если повторить пару Nonce+Key дважды — взлом тривиален
- Два шифротекста можно легко XOR-нуть и получить открытый текст

4.5. GCM (Galois/Counter Mode) — Золотой стандарт

Как работает:

- Комбинация CTR (скорость) + умножение в поле Галуа (аутентификация)
- Создаёт аутентификационный тег (Auth Tag)

Уникальность:

- Защищает не только от **прослушивания**, но и от **подмены данных**
- Если хакер изменит хоть один бит в зашифрованном файле → Tag не совпадёт

Применение:

- Стандарт де-факто для TLS (HTTPS)
- Используется в VPN и современных приложениях
- Обязателен для высокого уровня безопасности

5. ЗАДАНИЕ ПО ВАРИАНТАМ

Для выполнения работы необходимо выбрать вариант согласно списку группы. В качестве исходного текста для шифрования используется указанное ключевое слово в таблице вариантов. Ключ шифрования берется также из таблицы вариантов.

Важно! Программа принимает ключ в шестнадцатеричном (HEX) формате. Используйте значение из 3-го столбца таблицы.

Таблица вариантов

Номер варианта	Ключевое слово	HEX-значение ключа (для ввода в программу)
1, 6, 11, 16, 21, 26, 31, 36	**киви**	D0BAD0B8D0B2D0B80000000000000000
2, 7, 12, 17, 22, 27, 32, 37	**мора**	D0BCD0BED180D0B00000000000000000
3, 8, 13, 18, 23, 28, 33, 38	**изюм**	D0B8D0B7D18ED0BC0000000000000000
4, 9, 14, 19, 24, 29, 34, 39	**лайм**	D0BBD0B0D0B9D0BC0000000000000000
5, 10, 15, 20, 25, 30, 35, 40	**личи**	D0BBD0B8D187D0B80000000000000000

6. ОСНОВНЫЕ ЭТАПЫ ВЫПОЛНЕНИЯ

Этап 1: Ввод данных

1. Запустите программу MIREA AES Visualizer
2. Введите текст для шифрования в поле **"Текст"**
3. Скопируйте HEX-код ключа из Таблицы вариантов в поле **"Ключ"**
4. Убедитесь, что выбран режим **"AES-128 / ECB"**
5. Сделайте скриншот заполненного окна

[МЕСТО ДЛЯ СКРИНШОТА 6.1: Подготовка данных]

Этап 2: Пошаговое шифрование

1. Нажмите кнопку **"Зашифровать"**
2. Используя кнопку **"Вперёд"**, пройдите через все этапы, записывая ключевые значения:
    - Начальное состояние
    - После SubBytes раунда 1
    - После ShiftRows раунда 1
    - После MixColumns раунда 1
    - После первого AddRoundKey раунда 1
    - После каждого последующего раунда (выборочно)
    - Финальный шифротекст (раунд 10)
    - Расшифрование текста
3. Сделайте не менее 5–6 скриншотов ключевых этапов

[МЕСТО ДЛЯ СКРИНШОТА 6.2: Этап SubBytes]

[МЕСТО ДЛЯ СКРИНШОТА 6.3: Этап ShiftRows]

[МЕСТО ДЛЯ СКРИНШОТА 6.4: Этап MixColumns]

[МЕСТО ДЛЯ СКРИНШОТА 6.5: Финальный результат]

[МЕСТО ДЛЯ СКРИНШОТА 6.6: Расшифровка текста]

Этап 3: Изучение теории

1. Перейдите на вкладку **"Теория AES"**
2. Прочитайте информацию об алгоритме
3. Сделайте скриншот информационного материала

[МЕСТО ДЛЯ СКРИНШОТА 6.6: Теоретический материал]

Этап 4: Исследование режимов

1. Перейдите на вкладку **"Режимы AES"**
2. Изучите описание всех четырёх режимов
3. Запишите основные различия
4. Сделайте скриншот

[МЕСТО ДЛЯ СКРИНШОТА 6.7: Описание режимов]

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. **Архитектура**: В чём основное отличие архитектуры SPN (используемой в AES) от сети Фейстеля (используемой в DES)?
2. **Финальный раунд**: Почему в финальном раунде AES отсутствует операция MixColumns?
3. **Диффузия и конфузия**: Какие операции в AES обеспечивают "конфузию", а какие "диффузию"? Зачем нужны обе?
4. **ECB уязвимость**: Объясните, почему режим ECB небезопасен для шифрования изображений. Вспомните "проблему пингвина".
5. **CTR критичность**: В чём заключается критическая уязвимость режима CTR при повторном использовании пары Nonce+Key?
6. **GCM преимущества**: Какую дополнительную защиту обеспечивает режим GCM по сравнению с CBC и CTR?
7. **Расширение ключа**: Как изменится количество раундов при использовании AES-256 вместо AES-128?
8. **Каскадный эффект**: Что произойдёт с шифротекстом, если изменить один бит в открытом тексте? Будет ли полностью изменён финальный результат?

9. ОФОРМЛЕНИЕ ОТЧЁТА

Практическая часть (2–3 страницы)

1. **Скриншоты из программы** (не менее 5–6):
    - Главное окно с исходными данными
    - Окно с введённым текстом и ключом
    - Этап SubBytes раунда 1
    - Этап ShiftRows раунда 1
    - Этап MixColumns раунда 1
    - Финальный результат
    - Расшифрованный текст
2. **Таблица с результатами Задания 1**:
    - Заполненная таблица преобразований по раундам
    - Итоговый шифротекст в HEX-формате
    - Расшифрованный текст

Выводы

- Основные свойства алгоритма AES и его стойкость
- Роль каждой операции (SubBytes, ShiftRows, MixColumns, AddRoundKey) в обеспечении криптографической стойкости
- Сравнение режимов работы и рекомендации по их использованию
- Практические применения AES в современных системах безопасности

10. КРИТЕРИИ ОЦЕНКИ

Критерий	Баллы	Описание
Выполнение Задания 1	2	Корректное прохождение всех этапов, заполненная таблица, скриншоты
Выполнение Задания 2	1,5	Оба шифротекста, анализ различий, вывод о каскадном эффекте
Выполнение Задания 3	1,5	Таблица сравнения режимов, обоснованные рекомендации
Ответы на контрольные вопросы	2	Полные, развёрнутые ответы на все 8 вопросов
Качество скриншотов и визуализации	1	Достаточное количество скриншотов, хорошая читаемость
Теоретическая часть	1	Полное описание алгоритма, таблицы и схемы
Выводы и анализ	0,5	Глубина анализа, качество выводов
Оформление и грамотность	0,5	Структура отчёта, грамматика, читаемость
Максимальный балл	**10**	Всё выполнено идеально