Содержание

[Введение 5](#_Toc192763325)

[1 Описание функционала приложения 6](#_Toc192763326)

[1.1 Генерация ключевой пары 6](#_Toc192763327)

[1.2 Хэширование 7](#_Toc192763328)

[1.3 Генерация и проверка ЭЦП 8](#_Toc192763329)

[1.4 Шифрование и расшифрование 9](#_Toc192763330)

[2 Настройка и запуск 11](#_Toc192763331)

[3 Навыки, полученные в ходе практики 12](#_Toc192763332)

[Заключение 13](#_Toc192763333)

# Введение

Web-приложение для выполнения криптографических операций — это интернет-ресурс, предназначенный для выполнения криптографических операций.

Генерация ключевых пар, шифрование и расшифрование данных, хэширование, а также генерация и проверка электронной цифровой подписи (ЭЦП). Приложение работает с библиотеками OpenSSL и BEE2.

Цель веб-приложения — предоставить пользователям удобный инструмент для выполнения криптографических операций.

Для достижения указанной цели поставлены следующие задачи:

1. Разработать функционал для генерации ключевых пар с использованием различных алгоритмов (RSA, ECDSA, Ed25519) и библиотек (OpenSSL, BEE2).
2. Реализовать возможность шифрования и расшифрования текста с использованием алгоритмов AES (AES-128, AES-192, AES-256).
3. Создать инструмент для хэширования текста и файлов с использованием алгоритмов (SHA-256, SHA-512, MD5, Belt, Bash256, Bash512).
4. Реализовать генерацию и проверку электронной цифровой подписи (ЭЦП) с использованием библиотек (OpenSSL, BEE2).
5. Обеспечить удобный интерфейс для взаимодействия с приложением.

1 Описание функционала приложения

* 1. Генерация ключевой пары

Приложение поддерживает генерацию ключевых пар с использованием алгоритмов RSA, ECDSA и Ed25519. Для генерации ключей используются библиотеки OpenSSL и BEE2. Пользователь может выбрать алгоритм, длину ключа (для RSA) и пароль (для BEE2). Сгенерированные ключи сохраняются в базе данных и могут быть использованы для дальнейших операций. Форма для генерации пар ключей представлена на рисунках 1.1 (для OpenSSL) и 1.2 (для BEE2).

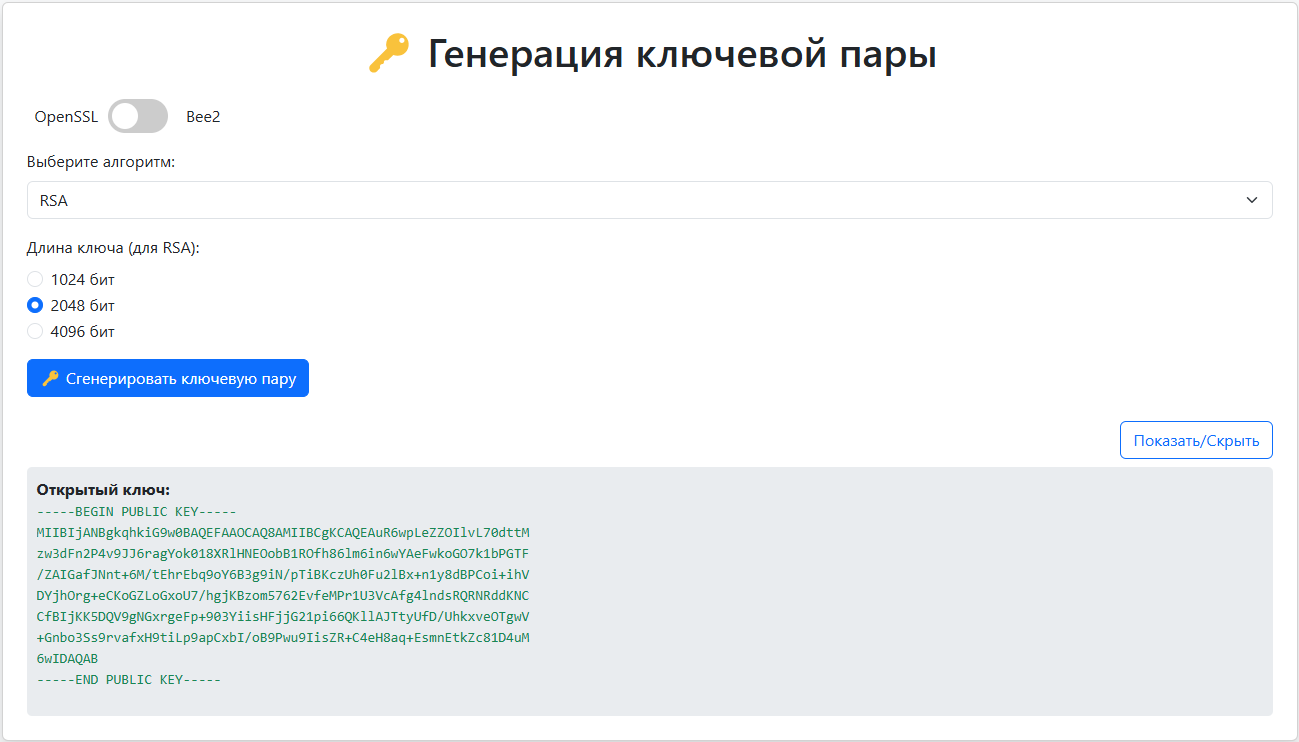


Рисунок 1.1 – Форма генерации пар ключей при помощи OpenSSL

Для генерации ключевой пары при помощи библиотеки OpenSSL нужно выбрать нужный вам алгоритм (RSA, ECDSA, ED25519). Выбрать длину ключа для RAS если вы используете этот алгоритм и нажать кнопку «Сгенерировать ключевую пару». После этого открытый ключ будет показан в форме «Открытый ключ:», а закрытый ключ будет сохранен в базу данных и доступен только на уровне сервера, его просмотреть нельзя. Ключ можно скрыть и показать, нажав на соответствующую кнопку.

Генерация ключа при помощи BEE2 представлена на рисунке 1.2. Основное отличие от OpenSSL, что нужно ввести пароль длинной не менее 6 символов.

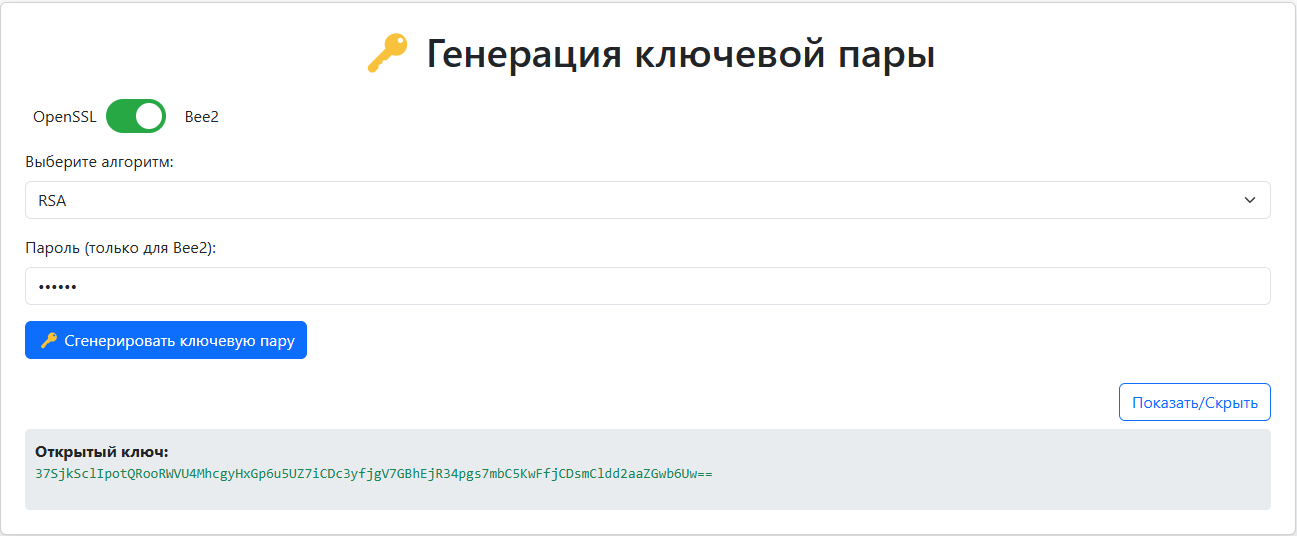
****

Рисунок 1.2 – Форма генерации пар ключей при помощи BEE2

* 1. Хэширование

Приложение поддерживает хэширование текста и файлов с использованием алгоритмов SHA-256, SHA-512, MD5, а также алгоритмов BEE2 (Belt, Bash256, Bash512). Пользователь может выбрать алгоритм и ввести текст или загрузить файл для хэширования. Примеры использования представлены на рисунках 1.3 (OpenSSL), 1.4 (BEE2).

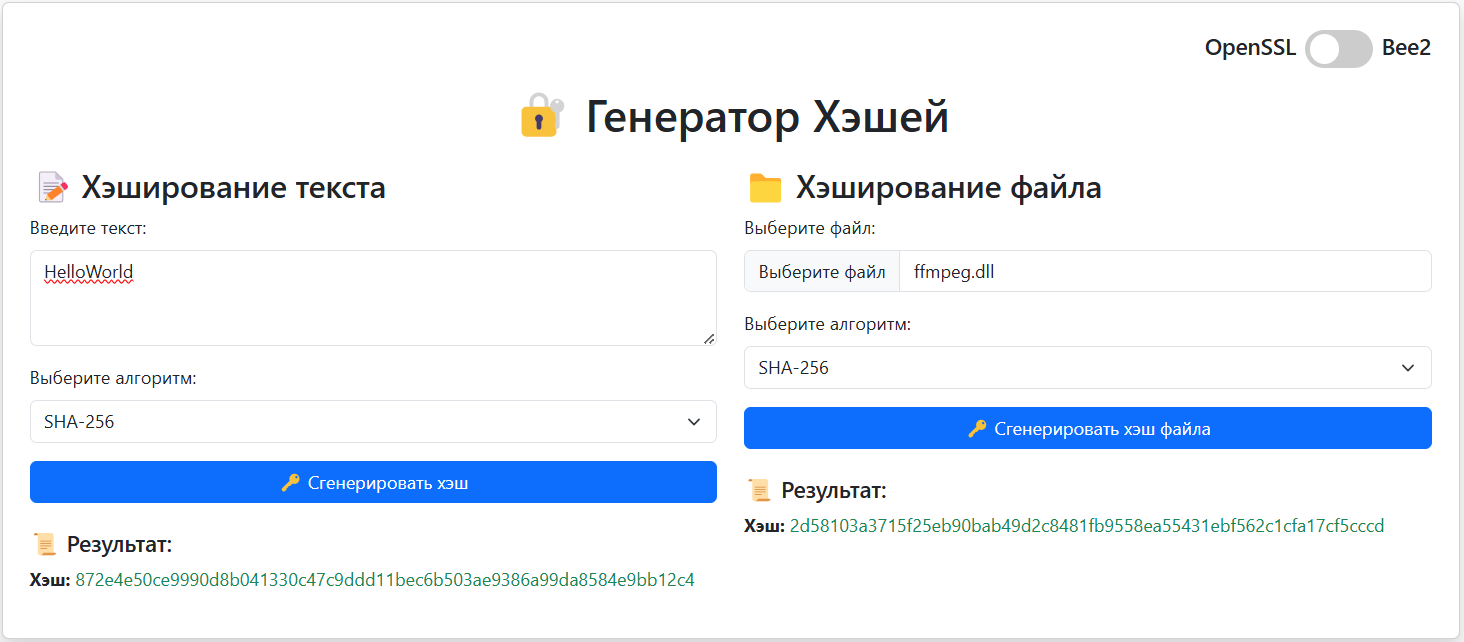


Рисунок 1.3 – Хэширование при помощи OpenSSL

Для хэширования нужно ввести текст и выбрать нужный алгоритм, далее нажать на кнопку «Сгенерировать хэш» и будет выдан результат в форме «Хэш».

Генерация хэша текста при помощи BEE2 не требует выбора алгоритма.

Для генерации хэша файла нужно выбрать любой файл на вашем ПК и загрузить его на страницу через форму, далее выбрать нужный вам алгоритм, для OpenSSL это (SHA-256, SHA-512, MD5), а для BEE2 (Belt, Bash256, Bash512). Далее в форме «Хэш» будет выведен результат. Результаты можно увидеть на рисунках 1.3 для (OpenSSL) и 1.4 (для BEE2).

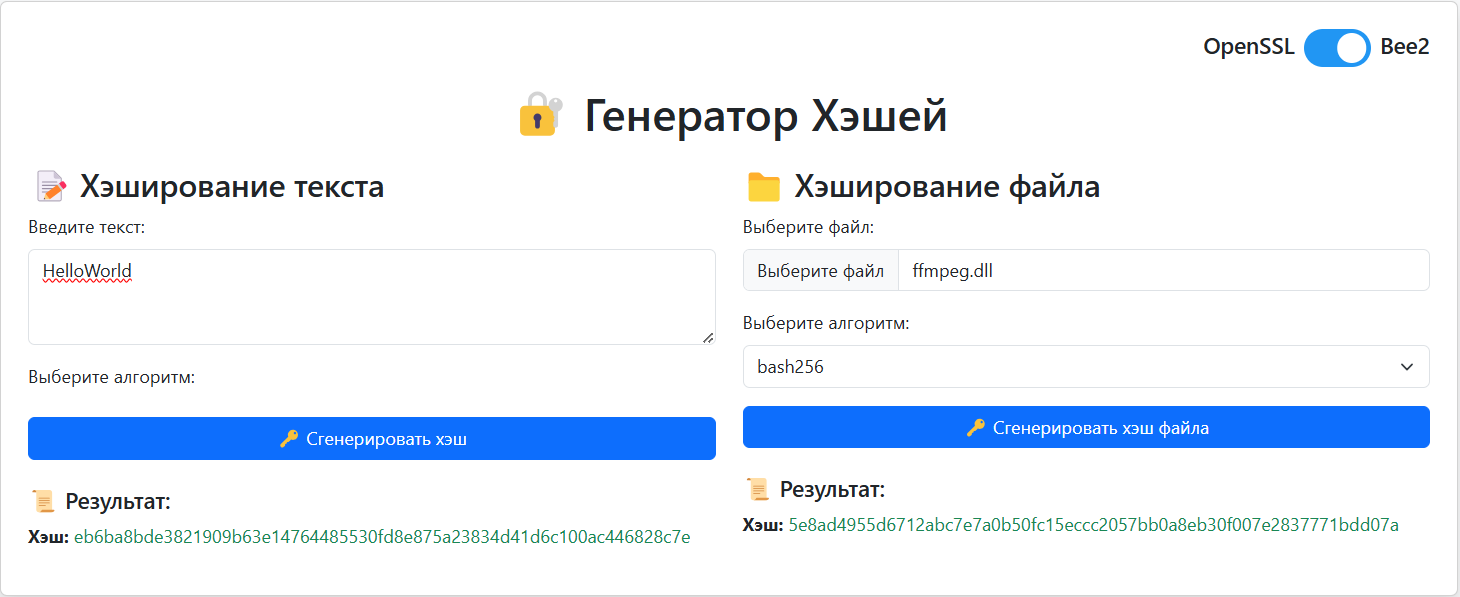


Рисунок 1.4 – Хэширование при помощи BEE2

* 1. Генерация и проверка ЭЦП

Приложение позволяет генерировать и проверять электронную цифровую подпись (ЭЦП) при помощи библиотек OpenSSL и BEE2. Приложение возвращает подпись, которую можно проверить с использованием открытого ключа. Пароль (для BEE2) должен совпадать с тем который был введен при создании пар ключей. Форма для генерации и проверки ЭЦП проставлена на рисунке 1.5 (для OpenSSL) и 1.6 (для BEE2).

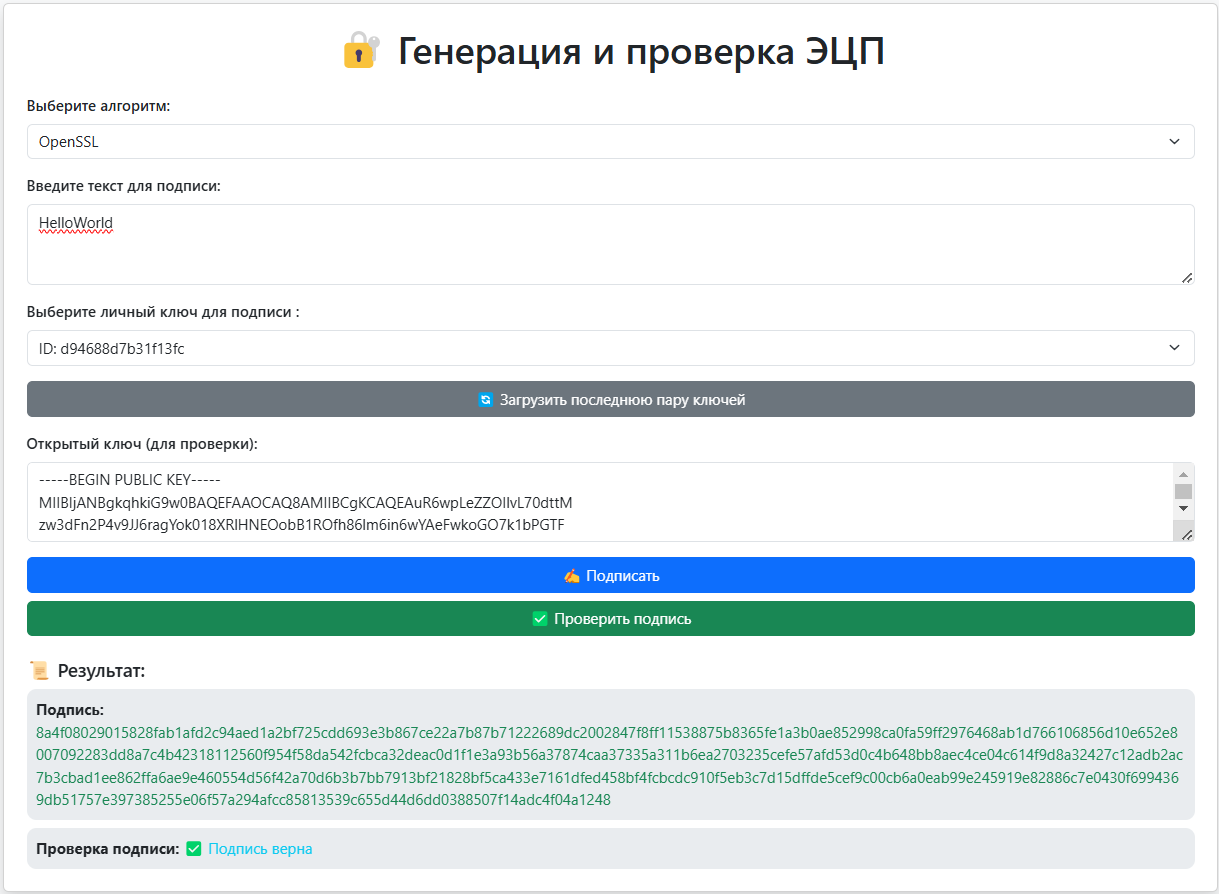


Рисунок 1.5 – Генерация и проверка ЭЦП при помощи OpenSSL.

Для того что бы сгенерировать подпись и ее проверить, нужно ввести текст, который мы хотим подписать, затем выбрать ключ из списка сгенерированных, можно загрузить последний сгенерированную пару, для этого нужно нажать кнопку «Загрузить последнюю пару ключей». Далее вы увидите открытый ключ, значит все загрузилось. После этого можно подписать текст, результат с подписанным текстом будет выведен в форму «Подпись». После подписи текста можно ее проверить, верна ли подпись, нужно нажать на кнопку «Проверить подпись» после чего будет выведено уведомление о корректности подписи в форму «Проверка подписи». Если подпись не верна, то будет выведено соответствующее сообщение.

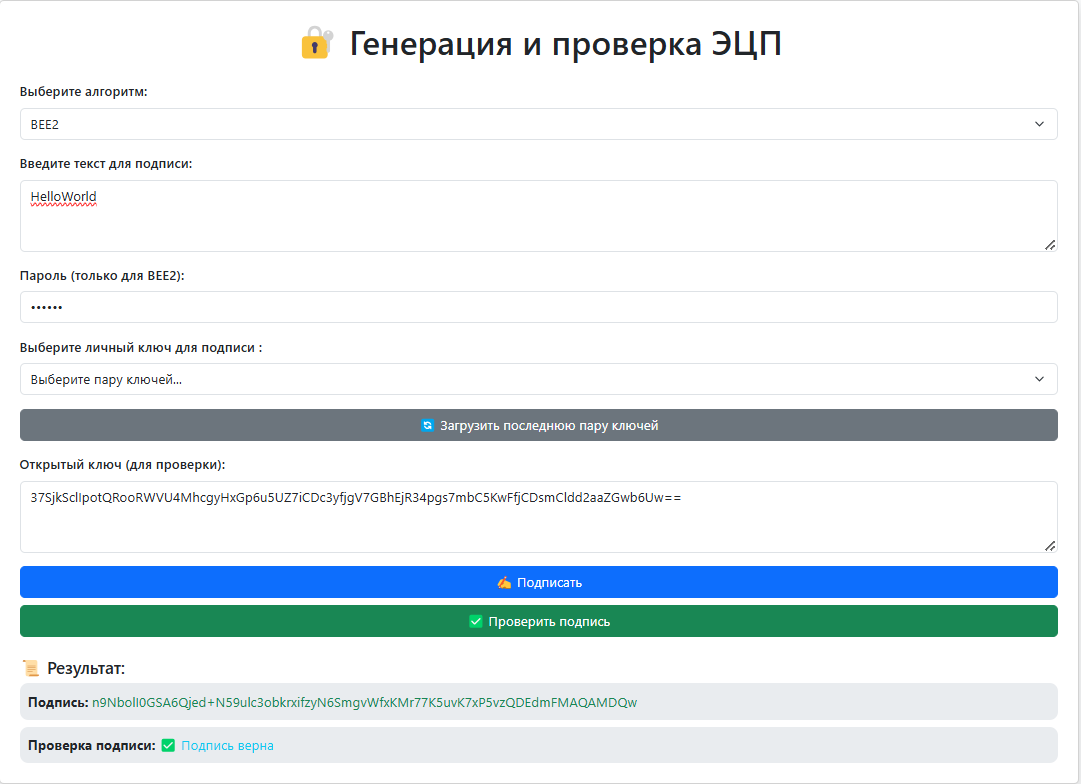


Рисунок 1.6 – Генерация и проверка ЭЦП при помощи BEE2.

Для подписания текста при помощи BEE2 нужно ввести пароль, который был введен при создании пар ключей. Пароль должен быть не менее 6 символов.

* 1. Шифрование и расшифрование

Приложение позволяет шифровать и расшифровывать текст с использованием алгоритмов AES (AES-128, AES-192, AES-256). Пользователь может выбрать ключ из списка сгенерированных или создать новый (случайное 32 битное число), все сгенерированные ключи сохраняются в базу данный. Для шифрования используется случайный вектор инициализации (IV), который возвращается вместе с зашифрованным текстом. Форма шифрования и расшифрования представлена на рисунке 1.7 (для OpenSSL).

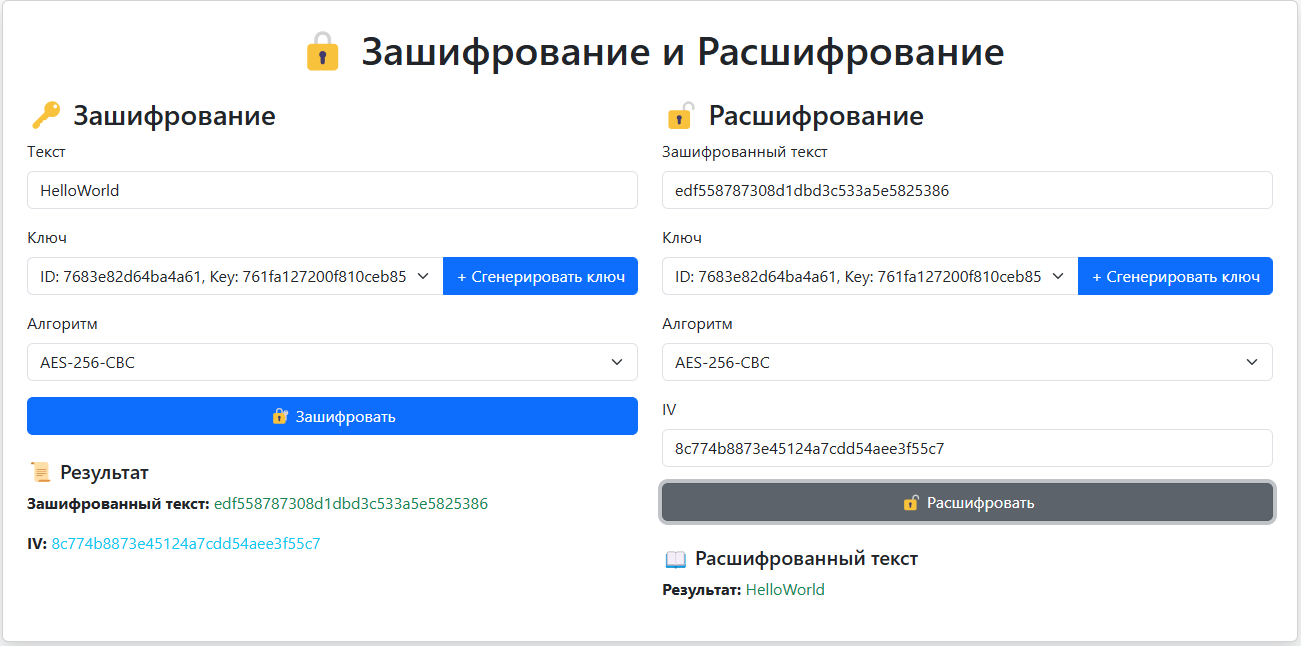


Рисунок 1.6 – Форма шифрования и расшифрования при помощи OpenSSL.

Для зашифрования нужно ввести текст для зашифрования в поле «Текст», далее выбрать ключ или сгенерироваться новый нажав на кнопку «Сгенерировать ключ» и выбрать алгоритм. После нажатии на кнопку «Зашифровать» будет получен зашифрованный текст и IV.

Что бы расшифровать текст нужно в поле «Зашифрованный текст» ввести значение, далее выбрать ключ, при помощи которого этот текст был зашифрован и выбрать алгоритм расшифрования, ввести IV и нажать на кнопку «Расшифровать». В форме «Результат» будет получено расшифрованное сообщение.

2 Настройка и запуск

Для того чтобы начать использовать приложение, необходимо развернуть его на локальной машине или сервере. Приложение поддерживает развертывание с использованием Docker, что значительно упрощает процесс установки и настройки.

1. Клонирование репозитория:

Весь исходный код лежит на платформе GitHub в открытом доступе. Для клонирования репозитория выполните команду:

git clone https://github.com/kl1mov1ch/pre-graduate-practice.git

2. Сборка:

Перейдите в директорию вашего проекта и соберите образ в Docker при помощи команды:

docker build -t <name> .

Эта команда создаст Docker-образ с выбранным именем, который содержит все необходимые зависимости и настройки для работы приложения.

3. Запуск контейнера

После успешной сборки запустите контейнер при помощи команды:

docker run -p <port>:<port> <name>

После успешного запуска приложения вы можете начать использовать его функционал. Приложение будет доступно по адресу [http://localhost:<port](http://localhost:%3cport)>.

3 Навыки, полученные в ходе практики

В ходе преддипломной практики я приобрел множество полезных навыков, которые будут полезны в дальнейшей профессиональной деятельности. Среди них:

* Работа с криптографическими библиотеками:

Я научился использовать такие библиотеки, как OpenSSL и BEE2, для выполнения различных криптографических операций, таких как генерация ключей, шифрование, хэширование и работа с электронной подписью.

* Разработка веб-приложений:

В процессе работы я углубил свои знания в разработке веб-приложений с использованием Node.js и Express. Я научился создавать RESTful API, работать с базами данных и организовывать взаимодействие между клиентом и сервером.

* Работа с Docker:

Я освоил основы работы с Docker, научился создавать Docker-образы и запускать контейнеры. Это позволило мне упростить процесс развертывания приложения и обеспечить его стабильную работу в различных средах.

* Использование Git и GitHub:

Я научился работать с системой контроля версий Git, создавать репозитории, управлять ветками и вносить изменения в код. Также я освоил использование GitHub для совместной работы над проектом и автоматизации процессов.

# Заключение

В ходе данной работы было разработано Веб-приложение для выполнения криптографических операций. Основные результаты разработки:

1. Приложение поддерживает генерацию ключевых пар с использованием алгоритмов RSA, ECDSA и Ed25519, а также библиотек OpenSSL и BEE2.
2. Реализованы функции шифрования и расшифрования текста с использованием алгоритмов AES.
3. Поддерживается хэширование текста и файлов с использованием алгоритмов SHA-256, SHA-512, MD5, Belt, Bash256 и Bash512.
4. Реализована генерация и проверка электронной цифровой подписи (ЭЦП) с использованием алгоритмов OpenSSL и BEE2.
5. Разработан удобный интерфейс для взаимодействия с приложением.

Приложение может быть применения в области информационной безопасности.