РАЗРАБОТКА СИСТЕМЫ ЗАЩИТЫ электронного документооборота на основе криптографических алгоритмов

Е.Ряполова, к.пед.н., доцент Оренбургского филиала ПГУТИ,
М.Студянникова, к.пед.н., доцент Оренбургского филиала ПГУТИ / studyannikovam@mail.ru,
К.Цветкова, к.пед.н., доцент Оренбургского филиала ПГУТИ

УДК 519.6, DOI: 10.22184/2070-8963.2020.90.5.58.63

Предложен проект системы защиты электронного документооборота на основе криптографических алгоритмов. Система представляет собой удобный REST API интерфейс для внедрения в существующую среду и обеспечения высокой производительности. Разработаны концептуальные схемы взаимодействия компонентов, диаграммы прецедентов, определена последовательность действий. Структурная схема системы защиты электронного документооборота состоит из четырех подсистем: аутентификации, управления доступом, криптографической защиты информации и подсистемы работы с электронными документами. Реализована система защиты на языке программирования Python и платформе Django.

Эффективная защита информации в корпоративных системах и сетях невозможна без использования современных методов и средств информационной безопасности. Потребность в защите информации актуальна для многих организаций. Игнорирование проблемы безопасности информации приводит к финансовым потерям. Постоянное появление новых угроз безопасности требует совершенствования методов и средств защиты информации в любой организации.

По мере увеличения объема информации возникает необходимость внедрения в организациях систем электронного документооборота, позволяющих не только повысить производительность труда сотрудников, но и обеспечить гибкость хранения и обработки информации. Но, с другой стороны, использование систем электронного документооборота приводит к появлению дополнительных рисков и уязвимостей конфиденциальной информации.

Электронный документ может содержать информацию разного уровня секретности. Утечка конфиденциальной информации негативно влияет на деятельность организации. Процесс передачи информации должен быть регламентирован и сопровождаться организационными мероприятиями, обеспечивающими защиту от несанкционированного доступа.

В защищенных информационных системах необходимо предусмотреть методы противодействия базовым угрозам: аутентификацию, безопасный доступ к ресурсам, конфиденциальность и целостность документов, логирование действий пользователей в системе.

На сегодняшний день при реализации системы защиты электронного документооборота

эффективнее всего использовать криптографические методы защиты информации: шифрование информации на сервере, в базах данных, шифрование учетных данных пользователей, шифрование соединения, аутентификация пользователей.

Архитектура предложенной системы изображена на рис.1. Программное средство представлено клиент-серверной архитектурой, где хранятся документы организации и база пользователей. В качестве клиента выступает веб-браузер, посредством которого осуществляется работа с сервисом. Хранение информации на сервере организовано с помощью системы управления базой данных.

Один из компонентов защиты электронного документооборота – подсистема аутентификации, посредством которой выполняются процедуры регистрации и авторизации пользователя в системе.

Мы предлагаем реализовать в системе защиты электронного документооборота многофакторную аутентификацию, основанную на использовании одноразового кода доступа по e-mail. Одно из преимуществ такой аутентификации – снижение вероятности кражи личных данных. Знания пароля в данном случае недостаточно для входа в информационную систему. Доступ к электронной почте осуществляется с помощью мобильного устройства, что весьма удобно, не требуются дополнительные токены.

Алгоритм аутентификации в системе предусматривает следующую процедуру:

- для регистрации пользователя директору компании необходимо указать адрес его электронной почты;
- на почту потенциальному пользователю приходит сообщение со ссылкой, где ему требуется задать пароль;
- данные отправляются на сервер, и процесс регистрации считается завершенным.

При последующем входе в систему пользователь вводит логин/пароль, на адрес электронной почты ему высылается код подтверждения, который он также вводит в форму авторизации. Если данные введены корректно, пользователь получает сообщение об успешной авторизации, в противном случае – уведомление об ошибке.

Второй компонент системы – подсистема управления доступом. Подсистема предназначена для управления доступом пользователей к ресурсам системы, а также для распределения и использования ресурсов и объектов системы зарегистрированными пользователями в соответствии с установленными правилами разграничения доступа.

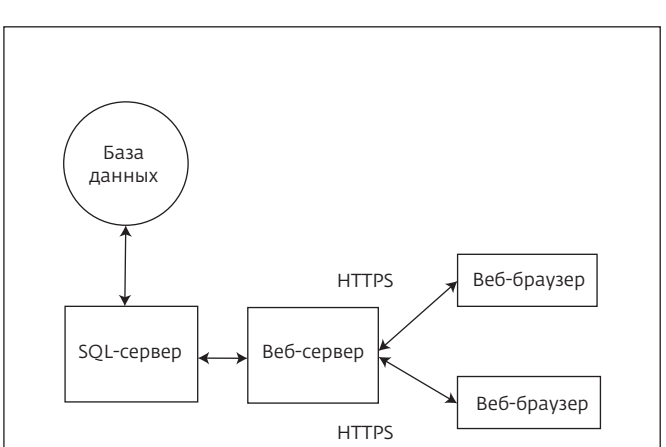

Рис.1. Структура разрабатываемой системы

В информационной системе рекомендуется реализовать ролевую модель управления доступом – принадлежность пользователя к той или иной роли определяет его права доступа к ресурсам системы электронного документооборота.

Пользователями системы являются администратор, руководитель организации и сотрудники отдела. Настройкой и поддержкой системы занимается администратор. До начала работы пользователю необходимо пройти регистрацию и авторизацию.

Третий компонент системы – подсистема криптографической защиты информации – реализует алгоритмы криптографического преобразования информации. Подсистема предназначена для защищенной передачи информации по каналам связи и для защиты от несанкционированного доступа.

В разрабатываемой системе в целях снижения риска взлома и предотвращения утечки паролей пользователей реализован алгоритм хеширования SHA256 с использованием соли. Длина выходной хеш-функции составляет 256 бит, благодаря чему процесс расшифровки и взлома становится трудоемким, основанным на последовательном переборе [1].

Алгоритм хеширования следующий: исходный текст разбивается на блоки одинаковой длины, размер одного блока – 64 бита. При необходимости последний блок заполняется до полного нулями. Каждый блок проходит через функцию сжатия с 64 или 80 итерациями. Функция сжатия преобразует два входных блока постоянной длины в выходной блок того же размера. Функция выполняется последовательно над результатом предыдущего

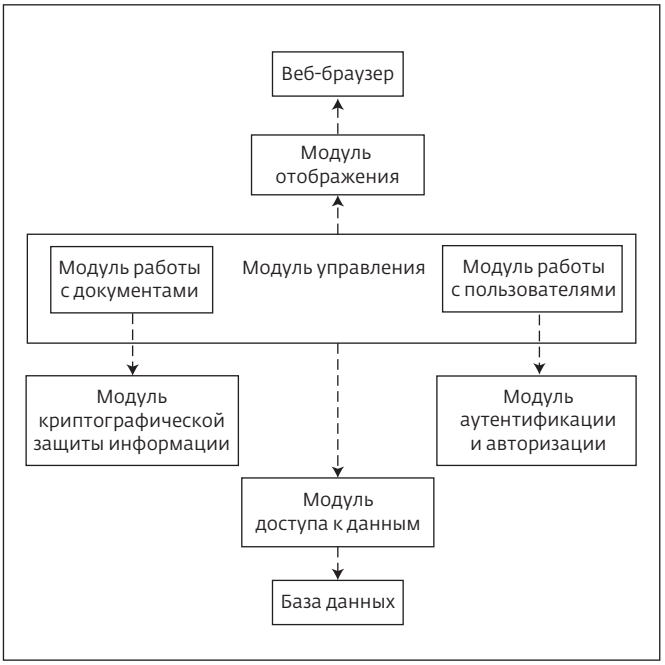

Рис.2. Диаграмма компонентов системы защиты электронного документооборота

Рис.3. Диаграмма последовательности действий прецедента "Авторизация пользователя"

прохода. Результаты каждого прохода складываются, полученная сумма – это значение хеш-функции.

Безопасная передача данных по сети обеспечивается с помощью криптографического протокола SSL, который основан на алгоритме асимметричного шифрования.

При каждом входе пользователя на сайт браузер и сервер устанавливают SSL-соединение. При вводе адреса браузер обращается к серверу с запросом о наличии сертификата у запрашиваемого сайта. В ответ сервер отправляет информацию о сертификате и открытый ключ. Браузер проверяет SSL-сертификат. Если проверка прошла успешно, браузер генерирует сеансовый ключ, который шифруется открытым ключом, и отправляет на сервер. Сервер расшифровывает сеансовый ключ, после чего между ними устанавливается безопасное соединение через HTTPS-протокол [1].

Хранение электронных документов в базе данных осуществляется в зашифрованном виде. Шифрование данных выполняется в браузере пользователя, для этого используется алгоритм AES с длиной ключа в 128 бит [2].

В разрабатываемой системе ключом шифрования документов служит пароль, который руководитель задает в личном кабинете при регистрации организации. Пароль хранится в базе данных в виде хеш-значения. Данный ключ шифрования отправляется системой на e-mail пользователя в том случае, если руководитель предоставил пользователю право доступа к зашифрованному документу. При попытке пользователя прочитать файл открывается окно с полем для ввода ключа, после чего обеспечивается его проверка в базе. Если ключ верный, документ расшифровывается и открывается в новом окне. В противном случае появляется уведомление об ошибке.

Четвертый компонент – подсистема работы с электронными документами – включает в себя функции добавления, удаления и чтения документов.

На рис.2 изображена диаграмма компонентов, которые представляют собой отдельные модули, реализованные в системе.

Модуль управления является главным и отвечает за логику работы с пользователями и электронными документами. Модуль доступа к данным наделен функциями для работы с базой данных: получение информации, ее сохранение и обновление. Для визуализации информации в браузере пользователя используется модуль отображения. Механизмы регистрации и входа пользователей в систему реализованы в модуле аутентификации и авторизации. В модуле криптографической защиты информации воплощены алгоритмы криптографического преобразования информации для ее безопасной передачи по линиям связи и хранения в базе данных.

Для демонстрации жизненного цикла объектов были созданы диаграммы последовательностей,

Рис.4. Диаграмма последовательности действий прецедента "Шифрование документа при добавлении"

одна из них – применительно к прецеденту "Авторизация пользователя" – показана на рис.3.

Для авторизации в системе пользователю необходимо указать логин и пароль, после чего на адрес электронной почты система пришлет сгенерированный код доступа. При указании некорректных данных система уведомит пользователя об ошибке.

На рис.4 представлена диаграмма последовательности действий применительно к прецеденту "Шифрование документа при добавлении".

При добавлении конфиденциального документа открывается окно, в котором пользователь указывает путь к файлу. После этого система с помощью секретного ключа шифрует документ и отправляет его в базу данных. Процесс установки безопасного соединения между браузером и сервером отображен на рис.5.

Для создания защищенного канала браузер посылает запрос веб-серверу, который в ответ отправляет копию сертификата. Затем браузер проверяет подлинность сертификата в центре сертификации. Если сертификат неподдельный, веб-сервер и браузер тайно договариваются о секретном ключе. С помощью секретного ключа браузер и сервер устанавливают защищенное HTTPS-соединение. С помощью секретного ключа шифруются данные пользователей.

Движение информационных потоков представлено на контекстной диаграмме (рис.6).

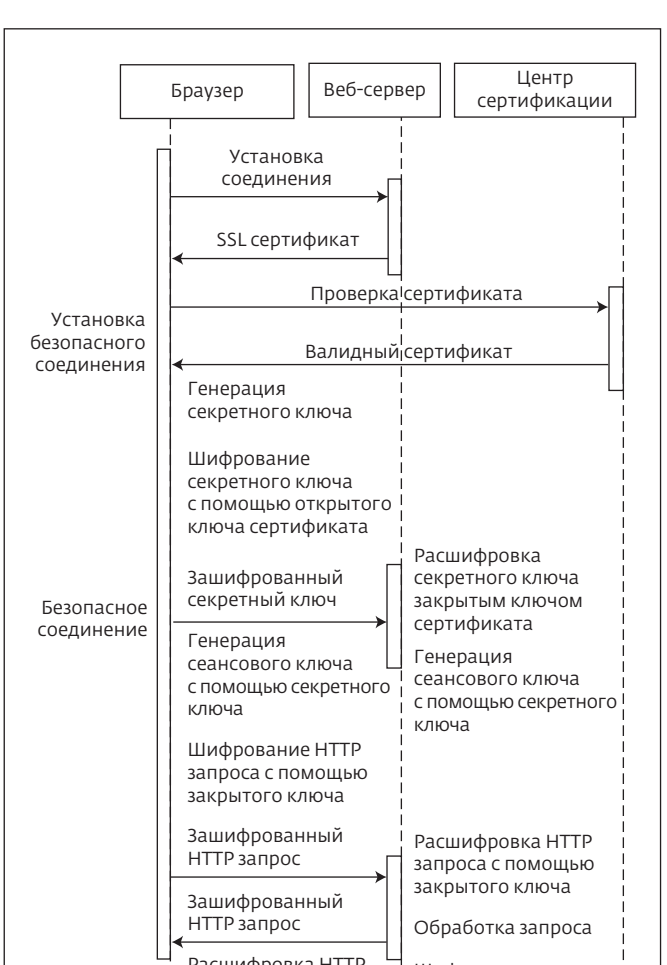

Рис.5. Диаграмма последовательности действий прецедента "Установка защищенного соединения"

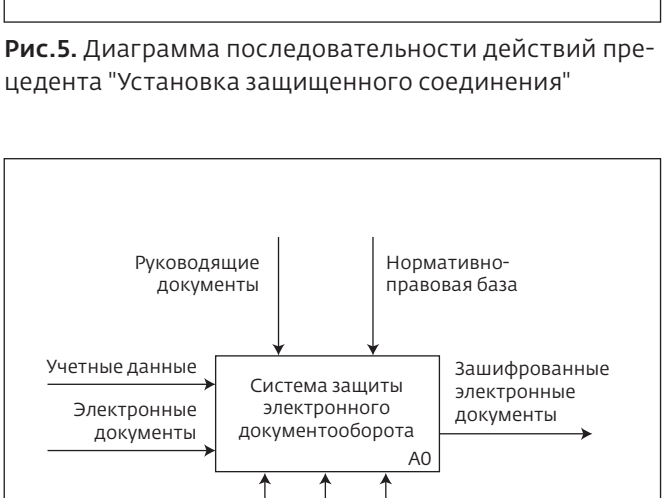

Рис.6. Контекстная диаграмма

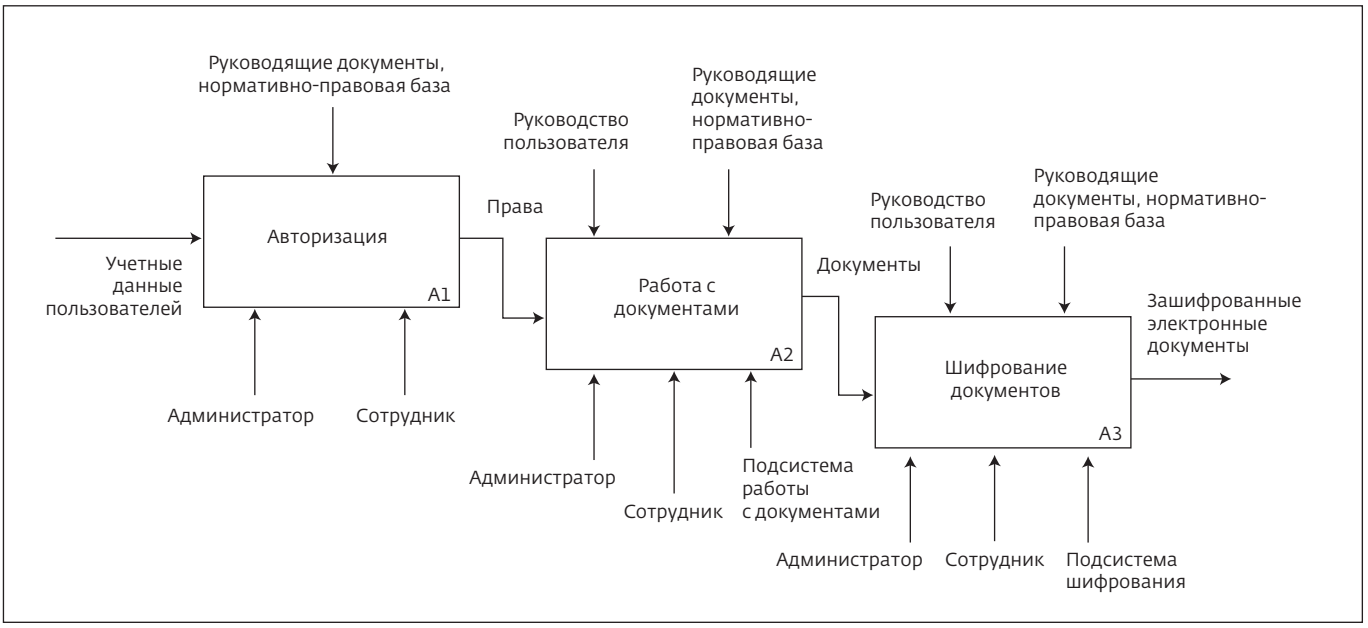

Рис.7. Декомпозиция контекстной диаграммы

Диаграмма на рис.7 позволяет определить основные информационные потоки разрабатываемой системы.

Выполним декомпозицию блока "Авторизация".

Из диаграмм на рис.2, 3, 4 видно, что поступающие в систему запросы обрабатываются сервером. Для доступа в личный кабинет пользователю необходимо ввести данные авторизации. После входа в личный кабинет ему предоставляются права доступа к документам.

При разработке системы предпочтение было отдано варианту клиент-серверной архитектуры. Для реализации серверной части были выбраны язык программирования Python и веб-фреймворк Django [3]. Основное преимущество данной платформы заключается в использовании шаблона проектирования MVC.

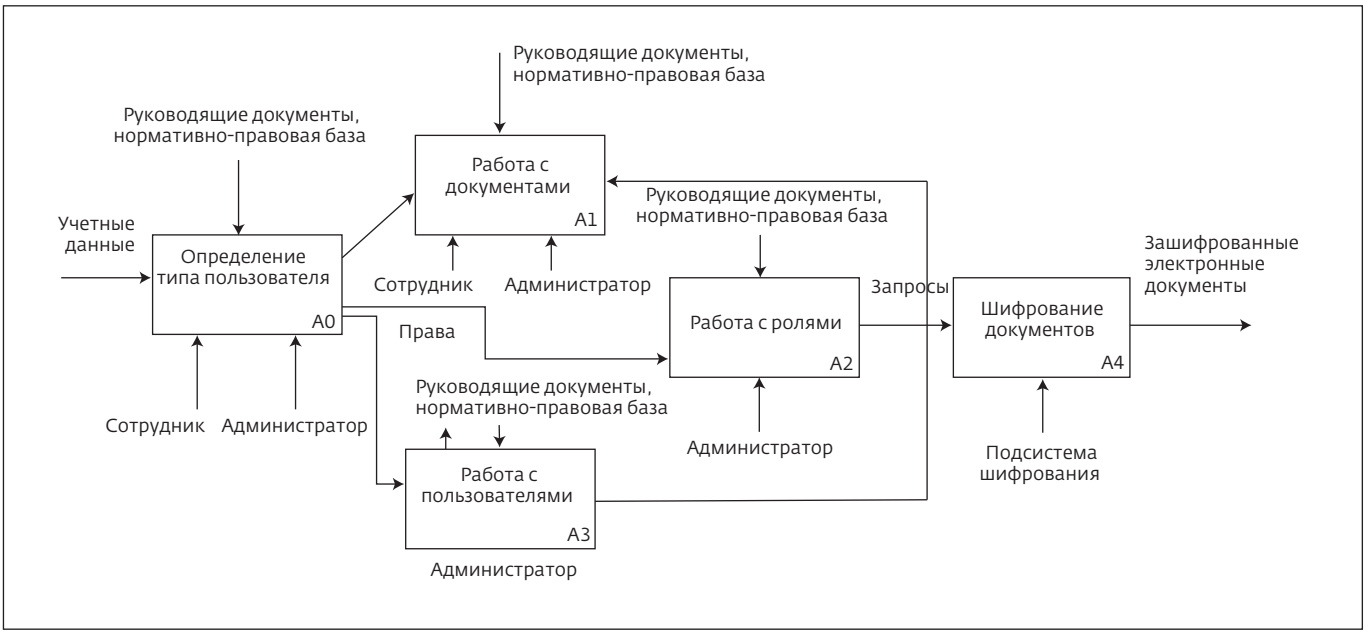

Рис.8. Декомпозиция блока "Авторизация"

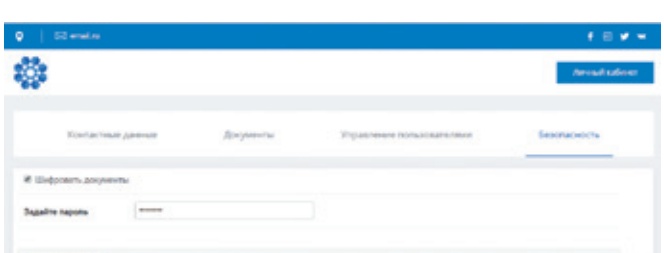

Рис.9. Форма задания пароля при входе в систему

В состав языка программирования Python входит система управления пакетами Pip, которая содержит большое количество библиотек и пакетов, написанных на данном языке [5]. Удобно использовать готовый код, уменьшается количество потенциальных ошибок, так как все пакеты находятся в открытом доступе и тестируются сообществом разработчиков.

Для развертывания проекта необходимо наличие файла requirement.txt, в котором содержится список необходимых пакетов. Pip самостоятельно разрешает зависимости между ними.

Для разработки бэкенда сервера использовалась среда разработки PyCharm от компании JetBrains. Основные преимущества этого инструмента – поддержка фреймворков Django, Flask, поиск дублирующего кода, навигация по проекту, умное автодополнение кода.

Для реализации API была выбрана технология GraphQL. Этот язык запросов к API-интерфейсам возвращает предоставленные сервером данные, из которых клиент самостоятельно может выбрать необходимые. Также есть возможность получить несколько ресурсов в рамках одного запроса к серверу, отпадает необходимость выполнять множество вызовов REST API.

Разработанная система защиты электронного документооборота представлена на рис.9.

Пользователю, которого добавил руководитель, назначается роль сотрудника. При входе в систему сотруднику необходимо ввести код доступа, который высылается ему на электронную почту.

Таким образом, разработанная система защиты электронного документооборота предоставляет удобный REST API интерфейс для внедрения в существующую среду и обеспечения высокой производительности.

Анализ базовой системы защиты электронного документооборота позволил выявить такие уязвимости, как слабые механизмы аутентификации и авторизации, SQL-инъекции, использование на сайтах открытой аутентификации HTTP, передача данных по открытому каналу связи. В рамках проекта были

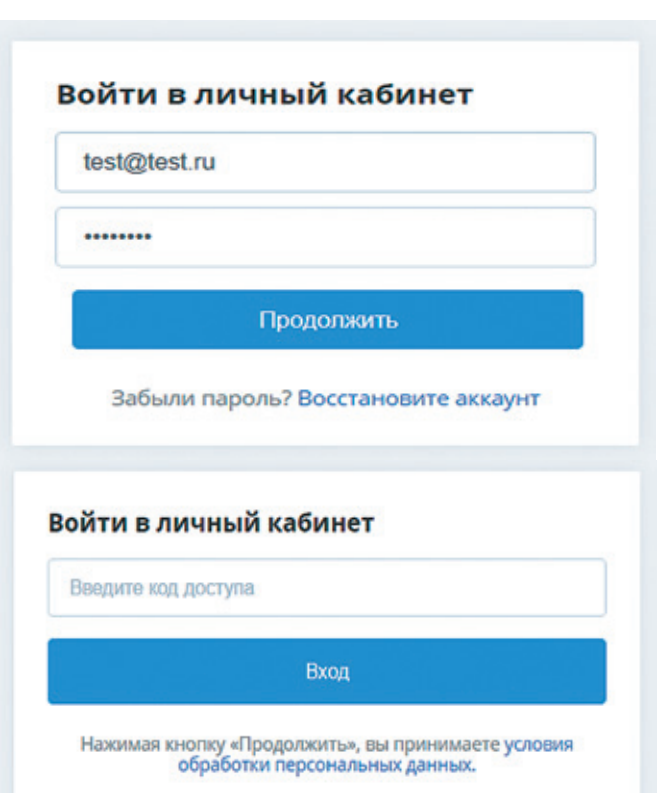

Рис.10. Окна входа в систему (вверху – окно ввода логина/пароля; внизу – окно ввода кода доступа)

составлены концептуальные схемы взаимодействия компонентов, диаграммы прецедентов и последовательности действий. Для разработки системы защиты электронного документооборота использовались язык программирования Python и платформа Django [3, 4]. Предложенная структурная схема системы включает в себя подсистемы аутентификации, управления доступом, криптографической защиты информации и работы с электронными документами.

ЛИТЕРАТУРА

1. **Смарт Н.** Криптография: учебное пособие. – М.: ТЕХНОСФЕРА, 2005. 528 с. – ISBN 5-94836-043-1.
2. **Рябко Б.Я.** Криптографические методы защиты информации: учебное пособие. – М.: Горячая линия – Телеком, 2005. 229 с.
3. Документация фреймворка Django – Режим доступа: https://www.djangoproject.com/
4. Документация JavaScript – Режим доступа: https://learn.javascript.ru/
5. Документация Python – Режим доступа: https://www.python.org