МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА РАСПРЕДЕЛЁННОЙ СИСТЕМЫ ХРАНЕНИЯ ФАЙЛОВ НА БАЗЕ ПРОТОКОЛА IPFS И БЛОКЧЕЙНА ETHEREUM

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета компьютерных наук и информационных технологий
Соловьева Яна Ярославовича

Научный руководитель
доцент, к. ф.-м. н. _____ А. С. Иванов

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. В условиях стремительного роста объёмов цифровых данных задача надёжного и безопасного хранения файлов приобретает особую значимость. Современные облачные хранилища, такие как Google Drive, Dropbox и iCloud, обеспечивают удобный доступ к данным, однако имеют ряд фундаментальных ограничений: зависимость от провайдера, единую точку отказа и подверженность цензуре. Пользователь не контролирует свои данные в полной мере — провайдер может ограничить доступ, удалить контент или прекратить работу сервиса.

Альтернативный подход к хранению данных предлагают децентрализованные технологии. Протокол IPFS (InterPlanetary File System) обеспечивает распределённое хранение файлов с контент-адресацией, а технология блокчейн позволяет создать неизменяемый реестр владения данными без центрального управляющего органа. Совместное использование этих технологий даёт возможность построить систему, в которой файлы хранятся распределённо, а права доступа обеспечиваются криптографически.

Таким образом, разработка децентрализованного файлового хранилища, объединяющего возможности IPFS и блокчейна Ethereum, является актуальной задачей, направленной на повышение безопасности, надёжности и независимости хранения пользовательских данных.

Цель бакалаврской работы — разработать прототип распределённой системы хранения файлов на базе протокола IPFS и блокчейна Ethereum, обеспечивающий безопасную загрузку, шифрование, хранение и управление файлами без зависимости от централизованных сервисов.

Поставленная цель определила следующие задачи:

— провести обзор существующих типов файловых хранилищ и проанализировать их недостатки;

— изучить технологию IPFS, её архитектуру и основные компоненты;

— исследовать технологию блокчейн, платформу Ethereum и концепцию смарт-контрактов;

— проанализировать возможности совместного использования блокчейна и IPFS для организации децентрализованного хранилища;

— сформулировать функциональные и нефункциональные требования к прототипу и обосновать выбор технологического стека;

— разработать прототип в виде веб-приложения с серверной частью на смарт-контракте и провести его тестирование.

Методологические основы работы составляют технологии распределённых систем хранения данных, протокол IPFS, платформа

блокчейн Ethereum, язык программирования смарт-контрактов Solidity, методы клиентского шифрования, а также современные средства разработки веб-приложений.

Теоретическая значимость бакалаврской работы заключается в систематизации сведений о технологиях IPFS и блокчейн, анализе архитектурных подходов к их совместному использованию и формализации требований к децентрализованным файловым хранилищам.

Практическая значимость бакалаврской работы заключается в разработке функционального прототипа децентрализованного файлового хранилища с поддержкой сквозного шифрования, который может быть использован как основа для создания полноценного продукта.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 8 приложений. Общий объём работы — 67 страниц, включая список использованных источников из 20 наименований и приложения с исходным кодом программы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Раздел 1. Общие сведения о файловых хранилищах и технология IPFS

Первый раздел посвящён обзору существующих типов хранилищ и подробному описанию протокола IPFS.

Обзор существующих хранилищ. В работе рассмотрены четыре основных типа файловых хранилищ: локальные (внутренние диски устройств), облачные (Google Drive, Dropbox, iCloud), внешние накопители (USB-флешки, портативные диски) и сетевые присоединённые хранилища NAS. Анализ показал, что все существующие решения либо централизованы и зависят от провайдера, либо имеют ограниченную доступность.

Технология IPFS. Рассмотрен протокол IPFS (InterPlanetary File System), разработанный компанией Protocol Labs в 2014 году. Описаны ключевые компоненты технологии: контент-адресация через идентификаторы CID, структура данных Merkle DAG, распределённая хеш-таблица DHT и протокол обмена данными BitSwap. Проведён анализ преимуществ (отказоустойчивость, проверяемость, цензуроустойчивость) и недостатков (зависимость от закрепления файлов, сложность освоения). Рассмотрены реальные проекты, использующие IPFS: Filecoin, OpenBazaar, Decentraland, DTube.

Раздел 2. Технология блокчейн

Второй раздел посвящён исследованию технологии блокчейн, её архитектуры, типов сетей, смарт-контрактов и возможностей взаимодействия с IPFS.

Основы блокчейн. Рассмотрена история развития технологии, начиная с концепции Сатоси Накамото и запуска Bitcoin в 2009 году до появления платформы Ethereum с поддержкой смарт-контрактов. Описаны криптографические основы блокчейна: хеширование, цифровые подписи, структура блоков и механизмы консенсуса. Проанализированы три типа блокчейн-сетей: публичные, приватные и консорциумные.

Смарт-контракты и децентрализованные приложения. Исследованы возможности смарт-контрактов как автоматически исполняемых программ в блокчейне. Рассмотрены платформы для разработки децентрализованных приложений (DApps) и примеры их применения.

Взаимодействие блокчейна и IPFS. Ключевым архитектурным решением данной работы является разделение ответственности: IPFS обеспечивает распределённое хранение содержимого файлов, тогда как блокчейн формирует неизменяемый реестр метаданных и обеспечивает контроль доступа. Такое сочетание устраняет главные недостатки каждой технологии в отдельности — хранение больших файлов в блокчейне экономически нецелесообразно, а контроль прав владения в IPFS отсутствует. Рассмотрены существующие решения (Filecoin, Ethereum Swarm).

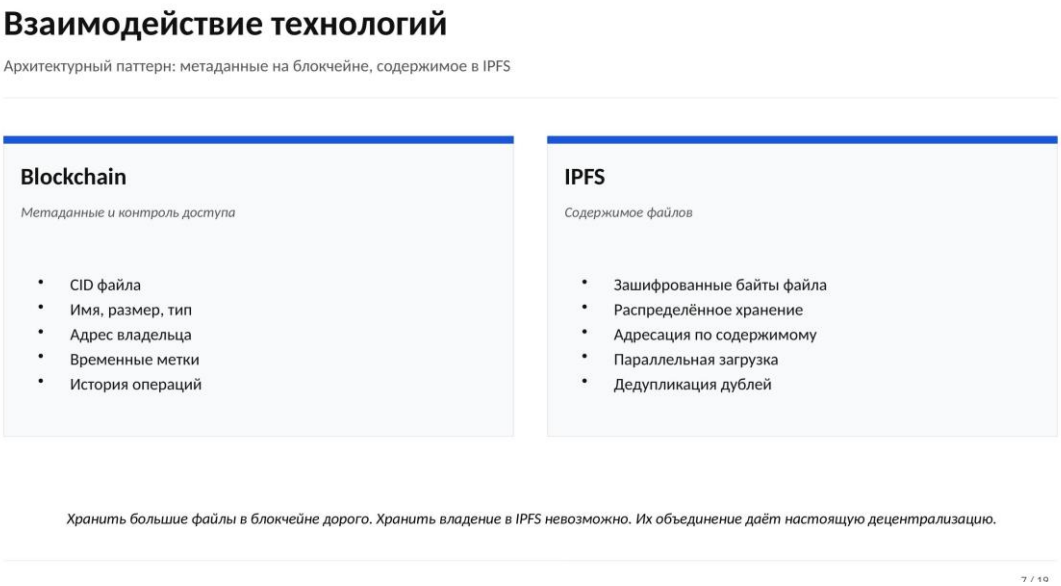

Рисунок 1 — Архитектурный паттерн взаимодействия Blockchain и IPFS

Раздел 3. Разработка прототипа децентрализованного файлового хранилища

Третий раздел посвящён практической реализации прототипа: формулировке требований, выбору технологий, разработке и тестированию.

Требования к прототипу. Сформулированы функциональные требования:

— подключение кошелька MetaMask для аутентификации пользователя;

— шифрование файла в браузере паролем пользователя (AES-256-GCM);

— загрузка зашифрованных файлов в IPFS;

— запись метаданных (CID, имя, размер, тип) в смарт-контракт;

— просмотр списка файлов, скачивание с расшифровкой;

— удаление записи о файле из реестра;

— поиск файлов по имени.

Нефункциональные требования включают: кроссплатформенность, минималистичный интерфейс, открытый исходный код (лицензия MIT), работу в тестовой сети без реальных финансовых затрат.

Технологический стек. Обоснован выбор технологий для каждого слоя системы.

Технологический стек

Слой	Технология	Назначение
Фронтенд	React 19, TypeScript, Vite	Компонентный пользовательский интерфейс
Стилизация	Tailwind CSS	Утилитарный CSS-фреймворк
Блокчейн	Solidity 0.8.20, Hardhat	Разработка и развёртывание смарт-контракта
Сеть	Polygon Amoy (тестнет Ethereum)	EVM-совместимый L2-блокчейн
Web3-библиотека	Ethers.js v6	Взаимодействие фронтенда с блокчейном
Кошелёк	MetaMask	Аутентификация и подпись транзакций
Хранение	IPFS через Pinata	Распределённое хранение зашифрованных файлов
Шифрование	Web Crypto API	AES-256-GCM, PBKDF2 (100 000 итераций)
Хостинг	Vercel	Публичное развёртывание фронтенда

Рисунок 2 — Технологический стек прототипа

Фронтенд разработан на React 19 с TypeScript и Vite, стилизация выполнена с помощью Tailwind CSS. Для разработки смарт-контракта использован язык Solidity 0.8.20 со средой Hardhat, а в качестве блокчейн-сети выбрана тестовая сеть Polygon Amoy — EVM-совместимый L2-блокчейн. Взаимодействие с блокчейном обеспечивается через библиотеку Ethers.js v6, аутентификация и подпись транзакций — через кошелёк MetaMask. Для хранения файлов используется IPFS через сервис Pinata, шифрование реализовано посредством Web Crypto API (AES-256-GCM, PBKDF2 с 100 000 итерациями). Фронтенд размещён на платформе Vercel.

Архитектура прототипа построена на принципе трёхслойного разделения ответственности.

Архитектура приложения

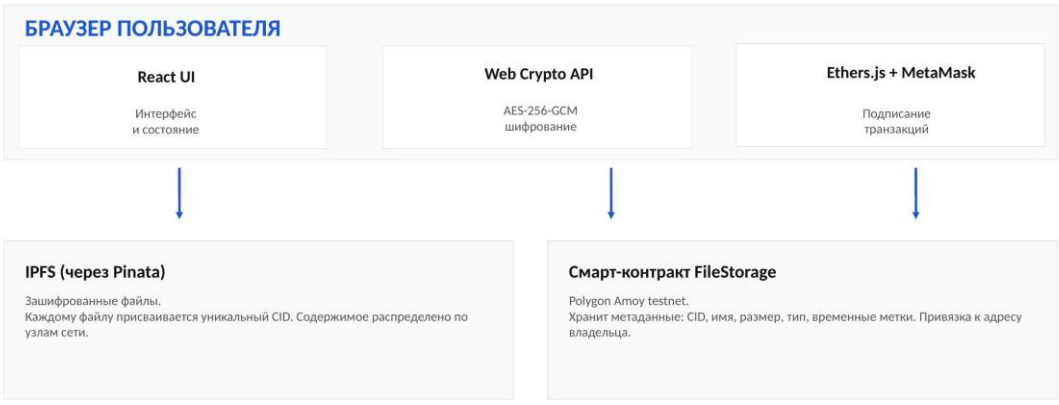

Рисунок 3 — Архитектура прототипа: браузер — IPFS — смарт-контракт

Первый слой — браузер пользователя: здесь выполняются шифрование файла алгоритмом AES-256-GCM и взаимодействие с блокчейном через Ethers.js и MetaMask. Второй слой — сеть IPFS через сервис Pinata: обеспечивает распределённое хранение зашифрованных файлов, каждому из которых присваивается уникальный CID. Третий слой — смарт-контракт FileStorage в сети Polygon Amoy: хранит метаданные файлов (CID, имя, размер, тип, временны́е метки) и контролирует права доступа через адрес кошелька владельца.

Принципиально важно, что пароль шифрования никогда не покидает браузер пользователя, что гарантирует конфиденциальность данных.

Реализация прототипа. Разработан смарт-контракт FileStorage на языке Solidity, реализующий хранение метаданных файлов и контроль доступа через адрес кошелька. Контракт развёрнут в тестовой сети Polygon Amoy. Создано клиентское веб-приложение на React с поддержкой сквозного шифрования: файл шифруется непосредственно в браузере, после чего зашифрованные данные загружаются в IPFS.

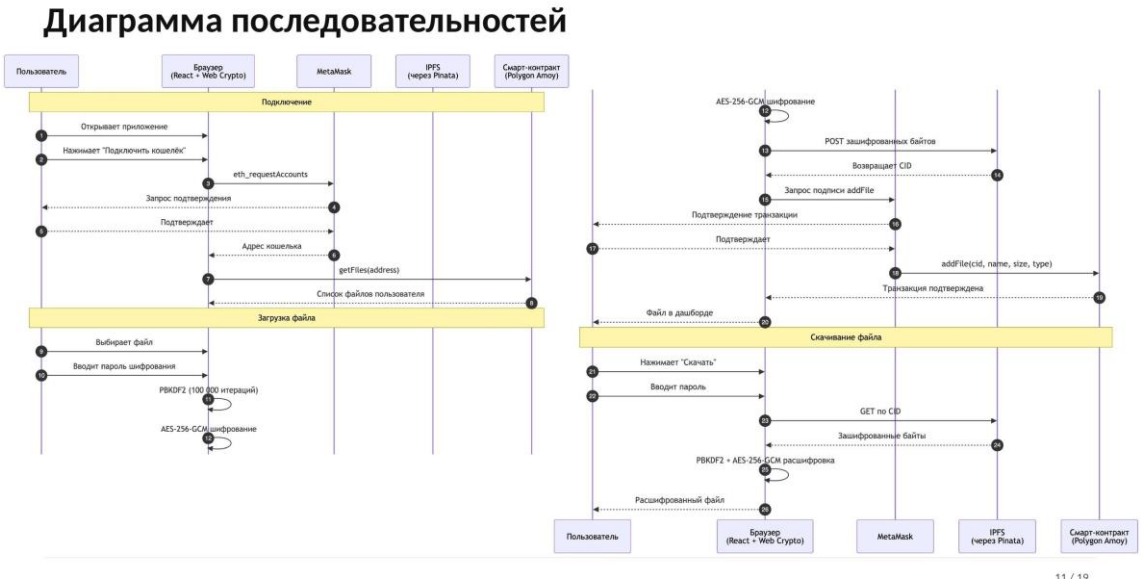

Рисунок 4 — Диаграмма последовательностей: подключение кошелька, загрузка и скачивание файла

Диаграмма последовательностей иллюстрирует три основных сценария работы с системой. При подключении кошелька пользователь авторизуется через MetaMask, и приложение получает адрес кошелька и список файлов из смарт-контракта. При загрузке файла пользователь вводит пароль; в браузере выполняется деривация ключа (PBKDF2, 100 000 итераций) и шифрование (AES-256-GCM); зашифрованные данные передаются в IPFS через Pinata, а полученный CID вместе с метаданными записывается в блокчейн. При скачивании выполняется обратная операция: зашифрованные байты получаются из IPFS по CID, расшифровываются в браузере с использованием введённого пользователем пароля.

Демонстрация работы. Проведено тестирование полного цикла работы с файлами: загрузка с шифрованием, скачивание с расшифровкой, поиск и удаление. Ниже представлен интерфейс разработанного приложения Vault3.

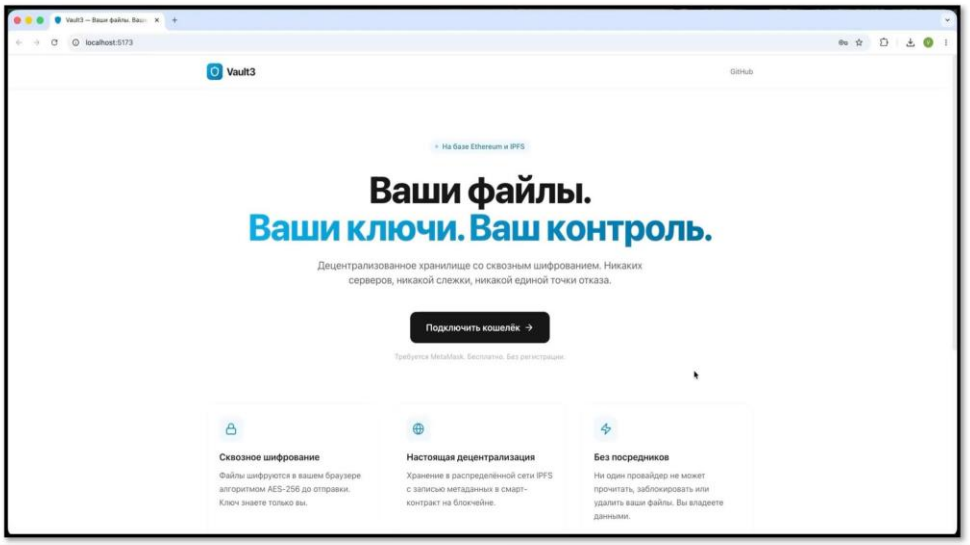

Рисунок 5 — Главная страница приложения Vault3

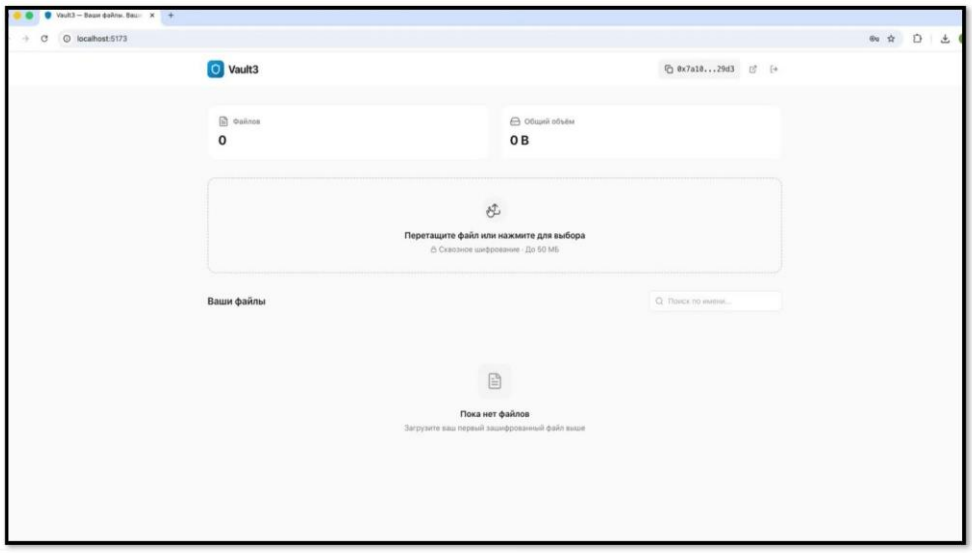

Рисунок 6 — Дашборд пользователя после подключения кошелька MetaMask

Продемонстрировано, что без пароля содержимое файла в IPFS представляет собой нечитаемые зашифрованные данные. Результаты подтвердили корректность работы прототипа и реализуемость выбранного архитектурного подхода.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был разработан прототип распределённой системы хранения файлов на базе протокола IPFS и блокчейна Ethereum, обеспечивающий безопасную загрузку, шифрование, хранение и управление файлами без зависимости от централизованных сервисов.

Перед началом разработки были проанализированы существующие типы файловых хранилищ и выявлены их ключевые ограничения: зависимость от провайдера, единая точка отказа и подверженность цензуре. Исследованы технологии IPFS и блокчейн, проанализированы архитектурные подходы к их совместному использованию.

На основе проведённого анализа были сформулированы требования к прототипу и выбран технологический стек. Разработан смарт-контракт FileStorage на языке Solidity, обеспечивающий хранение метаданных файлов и контроль доступа через блокчейн. Создано клиентское веб-приложение на React с TypeScript, реализующее сквозное шифрование файлов алгоритмом AES-256-GCM.

Архитектура прототипа построена на принципе разделения ответственности между тремя слоями: браузер пользователя выполняет шифрование и взаимодействие с блокчейном, сеть IPFS обеспечивает распределённое хранение зашифрованных файлов, а смарт-контракт в сети Polygon Amoy хранит метаданные и контролирует права доступа. Принципиально важно, что пароль шифрования никогда не покидает браузер пользователя, что гарантирует конфиденциальность данных.

Результаты тестирования подтвердили корректность работы прототипа: загрузка, шифрование, скачивание, поиск и удаление файлов функционируют в соответствии с требованиями. Продемонстрирована невозможность доступа к содержимому файлов без пароля.

Таким образом, цель выпускной квалификационной работы достигнута, а поставленные задачи решены. Разработанный прототип демонстрирует реальную применимость связки технологий IPFS и блокчейн для решения задач безопасного и независимого хранения данных и может быть использован как основа для дальнейшего развития в полноценный продукт.