Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №7**

**Дисциплина: Распределенные программные системы**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Козин

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Шиян

**Цель и задача**

**Цель работы**: смоделировать работу blockchain.

**Постановка задачи**: Разработать свою blockchain сеть с транзакциями и сетевым подключением.

**Введение**

**Blockchain** — это децентрализованная технология распределённого реестра, которая позволяет записывать и хранить данные в блоках, связанных и защищённых с помощью криптографии. Каждый блок в цепочке содержит набор транзакций, а также ссылку на предыдущий блок, образуя непрерывную цепь. Это обеспечивает неизменность и прозрачность данных, делая технологию идеальной для приложений, где важна надёжность и безопасность информации, например, в финансовых системах, смарт-контрактах, системах идентификации и многих других.

**История blockchain** начинается с работы по криптографии и распределённым системам, которые развивались на протяжении нескольких десятилетий. Основополагающие концепции были описаны ещё в 1990-х годах. Однако реальное воплощение технологии blockchain произошло с созданием Bitcoin в 2009 году человеком или группой людей под псевдонимом Сатоши Накамото. Bitcoin представил концепцию децентрализованной валюты, не требующей центрального регулирования, и впервые реализовал blockchain как часть своей инфраструктуры.

С тех пор blockchain развивался и применялся в различных областях, выходя за рамки криптовалют. Это включает в себя разработку платформ вроде Ethereum, которая позволяет создавать децентрализованные приложения и смарт-контракты.

**Конфликтные ситуации** в blockchain обычно связаны с вопросами консенсуса и безопасности. Два наиболее заметных примера включают форки (разветвления) и атаки 51%.

Форки, или разветвления, возникают, когда часть участников сети блокчейна начинает следовать новым правилам, в то время как другие продолжают работать по старым правилам. Это может быть результатом обновлений программного обеспечения или разногласий в сообществе. Форки делятся на два типа: мягкие форки, которые совместимы с предыдущими версиями blockchain, и жёсткие форки, которые создают новую цепь, несовместимую со старой. Жёсткие форки могут привести к разделению сети и созданию двух независимых blockchain.

Атака 51% является одним из самых серьёзных угроз для blockchain, работающих на механизме доказательства работы (Proof of Work). В такой атаке злоумышленник, контролирующий более 50% вычислительной мощности сети, может влиять на процесс добавления новых блоков, в том числе останавливать или обращать транзакции и проводить двойные расходы той же криптовалюты. Это подрывает доверие к сети и может привести к существенным финансовым потерям.

Механизм доказательства работы, применяемый во многих криптовалютах, включая Bitcoin, является основой для достижения консенсуса в децентрализованной сети. Участники сети, называемые майнерами, используют свои вычислительные ресурсы для решения сложных математических задач. Первый майнер, решивший задачу, получает право добавить новый блок в цепочку и вознаграждение в виде криптовалюты. Этот процесс обеспечивает безопасность сети, но также требует значительных вычислительных мощностей и энергии.

Транзакции в blockchain — это записи о передаче данных или ценностей между участниками сети. Они включают в себя отправителя, получателя, количество передаваемых ценностей (например, криптовалюты), а также уникальную криптографическую подпись, гарантирующую подлинность и неизменность транзакции. Транзакции собираются в блоки, которые после проверки участниками сети добавляются в цепочку блоков. Это обеспечивает прозрачность и безопасность, так как изменить данные в одном блоке потребует изменений во всех последующих блоках, что практически невозможно без одновременного контроля над большинством node сети.

**Дерево Меркла** (Merkle tree), также известное как хеш-дерево, является фундаментальным компонентом многих blockchain и криптографических систем. Это структура данных, которая позволяет эффективно и безопасно проверять содержимое больших наборов данных.

В дереве Меркла данные разбиваются на блоки, и каждому блоку присваивается его хеш-значение. Хеш — это уникальный цифровой отпечаток данных, созданный с помощью криптографической хеш-функции. Затем эти хеш-значения объединяются попарно и снова хешируются. Этот процесс повторяется, пока не останется одно хеш-значение, называемое корневым хешем или хешем вершины дерева.

Основные преимущества дерева Меркла:

1. Эффективность: Позволяет быстро и эффективно проверять, содержится ли определенный набор данных в большом объеме информации, не требуя доступа ко всему объему данных.

2. Безопасность: Любые изменения в данных нижнего уровня приведут к изменению хеша на верхнем уровне, что делает дерево Меркла чрезвычайно устойчивым к манипуляциям.

3. Оптимизация: В blockchain дерево Меркла используется для оптимизации верификации транзакций и эффективного хранения данных.

Например, в blockchain Bitcoin дерево Меркла используется для хранения хешей всех транзакций в блоке, что позволяет пользователям и node эффективно проверять наличие конкретных транзакций в блоке без необходимости загружать весь блок. Это особенно полезно для так называемых "легких" клиентов, которые не хранят всю цепочку блоков.

Для реализации проекта использовался **Golang**, известный как **Go**, это язык программирования, созданный в Google. Он спроектирован для быстроты, простоты и эффективности. Go обладает чистым синтаксисом, что делает его легким для изучения и использования. Одной из ключевых особенностей Go является его встроенная поддержка конкурентности через горутины и каналы, позволяющие легко писать многопоточные программы. Это особенно полезно для сетевых приложений и микросервисов. Также Go характеризуется отличной производительностью и масштабируемостью, благодаря чему он популярен среди разработчиков облачных и сетевых решений.

**Base58** — это метод кодирования, часто используемый в криптовалютах и приложениях блокчейна. Он похож на более известное кодирование Base64, но исключает некоторые символы, которые могут вызывать путаницу при визуальном восприятии, например, '0' (ноль), 'O' (большая буква о), 'I' (большая буква и) и 'l' (маленькая буква L), а также символы '+', '/' и '='. Это делает кодированные строки в Base58 более "чистыми" для чтения и использования в URL-адресах. Base58 часто применяется для кодирования биткоин-адресов и других ключей в блокчейн-системах.

**BadgerDB** — это высокопроизводительная, встраиваемая база данных ключ-значение, написанная на Go. Она разработана для предоставления быстрого доступа к большим объемам данных, при этом сохраняя низкое потребление ресурсов. Badger особенно полезна для приложений, требующих высокой скорости записи и чтения, таких как кэширование и блокчейн приложения. Она использует современные методы хранения, такие как LSM-деревья, для оптимизации производительности и эффективности использования диска. BadgerDB предлагает транзакционность, поддержку TTL (время жизни) для ключей и эффективную работу с большими данными, что делает её подходящей для широкого спектра приложений.

**Документация**

**Главный** исполняемый файл проекта эта *main.go*, в нем вызывается командная строка для взаимодействия с пользователем.

Файлы *go.mod* и *go.sum* используются для хранения требований к проекту. Например запрос к нужной базе данных и ее версия.

Файл *cli/cli.go* представляет собой логику командной строки, в ней предоставляется инструкция для работы программы пользователю.

Папка *tmp* и внутри нее папка *blocks* нужны для базы данных badger.

Внутри папки *blockchain* находятся 7 файлов:

1. block.go  
   Предоставляет структуру, создание первого Genesis блока и дальнейших.
2. blockchain.go  
   Предоставляет структуру цепочки блоков и итерации по ним. Используется для реализации функций цепочки блоков.
3. merkle.go  
   Предоставляет структуру дерева Меркла и его узлов.
4. proof.go  
   Предоставляет структуру доказательства подписания блоков и его дальнейшего майнинга.
5. transaction.go  
   Предоставляет структуру транзакций и их подписаний.
6. tx.go  
   Предоставляет структуру для входных и выходных данных для транзакций.
7. utxo.go  
   Предоставляет логику для реиндексации и обновления транзакций.

Папка *wallet* используется для хранения 3 файлов:

1. utils.go  
   Хранит функции для кодировки и декодировки base58.
2. wallet.go  
   Предоставляет структуру кошелька и всю его логику.
3. wallets.go  
   Предоставляет мапу всех кошельков и хранения их в файле.

Для создания блоков был разработан алгоритм шифрования, который состоит из 7 шагов.

ECDSA — это метод цифровой подписи, основанный на эллиптических кривых использующийся в blockchain для обеспечения безопасности и подтверждения подлинности транзакций. Это важный элемент в системах безопасности криптовалют, помогающий убедиться, что транзакции инициируются их законными владельцами.

SHA-256 — это хеш-функция, преобразующая данные в уникальный 256-битный (32-байтовый) хеш. В blockchain она используется для создания уникального отпечатка транзакций и блоков, обеспечивая их целостность и устойчивость к изменениям.

RIPEMD-160 — это ещё одна хеш-функция, преобразующая данные в 160-битный (20-байтовый) хеш. В контексте blockchain, она часто используется после SHA-256 для создания более коротких адресов, таких как адреса биткойн-кошельков.

Base58 — это метод кодирования, используемый в blockchain для преобразования бинарных данных в текстовую форму. Он часто применяется для кодирования адресов и приватных ключей, делая их более компактными и удобными для чтения человеком. Base58 исключает схожие символы, такие как '0' (ноль) и 'O' (большая буква О), чтобы уменьшить вероятность ошибки при вводе.

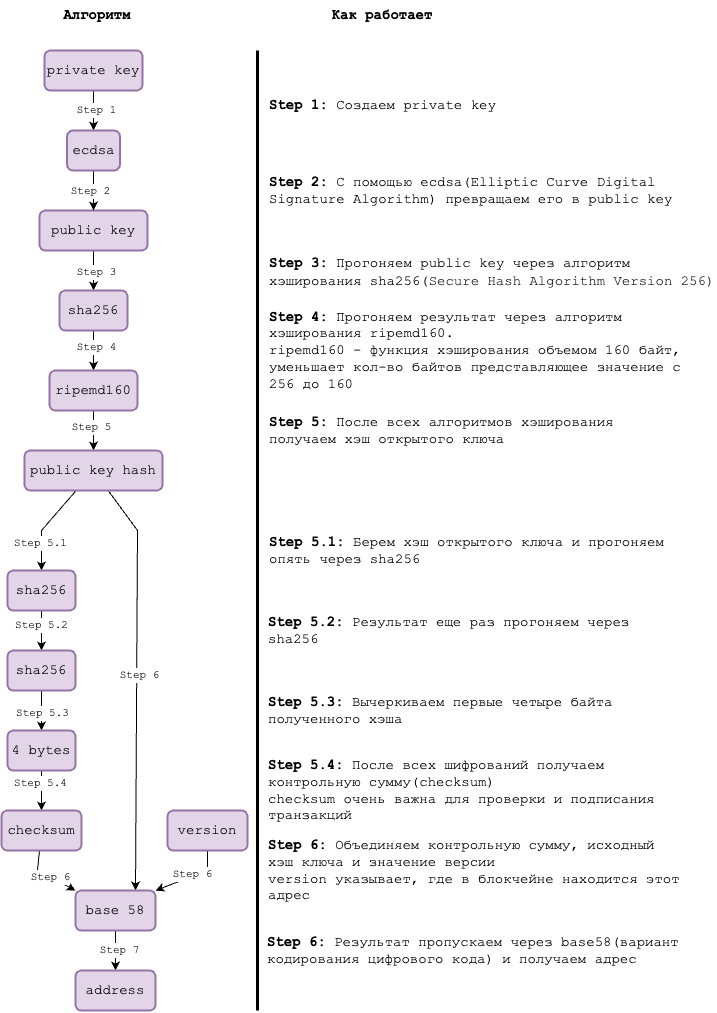


Рисунок 1 – алгоритм шифрования для публичного ключа.

В blockchain кошелек — это цифровой инструмент, который позволяет пользователям хранить и управлять своими криптовалютами и токенами. Он состоит из двух ключевых компонентов: приватного ключа, который используется для подписания транзакций и обеспечения доступа к средствам, и публичного адреса, который другие пользователи используют для отправки криптовалюты в этот кошелек.

A diagram of a private key

Description automatically generated

Рисунок 2 – структура кошелька.

Дерево Меркла — это структура данных в криптографии, используемая в blockchain для эффективной и безопасной организации данных в блоках. Это дерево составляется путем объединения хешей отдельных транзакций в блоке, затем эти хеши объединяются попарно, и процесс продолжается, пока не останется один хеш — корневой хеш дерева Меркла. Этот корневой хеш включается в заголовок блока и служит своеобразной цифровой "печатью" всех транзакций в блоке. Применение деревьев Меркла позволяет быстро и эффективно проверять целостность и наличие транзакций в blockchain, не обрабатывая каждую транзакцию индивидуально.



Рисунок 3 – структура древо Меркла.

**Примеры работы программы**

**Заключение**

Была изучена технология blockchain, его история, транзакции, шифрование. Посредством всего вышеперечисленного была смоделирована blockchain сеть.