Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «**Информатика и системы управления**»

КАФЕДРА «**Программное обеспечение ЭВМ и информационные технологии**»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

Реализация протокола транспортного уровня с поддержкой шифрования данных

Студент группы **ИУ7-32М** _____ **А. В. Романов**
(Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы _____ **А. М. Никульшин**
(Подпись, дата) (И.О. Фамилия)

***2023** г.*

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В этом контексте шифрование данных становится критически важным для обеспечения конфиденциальности и целостности информации. Применение криптографических протоколов и алгоритмов на уровне передачи данных позволяет защитить информацию от несанкционированного доступа и обеспечить ее конфиденциальность. Шифрование данных не только предотвращает возможность прочтения или модификации данных злоумышленниками, но и гарантирует аутентификацию и целостность передаваемой информации. Целью данной курсовой работы является разработка протокола транспортного уровня с поддержкой шифрования данных.

В ходе выполнения курсового проекта необходимо решить следующие задачи:

- провести анализ предметной области;
- спроектировать протокол транспортного уровня с поддержкой шифрования;
- разработать и реализовать данный протокол.

1 Аналитическая часть

В данном разделе приводится краткий обзор предметной области. Описаны протоколы поддерживающие шифрование данных.

1.1 Обзор предметной области

1.1.1 Модель OSI

Модель OSI (Open Systems Interconnection) является концептуальным рамочным протоколом, разработанным Международной организацией по стандартизации (ISO), чтобы стандартизировать связь между различными компьютерными системами [1]. Она была определена в 1984 году и является основным принципом организации и реализации сетевых протоколов.

Модель OSI состоит из семи уровней, каждый из которых выполняет определенные функции для обеспечения надежной и эффективной коммуникации (см. рис 1).

- физический уровень: обеспечивает физическое соединение между устройствами и передачу битов по сети;
- канальный уровень: управляет надежной доставкой данных внутри локальной сети;
- сетевой уровень: обеспечивает маршрутизацию и передачу данных между различными сетями;
- транспортный уровень: отвечает за установление, управление и контроль надежной передачи данных между приложениями;
- сеансовый уровень: управляет установлением, поддержкой и завершением сеансов связи между устройствами;
- представительный уровень: обеспечивает преобразование данных в формат, понятный для приложений;
- прикладной уровень: предоставляет интерфейс для взаимодействия с приложениями.

Модель OSI широко используется при разработке и реализации сетевых

Рисунок 1 – Модель OSI

протоколов, таких как TCP/IP, Ethernet и многих других. Она обеспечивает стандартизацию и согласованность в связи между различными системами и является основополагающей моделью для понимания работы сетевых сред.

1.1.2 Транспортный уровень

Транспортный уровень является третьим уровнем в сетевой архитектуре OSI. Он отвечает за передачу данных между конечными устройствами или хостами в сети. Основной задачей транспортного уровня является обеспечение эффективной и надежной передачи данных.

На транспортном уровне происходит сегментация данных на пакеты, каждый из которых содержит адрес отправителя и получателя, а также другую

необходимую информацию. Пакеты передаются через различные узлы сети до достижения адресата.

1.1.3 Протоколы транспортного уровня

На транспортном уровне используются различные протоколы для обеспечения надежной передачи данных. Наиболее распространенными из них являются протоколы TCP (Transmission Control Protocol) [**?**] и UDP (User Datagram Protocol) [**?**].

TCP является протоколом ориентированным на соединения. Он гарантирует доставку данных в правильном порядке и с контролем ошибок.

— TCP является соединительным протоколом. Он обеспечивает надежную, ориентированную на поток передачу данных между узлами в сети.
— Для установления соединения TCP использует трехстороннее рукопожатие (three-way handshake), включающее отправку и получение пакетов SYN (synchronize) и ACK (acknowledge).
— TCP контролирует порядок пакетов и гарантирует доставку данных без потерь, дублирования или повреждений.
— Обеспечивает контроль нагрузки и управление потоком данных, чтобы избежать перегрузки сети.
— TCP имеет встроенный механизм повторной передачи и контроля ошибок, что гарантирует целостность получаемых данных.

Протокол UDP используется в приложениях, где небольшие задержки более предпочтительны, например, в потоковой передаче видео и аудио. Основные особенности данного протокола:

— UDP является безсоединительным протоколом.
— Не обеспечивает надежную доставку данных, контроль порядка пакетов или ретрансляцию потерянных пакетов.
— UDP обеспечивает минимальное накладные расходы и более быструю передачу данных за счет отсутствия механизмов, используемых в TCP.
— Он является хорошим выбором для приложений, где небольшие задержки

важны, например, в реальном времени видео или голосовой связи.

— Протокол UDP также удобен для широковещательной и многоадресной передачи данных.

— В UDP-пакете нет гарантии доставки, но он прост и эффективен в простых сценариях, где периодическое обновление информации является приемлемым, а небольшие потери данных не критичны.

1.1.4 Шифрование данных

Шифрование данных является важным аспектом безопасности при передаче информации. Оно используется для защиты данных от несанкционированного доступа и предотвращения их изменения или подделки.

Шифрование данных может быть симметричным или асимметричным. В симметричном шифровании используется один и тот же ключ для шифрования и расшифрования данных.

Примерами симметричных алгоритмов шифрования является:

- AES – Advanced Encryption Standard [**?**];
- DES – Data Encryption Standard [**?**].

В асимметричном шифровании используется пара ключей: публичный и приватный. Публичный ключ используется для шифрования данных, а приватный ключ – для их расшифровки. Это обеспечивает большую безопасность, так как приватный ключ хранится в секрете. Ниже представлены примеры наиболее популярных алгоритмов асимметричного шифрования:

- RSA – Rivest-Shamir-Adleman [**?**];
- ECC – Elliptic Curve Cryptography [**?**];
- Diffie-Hellman [**?**].

1.2 Протоколы транспортного уровня с поддержкой шифрования

1.2.1 SSL / TLS

1.2.2 IPSec

1.2.3 DTLS

2 Конструкторская часть

3 Технологическая часть

ЗАКЛЮЧЕНИЕ

В ходе работы над данным проектом был проведён анализ предметной области, разработан блок лексического и синтаксического анализа с явным построением дерева разбора для заданного исходного кода, разработан блок семантического анализа и генерации кода LLVM IR.

В результате чего был реализован компилятор подмножества языка C с использованием ANTLR и LLVM.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Networks | IBM. [Электронный ресурс] – Режим доступа: https://www.ibm.com/docs/no/aix/7.1?topic=networks- – (01.11.2023)

ПРИЛОЖЕНИЕ А

Листинг 1: Сгенерированный LLVM IR код