Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"

Московский институт электроники и математики им. А.Н.Тихонова

Направление подготовки/специальности
10.05.01 «Компьютерная безопасность»
Образовательная программа «Компьютерная безопасность»

О Т Ч Е Т

о выполнении лабораторной работы на тему
«Методы и права доступа в ОС».

Студент Селях С.О. **СКБ-201**
(Фамилия И.О.) номер группы

Лектор Программирование Алгоритмов Защиты Информации:

Нестеренко Алексей Юрьевич, Доцент _____
(Фамилия И.О., должность, подпись)

Работа выполнена на ___ балла (-ов)

Дата 18.12.2024

Москва, 2024

Содержание

Введение

В рамках данной лабораторной работы была разработана клиент-серверная система, обеспечивающая защищённую передачу данных с использованием стандарта шифрования «Кузнечик» (ГОСТ Р 34.12-2015) и вычислением имитовставки (MAC) согласно рекомендациям ГОСТ Р 34.13-2015. Для реализации криптографических функций применена библиотека `libakrypt`, предоставляющая средства для работы с криптографическими стандартами.

Применение `libakrypt` позволило упростить реализацию, используя готовые функции для установки ключа, шифрования, расшифровки и вычисления CMAC. Данная библиотека взята из открытого репозитория https://github.com/CyberFatherRT/libakrypt. Хотя она не проверялась на предмет программных закладок, для учебных целей и в контексте данной работы это является приемлемым решением.

Разработанная система реализует модель «клиент-сервер»: клиент подключается к серверу, отправляет зашифрованное сообщение с имитовставкой, а сервер проверяет MAC, расшифровывает сообщение и отправляет ответ по тому же протоколу. Сетевое взаимодействие реализовано на базе TCP, с учетом корректной обработки ошибок и гарантированной передачи данных.

Цель работы:

- Изучить и применить на практике криптографические стандарты: ГОСТ Р 34.12-2015 (алгоритм «Кузнечик») и ГОСТ Р 34.13-2015 (режимы работы, включая вычисление имитовставки).
- Реализовать клиент-серверное приложение, использующее симметричное шифрование с имитовставкой для передачи сообщений.
- Освоить практику использования библиотеки libakrypt, а также методов сетевого взаимодействия через TCP-сокеты.
- Обеспечить корректный протокол обмена: передача IV, длины сообщения, шифротекста и MAC, проверка аутентичности сообщения и формирование ответа.

Теоретические основы

Симметричное шифрование «Кузнечик»

Алгоритм «Кузнечик» (ГОСТ Р 34.12-2015) — блочный симметричный шифр с размером блока 128 бит и длиной ключа 256 бит. Он обеспечивает высокий уровень криптостойкости при условии корректного выбора режима шифрования и генерации ключей.

Режим CBC и использование IV

Режим CBC (Cipher Block Chaining) обеспечивает стойкость к определенным атакам за счёт включения результата предыдущего блока шифротекста в следующий этап шифрования. Для первого блока используется инициализационный вектор (IV), который должен быть случайным и непредсказуемым. По ГОСТ Р 34.13-2015 IV нельзя повторять для разных сообщений, зашифрованных одним ключом, чтобы избежать детерминизма в шифровании.

При расшифровке CBC используется тот же IV, что и при шифровании. Первый блок расшифровывается, затем результат XOR-ится с IV, восстанавливая исходный текст. Для последующих блоков XOR применяется с предыдущим шифротекстом.

Имитовставка (MAC)

Для контроля целостности и аутентичности сообщений используется имитовставка (MAC). В работе применён режим CMAC по ГОСТ Р 34.13-2015, позволяющий вычислять имитовставку по зашифрованному тексту. Если при проверке MAC на стороне приемника обнаруживается несоответствие, сообщение отвергается.

MAC обеспечивает защиту от активных атак, предотвращая модификацию шифротекста злоумышленником без риска быть обнаруженным.

Описание протокола обмена

Протокол обмена включает передачу IV, длины зашифрованного сообщения, самого шифротекста и MAC. Клиент и сервер используют одинаковые вызовы функций `libakrypt` для вычисления CMAC и выполнения шифрования/расшифровки. Это позволяет легко проверить подлинность и целостность сообщения до расшифровки. Протокол обмена между клиентом и сервером выглядит так:

1. Клиент генерирует IV (16 байт).
2. Клиент дополняет исходное сообщение паддингом до 16-байтового блока.
3. Клиент шифрует сообщение в режиме CBC с ключом и IV.
4. Клиент вычисляет MAC (имитовставку) по шифртексту (CMAC).
5. Клиент отправляет:
    - IV (16 байт)
    - Длину зашифрованного сообщения (4 байта, сетевой порядок)
    - Зашифрованное сообщение (N байт)
    - MAC (16 байт)

Сервер получает данные в том же порядке, проверяет MAC, и только при его совпадении расшифровывает сообщение, выводит его содержимое, затем формирует аналогичный ответ («Принято!»), снова шифрует, вычисляет MAC и отправляет по тому же протоколу.

Практическая реализация

Среда разработки

- Операционная система: Linux (Debian)
- Компилятор: gcc
- Криптографическая библиотека: libakrypt (реализация ГОСТ-алгоритмов)
- Сетевое взаимодействие: сокеты TCP/IP

Сборка проекта:

Используйте следующую команду для компиляции клиентской или серверной программы:

```
gcc [client/server].c -o [client/server] -I/usr/local/include -L/usr/local/lib
-Wl,-rpath=/usr/local/lib -lakrypt -lakrypt-base
```

Здесь флаги -lakrypt и -lakrypt-base указывают на необходимость линковки с библиотеками libakrypt и libakrypt-base.

Ключ и файлы

Для шифрования используется 256-битный ключ, хранящийся в файле key.bin. Перед запуском клиентских и серверных приложений на обеих сторонах должен быть подготовлен одинаковый ключ. Пример генерации:

```
dd if=/dev/urandom of=key.bin bs=32 count=1
```

Структура проекта

- `client.c` — код клиентского приложения.
- `server.c` — код серверного приложения.
- `key.bin` — файл ключа.

Запуск

На сервере:

```
./server
```

На клиенте:

```
./client <IP-адрес сервера>
```

Отладка и проверка работоспособности

В коде реализованы функции `send_all` и `recv_all`, чтобы гарантированно считывать и отправлять указанный объём данных. Это предотвращает ошибки частичного чтения сетевых буферов.

При возникновении ошибок (например, неправильный MAC или неверный размер сообщения) программа выводит соответствующие сообщения и освобождает ресурсы.

Отладка показала, что при корректной настройке (одинаковом ключе и правильном IP) сообщения передаются и расшифровываются успешно. Клиент получает ответ от сервера и выводит его.

Безопасность и соответствие ГОСТ

Использование режима CBC вместе с уникальным IV для каждого сообщения соответствует требованиям ГОСТ Р 34.13-2015 к режимам блочного шифрования.

Вычисление имитовставки (CMAC) также прописано в ГОСТ Р 34.13-2015 для обеспечения целостности и подлинности сообщений. Это предотвращает успешные активные атаки (например, подмену шифротекста злоумышленником).

Паддинг (дополнение до размера блока) также соответствует требованиям ГОСТ, который предписывает заполнение оставшихся байтов блока значением длины дополнения.

Результаты

- Разработано клиент-серверное приложение на языке C с использованием библиотеки libakrypt.
- Применен блочный шифр «Кузнечик» в режиме CBC с IV.
- Добавлена имитовставка (MAC), вычисляемая с помощью CMAC, для обеспечения целостности.
- Реализованы гарантированная передача и прием данных в сетевом протоколе.
- Проверено, что при корректной настройке и использовании одинакового ключа сообщения успешно шифруются, передаются, проверяются на подлинность и целостность, расшифровываются, и клиент получает ожидаемый ответ.

Приложение

Ссылка на репозиторий с кодом:
https://github.com/The-Elder-One/OPZKS-lab/tree/main

В репозитории размещены исходные коды клиента и сервера, а также инструкций по сборке и запуску.