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

Кафедра теоретических основ компьютерной безопасности и криптографии

Система передачи сообщений со сквозным шифрованием

АВТОРЕФЕРАТ
дипломной работы

студентки 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Семёновой Надежды Михайловны

Научный руководитель
ассистент _____ А. А. Лобов
22.01.2022 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Ускорить передачу информации пытались разными способами во все времена: от сигнальных костров и почтовых голубей до телеграфа и интернета.

В век развивающихся технологий и с ростом темпа жизни людям понадобились сервисы быстрой отправки СМС и файлов. Это касается сферы бизнеса, ниши услуг и прочего.

Приложения для обмена сообщениями – это простое средство для поддержки связи с друзьями, семьей и коллегами. Однако при их использовании важно помнить о безопасности в интернете.

Основная проблема безопасности приложений для обмена сообщениями – это конфиденциальность. Она отражает, насколько личные сообщения доступны третьим лицам, какие компании разрабатывают приложения и собирают ли государственные органы данные о гражданах. При оценке безопасности приложений для обмена сообщениями необходимо учитывать данный фактор. Хорошим вариантом является использование сквозного шифрования в мессенджерах. Оно позволяет увидеть отправленные сообщения в расшифрованном виде только участникам двустороннего диалога.

Сейчас люди всё больше беспокоятся о своей приватности, и хотя существует множество мессенджеров, в том числе у больших компаний, не все доверяют этим компаниям. Разработанный продукт можно устанавливать на собственные ресурсы, все данные в нём хранятся локально в зашифрованном виде, об участниках диалога нет никакой информации, кроме зашифрованных сообщений.

Дипломная работа состоит из введения, 5 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 59 страниц, из них 41 страница – основное содержание, включая 27 рисунков, список использованных источников из 21 наименования.

КРАТКОЕ СОДЕРЖАНИЕ

1 Симметричное шифрование

Симметричным считается любой шифр, который использует один и тот же секретный ключ для шифрования и расшифровки.

В зависимости от принципа работы алгоритмы симметричного шифрования делятся на два типа: блочные и потоковые.

Блочные алгоритмы шифруют данные блоками фиксированной длины (64, 128 или другое количество бит в зависимости от алгоритма). К актуальным блочным алгоритмам относятся: AES, ГОСТ 28147-89, RC5 и др.

Симметричное шифрование используется для обмена данными во многих современных сервисах.

1.1 OpenPGP

OpenPGP – это широко используемый стандарт, который описывает формат зашифрованных сообщений.

Стандарт OpenPGP основан на оригинальном программном обеспечении PGP. OpenPGP определяет стандартные форматы для зашифрованных сообщений, подписей и сертификатов для обмена открытыми ключами.

1.2 OpenPGP.js

OpenPGP.js является JavaScript-реализацией стандарта OpenPGP.

OpenPGP.js реализует RFC4880 и части RFC4880bis. Специалисты из немецкой компании Recurity Labs разработали JavaScript-реализацию стандарта OpenPGP (RFC 4880) для подписи и шифрования писем в почтовых веб-интерфейсах.

2 Сквозное шифрование

Сквозное шифрование – это система связи, в которой только пользователи могут читать сообщения.

Сквозное шифрование не позволяет потенциальным прослушивателям получить доступ к криптографическим ключам, необходимым для расшифровки разговора. Только получатель может прочитать отправленное

сообщение, т.к. весь путь от отправителя до получателя сообщение проходит в зашифрованном виде.

3 Протоколы HTTP и WebSocket

HTTP и WebSocket – это протоколы связи, используемые в коммуникации клиент-сервер. Существуют разные способы передачи данных от браузера или приложения к серверам и обратно. Правила для этих способов передачи данных описаны в специальных протоколах.

3.1 Протокол HTTP

HTTP является однонаправленным протоколом, т.е. клиент отправляет запрос, а сервер отправляет ему ответ.

Каждое сообщение HTTP-запроса состоит из версии протокола, методов, заголовков, информации о хосте и т.д. Также HTTP-запрос имеет тело, которое содержит фактическое сообщение, передаваемое на сервер. Заголовки HTTP варьируются от 200 байтов до 2 КБ, общий размер заголовка HTTP составляет 700-800 байтов.

3.2 Протокол WebSocket

Протокол WebSocket, описанный в спецификации RFC 6455, обеспечивает возможность обмена данными между браузером и сервером через постоянное соединение. Данные передаются по этому соединению в обоих направлениях в виде “пакетов”, без дополнительных HTTP-запросов и разрыва соединения.

Протокол WebSocket используется в игровых и трейдерских приложениях, в чатах, в социальных сетях и т.д. WebSocket подходит для этих проектов лучше всего, так как в них клиент может не выполнять на своей стороне никаких вычислений, а лишь получать\передавать данные на сервер.

3.3 Сравнение HTTP и WebSocket

Протоколы HTTP и WebSocket основаны на TCP и являются надёжными протоколами передачи. Оба этих протокола являются протоколами прикладного уровня.

По сравнению с протоколом HTTP, протокол WebSocket устанавливает такой режим связи, при котором как только соединение WebSocket установлено, последующие данные передаются в виде последовательности кадров. Перед тем, как клиент отключит соединение WebSocket или серверное соединение отключится, клиенту и серверу не нужно повторно инициировать запрос соединения.

3.4 Библиотека PHP Workerman для WebSocket

Workerman – это асинхронный PHP-фреймворк, управляемый событиями, с высокой производительностью для создания быстрых и масштабируемых сетевых приложений. Для установки WorkerMan нужно воспользоваться Composer и прописать следующую команду: composer require workerman/workerman.

4 JSON

JSON (*JavaScript Object Notation*) – это простой формат обмена данными, удобный для чтения и написания как компьютером, так и человеком. JSON – это текстовый формат, который полностью независим от языка реализации, но он использует соглашения, знакомые программистам C-подобных языков, таких как C, C++, C#, Java, JavaScript, Perl, Python и многих других.

5 Реализация системы передачи сообщений со сквозным шифрованием

В ходе данной работы была реализована система передачи сообщений со сквозным шифрованием. Для её реализации были использованы следующие технологии: HTML, CSS, JavaScript, Vue.js, Vue Router, WebSocket, Workerman, PHP PDO, SQLite и OpenPGP.

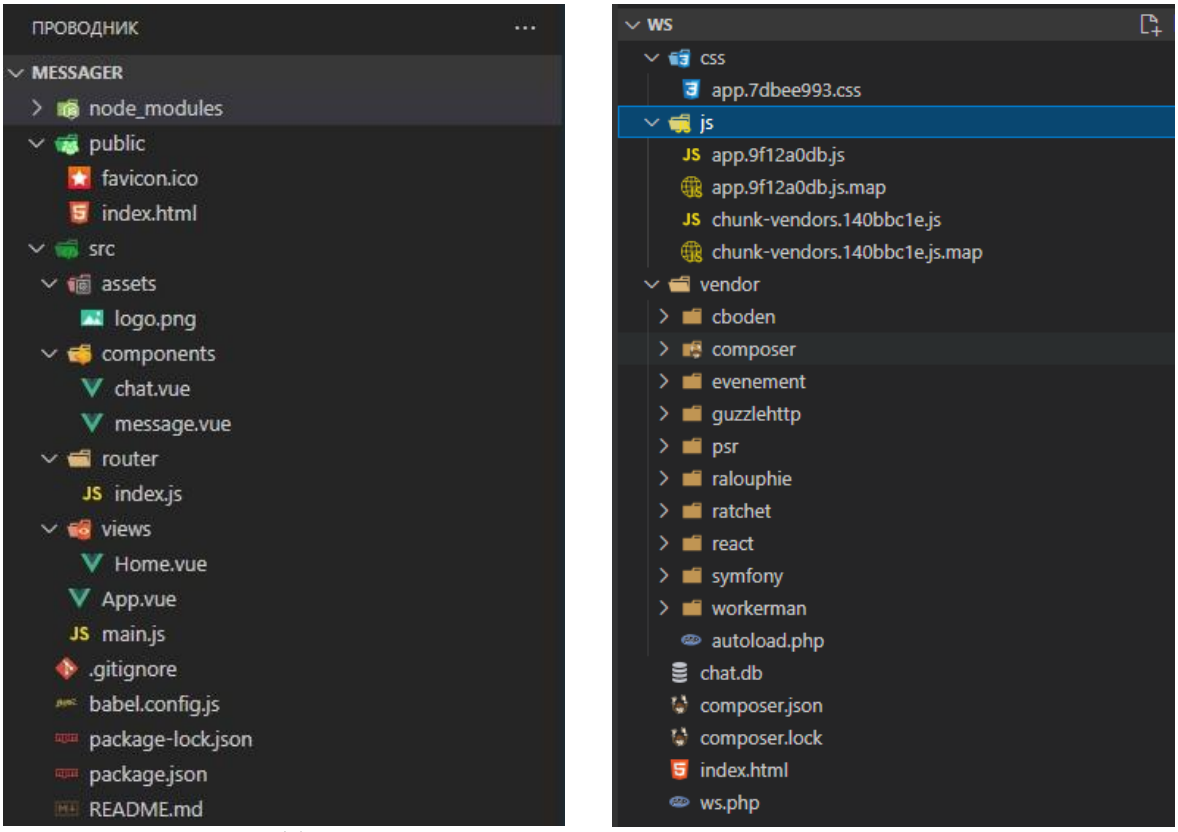
Работа включает в себя сервер, написанный на языке программирования PHP с использованием протокола WebSocket и библиотеки Workerman. Сервер принимает и отправляет данные в формате JSON. Для хранения сообщений используется реляционная база данных SQLite. Для шифрования сообщений используется JavaScript реализация

стандарта OpenPGP с открытым исходным кодом OpenPGP.js. Для организации пользовательского интерфейса используется фреймворк Vue.js, написанный на языке программирования JavaScript.

5.1 Демонстрация работы системы передачи сообщений со сквозным шифрованием

На рисунке 5.1.1а показана структура проекта.

Чтобы скомпилировать проект, необходимо прописать специальную команду в консоли: npm run build. В результате выполнения данной команды в папке messager создаётся папка dist, её содержимое копируется в папку сервера ws. Содержимое папки ws, после выполнения этого шага, показано на рисунке 5.1.1б.



(а) (б)

Рисунок 5.1.1 – Дерево решения и содержимое директории ws

Для хранения отправленных сообщений в проекте используется реляционная база данных SQLite, основанная на языке SQL. Для создания базы данных в консоли были прописаны специальные команды, которые представлены на рисунке 5.1.2.

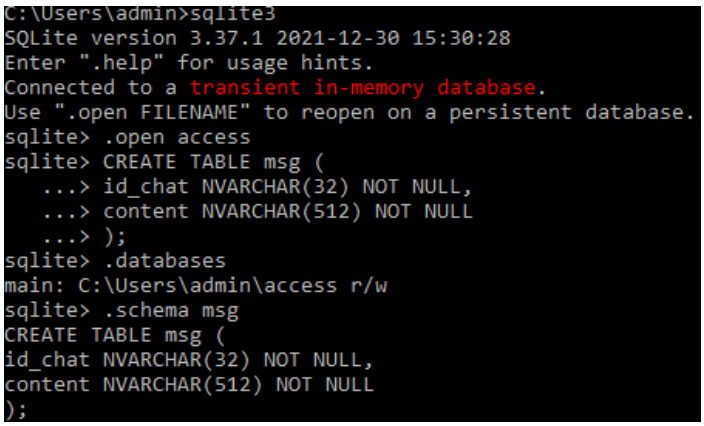


Рисунок 5.1.2 – Создание базы данных

Для запуска проекта нужно прописать команду php -S localhost:3000. Чтобы запустить web-сервер, необходимо прописать команду php ws.php start. Эти команды представлены на рисунках 5.1.3 и 5.1.4 соответственно.

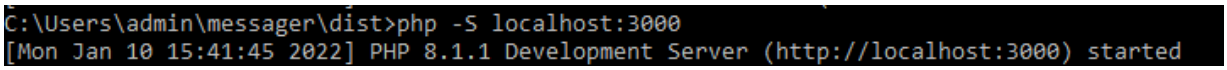


Рисунок 5.1.3 – Запуск проекта

Рисунок 5.1.4 – Запуск web-сервера

При открытии сайта пользователь попадает на основную страницу, где он может создать новый чат или войти с помощью соответствующего пароля в уже существующий чат. Главная страница сайта приведена на рисунке 5.1.5.

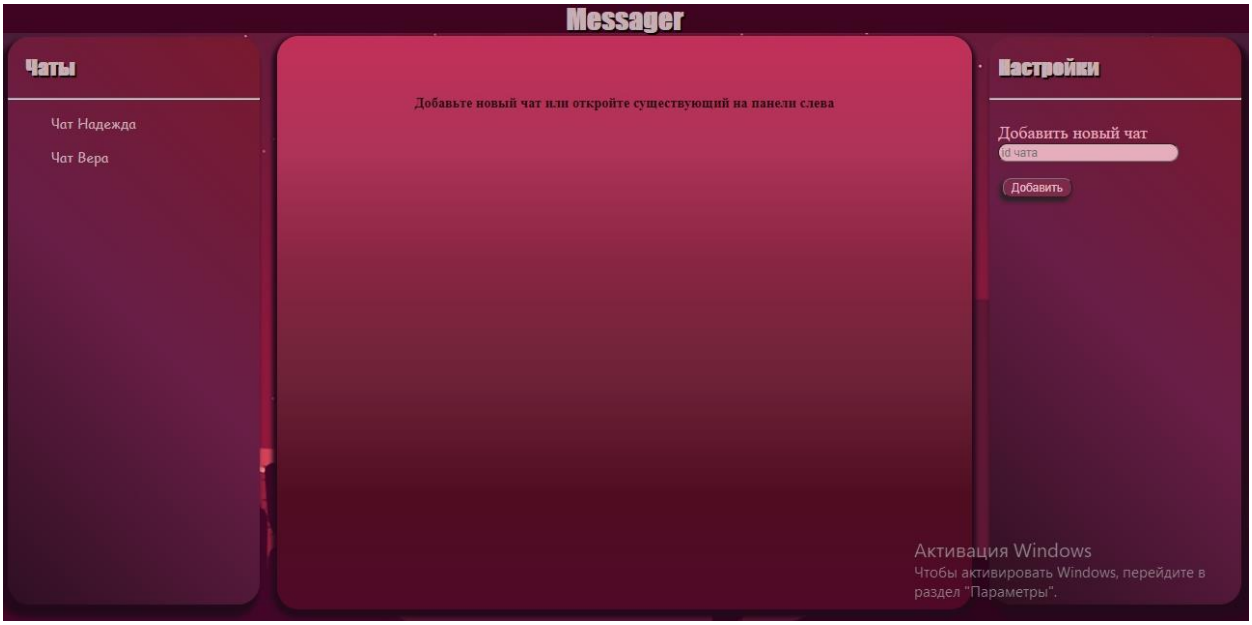


Рисунок 5.1.5 – Основная страница сайта

Для того чтобы войти в чат, пользователь должен знать соответствующий пароль, который будет использоваться OpenPGP для построения ключа сквозного шифрования сообщений. Если пользователь верно ввёл пароль, то он получает доступ к чату и может отправлять и получать сообщения. При отправке и получении сообщений данные предоставляются в расшифрованном виде, так как оба пользователя имеют пароль для сквозного шифрования. Однако на сервере эти данные хранятся в зашифрованном виде, что обеспечивает безопасную переписку между пользователями. Пример хранения данных на сервере представлен на рисунке 5.1.6.

Рисунок 5.1.6 – Пример хранения сообщений на сервере

При попытке подключиться к чату, не зная соответствующего пароля, пользователь попадёт на страницу данного чата, однако все сообщения будут представлены в зашифрованном виде, а имена пользователей чата и время отправки сообщений будут неизвестны.

Для создания нового чата нужно ввести в соответствующее поле название чата, которое будет служить его идентификатором.

Для проверки работы базы данных был произведен вход в чат "София" от двух разных пользователей, которым известен соответствующий пароль. После успешного входа в чат, пользователями были прописаны несколько сообщений, которые должны сохраниться в базе данных. Затем сервер был отключен и подключен заново. В специальной консоли sqlite3 были прописаны команды для отображения содержимого базы данных chat.db, в частности, таблицы msg. Результат выполнения этих команд представлен на рисунке 5.1.7.

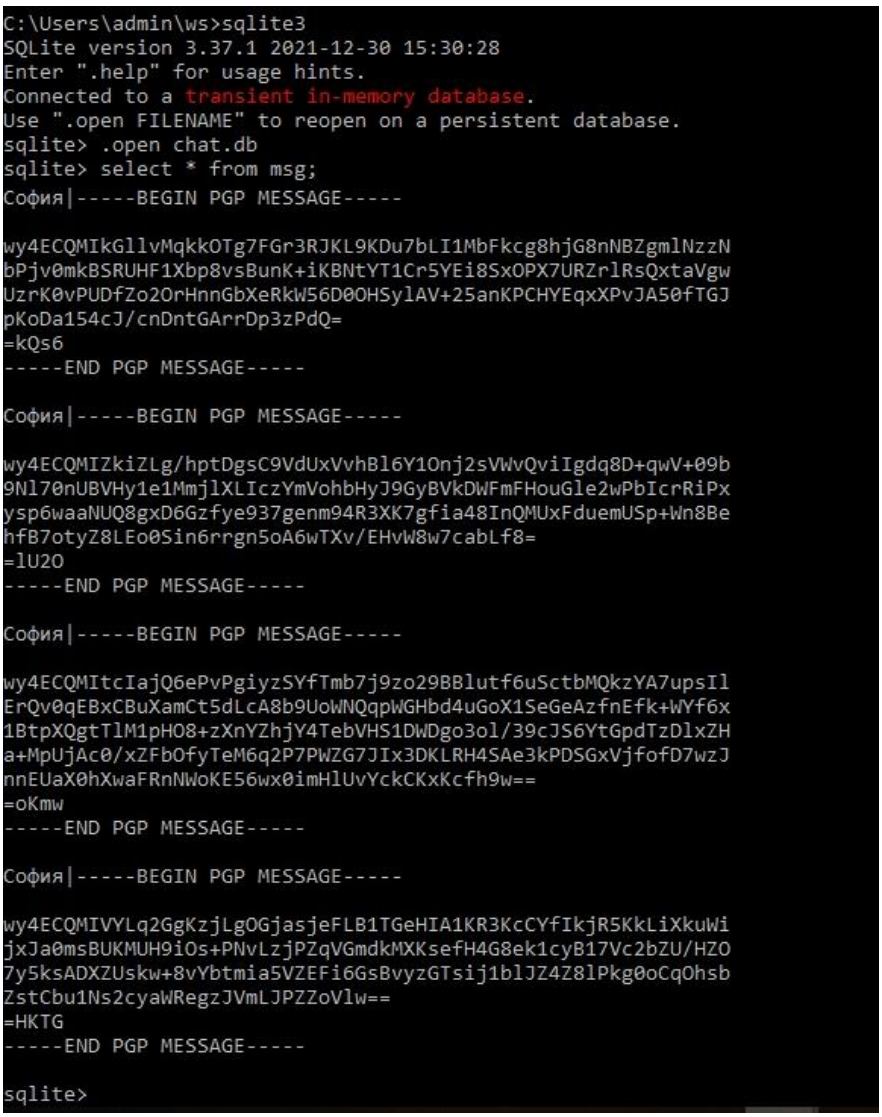


Рисунок 5.1.7 – Содержимое таблицы msg базы данных chat.db

ЗАКЛЮЧЕНИЕ

В ходе работы была разработана система передачи сообщений со сквозным шифрованием в виде интерактивного web-приложения, которое состоит из сервера, базы данных для хранения сообщений и пользовательского интерфейса.

При разработке веб-приложения были использованы следующие технологии и библиотеки: WebSocket – для организации оповещения о новом сообщении, OpenPGP.js – для шифрования сообщений, Workerman – для реализации WebSocket-сервера, Vue.js – для организации пользовательского интерфейса и SQLite – для хранения сообщений.

Данное приложение можно использовать для конфиденциального общения.