УДК 004.62

# АНАЛИЗ ПРОБЛЕМЫ БЕЗОПАСНОСТИ ХРАНЕНИЯ И ТРАНСПОРТИРОВКИ ИНФОРМАЦИИ И ПОДХОДОВ К ЕЕ РЕШЕНИЮ НА ПРИМЕРЕ ПРОГРАММНОГО СЕРВИСА ОБМЕНА ЗАЩИЩЕННЫМИ ДАННЫМИ

Нехорошева Е.М., магистрант гр. ПИм-221, II курс
Медведчиков М.Н., магистрант гр. ПИм-221, II курс
Научный руководитель: Пимонов А.Г., д.т.н., профессор
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Хранение информации на централизованных серверах представляет наибольшую уязвимость в сфере данных. Неохраняемые объемы информации, включая логины, пароли и данные кредитных карт, легко проникают в открытую сеть. Для защиты таких данных финансовые учреждения тратят огромные средства. Однако порой конфиденциальная информация всё равно попадает в руки злоумышленников. Один из недавних и крупных случаев утечки данных произошел в Сбербанке в марте 2023 года, когда сотрудники получили доступ к данным и могли их незаметно скопировать [1], подвергая конфиденциальность всего банка опасности.

Исходя из вышеизложенного, можно заключить, что для максимальной защиты данных необходимо осуществить их распределение (децентрализацию) в хранилищах. Эти сведения должны быть зашифрованы и размещены в различных уголках мира с предоставлением лимитированного доступа к ним. Идеальным вариантом является разделение информации и ее хранение на отдельных серверах без физического доступа к ним. Безопасность переписки также играет важную роль, поскольку текст сообщений может содержать конфиденциальные данные, требующие в обращении особой осторожности.

Информацией, о которой пойдет речь и которую разрабатываемый проект защищает, является личная переписка пользователей. Проект ставит своей целью полную защиту информации от несанкционированного доступа на этапах хранения и транспортировки данных.

Источниками информации являются «банки данных». Банками данных называются устройства пользователей, подключенные к системе, и хранящие зашифрованные части информации. Доступ к ним осуществляется с помощью других устройств, которые хотят восстановить и отобразить информацию.

В классических клиент-серверных приложениях сервер занимается хранением и обработкой всех поступающих в него данных. Взаимодействие с ним чаще всего происходит по одному из двух протоколов: API и Socket. В любом случае для того чтобы отобразить информацию на другом устрой-

стве необходимо сохранить ее на сервере, что не является безопасным для конфиденциальности информации, так как к серверу имеют доступ сотрудники, а также возможен доступ извне со стороны злоумышленников. Классический пример структуры системы передачи данных представлен на рисунке 1.

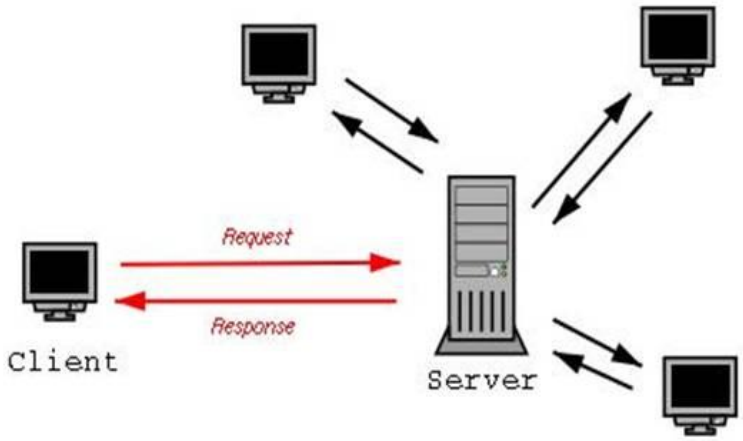


*Рисунок 1 – Структура клиент-серверного приложения*

В разрабатываемом проекте используется следующий стек технологий:

**Инструменты для реализации клиента:**

1. Java/Kotlin – язык программирования;
2. Android SDK – библиотека для взаимодействия с системой Андроид;
3. javax.crypto – шифрование;
4. Tor Server – маршрутизация [2] посредством Tor-сетей;
5. Realm – локальная БД для ОС Андроид.
6. GSON – парсер JSON-файлов.
7. Moxy – библиотека для работы с паттерном MVP (Model-View-Presenter);
8. OkHttp – библиотека для работы с сетью;
9. Reftrofit 2 – библиотека для работы с REST-запросами.

**Инструменты для реализации сервера:**

1. PHP – язык программирования;
2. MySQL – система управления базой данных.

Взаимодействие между клиентами осуществляется средствами TOR-сети [3]. Каждый клиент, по сути, является сервером, к которому подключается другой посредством Socket-соединения.

В случае ведения переписки данные не сохраняются в локальный БД Realm, а попадают в модуль разбиения/восстановления. После чего полученные данные необходимо расшифровать. Этим занимается библиотека Crypto. Сообщения, попавшие в модуль, могут быть отображены в читаемом виде.

Для отправки сообщения его необходимо передать из модуля сообщений в библиотеку Crypto, далее в модуль разбиения/восстановления. После чего сообщение отправляется частями на список устройств. Сам список создается модулем с сервера один раз и хранится на устройстве создателя беседы.

На устройстве пользователя хранятся зашифрованные части сообщений принадлежащих этой или другим беседам. Для этого устройству передается зашифрованная часть сообщения через его TOR-сервер. Эта часть записывается в локальную БД Realm и предоставляется при запросе.

Для доступа к системе используется простейшая система авторизации по номеру телефона и паролю. От несанкционированного доступа к методам сервера и TOR-сервера их API защищено динамическим ключом доступа.

Общая схема решения проекта представлена на рисунок 2.

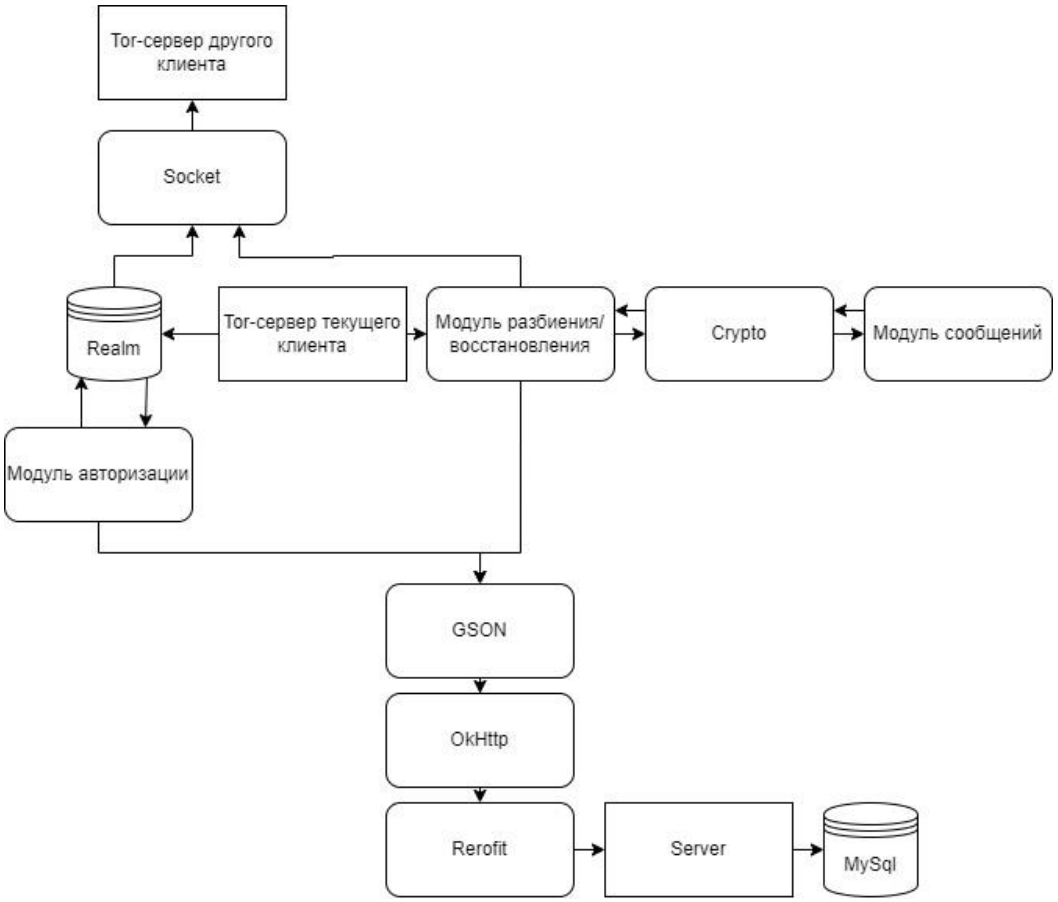


*Рисунок 2 – Схема клиентской части Системы*

Для хранения информации в рамках проекта используется адаптация системы хранения данных RAID 5. Она обеспечивает достаточно высокую отказоустойчивость и достаточно низкое потребление физической памяти. Адаптация отличается от оригинала тем, что в качестве жестких дисков выступают мобильные телефоны («банки данных»).

Для передачи информации используется псевдо p2p соединение посредством TOR-сети.

Сервер используется лишь как «дорожная карта» и платформа авторизации. Он знает лишь адреса банков памяти, которыми и являются клиенты, а также на нем хранятся хешированные номера телефонов и пароли. Сервер не знает информацию, которая хранится на клиентах, так как она идет в обход сервера, что и предотвращает возможность ее утечки.

Схема взаимодействия компонентов проекта представлена на рисунке 3.

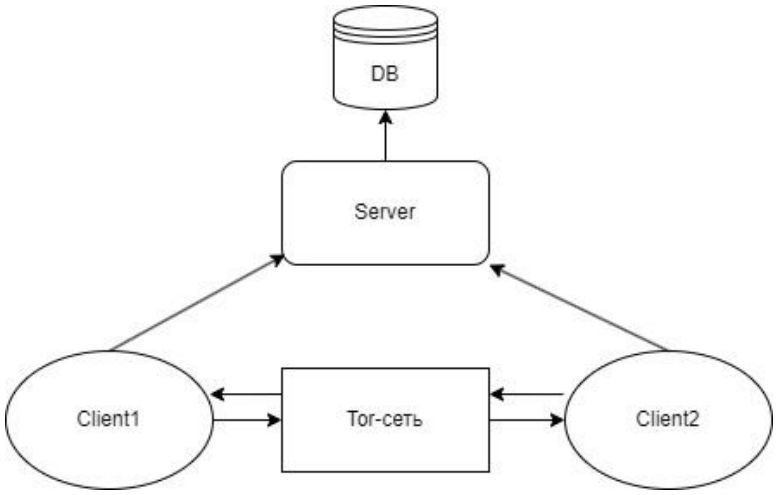


*Рисунок 3 – Схема взаимодействий компонентов проекта*

В разрабатываемом приложении пользовательский интерфейс программы предоставляет следующие возможности:

**1. Регистрация**

Регистрация доступна любому пользователю Системы. Для того чтобы зарегистрироваться необходимо на экране авторизации кликнуть по кнопке `Регистрация`. После чего пользователю откроется форма регистрации, в поле `Номер телефона` необходимо ввести свой номер телефона, в поле `Логин` необходимо ввести желаемый логин пользователя. Поле `Пароль` необходимо для ввода пароля, который необходимо придумать и ввести туда, а также продублировать его в поле `Подтвердите пароль`. После того как все поля заполнены, пользователь должен кликнуть по кнопке `Зарегистрироваться`. В случае успешной регистрации пользователю станет доступен основной функционал Системы.

**2. Авторизация**

Авторизация осуществляется на экране авторизации. Чтобы пройти авторизацию пользователю необходимо ввести свои данные, а именно логин и пароль. В случае, если пользователь по каким-то причинам не может вспомнить пароль, ему доступна функция восстановления доступа к аккаунту. Чтобы попасть на экран восстановления необходимо кликнуть по кнопке `Восстановить доступ`, после чего ввести в поле `Номер телефона` номер своего телефона, указанный при регистрации, и дождаться СМС с кодом подтверждения, который необходимо ввести в поле `Код подтверждения`.

**3. Управление диалогами**

Экран управления диалогами реализует несколько сценариев:

- поиск диалога;
- открытие диалога;
- создание диалога.

Для поиска нужного диалога необходимо воспользоваться полем поиска в верхней части экрана, начав вводить туда логин контакта, диалог с которым необходимо найти.

Для открытия диалога необходимо найти карточку с нужным контактом и кликнуть по ней.

Для создания диалога необходимо нажать на кнопку **+** в правом верхнем углу экрана, далее в открывшейся форме ввести логин желаемого контакта. После этого пользователь будет перемещен в созданный диалог.

Для удаления диалога необходимо найти желаемый диалог, после чего нажать на кнопку в виде красного крестика в правой части его карточки.

Экран ведения диалога представляет из себя историю диалога, в которой сообщения пользователя помещаются слева, а сообщения его собеседников – справа. Чтобы отправить сообщение необходимо использовать панель ввода, находящуюся в нижней части экрана. После того как сообщение написано, необходимо кликнуть по кнопке `Отправить`. Чтобы удалить сообщение необходимо нажатие на него, удерживая в течение трех секунд, и во всплывающем меню выбрать пункт `Удалить`.

Преимуществами данного решения являются:

- **Низкие затраты на сервер.**

Сервер выполняет роль дорожной карты и системы авторизации, т.е. хранит лишь данные о пользователе и о его сессиях.

- **Практически бесконечный объем памяти.**

Самый большой пласт информации распределенно хранится на устройствах клиентов. Каждый новый клиент увеличивает объем памяти без затрат со стороны команды поддержки.

- **Высокая отказоустойчивость.**

Система RAID 5 позволяет восстановить данные в случае отказа двух или менее банков данных. Клиент будет с определенным промежутком отсылать на сервер информацию о своей доступности, что позволит оперативно обновлять список банков данных, в которых хранятся данные переписок.

- **Почти полная конфиденциальность.**

Части системы являются изолированными друг от друга, что делает невозможным доступ к информации без получения доступа ко всем частям системы.

**Список литературы:**

1. Утечка данных из банков России // TAdviser. – URL: https://www.tadviser.ru/index.php/ (дата обращения: 25.03.2024).

2. Dingledine R., Mathewson N., Syverson P. Tor: Луковый маршрутизатор второго поколения. – URL: http://www.opennet.ru/soft/tordesign.pdf (дата обращения: 25.03.2024).

3. Создание скрытого сервиса в TOR наподобие Silk Road или DarkNet. – URL: https://codeby.net/blogs/sozdanie-skrytogo-servisa-v-tor-napodobii-silk-road-ili-darknet/ (дата обращения: 26.03.2024).