УДК 004.056:52 doi: 10.15827/0236-235X.142.230-237 2024. Т. 37. № 2. С. 230–237

Безопасный обмен файлами на основе сетей доверия и сертификатов открытых ключей с помощью разработанного приложения

П.Б. Хорев [1]✉, Д.А. Лосев [1]

[1] Национальный исследовательский университет «Московский энергетический институт», г. Москва, 111250, Россия

Ссылка для цитирования
Хорев П.Б., Лосев Д.А. Безопасный обмен файлами на основе сетей доверия и сертификатов открытых ключей с помощью разработанного приложения // Программные продукты и системы. 2024. Т. 37. № 2. С. 230–237. doi: 10.15827/0236-235X.142.230-237
Информация о статье
Группа специальностей ВАК: 2.3.5
Поступила в редакцию: 29.09.2023 После доработки: 18.12.2023 Принята к публикации: 25.12.2023

Аннотация. В статье анализируются существующие программные средства обмена файлами (в том числе облачные хранилища, программы-мессенджеры, электронная почта), выделены их недостатки. Для устранения этих недостатков авторами разработан проект клиент-серверного приложения, основанного на построении сети доверия с помощью сертификатов открытых ключей ее участников. В проекте использованы алгоритмы симметричной и асимметричной криптографии, методы построения сетей доверия и применения сертификатов открытых ключей для обеспечения подлинности, конфиденциальности и целостности данных. Реализованы серверная часть и клиентское мобильное приложение, позволяющие пользователям обмениваться файлами внутри одного конкретного файлового хранилища. Загружаемые файлы имеют электронные подписи их создателей и хранятся на сервере в зашифрованном виде. При разработке приложения использовалась технология, облегчающая его перенос на другие операционные платформы. Клиентское приложение создано и протестировано на устройствах под управлением операционной системы Android. Оно обладает необходимой для файлообменника функциональностью и обеспечивает подлинность, целостность и конфиденциальность передаваемых файлов. Это позволяет использовать разработанное приложение как группами частных пользователей, так и в корпоративных информационных системах.

Ключевые слова: клиент-серверное приложение, сертификаты открытых ключей, сеть доверия, безопасный обмен файлами

Введение. Задача обеспечения безопасного хранения и обмена файлами встала практически сразу после появления Интернета. Основное внимание при этом уделялось и уделяется до сих пор защите данных при их хранении на сервере с помощью шифрования, разграничения доступа к файлам на уровне пользователей и физической защиты сервера. Безопасность загрузки и выгрузки файлов с сервера и на сервер обычно обеспечивается использованием протокола HTTPS, являющегося объединением протокола передачи гипертекста HTTP и протоколов безопасной передачи данных SSL/TLS. Протокол HTTPS предполагает обязательное подтверждение подлинности (аутентификацию) сервера, а аутентификация клиента может быть дополнительной опцией, которая требует наличия у клиента так называемого сертификата открытого ключа, выданного удостоверяющим центром, которому доверяет сервер. Аутентификация клиента может быть осуществлена сервером после установки защищенного соединения с помощью одного из традиционных методов (например, по имени и паролю пользователя). Надежность такой аутентификации полностью определяется длиной, сложностью, регулярной сменой и безопасным использованием пароля пользователя.

В качестве примеров решений для обмена данными между пользователями Интернета можно рассмотреть облачные хранилища, программы-мессенджеры и электронную почту. Однако эти средства не всегда удобны и безопасны. Например, в социальной сети «ВКонтакте» есть ограничения на расширение файлов и на их размер, облачные хранилища имеют ограниченное дисковое пространство, а приложения электронной почты могут блокировать передачу исполнимых файлов (в том числе внутри архивов). Кроме того, использование ссылок на файлы в облачных хранилищах может привести к утечке информации, если эти ссылки станут известны посторонним лицам. Доступ к аккаунтам пользователей мессенджеров могут получить другие лица в результате человеческой ошибки владельцев аккаунтов или использования ими недостаточно стойкого к подбору пароля.

В ряде работ (например, [1, 2]) описываются угрозы безопасности данных при их обмене

между пользователями глобальной сети, анализируются достоинства и недостатки применяемых программных решений. В них не предлагаются методы устранения отмеченных недостатков, а для обеспечения безопасности данных рекомендуется использовать защищенные протоколы передачи файлов (HTTPS, SFTP и др.), наложение ограничений на минимальную длину и сложность паролей пользователей, двухфакторную аутентификацию пользователей (например, с помощью пароля и кода из SMS-сообщения на номер телефона, связанного с аккаунтом пользователя, как в сервисе «Яндекс. Почта»).

Однако эти решения имеют ряд недостатков:

– шифрование и контроль целостности файлов обеспечиваются только при их передаче, но не при хранении на стороне сервера;

– не гарантируется реальная регулярная смена пароля пользователя;

– двухфакторная аутентификация может быть неудобной для пользователя (например, из-за увеличения времени на обмен файлами);

– существует риск распространения недостоверной информации посторонними лицами, подобравшими или угадавшими пароль, получившими доступ к мобильному устройству пользователя и (или) применяемому им клиентскому приложению.

В связи с этим возникает необходимость в создании более защищенных файлообменников, которые позволят пользователям безопасно обмениваться файлами с сохранением удобств, предоставляемых другими подобными средствами. Предлагаемое решение основано на построении сети доверия пользователей файлообменника с применением возможностей, предоставляемых современной криптографией. Разработанное приложение имеет серверную и клиентскую части и допускает простой перенос на другие операционные платформы. Возможными применениями разработанного приложения могут быть организация и проведение учебного процесса в университетах, а также обмен конфиденциальными файлами между частными лицами и в информационных системах малого и среднего бизнеса.

Методы создания сетей доверия и использования сертификатов открытых ключей

В основе обеспечения безопасного хранения и обмена файлами в сети Интернет лежат методы современной криптографии. Традиционная симметричная криптография используется для обеспечения конфиденциальности данных путем их шифрования с помощью одного, известного отправителю и получателю данных секретного ключа (его безопасная передача от отправителя к получателю является основной проблемой симметричной криптографии). Симметричные криптосистемы используются для шифрования данных любого объема при их хранении и передаче.

Асимметричная криптография использует два ключа – открытый (публичный) и закрытый, который хранится у владельца и никому не должен передаваться. Открытый ключ используется для шифрования секретного ключа симметричного шифрования, которым защищены передаваемые данные, и для проверки электронной подписи под ними. Закрытый ключ применяется для расшифрования секретного ключа симметричного шифрования и вычисления электронной подписи.

Подлинность открытого ключа (его принадлежность конкретному владельцу двух ключей асимметричного шифрования) подтверждается сертификатом открытого ключа, выданным удостоверяющим центром (центром сертификации). Список доверенных удостоверяющих центров должен вестись у каждого пользователя сети Интернет на каждом его устройстве.

Для создания сети доверия пользователей файлообменника могут применяться методы и средства инфраструктуры открытых ключей (Public Key Infrastructure, PKI) [3], открытого стандарта OpenPGP (Open Pretty Good Privacy) [4] и его свободно распространяемой реализации GnuPG (GNU Privacy Guard) [5], технологии Blockchain [6].

Инфраструктура открытых ключей предоставляет возможность создавать, распространять, использовать сертификаты открытых ключей, которые позволяют аутентифицировать данные и пользователей, управлять ими, а также контролировать доступ к защищенным ресурсам в сети. В основе PKI лежит асимметричная криптография. Сертификаты открытого ключа пользователя, выданные доверенными центрами сертификации, являются ключевым компонентом PKI, поскольку удостоверяют подлинность ключа. Сертификат содержит информацию о пользователе, его открытом ключе, сроке действия сертификата и другую информацию, необходимую для аутентификации.

Достоинства PKI:

– обеспечение подлинности и целостности файлов с помощью их электронной подписи;

– обеспечение конфиденциальности файлов с помощью их шифрования;

– аутентификация пользователей с помощью проверки знания ими закрытых ключей, связанных с открытыми ключами в их сертификатах;

– совместимость с различными программными системами за счет использования международных стандартов X.509 и PKCS (Public Key Cryptography Standards);

– возможность управлять доступом пользователей к файлам, применяя различные уровни прав доступа.

Недостатки PKI:

– сложность создания, настройки и управления (особенно для частных лиц и организаций, не имеющих достаточных компетенций в сфере информационных технологий);

– сложность безопасного обновления сертификатов и их отзыва до истечения срока действия;

– риск компрометации закрытых ключей пользователей (особенно при их генерации на сервере);

– необходимость в дополнительных затратах на приобретение лицензионного ПО.

Открытый стандарт OpenPGP предоставляет инфраструктуру для создания ключей шифрования и управления ими, а также для защиты электронной почты и других данных. OpenPGP использует асимметричную криптографию и пары открытого и закрытого ключей пользователей для защиты их данных. GnuPG − бесплатная реализация OpenPGP, позволяющая пользователям шифровать и подписывать свои данные и сообщения, используя стандарты асимметричной криптографии. GnuPG − проект GNU, доступный бесплатно в большинстве ОС, включая Linux, MacOS и Windows.

Сертификаты открытого ключа в OpenPGP содержат информацию о владельце ключа, включая его имя и адрес электронной почты, дату создания сертификата и др. Они также могут содержать информацию о том, кто подтвердил имя владельца ключа (возможно, что сертификат подтвержден более чем одним лицом).

Достоинства OpenPGP и GnuPG:

– обеспечение подлинности, целостности и конфиденциальности файлов;

– отказ от использования доверенных удостоверяющих центров (центров сертификации) для выпуска сертификатов открытых ключей пользователей;

– открытость исходного кода с реализацией.

Недостатки OpenPGP и GnuPG:

– сложность использования для частных лиц и небольших организаций;

– сложность построения надежной сети доверия (в том числе и из-за транзитивного доверия, которое может привести к недостаточно надежному подтверждению подлинности открытого ключа участника сети доверия).

Блокчейн − технология создания и использования распределенной БД, информация в которой хранится в виде блоков, соединенных между собой цепочкой. Каждый блок содержит некоторую информацию (например, о завершенной транзакции) и имеет уникальный идентификатор, называемый хешем. Одной из основных особенностей этой технологии является децентрализация: информация хранится не на централизованном сервере, а на многих компьютерах, называемых сетевыми узлами. Каждый узел содержит копию всей цепочки, что обеспечивает безопасность и надежность хранения данных.

Сеть доверия на основе технологии Blockchain строится с помощью алгоритма консенсуса, который позволяет участникам сети достигать единства относительно состояния распределенного хранилища.

Достоинства технологии Blockchain:

– децентрализация файлового хранилища и управления сетью доверия;

– обеспечение подлинности, целостности и конфиденциальности файлов.

Недостатки технологии Blockchain:

– сложность использования частными лицами и небольшими организациями;

– сложность масштабирования;

– возможность дополнительных временных задержек при обработке транзакций из-за того, что новые блоки добавляются после выполнения сложных математических вычислений;

– необходимость достижения консенсуса участников сети доверия.

В таблице приведены результаты сравнения рассмотренных технологий.

PKI имеет широкое распространение, часто используется в корпоративной среде, но требует дополнительных затрат на управление ключами и сертификатами. OpenPGP и GnuPG предоставляют свободу и гибкость, не требуют централизованного управления, но могут быть сложными для неопытных пользователей. Блокчейн может обеспечить безопасность и де-

Сравнение технологий создания сетей доверия

Comparison of technologies for creating trust networks

Методы и средства	Шифрование данных	Аутентификация пользователей	Электронная подпись данных	Сложность использования частными лицами и малыми предприятиями	Необходимость поддерживать список доверенных удостоверяющих центров
PKI	Да	Да	Да	Да	Да
OpenPGP	Да	Да	Да	Да	Нет
GnuPG	Да	Да	Да	Нет	Нет
Blockchain	Нет	Да	Да	Да	Нет

централизацию, но требует больших затрат на вычислительные ресурсы и может быть неэффективным из-за большого объема вычислений при добавлении новых блоков в цепочку.

В предлагаемом решении будут объединены преимущества технологий PKI и GnuPG. Эти технологии обеспечивают безопасность, децентрализацию и гибкость, а также доступны широкому кругу пользователей.

Алгоритм работы файлообменника следующий.

1. Регистрация пользователя на сервере (с заданием имени пользователя, его пароля, генерацией пары ключей электронной подписи и выдачей сертификата открытого ключа).
2. Идентификация и аутентификация пользователя по уникальному имени (логину) и паролю, удовлетворяющему ограничениям на его минимальную длину и сложность.
3. Авторизация пользователя (получение доступа в файловые хранилища других пользователей, которые ранее были ему предоставлены).
4. Прием или отклонение запросов в друзья или приглашений в файловые хранилища, полученных от других пользователей.
5. Отправка запросов в друзья другим пользователям или удаление пользователей из списка своих друзей.
6. Создание файловых хранилищ с указанием минимального уровня доверия, при котором другой пользователь сможет получить к нему доступ (значения от 0 до 1). Для расчета уровня доверия для нового пользователя, получающего доступ в файловое хранилище, используется теорема Байеса: $P(H \mid E) == \frac{P(E \mid H)P(H)}{P(E)}$.

Значение априорной вероятности $P(H)$ принимается равным 0,9, если пользователь находится в друзьях у владельца хранилища, и 0,8 в противном случае. Значение условной вероятности $P(E|H)$ принимается равным уровню доверия владельца хранилища. Маргинальная вероятность $P(E)$ определяется на основе уровней доверия всех друзей пользователя, допущенных в его файловое хранилище: $P(E) = \sum P(E|H_i)P(H_i)$.

7. Загрузка и выгрузка файлов в хранилище. При выгрузке файла в хранилище пользователя файл шифруется и снабжается электронной подписью с помощью закрытого ключа владельца хранилища. При загрузке файла из хранилища другого пользователя проверяется подпись под файлом с помощью сертификата открытого ключа этого пользователя и файл расшифровывается.
8. Завершение сеанса работы пользователя.

Серверная часть файлообменника

Представим БД приложения, которая состоит из таблиц. В таблице users содержится информация о зарегистрированных пользователях, в storages – о файловых хранилищах, созданных пользователями, в friend_requests_users – о запросах пользователей в друзья, в friend_users – о пользователях, являющихся друзьями, в storage_invites – о приглашениях в файловые хранилища пользователей, в storage_users – о пользователях файловых хранилищ. Сертификаты открытых ключей всех пользователей хранятся в хранилище на сервере, выполняющем в приложении роль удостоверяющего центра, а закрытый ключ пользователя и сертификат его открытого ключа – на устройстве пользователя (закрытый ключ – в защищенной части его памяти). Для управления БД используется система PostgreSQL.

Взаимодействие серверной части файлообменника с БД реализовано с помощью фрейм-

ворка Exposed [7]. Разработаны модели и сущности для каждой таблицы из БД. Все возможные запросы к серверу разделены на три части: запросы для регистрации и авторизации пользователя, для получения информации о пользователях, для получения информации о файловых хранилищах.

Примеры запросов:

– certificate – получение сертификата сервера, необходимого для установления защищенного соединения по протоколу SSL/TLS;

– registration – регистрация нового пользователя с уникальным именем;

– users – получение списка всех зарегистрированных пользователей приложения;

– addFriendRequest – добавление запроса в друзья;

– storageInvites – получение приглашений конкретного пользователя в файловые хранилища;

– createStorage – создание файлового хранилища.

Запросы передаются в формате json.

Клиентская часть приложения

Проектирование клиентской части приложения проводилось на основе принципов разделения задач, управления пользовательским интерфейсом на основе моделей данных, единого достоверного источника информации (Single Source of Truth, SSoT) [8], однонаправленного потока данных (Unidirectional Data Flow, UDF) [9]. С учетом этих принципов в приложении были выделены три уровня: пользовательского интерфейса, отображающего данные приложения на экране, домена для упрощения и повторного использования взаимодействия между интерфейсом и данными и уровень данных, содержащий бизнес-логику приложения и предоставляющий его данные.

Для обеспечения кроссплатформенности разрабатываемого приложения использован инструмент Kotlin Multiplatform [10], который позволяет разработчикам создавать приложения для разных платформ (Android, iOS, Web и других) с применением общего кода на языке Kotlin. Для пробного тестирования разработано клиентское мобильное приложение для ОС Android с использованием фреймворка Jetpack Compose.

Пользовательский интерфейс клиентского приложения состоит из пяти частей: экраны авторизации и регистрации пользователя, домашний экран, экраны поиска пользователей, создания и выбора файлового хранилища, загрузки и выгрузки файлов.

При проведении тестирования разработанного файлообменника использовались как реальные мобильные устройства (Xiaomi Mi 6 с процессором Snapdragon 835 и 6 Гб оперативной памяти, работающий под управлением ОС Android 9), так и их эмуляторы (Google Pixel 6 Pro с процессором Google Tensor и 12 Гб оперативной памяти под управлением ОС Android 12, Google Pixel 7 Pro с процессором Google Tensor 2 и 12 Гб оперативной памяти под управлением ОС Android 13). Эмуляторы выполнялись на персональном компьютере с процессором AMD Ryzen 7 5800x3d и 64 Гб оперативной памяти, работающем под управлением ОС Windows 11 Pro.

Для тестирования были созданы 20 пользователей, для каждого из которых создавалось файловое хранилище. Случайным образом между пользователями распределялись приглашения к добавлению в друзья и к доступу в файловые хранилища. Для загрузки на сервер и выгрузки с него использовался файл (RAR-архив) размером 94,3 Мб. Время выгрузки для разных устройств и эмуляторов составляло от 12 до 20 секунд, а время загрузки – от 17 до 25 секунд. Увеличение времени загрузки файла связано с особенностью работы серверной части приложения: при выгрузке время считается до окончания выгрузки файла на сервер, после чего файл шифруется, но это время не включается в результат, так как операция производится вне соединения с клиентской частью. При загрузке файла клиент ожидает окончания расшифрования файла перед его передачей с сервера.

Результаты проведенного тестирования показали, что приложение работоспособно и позволяет пользователям создавать собственные сети доверия для безопасного и быстрого обмена файлами между собой. Разработанное приложение предназначено прежде всего для корпоративных информационных систем бюджетных организаций и предприятий малого и среднего бизнеса, а также для частных лиц.

Заключение

В статье представлены результаты разработки приложения для обмена файлами между пользователями в глобальной сети, обеспечивающего конфиденциальность, подлинность и целостность передаваемых файлов. Разработанное приложение объединяет преимущества

инфраструктуры открытых ключей (PKI) и сетей доверия типа GnuPG. При регистрации пользователя создаются пара его асимметричных ключей и сертификат открытого ключа (в формате стандарта X.509), а закрытый ключ сохраняется в защищенной от несанкционированного доступа части памяти его устройства. В создаваемых на сервере хранилищах файлы находятся в зашифрованном виде, а при передаче между сервером и клиентом используется протокол безопасной передачи данных HTTPS.

По сравнению с другими решениями для обмена файлами между пользователями Интернета разработанное приложение имеет ряд преимуществ.

- Использует более надежный способ подтверждения подлинности и целостности передаваемых файлов (каждый файл снабжается электронной подписью отправителя, которая проверяется получателем с помощью сертификата открытого ключа отправителя, выданного сервером сети доверия). Подбор закрытого ключа подписи в отличие от подбора пароля (при использовании подтверждения подлинности по имени и паролю) является практически не решаемой сегодня задачей (при правильном выборе длины ключа). Использование сервера сети доверия в качестве удостоверяющего центра для выдачи сертификатов открытых ключей пользователей файлообменника избавляет участников сети доверия от необходимости обращения в сторонние удостоверяющие центры и ведения на всех своих устройствах списков доверенных удостоверяющих центров.
- Обеспечивает конфиденциальность файлов с помощью их симметричного шифрования не только при передаче файлов по сети, но и при хранении на сервере, что минимизирует риски, связанные с похищением файлов пользователей с сервера.
- Использует сертификаты открытых ключей в формате общепринятого международного стандарта X.509, что позволяет применять для работы с ними имеющиеся в открытом доступе инструментальные программные средства.
- Позволяет управлять допуском к файловому хранилищу пользователя сети доверия других пользователей сети с помощью рассчитываемых по теореме Байеса значений уровней доверия.

Таким образом, предложенное решение обеспечивает большую безопасность и дополнительную функциональность по сравнению с передачей файлов с помощью других программных средств. Выбор инструментальных средств разработки позволил существенно уменьшить трудоемкость переноса разработанного клиентского приложения на другие операционные платформы.

Разработанное для обмена файлами приложение может применяться при организации и проведении учебного процесса в университете (например, при выдаче студентам индивидуальных заданий и приеме от них отчетов, содержащих и программный код). Также оно может использоваться как группами частных пользователей Интернета, так и пользователями корпоративных информационных систем предприятий малого и среднего бизнеса.

Направлениями развития разработанного файлообменника для устранения имеющихся в текущей версии недостатков могут быть замена аутентификации пользователей сети доверия по имени и паролю аутентификацией с помощью их закрытых ключей и одного из криптографических протоколов с нулевым разглашением; создание клиентских приложений для других операционных платформ; использование только криптоалгоритмов из российских стандартов; расчет уровня доверия к пользователю при предоставлении ему доступа в файловое хранилище с помощью разных методов для выбора наилучшего; реализация хранения закрытых ключей пользователей не в защищенной части памяти их мобильных устройств, а на внешних устройствах.

Список литературы

1. Дикий Д.И., Артемьева В.Д. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей Интернета // Науч.-технич. вестн. информационных технологий, механики и оптики. 2019. Т. 19. № 1. С. 109–117. doi: 10.17586/2226-1494-2019-19-1-109-117.

2. Hou R., Ren G., Zhou C., Yue H., Liu H., Liu J. Analysis and research on network security and privacy security in ubiquitous electricity Internet of things. Computer Communications, 2020, vol. 158, pp. 64–72. doi: 10.1016/j.comcom.2020.04.019.

3. Kent D., Cheng B.H.C., Siegel J. Assuring vehicle update integrity using asymmetric public key infrastructure (PKI) and public key cryptography (PKC). SAE Int. J. Transp. Cyber. & Privacy, 2019, vol. 2, no. 2, pp. 141–158. doi: 10.4271/11-02-02-0013.

4. Yeoh W.-Z., Teh J.S., Chen J. Automated enumeration of block cipher differentials: An optimized branch-and-bound GPU framework. JISA, 2022, vol. 65, art. 103087. doi: 10.1016/j.jisa.2021.103087.

5. Ray P.P., Chowhan B., Kumar N., Almogren A. BIoTHR: Electronic health record servicing scheme in IoT-blockchain ecosystem. IEEE Internet of Things J., 2021, vol. 8, no. 13, pp. 10857–10872. doi: 10.1109/JIOT.2021.3050703.

6. Mukherjee P., Pradhan C. Blockchain 1.0 to Blockchain 4.0 − The evolutionary transformation of blockchain technology. In: Blockchain Tech.: Applications and Challenges. ISRL, 2021, vol. 203, pp. 29–49. doi: 10.1007/978-3-030-69395-4_3.

7. Candas S., Muschner Ch., Buchholz S. et al. Code exposed: Review of five open-source frameworks for modeling renewable energy systems. Renewable and Sustainable Energy Reviews, 2022, vol. 161, art. 112272. doi: 10.1016/j.rser.2022.112272.

8. Mulyana E., Fakih G. Network automation with a single source of truth in a heterogeneous environment. Int. J. on Electrical Eng. and Informatics, 2022, vol. 14, no. 1, pp. 92–100. doi: 10.15676/ijeei.2022.14.1.6.

9. Воротницкий Ю.И., Румас Р.А. Архитектура аппаратно-программного средства однонаправленной передачи данных в компьютерных сетях // Докл. БГУИР. 2023. Т. 21. № 3. С. 96–101. doi: 10.35596/1729-7648-2023-21-3-96-101.

10. Кузнецова С.В. Особенности кроссплатформенной разработки мобильных приложений с использованием Xamarin // Тр. МАИ. 2022. № 125. С. 1–27. doi: 10.34759/trd-2022-125-21.

Software & Systems doi: 10.15827/0236-235X.142.230-237 2024, 37(2), pp. 230–237

Secure file sharing based on trust networks and public key certificates using a developed application

Pavel B. Khorev [1]✉, Dmitry A. Losev [1]

[1] National Research University "Moscow Power Engineering Institute", Moscow, 111250, Russian Federation

For citation

Khorev, P.B., Losev, D.A. (2024) 'Secure file sharing based on trust networks and public key certificates using a developed application', *Software & Systems*, 37(2), pp. 230–237 (in Russ.). doi: 10.15827/0236-235X.142.230-237

Article info

Received: 29.09.2023 After revision: 18.12.2023 Accepted: 25.12.2023

Abstract. The paper analyzes the shortcomings of existing software tools for file sharing (including cloud storage, messenger programs, e-mail). To eliminate the shortcomings, the authors of the paper have developed a project of a client-server application based on building a trust network using public key certificates of its participants. The project uses symmetric and asymmetric cryptography algorithms, methods for building trust networks and using public key certificates to ensure the authenticity, confidentiality and integrity of data. The implemented server part and the client mobile application allow users to exchange files with each other within one specific file storage. Uploaded files have electronic signatures of their creators and are stored on the server in encrypted form. When developing the application, the authors used the technology that facilitates its portability to other operating platforms. The client application is built and tested on devices running the Android operating system. The developed application has the functionality necessary for file hosting and ensures the authenticity, integrity and confidentiality of transferred files. This allows it to be used both by groups of private users and in corporate information systems.

Keywords: client-server application, public key certificates, trust network, secure files sharing

References

1. Dikii, D.I., Artemeva, V.D. (2019) 'MQTT data protocol in remote access control management model for internet networks', *Sci. and Tech. J. of Inform. Tech., Mech. and Optics*, 19(1), pp. 109–117 (in Russ.). doi: 10.17586/2226-1494-2019-19-1-109-117.

2. Hou, R., Ren, G., Zhou, C., Yue, H., Liu, H., Liu, J. (2020) 'Analysis and research on network security and privacy security in ubiquitous electricity Internet of things', *Computer Communications*, 158, pp. 64–72. doi: 10.1016/j.comcom.2020.04.019.

3. Kent, D., Cheng, B.H.C., Siegel, J. (2019) 'Assuring vehicle update integrity using asymmetric public key infrastructure (PKI) and public key cryptography (PKC)', *SAE Int. J. Transp. Cyber. & Privacy*, 2(2), pp. 141–158. doi: 10.4271/11-02-02-0013.

4. Yeoh, W.-Z., The, J.S., Chen, J. (2022) 'Automated enumeration of block cipher differentials: An optimized branch-and-bound GPU framework', *JISA*, 65, art. 103087. doi: 10.1016/j.jisa.2021.103087.

5. Ray, P.P., Chowhan, B., Kumar, N., Almogren, A. (2021) BIoTHR: Electronic health record servicing scheme in IoT-blockchain ecosystem. *IEEE Internet of Things J.*, 8(13), pp. 10857–10872. doi: 10.1109/JIOT.2021.3050703.

6. Mukherjee, P., Pradhan, C. (2021) 'Blockchain 1.0 to Blockchain 4.0 − The evolutionary transformation of blockchain technology', in *Blockchain Tech.: Applications and Challenges. ISRL*, 203, pp. 29–49. doi: 10.1007/978-3-030-69395-4_3.

7. Candas, S., Muschner, Ch., Buchholz, S. et al. (2022) 'Code exposed: Review of five open-source frameworks for modeling renewable energy systems', *Renewable and Sustainable Energy Reviews*, 161, art. 112272. doi: 10.1016/j.rser.2022.112272.

8. Mulyana, E., Fakih, G. (2022) 'Network automation with a single source of truth in a heterogeneous environment', *Int. J. on Electrical Eng. and Informatics*, 14(1), pp. 92–100. doi: 10.15676/ijeei.2022.14.1.6.

9. Varatnitski, Y.I., Rumas, R.A. (2023) 'Architecture of hardware and software for unidirectional data transmission in computer networks', *Doklady BGUIR*, 21(3), pp. 96–10 (in Russ.). doi: 10.35596/1729-7648-2023-21-3-96-101.

10. Kuznetsova, S.V. (2022) 'Features of cross-platform mobile applications development using Xamarin', *Proc. of MAI*, (125), pp. 1–27 (in Russ.). doi: 10.34759/trd-2022-125-21.

Авторы

Хорев Павел Борисович [1], к.т.н., доцент, профессор, pbkh@yandex.ru
Лосев Дмитрий Алексеевич [1], магистрант, lda-1028@mail.ru

[1] Национальный исследовательский университет «Московский энергетический институт», г. Москва, 111250, Россия

Authors

Pavel B. Khorev [1], Cand. of Sci. (Engineering), Associate Professor, Professor, pbkh@yandex.ru
Dmitry A. Losev [1], Graduate Student, lda-1028@mail.ru

[1] National Research University "Moscow Power Engineering Institute", Moscow, 111250, Russian Federation