КРИПТОГРАФИЧЕСКИЙ ФУНКЦИОНАЛ МЕССЕНДЖЕРА PEREGRINE НА БАЗЕ НАЦИОНАЛЬНЫХ СТАНДАРТОВ ШИФРОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Петров С. Н., Пакуль Е. С., Корчинский А. А.
Кафедра защиты информации,
Белорусский государственный университет информатики и радиоэлектороники
Национальный детский технопарк
Минск, Республика Беларусь
E-mail: petrov@bsuir.by

Разработан мессенджер Peregrine на языке программирования Rust, имеющий клиент-серверную архитектуру, в которой клиенты (устройства пользователей) взаимодействуют с центральным сервером, пересылающим сообщения от отправителя к получателю, выполняющим аутентификацию пользователей и управление доступом. Центральным звеном стала разработка библиотеки bee2-rs на языке программирования Rust. Библиотека создана для вызовов функций библиотеки Bee2 (от официального разработчика), написанной на языке программирования C. Приведено частичное описание криптографического функционала мессенджера.

Введение

Приложения для обмена мгновенными сообщениями (мессенджеры) стали важной частью современной жизни. Мессенджеры стали универсальным инструментом для коммуникации и совместной работы. Однако в связи с актуализацией проблематики обеспечения безопасности данных и возможными ограничениями (например, запретом на использование) становятся актуальными вопросы разработки альтернативных, безопасных мессенджеров.

Конфиденциальность информации является одной из уязвимых областей в системах мгновенных сообщений. Информация, передаваемая через системы обмена сообщениями, проходит через несколько точек раскрытия до достижения получателя. Сообщения проходят через серверы и сети, которые могут быть небезопасными, такк как нет гарантии того, что отправленная информация защищена. Ситуация усложняется тем, что службы обмена мгновенными сообщениями часто содержат программы обмена файлами, которые могут оставлять конфиденциальную информацию на сервере после завершения сеанса. Серверная инфраструктура систем обмена мгновенными сообщениями является критическим компонентом, который также подвержен различным типам атак, например, атак типа «отказ в обслуживании», которые могут нарушить работу серверов. Такие атаки часто осуществляются путем перегрузки серверов запросами, что приводит к исчерпанию ресурсов и неспособности обрабатывать законные запросы. Другой формой атаки на серверную инфраструктуру является несанкционированный доступ к серверам, где хранятся сообщения и метаданные пользователей, что может привести к утечке конфиденциальной информации. Особую опасность представляют атаки на централизованные серверы, где хранятся ключи шифрования или где происходит обработка сообщений перед их отправкой получателям.

I. Мессенджер Peregrine

Авторами разработан мессенджер с защитой пользовательских данных и end-to-end (сквозным) шифрованием на основе национальных криптографических стандартов. Мессенджер имеет клиент-серверную архитектуру, в которой клиенты (устройства пользователей) взаимодействуют с центральным сервером, пересылающим сообщения от отправителя к получателю, выполняющим аутентификацию пользователей и управление доступом. Определены функциональные возможности серверной и пользовательских частей. Мессенджер назван Peregrine.

Актуальность разработки нового мессенджера обусловлена следующими факторами:

- рост спроса на приватность и активный переход на альтернативные платформы после скандалов с утечками данных пользователей WhatsApp;
- доминирование Telegram и Viber создает зависимость от зарубежных сервисов, что повышает риски санкционных ограничений;
- государственные и коммерческие организации заинтересованы в решениях с серверами внутри страны.

Основные компоненты системы:

- Клиентское приложение (коммуникация с сервером происходит по API с использованием HTTP POST запросов).
- Сервер, выступающий в роли посредника между клиентами, управляет аутентификацией пользователей, хранением данных в базе данных. Сервер использует протокол без сохранения состояния (stateless protocol),то есть общение с сервером состоит из независимых пар запрос-ответ. Код сервера состоит из модуля базы данных, модуля хранили-

ща данных, где хранятся файлы, отправленные пользователями, а также модуля API.
- База данных MySQL (хранение пользовательских данных, токенов сессии и т.п.).
- Балансировщик нагрузки.

II. Криптографический функционал мессенджера

Основной акцент в проекте сделан на безопасность. Центральным звеном стала разработка библиотеки bee2-rs на языке программирования Rust. Библиотека создана для вызовов функций библиотеки Bee2. Официальный разработчик Bee2– НИИ прикладных проблем математики и информатики БГУ [1]. В библиотеке реализованы алгоритмы и протоколы, определенные в стандартах СТБ 34.101.31, (belt): шифрование, имитозащита, хэширование, управление ключами; СТБ 34.101.45 (bign): ЭЦП и транспорт ключа (шифрование с открытым ключом); СТБ 34.101.47 (brng): имитозащита, ГПСЧ, одноразовые пароли (HOTP, TOTP, OCRA); СТБ 34.101.60 (bels): разделение секрета; СТБ 34.101.66 (bake): протоколы формирования общего ключа (BMQV, BSTS, BPACE);СТБ 34.101.77 (bash): хэширование). Однако библиотека Bee2 написана на языке C, а мессенджер на языке Rust и напрямую использовать Bee2 в проекте невозможно. Для решения данной проблемы использован механизм binding для языка Rust (Rust Foreign Function Interface, FFI).

Были сделаны абстракции нулевой стоимости для повышения удобства и уменьшения вероятности неправильного использования библиотеки (например, в некоторые функции нельзя передать данные неправильного размера, иначе программа не скомпилируется). Также для блочного шифра Belt было реализовано дополнение байтов в конец изначальных данных, чтобы длина результирующих зашифрованных данных была кратна n (по умолчанию – 16, $1 < n < 256$) для невозможности определить оригинальную длину сообщения по длине зашифрованных данных.

Для проверки правильности функций и соответствия их функциям библиотеки Bee2 были написаны unit-тесты. Для Bash было написано 3 теста по 52 проверок в каждом, для Belt – 7 тестов с общим числом проверок 76, для Bign – 4 теста счислом проверок 311, для Brng написано 3 теста с 23 проверками.

Реализован расширенный тройной протокол Диффи-Хеллмана (X3DH) для обмена ключами симметричного алгоритма шифрования. X3DH устанавливает общий секретный ключ между двумя сторонами, которые взаимно аутентифицируют друг друга на основе открытых ключей. Реализован режим belt-dwp алгоритма шифрования Belt (аутентификационное шифрование с ассоциированными данными). Для предотвращения кибератак типа bruteforce реализован следующий алгоритм аутентификации – клиент генерирует пару ключей для асимметричного алгоритма шифрования по паролю с использованием PBKDF2, после чего подписывает приватным ключом текущую временную метку (timestamp), а также период валидности сигнатуры до и после этого timestamp для принятия запроса даже в случае рассинхронизации времени между сервером и клиентом функции для протокола X3DH.

В структуре мессенджера был реализован функционал для использования криптографических стандартов отличных от национальных. На данный момент поддерживается только AES-GCM как альтернатива Belt. Открытый исходный код мессенджера позволяет добавить альтернативную библиотеку. При необходимости можно использовать разные криптографические библиотеки для разных чатов. В будущем планируется реализация поддержки следующих криптографических библиотек: chacha20poly1305, x25519-dalek, ed25519-dalek, k256, pbkdf2, sha3. У большинства из указанных ранее библиотек был проведен аудит безопасности [2], который либо не выявил серьезных уязвимостей, либо они были исправлены.

Исходный код разработанной библиотеки bee2-rs доступен на GitHub [3], а также опубликован в реестр библиотек (crates) сообщества Rust. Исходный код мессенджера был опубликован на платформае GitHub [4].

III. Заключение

Мессенджер Peregrine, использующий end-to-end шифрование на основе национальных стандартов, призван решить проблемы, обусловленные растущим спросом на приватность, стремлением снизить зависимость от зарубежных сервисов и необходимостью создания отечественных решений для государственных и коммерческих организаций. Проект успешно решает сложную задачу интеграции национальных криптографических стандартов в современную, безопасную среду Rust, делая основной акцент на предотвращении ошибок разработчика (через абстракции) и использовании надежных протоколов (X3DH, AEAD).

1. Криптографические стандарты Республики Беларусь [Электронный ресурс] //НИИ прикладных проблем математики и информатики БГУ – Режим доступа: `https://apmi.bsu.by/resources/std.html`. – Дата доступа: 01.10.2025.
2. bee2-rs [Электронный ресурс] / Е.С. Пакуль – Режим доступа: `https://github.com/tpyauheni/bee2-rs`. – Дата доступа: 01.10.2025.
3. Peregrine [Электронный ресурс] / Е.С. Пакуль – Режим доступа: `https://github.com/tpyauheni/peregrine`. – Дата доступа: 01.10.2025.