Метод повышения безопасности передачи изображений в мессенджерах с использованием одноразовых паролей

И.В. Саварин

Финансовый университет при Правительстве Российской Федерации, Москва

Аннотация: В статье представлен метод защиты передаваемых изображений в мессенджерах с использованием одноразовых паролей на основе времени (Time-based One-Time Password – TOTP). Предлагается дополнительный уровень защиты, основанный на комбинации маскирования изображений с помощью ортогональных матриц и двухфакторной аутентификации на основе TOTP. Разработан и протестирован прототип приложения на языке Python с использованием протокола удалённого вызова процедур gRPC для обеспечения безопасного обмена данными между клиентом и сервером. Представлены результаты реализации предложенного метода в предотвращении несанкционированного доступа к конфиденциальным изображениям.
Ключевые слова: Информационная безопасность, мессенджер, обмен сообщениями, коммуникации, системы мгновенного обмена сообщениями, одноразовый пароль.

1. Введение

Современные мессенджеры стали неотъемлемыми инструментами для эффективного взаимодействия пользователей в посведневной жизни, а также внутри компаний и организаций. Они предоставляют удобный функционал для обмена текстовыми сообщениями, документами, а также мультимедийными файлами, такими, как изображения и видеозаписи. Вместе с тем, растет осознание важности защиты передаваемой информации, особенно в условиях возрастающей активности киберпреступников и потенциальных рисков утечки данных.

В статье [1] было уделено значительное внимание вопросам безопасности передаваемых изображений. В частности, был предложен метод маскирования изображений с использованием ортогональных матриц [2], который продемонстрировал свою эффективность в сокрытии визуальной информации и предотвращении несанкционированного доступа. Однако остаются нерешёнными задачи, связанные с обеспечением дополнительной защиты данных в случаях физического доступа к устройствам или компрометации учетных записей пользователей.

Настоящее исследование направлено на разработку и внедрение дополнительного уровня защиты передаваемых конфиденциальных изображений в мессенджерах, основанный на использовании технологии одноразовых паролей основанных на времени (Time-Based One-Time Password – TOTP). Эта технология уже зарекомендовала себя в системах двухфакторной аутентификации, обеспечивая повышенную безопасность за счет временного характера генерируемых кодов. Внедрение TOTP в процесс открытия защищенных изображений позволит значительно снизить риски несанкционированного доступа даже в случае компрометации учетной записи пользователя.

Таким образом, цель настоящей работы заключается в расширении существующих методов защиты передаваемых изображений путем добавления механизма временной аутентификации на базе TOTP.

Статья состоит из 4 разделов. В разделе 2 проанализированы дополнительные меры защиты от уязвимости, связанной с хранением файлов изображений в кэше устройств в незашифрованном виде, описанной в статье [1]. В разделе 3 рассмотрены теоретические аспекты метода аутентификации с помощью одноразовых паролей TOTP. Также рассмотрен алгоритм его использования в приложении мессенджера. В разделе 4 представлен практический пример реализации предложенного дополнительного метода защиты данных посредством написания кода модуля проверки одноразовых паролей.

2. Анализ уязвимости

Современные мессенджеры представляют собой сложные программные комплексы, обеспечивающие обмен различными видами данных, включая текстовые сообщения, документы и мультимедийный контент. Несмотря на наличие множества мер безопасности, таких, как шифрование и

аутентификация, существуют определенные уязвимости, которые могут поставить под угрозу конфиденциальность передаваемой информации.

Одной из основных проблем является сохранение изображений в кэше устройства в незашифрованном виде. Хотя многие современные мессенджеры используют шифрование для защиты данных в процессе передачи, они зачастую не применяют аналогичные меры для защиты кэша на устройстве пользователя. Это означает, что злоумышленник, получивший физический доступ к устройству, может извлечь изображения из кэша и просмотреть их без каких-либо препятствий.

Кроме того, существует риск компрометации учетной записи пользователя. Если злоумышленнику удастся получить доступ к учетной записи пользователя или сотрудника компании, он сможет беспрепятственно открывать и просматривать любые ранее полученные изображения, независимо от степени их конфиденциальности.

Эти уязвимости создают значительные риски для безопасности информации. Потенциальные последствия включают:

1. Утрату конфиденциальных данных: Злоумышленник может получить доступ к документам, фотографиям, схемам и другим важным материалам, что приведет к нарушению коммерческой тайны или утечке персональных данных.

2. Злоупотребление полученной информацией: Злоумышленник может использовать украденные данные для шантажа, мошенничества или других противоправных действий.

3. Репутационные потери: утечка конфиденциальной информации может нанести серьезный ущерб репутации компании, что негативно скажется на доверии клиентов и партнеров.

Для устранения этих уязвимостей требуется дополнительный уровень защиты, который предотвратит несанкционированный доступ к

изображениям даже в случае физического доступа к устройству или компрометации учетной записи. В этом контексте внедрение механизма временной аутентификации на основе TOTP представляется эффективным решением.

Использование одноразовых паролей TOTP требует от пользователя ввода специального кода, сгенерированного на его устройстве, для каждого случая открытия защищенного изображения. Это добавляет еще один барьер для злоумышленников, поскольку даже при наличии физического доступа к устройству или учетной записи им потребуется действующий одноразовый пароль, чтобы открыть изображение. Код TOTP изменяется каждые N-секунд, что делает его практически невозможно предсказать или подделать.

Таким образом, введение TOTP-аутентификации в процесс открытия изображений в мессенджере значительно снижает риски, связанные с несанкционированным доступом к конфиденциальной информации, и повышает общий уровень безопасности системы.

3. TOTP как дополнительный способ защиты сообщений в мессенджере

Введение дополнительных уровней защиты в мессенджеры является необходимым условием для предотвращения утечек конфиденциальной информации и обеспечения высокого уровня безопасности данных. Одним из эффективных способов достижения этой цели является интеграция механизма аутентификации на основе одноразовых паролей. Рассмотрим подробнее, каким образом эта технология может быть использована для защиты передаваемых файловых сообщений в мессенджере.

TOTP — это метод двухфакторной аутентификации, который генерирует одноразовые пароли на основе текущего времени [3]. Основной целью TOTP является повышение уровня безопасности учетных записей и систем путем введения дополнительного фактора аутентификации, помимо

традиционного пароля. В отличие от статичных паролей, которые могут быть перехвачены или взломаны, одноразовые пароли TOTP действуют только в течение определенного периода времени, что значительно снижает риск несанкционированного доступа.

TOTP базируется на стандарте инженерного совета Интернета (Internet Engineering Task Force – IETF) RFC 6238, который определяет процесс генерации и верификации одноразовых паролей на основе времени. Основой этого метода является использование кода проверки подлинности сообщений, основанного на хэш-функции (Hash-based Message Authentication Code – HMAC) [4] совместно с текущим временем.

Процесс генерации одноразового пароля TOTP включает следующие этапы:

1. Инициализация: пользователь регистрирует свое устройство (например, смартфон) в системе, используя специальный QR-код или секретный ключ. Этот ключ известен только пользователю и серверу.

2. Генерация пароля: на основе текущего времени и секретного ключа устройство генерирует одноразовый пароль. Время делится на фиксированные интервалы (обычно 30 секунд), и для каждого интервала создается уникальный код.

3. Проверка: когда пользователь пытается войти в систему или выполнить другую защищённую операцию, он вводит сгенерированный код. Сервер проверяет введённый код, сравнивая его с тем, который сам рассчитывает на основе текущего времени и известного секретного ключа.

4. Действие: если пароли совпадают, операция считается успешной, и пользователь получает доступ. Если пароль неверен или истек срок его действия, система отклоняет запрос.

TOTP обладает рядом важных преимуществ:

1. Устойчивость к перехвату: даже если злоумышленник перехватил одноразовый пароль, он не сможет воспользоваться им, так как тот станет недействительным через короткий промежуток времени [5].

2. Независимость от сети: генерация паролей происходит на устройстве пользователя, поэтому нет необходимости в постоянном подключении к интернету.

3. Простота использования: пользователям достаточно иметь мобильное приложение, которое автоматически генерирует коды.

Для успешного внедрения механизма TOTP в мессенджер необходимо выполнить ряд шагов, направленных на настройку и синхронизацию системы. Эти шаги включают регистрацию устройства пользователя, создание и хранение секретного ключа, а также синхронизацию таймеров между устройством и сервером. Рассмотрим подробнее данные шаги:

1. Установка и настройка приложения: при первом запуске мессенджера пользователь проходит процедуру регистрации устройства. Это включает создание уникальной пары ключей (открытый и закрытый), которые будут использоваться для генерации и проверки одноразовых паролей. Открытый ключ передается на сервер, а закрытый хранится на устройстве пользователя в зашифрованном виде.

2. Синхронизация таймеров: важным элементом работы TOTP является синхронизация таймеров между устройством пользователя и сервером. Оптимальным решением будет использование временной метки Unix Time Stamp, которая не зависит от часовых поясов [6]. Такая синхронизация гарантирует, что временные окна для генерации паролей будут одинаковыми на обеих сторонах.

3. Создание резервных копий: для предотвращения ситуации, когда пользователь теряет доступ к своему устройству или случайно удаляет приложение, рекомендуется создать резервную копию секретного ключа. Эта

копия должна храниться в надежном месте, доступ к которому ограничен только самим пользователем. В противном случае, пользователю будет предложено повторно пройти процедуру авторизации в приложении.

Используя рассмотренный вариант защиты передаваемых изображений и файлов [1], добавим дополнительный механизм защиты от несанкционированного доступа к данным. Так как доступ к просмотру отправленного изображения возможен только после процедуры демаскирования, внедрим перед ней новый шаг – запрос одноразового пароля. Алгоритм работы будет следующим:

1. Запрос на открытие: когда получатель решает открыть изображение, он инициирует процедуру демаскирования. Мессенджер запрашивает у пользователя одноразовый пароль TOTP.

2. Генерация одноразового пароля: одноразовый пароль генерируется на устройстве пользователя с использованием секрета и текущего времени. Важно отметить, что каждый новый пароль уникален и действителен только в течение короткого промежутка времени (по умолчанию установлен интервал 30 секунд).

3. Проверка кода: введённый пользователем одноразовый пароль передается на сервер для проверки с помощью соответствующего gRPC запроса. Сервер сравнивает полученный код с ожидаемым значением, рассчитанным на основе текущего времени и секретного ключа.

4. Разрешение на демаскирование: если код верен, сервер разрешает демаскирование изображения. Клиентское приложение выполняет необходимые вычислительные операции и отображает оригинальное изображение.

5. Ошибка доступа: если код неверен или истёк срок его действия, сервер возвращает сообщение об ошибке, и процедура демаскирования не

производится. Пользователь должен повторно ввести правильный код для продолжения.

На рис. 1 изображен алгоритм работы модуля одноразовых паролей в мессенджере.

Также, для дальнейшего укрепления защиты передаваемых данных и предотвращения несанкционированного доступа к информации, необходимо рассмотреть дополнительные меры безопасности, дополняющие основную схему использования TOTP. Эти меры направлены на снижение риска компрометации системы, ограничение числа попыток ввода неверных паролей, ведение журнала событий и регулярное обновление секретных ключей. Их правильная реализация обеспечит более высокий уровень безопасности и уверенность в сохранности конфиденциальных данных. Возможные варианты дополнительных мер защиты:

1. Ограничение количества попыток: чтобы предотвратить атаки перебора (brute force атаки), направленные на перебор всех возможных комбинаций [7], можно ограничить количество неудачных попыток ввода одноразового пароля. Например, после трёх неправильных попыток доступ к процедуре демаскирования может быть заблокирован на определённое время.

2. Журналирование событий: все попытки открытия изображений должны регистрироваться в журнале событий. Это поможет администраторам системы отслеживать подозрительную активность и принимать соответствующие меры.

3. Обновление секретного ключа: периодически (например, раз в месяц) следует обновлять секретный ключ, используемый для генерации одноразовых паролей. Это повысит общую безопасность системы и снизит риск компрометации ключа.

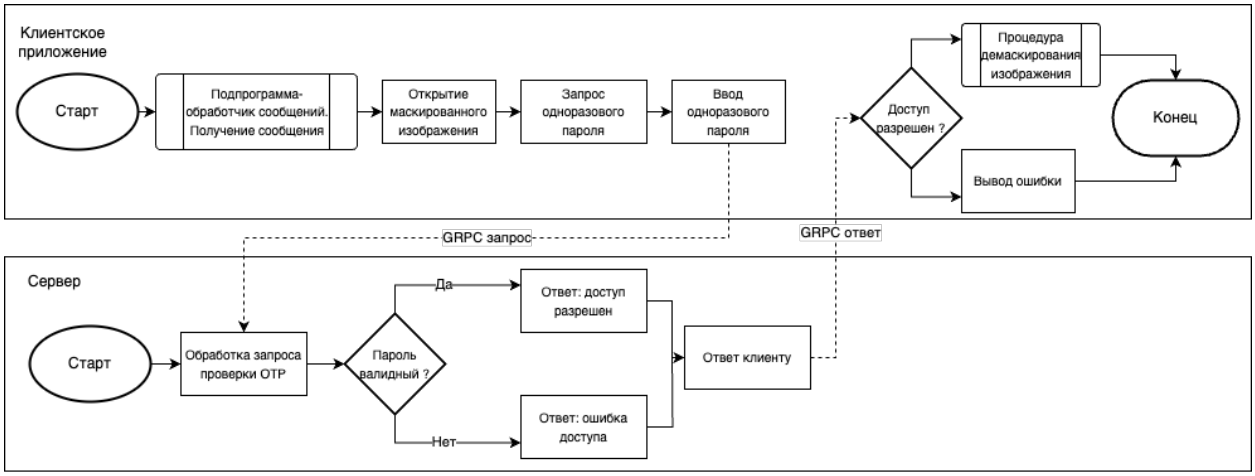

Рис. 1 – Алгоритм работы модуля одноразовых паролей в мессенджере

Рассмотрим пример реализации вышеописанного механизма дополнительной защиты.

4. Пример реализации

Для демонстрации предложенного дополнительного метода защиты передаваемых изображений в мессенджере, рассмотрим пример его практической реализации. В статье [1] был создан прототип приложения на языке Python, в котором для обеспечения безопасной передачи данных между клиентом и сервером использовался транспортный протокол удалённого вызова процедур (gRPC Remote Procedure Calls – gRPC) [8]. Прототип включает в себя как клиентскую, так и серверную части, взаимодействующие друг с другом через gRPC-соединение. В клиентской части реализованы механизмы загрузки и отправки изображений, а также их маскирования с использованием ортогональных матриц. Серверная часть отвечает за обработку запросов, проверку подлинности пользователей и управление сессиями.

Для реализации функции проверки валидности одноразового пароля (TOTP) в приложении, будут использованы две библиотеки:

pyotp — это библиотека, предназначенная для работы с одноразовыми паролями (OTP). Она поддерживает различные стандарты OTP, включая

TOTP, и предоставляет простые средства для генерации и проверки одноразовых паролей [9]. Библиотека pyotp облегчает интеграцию механизма TOTP в приложение мессенджера.

base64 — стандартная встроенная библиотека Python, предназначенная для кодирования и декодирования данных в формате Base64 [10]. В контексте приложения, она будет использован для кодирования и декодирования секретного ключа, необходимого для генерации и проверки одноразовых паролей.

Комбинация этих библиотек позволяет реализовать надежный и удобный метод проверки одноразовых паролей.

```
    def CheckOtp(self, request, context: grpc.ServicerContext):
    # Получение идентификатора пользователя из метаданных
    header = dict(context.invocation_metadata()).get('user-id')
    # Получение информации о пользователе из базы данных
    user = self.db.get_user_by_uid(header)
    b_hashed_secret = b32encode(user.secret.encode())
    otp = pyotp.TOTP(b_hashed_secret, digest=hashlib.sha512)
    otp_now = otp.now() # Получение текущего значения TOTP
    if otp_now != request.otp:
        return auth.ResponseCheckOtp(valid=False)
    return auth.ResponseCheckOtp(valid=True)
```

Рассмотрим пример открытия изображения с использованием механизма аутентификации на основе TOTP. Допустим, пользователь получил конфиденциальное изображение через мессенджер и теперь хочет его открыть. Программа запрашивает у пользователя одноразовый пароль, который был сгенерирован на его устройстве. После ввода пароля программа проверяет его валидность, и если пароль верный, изображение открывается и

демонстрируется пользователю. На рис. 2 изображено поведение программы в момент запроса TOTP-кода.

При введении валидного, на текущий момент времени, одноразового пароля, приложение производит процедуру демаскирования и отображает переданное изображение.

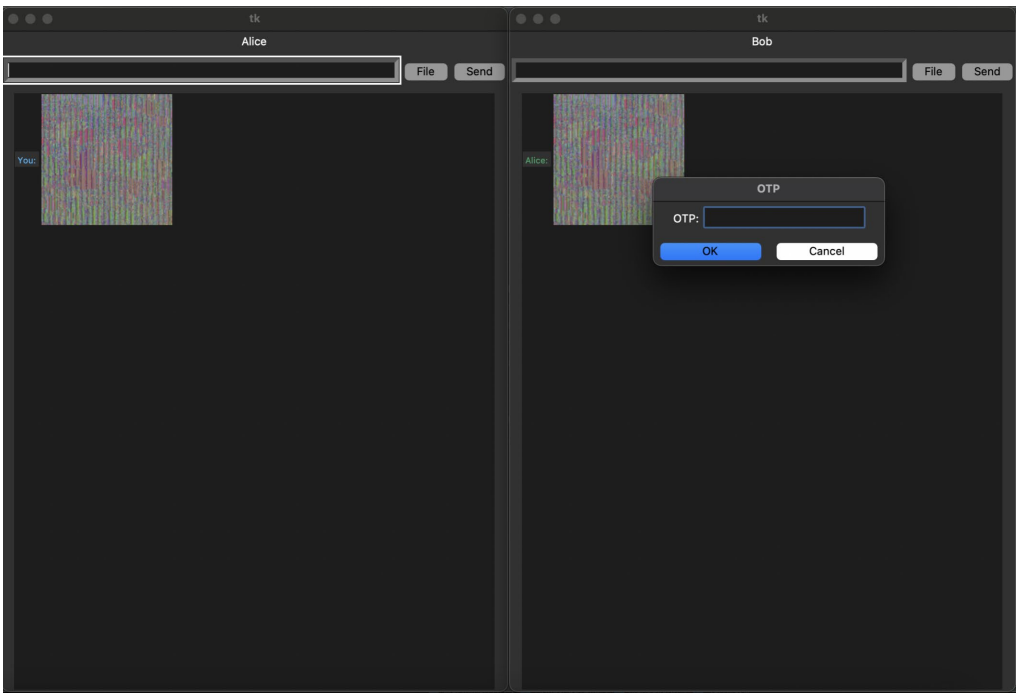

Рис. 2. – Поведение программы в момент открытия изображения.

В противном случае, приложение возвращает ошибку и процедура не выполняется пока не будет введён валидный код. На рис. 3 изображено поведение приложения при вводе невалидного одноразового кода.

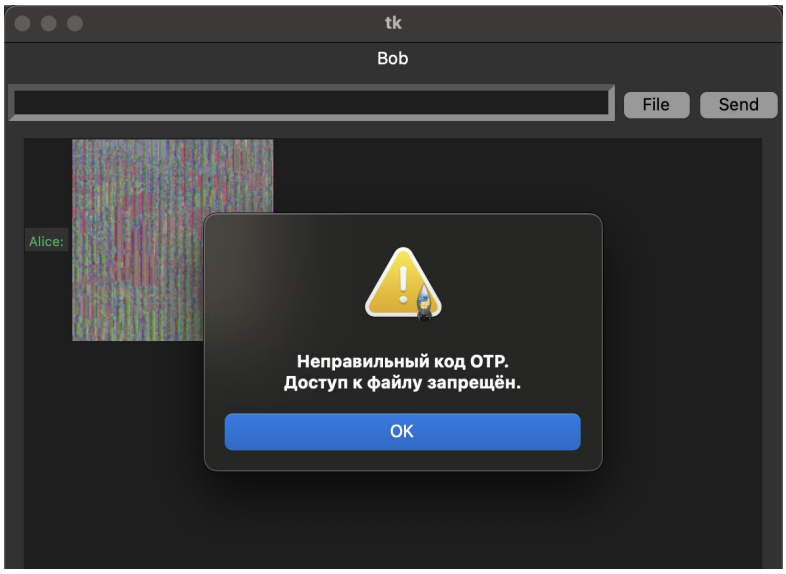

Рис. 3. – Поведение программы при вводе невалидного кода.

Выводы

В статье описан метод защиты передаваемых изображений в мессенджерах, основанный на сочетании маскирования данных с использованием ортогональных матриц и двухфакторной аутентификации на основе одноразовых паролей на основе времени (TOTP).

В ходе работы был разработан и интегрирован в прототип мессенджера [1] модуль аутентификации на основе одноразовых паролей (TOTP) для открытия конфиденциальных изображений. Данный модуль усиливает защиту передаваемых данных, обеспечивая дополнительный уровень безопасности, который требует от пользователя ввода одноразового пароля для доступа к конфиденциальному изображению или файлу.

Литература

1. Саварин И.В. Метод защиты изображений, передаваемых через мессенджер // Инженерный вестник Дона, 2024, №. 12. URL: ivdon.ru/ru/magazine/archive/n12y2024/9726

2. Сергеев М.Б., Татарникова Т.М., Сергеев А.М., Боженко В.В. Метод обеспечения конфиденциальности данных с применением ортогональных

матриц // Инженерный вестник Дона, 2024, № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8967

3. M'Raihi D. et al. Totp: Time-based one-time password algorithm. – 2011. – №. rfc6238. – P. 4

4. Bellare M., Canetti R., Krawczyk H. Message authentication using hash functions: The HMAC construction //RSA Laboratories' CryptoBytes. – 1996. – V. 2. – №. 1. – P. 12-15.

5. Cunha V. A. et al. TOTP Moving Target Defense for sensitive network services //Pervasive and Mobile Computing. – 2021. – V. 74. – P. 101412.

6. TOTP (Time-based one-time Password algorithm) // Хабр. URL: habr.com/ru/articles/534064/ (дата обращения: 02.01.2025).

7. Dave K. T. Brute-force attack 'seeking but distressing' // Int. J. Innov. Eng. Technol. Brute-force. – 2013. – V. 2. – №. 3. – P. 76.

8. What is grpc // gRPC URL: grpc.io/docs/what-is-grpc/introduction/ (дата обращения: 02.01.2025).

9. PyOTP - The Python One-Time Password Library // PyOTP documentation URL: pyauth.github.io/pyotp/ (дата обращения: 02.01.2025).

10. base64 — Base16, Base32, Base64, Base85 Data Encodings // Python 3.13.1 documentation. URL: docs.python.org/3/library/base64.html (дата обращения: 02.01.2025).

References

1. Savarin I.V. Inzhenernyi vestnik Dona, 2024, №. 12. URL: ivdon.ru/ru/magazine/archive/n12y2024/9726

2. Sergeev M.B., Tatarnikova T.M., Sergeev A.M., Bozhenko V.V. Inzhenernyi vestnik Dona, 2024, № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8967

3. M'Raihi D. et al. Totp: Time-based one-time password algorithm. 2011. №. rfc6238. P. 4.

4. Bellare M., Canetti R., Krawczyk H. RSA Laboratories' CryptoBytes. 1996. V. 2. №. 1. pp. 12-15.

5. Cunha V. A. Pervasive and Mobile Computing. 2021. V. 74. P. 101412.

6. TOTP (Time-based one-time Password algorithm). URL: habr.com/ru/articles/534064/ (date accessed: 02.01.2025).

7. Dave K. T. Int. J. Innov. Eng. Technol. Brute-force. 2013. V. 2. №. 3. P. 76.

8. What is grpc URL: grpc.io/docs/what-is-grpc/introduction/ (date accessed 02.01.2025).

9. PyOTP The Python One-Time Password Library. URL: pyauth.github.io/pyotp/ (date accessed: 02.01.2025).

10. base64 Base16, Base32, Base64, Base85 Data Encodings URL: docs.python.org/3/library/base64.html (date accessed 02.01.2025).

Дата поступления: 4.12.2024

Дата публикации: 12.01.2025