Метод защиты изображений, передаваемых через мессенджер

И.В. Саварин

Финансовый университет при Правительстве Российской Федерации, Москва

Аннотация: В настоящей статье рассматривается уязвимость, связанная с сохранением файлов изображений в кэше на жёстком диске устройства в незашифрованном виде. Исследуется характер данной проблемы и возможные последствия её эксплуатации, включая утечку конфиденциальных данных, злоупотребление полученной информацией и риски для корпоративных информационных систем. Основное внимание уделяется способу защиты от данной уязвимости, который основан на применении техники маскирования с использованием ортогональных матриц. Представлен разработанный прототип мессенджера, в котором реализован данный метод: изображения передаются и хранятся в файловой системе в маскированном виде, процесс демаскирования осуществляется непосредственно в самом приложении мессенджера.
Ключевые слова: информационная безопасность, мессенджер, обмен сообщениями, коммуникации, системы мгновенного обмена сообщениями, шифрование, ортогональные матрицы.

1. Введение

Современные мессенджеры играют ключевую роль в обмене информацией между пользователями, включая передачу различных типов данных, таких, как текстовые сообщения, голосовые записи и графические файлы. С ростом популярности мессенджеров и увеличением объёма передаваемого контента вопросы обеспечения конфиденциальности и безопасности данных становятся все более актуальными. Одной из серьёзных угроз является уязвимость систем хранения данных на стороне клиента, особенно когда речь идёт о передаче конфиденциальных изображений.

Передача мультимедийной информации посредством современных мессенджеров представляет собой одну из наиболее востребованных функций информационных технологий. Однако безопасность данных, передаваемых через такие платформы, остаётся актуальной проблемой. Одним из аспектов угрозы является открытый доступ к кэшированным изображениям на устройствах пользователей. Поскольку большинство мессенджеров хранят загруженные изображения в незашифрованном виде,

существует вероятность их несанкционированного извлечения третьими лицами.

Настоящая работа посвящена разработке метода защиты изображений, передаваемых с использованием мессенджеров, который основан на применении техники маскирования с использованием ортогональных матриц. Этот метод предполагает преобразование исходного изображения таким образом, что его визуальное содержание становится неразличимым до момента демаскирования, которое осуществляется непосредственно в самом мессенджере. Таким образом, даже при физическом доступе злоумышленника к устройству или при перехвате трафика изображение будет сохранено в форме, существенно затрудняющей его распознавание без знания ключа, хранимого в пределах защищённой среды приложения.

Целью исследования является разработка метода, направленного на обеспечение безопасности передаваемых изображений в условиях потенциального доступа третьих лиц к кэшируемым данным. В работе рассмотрены теоретические аспекты предлагаемого подхода, представлены результаты экспериментов, подтверждающие его работоспособность, а также обсуждён потенциал дальнейших усовершенствований и возможностей интеграции в существующие системы обмена сообщениями.

Статья состоит из 4 разделов. В разделе 2 проанализирована уязвимость, связанная с хранением файлов изображений в кэше устройств в незашифрованном виде. Рассмотрена угроза информационной безопасности, которая возникает при сохранении конфиденциальных данных в открытом доступе в кэше приложений. Описаны возможные последствия эксплуатации данной уязвимости. В разделе 3 рассмотрены теоретические аспекты и преимущества метода маскирования изображений с использованием ортогональных матриц. В разделе 4 представлен пример реализации предложенного метода защиты данных посредством создания

специализированного программного обеспечения – прототипа мессенджера, обеспечивающего безопасное хранение и передачу изображений с применением технологии маскирования с помощью ортогональных матриц.

2. Анализ уязвимости

Настоящая работа посвящена исследованию уязвимости, возникающей вследствие сохранения файлов изображений в кэше на жёстком диске устройства в незашифрованном виде. Данная уязвимость представляет собой потенциальную угрозу информационной безопасности пользователей, поскольку предоставляет возможность несанкционированного доступа к конфиденциальным данным, временно размещённым в кэше приложения.

При эксплуатации приложений пользователи нередко загружают и просматривают изображения. Эти данные обычно кэшируются приложением для оптимизации скорости повторного доступа к ним [1]. Однако, если файлы изображений сохраняются в кэше в оригинальном виде, существует риск того, что злоумышленник, обладающий физическим доступом к устройству, сможет извлекать эти файлы без применения специализированных средств или методов.

Подобная уязвимость была отмечена, например, в мессенджере Telegram, где сообщения с вложенными изображениями также сохранялись в открытом виде в локальной файловой системе устройства [2]. Аналогичные проблемы могут возникать и в других приложениях, использующих сходные методы обработки файлов.

Эксплуатация рассматриваемой уязвимости может приводить к серьёзным последствиям для пользователей и организаций:

1. Утрата конфиденциальных данных. Если пользователи пересылают или получают изображения, содержащие персональные или коммерчески важные сведения (например, документы, фотографии, сканированные копии удостоверений личности и т.п.), то такая информация может попасть в руки

злоумышленников. Это создаёт значительный риск нарушения приватности пользователей или коммерческой тайны предприятий.

2. Злоупотребление полученной информацией. Обладая доступом к таким данным, злоумышленники могут использовать их для вымогательства, мошенничества или иных противоправных действий.

3. Риски для корпоративных информационных систем. При использовании подобных приложений в рамках бизнеса утечка информации способна повлечь за собой существенные убытки для организации, включая утрату доверия клиентов, правовые санкции и подрыв деловой репутации.

На основании проведённого анализа становится очевидным, что рассмотренная уязвимость представляет собой угрозу для безопасности пользователей и организаций. Далее предлагается возможный вариант решения данной проблемы.

3. Способ обеспечения безопасности

Анализ научных исследований показывает, что проблема защиты данных и конфиденциальной информации остаётся актуальной задачей, особенно в условиях быстрого развития цифровых технологий. В последние годы особое внимание уделяется разработке методов обработки изображений, обеспечивающих высокий уровень безопасности. Одним из возможных направлений является использование ортогональных матриц для защиты изображений.

Так, в работе [3] представлен алгоритм цифровой обработки изображений с использованием унимодулярных матриц и логистической карты, реализованный на языке программирования Python. Авторы предлагают способ преобразования изображений, который позволяет повысить конфиденциальность передаваемых данных. Другой важный вклад в исследование данной области сделан в статье [4], где рассмотрено

применение ортогонального преобразования для защиты изображений от несанкционированного доступа. Авторами предложено использовать маскирование с помощью ортогональных матриц для защиты аудиофайлов и изображений. В качестве преимуществ данного метода авторы отмечают следующие:

1. Скорость работы за счёт симметричной криптосистемы.

2. Абсолютная стойкость в случае совпадения размера изображения или аудиофайла с порядком матрицы.

3. Возможность передачи только параметров матрицы-ключа, а не её значений.

Ортогональные матрицы обладают особыми математическими свойствами, которые позволяют эффективно скрывать исходные данные. Главное преимущество данного метода заключается в том, что он не только предотвращает прямой просмотр защищённых данных, но и значительно усложняет их восстановление даже при наличии физической копии файла.

В контексте приложения мессенджера можно выделить следующие преимущества данного метода защиты конфиденциальности пересылаемых изображений:

1. Ортогональное преобразование позволяет эффективно скрывать информацию, содержащуюся в изображениях, делая их недоступными для несанкционированного просмотра. Это особенно важно в корпоративных коммуникациях, где защита конфиденциальных данных имеет первостепенное значение.

2. Операции с ортогональными матрицами могут выполняться быстро, что минимизирует задержки при передаче и обработке сообщений. Это критично для современных мессенджеров, где пользователи ожидают мгновенной доставки информации.

3. При правильном использовании ортогонального преобразования качество изображения сохраняется даже после его маскирования. Получатель сможет восстановить оригинальную картинку без потерь в качестве.

4. Метод маскирования с использованием ортогональных матриц применим ко всем основным формам мультимедийных данных, включая фотографии, видео и аудио файлы. Это делает его универсальным инструментом для защиты различных типов медиаконтента он несанкционированного доступа.

Несмотря на сложность математического аппарата, лежащего в основе этого метода, его реализация относительно проста и может быть интегрирована в существующие системы обмена сообщениями без значительных изменений архитектуры приложения.

4. Реализация маскирования на базе мессенджера

Архитектура разработанного тестового приложения основывается на принципах модульности и масштабируемости, что позволяет легко адаптировать систему под изменяющиеся требования и обеспечивать высокую производительность даже при больших объёмах передаваемых данных. В качестве основного языка программирования выбран Python, благодаря его гибкости, обширной экосистеме библиотек и простоте разработки [5].

Для создания графического интерфейса пользователя (GUI) использовалась библиотека Tkinter, предоставляющая базовые элементы управления и возможность быстрого прототипирования [6]. Выбор этой библиотеки обусловлен ее простотой интеграции с Python и возможностью легкой адаптации под нужды демонстрационного проекта. Интерфейс приложения выполнен минималистично, с акцентом на функциональность, а не на визуальную привлекательность, что позволило сосредоточиться на

основной задаче – проверке корректности работы механизма маскирования изображений.

Серверная часть приложения реализована с использованием gRPC (gRPC Remote Procedure Calls), который представляет собой высокопроизводительный фреймворк для удалённого вызова процедур [7]. Этот выбор был сделан ввиду ряда преимуществ gRPC перед традиционными RESTful API, включая меньшую задержку при передаче больших объёмов данных, улучшенную производительность и поддержку двунаправленной потоковой передачи данных [8]. Кроме того, gRPC предоставляет встроенную поддержку аутентификации и шифрования, что особенно важно для обеспечения безопасности передаваемых сообщений.

Для маскирования изображений в рамках данного исследования воспользуемся криптографическим методом, известным как алгоритм Хилла. Этот подход основывается на применении матричных операций над фрагментами изображения, что позволяет эффективно скрыть визуальную информацию путём её математической трансформации. Алгоритм Хилла использует ключевые матрицы заданного порядка, которые обеспечивают нелинейность преобразования, затрудняя тем самым восстановление исходного изображения без знания правильного ключа [9]. В матричном виде данный метод описывается следующим образом:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} * \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

где p – открытый текст, k – ключ, c – шифртекст.

Метод заключается в следующем: изображение на стороне клиентского приложения разбивается на блоки пикселей, каждый из которых представлен вектором значений интенсивности цвета. Затем эти векторы подвергаются

операции умножения на заранее выбранную ключевую матрицу, которая действует как «шифрующий» элемент. После применения этой процедуры к каждому блоку пикселей получается замаскированный вариант изображения, который внешне выглядит как случайный набор пикселей. Изображение преобразуется в base64 строку и передаётся по защищённому gRPC соединению получателю. Получатель в своём клиентском приложении производит декодирование base64 строки и демаскирование полученного защищенного изображения. Стоит отметить, что для ускорения открытия изображений в последующие разы, результат демаскирования сохраняется во внутренней памяти приложения, не сохраняя его в файловой системе. Таким образом отпадает необходимость производить вычисления каждый раз при открытии одно и того же файла. Однако, при перезагрузке приложения. данную операцию необходимо повторить снова. На рис.1 изображена схема работы маскирования на базе мессенджера.

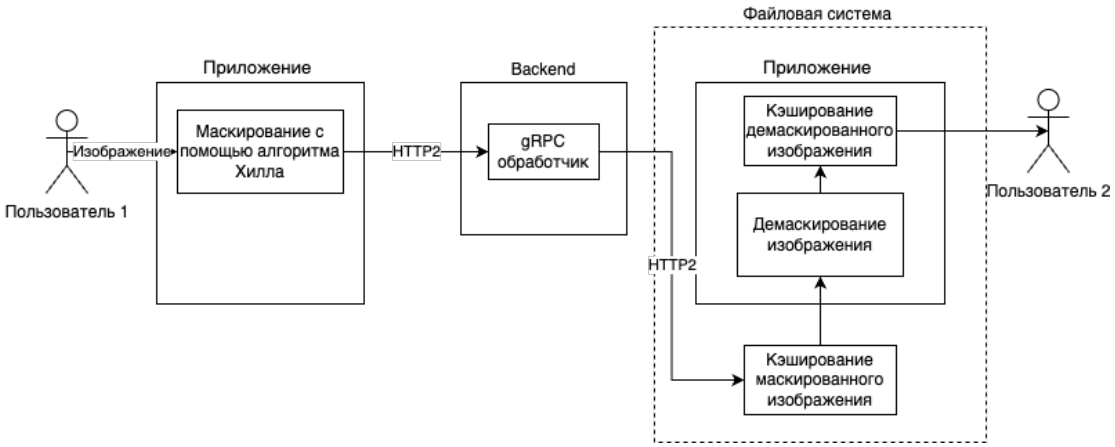

Рис. 1. – Схема работы приложения.

В качестве примера маскирование и демаскирования возьмём два изображения: рисунок и скриншот текста. На рис. 2 изображён результат маскирования изображения с рисунком (в ленте сообщений) и открытого внутри приложения, демаскированного оригинала.

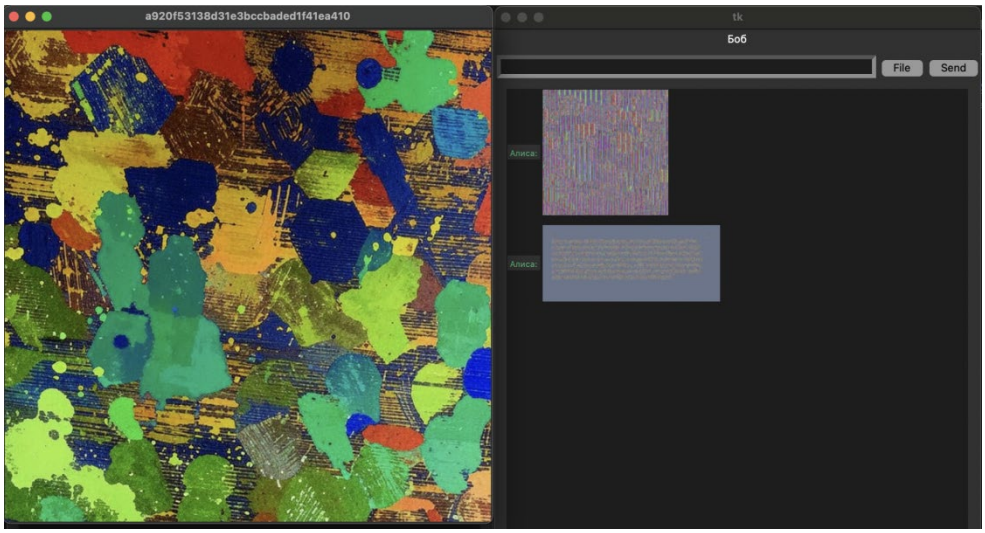

Рис. 2. – Результат маскирования и демаскирования рисунка.

Отправим в качестве изображения скриншот с текстом. На рис.3 изображён результат маскирования скриншота (в ленте сообщений) и его демаскированный оригинал, открытый внутри приложения.

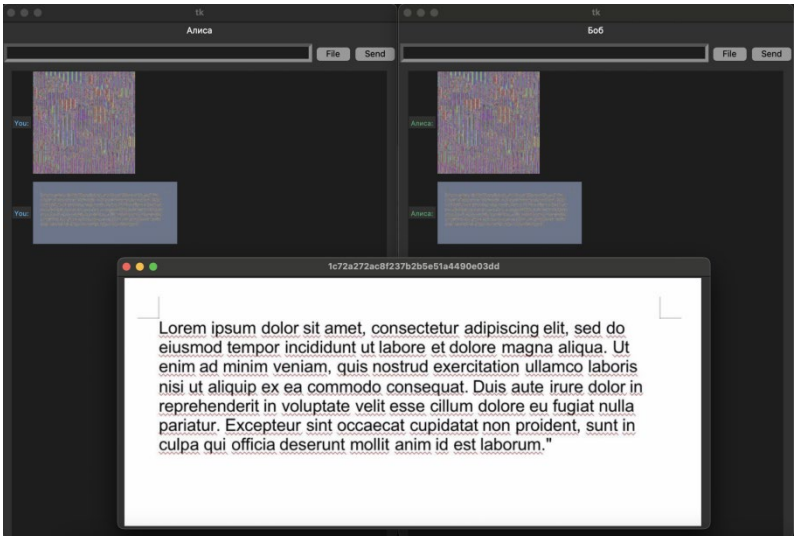

Рис. 3. – Результат маскирования и демаскирования скриншота с текстом.

В целях повышения уровня безопасности можно интегрировать в процесс открытия маскированных изображений механизм двухфакторной аутентификации на основе одноразовых паролей TOTP (Time-based One-Time

Password). Перед началом процедуры демаскирования пользователь будет обязан ввести действующий одноразовый код, сгенерированный его устройством. Только после успешной проверки подлинности кода приложение выполнит преобразование изображения в оригинальный вид. Данная дополнительная мера усилит защиту медиафайлов и может быть применима, например, в корпоративном мессенджере организации с развитой инфраструктурой аутентификации сотрудников. Такой подход позволяет исключить риск несанкционированного просмотра файла из приложения сотрудника. Это особенно важно, поскольку на текущий момент вопрос информационной безопасности корпоративных мессенджеров является актуальной темой и требует всестороннего исследования для предотвращения утечек данных и защиты конфиденциальной информации [10].

Выводы

В представленной статье исследовалась проблема сохранения файлов изображений, передаваемых посредством мессенджера, в кэше на жёстком диске устройства. Проанализированы потенциальные угрозы, возникающие вследствие этого, включая возможность несанкционированного доступа к конфиденциальным данным. Для устранения указанной проблемы предложено решение, основанное на использовании метода маскирования изображений с применением ортогональных матриц. Рассмотрены преимущества интеграции данного подхода в архитектуру приложений мессенджеров.

Разработана экспериментальная версия приложения, реализующая функционал отправки маскированных изображений с последующим демаскированием непосредственно в контуре самого приложения. Дополнительно предложены рекомендации по внедрению дополнительных

мер безопасности для защиты конфиденциальных изображений в рамках функционала мессенджера.

Литература:

1. Сила кэширования: повышаем производительность API и масштабируемость // Хабр URL: habr.com/ru/companies/spaceweb/articles/825030/ (дата обращения: 09.12.2024).

2. Telegram Self-Destruct? Not Always. URL: trustwave.com/en-us/resources/blogs/spiderlabs-blog/telegram-self-destruct-not-always/ (дата обращения 17.11.2024)

3. Muktyas I. B., Sulistiawati S., Arifin S. Digital image encryption algorithm through unimodular matrix and logistic map using Python //AIP Conference Proceedings. – AIP Publishing, 2021. – Т. 2331. – №. 1. – P. 020006.

4. М Б. Сергеев, Т М. Татарникова, А М. Сергеев, В В. Боженко Метод обеспечения конфиденциальности данных с применением ортогональных матриц // Инженерный вестник Дона, 2024, № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8967

5. Nagpal A., Gabrani G. Python for data analytics, scientific and technical applications //2019 Amity international conference on artificial intelligence (AICAI). – IEEE, 2019. – pp. 140-145.

6. Lundh F. An introduction to tkinter // URL: pythonware. com/library/tkinter/introduction/index. htm. – 1999. – Т. 539. – P. 540.

7. What is grpc // gRPC URL: grpc.io/docs/what-is-grpc/introduction/ (дата обращения: 25.11.2024).

8. Buzhin I. G. et al. Comparative Analysis of the REST and gRPC Used in the Monitoring System of Communication Network Virtualized Infrastructure //T-Comm-Телекоммуникации и Транспорт. – 2023. – Т. 17. – №. 4. – pp. 50-55.

9. Acharya B. et al. Image encryption using advanced hill cipher algorithm //International Journal of Recent Trends in Engineering. – 2009. – Т. 1. – №. 1. – pp. 663-667.

10. Коренева А. М., Саварин И. Сравнительный обзор безопасности популярных корпоративных мессенджеров // Инженерный вестник Дона, 2024, №. 8. URL: ivdon.ru/ru/magazine/archive/n8y2024/9416

References

1. Sila keshirovaniya: povyshaem proizvoditel'nost' API i masshtabiruemost' [The power of caching: improving API performance and scalability] Khabr. URL: habr.com/ru/companies/spaceweb/articles/825030/ (date assessed: 09.12.2024).

2. Telegram Self-Destruct? Not Always. URL: trustwave.com/en-us/resources/blogs/spiderlabs-blog/telegram-self-destruct-not-always/ (date assessed 17.11.2024)

3. Muktyas I. B., Sulistiawati S., Arifin S. AIP Conference Proceedings. AIP Publishing, 2021. Т. 2331. №. 1. P. 020006.

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

5. Nagpal A., Gabrani G. 2019 Amity international conference on artificial intelligence (AICAI). IEEE, 2019. P. 140-145.

6. Lundh F. An introduction to tkinter. URL: pythonware. com/library/tkinter/introduction/index. htm. 1999. Т. 539. P. 540.

7. What is grpc URL: grpc.io/docs/what-is-grpc/introduction/ (date assessed: 25.11.2024).

8. Buzhin I. G. et al. T-Comm-Telekommunikatsii i Transport. 2023. Т. 17. №. 4. P. 50-55.

9. Acharya B. International Journal of Recent Trends in Engineering. 2009. Т. 1. №. 1. pp. 663-667.

10. Koreneva A. M., Savarin I. Inzhenernyi vestnik Dona, 2024, №. 8. URL: ivdon.ru/ru/magazine/archive/n8y2024/9416

Дата поступления: 15.09.2024
Дата публикации: 26.12.2024