Хакер - Криптостойкие андроиды. Почему шифрование в Signal, WhatsApp, Telegram и Viber не защитит твою переписку от взлома
nopaywall

https://t.me/nopaywall

Содержание статьи

Сквозное (end-to-end) шифрование в мессенджерах завоевало популярность тем, что оно происходит совершенно незаметно для пользователей. Им не надо самостоятельно генерировать пары ключей, подписывать их, распространять открытые и оберегать секретные ключи, вовремя отзывать старые и скомпрометированные — все делается автоматически, а переписка волшебным образом оказывается защищенной. Но так ли все хорошо на самом деле?

WARNING

Вся информация в этой статье предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Еще в 2004 году наш соотечественник Никита Борисов совместно с Ианом Голдбергом разработал универсальный криптографический протокол для систем мгновенного обмена сообщениями. Протокол получил название OTR (Off-the-Record Messaging) и начал открыто распространяться под лицензией GPL в виде готовой библиотеки. В дальнейшем OTR стал основой других популярных протоколов с дополнительными методами повышения безопасности. В частности, протокола Signal, ранее известного как TextSecure. На базе Signal работает и большинство других современных мессенджеров.

Принципы шифрования переписки

Концептуально все криптографические способы защиты переписки должны обеспечивать как минимум два базовых свойства: конфиденциальность и целостность сообщений. Конфиденциальность подразумевает, что только собеседники могут расшифровать сообщения друг друга. Ни интернет-провайдер, ни разработчик мессенджера, ни какая-то иная третья сторона не должны иметь технической возможности выполнять дешифровку за разумное время. Целостность обеспечивает защиту от случайных искажений и целенаправленных атак подмены. Любое измененное при передаче сообщение будет автоматически отклонено принимающей стороной как поврежденное и утратившее доверие.

В современных протоколах обмена мгновенными сообщениями также решаются дополнительные задачи, повышающие удобство и безопасность. В протоколе Signal и его ближайших аналогах это такие свойства, как асинхронность передачи, прямая и обратная секретность.

Наверняка ты замечал, что в мессенджерах доставляются пропущенные сообщения. Они приходят даже в том случае, если ты беседовал в групповом чате и вдруг надолго отключился посреди разговора. Это и есть асинхронность: сообщения шифруются и доставляются независимо друг от друга. При этом за счет временных меток и некоторых дополнительных механизмов сохраняется их логическая последовательность.

Такое свойство, как **прямая секретность**, подразумевает, что при компрометации ключа шифрования текущего сообщения с его помощью нельзя будет расшифровать предыдущую переписку. Для этого у мессенджеров часто меняются сессионные ключи, каждый из которых шифрует свою небольшую порцию сообщений.

Аналогично **обратная секретность** обеспечивает защиту будущих сообщений при компрометации текущего ключа. Новые ключи генерируются таким образом, что их взаимосвязь с предыдущими вычислить крайне сложно.

Прямая и обратная секретность реализованы в современных механизмах управления ключами. В протоколе Signal для этого используется алгоритм «Двойной храповик» (Double ratchet, DR). Он был разработан в 2013 году консультантом по криптографии Тревором Перрином (Trevor Perrin) и основателем Open Whisper Systems Мокси Марлинспайком (Moxie Marlinspike).
Название является отсылкой к механической шифровальной машине Enigma, в которой использовались храповики — шестеренки с наклонными зубцами, двигающиеся только в одном направлении. За счет этого в шифровальной машине исключалось состояние шестеренок, повторяющее одно из недавно использованных.
По аналогии с ними «цифровой храповик» также препятствует повторному использованию прежних состояний шифрсистемы. DR часто меняет сессионные ключи, при этом не давая повторно использовать ранее сгенерированные. Этим он как раз и обеспечивает прямую и обратную секретность, то есть дополнительную защиту отдельных сообщений. Даже в случае удачного подбора одного сессионного ключа атакующая сторона сможет расшифровать только зашифрованные им сообщения, а это всегда малая часть переписки.
В протоколе Signal реализовано и множество других интересных механизмов, описание которых выходит за рамки статьи. С результатами его аудита можно ознакомиться здесь.

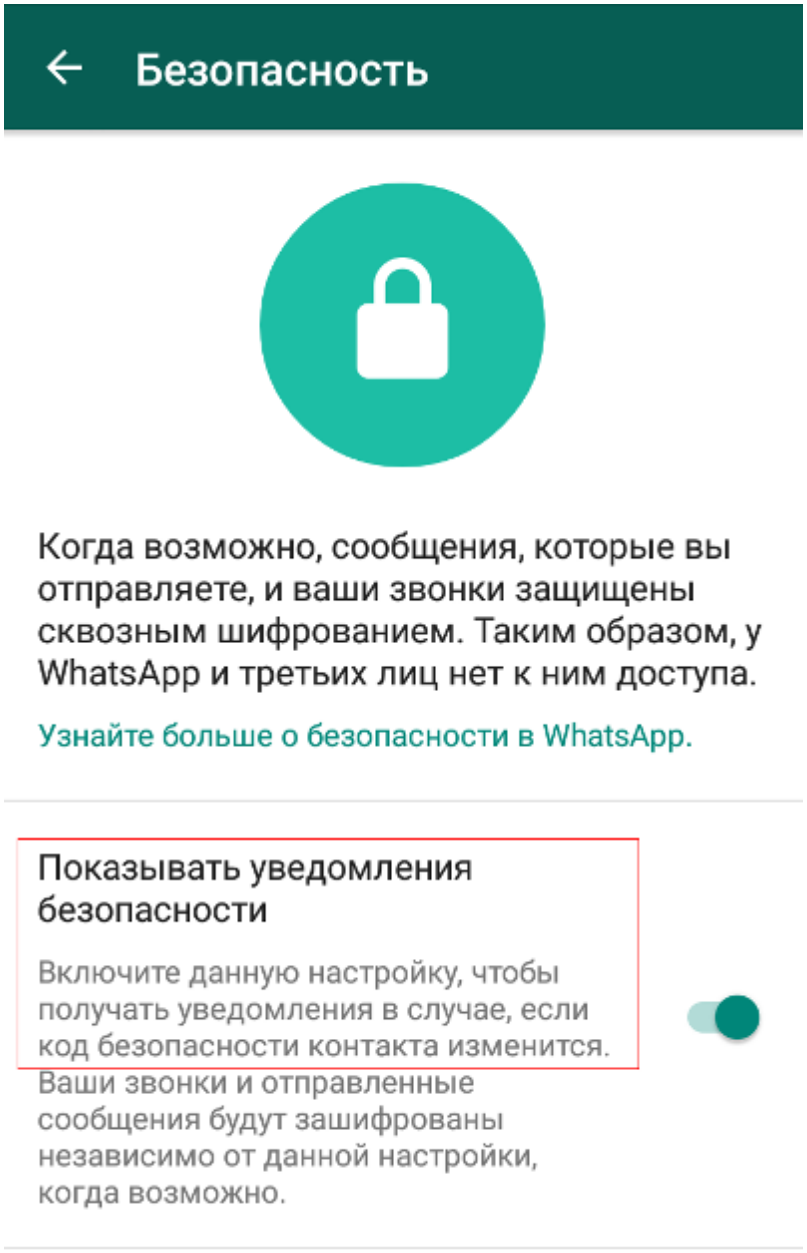
Signal и его аналоги

Предоставляемое Signal сквозное шифрование сегодня применяется как в одноименном мессенджере от Open Whisper Systems, так и во многих сторонних: WhatsApp, Facebook Messenger, Viber, Google Allo, G Data Secure Chat — все они используют оригинальную или слегка модифицированную версию Signal Protocol, иногда давая им собственные названия. Например, у Viber это протокол Proteus — по сути, тот же Signal с другими криптографическими примитивами.
Однако при схожей реализации сквозного шифрования приложение может компрометировать данные другими способами. Например, WhatsApp и Viber имеют функцию резервного копирования истории переписки. Вдобавок WhatsApp отправляет статистику общения на серверы Facebook. Защита у локальной и облачной копии переписки формальная, а метаданные вообще никак не шифруются — об этом открыто говорится в лицензионном соглашении.
По метаданным видно, кто с кем общается и как часто, какие устройства для этого использует, где при этом находится и так далее. Это огромный пласт косвенной информации, которую можно использовать против собеседников, считающих свой канал связи защищенным. Например, АНБ неважно, какими именно словами подозреваемый

поздравил Ассанжа с оставлением Обамы в дураках и что Джулиан ему ответил. Важно то, что они переписываются.

Как уже говорилось выше, все мессенджеры периодически меняют сессионные ключи шифрования, и это нормальный процесс. Основной же ключ может смениться, если собеседник перебрался на другое устройство, надолго ушел в офлайн… или кто-то начал писать от его имени, угнав аккаунт.

В оригинальном приложении Signal всем участникам беседы в таком случае отправляется уведомление о смене ключа. В WhatsApp и других мессенджерах эта настройка по умолчанию отключена, так как она не несет большинству пользователей значимой информации. Также ключ меняется при долгом отсутствии собеседника онлайн — это и баг, и фича одновременно.



Включение уведомления о смене ключа в WhatsApp

Как писал по этому поводу исследователь из Калифорнийского университета в Беркли Тобиас Бёлтер (Tobias Boelter), при атаке на сервис возможно создать новый ключ и

получить сообщения вместо адресата. Более того, то же самое могут сделать и сами операторы серверов WhatsApp — например, по запросу спецслужб.
Разработчики протокола Signal опровергают выводы Бёлтера и встают на защиту WhatsApp. По их словам, подмена ключа дает доступ только к недоставленным сообщениям. Слабое утешение.
Включить уведомление о смене ключа можно в настройках, вот только на практике этот режим параноика вряд ли что-то даст. Мессенджер уведомляет о смене ключа только после повторной отправки сообщений. Считается, что так удобнее самим пользователям.

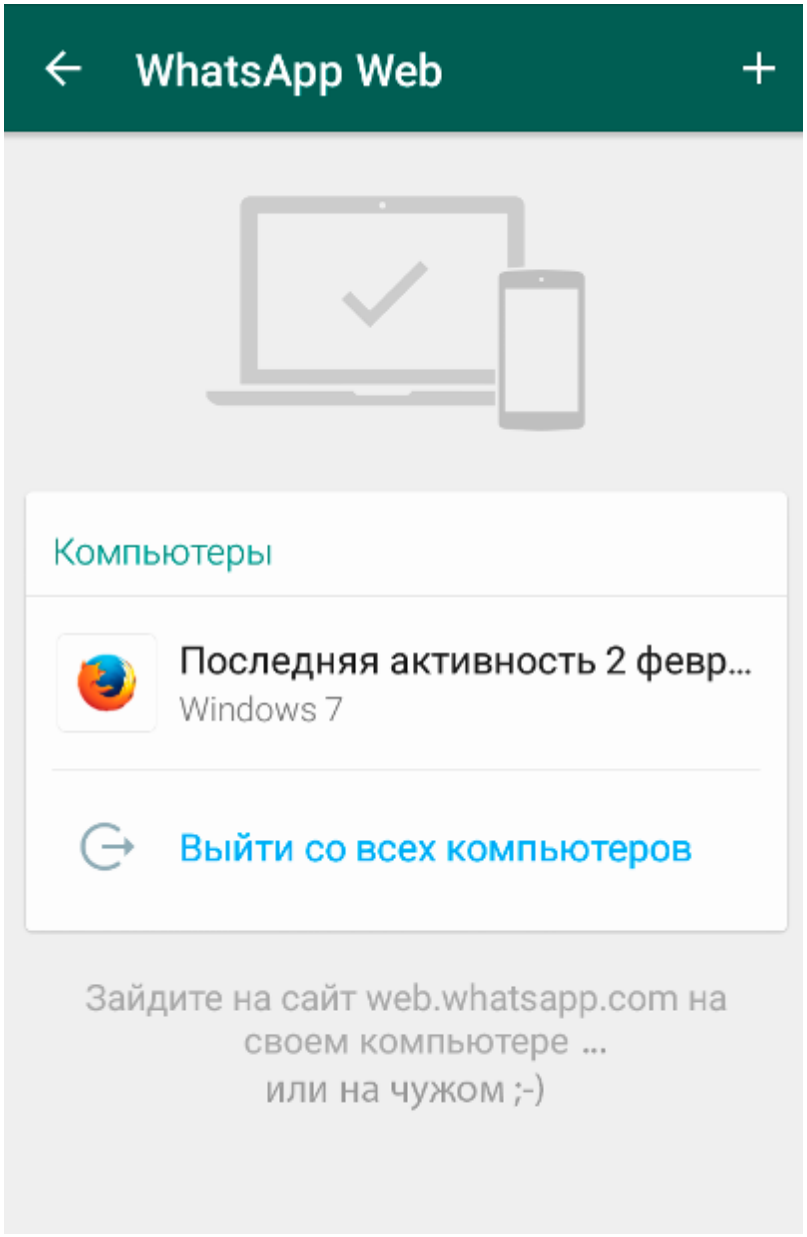
Способы вскрытия

Допустим, мы вняли этим аргументам. Примем в качестве рабочего предположения, что протокол Signal не имеет практически значимых уязвимостей. И что же? Проблема шифрования переписки остается, поскольку у Signal, WhatsApp, да и у других мессенджеров сквозное шифрование гарантирует конфиденциальность только в том случае, когда у атакующей стороны нет ничего иного, кроме перехваченных сообщений в зашифрованном виде.
На практике ФБР и родственные этому бюро ведомства при фоновом наблюдении за человеком обходятся метаданными его коммуникаций, а сами сообщения при необходимости получают другими способами, не требующими ни вскрытия стойкого протокола шифрования, ни факторизации длинных ключей.
В качестве доказательства надежности какой-либо криптосистемы часто приводят результаты соревнований по ее взлому. Дескать, никто так и не забрал объявленный приз, а значит, не смогли взломать. Здесь происходит типичная подмена понятий. Одно дело — прочитать секретные сообщения живого собеседника, и совсем другое — выполнить условия конкурса на взлом диалога ботов (или разработчиков мессенджера, ожидающих подвоха в каждом сообщении). Обычно условия пишутся так, что конкурсантам в итоге предъявляется задача, заведомо не решаемая за отведенное время.
В реальных условиях охотники за чужой перепиской не ограничены какими-либо правилами. Они не обязательно станут искать дыры в самом протоколе сквозного шифрования, а будут ломать то, что проще. Использовать социальный инжиниринг (поэтому я и написал про живых людей), уязвимости в ОС (в Android их тысячи), драйверах и стороннем ПО — любые мыслимые трюки. Нормальные герои всегда идут в обход, и сотрудники трехбуквенных ведомств не исключение.
При наличии физического доступа к смартфону (даже кратковременного и без рута) тем более появляется множество новых векторов атаки, выходящих за рамки конкурса на

взлом мессенджера. Обычно в таком случае удается использовать «не баг, а фичу» приложения, оставленную разработчиками для удобства (взлома).
Приведу пример. В нашей лаборатории нередко бывало, что сотрудник выходил на пару минут и оставлял свой смартфон на зарядке. Смартфоны были у каждого, а розеток не хватало. Поэтому мы выделили специальный стол с сетевым фильтром — эдакую заправочную станцию, где в течение дня лежали все или почти все смартфоны. Естественно, мы по десять раз на день подходим к этому столу, берем свои (а иногда и чужие — по ошибке) смартфоны и кладем их заряжаться дальше. Однажды мне потребовалось узнать, что пишет в мессенджерах Вася. Было подозрение, что он сливает информацию по проектам, а наша служба безопасности лишь разводила руками. Сквозное шифрование — неприступная стена. Концепция BYOD у нас не прижилась. Запретить же пользоваться мессенджерами и смартфонами вообще тоже пытались, но ничего хорошего из этого не вышло. Слишком много коммуникаций на них завязано сегодня. Поэтому с одобрения службы безопасности (п. 100500: «…в исключительных случаях имеет право…») я просто выбрал удобный момент и сделал вот что:

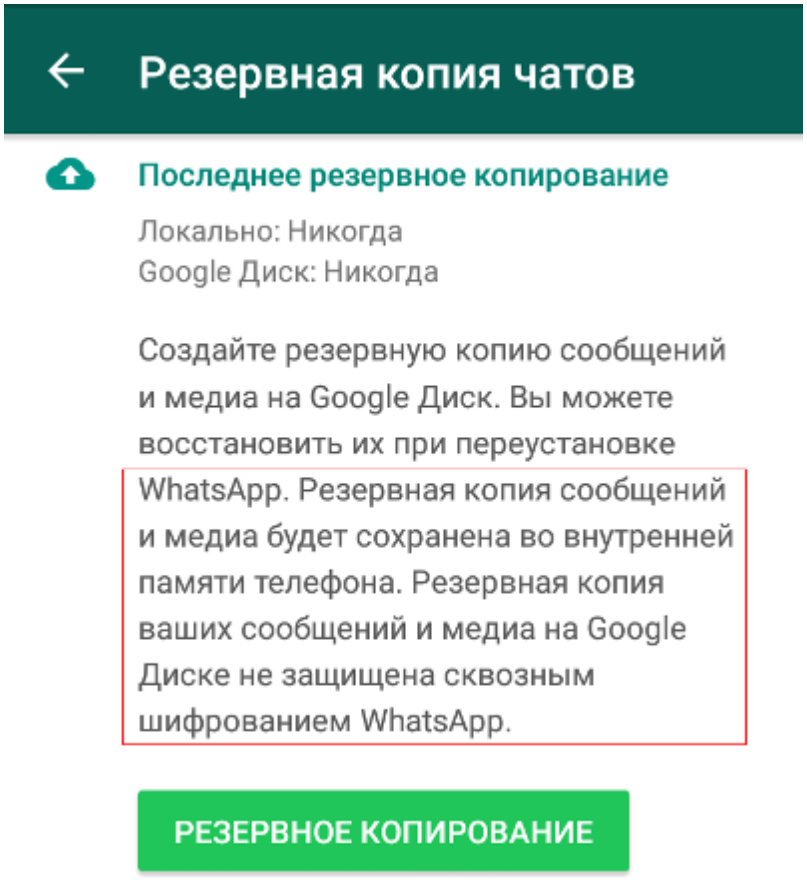
1. Дождался, когда Вася пойдет за едой. Это минимум три минуты, а мне хватит и двух.
2. Спокойно беру его смартфон и сажусь обратно на свое место.
3. Смартфон заблокирован, но я знаю графический ключ. Вася его сотни раз использовал при мне. Поневоле запомнишь эту «букву зю».
4. Запускаю на своем компьютере браузер и перехожу на страницу веб-интерфейса WhatsApp. На ней генерируется QR-код синхронизации.
5. Открываю на смартфоне Васи WhatsApp. Иду в «Чаты → Настройки → WhatsApp Web».
6. Сканирую смартфоном QR-код.
7. Всё. Полная история чатов Васи загружена в моем браузере.
8. Удаляем следы и возвращаем чужой смартфон на место.

Теперь я вижу всю прошлую и текущую переписку Васи. Я буду видеть ее как минимум до конца дня, пока WhatsApp не сменит ключ или Вася вручную не отключит веб-сессию. Чтобы ее отключить, он должен заподозрить неладное, затем войти в тот же пункт меню WhatsApp Web. Там он увидит сообщение о последней веб-сессии… которое будет совершенно неинформативным. В нем указывается только город (по GeoIP), браузер и ОС. У нас с Васей все эти переменные полностью совпадают (одна лаборатория, одна сеть, типовые компы с одинаковым софтом). Поэтому повода для беспокойства эта запись ему не дает.



Открытые сессии WhatsApp на других устройствах

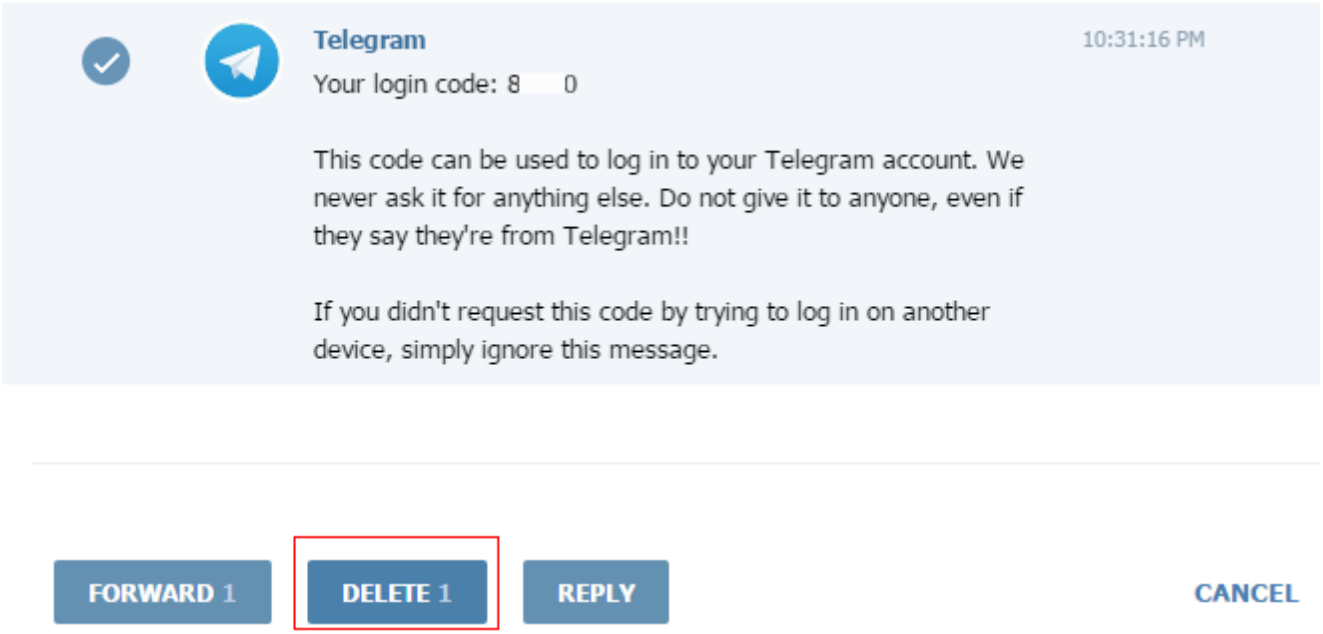
Веб-сессия удобна для текущего наблюдения. Дополнительно можно сделать резервную копию чатов — уже для протокола.



Сохраняем резервную копию сообщений

Еще через несколько дней Вася перешел на Telegram. Метод контроля его переписки в общих чертах был тот же.

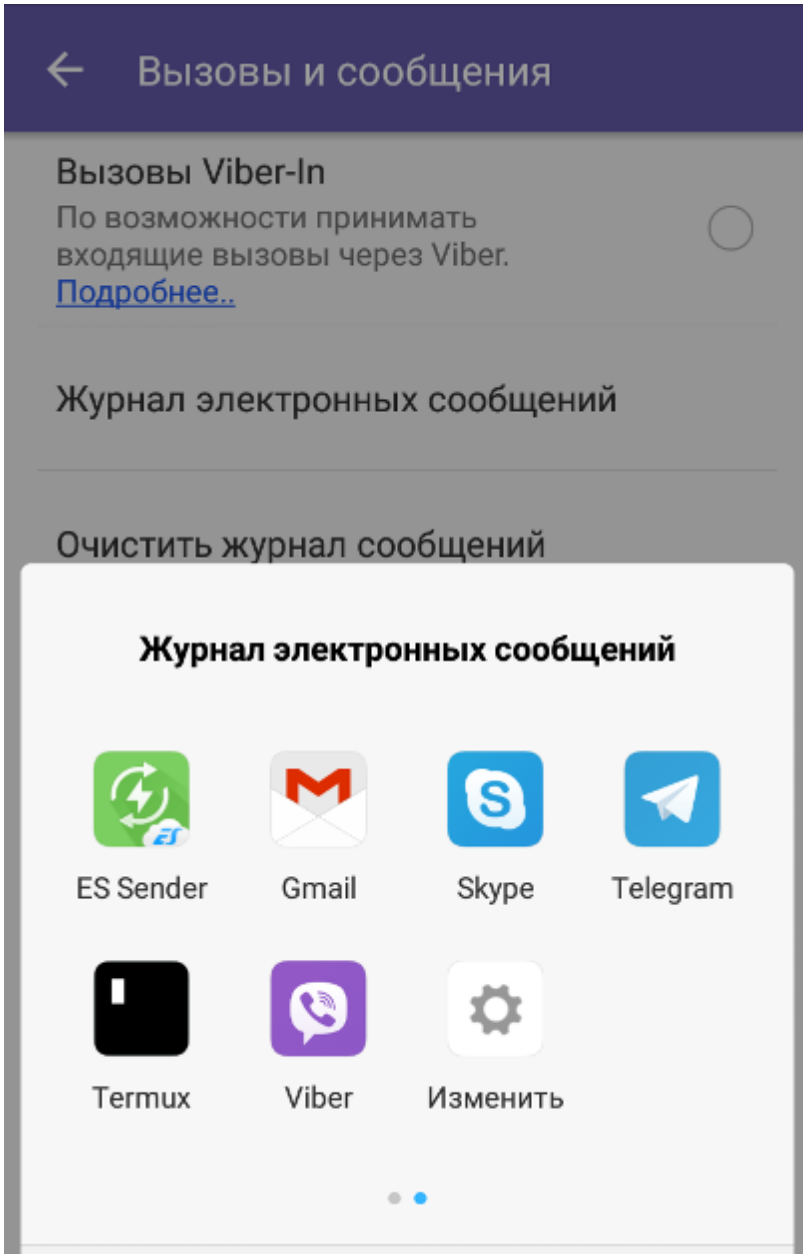
1. Берем его смартфон, разблокируем привычной «буквой зю» и открываем Telegram.
2. Заходим в своем браузере на сайт Telegram.
3. Вводим номер телефона Васи.
4. Ловим код подтверждения, пришедший в его Telegram.
5. Вводим его в окне своего браузера.
6. Удаляем сообщение и все следы.



Заметаем следы после клонирования веб-сессии в Telegram

Вскоре Вася поставил Viber, и мне пришлось проделать новый трюк.

1. Берем на пару минут его смартфон.
2. Открываем Viber → «Настройки → Вызовы и сообщения → Журнал электронных сообщений».
3. Копируем архив на флешку (OTG) или отправляем его себе любым другим способом. Благо Viber предоставляет их десятки.
4. Возвращаем смартфон и удаляем следы.

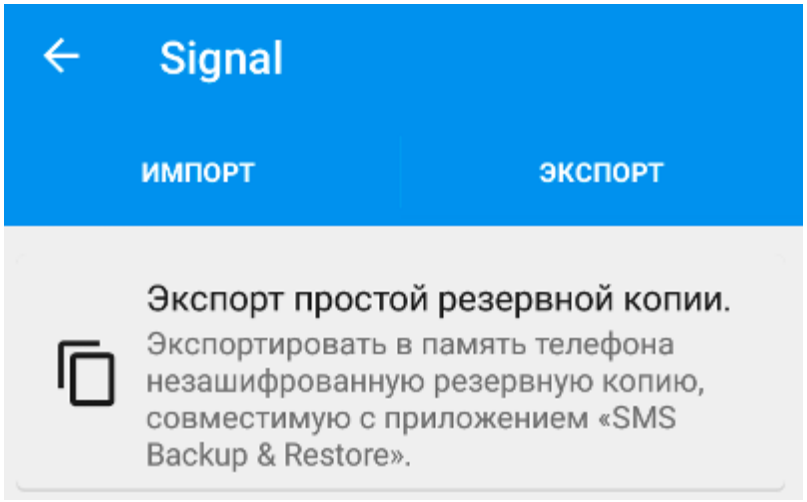


Экспортируем журнал Viber

Дело в том, что у Viber нет веб-версии. Можно было бы установить десктопную и так же связать ее с мобильным аккаунтом Viber Васи, но я выбрал тот метод, который проще было реализовать.

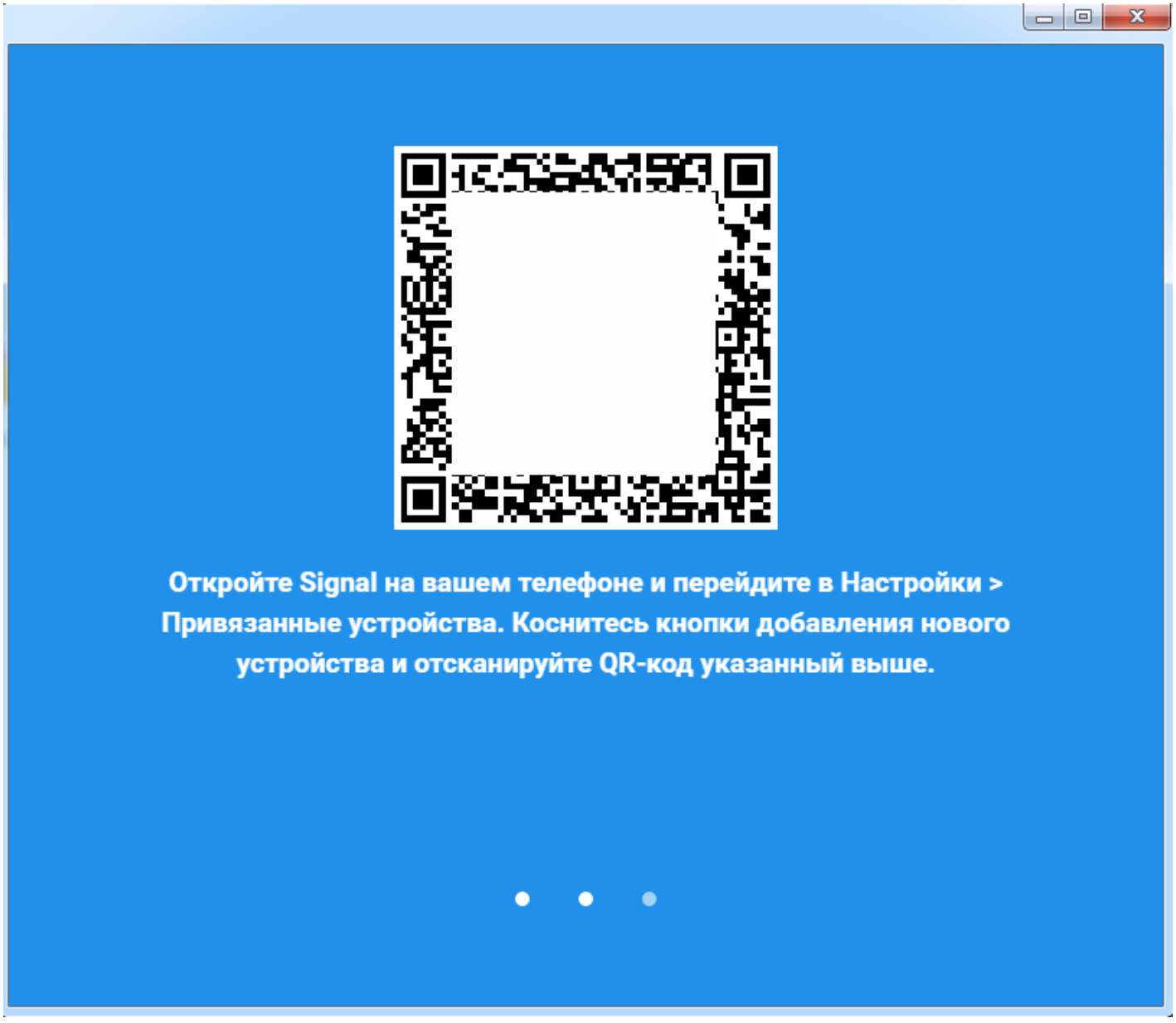
Вася «сел на измену» и поставил Signal. Черт, это же образцовый мессенджер, рекомендованный Шнайером, Сноуденом и Фондом электронных рубежей! Он даже скриншоты чата не дает сделать самому пользователю. Как же быть?

Снова дожидаемся удобного момента и запускаем Signal на Васином смартфоне. Мессенджер требует ввести парольную фразу, которую я не знаю… но я знаю Васю! Пробую его день рождения — не подходит. Пробую код от нашего лабораторного дипломата — подошел. Даже скучно. Идем в настройки мессенджера и останавливаемся, словно витязь на распутье. Оказывается, вариантов добраться до чатов много. Например, Signal позволяет одной командой экспортировать всю переписку, причем только в открытом виде.



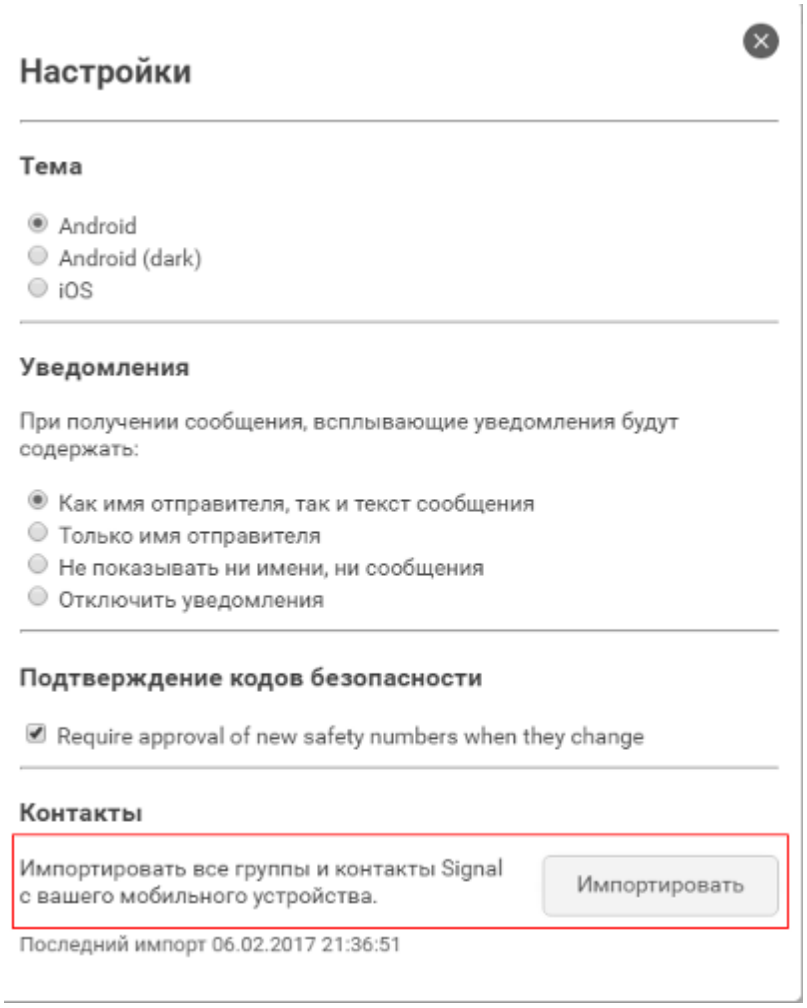
Экспортные ограничения? Не слышали!

Затем можно направиться в «Настройки → Привязанные устройства» и повторить трюк, уже проделанный с WhatsApp ранее. Signal точно так же открывает веб-сессию через QR-код. Для этого даже есть отдельное расширение в Google Chrome.



Клонируем все чаты Signal в Chrome

Бонусом из веб-сессии Signal можно утащить все контакты. Пригодятся.



Импортируем контакты через веб-сессию Signal

Итог: я не знаю ключей шифрования Васи (да он и сам их не знает!), но могу читать его прошлую и текущую переписку во всех мессенджерах. Вася ничего не подозревает и продолжает верить в то, что «сквозное шифрование» гарантирует ему полную конфиденциальность.

С физическим доступом к смартфону получить контроль над любым мессенджером становится просто, но даже он необязателен для взлома переписки. Можно заманить жертву на фишинговую ссылку и удаленно протроянить смартфон — в старых версиях Android и предустановленном браузере дыр хватает. Троян получит рут (сейчас это рутинная автоматическая процедура), начнет делать скриншоты, дампы памяти… или просто облегчит резервное копирование всех чатов очередного мессенджера в открытом виде.

Telegram

Про этот мессенджер стоит поговорить отдельно по целому ряду причин. Во-первых, он использует другой протокол сквозного шифрования — MTProto. После избавления от детских болезней (раз, два, три) и публикации большей части исходных кодов его можно рассматривать как перспективную альтернативу протоколу Signal.

Во-вторых, само приложение Telegram не хранит локальные копии переписки. Вся она загружается с сервера. Поэтому скопировать лог чатов, как в Viber и многих других мессенджерах, не получится.
В-третьих, в Telegram есть секретные чаты, для которых реализованы дополнительные механизмы защиты. Например, во всплывающих уведомлениях не показывается текст секретных чатов. При удалении у одного собеседника секретные сообщения удаляются и у другого. Можно даже запрограммировать автоматическое уничтожение сообщения через заданное время. В любом случае после доставки все секретные сообщения удаляются с серверов Telegram.
Проверить это можно только косвенно, сменив устройство. На новом смартфоне после авторизации в Telegram удастся восстановить всю переписку, кроме секретных чатов. Происходит так потому, что ключ шифрования переписки в секретных чатах привязан к конкретному устройству. Во всяком случае, так утверждается в официальном FAQ.
Что же делать, если Вася будет использовать секретные чаты Telegram с автоматически удаляющимися сообщениями? Как вариант, воспользоваться одной особенностью Android под названием Screen after Previous Screens.
Кратко суть метода состоит в том, что Android кеширует в оперативной памяти изображения экранов запущенных приложений и хранит их некоторое время. Делается это для того, чтобы пользователь быстрее мог переключаться между приложениями, не дожидаясь полной отрисовки их окон заново.
Утилита RetroScope с открытым исходным кодом умеет вытаскивать из памяти смартфона десяток последних снапов экрана (или больше, если повезет), среди которых попадается что угодно — включая секретные чаты Telegram (в том числе уже удаленные) и чаты Signal, которые вообще нельзя заскринить штатным образом.

Криптофоны

Android — сложная операционная система, а степень защиты любой системы соответствует таковой для самого слабозащищенного компонента. Поэтому на рынке появляются криптофоны — максимально защищенные смартфоны. Silent Circle выпустила две версии Blackphone. BlackBerry создала Priv, а Macate Group представила в прошлом году GATCA Elite.
Парадокс в том, что, пытаясь сделать Android надежнее, все эти компании создают его более консервативную, усложненную… и менее надежную версию. Например, в Blackphone приходится использовать старые версии приложений из проекта AOSP, в которых накапливается много известных уязвимостей. Обновлять их вручную оперативно не получится, а ставить магазин приложений — это самому создавать брешь в охранном периметре.

Доходит до абсурда: предустановленное приложение для защищенной переписки SilentText долгое время использовало библиотеку libscimp, в которой давно выявлена утечка памяти. Достаточно было отправить модифицированное сообщение, чтобы его команды выполнились от локального пользователя и предоставили удаленный доступ к содержимому Blackphone.

Проблема сотовых операторов

Двухфакторная аутентификация была призвана усложнить взлом аккаунта, но на деле она лишь переложила проблемы безопасности на хрупкие плечи сотовых операторов и создала новые бреши. К примеру, в апреле прошлого года два сотрудника «Фонда борьбы с коррупцией» (ФБК) сообщили о взломах своих аккаунтов Telegram. Оба пострадавших использовали двухфакторную аутентификацию. Они считают, что взлом их аккаунтов произошел при непосредственном участии МТС. Предполагается, что недобропорядочные сотрудники сотового оператора клонировали SIM-карты и передали их злоумышленникам. Это позволило получить им СМС-коды подтверждения для авторизации в Telegram.
Клонирование SIM-карты — тривиальная процедура, выполняемая в любом салоне оператора связи. Я много раз пользовался ей для замены испорченной симки… и не всегда у меня спрашивали паспорт. Более того, между активацией новой симки и отключением старой есть небольшое временное окно. Об этом я узнал случайно, когда забыл отдать испорченную симку и она вдруг ожила у меня уже после выдачи новой.

Уязвимости старых версий Android

Любое криптографическое приложение может безопасно использоваться только в том случае, если запущено в доверенной среде. Смартфоны с ОС Android этому условию не удовлетворяют совершенно. WhatsApp, Telegram и другие мессенджеры готовы запускаться даже на старой Android 4.0 Ice Cream Sandwich, эксплоитов для которой просто тьма. Если же мессенджеры ограничат возможность запуска только последними версиями Android, то лишатся 99% пользователей.

Шифрование де-юре и де-факто

Использование сквозного шифрования во всех популярных мессенджерах стало стандартом де-факто. Его юридический статус в настоящее время не вполне определен.

С одной стороны, свобода переписки и запрет на цензуру гарантируются конституцией во многих странах. С другой — такое шифрование противоречит новым российским законам из «пакета Яровой» и законодательным актам «антитеррористической направленности» в США. Google, Facebook и другие компании обязаны соблюдать законы тех стран, в которых работают. Если их принудят дать доступ к переписке, они будут вынуждены «сотрудничать» с правительством.
Пока отработанного механизма принуждения еще нет, стойкое шифрование в мессенджерах называют головной болью для спецслужб всего мира. Директор ФБР, министр внутренних дел Франции и многие другие высокопоставленные чиновники заявляли, что их ведомства не могут контролировать такую переписку.
На мой взгляд, это лишь игра на публику. Как говорил Братец Кролик: «Только не бросай меня в терновый куст!» Хотя используемый мессенджерами протокол защищенной передачи данных Signal и его аналоги считаются надежными (а порой и прошедшими серьезный аудит), реальная степень криптографической защиты переписки в них оказывается невысокой из-за человеческого фактора и дополнительных функций в самих приложениях. Формально архивация чатов, их дублирование в облако, перенос на другое устройство и автоматическая смена ключей создавались ради удобства… вот только чьего именно?

А как же PGP?

Первое, что обычно приходит на ум при упоминании зашифрованной переписки, — это PGP. Однако далеко не все реализации этой популярной криптосистемы с открытым ключом в равной степени безопасны. Различные методы ослабления криптостойкости официально использовались США для экспортных продуктов, а неофициально — и для всех массовых. Компания Symantec, купившая у Филиппа Циммермана права на PGP и закрывшая исходный код своих продуктов, просто обязана соблюдать действующие ограничения американского законодательства и следовать негласным «рекомендациям» своего правительства.
Поэтому сторонники приватности долгое время считали заслуживающими доверия лишь авторские версии PGP 2.x, которые использовали для шифрования сессионных ключей алгоритм RSA или IDEA. Однако после того как в 2010 году методом решета числового поля удалось за приемлемое время вычислить ключ RSA длиной 768 бит, их надежность тоже перестала считаться достаточно высокой.
Внимание современных хактивистов и прочих правозащитников переключилось на свободные реализации PGP с открытым исходным кодом. Большинство из них позволяет выбирать из нескольких алгоритмов и генерировать более длинные ключи. Однако и здесь не все так просто. Более длинный ключ еще не гарантирует большей

криптостойкости системы. Для этого в ней должны отсутствовать другие недостатки, а все биты ключа быть в равной степени случайными. На практике это часто оказывается не так.
Битовую последовательность ключа всегда формирует какой-то известный генератор псевдослучайных чисел. Обычно это предустановленный в ОС или взятый из готовых библиотек ГПСЧ. Его случайное или преднамеренное ослабление — самая часто встречающаяся проблема. Некогда популярный Dual_EC_DRBG (использовавшийся и в большинстве продуктов компании RSA) непосредственно был разработан в АНБ и содержал закладку. Выяснили это спустя семь лет, уже когда Dual_EC_DRBG использовался повсеместно.
Все реализации PGP, соответствующие стандарту OpenPGP (RFC 2440 и RFC 4880), сохраняют базовую совместимость друг с другом. На смартфонах с ОС Android добавить шифрование PGP к почте можно, например, с помощью приложения OpenKeychain.

OpenKeychain

OpenKeychain имеет открытый исходный код, проверенный компанией Cure53 на безопасность, прозрачно интегрируется с почтовым клиентом K-9 Mail, Jabber-клиентом Conversations и даже может передавать зашифрованные файлы в приложение EDS (Encrypted Data Store), о котором мы писали в прошлой статье цикла.

Выводы

Защиту от перехвата сообщений третьей стороной сегодня в Android эффективно реализуют как классические почтовые приложения (использующие OpenPGP), так и современные мессенджеры, в основе которых лежит принцип сквозного шифрования. При этом конфиденциальность сообщений сохраняется ровно до тех пор, пока у атакующей стороны нет дополнительных преимуществ — таких как физический доступ к устройству или возможность удаленно протроянить его.
Разговоры о том, насколько надежен сам протокол Signal, чем он лучше Proteus или MTProto, интересны лишь самим криптографам. Для пользователей они лишены практического смысла до тех пор, пока в мессенджерах можно делать незашифрованные копии переписки и клонировать текущие сессии. Даже если все

приложения для переписки станут настолько же бронированными (и неудобными), как оригинальный мессенджер Signal, все равно останется множество уязвимостей на уровне ОС Android и человеческий фактор.

WWW

- FAQ про GnuPG на русском
- Использование OpenPGP в Android
- Библиотека Signal Protocol на Java для Android
- Бэкдор в ГПСЧ
- Особенности и ограничения криптосистемы PGP