Сергей В. Запечников, Полина О. Кожухова
Национальный исследовательский ядерный университет «МИФИ», Каширское шоссе, 31,г. Москва, 115409, Россия
e-mail: SVZapechnikov@mephi.ru, ORCID 0000-0002-7975-6040
e-mail: PKozhukhova@yandex.ru, ORCID 0000-0002-4004-5209

О КРИПТОГРАФИЧЕСКОЙ СТОЙКОСТИ СКВОЗНЫХ ЗАЩИЩЕННЫХ СОЕДИНЕНИЙ В МЕССЕНДЖЕРАХ WHATSAPP И TELEGRAM

DOI: http://dx.doi.org/10.26583/bit.2017.4.04

Ключевые слова: криптография, сквозное соединение, шифрование, WhatsApp, Telegram

Аннотация. В статье анализируются возможности повышения стойкости защищенных соединений между пользователями мессенджеров в условиях воздействия внешнего нарушителя и недоверия к провайдеру сервиса. В работе проведено сравнение методов и механизмов криптографической защиты информации, заложенных в основу двух широко распространенных мессенджеров: Telegram и WhatsApp. При этом установлено, что для защиты сквозных соединений в мессенджере Telegram используется протокол MTProto, а в мессенджере WhatsApp — протокол Signal.Изучены особенности реализации мессенджеров на наиболее распространенной мобильной платформе Android, связанные с генерацией случайных чисел. В результате детального анализа каждого из них было выявлено, что лучшим по совокупности свойств безопасности является Signal. Помимо WhatsApp, он используется в ряде других популярных мессенджерах, таких как TextSecure, RedPhone, GoogleAllo, FacebookMessenger, Signal. Выявлены и проанализированы возможные атаки на оба мессенджера. В частности, установлено, что в обоих мессенджерах не защищаются метаданные. Обеспечение безопасности метаданных может стать одной из целей дальнейших исследований.

Для цитирования. **ЗАПЕЧНИКОВ, Сергей В.; КОЖУХОВА, Полина О.** О КРИПТОГРАФИЧЕСКОЙ СТОЙКОСТИ СКВОЗНЫХ ЗАЩИЩЕННЫХ СОЕДИНЕНИЙ В МЕССЕНДЖЕРАХ WHATSAPP И TELEGRAM. Безопасность информационных технологий, [S.l.], v. 24, n. 4, p. 35-43, nov. 2017. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/275>. Дата доступа: 29 nov. 2017. doi:http://dx.doi.org/10.26583/bit.2017.4.04.

Sergey V. Zapechnikov, Polina O. Kozhukhova
National Research Nuclear University MEPhI, Kashirskoe shosse, Moscow, 31, 114509, Russian Federation
e-mail: SVZapechnikov@mephi.ru, ORCID 0000-0002-7975-6040
e-mail: PKozhukhova@yandex.ru, ORCID 0000-0002-4004-5209

On cryptographic security of end-to-end encrypted connections in WhatsApp and Telegram messengers

DOI: http://dx.doi.org/10.26583/bit.2017.4.04

Keywords: cryptography, end-to-end connection, encryption, WhatsApp, Telegram

Abstract. The aim of this work is to analyze the available possibilities for improving secure messaging with end-to-end connections under conditions of external violator actions and distrusted service provider. We made a comparative analysis of cryptographic security mechanisms for two widely used messengers: Telegram and WhatsApp. It was found that Telegram is based on MTProto protocol, while WhatsApp is based on the alternative Signal protocol. We examine the specific features of messengers implementation associated with random number generation on the most popular Android mobile platform. It was shown that Signal has better security properties. It is used in several other popular messengers such as TextSecure, RedPhone, GoogleAllo, FacebookMessenger, Signal along with WhatsApp. A number of possible attacks on both messengers were analyzed in details. In particular, we

demonstrate that the metadata are poorly protected in both messengers. Metadata security may be one of the goals for further studies.

For citation. **ZAPECHNIKOV, Sergey V.; KOZHUKHOVA, Polina O.** On cryptographic security of end-to-end encrypted connections in WhatsApp and Telegram messengers. IT Security, [S.l.], v. 24, n. 4, p. 35-43, nov. 2017. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/275>. Date accessed: 29 nov. 2017. doi:http://dx.doi.org/10.26583/bit.2017.4.04.

Введение

Одним из самых популярных средств персонального обмена информацией в современном обществе становятся мессенджеры – программы для мгновенного обмена сообщениями и файлами с мобильных устройств и настольных компьютеров. Создатели мессенджеров борются за право называть свои продукты самыми защищенными.

Важным аспектом защиты передаваемых с использованием мессенджеров данных является способ шифрования и аутентификации этих данных. Наилучшим на сегодняшний день считается сквозное защищенное соединение, когда исходящее сообщение обрабатывается на мобильном устройстве отправителя, а обработка входящего сообщения также целиком происходит на мобильном устройстве получателя. Оно используется в таких популярных мессенджерах как WhatsApp и Telegram. Общей чертой такого способа обмена сообщениями в мессенджерах является использование сервера только в роли «почтового ящика». Сервер не получает доступа к содержанию пересылаемых им сообщений. Далее в статье будут рассматриваться только такие способы пересылки сообщений.

Большая распространенность персональных средств интерактивного обмена текстовой информацией и файлами в настоящее время сочетается с недостаточной защищенностью используемых этими программными средствами соединений между участниками диалога, что определяет актуальность исследования этой темы.

1 Обмен сообщениями в Telegram

Telegram — бесплатный кроссплатформенный мессенджер для смартфонов и других устройств, позволяющий обмениваться текстовыми сообщениями и медиафайлами различных форматов [1]. По данным на февраль 2016 года количество активных пользователей в месяц у мессенджера Telegram превысило 100 млн [2].

Для безопасной передачи сообщений между собеседниками в Telegram разработчиками предусмотрен специальный тип соединений — секретные чаты, которые обеспечивают сквозное шифрование и аутентификацию сообщений. Секретные чаты предназначены для общения только двух собеседников, в групповых чатах сквозное шифрование не используется. Для передачи сообщений через секретный чат мессенджер Telegram использует собственный протокол MTProto. Анализ протокола и отдельных элементов его реализации в мессенджере Telegram проводился по материалам [3, 4].

В Telegram обмен сообщениями состоит из трех этапов.

1. *Регистрация клиента*. Клиент вводит свой номер телефона, на который по SMS приходит пятизначный код для подтверждения номера. После этого клиент авторизуется на сервере.

2. *Обмен ключами*. На данном этапе Клиенты вырабатывают общий секретный ключ путем обмена параметрами по протоколу Диффи-Хеллмана.

3. *Обработка исходящего сообщения*. Схема обработки сообщений отправителем представлена на рис. 1.

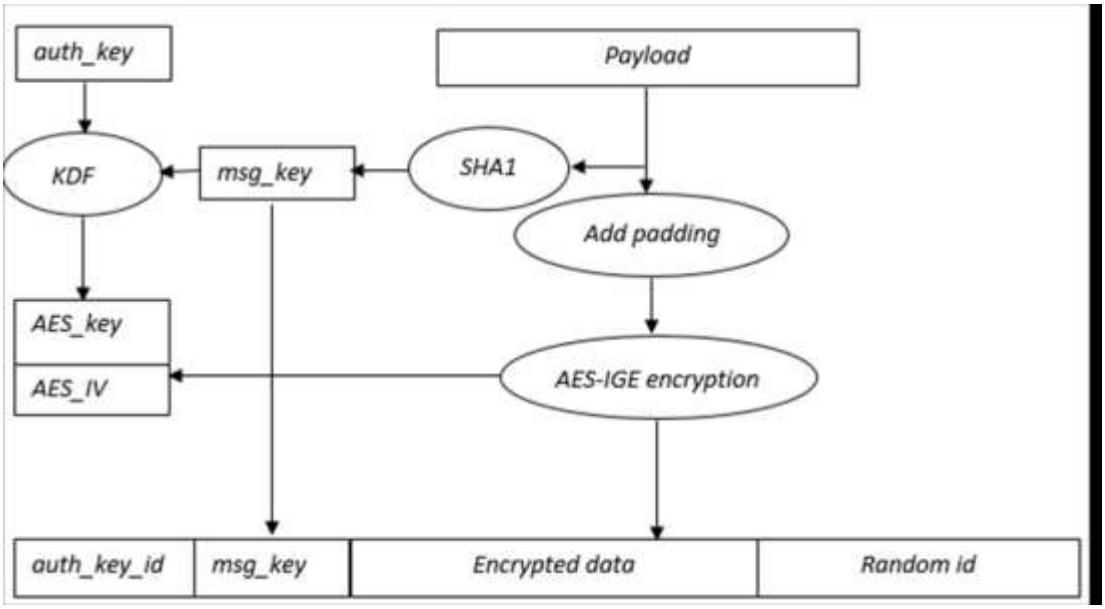

Рисунок 1 - Шифрование сообщений в секретных чатах Telegram
(Fig. 1 - Encryption of messages in a secret chat Telegram)

На рис. 1 приняты следующие обозначения:

-*auth_key*: 2048-битный общий секретный ключ;

-*msg_key*: последние 128 бит SHA1 от сообщения, которое будет зашифровано;

-*padding*: дополнение. 0-96 случайных бит, сгенерированных клиентом, которые добавляются к каждому блоку AES до размера 128 бит;

-*AES key and IV*: 256-битный ключ и 256-битный вектор инициализации, полученные с помощью AES в режиме IGE;

-*auth_key_id*: 64 последних бита SHA1 от auth_key;

-*payload*: заполнение. Заполнение состоит из следующих частей:

• *length*: 32-битное целое число, характеризующее длину Payload (за исключением Length и Padding);

•*header*: 32-битное число, связанное с версией протокола и указанием типа передаваемых медиа-сообщений;

•*random bits*: 128 случайных бит, сгенерированных клиентом, использующихся как соль сообщения;

•*layer*: 32-битное целое число, которое передает информацию о версии протокола клиента;

•*seq_in*: 32-битный счетчик сообщений, отправленных клиентом;

•*seq_out*: 32-битный счетчик сообщений, принятых клиентом;

•*random id*: 64-битное случайное число, сгенерированное клиентом;

•*ttl*: 32-битное целое число, содержащее информацию о количестве секунд, в течение которых принимающий пользователь может увидеть сообщение до того, как оно будет удалено;

•*Padding*: Дополнение. Добавляется перед шифрованием.

-*Message:* отправляемое сообщение.

Новый общий секретный ключ по умолчанию генерируется после отправки каждых очередных ста сообщений.

4. *Обработка входящего сообщения* происходит аналогично этапу 3, но в обратном порядке.

2 Обмен сообщениями в WhatsApp

WhatsApp – бесплатный кроссплатформенный мессенджер с поддержкой голосовой связи. Он позволяет пересылать текстовые сообщения, изображения, видео- и аудиоданные через Интернет. По данным[5], в феврале 2016 г. оценочное число активных пользователей мессенджера WhatsApp превысило миллиард человек и продолжает возрастать.

С апреля 2016 г. в WhatsApp используется сквозное шифрование соединений между пользователями.Сквозное соединение в WhatsApp устанавливается на базе протокола Signal, хотя некоторые точные детали спецификации разработчиками до сих пор не объявлены. В связи с этим анализ протокола и отдельных элементов его реализации в мессенджере WhatsApp проводился по материалам [6 – 8].

В WhatsApp порядок взаимодействия клиентов таков.

1. *Регистрация клиента*. В момент установки мессенджера Клиент генерирует ряд криптографических ключей и регистрируется на Сервере.

2. *Установление сессии*. Клиент, устанавливающий сессию запрашивает с сервера открытые ключи собеседника и оба клиента вырабатывают общие секретные ключи.

3. *Отправка сообщения*. На данном этапе происходит шифрование сообщения ключами, полученными на этапе 2 общими секретными ключами с помощью AES256 в режиме сцепления блоков (CBC).

4. *Обновление «храповика»*. В мессенджере используется специальная криптографическая конструкция – так называемый храповой механизм (ratchet). Он подразумевает обновление ключей собеседников после каждого отправленного и принятого сообщения. На рис. 2 показано, как происходит обновление ключей у клиента при отправке и приеме сообщений. Старые ключи после использования удаляются.

На рис. 3 приняты следующие обозначения:

- *PubK* – открытый ключ;
- *PrK* – секретный ключ;
- *PB1* – открытый ключ клиента B;
- *A1, B*1 – секретные ключи клиентов A и B для шифрования сообщения;
- *RK* – корневой ключ;
- *CH* – ключ цепи.

HKDF определяется следующим образом [9]:

$$\mathrm{HKDF}(XTS, SKM, CTXinfo, L) = K(1)\|K(2)\|...\|K(t), \quad (1),$$

где:

L — число ключевых битов, получаемых в результате применения функции.

K(i) вычисляется по алгоритму, пошагово описанному ниже.

Шаг 1. Вычисляется значение PRK:

$$PRK = \mathrm{HMAC}(XTS, SKM), \quad (2),$$

где:

XTS — случайное заполнение, называемое «солью» (salt), SKM — ключ исходного текста.

Шаг 2. Вычисляется значение K(1):

$$K(1) = \mathrm{HMAC}(PRK, CTXinfo\|0), \quad (3),$$

где:

CTXinfo — ключевая информация (например, идентификаторы алгоритмов, сеансов).

Шаг 3. Вычисляются значения $K(i+1)$:

$$K(i+1) = \mathrm{HMAC}\left(PRK, K(i) \| CTXinfo \| i\right),\ 1 \le i < t\,. \quad (4)$$

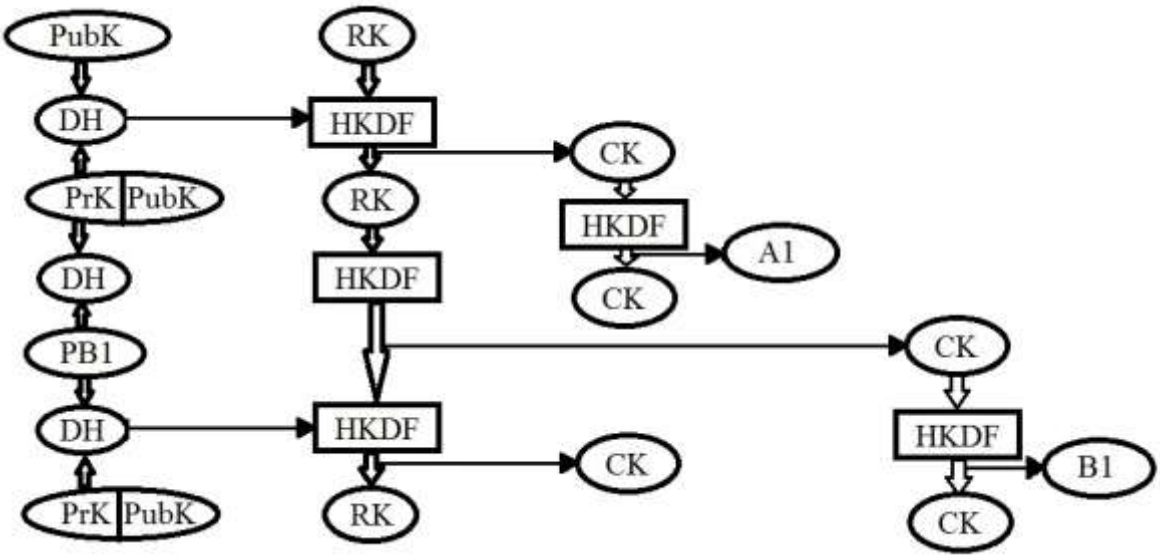

Рисунок 2 - Обновление храповика при отправке и приеме сообщения
(Fig. 2 - Update of the ratchet when sending and receiving messages)

5. *Получение сообщения*. Осуществляется аналогично этапу 3, но в режиме расшифрования.

3 Генерация случайных чисел в устройствах на платформе Android

Мессенджеры используются главным образом на мобильных устройствах. Стойкость используемых в мобильных устройствах криптографических механизмов напрямую зависит от качества генерации случайных величин их операционными системами (ОС). В ходе работы проанализировано устройство генератора случайных величин (ГСЧ) в ОС Android [10]. Выявлено, что он имеет два интерфейса:

- */dev/random* (блокирующий) — возвращает только максимальное число случайных битов, которые содержатся в пуле энтропии. Если в пуле не хватает случайности, то он будет блокировать процесс до того момента, пока счетчик энтропии не накопит достаточное значение;
- */dev/urandom* (неблокирующий) — у данного устройства нет ограничений: будет возвращаться столько битов, сколько запрашивается. Если случайности недостаточно, то поток случайных чисел все равно не прекратится, просто последующие числа будут менее криптостойкими.

В большинстве случаев в качестве случайных чисел берутся значения из неблокирующего пула, что может привести к недостаточному количеству энтропии, и, следовательно, к снижению качества сгенерированных случайных чисел.

Реализация протокола Диффи – Хеллмана на основе эллиптических кривых обладает тем преимуществом, что требуются более короткие ключи, а, следовательно, меньшая зависимость от качества ГСЧ, не способных за короткое время сгенерировать длинную случайную последовательность.

Подробных сведений об устройстве ГСЧ в другой наиболее часто используемой ОС iOS в настоящее время не опубликовано, поэтому судить о качестве этого ГСЧ не представляется возможным.

4 Достоинства и недостатки секретных чатов Telegram

В результате детального анализа алгоритма обработки сообщений в секретных чатах Telegram были выявлены следующие основные достоинства:

- обеспечение защиты от чтения назад, то есть при компрометации будущих ключей предыдущие ключи скомпрометированы не будут;
- обеспечение защиты от чтения вперед, то есть при компрометации предыдущих ключей будущие ключи скомпрометированы не будут;
- возможность устанавливать время жизни отправляемых сообщений.

В то же время следует отметить ряд недостатков принятого в секретных чатах Telegram способа обработки сообщений:

- авторизация по отправке СМС на номер телефона. Недостаток связан с уязвимостью сети SS7. Можно получить доступ к чатам Telegram, секретные чаты прочесть невозможно, зато можно инициировать любой чат от имени жертвы;
- «кустарность» протокола. Разработчики нарушили известный в криптографии принцип – не изобретать самостоятельно новых протоколов, если уже есть протоколы с доказанными оценками стойкости, решающие те же задачи;
- использование обычного «числового» протокола Диффи-Хеллмана. Протокол Диффи-Хеллмана на эллиптических кривых было бы предпочтительнее, так как для него достаточно более коротких случайных чисел. Это позволило бы снизить зависимость стойкости протокола от проблемы качества генерации случайных чисел, описанной в п. 3;
- способ генерации случайных величин существенно зависит от качества ГСЧ в ОС мобильного устройства;
- не обеспечивается безопасность метаданных. Недостаток является существенным, так как на Сервере можно отследить факт передачи сообщений. В дополнение к этому, любой пользователь может добавить к себе в адресную книгу любой номер телефона, который является Клиентом Telegram, и будет знать, когда этот Клиент находится в сети.

5 Достоинства и недостатки чатов WhatsApp

Анализ способа обработки сообщений в WhatsApp позволил выявить следующие его достоинства:

- обеспечение защиты ключей от чтения назад;
- обеспечение защиты ключей от чтения вперед;
- смена ключей после каждого сообщения;
- выработка общего секретного ключа с помощью протокола Диффи-Хеллмана на эллиптических кривых;
- использование протокола с доказанной оценкой стойкости.

В то же время можно заключить, что этому способу присущи следующие недостатки:

- отправка СМС на номер телефона, используемая для аутентификации, является недоверенным способом связи;
- синхронизация с ПК осуществляется через QR-код: такой прием дает множество возможностей для фишинговых атак (пример см. в п. 6);
-способ генерации случайных величин существенно зависит от качества ГСЧ в ОС мобильного устройства;
- не обеспечивается безопасность метаданных о пересылаемых сообщениях.

6 Возможные атаки на клиентов WhatsApp и Telegram

Несмотря на достаточно высокую стойкость протоколов, используемых в мессенджерах Telegram и WhatsApp, остается возможность осуществления целого ряда атак на эти протоколы.

UKS-атака в WhatsApp. Атака описана на основе материалов статьи [8].

A доверяет B, делится ключом. При этом A по факту делится ключом с E.

E — объект нападения, так как A думает, что пишет B, а на самом деле пишет E.

Схема атаки пошагово описана ниже.

Шаг 1. B запрашивает $eprepk_E$ с Сервера.

Шаг 2. B отправляет $eprepk_E$ и ipk_E на Сервер.

Шаг 3. Когда A хочет отправить сообщение B, A запрашивает $prepk_B$, Сервер возвращает $prepk_B$ и ipk_E.

Шаг 4. A вычисляет общий секрет, цепной и корневой ключи.

Шаг 5. A шифрует сообщение и отправляет B.

Шаг 6. B отправляет это сообщение E.

Шаг 7. E расшифровывает сообщение.

Данный вид атаки на сегодняшний момент можно предотвратить только сверкой хэш-кодов ключей.

Атака QRLJacking. Связана с возможностью синхронизации приложения WhatsApp с ПК. При сканировании из WhatsApp QR-кода с ПК все данные синхронизируются. Описание атаки основано на материалах [11].

Схема атаки пошагово описана ниже.

Шаг 1. Злоумышленник инициирует клиентскую QR-сессию и копирует код QR-логина на фишинговую страницу.

Шаг 2. Ссылка на страницу отправляется жертве.

Шаг 3. Жертва сканирует QR-код.

Шаг 4. Происходит процесс аутентификации.

Шаг 5. Злоумышленник получает доступ к аккаунту жертвы.

Атака IND-CCA: Неразличимость шифртекста. Свойство неразличимости шифртекста определяется тем, что противник не должен определить выбранное собеседником сообщение с вероятностью значительно большей, чем 1/2. Если противник может добиться успеха в различении выбранного шифртекста с вероятностью значительно большей, чем 1/2, то считается, что он имеет «преимущество» в различении шифртекста, а схема не считается безопасной с точки зрения неразличимости. Описание атаки основано на материалах исследовании [4].

A будет знать, какое из сообщений зашифровано. Свойство неразличимости не выполняется.

Схема атаки пошагово описана ниже.

Шаг 1. A отправляет B два различных сообщения m0 и m1 одинаковой длины.

Шаг 2. B случайным образом выбирает b = {0, 1} и шифрует сообщение $c = \mathrm{Enc}_{\mathrm{k}}(m_b)$, отправляет его A.

Шаг 3. A добавляет «лишний» блок 128 случайных битов c_r к c: $c' = c \| c_r$ и просит B расшифровать *c'*.

Шаг 4. B расшифровывает *c'* и считывает заполнение *payload*. Все, что идет ниже заполнения *payload*, в том числе и дополнение *padding*, отбрасывается (включая «лишний» блок). B передает $m' = \mathrm{Dec}_{\mathrm{k}}(c') = m_b$.

Шаг 5. A возвращает 1, если $m' = m_1$ и 0, если $m' = m_0$.

Так как длина дополнения *padding* в алгоритме расшифровки не проверяется, то в эту часть передаваемых сообщений можно добавить “лишние” блоки.

Атака, использующая уязвимость «принудительной» смены ключей в WhatsApp. В чатах WhatsApp в случае, когда получатель зашифрованного сообщения долгое время находится в статусе «не в сети» или сменил устройство, генерируется новый ключ. Сообщение пересылается на сервер WhatsApp, где будет ждать появления получателя в сети. То есть появляется посредник с еще одним ключом шифрования, к которому могут получить доступ разработчики. По утверждению [12], администрация мессенджера WhatsApp таким образом может предоставлять сообщения третьим лицам.

Заключение

Таким образом, основные результаты работы заключаются в следующем:

-проанализированы механизмы обеспечения безопасности сквозных защищенных соединений в мессенджерах WhatsApp и Telegram;

-проанализированы способы генерации случайных величин в устройствах на Android;

-выявлены достоинства, недостатки и вероятные атаки в Telegram и WhatsApp.

Результаты работы являются значимыми как в практическом, так и в научном плане, поскольку могут быть использованы для дальнейшего исследования защищенных соединений в мессенджерах с целью усовершенствования алгоритмов передачи информации между клиентами мессенджеров.

СПИСОКЛИТЕРАТУРЫ:

1 Telegram [Электронный ресурс]. URL: https://planfix.ru/docs/Telegram (Дата обращения: 13.11.2016).
2 Число пользователей Telegram превысило 100 миллионов [Электронный ресурс]. URL: https://lenta.ru/news/2016/02/23/telegram/ (Датаобращения: 13.11.2016).
3 Secretchats, end-to-endencryption [Электронныйресурс]. URL: https://core.telegram.org/api/end-to-end(Дата обращения: 13.11.2016).
4 Jacobsen, J. B. A practical cryptanalysis of the Telegram messaging protocol. Ph.D. Theses [Электронныйресурс] / J. B. Jacobsen; Aarhus University: Department of Computer Science. 2015. 79 pp. URL: https://cs.au.dk/~jakjak/master-thesis.pdf (Дата обращения: 20.02.2017).
5 Официальный сайт мессенджера WhatsApp [Электронный ресурс]. URL: https://https://www.whatsapp.com/ (Дата обращения: 27.09.2017).
6 WhatsAppEncryptionOverview [Электронныйресурс]. URL: https://www.whatsapp.com/security/WhatsApp-SecurityWhitepaper.pdf (Дата обращения: 30.11.2016).
7 Cohn-Gordon, K. A Formal Security Analysis of the Signal Messaging Protocol [Электронныйресурс] / K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, D. Stebila. 2016. 30 pp. URL: https://eprint.iacr.org/2016/1013.pdf (Датаобращения: 20.02.2017)
8 Frosch, T. How secure is TextSecure? [Электронныйресурс] / T. Frosch, C. Mainka, C. Bader, F. Bergsma, J. Schwenk, T. Holz. 2014.17 p. URL: https://eprint.iacr.org/2014/904.pdf (Датаобращения: 20.02.2017)
9 Krawczyk, H. Cryptographic Extraction and Key Derivation: The HKDF Scheme [Электронныйресурс] / H. Krawczyk. 2010. 34 pp. URL: https://eprint.iacr.org/2010/264.pdf (Датаобращения: 20.02.2017)
10 Saritas, S. Analysis of Android random number generator. Ph.D. Theses [Электронныйресурс] / S.Saritas, Bilkent University. 2013. 84 pp. URL: http://www.thesis.bilkent.edu.tr/0006566.pdf (Дата обращения: 20.02.2017)
11 Атака QRLJacking доказывает небезопасность авторизации с использованием SQRL [Электронный ресурс]. URL: https://xakep.ru/2016/08/02/qrljacking/ (Дата обращения: 21.12.2016).
12 В WhatsApp найдена уязвимость, позволяющая читать сообщения [Электронный ресурс].URL:https://xakep.ru/2017/01/13/whatsapp-retransmission-problem/ (Датаобращения: 15.01.2017).

REFERENCES:

[1] Telegram. Available at: https://planfix.ru/docs/Telegram (accessed 13.11.2016).
[2] Chislo pol'zovateley Telegram prevysilo 100 millionov (in Russian). Available at: https://lenta.ru/news/2016/02/23/telegram/ (accessed 13.11.2016)
[3] Secret chats, end-to-end encryption. Available at: https://core.telegram.org/api/end-to-end (accessed 13.11.2016)
[4] Jacobsen, J. B. A practical cryptanalysis of the Telegram messaging protocol. Ph.D. Theses / J. B. Jacobsen; Aarhus University: Department of Computer Science. 2015. 79 pp. Available at: https://cs.au.dk/~jakjak/master-thesis.pdf (accessed 20.02.2017).
[5] Official site of WhatsApp messenger. Available at: https://ru.wikipedia.org/wiki/WhatsApp (accessed 30.11.2016).
[6] WhatsApp Encryption Overview. Available at: https://www.whatsapp.com/security/WhatsAppSecurity-Whitepaper.pdf (accessed 30.11.2016).
[7] Cohn-Gordon, K. A Formal Security Analysis of the Signal Messaging Protocol / K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, D. Stebila. 2016. 30 pp. Available at: https://eprint.iacr.org/2016/1013.pdf (accessed 20.02.2017)
[8] Frosch, T. How secure is TextSecure? / T. Frosch, C. Mainka, C. Bader, F. Bergsma, J. Schwenk, T. Holz. 2014. 17 p. Available at: https://eprint.iacr.org/2014/904.pdf (accessed 20.02.2017)

[9] Krawczyk, H. Cryptographic Extraction and Key Derivation: The HKDF Scheme / H. Krawczyk. 2010. 34 pp. Available at: https://eprint.iacr.org/2010/264.pdf (accessed 20.02.2017)
[10] Saritas, S. Analysis of Android random number generator. Ph.D. Theses / S. Saritas, Bilkent University. 2013. 84 pp. Available at: http://www.thesis.bilkent.edu.tr/0006566.pdf (accessed 20.02.2017)
[11] Ataka QRLJacking dokazyvayet nebezopasnost' avtorizatsii s ispol'zovaniyem SQRL(in Russian). Available at: https://xakep.ru/2016/08/02/qrljacking/ (accessed 21.12.2016) (in Russian)
[12] V WhatsApp naydena uyazvimost', pozvolyayushchaya chitat' soobshcheniya(in Russian). Available at: https://xakep.ru/2017/01/13/whatsapp-retransmission-problem/ (accessed 15.01.2017)

Поступила в редакцию - 12 июня 2017 г. Окончательный вариант – 14 ноября 2017 г.
Received – June 12, 2017. The final version – November 14, 2017.