УДК 004.056

В.А. Кучер, М.М. Путято, А.С. Макарян, М.А. Карманов

Исследование безопасности пользовательских данных мессенджера Signal в операционной системе Android

Представлены анализ безопасности локально хранящихся данных конечного пользователя, а также особенности работы с ними в приложении signal операционный системы Android. В качестве тестовой версии приложения выступала актуальная на момент написания статьи сборка версии 5.3.12. По определенному сценарию были сгенерированы тестовые пользовательские данные в приложении, из которых были выделены источники информации с критичными данными. По имеющемуся в открытом доступе исходному коду были идентифицированы и проанализированы механизмы работы приложения, включая реализации мер защиты, с выявленными критичными данными. Произведена качественная оценка реализации механизмов защиты локально хранящихся критичных данных приложения по признаку наличия в приложении с типовыми мерами защиты, присущими для любых мобильных приложений, а также со специфичными конкретно для приложений класса мессенджеры. В результате были обнаружены недостатки, связанные с недоступностью определенных защитных механизмов на конкретных версиях операционной системы Android, которые могут повлечь компрометацию данных пользователя. Из преимуществ реализации защиты мессенджера был выделен модуль шифрования баз данных, обеспечивающий стойкую защиту от несанкционированного доступа к сведениям в них ввиду отсутствия определенной версии сборки для персональных компьютеров. В данной статье также предлагается подход к расшифровке баз данных мессенджера, при котором потребуется либо эмулятор устройства с операционной системой Android, либо реальное мобильное устройство в совокупности со специально разработанным приложением.

Ключевые слова: мобильные приложения, Android, статический анализ, динамический анализ, декомпиляция, обратная разработка, база данных, кибербезопасность, защита данных.
doi: 10.21293/1818-0442-2021-24-2-23-28

Системы мгновенного обмена текстовыми сообщениями для мобильных устройств с поддержкой голосовой и видеосвязи все больше и больше внедряются в жизнь человека. Мессенджеры становятся неотъемлемой частью не только повседневной жизни людей, но и часто решают личные, конфиденциальные, рабочие и деловые вопросы, от которых зависят благосостояние, безопасность и даже здоровье человека.

Целью данной работы является проведение исследования безопасности мессенджера Signal при наличии физического доступа к устройству для версии данного приложения: 5.3.12 (дата публикации – 01.2021 г.).

При постановке цели данной работы были определены следующие задачи:

1) определить перечень критичных данных приложения Signal, хранящихся локально на мобильном устройстве;

2) установить специфику работы с критичными данными в мобильном приложении;

3) определить перечень используемых мер защиты на основе определённого списка возможных механизмов защиты;

4) провести оценку защищенности информации приложения Signal.

Генерация и описание данных приложения

Для генерации данных в приложении были проведены тестовые сценарии работы, включающие в себя:

1) регистрацию в приложении;

2) работу с текстовыми сообщениями (отправка исходящих и прием входящих);

3) работу с вложениями сообщений;

4) выполнение аудио- и видеозвонков;

5) создание групповых чатов и выполнение схожих действий, как в случае со стандартными диалогами;

6) удаление определенных объектов из истории сообщений с фиксацией состояния критичных данных до изменения и после.

Для получения доступа к файлам приложения использовались устройства с учетными записями пользователя уровня администратор (root) [10].

В результате проведенных манипуляций были получены данные приложения Signal. Архитектура данных программы представлена в табл. 1.

Таблица 1

Архитектура данных программы

Имя объекта	Описание
db	Директория, содержащая базы данных приложения
f	Директория, содержащая файлы, создаваемые в ходе работы приложения
r	Директория, содержащая ресурсы программы или кэш
sp	Директория, содержащая конфигурационные файлы приложения
_manifest	Закодированный манифест приложения, содержащий все его разрешения

В ходе анализа были обнаружены следующие программные файлы, содержащие потенциально критичную информацию:

1) файл базы данных формата SQLite v3 signal.db, содержащий истории сообщений, данные

контактов из телефонной книги устройства, которые зарегистрированы в системе [2];

2) конфигурационный файл org.thoughtcrime.securesms_preferences.xml, который, по сравнению с прошлой рассмотренной версией мессенджера, дополнительно содержит секретные ключи для расшифровки медиа-вложений, базы данных, log-файлов;

3) конфигурационный файл SecureSMS-Preferences.xml, содержащий составные элементы, необходимые для построения мастер-ключа приложения;

4) файлы формата part-{timestamp}.mms, являющиеся зашифрованными файлами медиа-вложений;

5) журналы работы приложения (далее – log-файлы), включающие факты отправки сообщений (обезличены), однако в них имеется информация по отправленным вложениям. Также в файлах содержатся факты совершения звонков со сведениями об их типах (длительность звонка можно установить косвенно по содержанию log-файла). Количество генерируемых файлов по умолчанию не ограничено.

База данных (далее – БД) мессенджера signal.db недоступна для непосредственного чтения, так как она зашифрована [12]. В качестве провайдера криптографических операций базы данных выступает модуль SQLCipher компании Zetetic [15].

Для определения особенностей получения доступа к информации, содержащейся в рассматриваемой базе данных, необходимо изучить исходный код клиента приложения. Нет необходимости проводить обратную разработку путем декомпиляции установочного пакета приложения, так как исходный код приложения Signal для операционной системы Android (далее – ОС Android) находится в открытом доступе [12].

В ходе изучения исходного кода касательно получения доступа к БД было установлено следующее:

1) модуль SQLCipher имеет версию сборки 3.5.9;

2) используются параметры PRAGMA при открытии базы данных «cipher_default_kdf_iter = 1; cipher_default_page_size = 4096; kdf_iter = ‘1’; cipher_page_size = 4096;»;

3) секретный ключ подается в виде последовательности байт (decipher key \"x'{hex-последовательность секретного ключа}'").

Что касается самого секретного ключа, согласно алгоритму хранения приложения Signal, возможны 2 сценария:

1) если версия ОС Android меньше 6.0, то ключ хранится в открытом виде в файле org.thoughtcrime.securesms_preferences.xml в поле pref_database_unencrypted_secret уже в виде строчного представления шестнадцатеричных значений байт ключевой последовательности;

2) если версия ОС Android 6.0 и выше, то ключ хранится в зашифрованном виде, предварительно обрабатываемый средствами Android Keystore [3], и хранится в поле параметра pref_database_encrypted_secret.

Стоит отметить, что извлечь ключ шифрования базы данных не представится возможным только в случае, если защищенное хранилище Android Keystore реализовано аппаратно, во всех остальных случаях ключевую информацию можно получить и расшифровать БД.

Использование версий SQLCipher для стационарных решений (далее – ПК) 3.4.2 и 4.2.0 не дало результатов, и базу расшифровать не удалось с имеющимися данными. В качестве альтернативного способа была реализована тестовая утилита для ОС Android, которая использует идентичную используемой в целевом приложении версию модуля SQLCipher, расшифровывает БД и сохраняет ее копию. Приложение принимает на входе путь к зашифрованной базе, опциональные PRAGMA-параметры и ключ шифрования. В результате удалось получить расшифрованную БД signal_decrypted.db.

Стоит отметить, что информация в самой БД уже доступна для непосредственного чтения и хранится в открытом виде.

Log-файлы недоступны для непосредственного просмотра содержащихся в них данных ввиду использования криптографических преобразований информации. Согласно исходному коду приложения Signal, данные файлы используют симметричное шифрование AES в режиме CBC с ключом шифрования 256 бит. Данный ключ хранится в файле org.thoughtcrime.securesms_preferences.xml в поле параметра pref_log_unencrypted_secret или pref_log_encrypted_secret в зависимости от версии ОС Android конечного мобильного устройства аналогично сценариям при рассмотрении вопроса получения доступа к зашифрованной БД signal.db [12]. Стоит отметить, что информация в данных log-файлах шифруется не целым файлом, а построчно. Типовая структура зашифрованной строки log-файла представлена ниже:

1. 0x{71 59 49 D5 DE C1 31 C1 0D 44 61 D7 7F 8B A1 FA} – инициализирующий вектор (далее – IV) записи.
2. 0x{00 00 00 50} – длина зашифрованной последовательности строки log-файла (в данном примере длина зашифрованной последовательности равна 80 байтам, так как $50_{16} = 80_{10}$).
3. 0x{76 09 AA FF 89 03 AE 95 3D 1F B5 FF E2 95 93 79 0B 53 48 15 97 93 E5 06 2E A3 81 7D B2 FA 33 A2 78 98 60 67 4B B2 35 79 01 DD 5E 0B 5D 4C 64 63 9C 88 D3 F7 5C 1D C2 91 20 6B 56 A3 A5 53 31 43 66 F3 B7 0B A9 81 A5 1D BB 16 94 9C A5 5C 07 73} – зашифрованная информация log-файла (полезная часть).

Для автоматизации процесса расшифровки log-файлов была написана соответствующая программа для ПК, принимающая в качестве входных параметров директорию с зашифрованными log-файлами и их ключ шифрования.

Представленная выше запись в расшифрованном виде: «2020-09-13 12:24:05.231 GMT+03:00 I ApplicationContext: onCreate()».

Исследование данных после изменения в приложении

Далее необходимо проанализировать изменения состояний данных приложения до удаления объекта

из истории сообщений и после, используя графический интерфейс самого мессенджера. Под удаляемыми объектами будут пониматься следующие понятия:

1) текстовое сообщение в стандартном чате;
2) текстовое сообщение в групповом чате;
3) сообщение с вложением;
4) информация о вызове (аудио- и видеозвонок) через приложение.

Для качественной оценки изменений данных будут применяться следующие признаки:

1) соответствовала ли информация в графическом интерфейсе приложения новому состоянию;
2) соответствовала ли информация в хранилищах данных без побитового поиска новому состоянию;
3) соответствовала ли информация в хранилищах данных с побитовым поиском новому состоянию;
4) возможность восстановления удаленной информации из различных источников и максимальный срок между удалением и восстановлением [1];
5) возможность восстановления удаленной информации по метаданным приложения и максимальный срок между удалением и восстановлением [1].

Результаты анализа данных при изменении представлены в табл. 2.

Таблица 2

Результаты анализа данных при изменении

Наименование признака	Текстовое сообщение в диалоге	Текстовое сообщение в групповом чате	Удаление сообщения с вложением	Удаление информации о вызове
Соответствие в интерфейсе приложения	+	+	+	+
Соответствие в хранилище без расширенного поиска	+	+	+	+
Соответствие в хранилище с расширенным поиском	±	±	±	±
Возможность восстановления из различных источников	–	–	–	–
Возможность восстановления по метаданным	–	–	–	–

Стоит отметить, что при всех сценариях ручного удаления информация может быть восстановлена в исходном виде из-за организации структуры хранения информации в БД, где логические блоки файла с удаляемыми данными лишь помечаются как удаленные, которые в будущем могут быть перезаписаны другой информацией. Вследствие этого данные некоторое время спустя после удаления их пользователем через графический интерфейс приложения все еще будут доступны.

Анализ защищенности данных в приложении

Для оценки защищенности критичной информации в мессенджере будет использоваться следующий список возможных мер защиты:

1) аутентификация при открытии приложения (ПИН-код, графический ключ, отпечаток пальца, снимок лица и т.д.);
2) защищенное хранение контрольных данных аутентификации при входе в приложение (использование средств шифрования, хэш-функций) [4–6, 8];
3) защита содержимого на текущем экране приложения от несанкционированного просмотра вне контекста работы приложения на примере списка активных приложений, где представлены снимки экранов на момент сворачивания приложения [7] (далее – защита от несанкционированного просмотра);
4) ввод конфиденциальной информации реализован в виде собственного локального решения (к примеру, собственная виртуальная клавиатура) [7];
5) участие данных аутентификации в защите информации приложения;
6) наличие проверок в приложении повышенных привилегий пользователя (root – доступ) [7, 9, 13, 14];
7) ограничение копирования данных в файл резервной копии устройства [16];
8) устойчивость защиты данных приложения вследствие эксплуатации уязвимостей операционной системы;
9) восстановление измененных или удаленных данных на логическом уровне организации памяти;
10) восстановление измененных или удаленных данных на аппаратном уровне организации памяти;
11) механизм идентификации несанкционированной подмены данных в основных хранилищах критичных данных;
12) наличие мер усложнения обратной разработки приложения [4, 11];
13) наличие механизмов защиты у основных хранилищ критичной информации приложения [16];
14) наличие механизмов защиты у файлов, образующихся в ходе работы с приложением, с конфиденциальными данными [16];
15) наличие конфиденциальной информации в метаданных приложения [4, 5];
16) наличие конфиденциальных сведений в исходном коде приложения;
17) оптимальная организация хранения конфиденциальных данных приложения [5, 16];
18) наличие дополнительных факторов аутентификации при входе в учетную запись.

Результат оценки защищенности критичной информации в мессенджере по указанному выше списку представлен в табл. 3.

В качестве нестандартной реализации защиты стоит отметить использование шифрования базы данных с помощью SQLCipher, сборки которой нет для ПК, что исключает возможность получения доступа к данным только с помощью исследовательского стенда в виде ПК. Использование шифрования в log-файлах компенсирует факт наличия конфиденциальных данных в них.

Как итог можно сказать следующее: исследованное приложение можно рассматривать как достаточно защищенное решение для приложений класса мессенджер при условии использования их на устройстве с ОС Android с версией 6.0 и выше совместно с реализованным аппаратно защищенным хранилищем Android KeyStore. Выполнение данных условий обеспечит невозможность получения непосредственного доступа к критичным данным приложения.

Таблица 3

Результат оценки защищенности критичной информации в мессенджере

Механизм защиты	Наличие	Примечание
Аутентификация при открытии приложения	+	Опциональный ввод PIN-кода при открытии приложения
Защищенное хранение контрольных данных аутентификации	±	Использование криптографических средств при хранении PIN-кода для открытия приложения и ключей шифрования при версии ОС Android 6.0 и выше [8, 14]
Защита от несанкционированного просмотра	–	
Защищенный ввод конфиденциальной информации приложения [9]	–	
Участие данных аутентификации в защите информации приложения	+	Установленный опциональный PIN-код может использоваться как дополнительный составной элемент генерации ключей шифрования [12]
Наличие проверок в приложении повышенных привилегий пользователя	–	
Ограничение копирования данных в файл резервной копии устройства	+	
Устойчивость защиты при эксплуатации уязвимостей операционной системы	±	Устойчивость обеспечивается на устройствах с ОС Android с версии 7.0 и новее
Возможность восстановления измененных или удаленных данных на логическом уровне организации памяти [1]	±	Имеется возможность восстановить сообщения, удаленные пользователем вручную
Возможность восстановления измененных или удаленных данных на аппаратном уровне организации памяти	+	
Механизм идентификации несанкционированной подмены данных в основных хранилищах критичных данных	+	Для каждого сообщения в базе данных сохраняется его контрольная сумма
Наличие мер усложнения обратной разработки	–	Исходный код клиента находится в открытом доступе
Наличие механизмов защиты у основных хранилищ критичной информации приложения	+	БД шифруется с помощью SQLCipher
Наличие механизмов защиты у файлов, образующихся в ходе работы с приложением, с конфиденциальными данными	+	Медиа-файлы мессенджера шифруются по алгоритму симметричного шифрования AES с длиной ключа 256 бит
Наличие конфиденциальной информации в метаданных приложения	+	Факты отправки сообщений и совершения звонков, информация по медиа-вложениям
Оптимальная организация хранения конфиденциальных данных приложения	±	Хранение идентификатора учетной записи в Signal в открытом виде, хранение секретных ключей шифрования базы данных, log-файлов, медиа-вложений в открытом виде для устройств с версией ОС Android до 6.0
Наличие дополнительных факторов аутентификации при входе в учетную запись	+	Опциональная двухфакторная аутентификация при входе в учетную запись

Заключение

В работе рассмотрены сценарии получения выборки тестовых данных, методы получения этих данных из устройства, определены критичные данные для приложений типа мессенджер, хранящиеся локально на устройстве. Эти данные можно использовать для других приложений родственного класса, установлена специфика работы с критичными данными, определен перечень используемых мер защиты для приложения, а также проведена оценка защищенности информации приложения Signal.

Литература

1. Исследование остаточных артефактов Viber и Telegram в операционной системе Windows / А.И. Бородин, Р.Р. Вейнберг, Д.В. Писарев, О.В. Литвишко // Бизнес-информатика. – 2019. – Т. 13, № 4. – С. 39–48.

2. Безопасность – Поддержка Signal [Электронный ресурс]. – Режим доступа: https://support.signal.org/hc/ru/categories/360000674811-Безопасность, свободный (дата обращения: 30.09.2020).

3. Бюллетень по безопасности Android – август 2020 [Электронный ресурс]. – Режим доступа: https://source.android.com/security/bulletin/2020-08-01, свободный (дата обращения: 03.10.2020).

4. Путято М.М. Кибербезопасность как неотъемлемый атрибут многоуровнего защищенного киберпространства / М.М. Путято, А.С. Макарян // Прикаспийский журнал: управление и высокие технологии. – 2020. – № 3. – С. 94–102.

5. Путято М.М. Классификация мессенджеров на основе анализа уровня безопасности хранимых данных / М.М. Путято, А.С. Макарян // Прикаспийский журнал: управление и высокие технологии. – 2019. – № 4. – С. 135–143.

6. Исследование системы идентификации и подтверждения легитимности доступа на основе динамиче-

ских методов биометрической аутентификации / М.М. Путято, А.С. Макарян, Ш.М. Чич, В.К. Маркова // Прикаспийский журнал: управление и высокие технологии. – 2020. – № 3. – С. 83–93.

7. Уязвимости и угрозы мобильных приложений, 2019 [Электронный ресурс]. – Режим доступа: https://www.ptsecurity.com/ru-ru/research/analytics/mobile-application-security-threats-and-vulnerabilities-2019/#id6, свободный (дата обращения: 27.09.2020).

8. App security best practices, 2020 [Электронный ресурс]. – Режим доступа: //developer.android.com/topic/security/best-practices, свободный (дата обращения: 03.10.2020).

9. NISTIR 8144 (Draft) Assessing Threats to Mobile Devices & Infrastructure. / C. Brown, S. Dog, J. Franklin, N. McNab, S. Voss-Northrop, M. Peck, B. Stidham. – The Mobile Threat Catalogue, 2016. – 44 p.

10. The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps / N.D.W. Cahyani, N.H. Ab Rahman, W.B. Glisson, K.-K.R. Choo // Mobile Networks and Applications. – 2017. – Vol. 22, No. 2. – P. 240–254. DOI: 10.1007/ s11036-016-07919.

11. Github – rehmanmuradali/android-security-guides: Is Your App Ready To Get Live? [Электронный ресурс]. – Режим доступа: https://github.com/rehmanmuradali/android-security-guides, свободный (дата обращения: 02.10.2020).

12. Github – signalapp / Signal-Android: A private messenger for Android [Электронный ресурс]. – Режим доступа: https://github.com/signalapp/Signal-Android, свободный (дата обращения: 30.09.2020).

13. Mueller B. OWASP Mobile Security Testing Guide / B. Mueller, S. Schleier, J. Willemsen. – 2020. – 536 p.

14. Schleier S. OWASP Mobile Application Security Verification Standard, Mobile application security check guide standard / S. Schleier, J. Willemsen, C. Holguera. – 2020. – 49 p.

15. SQLCipher – Zetetic [Электронный ресурс]. – Режим доступа: https://www.zetetic.net/sqlcipher/, свободный (дата обращения: 03.10.2020).

16. Telegram, Signal, Wickr Me: выбираем самый безопасный мессенджер и разбираемся, существует ли он [Электронный ресурс]. – Режим доступа: https://habr.com/en/company/group-ib/blog/522178/, свободный (дата обращения: 20.12.2020).

Кучер Виктор Алексеевич
Канд. техн. наук, профессор каф.
компьютерных технологий
и информационной безопасности
Кубанского государственного технологического
университета (КубГТУ)
Московская ул., 2, г. Краснодар, Россия, 350072
Тел.: +7-905-402-64-98
Эл. почта: vakucher@bk.ru

Путято Михаил Михайлович
Канд. техн. наук, доцент каф.
компьютерных технологий
и информационной безопасности КубГТУ
Московская ул., 2, г. Краснодар, Россия, 350072
Тел.: +7-964-904-05-55
Эл. почта: putyato.m@gmail.com

Макарян Александр Самвелович
Канд. техн. наук, доцент каф.
компьютерных технологий
и информационной безопасности КубГТУ
Московская ул., 2, г. Краснодар, Россия, 350072
Тел.: +7-918-444-64-47
Эл. почта: msanya@yandex.ru

Карманов Михаил Александрович
Аспирант каф. компьютерных технологий
и информационной безопасности КубГТУ
Московская ул., 2, г. Краснодар, Россия, 350072
Тел.: +7-905-470-63-74
Эл. почта: michaelkdev15@gmail.com

Kucher V.A., Putyato M.M., Makaryan A.S., Karmanov M.A.

Investigation of User Data Security for Android-based «Signal» Messenger

The article presents the security analysis of locally stored end-user data, as well as the specifics of working with them in the application called “Signal” based on Android OS. The investigated version 5.3.12 was the most recent one up to the time of writing this article. According to a certain scenario, test user data was generated in the application, and then the sources information with critical data was extracted from this data. Using the open-source code available, the mechanisms of the application's operation, including the implementation of protection measures, with the specified critical data were identified and analyzed. A qualitative assessment of implementing protection mechanisms for locally stored critical data was made to distinguish the data with typical protection measures inherent for any mobile applications, and the one specific for applications of this particular class of messengers. As a result, the flaws were discovered related to the inaccessibility of certain protective mechanisms on specific versions of the Android operating system, which could compromise the user data. As an advantage of the messenger protection, the database encryption module could be specified, which provides strong protection against unauthorized access to information due to the lack of a specific version of the assembly for personal computers. This article also proposes an approach to decrypt messenger databases, which requires either an emulator of an Android-based device, or a real mobile device with a specially developed application.

Keywords: mobile apps, Android, static analysis, dynamic analysis, decompilation, reverse engineering, database, cybersecurity, data protection.
doi: 10.21293/1818-0442-2021-24-2-23-28

References

1. Borodin A.I., Veynberg R.R., Pisarev D.V., Litvishko O.V. [Simulation of artifact detection in Viber and Telegram instant messengers in Windows operations systems]. *Business informatics*, 2019, vol. 13, no. 4, pp. 39–48 (in Russ.).

2. Security – Signal Support. Available at: https://support.signal.org/hc/en/categories/360000674811-Security (Accessed: September 30, 2020) (in Russ.).

3. Android Security Bulletin – august 2020. Available at: https://source.android.com/security/bulletin/2020-08-01 (Accessed: October 3, 2020) (in Russ.).

4. Putyato M.M., Makaryan A.S. [Cybersecurity as an integral attribute of a multilevel protected cyberspace]. *Caspian journal: management and high technologies*. 2020, no. 3, pp. 94–102 (in Russ.).

5. Putyato M.M, Makaryan A.S. [Classification of messengers based on analysis of the security level of stored data]. *Caspian journal: management and high technologies*. 2020, no. 3, pp. 94–112 (in Russ.).

6. Putyato M.M., Makaryan A.S., Chich S.M., Markova V.K. [Investigation of the system of identification and confirmation of the legitimacy of access based on dynamic methods of biometric authentication]. *Caspian journal: management and high technologies*. 2020, no. 3, pp. 83–93 (in Russ.).

7. Uyazvimosti i ugrozy mobil'nykh prilozhenii [Weak points and threats of mobile applications], 2019. Available at: https://www.ptsecurity.com/ru-ru/research/analytics/mobile-application-security-threats-and-vulnerabilities-2019/#id6 (Accessed: September 27, 2020 (in Russ.).

8. App security best practices, 2020. Available at: https://developer.android.com/topic/security/best-practices (Accessed: October 3, 2020).

9. Brown C., Dog S., Franklin J., McNab N., Voss-Northrop S., Peck M., Stidham B. NISTIR 8144 (Draft) Assessing Threats to Mobile Devices & Infrastructure. *The Mobile Threat Catalogue*. 2016. 44 p.

10. Cahyani N.D.W., Ab Rahman N.H., Glisson W.B., Choo K.-K.R. [The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps]. *Mobile Networks and Applications*. 2017. vol. 22, no. 2, pp. 240–254. doi: 10.1007/ s11036-016-07919.

11. Github – rehmanmuradali/android-security-guides: Is Your App Ready To Get Live? Available at: https://github.com/rehmanmuradali/android-security-guides (Accessed: October 2, 2020).

12. Github – signalapp / Signal-Android: A private messenger for Android. Available at: https://github.com/signalapp/Signal-Android (Accessed: September 30, 2020).

13. Mueller B., Schleier S., Willemsen J. *OWASP Mobile Security Testing Guide*, 2020, 536 p.

14. Schleier S., Willemsen J., Holguera C. *OWASP Mobile Application Security Verification Standard, Mobile application security check guide standard*, 2020, 49 p.

15. SQLCipher – Zetetic. Available at: https://www.zetetic.net/sqlcipher/ (Accessed: October 3, 2020).

16. Telegram, Signal, Wickr Me: vybiraem samyi bezopasnyi messendzher i razbiraemsya , sushcestvuet li on. [Choosing the safest messenger, if ever it exists] (in Russ.) Available at: https://habr.com/en/company/group-ib/blog/522178/ (Accessed: December 20, 2020).

Victor A. Kucher
Candidate of Science in Engineering, Professor,
Department of Computer Technologies and Information Security, Kuban State Technological University (KubSTU)
2, Moskovskaya st., Krasnodar, Russia, 350072
Phone: +7-905-402-64-98
Email: vakucher@bk.ru

Mikhail M. Putyato
Candidate of Science in Engineering, Assistant Professor,
Department of Computer Technologies and Information Security, KubSTU
2, Moskovskaya st., Krasnodar, Russia, 350072
Phone: +7-964-904-05-55
Email: putyato.m@gmail.com

Alexander S. Makaryan
Candidate of Science in Engineering, Assistant Professor,
Department of Computer Technologies and Information Security, KubSTU
2, Moskovskaya st., Krasnodar, Russia, 350072
Phone: +7-918-444-64-47
Email: msanya@yandex.ru

Mikhail A. Karmanov
Postgraduate student, Department of Computer Technologies and Information Security, KubSTU
2, Moskovskaya st., Krasnodar, Russia, 350072
Phone: +7-964-904-05-55
Email: michaelkdev15@gmail.com