УДК 51-3:519.6

Д. С. ЧУЙКО, А. С. ГУБКА

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ШИФРОВАНИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ ХРАНИЛИЩ

В статье рассматриваются вопросы выявления и анализа угроз информационной безопасности при использовании мобильных устройств, а также возможные пути их решения. Производится анализ криптографических методов с использованием ключа, а именно симметричная и ассиметричная методологии, приводятся численные примеры работы криптографических алгоритмов. Описываются механизмы распределения ключей, а также детально рассматривается реализация такого механизм с помощью алгоритма Диффи-Хеллмана, используемая в реализуемой криптосистеме. Предлагается реализация комплекса мер по обеспечению информационной безопасности при использовании мобильных устройств.

Ключевые слова: *мобильная операционная система, криптография, криптосистема, симметричная/ассиметричная методология, криптографический ключ, алгоритм RSA, схема Эль-Гамаля, алгоритм RC4.*

Введение

За последние годы использование мобильных устройств набирает всё большую популярность. Мобильные устройства неуклонно захватывают рынок как персонального, так и корпоративного использования. Переход пользователей и бизнеса на мобильные устройства практически неизбежен. Так, по некоторым оценкам, количество мобильных устройств, используемых в корпоративной среде, непрерывно увеличивается, и на 2015 год почти половина используемых в бизнесе устройств относится к мобильным.

Помимо этого, количество вредоносного кода и выявляемых уязвимостей в мобильном коде также растет год от года практически в геометрической прогрессии. Одним из самых распространенных способов внедрения вредоносного кода на мобильное устройство является установка приложений из недостоверных источников.

Наиболее распространенными на сегодняшний день мобильными операционными системами являются iOS (от компании Apple) и Android (от компании Google). Если говорить об устройствах, работающих под управлением iOS, где jailbreak является наиболее популярным способом обхода правил установки приложений только через Apple AppStore, и устройства на платформе Android, где получение root-прав занимает всего пару минут, то проблема внедрения вредоносного кода на мобильное устройство является актуальной на сегодняшний день. И, как следствие, угроза информационной безопасности также непрерывно растет.

Под информационной безопасностью обычно понимают состояние защищенности обрабатываемых, хранимых и передаваемых данных в информационно-телекоммуникационных системах от незаконного ознакомления, преобразования и уничтожения, а также состояние защищённости информационных ресурсов от воздействий, направленных на нарушение их работоспособности [1].

Многочисленные публикации последних лет показывают, что злоупотребления информацией, циркулирующей в информационных системах или передаваемой по каналам связи, совершенствовались не менее интенсивно, чем меры защиты от них. В работах [2, 3] рассматриваются современные способы обеспечения информационной безопасности, согласно которым для обеспечения защиты информации требуется не просто разработка частных механизмов защиты, а реализация системного подхода, включающего комплекс взаимосвязанных мер (использование специальных технических и программных средств, организационных мероприятий, нормативно-правовых актов, и т.д.). Комплексный характер защиты проистекает из комплексных действий злоумышленников, стремящихся любыми средствами добыть важную для них информацию.

Таким образом, можно выделить следующие проблемы информационной безопасности при использовании мобильных устройств [4]:

– нарушение конфиденциальности в результате кражи устройства;

– нарушение конфиденциальности информации в результате доступа посторонних лиц к устройству, оставленному без присмотра;

– доступ к конфиденциальной информации внешними нарушителями посредством использования вредоносного программного кода;

– нарушение конфиденциальности информации в процессе ее передачи по внешним каналам связи (с использованием личной почты, облачных хранилищ, социальных сетей).

Существуют различные методы, предназначенные для защиты конфиденциальной информации от несанкционированного доступа в процессе ее передачи и хранения [5]:

– скрыть канал передачи информации, используя нестандартный способ передачи сообщений;

– замаскировать канал передачи закрытой информации в открытом канале связи, например, спрятав информацию в безобидном «контейнере» с использованием тех или других стенографических способов либо обмениваясь открытыми сообщениями, смысл которых согласован заранее;

– существенно затруднить возможность перехвата противником передаваемых сообщений, используя специальные методы передачи по широкополосным каналам сигнала под уровнем шумов либо с использованием «прыгающих» несущих частот и т.п.;

– использовать криптографическое преобразование.

В отличие от перечисленных методов криптография не «прячет» передаваемые сообщения, а преобразует их в форму, недоступную для понимания противником.

Целью данной статьи является описание реализации криптографической системы для защиты конфиденциальной информации пользователей мобильных устройств от несанкционированного доступа как при ее хранении на устройстве, так и в процессе ее передачи с использованием облачных сервисов.

Постановка задачи исследования

Рассмотрев проблемы информационной безопасности при использовании мобильных устройств, необходимо разработать приложение, представляющее реализацию комплекса мер по обеспечению защиты конфиденциальной информации пользователей. При решении данной проблемы необходимо рассмотреть следующие задачи:

– защиту текстовой информации, которую пользователь может ввести с помощью клавиатуры или из соответствующего файла;

– защиту пользовательских файлов, независимо от содержания этих файлов;

– организацию канала передачи данных: ключей шифрования и дешифрования, а также самой зашифрованной информации.

Для решения вышеуказанных задач необходимо рассмотреть различные криптографические алгоритмы с учетом скорости их работы, а также требований к аппаратным ресурсам. В данной работе рассматриваются криптографические алгоритмы с использованием ключа, так как они соответствуют современным требованиям стойкости, а также позволяют организовывать конфиденциальный обмен данными среди большой или изменяющейся группы пользователей. В случае, если пользователь выходит из группы, то оставшимся участникам группы нет необходимости менять алгоритм шифрования, а достаточно всего лишь заменить ключ.

Анализ криптографических методов

Безопасность современных алгоритмов шифрования полностью основана на ключах, а не на деталях алгоритмов. Это означает, что алгоритм может быть опубликован и проанализирован. Продукты, использующие некоторый алгоритм, могут широко тиражироваться. Не имеет значения, что злоумышленнику известен алгоритм, если ему не известен конкретный ключ, то он не сможет получить доступ к конфиденциальной информации. В зависимости от типов и количества ключей выделяют симметричную и ассиметричную методологию [6].

Основная проблема современных алгоритмов состоит в том, чтобы сгенерировать и безопасно передать ключи участникам взаимодействия. Также другой важной проблемой является аутентификация:

– сообщение шифруется лицом, которое владеет ключом в данный момент. Это лицо может быть законным владельцем ключа, но если система скомпрометирована, это может быть и злоумышленник;

– когда участники взаимодействия получают ключи, они не могут знать, что эти ключи были сгенерированы и посланы уполномоченным лицом.

В качестве алгоритмов, рассматриваемых в данной статье, были выбраны алгоритмы Диффи-Хелмана, RSA, RC4 и схема Эль-Гамаля.

Механизм распределения ключей

В обеих вышеуказанных методологиях необходимо решать проблему распространения ключей.

В симметричных методологиях эта проблема стоит наиболее остро, поэтому в них ясно определяется, как передавать ключи между участниками взаимодействия до начала взаимодействия. Кон-

кретный способ выполнения этого зависит от требуемого уровня безопасности. Если не требуется высокий уровень безопасности, то ключи можно рассылать с помощью некоторого механизма доставки. Для обеспечения более высокого уровня безопасности более уместна ручная доставка ключей ответственными за это людьми.

Асимметричные методологии пытаются обойти эту проблему с помощью шифрования симметричного ключа и присоединения его в таком виде к зашифрованным данным. А для распространения открытых асимметричных ключей используются центры сертификации ключей Certification Authority.

Сторонники асимметричных систем считают, что такого механизма достаточно для обеспечения аутентичности абонентов взаимодействия. Но проблема все равно остается. Пара асимметричных ключей должна создаваться совместно. Оба ключа, независимо от того, доступны они всем или нет, должны быть безопасно посланы владельцу ключа, а также центру сертификации ключей. Единственный способ сделать это - использовать какой-либо способ доставки при невысоких требованиях к уровню безопасности, и доставлять их вручную - при высоких требованиях к безопасности.

Проблема с распространением ключей в асимметричных системах состоит в следующем [7]:

– не существует стандартов для решения проблемы распространения ключей;

– электронная подпись ключей центром сертификации не всегда гарантирует их аутентичность;

– нет надежного способа проверить, между какими компьютерами осуществляется взаимодействие. Существуют атаки, при которых атакующий маскируется под СА и получает данные, передаваемые в ходе взаимодействия. Для этого атакующему достаточно перехватить запрос к центру сертификации ключей и подменить его ключи своими.

Алгоритм Диффи-Хеллмана

Одним из возможных решений проблемы распределения ключей является использование алгоритма Диффи-Хеллмана, схема работы которого представлен на рис. 1. Данный алгоритм основан на трудности вычисления дискретного логарифма [8].

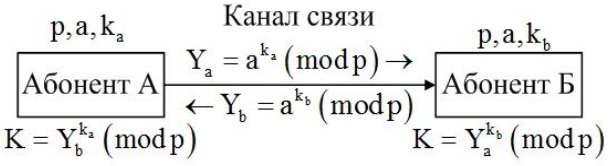

Рис. 1. Схема обмена ключами при использовании алгоритма Диффи-Хеллмана

Сторона А выбирает случайное число k_a, а сторона В – случайное число k_b таким образом, чтобы выполнялись условия $1 < k_a < p-1$ и $1 < k_b < p-1$. Числа k_a и k_b держатся в секрете. Сторона А формирует открытый ключ $Y_a = a^{k_a} \pmod p$, а сторона В аналогично формирует ключ $Y_b = a^{k_b} \pmod p$. После обмена ключами Y_a и Y_b стороны вычисляют значение секретного ключа K:

$$K = Y_a^{k_b} \pmod p = a^{k_a k_b} \pmod p,$$

$$K = Y_a^{k_b} \pmod p = a^{k_a k_b} \pmod p.$$

Таким образом, полученный ключ K является секретным, поскольку решение уравнений Y_a и Y_b для больших чисел не представляется возможным.

Пример. Пусть $p = 13$, $a = 7$, $k_a = 3$, $k_b = 4$.

Открытый ключ, посылаемый стороной А:

$$Y_a = 7^3 \pmod{13} = 343 \pmod{13} = 5.$$

Открытый ключ, посылаемый стороной Б:

$$Y_b = 7^4 \pmod{13} = 2401 \pmod{13} = 9.$$

Секретное число, вычисляемое обеими сторонами:

$$K = 5^4 \pmod{13} = 625 \pmod{13} = 1,$$

$$K = 9^3 \pmod{13} = 729 \pmod{13} = 1.$$

Таким образом, из вышеуказанного примера видно, что в случае, если переписка абонентов А и Б будет перехвачена третьей стороной, то на основании значений p и a у нее не будет никакой возможности определить значение секретных ключей абонентов А и Б, а также значения ключа K.

Симметричная методология

В данной методологии для шифрования и дешифрования применяется один и тот же криптографический ключ. Выделяют две основные группы симметричных криптосистем [9]:

– блочные шифры, обрабатывают информацию: блоками определённой длины (обычно 64, 128 бит и более), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми раундами. Результатом повторения раундов является лавинный эффект – нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных;

– поточные шифры, шифрование проводится: над каждым битом либо байтом исходного (откры-

того) текста с использованием гаммирования. Поточный шифр может быть легко создан на основе блочного шифра, запущенного в специальном режиме.

Блочные шифры являются основой, на которой реализованы практически все криптосистемы. Методика создания цепочек из зашифрованных блочными алгоритмами байтов позволяет шифровать ими пакеты информации неограниченной длины. Такое свойство блочных шифров, как быстрота работы, используется асимметричными криптографическими алгоритмами, медлительными по своей природе. Отсутствие статистической корреляции между битами выходного потока блочного шифра используется для вычисления контрольных сумм пакетов данных и в хешировании паролей.

Симметричные криптосистемы обладают следующими достоинствами, по сравнению с ассиметричными криптосистемами [10]:

– скорость;

– простота реализации (за счёт более простых операций);

– меньшая требуемая длина ключа для сопоставимой стойкости;

– изученность (за счёт большего возраста).

А также следующими недостатками:

– сложность управления ключами в большой сети. Означает квадратичное возрастание числа пар ключей, которые надо генерировать, передавать, хранить и уничтожать в сети;

– сложность обмена ключами. Для применения необходимо решить проблему надёжной передачи ключей каждому абоненту, так как нужен секретный канал для передачи каждого ключа обеим сторонам.

Ассиметричная методология

Симметричные криптосистемы, несмотря на множество преимуществ, обладают серьезным недостатком. Связан он с ситуацией, когда общение между собой производят большое количество людей. В этом случае для каждой пары, переписывающейся между собой, необходимо создавать свой секретный симметричный ключ. Это в итоге приводит к существованию в системе из N пользователей $\frac{N^2}{2}$ ключей. Кроме того, при нарушении конфиденциальности какой-либо рабочей станции, злоумышленник получает доступ ко всем ключам этого пользователя и может отправлять сообщения от его имени всем абонентам.

Своеобразным решением этой проблемы явилось появление асимметричной криптографии. Основная идея асимметричных криптографических алгоритмов состоит в том, что ключи для шифрования и расшифровки разные, хотя и создаются вместе. Один ключ делается известным всем, а другой держится в тайне. Хотя можно шифровать и дешифровать обоими ключами, данные, зашифрованные одним ключом, могут быть расшифрованы только другим ключом.

Процедура шифрования выбрана так, что она необратима даже по известному ключу шифрования. То есть, зная ключ шифрования и шифрованный текст, невозможно восстановить исходное сообщение – прочесть его можно только с помощью второго ключа – ключа дешифрования. А раз так, то ключ шифрования для отправки писем какому-либо лицу можно вообще не скрывать – зная его, все равно невозможно прочесть зашифрованное сообщение.

Данный подход представлен на рис. 2, где e и d – ключи шифрования и дешифрования, E – функция шифрования для произвольного ключа e, D – функция дешифрования для произвольного ключа d, а m и c исходное и шифрованное сообщение соответственно.

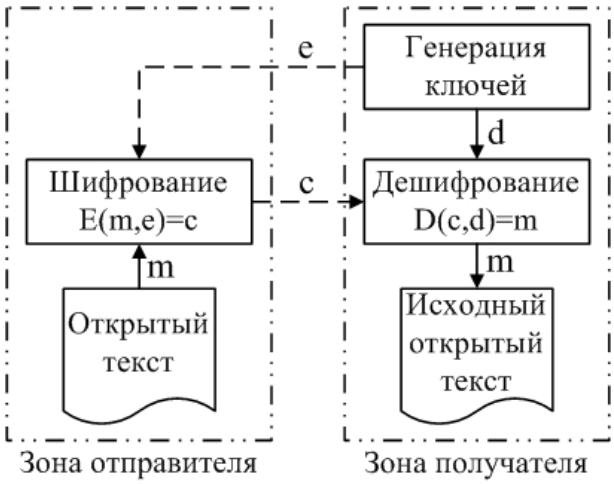

Рис. 2. Принцип работы ассиметричных систем шифрования

Процедура шифрования выбрана так, что она необратима даже по известному ключу шифрования – это одно из необходимых условий асимметричной криптографии. Таким образом, зная ключ шифрования и шифрованный текст, невозможно восстановить исходное сообщение – прочесть его можно только с помощью второго ключа – ключа дешифрования. А раз так, то ключ шифрования для отправки писем какому-либо лицу можно вообще не скрывать.

Все асимметричные криптосистемы являются объектом атак путем прямого перебора ключей, поэтому в них должны использоваться гораздо более длинные ключи, чем те, которые используются в

симметричных, для обеспечения эквивалентного уровня защиты. Это сразу же сказывается на вычислительных ресурсах, требуемых для шифрования и дешифрования. В таблице 1 приведена сравнительная характеристика ключей из двух вышеупомянутых методологий.

Таблица 1
Сравнительная характеристика размерности ключей различных методологий

Размерность симметричного ключа	Размерность ассиметричного ключа
56 бит	384 бит
64 бит	512 бит
80 бит	768 бит
112 бит	1762 бита
128 бит	2304 бита

Асимметричные криптосистемы обладают следующими достоинствами, по сравнению с симметричными криптосистемами [11]:

– в симметричной криптографии ключ держится в секрете для обеих сторон, а в асимметричной криптосистеме только один секретный;

– при симметричном шифровании необходимо обновлять ключ после каждого факта передачи, тогда как в асимметричных криптосистемах пару (E, D) можно не менять значительное время;

– отсутствие необходимости предварительной передачи секретного ключа по надёжному каналу;

– число ключей в асимметричной криптосистеме значительно меньше, чем в симметричной криптосистеме такого же размера.

Алгоритм RSA

Алгоритм RSA был разработан в 1977 году Роном Ривестом, Ади Шамиром и Леном Адлеманом и опубликован в 1978 году. Стойкость RSA базируется на сложности факторизации больших целых чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел разрядностью 100…200 десятичных цифр и даже больше. Восстановление открытого текста по шифрованному тексту и открытому ключу равносильно разложению числа на два больших простых множителя. Многие годы алгоритм RSA противостоит многочисленным попыткам криптографического вскрытия. Криптоанализ ни доказывает, ни опровергает безопасность алгоритма RSA, тем самым обосновывая степень доверия к алгоритму [12].

Пример. Дано сообщение $M = 15$ и $p = 11$, $q = 5$. Вычисляем $n = p \cdot q = 11 \cdot 5 = 55$. Определяем функцию Эйлера $\varphi(55) = (11-1)(5-1) = 40$. Выбираем ключ для шифрования $e = 7$, который удовлетворяет условиям, $7 < 40$ и $НОД(7, 40) = 1$. Вычисляем ключ дешифрования d из уравнения $7d = 1 \pmod{40}$ с помощью алгоритма Евклида:

$$\begin{cases} 40 = 7 \cdot 5 + 5, \\ 7 = 5 \cdot 1 + 2, \\ 5 = 2 \cdot 2 + 1, \\ 2 = 1 \cdot 2 + 0. \end{cases}$$

Обратная подстановка дает:

$$1 = 5 - 2 \cdot 2 = 5 - (7 - 5 \cdot 1) \cdot 2 = 5 \cdot 3 + 7(-2) =$$
$$= (40 - 7 \cdot 5) \cdot 3 + 7(-2) = 40 \cdot 3 + 7(-17).$$

Таким образом, $d = 23$, так как $-17 = 23 \pmod{40}$.

В результате открытый ключ представляет собой пару чисел $(55, 7)$, секретный ключ – $(55, 23)$.

Шифрование сообщения:

$$C = 15^7 \pmod{55} = 5.$$

Дешифрование сообщения:

$$M = 5^{23} \pmod{55} = 15.$$

Таким образом, из вышерассмотренного примера видно, что для выполнения операции шифрования и дешифрования применяются трудоемкие математические операции, однако данное обстоятельство компенсируется высокой стойкостью к криптоанализу.

Схема Эль-Гамаля

Схема Эль-Гамаля была предложена Тахером Эль-Гамалем в 1984 году. Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации.

Криптостойкость схемы Эль-Гамаля основана на том, что можно легко вычислить степень целого числа, то есть произвести умножение его самого на себя любое число раз также, как и при операциях с обычными числами. Однако трудно найти показатель степени, в которую нужно возвести заданное число, чтобы получить другое, тоже заданное. В общем случае эта задача дискретного логарифмирования кажется более трудной, чем разложение больших чисел на простые сомножители, на основании чего можно предположить, что сложности вскрытия систем RSA и Эль-Гамаля будут сходными [13].

Пример. Дано сообщение $M = 5$, $p = 11$, $g = 2$.

Выбирается случайное целое число $x = 8$, удовлетворяющее условию $1 < x < p - 1$. Далее вычисляется $y = 2^8 \bmod p = 3$. Таким образом, открытый ключ представляет собой $(11,2,3)$, а закрытый ключ – $(11,8)$.

Шифрование сообщения:

$$a = 2^9 \bmod 11 = 512 \bmod 11 = 6,$$

$$b = 3^9 \cdot 5 \bmod 11 = 19683 \cdot 5 \bmod 11 = 9.$$

В итоге, шифрованное сообщение представляет собой пару чисел $(6,9)$.

Дешифрование сообщения:

$$M = 9\left(6^8\right)^{-1} \bmod 11 = 5.$$

Таким образом, из вышерассмотренного примера видно, что в результате шифрования исходного текста ($M = 5$), полученный зашифрованный текст, представляет собой пару чисел. Из этого следует, что в результате шифрования объем зашифрованных данных будет увеличиваться в два раза по сравнению с исходными данными, однако этот недостаток компенсируется высокой криптостойкостью.

Алгоритм RC4

RC4, алгоритм потокового шифрования, был предложен в 1987г. Рональдом Линном Ривестом, известным американским специалистом в области криптографии. С 1994г. он нашел широкое применение в целом ряде криптографических приложений, включая такие известные, как SSL и TLS – для шифрования данных, передаваемых по сетям передачи данных, не предусматривающим защиты пользовательских данных, WPA и WEP – для защиты беспроводных соединений.

Таким широким распространением алгоритм обязан ряду свойств, не утративших актуальности за время своего существования. Одно из них – высокое быстродействие. Помимо, этого данный алгоритм имеет высокую экономность используемых вычислительных ресурсов, а, следовательно, имеет низкое энергопотребление, что является ключевой характеристикой при использовании на мобильных устройствах [14].

Алгоритм RC4 (рис. 3) строится, как и любой потоковый шифр, на основе параметризованного ключом генератора псевдослучайных битов с равномерным распределением. Длина ключа может составлять от 40 до 256 бит.

Ядро алгоритма состоит из функции генерации ключевого потока. Эта функция генерирует последовательность битов k_i, которая затем объединяется с открытым текстом m_i посредством суммирования по модулю два, в результате получается шифрованное сообщение c_i:

$$c_i = m_i \oplus k_i.$$

Расшифровка заключается в регенерации этого ключевого потока k_i и сложении его и шифрограммы c_i по модулю два. В силу свойств суммирования по модулю два, на выходе мы получим исходный незашифрованный текст m_i:

$$m_i = c_i \oplus k_i = \left(m_i \oplus k_i\right) \oplus k_i.$$

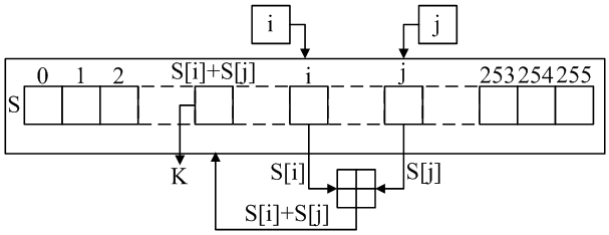

Рис.3. Схема работы алгоритма RC4

Результаты

В результате проведенного анализа основных угроз информационной безопасности при использовании мобильных устройств было принято решение о создании четырех режимов работы приложения:

– робота с текстом, введенным пользователем (шифрование и дешифрование);

– работа с пользовательскими файлами (шифрование и дешифрование).

В соответствии с этим проводился анализ криптографических методов и алгоритмов для их применимости при организации информационной безопасности с учетом затрат аппаратных и временных ресурсов.

Для реализации были выбраны следующие алгоритмы:

– алгоритм RSA;

– схема Эль-Гамаля;

– алгоритм RC4.

Для определения скорости работы реализованных алгоритмов была составлена программа на языке программирования Java. В качестве исходных данных использовался текстовый файл, содержащий в себе 10^6 случайно сгенерированных символов, и произвольный файл размеров 10 Мб, который также состоял из случайной последовательности байтов. Тестирование производилось на устройстве под управлением мобильной операционной системы Android 4.2.2, обладающим следующими аппаратными характеристиками:

– 2-х ядерный процессор MediaTek MT8125/ 8389 (1,2 ГГц);

– 1 Гб оперативной памяти;

– 16 Гб встроенной памяти.

В таблице 2 представлена скорость работы разработанных алгоритмов в режиме шифрования при различной сложности ключа, а в таблице 3 – скорость работы в режиме дешифрования.

На основании данных из таблицы 2 были построены графики, отражающие скорость работы алгоритмов в режиме шифрования в зависимости от сложности ключа и типа входных данных. На рис. 4 представлен график отражающий скорость шифрования текстовых данных, а на рис.5 – скорость шифрования файлов. На основании данных из таблицы 3 также были составлены графики, которые отражают скорость работы алгоритмов, но в режиме дешифрования. На рис. 6 представлен график отражающий скорость дешифрования текстовой информации, на рис. 7 – скорость дешифрования файлов.

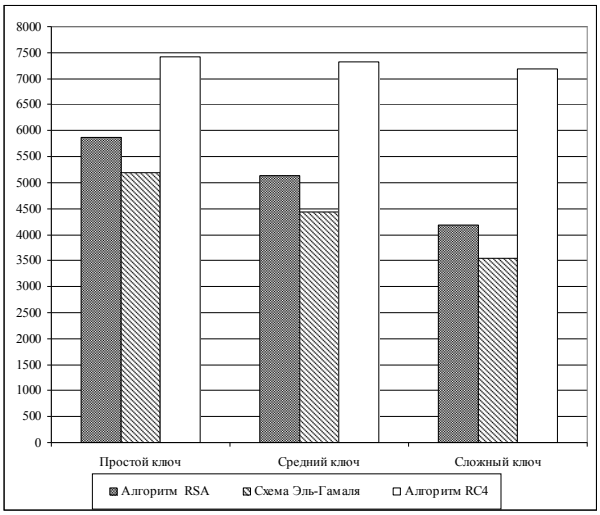

Рис. 5. Скорость работы приложения при шифровании файлов

Таблица 2

Скорость работы алгоритмов в режиме шифрования

	Ключ	Алгоритм RSA	Схема Эль-Гамаля	Алгоритм RC4
Текст	Простой	5179,43 символов/с	4588,45 символов/с	6542,06 символов/с
	Средний	4571,70 символов/с	3941,44 символов/с	6515,20 символов/с
	Сложный	3714,81 символов/с	3149,17 символов/с	6378,83 символов/с
Файл	Простой	5867,11 байт/с	5197,67 байт/с	7410,67 байт/с
	Средний	5141,24 байт/с	4432,46 байт/с	7326,87 байт/с
	Сложный	4181,04 байт/с	3544,40 байт/с	7179,41 байт/с

Таблица 3

Скорость работы алгоритмов в режиме дешифрования

	Ключ	Алгоритм RSA	Схема Эль-Гамаля	Алгоритм RC4
Текст	Простой	4911,04 символов/с	3739,80 символов/с	6577,77 символов/с
	Средний	3893,52 символов/с	3113,56 символов/с	6441,52 символов/с
	Сложный	1787,85 символов/с	2450,89 символов/с	6340,92 символов/с
Файл	Простой	5408,69 байт/с	4118,76 байт/с	7244,32 байт/с
	Средний	4293,64 байт/с	3433,52 байт/с	7103,49 байт/с
	Сложный	1977,48 байт/с	2710,84 байт/с	7013,48 байт/с

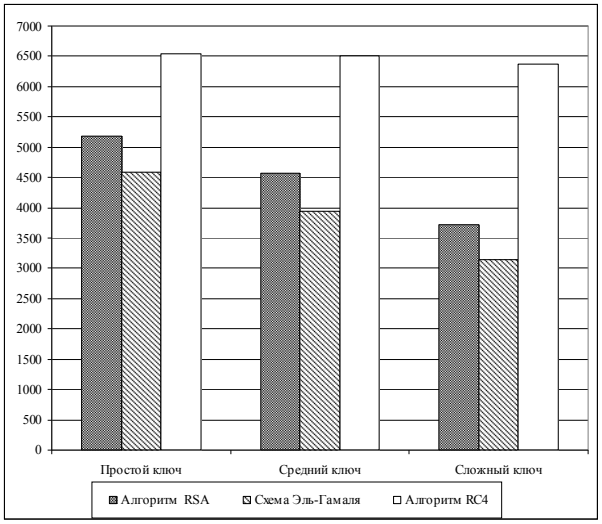

Рис. 4. Скорость работы приложения при шифровании текстовой информации

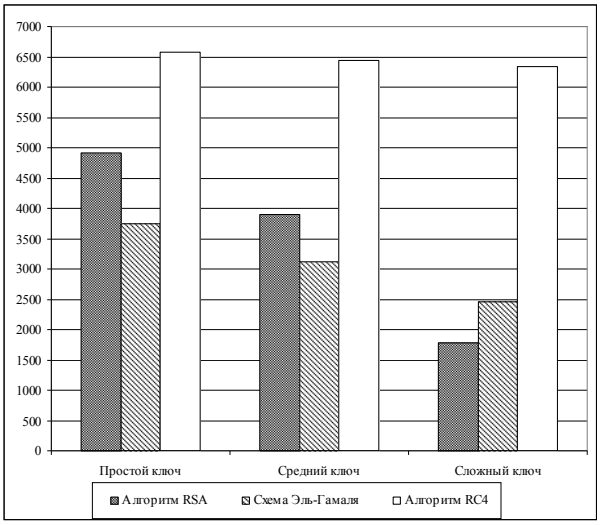

Рис. 6. Скорость работы приложения при дешифровании текстовой информации

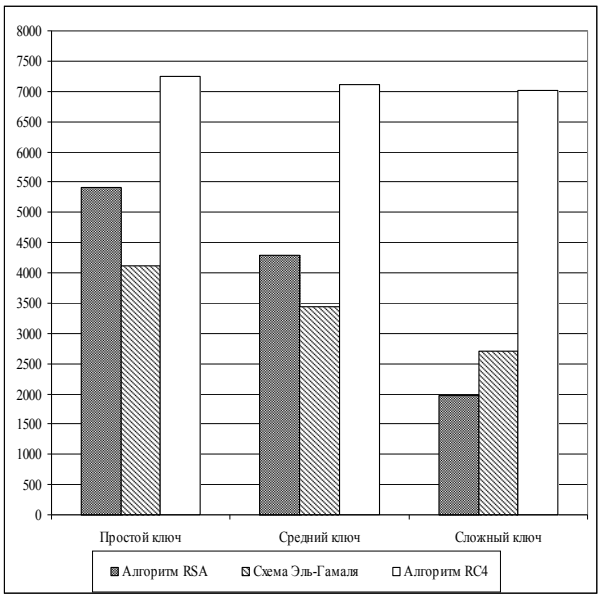

Рис. 7. Скорость работы приложения при дешифровании файлов

Из таблиц 2 и 3 видно, что при работе с текстовыми данными и пользовательскими файлами наибольшую скорость имеет алгоритм RC4, следующим по скорости является RSA и наименьшую скорость показывает схема Эль-Гамаля. Следует также заметить, что при увеличении сложности ключа скорость работы алгоритма RSA и схемы Эль-Гамаля значительно уменьшается как в режиме шифрования, так и в режиме дешифрования, в то время как скорость работы алгоритма RC4 в обоих режимах изменяется незначительно при увеличении сложности ключа.

Основываясь на результатах анализа, было принято решение об использовании алгоритмов RSA, Эль-Гамаля и RC4 для защиты текстовых данных. А для защиты пользовательских файлов наиболее оптимальным является алгоритм RC4, поскольку файлы зашифрованные данным алгоритмом имеют незначительный прирост в объеме занимаемого пространства на диске. Несмотря на то, что алгоритмы RSA и Эль-Гамаля показали достаточно высокую скорость работы при шифровании и дешифровании файлов, они обладают существенным недостатком, а именно значительным приростом объема зашифрованного файла по сравнению с исходным. Поэтому рекомендуется применять данные алгоритмы при шифровании файлов, имеющих незначительный размер.

Для алгоритмов RSA и Эль-Гамаля используются ключи размерностью 512, 1024 и 2048 бит, а для алгоритма RC4 – 40, 128 и 256 бит (простой, средний и сложный соответственно).

На рис. 8 приведена схема работы приложения, отражающая последовательность работы с ним.

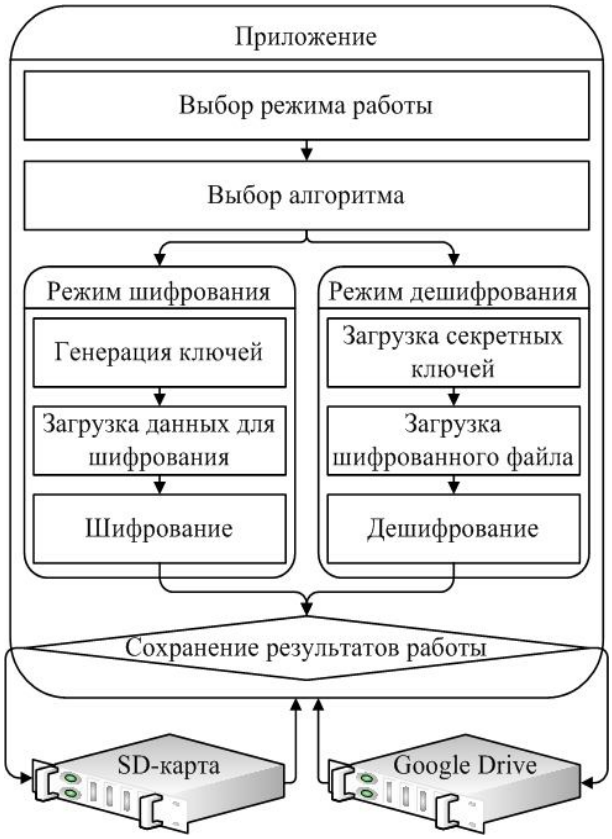

Рис. 8. Схема работы приложения

Заключение

В данной работе рассмотрены вопросы обеспечения информационной безопасности при использовании мобильных устройств, приведено сравнение симметричной и ассиметричной методологий шифрования, выявлены их преимущества и недостатки. Рассмотрены способы распределения ключей шифрования.

Предложен комплекс мер по обеспечению информационной безопасности при использовании мобильных устройств. Рассмотрены результаты тестирования производительности реализованных алгоритмов в режиме шифрования и дешифрования при различной сложности ключа и входных данных.

Применение предложенного комплекса позволяет гарантировать сохранность конфиденциальности пользовательских данных в течение длительного периода времени.

Литература

1. Барсуков, В. С. Безопасность: технологии, средства и услуги [Текст] / В. С. Барсуков. – М. : КУДИЦ-ОБРАЗ, 2001. – 496 с.

2. Bishop, M. What is computer security? [Text] / M. Bishop // IEEE Security Practice. – 2009.

– Vol. 1, Iss. 1. – P. 67-69.

3. Bellare, M. Introduction to modern cryptography [Text] / M. Bellare, P. Rogaway // IEEE Transactions on Information Theory. – 2008. – Vol. 2, Iss. 6. – P. 283-286.

4. Neidhardt, E. Asymmetric Cryptography for Mobile Devices [Text] / E. Neidhardt. – Service-centric Networking, 2011. – P. 1-12.

5. Основы информационной безопасности [Текст] : учеб. для вузов / Е. Б. Белов [и др.] ; под ред. А. А. Шелупанов. – М. : Горячая линия – Телеком, 2006. – 544 с.

6. Каханович, Г. Ф. Компьютерная криптография. Теория и практика [Текст] / Г. Ф. Каханович, А. Ю. Пузыренко. – К. : МК-Пресс, 2006. – 288 с.

7. Самсонов, Б. Б. Теория информации и кодирования [Текст] / Б. Б. Самсонов. – М. : Феникс, 2002. – 288 с.

8. Аграновский, А. В. Практическая криптография. Алгоритмы и их программирование [Текст] / А. В. Аграновский, Р. А. Хади. – М. : Солон-Пресс, 2002. – 256 с.

9. Асосков, А. В. Поточные шифры [Текст] / А. В. Асосков, М. А. Иванов, А. А. Мирский. – М. : КУДИЦ-ОБРАЗ, 2003. – 336 с.

10. Schneier, B. Applied Cryptography. Protocols, algorithms, source code in C language [Text] / B. Schneier. – M. : Triumph, 2003. – 816 p.

11. Schneier, B. Practical Cryptography [Text] / B. Schneier, N. Ferguson. – M. : John Wiley & Sons, 2006. – 432 p.

12. Онацкий, А. В. Ассиметричные методы шифрования [Текст] : учеб. пособие / А. В. Онацкий, Л. Г. Йона ; под ред. Н. В. Захарченко. – Одесса : ОНАС им. А. С. Попов. – 2010. – 148 с.

13. Simmons, G. J. Symmetric and asymmetric encryption [Text] / G. J. Simmons // ACM Computing Surveys. – 2008. – Vol. 11, Iss. 4. – P. 305-330.

14. Нечаев, В. И. Элементы криптографии [Текст] : учеб. пособие / В. И. Нечаев ; под ред. В. А. Садовничьего. – М. : Высш. шк., 2007. – 142 с.

Поступила в редакцию 5.10.2015, рассмотрена на редколлегии 18.11.2015

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ШИФРУВАННЯ ДАНИХ З ВИКОРИСТАННЯМ ХМАРНИХ СХОВИЩ

Д. С. Чуйко, А. С. Губка

В даній статті розглядаються питання виявлення та аналізу загроз інформаційної безпеки при використанні мобільних пристроїв, а також можливі шляхи їх вирішення. Проводиться аналіз криптографічних методів з використанням ключа, а саме симетрична та асиметрична методології, наводяться чисельні приклади роботи криптографічних алгоритмів. Описуються механізми розподілу ключів, а також детально розглядається реалізація такого механізму з використанням алгоритму Діффі-Хеллмана, яка була використана в проектованій криптосистемі. Пропонується реалізація комплексу заходів щодо забезпечення інформаційної безпеки при використанні мобільних пристроїв.

Ключові слова: мобільна операційна система, криптографія, криптосистема, симетрична/асиметрична методологія, криптографічний ключ, алгоритм RSA, схема Ель-Гамаля, алгоритм RC4.

MOBILE APPLICATIONS DEVELOPMENT FOR DATA ENCRYPTION USING CLOUD STORAGE

D. S. Chuico, A. S. Gubka

Identification and analysis issues of information security threats using mobile devices are considered in the article as well as solutions for its solving. The analysis of cryptographic methods using keys is reviewed, specifically symmetric and asymmetric methodology, also examples of cryptographic algorithm is shown. Key distribution mechanisms are described, besides this the implementation of such a mechanism in details is considered using the Diffie-Hellman algorithm, which is used in the realized cryptosystem. The implementation of measures set to ensure information security using mobile devices is proposed.

Key words: mobile operating system, cryptography, cryptosystem, symmetric / asymmetric methodology, cryptographic key, RSA algorithm, El-Gamal algorithm, RC4 algorithm.

Чуйко Дмитрий Сергеевич – магистр кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Губка Алексей Сергеевич – канд. техн. наук, доцент, доцент кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.