УДК 004.032.2

А.С. ДАЦЬКО, магистр НТУ «ХПИ»
Л.В. ДЕРБУНОВИЧ, д-р техн. наук, проф. НТУ «ХПИ»
П.А. КАЧАНОВ, д-р техн. наук, зав. кафедры НТУ «ХПИ»

МЕТОДЫ ПОВЫШЕНИЯ КАЧЕСТВА АЛГОРИТМОВ ШИФРОВАНИЯ ДАННЫХ С ОТКРЫТЫМ КЛЮЧОМ

У даній статті розглянуті методи підвищення якості шифрування даних. Запропоновані різні варіанти підвищення кількості повторень алгоритмів в обидва напрямки. Також запропоновано такі методи: використання різних алгоритмів і одного ключа, використання одного алгоритму і різних ключів а також комбінація цих методів.

This article describes methods of improving the quality of encryption. Suggested various options for increasing the number of repetitions of algorithms in both directions. Also offered the following methods: the use of different algorithms and a key, use one algorithm and the various options and combinations of these methods.

Постановка проблемы. Существует множество способов объединять блочные алгоритмы для получения новых алгоритмов. Стимулом создавать подобные схемы является желание повысить безопасность, не пробираясь через тернии создания нового алгоритма.

Анализ литературы. В [1] приведены основы и предпосылки шифрования информации. В работе [2] описаны основы шифрования информации с использованием односторонних обработки данных. В [3] приведен криптографический анализ практически самого первого и самого распространенного ассиметричного алгоритма шифрования – RSA. В работе [4] приведены примеры программной реализации различных алгоритмов на языке высокого уровня С. Работа [5] описывает блочные шифры.

Цель статьи – найти методы повышения качества шифрования существующих криптографических алгоритмов.

Многократное шифрование. Одним из способов объединения является многократное шифрование – для шифрования одного и того же блока открытого текста алгоритм шифрования используется несколько раз с несколькими ключами. Шифрование каскадом похоже на многократное шифрование, но использует различные алгоритмы. Существуют и другие методы. Повторное шифрование блока открытого текста одним и тем же ключом с помощью того же или другого алгоритма неразумно. Повторное использование того же алгоритма не увеличивает сложность вскрытия грубой силой. (Не забывайте, мы предполагаем, что алгоритм, включая количество шифрований, известен криптоаналитику.) При различных алгоритмах сложность вскрытия грубой силой может возрастать, а может и

остаться неизменной. Чтобы использовать методы, описанные в этой статье, следует убедиться, что ключи для последовательных шифрований различны и независимы.

Двойное шифрование. Наивным способом повысить безопасность алгоритма является шифрование блока дважды с двумя различными ключами. Сначала блок шифруется первым ключом, а затем получившийся шифротекст шифруется вторым ключом. Дешифрирование является обратным процессом.

Если блочный алгоритм образует группу, то всегда существует K_3. Если алгоритм не образует группу, то при помощи исчерпывающего поиска взломать получающийся дважды зашифрованный блок шифротекста намного сложнее. Вместо 2^n (где n – длина ключа в битах), потребуется 2^{2n} попыток. Если алгоритм использует 64-битовый ключ, для обнаружения ключей, которыми дважды зашифрован шифротекст, потребуется 2^{128} попыток.

Но при вскрытии с известным открытым текстом это не так. Меркл и Хеллман придумали способ обменять память на время, который позволяет вскрыть такую схему двойного шифрования за 2^{n+1} шифрований, а не за 2^{2n}. (Они использовали эту схему против *DES*, но результаты можно обобщить на все блочные алгоритмы.) Это вскрытие называется «встреча посередине»**,** с одной стороны выполняется шифрование а с другой – дешифрирование, получившиеся посередине результаты сравниваются.

Для такого вскрытия нужен большой объем памяти: 2^n блоков. Для 56-битового ключа нужно хранить 2^{56} 64-битовых блоков, или 10^{17} байтов. Такой объем памяти пока еще трудно себе представить, но этого хватает, чтобы убедить самых параноидальных криптографов в том, что двойным шифрованием пользоваться не стоит.

При 128-битовом ключе для хранения промежуточных результатов потребуется 10^{39} байтов. Если предположить, что есть способ хранить бит информации, используя единственный атом алюминия устройство памяти, нужное для выполнения такого вскрытия, будет представлять собой алюминиевый куб с ребром, длиной 1 км. Кроме того, вам понадобится куда-то его поставить. Вскрытие «встреча посередине» кажется невозможным для ключей такого размера.

Утверждается, что «у этого режима нет никаких особых достоинств», к тому же он, по видимому, так же чувствителен ко вскрытию «встреча посередине» как и другие режимы двойного шифрования.

Тройное шифрование с двумя ключами. В более интересном методе, предложенном Тачменом, блок обрабатывается три раза с помощью двух ключей: первым ключом, вторым ключом и снова первым ключом. Он предлагает, чтобы отправитель сначала шифровал первым ключом, затем дешифрировал вторым, и окончательно

шифровал первым ключом. Получатель расшифровывает первым ключом, затем шифрует вторым и, наконец, дешифрирует первым.

Иногда такой режим называют шифрование-дешифрирование-шифрование (*encrypt-decrypt-encrypt, EDE*). Если блочный алгоритм использует *n*-битовый ключ, то длина ключа описанной схемы составляет *2n* бит. Любопытный вариант схемы шифрование-дешифрирование-шифрование был разработан в *IBM* для совместимости с существующими реализациями алгоритма: задание двух одинаковых ключей эквивалентно одинарному шифрованию этим ключом.

Тройное шифрование с двумя ключами устойчиво, но Меркл и Хеллман разработали другой способ размена памяти на время, который позволяет взломать этот метод шифрования за 2^{n-1} действий, используя 2^n блоков памяти.

Понадобится 2^n времени и памяти, а также 2^m выбранных открытых текстов. Вскрытие не очень практично, но все же чувствительность к нему является слабостью алгоритма.

Тройное шифрование с тремя ключами. Если вы собираетесь использовать тройное шифрование, я рекомендую три различных ключа. Общая длина ключа больше, но хранение ключа обычно не является проблемой. Для наилучшего вскрытия с разменом памяти на время, которым является «встреча посередине», потребуется 2^{2n} действий и 2^n блоков памяти. Тройное шифрование с тремя независимыми ключами безопасно настолько, насколько на первый взгляд кажется безопасным двойное шифрование.

Тройное шифрование с минимальным ключом (TEMK). Существует безопасный способ использовать тройное шифрование с двумя ключами, противостоящий описанному вскрытию и называемый Тройным шифрованием с минимальным ключом (*Triple Encryption with Minimum Ke*y, TEMK). Фокус в том, чтобы получить три ключа из X_1 и X_2. T_1, T_2 и T_3 представляют собой константы, которые необязательно хранить в секрете. Эта схема гарантирует, что для любой конкретной пары ключей наилучшим будет вскрытие с известным открытым текстом.

Удвоение длины блока. В академическом сообществе давно спорят на тему, достаточна ли 64-битовая длина блока. С одной стороны 64-битовый блок обеспечивает диффузию открытого текста только в 8 байтах шифротекста. С другой стороны более длинный блок затрудняет безопасную маскировку структуры, кроме того, больше возможностей ошибиться. Существуют предложения удваивать длину блока алгоритма с помощью многократного. Прежде, чем реализовывать одно из них, оцените возможность вскрытия «встреча посередине». Схема Ричарда Аутбриджа не более безопасна, чем тройное шифрование с одинарным блоком и двумя ключами. Однако не рекомендуется использовать подобный прием. Он не быстрее

обычного тройного шифрования: для шифрования двух блоков данных все также нужно шесть шифрований. Характеристики обычного тройного шифрования известны, а за новыми конструкциями часто прячутся новые проблемы.

Отбеливание. Отбеливанием называется способ, при котором выполняется *XOR* части ключа с входом блочного алгоритма и *XOR* другой части ключа с выходом блочного алгоритма. Впервые этот метод был применен для варианта *DESX*, а затем (по-видимому, независимо) в *Khufu* и *Khafre*.

Смысл этих действий в том, чтобы помешать криптоаналитику получить пару «открытый текст/шифротекст» для лежащего в основе блочного алгоритма. Метод заставляет криптоаналитика угадывать не только ключ алгоритма, но и одно из значений отбеливания. Так как *XOR* выполняется и перед, и после блочного алгоритма, считается, что этот метод устойчив против вскрытия «встреча посередине».

Многократное последовательное использование блочных алгоритмов. Этот прием, иногда называемый последовательным использованием (каскадирование), можно распространить и на большее количество алгоритмов и ключей.

Пессимисты утверждали, что совместное использование двух алгоритмов не гарантирует повышения без опасности. Алгоритмы могут взаимодействовать каким-то хитрым способом, что на самом деле даже уменьшит. Даже тройное шифрование тремя различными алгоритмами может не быть настолько безопасным, насколько вам это кажется. Криптография – достаточно темное искусство, если вы не совсем понимаете, что делаете, то можете легко попасть в беду.

Действительность намного светлее. Упомянутые предостережения верны, только если различные ключи зависят друг от друга. Если все используемые ключи независимы, то сложность взлома последовательности алгоритмов по крайней мере не меньше, чем сложность взлома первого из применяемых алгоритмов. Если второй алгоритм чувствителен к вскрытию с выбранным открытым текстом, то первый алгоритм может облегчить это вскрытие и при последовательном использовании сделать второй алгоритм чувствительным к вскрытию с известным открытым текстом. Такое возможное облегчение вскрытия не ограничивается только алгоритмами шифрования: если вы позволите кому-то другому определить любой из алгоритмов, делающих что-то с вашим сообщением до шифрования, стоит удостовериться, что ваше шифрование устойчиво по отношению к вскрытию с выбранным открытым текстом.

Это можно сформулировать и иначе: При использовании вскрытия с выбранным открытым текстом последовательность шифров взломать не легче, чем любой из шифров последовательности. Ряд

результатов показал, что последовательное шифрование взломать по крайней мере не легче, чем самый сильный из шифров последовательности, но в основе этих результатов лежат некоторые несформулированные предположения. Только если алгоритмы коммутативны, как в случае каскадных потоковых шифров, надежность их последовательности не меньше, чем у сильнейшего из используемых алгоритмов.

Не стоит забывать, что ключи для каждого алгоритма последовательности должны быть независимыми. Если алгоритм А использует 64-битовый ключ, а алгоритм В - 128-битовый ключ, то получившаяся последовательность должна использовать 192-битовый ключ. При использовании зависимых ключей у пессимистов гораздо больше шансов оказаться правыми.

Выводы. Таким образом, в статье рассмотрены и проанализированы основные методы шифрования данных. По предварительным оценкам, наиболее оптимальными выбраны следующие криптографические алгоритмы: *RSA* и методы шифрования на основе эллиптических кривых над конечными полями, но данные нуждаются в дальнейшей экспериментальной проверке путем программной реализации на языке программирования высокого уровня и измерения скорости шифрования одинаковых пакетов данных различной длины и сложности.

Список литературы: **1.** *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В* Основы криптографии. **2.** *Саломаа А.* Криптография с открытым ключом: Пер. с англ. – М.: Мир, 1995. – 318 с., ил. **3.** *Ян С. Й.* Криптоанализ RSA. – М.-Ижевск: НИЦ «Регулярная и хаотическая динамика», Ижевский институт компьютерных исследований, 2011. – 312 с. **5.** *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

Поступила в редакцию 31.01.2011