УДК 004.032.2

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

АНАЛИЗ АЛГОРИТМОВ ШИФРОВАНИЯ ДАННЫХ С ОТКРЫТЫМ КЛЮЧОМ

Розглянуто основи шифрування даних з відкритим ключем. Проаналізовано найбільш популярні алгоритми та визначено їх переваги і недоліки. Відібрані найбільш оптимальні та універсальні алгоритми для подальшого дослідження шляхом програмної реалізації на мові високого рівня та проведення низки експериментів з потоками даних різної довжини та складності.

A base data encryption with a public key. Analyzed the most popular algorithms and defines their advantages and disadvantages. Selected the most optimal and universal algorithms for further investigation by the software implementation of high-level language and a series of experiments with data streams of varying length and complexity.

Постановка проблемы. Задача защиты информации от искажения и от несанкционированного доступа давно является одной из самых основных при передаче любых данных. На данный момент существует достаточно большое количество алгоритмов позволяющих защитить информацию, но практически не существует достаточно полноценной картины их сравнения и как следствие достаточно трудно правильно подобрать алгоритм для каждой новой задачи.

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

Цель статьи – проанализировать существующие алгоритмы шифрования данных с открытым ключом.

Основы криптографии с открытым ключом. Идея криптографии с открытым ключом очень тесно связана с идеей односторонних функций, то есть таких функций $f(x)$, что по известному x довольно просто найти значение $f(x)$, тогда как определение x из $f(x)$ сложно в смысле теории. Но сама односторонняя функция бесполезна в применении: ею можно зашифровать сообщение, но расшифровать нельзя. Поэтому криптография с открытым ключом использует односторонние функции с лазейкой.

Лазейка – это некий секрет, который помогает расшифровать. То есть существует такой y, что зная $f(x)$ и y, можно вычислить x. К примеру, если разобрать часы на множество составных частей, то очень сложно собрать вновь работающие часы. Но если есть инструкция по сборке (лазейка), то можно легко решить эту проблему.

Основные принципы построения криптосистем с открытым ключом. Начинаем с трудной задачи P. Она должна решаться сложно в смысле теории: не должно быть алгоритма, с помощью которого можно было бы перебрать все варианты решения задачи P за полиномиальное время относительно размера задачи. Более правильно сказать: не должно быть известного полиномиального алгоритма, решающего данную задачу — так как ни для одной задачи еще пока не доказано, что для нее подходящего алгоритма нет в принципе. Можно выделить легкую подзадачу P' из P. Она должна решаться за полиномиальное время, лучше, если за линейное. «Перетасовываем и взбалтываем» P', чтобы получить задачу P'', совершенно не похожую на первоначальную. Задача P'', по крайней мере, должна выглядеть как оригинальная труднорешаемая задача P. P'' открывается с описанием, как она может быть использована в роли ключа зашифрования. Как из P'' получить P', держится в секрете как секретная лазейка. Криптосистема организована так, что алгоритмы расшифрования для легального пользователя и криптоаналитика существенно различны. В то время как второй решает P'' задачу, первый использует секретную лазейку и решает P' задачу.

Большинство криптосистем с открытым ключом основаны на проблеме факторизации больших чисел. К примеру, *RSA* использует в качестве открытого ключа n произведение двух больших чисел. Сложность взлома такого алгоритма состоит в трудности разложения числа n на множители. Но эту задачу решить реально. И с каждым годом процесс разложения становится все быстрее. Также задачу разложения потенциально можно решить с помощью Алгоритма Шора при использовании достаточно мощного квантового компьютера. Для многих методов несимметричного шифрования криптостойкость, полученная в результате криптоанализа, существенно отличается от величин, заявляемых разработчиками алгоритмов на основании теоретических оценок. Поэтому во многих странах вопрос применения алгоритмов шифрования данных находится в поле законодательного регулирования.

RSA. Это буквенная аббревиатура от фамилий *Rivest, Shamir* и *Adleman*. *RSA* – криптографический алгоритм с открытым ключом. *RSA* стал первым алгоритмом такого типа, пригодным и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений. Безопасность *RSA* основана на трудности разложения на множители больших чисел. Открытый и

закрытый ключи являются функциями двух больших (100 – 200 разрядов или даже больше) простых чисел. Предполагается, что восстановление открытого текста по шифротексту и открытому ключу эквивалентно разложению на множители двух больших чисел. Для генерации двух ключей используются два больших случайных простых числа p и q. Для максимальной безопасности выбирайте p и q равной длины. Рассчитывается произведение (1).

$$n = p * q. \quad (1)$$

Затем случайным образом выбирается ключ шифрования e, такой что e и (p–1)(q–1) являются взаимно простыми числами. Наконец расширенный алгоритм Эвклида используется для вычисления ключа дешифрирования d, такого что:

$$ed = 1 \ (mod \ (p–1) \ (q–1)). \quad (2)$$

Другими словами:

$$d = e^{-1} \ mod \ ((p–1) \ (q–1)) \ . \quad (3)$$

Заметим, что d и n также взаимно простые числа. Числа e и n – это открытый ключ, а число d – закрытый. Два простых числа p и q больше не нужны. Они должны быть отброшены, но не должны быть раскрыты. Для шифрования сообщения m оно сначала разбивается на цифровые блоки, меньшие n (для двоичных данных выбирается самая большая степень числа 2, меньшая n). То есть, если p и q – 100-разрядные простые числа, то n будет содержать около 200 разрядов, и каждый блок сообщения m_i должен быть около 200 разрядов в длину. Если нужно зашифровать фиксированное число блоков, их можно дополнить несколькими нулями слева, чтобы гарантировать, что блоки всегда будут меньше n. Зашифрованное сообщение c будет состоять из блоков c_i той же самой длины. Формула шифрования выглядит так:

$$c_i = m_i^e \ mod \ n \quad (4)$$

Для расшифровки сообщения берется каждый зашифрованный блок c_i и вычисляется (5).

$$m_i = c_i^d \ mod \ n \quad (5)$$

Так как

$$c_i^d = (m_i^e)^d = m_i^{ed} = m_i^{k(p-1)(q-1)+1} = m_i m_i^{k(p-1)(q-1)} = m_i * 1 = m_i \quad (6)$$

Формула (6) восстанавливает сообщение.

Схема Рабина-Вильямса. Безопасность схемы Рабина (Rabin) опирается на сложность поиска квадратных корней по модулю составного числа. Эта проблема аналогична разложению на множители. Вот одна из реализаций этой схемы.

Сначала выбираются два простых числа p и q, конгруэнтных 3 mod 4. Эти простые числа являются закрытым ключом, а их произведение (1) – открытым ключом. Для шифрования сообщения M (M должно быть меньше n), просто вычисляется

$$C = M^2 \ mod \ n. \quad (7)$$

Дешифрирование сообщения также несложно, но немного скучнее. Так как получатель знает p и q, он может решить две конгруэнтности с помощью китайской теоремы об остатках. Вычисляется (8)–(11)

$$m_1 = C^{(p+1)/4} mod\ p; \quad (8)$$

$$m_2 = (p - C^{(p+1)/4}) mod\ p; \quad (9)$$

$$m_3 = C^{(q+1)/4} mod\ q; \quad (10)$$

$$m_4 = (q - C^{(q+1)/4}) mod\ q. \quad (11)$$

Затем выбирается целые числа

$$a = q\ (q^{-1}\ mod\ p); \quad (12)$$

$$b = p\ (p^{-1}\ mod\ q). \quad (13)$$

Четырьмя возможными решениями являются:

$$M_1 = (am_1 + bm_3)\ mod\ n, \quad (14)$$

$$M_2 = (am_1 + bm_4)\ mod\ n, \quad (15)$$

$$M_3 = (am_2 + bm_3)\ mod\ n, \quad (16)$$

$$M_4 = (am_2 + bm_4)\ mod\ n. \quad (17)$$

Один из четырех результатов M_1, M_2, M_3 и M_4, равно M. Если сообщение написано по английски, выбрать правильное М, нетрудно. С другой стороны, если сообщение является потоком случайных битов (скажем, для генерации ключей или цифровой подписи), способа определить, какое M, – правильное, нет. Одним из способов решить эту проблему служит добавление к сообщению перед шифрованием известного заголовка.

Впоследствии Вильямс улучшил эту схему. Вместо возведения в квадрат открытого текста сообщения, возведите его в третью степени. Большие простые числа должны быть конгруэнтны 1 по модулю 3, иначе открытый и закрытый ключи окажутся одинаковыми. Даже лучше, существует только одна уникальная расшифровка каждого шифрования.

Преимущество схем Рабина и Вильямса перед *RSA* в том, что доказано, что они также безопасны, как и разложение на множители. Однако перед вскрытием с выбранным шифротекстом они совершенно беззащитны. Если вы собираетесь использовать эти схемы для случаев, когда взломщик может выполнить такое вскрытие (например, алгоритм цифровой подписи, когда взломщик может выбирать подписываемые сообщения), не забывайте использовать перед подписанием однонаправленную хэш-функцию.

Рабин предложил другой способ защититься от такого вскрытия: к каждому сообщению перед хэшированием и подписанием добавляется уникальная случайная строка. К несчастью, после добавления однонаправленной хэш-функцией тот факт, что система столь же безопасна, как и разложение на множители, больше не является доказанным. Хотя с практической точки зрения добавление хэширования не может ослабить систему.

Схема ElGamal. Ее можно использовать как для цифровых подписей, так и для шифрования, его безопасность основана на трудности вычисления дискретных логарифмов в конечном поле.

Для генерации пары ключей сначала выбирается простое число p и два случайных числа, g и x, оба эти числа должны быть меньше p. Затем вычисляется

$$y = g^x \bmod p. \tag{16}$$

Открытым ключом являются y, g и p. И g, и p можно сделать общими для группы пользователей. Закрытым ключом является x.

Схема McEliece. В 1978 году Роберт МакЭлис разработал криптосистему с открытыми ключами на основе теории алгебраического кодирования. Этот алгоритм использует существование определенного класса исправляющих ошибки кодов, называемых кодами Гоппа. Он предлагал создать код Гоппа и замаскировать его как обычный линейный код. Существует быстрый алгоритм декодирования кодов Гоппа, но общая проблема найти слово кода по данному весу в линейном двоичном коде является NP-полной**.** Ниже приведен только краткий обзор.

Пусть $dH(x,y)$ обозначает расстояние Хэмминга между x и y. Числа n, k и t служат параметрами системы.

Закрытый ключ состоит из трех частей: G' – это матрица генерации года Гоппа, исправляющего t ошибок. P – это матрица перестановок размером $n*n$. S - это *nonsingular* матрица размером $k*k$. Открытым ключом служит матрица G размером $k*n$.

$$G = SG'P \tag{17}$$

Открытый текст сообщений представляет собой строку k битов в виде k-элементного вектора над полем $GF(2)$.

Для шифрования сообщения случайным образом выбирается n-элементный вектор z над полем $GF(2)$, для которого расстояние Хэмминга меньше или равно t.

$$c = mG + \mathrm{z} \tag{18}$$

Для дешифрирования сообщения сначала вычисляется $c' = cP^{-1}$. Затем с помощью декодирующего алгоритма для кодов Гоппа находится m' , для которого $dH(m'G,c)$ меньше или равно t. Наконец вычисляется $m = m'S^{-1}$.

В своей оригинальной работе МакЭлис предложил значения n = 1024, t = 50 и k = 524. Это минимальные значения, требуемые для безопасности.

Хотя этот алгоритм был одним из первых алгоритмов с открытыми ключами, и вне появлялось публикаций о его успешном криптоаналитическом вскрытии, он не получил широкого признания в криптографическом сообществе. Схема на два-три порядка быстрее, чем *RSA*, но у нее есть ряд недостатков. Открытый ключ огромен: 2^{19} битов. Сильно увеличивается объем данных – шифротекст в два раза длиннее открытого текста.

Ни одна из попыток криптоанализа не достигла успеха для общего случая, хотя сходство между алгоритмом МакЭлиса и алгоритмом рюкзака немного волнует.

Криптосистемы с эллиптическими кривыми. Эллиптические кривые изучались многие годы, и по этому вопросу существует огромное количество литературы. В 1985 году Нил Коблиц и В.С. Миллер независимо предложили использовать их для криптосистем с открытыми ключами. Они не изобрели нового криптографического алгоритма, использующего эллиптические кривые над конечными полями, но реализовали существующие алгоритмы, подобные *Diffie-Hellman*, с помощью эллиптических кривых.

Эллиптические кривые вызывают интерес, потому что они обеспечивают способ конструирования «элементов» и «правил объединения», образующих группы. Свойства этих групп известны достаточно хорошо, чтобы использовать их для криптографических алгоритмов, но у них нет определенных свойств, облегчающих криптоанализ. Например, понятие «гладкости» неприменимо к эллиптическим кривым. То есть, не существует такого множества небольших элементов, используя которые с помощью простого алгоритма с высокой вероятностью можно выразить случайный элемент. Следовательно, алгоритмы вычисления дискретного логарифма показателя степени не работают.

Особенно интересны эллиптические кривые над полем $GF(2^n)$. Для n в диапазоне от 130 до 200 несложно разработать схему и относительно просто реализовать арифметический процессор для используемого поля. Такие алгоритмы потенциально могут послужить основой для более быстрых криптосистем с открытыми ключами и меньшими размерами ключей. Эллиптические кривые используются двумя аналогами *RSA*.

Выводы. Таким образом, в статье рассмотрены и проанализированы основные методы шифрования данных. По предварительным оценкам, выбраны наиболее оптимальные и универсальные 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.** *Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А.* Алгоритмические основы эллиптической криптографии. – Москва: МЭИ, 2000. – 100 с. **4.** *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