УДК 681.324.067

О.В. ЖИЛИН, А.И. БАЗИЛЕВСКИЙ

ИСПОЛЬЗОВАНИЕ ОДНОСТОРОННЕГО ПРЕОБРАЗОВАНИЯ, ОСНОВАННОГО НА ФУНКЦИЯХ ЛЮКА В НЕСИММЕТРИЧНЫХ КРИПТОСИСТЕМАХ

Введение

В настоящее время вопросы защиты информации в компьютерных системах приобрели особую актуальность. Это связано с более широким применением компьютерной техники в государственных и коммерческих структурах. В связи с все возрастающими требованиями к целостности, конфиденциальности и аутентичности сообщений и их источников все большее внимание уделяется несимметричным алгоритмам шифрования.

Самым первым, и в то же время по-прежнему наиболее популярным несимметричным алгоритмом является алгоритм RSA [1], разработанный Ривестом, Шамиром и Адлманом в 1978 году. На данный момент существует множество реализаций этого алгоритма, он стандартизирован и широко применяется. Разумеется, что хорошо изучены и слабые стороны этого алгоритма. В связи с этим постоянно происходит поиск новых криптоалгоритмов, которые бы превзошли RSA по показателю Быстродействие/Стойкость. Одной из таких попыток являются системы класса Эль-гамаля [2]. Эти системы более стойкие, чем RSA, однако работает на порядок медленнее.

В данной работе рассматривается криптосистема, основанная на математическом аппарате функций Люка. Эти системы имеют перед RSA ряд преимуществ в стойкости, но, к сожалению, их быстродействие уступает RSA. Реально реализуемые на ЭВМ методы построения таких систем были предложены П. Смитом и М. Ленном в [3].

1. Анализ несимметричных криптосистем

Впервые принципы построения несимметричных криптосистем были рассмотрены Диффи и Хелманом в [4]. Они предложили использовать в криптографии односторонние функции. Эти функции характерны тем, что вычисление обратной функции за разумное время вычислительно сложно или невозможно, если неизвестна дополнительная информация, обычно называемая ключом. Вид функции и ключ не зависят от посылаемого сообщения. Обычно используется числовая функция, в качестве ключа используется число определенной длины, вычисление функции называется преобразованием по закону открытого ключа, а вычисление обратной функции – преобразованием по закону секретного ключа. Наибольшая безопасность в криптосистеме достигается, если имеется возможность производить секретные преобразования только у одного из абонентов. Это позволяет хранить секретный ключ в строгой тайне. Открытый ключ, предназначенный для шифрования сообщений, может быть широко распространен, однако зашифрованные сообщения будет иметь возможность читать только хозяин секретного ключа.

Диффи и Хелман указали на то, что такая же методология может быть использована для цифровой подписи сообщений, то есть обеспечения их аутентичности. В этом случае подписывающий сообщение абонент выполняет над открытой подписью преобразование по закону секретного ключа, а все владельцы открытого ключа могут проверить подпись, осуществив преобразование по закону открытого ключа и проанализировав полученную открытую подпись.

После публикации статьи Диффи и Хелмана было предложено множество односторонних функций. Наиболее широко известной и применяемой является функция, используемая в криптоалгоритмах RSA и Эль-Гамаля. Эти алгоритмы основаны на возведении сообщения, рассматриваемого как число, в степень по модулю большого числа.

Преобразование по закону открытого ключа в системе RSA представляется следующим соотношением:

$$f_{RSA} = M^e \pmod N. \qquad (1)$$

Если M – шифруемое сообщение, то $f_{RSA}(M)$ – криптограмма M'. Для того чтобы сделать f_{RSA} односторонней функцией, N выбирается как произведение 2-х больших простых чисел p, q. Для 1преобразования по закону секретного ключа необходимо число d, такое что:

$$e \cdot d \equiv 1 (\mathrm{mod}\,(p-1)\cdot(q-1)) \quad (2)$$

Если e взаимнопросто с (p-1)(q-1), то d всегда может быть вычислено при известных p, q. Преобразование по закону секретного ключа выглядит следующим образом:

$$M = (M')^d \,(\mathrm{mod}\, N). \quad (3)$$

При этом необходимо соблюдение условия M<N. Как видно, преобразования (1) и (3) отличаются только используемым ключом e или d.

Первой трудностью при реализации RSA является то, что в качестве N используются длинные числа (минимум 512 бит). В связи с этим возникает проблема реализации арифметики многократной точности. В настоящее время существуют алгоритмы возведения в степень, позволяющие возводить длинные числа в степень за время $O(\log_2 N)$, где N – показатель степени.

Функция f_{RSA} является односторонней, так как не существует способа вычисления M из M' по формуле (3) без знания d, а также способа вычислить d из e и N, кроме факторизации N. Было доказано, что любой метод получения d приблизительно эквивалентен или сложнее факторизации. В настоящее время методом решета общего числового поля было факторизовано число длиной 428 бит, однако, выбирая N достаточно большим (например, 1024 бита) можно обеспечить определенную стойкость системы. Сегодня RSA считается системой с доказуемо стойкостью (вероятно стойкой). Стойкость RSA обеспечивается сложностью задачи факторизации модуля.

Одной из слабых сторон RSA системы является то, что для цифровой подписи (ЦП) по RSA возможен адаптивный криптоанализ с выбранным текстом. При этом возможно получить подпись ложного сообщения после того, как абонент-жертва атаки подпишет несколько сообщений, вид которых специально подобран злоумышленником. Эта атака возможна из-за мультипликативности ЦП по RSA, то есть

$$M^d \cdot L^d = (M \cdot L)^d. \quad (4)$$

Если злоумышленнику известны M, M^d, L, L^d, то ML и $(ML)^d$ могут быть вычислены без знания d.

Такая ситуация, возможно, трудно реализуема, так как сообщения перед подписью обычно хешируют, однако применение хеш-функций (ХФ) не может гарантировать невозможность подделки подписи. Возможна следующая последовательность действий злоумышленника:

Выбирается ложное сообщение, хешируется, если необходимо и раскладывается на простые сомножители.

Генерируется множество "невинных" сообщений, они, если необходимо, хешируются, факторизуются и среди них выбираются те сообщения, которые содержат множители, полученные на шаге 1 или множители других сообщений, полученных на данном шаге.

Добываются подписи сообщений, полученных на шаге 2.

В процессе элиминации получают подписи простых чисел-сомножителей сообщений.

Путем перемножения подписей сомножителей ложного сообщения в соответствующей степени получаем подпись ложного сообщения.

Для реализации такой атаки необходимо большое количество сообщений, однако существуют и другие способы добычи подписей набора простых чисел, и данная атака может оказаться реально осуществимой. Это означает, что хеширование не может гарантированно защитить подпись от подделки. В отличие от RSA подписи в описываемой здесь системе, надежно защищены от таких атак, так как они не мультипликативны.

2. Определение и свойства функции Люка

Данная система основана на односторонней функции, отличной от тех, что используются в системах RSA и Эль-Гамаля, определяемой функциями Люка. Поскольку эти функции можно считать обобщением степеней, то преобразования по закону открытого и секретного ключей производится

аналогичным образом и все атаки на эту систему могут быть использованы и для RSA, но не наоборот что доказывает, что LUC является более стойкой системой.

Функции Люка являются частным случаем линейных рекуррентных соотношений высшег порядка. Если $P_1, P_2, ..., P_m$ - целые числа то можно определить последовательность целых чисел $\{T_n\}$ следующим образом:

$$T_n = P_1 \cdot T_{n-1} + P_2 \cdot T_{n-2} + .. + P_m \cdot T_{n-m} \quad (5)$$

$T_1, T_2, .. T_{m-1}$ определяются независимо. Выражение (5) называется линейным рекуррентным соотношением (ЛРС) m-го порядка. Можно доказать, что последовательность, определяемая ЛРС первого порядка состоит из степеней P_1, умноженных на константу T_0. Следовательно, ЛРС более высоких порядков являются обобщением степеней и не удивительно, что RSA система может быть реализована с использованием ЛРС порядка, большего, чем первый.

В данном случае мы обсудим только ЛРС 2-го порядка, которую можно представить в виде:

$$T_n = P \cdot T_{n-1} - Q \cdot T_{n-2} \quad (6)$$

В виде (5) выражение (6) можно представить следующим образом:

Пусть a и b - корни следующего полинома 2-го порядка:

$$x^2 - P \cdot x + Q = 0 \quad (7)$$

Пусть c_1 и c_2 – целые числа, тогда последовательность $\{c_1 a^n + c_2 b^n\}$ обладает следующим свойством:

$$P \cdot (c_1 \cdot a^{n-1} + c_2 \cdot b^{n-1}) - Q \cdot (c_1 \cdot a^{n-2} + c_2 \cdot b^{n-2}) = c_1 \cdot a^n + c_2 \cdot b^n \quad (8)$$

Таким образом, любая последовательность вида (6) может быть представлена в виде $\{c_1 a^n + c_2 b^n\}$, где $T_0 = c_1 + c_2$, $T_1 = c_1 a + c_2 b$. Следует отметить, что если T_0 и T_1 – целые, то последовательность будет состоять из целых чисел, хотя c_1, c_2, a, b могут быть комплексными.

Нас интересует два подмножества последовательностей вида (6). Они обозначаются $\{U_n\}$ и $\{V_n\}$ и определяются:

$$U_n = \frac{a^n - b^n}{a - b} \quad \left(c_1 = \frac{1}{a - b} = -c_2\right), \quad (9)$$

$$V_n = a^n + b^n \quad (c_1 = 1 = c_2). \quad (10)$$

Это всегда будут последовательности целых чисел, первые два члена которых

$$U_0 = 0, U_1 = 1, \; V_0 = 1, V_1 = P. \quad (11)$$

Эти последовательности зависят только от P и Q и называются функциями Люка от P и Q. Иногда они обозначаются Un(P,Q) и Vn(P,Q). Расширенная теория этих функций обсуждается Леммером в работе [5].

Следует отметить, что для любого целого N выполняется соотношение:

$$U_n(P \bmod N, Q \bmod N) = U_n(P, Q) \pmod N \quad (12)$$

Доказательство этого соотношения можно получить методом математической индукции. Аналогично

$$V_n(P \bmod N, Q \bmod N) = V_n(P, Q) \pmod N \quad (13)$$

Поскольку корни полинома (7) a и b удовлетворяют соотношениям:

$$a + b = P, \; a \cdot b = Q, \quad (14)$$

то нетрудно вывести множество соотношений между функциями Люка Un и Vn и коэффициентами ЛРС (6), P и Q. Дискриминант полинома (7), $D = P^2 - 4Q$, может быть выражен через a и b:

$$D = (a - b)^2. \quad (15)$$

Приведем некоторые из этих соотношений:

$$V_{2n} = V_n^2 - 2 \cdot Q^n, \tag{16}$$

$$V_{2n-1} = V_n \cdot V_{n-1} - P \cdot Q^n, \tag{17}$$

$$V_{2n+1} = P \cdot V_n^2 \cdot V_{n-1} - Q \cdot V_n \cdot V_{n-1} - P \cdot Q^n, \tag{18}$$

$$V_n^2 = D \cdot U_n^2 + 4 \cdot Q^n, \tag{19}$$

$$2 \cdot V_{n+m} = V_n \cdot V_m + D \cdot U_n \cdot U_m, \tag{20}$$

$$2 \cdot Q^m \cdot V_{n-m} = V_n \cdot V_m - D \cdot U_n \cdot U_m. \tag{21}$$

Еще одним важным свойством функций Люка является:

$$V_n(V_k(P,Q),Q^k) = V_{nk}(P,Q). \tag{22}$$

Из данного выражения следует, что мы можем сформулировать правило умножения функций Люка, аналогичное правилу умножения степеней, где индекс функции Люка будет выступать аналогом показателя степени. Приняв Q=1 получим более простое соотношение:

$$V_{nk}(P,1) = V_n(V_k(P,1),1). \tag{23}$$

Существует связь между индексом функции Люка и делителями ее значения. Для пояснения необходимо ввести определение символа Лежандра:

Пусть a – целое, p – простое. Символ Лежандра $L(a,p)$ равен 1, если существует b, такое что $b^2 \equiv a \pmod p$, иначе он равен -1.

Если p - простое, не делитель Q или D, и e=L(D,p), то, как Леммер показал в [5],

$$U_{k(p-e)}(P,Q) \equiv 0 \pmod p \text{ для любого целого к.} \tag{24}$$

а также

$$V_{k(p-e)}(P,Q) \equiv 2 \cdot Q^{k(1-e)/2} \pmod p \text{ для любого целого к.} \tag{25}$$

Для функций Люка, зависящих от взаимнопростых целых P и Q, существует обобщение функции Эйлера, называемой функцией Леммера [5]. В данном случае нам необходимо применить ее для числа N вида N=pq, где p и q различные нечетные простые числа. Тогда функция Леммера определяется следующим образом:

$$T(N) = [p - L(D,p)] \cdot [q - L(D,q)]. \tag{26}$$

Как и в случае функции Эйлера полное произведение необязательно для получения нужного результата, достаточно найти наименьшее общее кратное (НОК) сомножителей (26). Тогда обобщенная функция Леммера определяется следующим образом:

$$S(N) = НОК([p - L(D,p)] \cdot [q - L(D,q)]). \tag{27}$$

Из выражения (27) следует, что выражения (24) и (25) можно записать в виде:

$$U_{kS(N)}(P,1) \equiv 0 \pmod N \text{ для любого целого к.} \tag{28}$$

$$V_{kS(N)}(P,Q) \equiv 2 \pmod N \text{. Для любого целого к.} \tag{29}$$

Тогда если N=pq – произведение двух различных нечетных простых чисел, P<N и взаимнопростое с N, e – любое число взаимнопростое с S(N), а d вычислено по расширенному алгоритму Евклида из соотношения ed=kS(N)+1, тогда:

$$V_d(V_e(P,1),1) \equiv P \pmod N. \tag{30}$$

Этот результат позволяет определить одностороннюю функцию, аналогичную задаваемой формулами (1) и (3), основанную на функциях Люка. Однако явно видно отличие – отсутствие

симметрии между $V_e(P,1)$ и инверсной функцией $V_d(R,1)$. d и e связаны через S(N), то есть через квадратичные вычеты дискриминанта D, который зависит от P.

Для функций Люка справедливо также соотношение

$$L(D,p) = L\left(P^2 - 4, p\right) = L(V_e^2(P,1) - 4, p) \quad (31)$$

Это означает, что значение S(N) одно и то же для P и функции Люка $V_e(P,1)$.

Используя полученные результаты можно построить систему с открытыми ключами, аналогичную RSA. Пусть выбраны числа N и e, такие что N - произведение двух различных нечетных простых чисел p и q, e взаимнопросто с (p-1)(q-1)(p+1)(q+1). Пусть M - сообщение, взаимнопростое с N. Определим преобразование по закону открытого ключа как

$$f_{LUC}(M) = V_e(M,1) \pmod N. \quad (32)$$

Для определения соответствующего преобразования по закону открытого ключа необходимо число d, вычисляемое из соотношения

$$de = 1 \pmod{S(N)}. \quad (33)$$

Преобразование по закону секретного ключа определяется заменой ключа e на d:

$$M = V_d(M',1) \pmod N. \quad (34)$$

Из соотношения (30) следует, что соотношение (34) будет всегда верно при соблюдении всех условий.

3. Возможности реализации криптографических систем с использованием функций Люка

При реализации системы LUC возникают две главные проблемы. Первая из них - вычисление значений функций V_e и V_d, для больших значений e и d, а также то, что значение ключа d зависит от сообщения. Первая проблема решаема, так как из соотношений (16)-(19) следует, что для вычисления функций Люка можно использовать обычный двоичный алгоритм для возведения длинных чисел в степень по модулю. При этом могут быть использованы любые эвристические процедуры, применяемые для данного алгоритма (например, блочный метод) [6]. Такой метод вычисления гарантирует, что вычисление функции Люка по временным затратам сравнимо с возведением числа в соответствующую степень и превосходит его всего на 50%.

Что касается второй трудности, то на самом деле существует всего 4 возможных значения ключа d, так как существует всего 4 возможных значения S(N): НОК[(p+1)(q+1)], НОК[(p+1) (q−1)], НОК[(p−1) (q+1), НОК[(p−1) (q−1). Все эти значения известны когда известен модуль N, поэтому соответствующие 4 значения ключа d могут быть вычислены заранее. Для выбора нужного ключа при произведении преобразования по закону секретного ключа вычисляется дискриминант D сообщения или криптограммы, затем вычисляются символы Лежандра L(D,p), L(D,q) и выбирается соответствующий ключ. Это также означает, что, в отличие от RSA, делители модуля p и q должны храниться вместе с секретными ключами и, разумеется, в строгом секрете.

Наличие 4 ключей приводит к возрастанию времени преобразования по закону секретного ключа. Вычисление символов Лежандра L(D,p), L(D,q) по вычислительной сложности равно $O(\log_2 p)+O(\log_2 q)$. Таким образом, данный процесс занимает приблизительно на 80% больше времени, чем аналогичный процесс в RSA.

Существует способ избежать вычисления и значения четырех секретных ключей и составляющих модуля. Для этого вместо функции S(N) в выражение (33) подставляется функция R(N), определяемая следующим образом:

$$R(N) = (p-1)(q-1)(p+1)(q+1). \quad (35)$$

При этом размер ключа d возрастет в два раза, и, соответственно, возрастет время вычисления функции V_d, причем практически в два раза.

Аналогично RSA, преобразование по закону секретного ключа может быть осуществлено злоумышленником только если он найдет способ вычисления V_d без знания d, или вычисления d из e и

N. Решение первой проблемы, как и в RSA, - полный перебор значений. Отличие в решении второй проблемы заключается в том, что для каждой пары e и N существует 4 ключа d, применяемые в зависимости от сообщения.

Поскольку функции Люка - обобщение степеней, то удачная криптографическая атака на LUC является успешной и для RSA. Однако, так как в LUC применяются дополнительные усложнения, то обратное утверждение неверно. Например, поскольку подпись LUC не мультипликативна, то адаптивный криптоанализ для такой системы не применим. Из этого следует, что LUC криптографически сильнее RSA.

Фактически, любая криптографическая система, основанная на возведении в степень, может быть преобразована для использования функций Люка для определения односторонней функции. Например, можно разработать систему класса Эль-Гамаля, основанную на функциях Люка.

По классическому алгоритму шифрования по Эль-Гамалю, в системе генерируется простое число p и первообразный элемент a. Затем каждый абонент генерирует себе секретный ключ x, затем рассылает открытый ключ $y=a^x$. Для того, чтобы зашифровать сообщение генерируется случайный сеансовый ключ k и вычисляется $L=y^k \bmod p$. Вычисляются две части криптограммы: $c1=a^k$, $c2=L\cdot M$, которые отправляются абоненту.

На приемной стороне вычисляется $L=(ak)^x=(c1)^x$, с использованием секретного ключа x. Затем вычисляется обратный элемент L, $L'L=1 \bmod p$, и восстанавливается исходное сообщение $M=c2\cdot L'$.

Для шифрования с использованием функций Люка процесс происходит аналогично. Открытый ключ y в этом случае $y=V_x(g,1) \bmod p$. Нам также необходим сеансовый ключ k. Затем мы вычисляем $G=V_k(y,1) \bmod p$. После этого вычисляются две части криптограммы $d1=V_k(g,1) \bmod p$, $d2=G\cdot M \bmod p$.

На приемной стороне вычисляется $G=V_x(d1,1) \bmod p$, с использованием секретного ключа x. Затем вычисляется обратный элемент G, $G'\cdot G=1 \bmod p$, и восстанавливается исходное сообщение $M=d2\cdot G'$.

Следует заметить, что, как и в классическом алгоритме, криптограмма в два раза больше модуля.

Такое же преобразование к функциям Люка можно провести для прикладных алгоритмов класса Эль-Гамаля, в которых длина криптограммы гораздо меньше за счет использования двухмодульного преобразования.

На основании теоретических сведений, изложенных выше, были алгоритмически реализованы подпрограммы для вычисления функций Люка, а также производящие генерацию ключей и производящие преобразования по закону этих ключей для системы LUC, а также для шифрования и цифровой подписи, классической и прикладной по алгоритмам класса Эль-Гамаля.

Разработка производилась на алгоритмическом языке C. Результатом разработки является модули с подпрограммами, которые могут быть откомпилированы для подключения и использования в 16-битных и 32-битных приложениях. Компиляция для 16-битных приложений производилась с помощью компилятора Borland C++ 3.1, а для 32-битных – с помощью Borland C++ Builder.

Далее рассматриваются временные характеристики разработанного программного обеспечения.

В таблице 1 приводятся временные характеристики вычисления функций Люка для 512 битного модуля под управлением ОС MS-DOS. Данные в таблице 1 показывают, что вычисление V-функции Люка приблизительно в два раза дольше возведения в степень по модулю, что соответствует теоретическим оценкам.

Таблица 1

Функция	Время выполнения, с
Блочное возведение в степень по Монтгомери	0.143900
Вычисление V- функции Люка	0.301000
Вычисление UV - функции Люка	0.472400
Вычисление UV - функции Люка по блочному алгоритму	0.420200

В таблице 2 приведены временные характеристики системы LUCRSA.

Таблица 2

Действие	Время выполнения, с
Подпись (Расшифровка)	0.362500
Проверка подписи (Шифрование)	0.316900

В таблице 3 приведены временные характеристики криптосистемы шифрования по классическому Эль-Гамалю с использованием функций Люка.

Таблица 3

Действие	Время выполнения, с
Шифрование	0.580100
Расшифровка	0.294400

В таблице 4 приведены временные характеристики криптосистемы ЦП по классическому Эль-Гамалю с использованием функций Люка.

Таблица 4

Действие	Время выполнения, с
Подпись	0.431700
Проверка	1.132600

В таблице 5 приведены временные характеристики криптосистемы ЦП DSS с использованием функций Люка.

Таблица 5

Действие	Время выполнения, с
Подпись	0.075800
Проверка	0.288900

Полученные результаты показывают, что криптосистемы с использованием функций Люка уступают традиционным аналогам по быстродействию приблизительно в два раза.

Заключение

В данной работе проведено исследование о применимости функций Люка в качестве односторонних функций в несимметричных алгоритмах, а также приведены практические результаты такого использования. Результаты показывают возможность использования разработанных алгоритмов в криптографических системах. Они показывают, что по времени работы эти алгоритмы сравнимы с традиционными, однако теоретические исследования показывают более высокую стойкость этих алгоритмов. В частности показана невозможность применения адаптивного криптоанализа в системах с использованием алгоритмов, основанных на функциях Люка.

Как показал Шеннон, применение в криптосистеме разнообразных алгоритмов позволяет существенно увеличить стойкость системы. В настоящее время в мире в большинстве применяются традиционные криптосистемы, основанные на возведении в степень по модулю, следовательно, хорошо изучены сильные и слабые стороны таких систем. Изменение алгоритма при остающейся неизменной математической базе редко позволяет получить алгоритм, стойкость которого на порядок выше предшественников. Поэтому необходим постоянный поиск и обоснование нового математического аппарата для построения криптосистем. Одним из примеров такого математического аппарата являются криптосистемы, основанные на функциях Люка.

Список литературы: **1.** *R.L. Rivest, A. Shamir, L.M. Adleman.*– A method for obtaining digital signatures and public key cryptosystems. Comm. ACM, 1978. pp120–126. **2.** *T. El Gamal.* A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions of Information Theory (July 1985) pp100–140. **3.** *P. J. Smith, M. J. Lennon* – LUC: A New Public Key System. AsiaCrypt'93. pp 230–250. **4.** *W. Diffie, M. Hellman* - New directions in cryptography, IEEE Transaction of Information Theory, 22 (1976) pp644-654. **5.** *H. Lehmer* - An extended theory of Lucas' functions, Annals of Math., 31 (1930) pp 419-448.

Харьковский государственный технический университет радиоэлектроники

Поступила в редколлегию 15.03.2000