УДК 004.056.55

**М. И. Вахрушев**, **Е. А. Загурских**

*Новосибирский государственный университет*
*ул. Пирогова*, *1*, *Новосибирск*, *630090*, *Россия*

*vakhrushev.maxim@gmail.com*, *eugeny.zagurskih@yandex.ru*

# РАЗРАБОТКА КРИПТОГРАФИЧЕСКИХ СИСТЕМ С ОТКРЫТЫМ КЛЮЧОМ, ОСНОВАННЫХ НА ОБОБЩЕНИИ ЗАДАЧИ О РАНЦЕ

Развитие квантовых компьютеров ставит под угрозу множество распространенных на данный момент криптосистем, основанных на задачах факторизации, дискретного логарифмирования и других, которые могут быть решены за полиномиальное время на квантовом компьютере.

Однако на данный момент не предложено алгоритмов, позволяющих за полиномиальное время решать NP-полные задачи квантовыми компьютерами. Мы рассматриваем две криптосистемы с открытым ключом, основанные на NP-полных задачах о сумме подмножеств и целочисленного программирования.

*Ключевые слова*: асимметричная криптография, постквантовая криптография, рюкзачная криптография, NP-полнота, задача о сумме подмножеств.

## Введение

В данной работе рассматриваются два подхода к построению систем шифрования с открытым ключом на основе методов, использующих некоторые обобщения задач subset sum и целочисленного программирования.

Обоснование стойкости предлагаемых алгоритмов основано на том, что в результате шифрования получается система линейных уравнений, в которой число уравнений меньше, чем число неизвестных, к тому же на неизвестные налагаются ограничения. Доказано, что задача решения таких уравнений NP-полна [1]. На данный момент не существует алгоритмов, позволяющих решать данную задачу эффективно с помощью квантовых компьютеров, поэтому предлагаемые алгоритмы являются алгоритмами постквантовой криптографии.

## Задача subset sum

Задача subset sum заключается в поиске такого непустого подмножества некоторого набора чисел, что сумма чисел этого подмножества равна нулю. Эту задачу можно считать частным случаем задачи о ранце [2]. В данной статье будем рассматривать эквивалентную subset sum задачу, заключающуюся в нахождении подмножества, сумма элементов которого равна некоторому заданному числу $s$. Несмотря на то, что обе эти задачи являются NP-полными, существуют такие наборы чисел, для которых задачи легко решаются за полиномиальное время.

В данном параграфе рассматриваются методы построения таких наборов целых положительных чисел, что всевозможные суммы различных чисел не совпадают. Другими словами,

*Вахрушев М. И.*, *Загурских Е. А.* Разработка криптографических систем с открытым ключом, основанных на обобщении задачи о ранце // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2016. Т. 14, № 4. С. 31–38.

если нам известно некоторое число, то можно однозначно, с точностью до перестановки, найти те числа, сумма которых равна данному числу. Для таких наборов чисел легко решить задачу subset sum.

Один из вариантов такого набора – сверхвозрастающая (супервозрастающая) последовательность [3], однако для последовательностей такого вида имеет место проблема, заключающаяся в том, что представимые числа расположены очень редко. Приведем другой алгоритм построения наборов чисел, избавленных от данного недостатка.

*Алгоритм 1*

Будем одновременно строить две последовательности: сам набор чисел и набор всех возможных сумм элементов первого набора.

1. Прежде всего, в последовательность сумм записывается число 0, которое означает, что для составления суммы никаких чисел не берется.
2. Теперь выберем первое натуральное число $X$. Множество возможных сумм на данном этапе: $\{0, X\}$.
3. Далее будем последовательно, до момента достижения необходимой длины последовательности, выбирать некоторое число, не представимое в виде сумм предыдущих и большее любого из уже выбранных чисел.

Заметим, что всевозможные суммы различных чисел полученного набора не совпадают по построению и получившаяся последовательность не будет обязательно супервозрастающей. Кроме того, из построенных таким образом наборов чисел можно строить другие, воспользовавшись сильным модульным умножением [4].

Приведем два алгоритма построения новых последовательностей с помощью сильного модульного умножения. Пусть дана некоторая последовательность целых положительных чисел $\{a_i\}$ и последовательность возможных сумм ее элементов $\{s_j\}$. Выберем некоторое число $z$, большее максимальной возможной суммы, и посчитаем количество обратимых элементов по модулю данного числа, воспользовавшись функцией Эйлера. Пусть существует $N$ обратимых элементов, тогда с помощью модуля $z$ мы можем построить $N - 1$ новую последовательность каждым из алгоритмов (не $N$, так как среди обратимых элементов будет 1). Рассмотрим предлагаемые алгоритмы:

*Алгоритм 2.1*

Выбираем любой обратимый элемент $x$ по модулю $z$ и преобразовываем последовательность $\{a_i\}$ в $\{a_i x \ mod(z)\}$, а последовательность сумм $\{s_j\}$ пересчитываем заново, с учетом изменений в исходной последовательности.

*Алгоритм 2.2*

Выбираем любой обратимый элемент $x$ по модулю $z$ и добавляем к последовательности $\{a_i\}$ последовательность $\{a_i x \ mod(z)\}$, тем самым удлиняя исходный набор чисел в два раза. Последовательность сумм $\{s_j\}$ также необходимо пересчитать. Далее применяем Алгоритм 2.1 к новой последовательности и на выходе получаем новый набор чисел.

*Пример 1*

Ниже приводится пример построения набора чисел из 4 элементов с помощью Алгоритма 1:

- Пусть выполнен первый пункт алгоритма и на втором этапе в первый раз было выбрано число 5, тогда текущая последовательность чисел – {5} и возможные суммы – {0, 5}. Будем выполнять третий пункт алгоритма.

- Пусть сначала выбрали число 7, оно не может быть представлено в виде сумм 0 и 5 и больше обоих этих чисел. Тогда имеем набор {5, 7} и возможные суммы {0, 5, 7, 12}. Кандидаты на следующее число находятся в множестве чисел больших 7 и не равных 12.
- Пусть следующее число 11, значит, получаем набор чисел {5, 7, 11} и набор сумм {0, 5, 7, 11, 12, 16, 18, 23}. Следующее число должно быть больше 11 и не входить в множество {0, 5, 7, 11, 12, 16, 18, 23}.
- Пусть это 14, значит, последовательность – {5, 7, 11, 14} и возможные суммы – {0, 5, 7, 11, 12, 14, 16, 18, 19, 21, 23, 25, 26, 30, 32, 37}.
- Последовательность требуемой длины построена

*Пример 2*

Приведем примеры построения из последовательности {5, 7, 11, 14} других с помощью сильного модульного умножения.

Рассмотрим последовательность возможных сумм {0, 5, 7, 11, 12, 14, 16, 18, 19, 21, 23, 25, 26, 30, 32, 37}. Максимальное возможное значение суммы – 37. Для начала воспользуемся Алгоритмом 2.1. Выберем некоторое число большее этого значения, например 44. Обратимых элементов по модулю 44: $\varphi(44) = \varphi(4 \times 11) = 2 \times 10 = 20$, значит можно изготовить 19 новых последовательностей (не 20, т.к. $1 \times 1 = 1 mod(44)$). Например, рассмотрим число 5, обратным является 9, т.к $5 \times 9 = 1 mod(44)$. Значит относительно сильного модульного умножения на 5 последовательность {5, 7, 11, 14} перейдет в последовательность {25, 35, 11, 26}, а последовательность сумм перейдет в новую – {0, 11, 25, 26, 35, 36, 37, 46, 51, 60, 61, 62, 71, 72, 86, 97}.

Воспользуемся Алгоритмом 2.2 построения нового набора чисел. Выберем число, большее 37, например, 39. Тогда промежуточной последовательностью будет $\{5,7,11,14,5 \times 39,7 \times 39,11 \times 39,14 \times 39\} = \{5,7,11,14,195,273,429,546\}$. Максимальная сумма равна 1480. Заметим, что $31 \times 48 = 1 mod(1487)$ и $1480 < 1487$. Применим сильное модульное умножение и получим новую последовательность {0, 240, 336, 528, 672, 438, 1208, 1261, 929}, которая имеет длину в два раза больше, чем исходная.

**Прямой подход**

*Построение ключей*

Рассмотрим матрицу $B \in Z^{k \times n}$, $k \leq n/2$, строки которой построены следующим образом. Пусть $N=[n/k]$, тогда первые $(i-1) \cdot N$ элементов $i$-й строки ($i = 1 \ldots k$) – случайные, следующие $N$ элементов – последовательность, являющаяся легко разрешимой для задачи subset sum, например, построенная одним из вышеописанных алгоритмов, остальные элементы – нулевые. Система уравнений вида $Bx = y$ (1) при $x \in \{0,1\}^n$ либо не имеет решения, либо имеет единственное решение, которое находится за полиномиальное время. По построению $B$ эту систему легко решить следующим образом: из первого уравнения находим первые $N$ бит сообщения путем последовательного деления с остатком правой части уравнения на коэффициенты, начиная с наибольшего элемента легко разрешимой последовательности. Частное от каждого деления будет являться соответствующим битом сообщения, а остаток – делимым, используемым при вычислении следующего бита, и т. д.

Далее необходимо замаскировать матрицу $B$. Обозначим новую, замаскированную, матрицу $A$ и выберем число строк в ней $m \in [k, n/2]$. Для начала, первые $k$ строк новой матрицы выберем равные строкам матрицы $B$, а остальные выберем произвольные. Соответствующим образом расширив правую часть, получим систему уравнений (2): $Ax = y$, которая, так же, как и система (1) при $x \in \{0;1\}^n$, имеет единственное решение, если оно существует. Для увеличения криптостойкости можно осуществить перестановку уравнений системы.

Следующим шагом находим максимальную сумму по строке (максимум среди максимально возможных значений правой части разрешимой системы (2)) матрицы $A$. Выберем некоторое $p$, большее этой суммы, и произвольную матрицу $H \in Z^{m \times m}$, обратимую в кольце по модулю $p$.

Теперь изготовим из матриц $A$ и $H$ матрицу $R$: $R = HAmod(p)$. Полученная матрица $R$ будет являться открытым ключом, а матрицы $B$, $A$ и $H$ – закрытым. Система уравнений теперь принимает вид (3): $Rx = Hymod(p)$ и она также, при $x$ $\{0;1\}^n$, имеет единственное решение, если оно существует.

*Алгоритм 3* (*шифрование*)

Пусть имеем сообщение $e \in \{0;1\}^n$. Шифротекст получаем умножением на матрицу $R$: $c = \Re$.

*Алгоритм 4* (*дешифрование*)

Процедура дешифрования шифротекста представляет собой следующий алгоритм.

1. Первым делом возвращаемся к системе (2) путем домножения на матрицу, обратную к $H$: $H^{-1}c = H^{-1}HAemod(p)Ae \rightarrow H^{-1}cmod(p)$
2. Полученную систему мы можем решить (предварительно осуществив обратную перестановку, если мы переставляли строки на этапе построения ключей), сведя ее к виду (1) путем отбрасывания маскирующих уравнений.
3. Систему вида (1) мы можем решить в силу знания матрицы $B$ и способа построения ее строк.

*Пример 3*
*Построение ключей*

Рассмотрим в качестве примера систему вида (1), в которой будет 2 уравнения и 10 неизвестных. Пусть матрица $B$ образована с помощью двух последовательностей, таких, что всевозможные суммы различных чисел не совпадают, например последовательностями {2, 3, 7, 14, 27} и {4, 5, 10, 17, 41}. Первая последовательность задает коэффициенты при $x_1$, $x_2$, $x_3$, $x_4$, $x_5$ в первом уравнении, а вторая – коэффициенты при $x_6$, $x_7$, $x_8$, $x_9$, $x_{10}$ во втором. Таким образом,

$$B = \begin{pmatrix} 2 & 3 & 7 & 14 & 27 & 0 & 0 & 0 & 0 & 0 \\ 3 & 12 & 7 & 2 & 1 & 4 & 5 & 10 & 21 & 41 \end{pmatrix}$$

Система уравнений $Bx = b$ имеет вид

$$2x_1 + 3x_2 + 7x_3 + 14x_4 + 27x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = b_1$$
$$3x_1 + 12x_2 + 7x_3 + 2x_4 + 1x_5 + 4x_6 + 5x_7 + 10x_8 + 21x_9 + 41x_{10} = b_2$$

Добавив еще два произвольных уравнения, получаем матрицу

$$A = \begin{pmatrix} 2 & 3 & 7 & 14 & 27 & 0 & 0 & 0 & 0 & 0 \\ 3 & 12 & 7 & 2 & 1 & 4 & 5 & 10 & 21 & 41 \\ 5 & 4 & 1 & 40 & 12 & 3 & 7 & 2 & 3 & 21 \\ 0 & 11 & 3 & 2 & 10 & 7 & 4 & 1 & 11 & 8 \end{pmatrix}$$

Система уравнений $Ax = b$ представляет собой систему вида (2)

$$2x_1 + 3x_2 + 7x_3 + 14x_4 + 27x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = b_1$$
$$3x_1 + 12x_2 + 7x_3 + 2x_4 + 1x_5 + 4x_6 + 5x_7 + 10x_8 + 21x_9 + 41x_{10} = b_2$$
$$5x_1 + 4x_2 + 1x_3 + 40x_4 + 12x_5 + 3x_6 + 7x_7 + 2x_8 + 3x_9 + 21x_{10} = b_3$$
$$0x_1 + 11x_2 + 3x_3 + 2x_4 + 10x_5 + 7x_6 + 4x_7 + 1x_8 + 11x_9 + 8x_{10} = b_4$$

Следующим шагом находим максимальные значения для правых частей системы уравнений: $m_1=max(b_1)=53$, $m_2=max(b_2)=106$, $m_3=max(b_3)=98$, $m_4=max(b_4)=57$, $max\{m_1, m_2, m_3, m_4\}=106$. Возьмем простое $p=149>106$ и матрицу $H$:

$$H=\begin{pmatrix}3 & 7 & 0 & 2\\ 2 & 11 & 5 & 0\\ 0 & 0 & 2 & 1\\ 0 & 0 & 0 & 3\end{pmatrix}$$

Матрица обратима, так как $det(H)=114$ и $gcd(106, 149)=1$. Теперь получим матрицу $R = HAmodp$:

$$R=\begin{pmatrix}27 & 115 & 76 & 60 & 108 & 42 & 43 & 72 & 20 & 5\\ 62 & 9 & 96 & 101 & 125 & 59 & 90 & 120 & 97 & 109\\ 10 & 19 & 5 & 82 & 34 & 13 & 18 & 5 & 17 & 50\\ 0 & 33 & 9 & 6 & 30 & 21 & 12 & 3 & 33 & 24\end{pmatrix}$$

Таким образом, построенная матрица R является открытым ключом, а матрицы $B$, $A$ и $H$ – закрытым.

*Шифрование*

Пусть необходимо зашифровать сообщение $e=(1, 0, 0, 1, 0, 1, 1, 1, 0, 0)$. Вычисляя $c = \Re$ получим шифротекст $c = (244,581,128,42)$.

*Дешифрование*

Имея шифротекст $c$ и набор матриц из приватного ключа, восстановим сообщение $e$:

1. Домножаем уравнение $c = HAe$ на $H^{-1}$:

$$2x_1 + 3x_2 + 7x_3 + 14x_4 + 27x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = 16$$
$$3x_1 + 12x_2 + 7x_3 + 2x_4 + 1x_5 + 4x_6 + 5x_7 + 10x_8 + 21x_9 + 41x_{10} = 24$$
$$5x_1 + 4x_2 + 1x_3 + 40x_4 + 12x_5 + 3x_6 + 7x_7 + 2x_8 + 3x_9 + 21x_{10} = 57$$
$$0x_1 + 11x_2 + 3x_3 + 2x_4 + 10x_5 + 7x_6 + 4x_7 + 1x_8 + 11x_9 + 8x_{10} = 14$$

2. Мы знаем, что два последних уравнения не являются для нас существенными. Кроме того, мы знаем, что первые два уравнения построены с использованием специальных последовательностей, которые гарантируют, что всевозможные суммы разных чисел не совпадают. Из первого уравнения найдем первые 5 бит сообщения: $e_5=16/27=0$, $e_4=16/14=1$, $e_3=(16-14)/7=0$, $e_2=2/3=0$, $e_1=2/2=1$.

3. Аналогично найдем остальные биты сообщения из второго уравнения, полностью восстановив исходное сообщение $e=(1, 0, 0, 1, 0, 1, 1, 1, 0, 0)$.

**Двойственный подход**

*Построение ключей*

Рассмотрим систему уравнений вида (1) $Ax = b$, где $A \in Z^{n\times m}$, $b \in Z^m$ и сопоставим ей однородную систему $Ax = 0$. Заметим, что для увеличения стойкости алгоритма наравне с системой (1) можно рассматривать и системы (2) или (3), вся разница будет заключаться только в сложности решения конечной системы уравнений в фазе дешифрования, однако устойчивость схемы зависит не от этого фактора.

Для построения ключей необходимо найти максимальный набор линейно независимых решений этой системы, который состоит из $n$-$m$ векторов. Один из стандартных способов решения этой системы рассматривается в Примере 5.

Пусть $v_1 \dots v_{n-m}$ – полный набор решений. Выберем некое число $N<n$-$m$ и любые $N$ векторов (или $N$ линейных комбинаций векторов) из этого набора. Пусть это вектора $u_1 \dots u_N$. Далее выберем некоторое целое число $M$ и сгенерируем $N$ случайных векторов $\alpha_1 \dots \alpha_N$ длины $M$, после чего построим вектора $w_1 \dots w_M$ длины $n$ следующим образом:

$$w_i = \sum_{j=1}^{N} \alpha_{ji} u_j \,, i = 1 \dots M$$

Эти векторы также ортогональны векторам матрицы $A$ и являются открытым ключом, а матрица $A$ – закрытым. Заметим, что для увеличения стойкости алгоритма можно выполнить перестановку векторов $w_1 \dots w_M$. Кроме того, количество векторов $M$ должно быть выбрано таким образом, чтобы ранг матрицы, для которой эти вектора являются столбцами, не был максимальным, т.е. равным $n$.

Теперь рассмотрим алгоритмы шифрования и дешифрования в данной криптосистеме.

*Алгоритм 5* (*шифрование*)

На вход алгоритм принимает вектор $e$ длины $n$, такой, что $e_i \in \{0; 1\}, i \in [1 \dots n]$. Для начала выбираются $M$ случайных чисел $\lambda_1 \dots \lambda_M$ и после этого выполняется шифрование: $c = \sum_{i=1}^{M} \lambda_i w_i + e$. Построенный вектор $c$ – шифротекст для сообщения $e$.

*Алгоритм 6* (*дешифрование*)

Алгоритм принимает на вход вектор $c$ длины $n$. Для расшифровки сначала вычисляется произведение $y = Ac$. Т.к. вектора $w_1 \dots w_M$ ортогональны строкам $A$, то $Aw_i = 0, i = 1 \dots M$, а значит, $y = Ac = Ae$. Данную систему известно, как решать, а значит, вектор сообщения $e$ находится с помощью описанного алгоритма.

Рассмотрим пример использования криптосистемы, основанной на двойственном подходе.

*Пример 4*

Рассмотрим систему уравнений (4):

$$2x_1 + 3x_2 + 7x_3 + 14x_4 + 27x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = b_1$$
$$3x_1 + 12x_2 + 7x_3 + 2x_4 + 1x_5 + 4x_6 + 5x_7 + 10x_8 + 17x_9 + 41x_{10} = b_2$$

С данной системой сопоставим однородную систему уравнений

$$2x_1 + 3x_2 + 7x_3 + 14x_4 + 27x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = 0$$
$$3x_1 + 12x_2 + 7x_3 + 2x_4 + 1x_5 + 4x_6 + 5x_7 + 10x_8 + 17x_9 + 41x_{10} = 0$$

Эта система имеет 8 линейно независимых решений, которые необходимо найти для построения открытого ключа. Пусть $a_1$=(*2*, *3*, *7*, *14*, *27*, *0*, *0*, *0*, *0*, *0*), $a_2$=(*3*, *12*, *7*, *2*, *1*, *4*, *5*, *10*, *17*, *41*), тогда матрица системы (1) $A = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$.

*Построение ключей*

Рассмотрим один из возможных способов решения однородной системы. Заметим, что матрица при первых двух переменных: $C = \begin{pmatrix} 2 & 3 \\ 3 & 12 \end{pmatrix}$, ее определитель *det=15*. Если планируется решать данную систему в целых числах, то желательно, чтобы определитель равнялся 1. Этого можно добиться выбором другого набора случайных чисел во второй строчке уравнения.

Теперь применим стандартный подход к решению этой системы. Построим матрицу L с определителем 1 такую, что $U = LC$ – верхнетреугольная матрица. Напимер, при $L = \begin{pmatrix} -1 & 1 \\ -3 & 2 \end{pmatrix}$ получим: $U = \begin{pmatrix} 1 & 9 \\ 0 & 15 \end{pmatrix}$.

Умножим систему (5) слева на матрицу $L$, получим эквивалентную систему уравнений (6):

$$1x_1 + 9x_2 + 0x_3 - 12x_4 - 26x_5 + 4x_6 + 5x_7 + 10x_8 + 17x_9 + 41x_{10} = 0$$
$$0x_1 + 15x_2 - 7x_3 - 38x_4 - 79x_5 + 8x_6 + 10x_7 + 20x_8 + 34x_9 + 82x_{10} = 0$$

В данной системе во втором уравнении есть два взаимно простых числа -7 и 10 при $x_3$ и $x_7$ соответственно. Это означает, что переменные $x_2$, $x_4$, $x_5$, $x_6$, $x_8$, $x_9$, $x_{10}$ можно выбирать произвольно. Для упрощения примера выберем $x_2=x_4=x_5=x_6=x_8=x_9=x_{10}=1$, подставим эти значения во второе уравнение системы и получим уравнение $7x_3 - 10x_7 = 42$.

Используя алгоритм Евклида найдем переменные $x_3$ и $x_7$: $x_3=6$, $x_7=0$. Теперь подставим в первое уравнение системы (3) полученные значения и найдем $x_1=-78$. В результате получим первое решение однородной системы: $u_1$=(*-43*, *1*, *6*, *1*, *1*, *1*, *0*, *1*, *1*, *1*).

Повторив описанные выше действия для других пар взаимно простых чисел из коэффициентов второго уравнения системы (6) найдем полный набор решений системы однородных уравнений (5). Кроме того, можно рассмотреть матрицу $C$ при других переменных системы, построить для нее матрицу $L$ и повторить дальнейшие рассуждения. Если же уравнений в исходной системе больше двух, то рассуждения аналогичны.

Пусть $N=2$, тогда нужно найти еще одно решение. В этот раз рассмотрим пару взаимно простых чисел 15 и 8 при $x_2$ и $x_6$ и, повторив вышеописанные действия, найдем решение $u_2$= =(*-33*, *6*, *1*, *1*, *1*, *-14*, *1*, *1*, *1*, *1*).

Полученные вектора $u_1$, $u_2$ ортогональны векторам-строкам матрицы B по построению. Далее выберем $M=3$ и построим вектора открытого ключа $w_1$, $w_2$, $w_3$, выбрав случайные вектора $\alpha_1 = (1,2,3)$ и $\alpha_2 = (5,6,7)$: $w_1$=(*-208*, *31*, *11*, *6*, *6*, *-69*, *5*, *6*, *6*, *6*), $w_2$=(*-284*, *38*, *18*, *8*, *8*, *-82*, *6*, *8*, *8*, *8*), $w_3$=(*-360*, *45*, *25*, *10*, *10*, *-95*, *7*, *10*, *10*, *10*). Эти вектора также ортогональны векторам $a_1$, $a_2$ и являются элементами открытого ключа. В то же время матрица $A$ – секретный ключ.

*Шифрование*

Пусть сообщением является вектор $e$=(*0*, *1*, *0*, *1*, *0*, *1*, *0*, *1*, *0*, *1*). Выберем случайно 4 целых числа $\lambda_1 = 10, \lambda_2 = 2, \lambda_3 = 5$ и вычислим шифротекст:

$$c = \lambda_1 w_1 + \lambda_2 w_2 + \lambda_3 w_3 + e$$

В данном примере получим $c$=(*-3369*, *429*, *293*, *84*, *83*, *-951*, *167*, *84*, *83*, *83*).

*Дешифрование*

Вычисляем

$$y = Ac = \lambda_1 A w_1 + \lambda_2 A w_2 + \lambda_3 A w_3 + Ae = Ae$$

Решая систему уравнений $Ae = y = (17,69)$ согласно методу, описанному в параграфе 2, находим вектор зашифрованного сообщения $e$=(*0*, *1*, *0*, *1*, *0*, *1*, *0*, *1*, *0*, *1*).

Заметим, что для упрощения примера были выбраны заведомо неудачные «случайные» числа на разных этапах алгоритма, вследствие чего по шифротексту можно догадаться об элементах сообщения и секретного ключа.

Основным недостатком описанного подхода является рост размера ключа, поскольку все вычисления проводятся в целых числах. Однако эту проблему можно устранить, если выполнять все вычисления по некоторому модулю $p$, выбранному больше максимальной возможной суммы последовательности чисел, состоящей из коэффициентов уравнений системы.

**Заключение**

В данной работе предложено два способа построения криптосистем, основанных на обобщении частного случая задачи о ранце – задачи subset sum. Рассмотренные подходы ос-

нованы на том факте, что кольцо целых чисел является евклидовым. Таким образом, возможно построение криптографических систем на базе других евклидовых колец, например, на базе кольца многочленов с коэффициентами из конечного поля. Однако это тема для отдельных исследований, не рассматриваемых в данной работе.

**Список литературы**

1. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ. 3-е изд. М.: Вильямс, 2013. 1328 с.

2. *Martello S., Toth P.* Knapsack problems: Algorithms and computer interpretations. Wiley-Interscience, 1990. 306 с.

3. *Шнайер Б.* Прикладная криптография: протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. 610 с.

4. *Саломаа А.* Криптография с открытым ключом. М.: Мир, 1995. 320 с.

*Материал поступил в редколлегию 23.08.2016*

**M. I. Vakhrushev**, **E. A. Zagurskikh**

*Novosibirsk State University*
*1 Pirogov Str*., *Novosibirsk*, *630090*, *Russian Federation*

*vakhrushev.maxim@gmail.com*, *eugeny.zagurskih@yandex.ru*

**DESIGN OF KNAPSACK CRYPTOSYSTEMS**

The development of quantum computers endangers the great number modern cryptosystems based on factorization problem, discrete logarithm problem and other, which can be solved in polynomial time on a quantum computer. However, quantum computing algorithms can't efficiently solve NP-hard problems at present. In this paper, we consider two public key cryptosystems based on NP-hard problems: subset sum and integer programming.

*Keywords*: post-quantum cryptography, knapsack public-key cryptosystem, subset sum problem, NP-hard, knapsack problem.

**References**

1. T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms. 3rd. MIT Press, 2009. 1312 p.

2. S. Martello, P. Toth. Knapsack problems: Algorithms and computer interpretations. Wiley-Interscience, 1990. 306 p.

3. B. Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, 1996. 784 p.

4. A. Salomaa. Public-Key Cryptography – Springer Science & Business Media, 1996. 275 p.