УДК 515.171; 681.3

Ю. Е. Бояринова[1], **Я. В. Одарич**[2], **П. В. Трубников**[1]
[1]Институт проблем регистрации информации НАН Украины
ул. Н. Шпака, 2, 03113 Киев, Украина
[2]Национальный технический университет Украины «КПИ»
проспект Победы, 37, 03056 Киев, Украина

Реализация алгоритма Евклида для задачи разделения секрета

Рассмотрена реализация алгоритма Евклида для задачи разделения секрета на языке C++ с использованием гиперкомплексных числовых систем: комплексных, двойных и дуальных чисел.

Ключевые слова: *алгоритм Евклида, задача разделения секрета, комплексные числа, двойные числа, дуальные числа.*

Модулярный подход к формулировке и решению задач разделения секрета впервые предложил Ч. Асмус и Л. Блюм. Он состоит в следующем.

Пусть $m_1, m_2, ..., m_n$ — система попарно взаимно простых натуральных модулей. Предположим, что они упорядочены $m_1 < m_2 < ... < m_n$, и выполнено условие

$$M_2 = m_1 m_2 ... m_k > m_{n-k+1} m_{n-k+2} ... m_n = M_1,$$

а секрет взят из промежутка (M_1, M_2). Тогда часть секрета *i*-го участника a_i определяется наименьшим неотрицательным вычетом секрета x по модулю m_i. Получаем систему сравнений:

$$x \equiv a_i (\mathrm{mod}\, m_i), i = 1...n.$$

Полученное множество $\{a_1 ... a_n\}$ называется (n, k)-пороговой схемой.

Любая подсистема из k сравнений данной системы имеет единственное решение в промежутке (M_1, M_2). Это решение можно найти исходя из китайской теоремы об остатках.

Обозначим через M произведение всех модулей. Пусть $M_i = M / m_i$, N_i — число, обратное M_i по модулю m_i, $i = 1...k$. Таким образом, $M_i N_i \equiv 1 (\mathrm{mod}\, m_i)$. Тогда:

$$x = \sum_{i=1}^{k} a_i M_i N_i .$$

Для нахождения числа N_i в области вещественных чисел можно использовать функцию Эйлера:

$$N_i = M_i^{\varphi(m_i)-1} (\mathrm{mod}\, m_i),$$

которая рассмотрена только для поля вещественных чисел.

Для нахождения N_i наряду с применением функции Эйлера можно использовать и другие подходы. Нам предоставляется наиболее приемлемым подход, основанный на применении алгоритма Евклида. На первом этапе рассмотрим решение для вещественных чисел.

Из определения евклидова кольца следует, что для любых элементов $a, b \in R$, где $b \neq 0$, в кольце вещественных чисел можно так подобрать элементы q и r, что $a = bq + r$, причем или $r = 0$, или же $n(r) < n(b)$.

Тогда для отыскания наибольшего общего делителя применяется алгоритм Евклида, который состоит в следующем. Пусть a и b — положительные целые. Находим ряд равенств

$$\left.\begin{array}{l} a = b \cdot q_1 + r_2, 0 < r_2 < b \\ b = r_2 \cdot q_2 + r_3, 0 < r_3 < r_2 \\ \ldots\ldots\ldots \\ r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n, 0 < r_n < r_{n-1} \\ r_{n-1} = r_n \cdot q_n \end{array}\right\},$$

заканчивающийся тогда, когда получаем некоторое значение $r_{n+1} = 0$. Последнее неизбежно, так как ряд $b, r_2, r_3, \ldots$ как ряд убывающих чисел не может содержать более чем b положительных чисел.

Общие делители чисел a и b одинаковы с общими делителями чисел b и r_2, далее одинаковы с общими делителями чисел r_2 и r_3, чисел r_3 и r_4 и т.д., и, наконец, с делителем числа r_n. Одновременно с этим имеем:

$$(a, b) = (b, r_2) = (r_2, r_3) = \ldots = (r_{n-1}, r_n) = r_n .$$

Следовательно, наибольший общий делитель равен r_n. Для взаимно простых чисел $r_n = 1$.

Из теории чисел известно, что для любых взаимно простых a и b найдутся такие x и y, что $ax + by = 1$. Причем $ax = 1(\mathrm{mod}\, b)$ и $by = 1(\mathrm{mod}\, a)$.

Предположим, $a > b$. Тогда мы можем решить уравнения:

$$a \cdot x + b \cdot y = a,$$
$$a \cdot x + b \cdot y = b.$$

Первое уравнение имеет решение $x_0 = 1$, $y_0 = 0$, второе уравнение имеет решение $x_1 = 0, y_1 = 1$. Выполняя последовательно шаги алгоритма Евклида, получим систему уравнений для вычисления $x_i, y_i, x_{i-1}, y_{i-1}$:

$$\begin{aligned} &r_{i-1} \cdot x_{i-1} + r_i \cdot y_{i-1} = r_{i-1}, \\ &r_{i-1} \cdot x_i + r_i \cdot y_i = r_i. \end{aligned} \qquad i = 1 \ldots n.$$

Инициализируем начальные значения: $r_0 = a, r_1 = b$.

Далее выразим r_{i+1}:

$$r_{i+1} = r_{i-1} - q_i \cdot r_i = r_{i-1} \cdot x_{i-1} + r_i \cdot y_{i-1} - q_i \cdot (r_{i-1} \cdot x_i + r_i \cdot y_i),$$
$$r_{i+1} = r_{i-1} \cdot (x_{i-1} - q_i \cdot x_i) + r_i \cdot (y_{i-1} - q_i \cdot y_i) = r_{i-1} \cdot x_{i+1} + r_i \cdot y_{i+1}.$$

Следовательно,

$$x_{i+1} = x_{i-1} - q_i \cdot x_i,$$
$$y_{i+1} = y_{i-1} - q_i \cdot y_{i.}$$

Поскольку $r_{n+1} = 0$, и для взаимно простых чисел $r_n = 1$, искомые переменные будут равняться x_n и y_n.

Приведем короткий пример.
Необходимо решить сравнение $7 \cdot x = 1 (\mathrm{mod}\ 5)$
Следовательно, $7 \cdot x + 5 \cdot y = 1$.
$r_0 = 7 = 5 \cdot 1 + 2, \qquad q_1 = 1,$
$r_1 = 5 = 2 \cdot 2 + 1, \qquad q_2 = 2,$
$r_2 = 2 = 1 \cdot 2, \qquad q_3 = 2.$

$x_0 = 1, \qquad y_0 = 0,$
$x_1 = 0, \qquad y_1 = 1,$
$x_2 = 1 - 1 \cdot 0 = 1, \qquad y_2 = 0 - 1 \cdot 1 = -1,$
$x_3 = 0 - 2 \cdot 1 = -2, \qquad y_3 = 1 - 2 \cdot (-1) = 3.$
Действительно, $7 \cdot (-2) + 5 \cdot 3 = 1$.

Рассмотрим реализацию алгоритма Евклида для трех числовых систем вида:
– комплексные числа $a + ib$, $i^2 = -1$;
– дуальные числа $\alpha + \varepsilon\beta$, $\varepsilon^2 = 0$;
– двойные числа $p + eq$, $e^2 = 1$.

Для взаимно простых комплексных чисел $(a,b)=1$ выполняется следующее свойство: $ax+by=1$.

Алгоритм решения уравнения в целом аналогичен алгоритму решения для вещественных чисел, если не брать во внимание нахождение величин $q_i = q_i' + iq_i''$.

Поскольку $\frac{a}{b} = \frac{a_1 \cdot b_1 + a_2 \cdot b_2}{b_1^2 + b_2^2} + i \cdot \frac{a_1 \cdot b_2 - a_2 \cdot b_1}{b_1^2 + b_2^2} = r_1 + i \cdot r_2$ — нецелое число, то целую величину q находим из неравенства:

$$(r_1 - q_1)^2 + (r_2 - q_2)^2 \le \frac{1}{2}.$$

При этом следует учитывать, что в выражении $(a,b)=1$, где 1 — единица кольца комплексных чисел. Поэтому в процессе решения существует возможность получить последний ненулевой остаток r_n, который может быть равен –1, i или $-i$. Следовательно, для приведения уравнения к виду $ax+by=1$, необходимо полученные величины x и y разделить на r_n.

Пример
$(4 + i\cdot5) = 1 \bmod (3 + i\cdot4)$.
Следовательно, $(4 + i\cdot5)\cdot x + (3 + i\cdot4)\cdot y = 1$.
1) $\frac{4+i\cdot5}{3+i\cdot4} = 1{,}28 - i\cdot0{,}04$,
$(1{,}28 - q_1')^2 + (-0{,}04 - q_1'')^2 \le 0{,}5$,
$q_1 = 1$,
$r_0 = 4 + i\cdot5 = (3 + i\cdot4)\cdot1 + (1 + i)$;
2) $\frac{3+i\cdot4}{1+i} = 3{,}5 + i\cdot0{,}5$,
$(3{,}5 - q_2')^2 + (0{,}5 - q_2'')^2 \le 0{,}5$,
$q_2 = 3$,
$r_0 = 3 + i\cdot4 = (1 + i)\cdot3 + i$;
3) $\frac{1+i}{i} = i - 1$,

$x_0 = 1$,	$y_0 = 0$,
$x_1 = 0$,	$y_1 = 1$,
$x_2 = 1 - 1\cdot0 = 1$,	$y_2 = 0 - 1\cdot1 = -1$,
$x_3 = 0 - 3\cdot1 = -3$.	$y_3 = 1 - 3\cdot(-1) = 4$.

Поскольку $r_n = i$, то $x_3' = x_3 / r_n = i\cdot3$, $y_4' = y_4 / r_n = -i\cdot4$.
Действительно, $(4 + 5\cdot i)\cdot(i\cdot3) + (3 + i\cdot4)\cdot(-i\cdot4) = 1$.

Для двойных чисел выполняются следующие свойства.
Для любых целых двойных чисел α и β ($\beta \ne 0$ и не является делителем ну-

ля) существуют целые числа ρ и γ такие, что $\alpha = \beta\rho + \gamma$, причем $N(\gamma) < N(\beta)$.

Два целых двойных числа α, β называют взаимно простыми, если их нормы взаимно простые целые числа $(N(\alpha), N(\beta)) = 1$ и существуют целые двойные числа γ_1 и γ_2 такие, что

$$\alpha \cdot \gamma_1 + \beta \cdot \gamma_2 = 1.$$

Алгоритм нахождения чисел γ_1 и γ_2 ничем не отличается от алгоритма для комплексных чисел, кроме контроля возникновения делителей нуля.

Пример
$(7 + e\cdot2) = 1 \bmod (3 + e\cdot5)$.
Следовательно, $(7 + e\cdot2)\cdot x + (3 + e\cdot5)\cdot y = 1$:
1) $\dfrac{7+e\cdot2}{3+e\cdot5} = -0{,}69 + e\cdot1{,}81$,
$(-0{,}69 - q_1')^2 + (1{,}81 - q_1'')^2 \le 0{,}5$,
$q_1 = -1 + e\cdot2$,
$r_0 = 7 + e\cdot2 = (3 + e\cdot5)\cdot(-1 + e\cdot2) + e$;
2) $\dfrac{3+e\cdot5}{e} = 5 + e\cdot3$,
$x_0 = 1$, $y_0 = 0$,
$x_1 = 0$, $y_1 = 1$,
$x_2 = 1 - (-1 + e\cdot2)\cdot0 = 1$. $y_2 = 0 - (-1 + e\cdot2)\cdot1 = 1 - e\cdot2$.

Поскольку $r_n = e$, то $x_2' = x_2 / r_n = e$, $y_2' = y_2 / r_n = -2 + e$.
Действительно, $(7 + e\cdot2)\cdot(e) + (3 + e\cdot5)\cdot(-2 + e) = 1$.

Для любых целых дуальных чисел α и β ($\beta \neq 0$ и не является делителем нуля) существуют целые дуальные числа p и q такие, что $\alpha = \beta p + q$, причем $N(q) < N(\beta)$.

Два целых дуальных числа α, β называют взаимно простыми, если их нормы взаимно простые целые числа $(N(\alpha), N(\beta)) = 1$ и существуют целые двойные числа x и y такие, что $\alpha x + \beta y = 1$.

Пример
$(5 + \varepsilon) = 1 \bmod (3 + \varepsilon\cdot2)$,
$(5 + \varepsilon)\cdot x + (3 + \varepsilon\cdot2)\cdot y = 1$.
1) $\dfrac{5+\varepsilon}{3+\varepsilon\cdot2} = 1{,}667 - \varepsilon\cdot0{,}4778$,
$(1{,}167 - q_1')^2 + (1{,}28 - q_1'')^2 \le 0{,}5$,
$q_1 = 2 - \varepsilon$,
$r_0 = 5 + \varepsilon = (3 + \varepsilon\cdot2)\cdot(2 - \varepsilon) + (-1)$;
2) $\dfrac{3+\varepsilon\cdot2}{-1} = -3 - \varepsilon\cdot2$,

$x_0 = 1,$ $y_0 = 0,$
$x_1 = 0,$ $y_1 = 1,$
$x_2 = 1 – (2 – \varepsilon)\cdot 0 = 1.$ $y_2 = 0 – (2 – \varepsilon)\cdot 1 = –2 + \varepsilon.$

Поскольку $r_n = –1$, то $x_3' = x_3 / r_n = –1,$ $y_4' = y_4 / r_n = 2 – \varepsilon.$
Действительно, $(5 + \varepsilon)\cdot(–1) + (3 + \varepsilon\cdot 2)\cdot(2 – \varepsilon) = 1.$

Исходные тексты реализации алгоритма для вещественных чисел на языке C++ приведены ниже.

```
long extended_euclid_real(long a, long b)
{
  long Xs(0), Xs_1(1), Ys(1), Ys_1(0), x(0);
  if (b! = 0)
          real_recurse(a, b, &Xs, &Xs_1, &Ys, &Ys_1, x);
  return x;
}

void real_recurse(long a, long b, long *Xs_1, long *Xs_2, long *Ys_1, long *Ys_2, long &x)
{
 long q, r, Xs, Ys;

 if (b > 0)
 {
          q = a / b;
          r = a - q * b;                          // считаем остаток

          Xs = *Xs_2 - q * (*Xs_1);
          Ys = *Ys_2 - q * (*Ys_1);

      real_recurse (b, r, &Xs, Xs_1, &Ys, Ys_1, x);
 }
       else
       {
          if (a != 1)  x = 0;
          else x = *Xs_2;
       }
}
```

При реализации алгоритма для расширений поля вещественных чисел использован механизм шаблонов (templates) языка C++. Основные операции для комплексных, двойных, дуальных чисел описаны в классах clsComplexNumber, clsDoubleNumber, clsDualNumber, которые наследуются от общего класса clsComplexBase. В данной статье приведем только исходные тексты реализации алгоритма.

```
template <class __Number> __Number template_euclid(__Number a, __Number b)
{
   __Number Xs(0,0), Xs_1(1,0), Ys(1,0), Ys_1(0,0), x(0,0);

   if (b != __Number(0,0))
      template_recurse(a, b, &Xs, &Xs_1, &Ys, &Ys_1, x);

   return x;
}

template <class __Number> void template_recurse(__Number a,
                                    __Number b,
                                    __Number *Xs_1,
                                    __Number *Xs_2,
                                    __Number *Ys_1,
                                    __Number *Ys_2,
                                    __Number &x)
{
   __Number q, r, Xs, Ys;
   long q1,q2;

   if ( b.norm())                    // N(b)   ≠  0      - не делитель нуля
   {
      __Number __q = a  b;

   // (0.5 - q1)^2 + (0.5 - q2)^2 <= 0/5
      long double dQ1 = __q.real() - 0.5;
      long double dQ2 = __q.imag() - 0.5;

      q1 = (dQ1 > (long) dQ1) ? (dQ1 + 1) : dQ1;
      q2 = (dQ2 > (long) dQ2) ? (dQ2 + 1) : dQ2;

      q = __Number(q1,q2);
      r = a - q * b;

      Xs = *Xs_2 - q * (*Xs_1);
      Ys = *Ys_2 - q * (*Ys_1);

      template_recurse (b, r, &Xs, Xs_1, &Ys, Ys_1, x);
   }
   else
   {
      // проверка на единицу в кольце
      if  (!a.isUnit()) x = 0.0;
      else x = (*Xs_2) / a;
   }
}
```

В статье приведены примеры реализации алгоритма Евклида на языке C++ для некоторых гиперкомплексных числовых систем. В дальнейшем этот алгоритм будет реализован в языковой среде Maple.

Это целесообразно, так как в отделе «Специализированных средств моделирования» ИПРИ НАН Украины разработана библиотека процедур для выполнения символьных и численных операций, алгебраических операций, нелинейных операций, моделирования динамических процессов, модульных вычислений и др. для аппарата гиперкомплексных чисел в среде Maple.

При этом возможность генерации программ из Maple в C++ и использование программ C++ как подключаемой процедуры открывает новые перспективы для моделирования в гиперкомплексных числовых системах широкого спектра задач из различных областей науки и техники.

Материалы данной статьи подготовлены и написаны при участии нашего научного руководителя профессора, д.т.н. М.В.Синькова.

1. *Синьков М.В., Бояринова Ю.Е., Калиновский Я.А., Трубников П.В.* Развитие задачи разделения секрета // Реєстрація, зберігання і оброб. даних. — 2003. — Т. 5, № 4. — С. 90–96.

2. *Бояринова Ю.Е., Трубников П.В.* Расширение задачи разделения секрета для случая использования двойных чисел // Реєстрація, зберігання і оброб. даних. — 2004. — Т. 6, №1. — С. 47–52.

3. *Ноден П., Китте К.* Алгебраическая алгоритмика с упражнениями и решениями. — М.: Мир, 1999. — 720 с.

4. *Синьков М.В., Губарени Н.М.* Непозиционные представления в многомерных числовых системах. — К.: Наук. думка, 1979. — 138 с.

Поступила в редакцию 08.09.2004