УДК 681.511:3

А.Н. ТЕРЕЩЕНКО

ОПТИМИЗАЦИЯ МЕТОДА МОНТГОМЕРИ ЗА СЧЕТ ИСПОЛЬЗОВАНИЯ ОДНОСЛОВНЫХ УМНОЖЕНИЙ ПО ОДНОСЛОВНОМУ МОДУЛЮ

Оптимизация вычислений

Рассматривается операция умножения по модулю, от быстродействия которой в основном зависит быстродействие ассиметричной криптографии. Предлагается оптимизация метода Монтгомери нахождения остатка. Показано, что умножение n-разрядных чисел по n-разрядному модулю можно привести к однословным умножениям по однословному модулю.

Введение. При реализации криптографических схем защиты информации, которые основаны на использовании односторонних и односторонних с лазейкой функций, основная вычислительная нагрузка падает на выполнение модулярных арифметических операций над большими числами. В связи с этим последнее время значительное внимание уделяется поиску вычислительно эффективных алгоритмов, которые выполняют операции модулярной арифметики [1 – 3].

Общее описание метода Монтгомери

В 1985 году П. Монтгомери (P. Montgomery) предложил эффективный алгоритм [4] вычисления $R = \langle U \cdot V \rangle_N$, где R, U, V, N – битовые двоичные числа. Этот алгоритм особенно удобен при использовании на универсальных компьютерах (сигнальных процессорах или микропроцессорах), которые способны быстро выполнять арифметические действия по модулю степени двойки. Особенностью алгоритма Монтгомери является то, что этот алгоритм позволяет вычислить остаток R без выполнения деления на N. Благодаря особенному представлению остаточного класса по модулю N алгоритм заменяет операцию деления на N делением на степень двойки, которая легко реализуется линейными сдвигами на компьютере. R и N должны быть взаимно простыми числами.

Опишем основную идею алгоритма Монтгомери вычисления остатка [2]. Пусть дано число $U < N$. Определим его N-остаток, как $\overline{U} = \langle U \cdot M \rangle_N$. Легко показать, что множество $\left\{ \langle I \cdot M \rangle_N \middle| 0 \le I < N-1 \right\}$ – полная система остатков, т. е. содержит все числа между 0 и $N-1$. Поэтому существует взаимно однозначное соответствие между числами из области от 0 до $N-1$ и вышеуказанным множеством. Алгоритм Монтгомери использует это свойство, используя более быструю процедуру умножения для вычисления *N*-остатка умножения двух целых чисел, у которых известны их *N*-остатки. Когда известны *N*-остатки чисел $\overline{U}$ и $\overline{V}$, умножение Монтгомери определяется как *N*-остаток следующим образом:

$$\overline{R} = \left\langle \overline{U} \cdot \overline{V} \cdot M^{-1} \right\rangle_N,$$

где M^{-1} такое, что $\left\langle M \cdot M^{-1} \right\rangle_N = 1$.

Тогда $\overline{R}$ – *N*-остаток произведения $R = \langle U \cdot V \rangle_N$, так как

$$\overline{R} = \left\langle \overline{U} \cdot \overline{V} \cdot M^{-1} \right\rangle_N = \left\langle U \cdot M \cdot V \cdot M \cdot M^{-1} \right\rangle_N = \langle U \cdot V \cdot M \rangle_N.$$

Для описания алгоритма Монтгомери вычисления остатка, нам понадобится величина N^{-1}, для которого выполняется соотношение $M \cdot M^{-1} - N \cdot N^{-1} = 1$. Целые числа M^{-1} и N^{-1} вычисляются расширенным алгоритмом Эвклида.

Пошаговое описание алгоритма Монтгомери вычисления произведения $\overline{R} = \left\langle \overline{U} \cdot \overline{V} \cdot M^{-1} \right\rangle_N$, где $\overline{U}$ и $\overline{V}$ – заданные числа выглядят таким образом:

Алгоритм 1. Функция MonPro($\overline{U}$,$\overline{V}$)

Шаг 1. $T = \overline{U} \cdot \overline{V}$.

Шаг 2. $S = \left\langle T \cdot N^{-1} \right\rangle_M$.

Шаг 3. $\overline{R} = (T + N \cdot S)/M$.

Шаг 4. Если $\overline{R} > N$, то $\overline{R} = \overline{R} - N$.

Важной особенностью алгоритма Монтгомери является то, что $M = 2^{\gamma}$ является степенью двойки, что позволяет заменить операции умножение по модулю M и деление на M операцией быстрого линейного сдвига.

Лема 1. Если числа $\overline{U}$ и $\overline{V}$ n-разрядные, то количество операций однословного умножения в алгоритме 1 можно выразить следующей априорной оценкой: $C^*(n) = 3n^2$.

Доказательство. В алгоритме 1 шаги 1, 2 и 3 содержат операцию умножения, каждая из которых имеет сложность n^2.

Примечание 1. Сложность n^2 имеет диагональный метод, который является эффективным при умножении больших чисел, длина которых не превышает $n < 90$ машинных слов.

Предлагается в методе Монтгомери использовать два дополнительных числа M и M' вместо одного: $N < M < M'$. Введение второго дополнительного числа M' дает возможность использовать китайскую теорему об остатках и, как результат, привести s-словную операцию умножения по модулю к однословным операциям умножения по модулю.

Теорема 1 [2]. Сравнения $\langle U \rangle_{m_i} = u_i$ имеют единственное решение среди классов остатков по модулю $M = \prod_{i=0}^{n-1} m_i$ такому, что $(m_j, m_i) = 1$, $i \neq j$.

Доказательство. Сначала построим сложный модуль $M = \prod_{i=0}^{n-1} m_i$.

Потом запишем величины $M_0 = M/m_0$, $M_1 = M/m_1$, ..., $M_{n-1} = M/m_{n-1}$, где $(M_i, m_i) = 1$, $i = \overline{0, n-1}$ взаимно простые между собой.

Поэтому можно найти число h_i, развязывая равенство $\langle h_i \cdot M_i \rangle_{m_i} = 1$ [4].

Теперь рассмотрим величину

$$U = \langle u_0 \cdot h_0 \cdot M_0 + u_1 \cdot h_1 \cdot M_1 + \ldots + u_{n-1} \cdot h_{n-1} \cdot M_{n-1} \rangle_M . \qquad (1)$$

Найдем остаток U по модулю m_0. Так как каждое из чисел M_1, M_2,..., M_{n-1} делится на m_0, то $\langle U \rangle_{m_0} = u_0$. Аналогично $\langle U \rangle_{m_i} = u_i$.

Поэтому U – решение заданной системы равенств. Легко убедиться, что решение (1) – единственное (как класс остатков по модулю M).

Лемма 2. Если модуль $M = \prod_{i=0}^{n-1} m_i$ такой, что $(m_j, m_i) = 1$, $i \neq j$, и известны числа h_i такие, что $\langle h_i \cdot M_i \rangle_{m_i} = 1$, $M_i = M/m_i$, $(M_i, m_i) = 1$, $i = \overline{0, n-1}$, то для любого числа $U < M$ справедлива следующая оценка:

$$\sum_{i=0}^{n-1} \left(\langle u_i \cdot h_i \rangle_{m_i} / m_i \right) < n , \qquad (2)$$

где $\langle U \rangle_{m_i} = u_i$, $i = \overline{0, n-1}$.

Доказательство. Очевидно, что $\langle u_i \cdot h_i \rangle_{m_i} < m_i$, $i = \overline{0, n-1}$. Умножая обе части на M_i, получаем следующее выражение:

$$\left(\langle u_i \cdot h_i \rangle_{m_i} \cdot M_i \right) < M , \; i = \overline{0, n-1} .$$

Откуда вытекает, что сумма по i будет меньше, чем $n \cdot M$

$$\sum_{i=0}^{n-1}\left(\left\langle u_i \cdot h_i \right\rangle_{m_i} \cdot M_i\right) < n \cdot M .$$

Разделив обе части на M, получим (2), что и требовалось доказать.

Задача восстановления большого числа U по его остаткам u_i, $i = \overline{0, n-1}$ имеет квадратичную сложность $O(n^2)$, где n-разрядность числа U. Лемма 2 показывает, что при восстановлении числа U по его остаткам может превышать модуль M, и что задача проверки превышения модуля M числом U имеет линейную сложность.

Из соотношения (2) видно, что если u_i, h_i, m_i не превышают машинное слово, то необходимо n однословных операций умножения и $2n$ однословных операций деления. С учетом того, что деление происходит нацело, в соотношение (2) нужно учесть выравнивающий коэффициент 2^{ω}.

$$\left\lfloor \frac{1}{2^{\omega}} \sum_{i=0}^{n-1} \left\lfloor \frac{\left\langle u_i h_i \right\rangle_{m_i} \cdot 2^{\omega}}{m_i} \right\rfloor \right\rfloor < n,$$

где ω – количество бит в выравнивающем коэффициенте.

Операции с выравнивающим коэффициентом 2^{ω} являются быстрыми, так как они реализуются линейными сдвигами.

Следующая теорема показывает, что перевод числа из одной системы остатков в другую можно выполнить без восстановления числа (при этом учитывается ранее доказанный критерий проверки (лемма 2)).

Теорема 2. Пусть заданы две системы в остатках с модулями $M = \prod_{i=0}^{n-1} m_i$, $M' = \prod_{i=0}^{n-1} m'_i$ такими, что $(m_j, m_i) = 1$, $(m'_j, m'_i) = 1$, $i \neq j$, и число $U < M$ имеет следующие остатки: $\left\langle U \right\rangle_{m_i} = u_i$, $i = \overline{0, n-1}$. Тогда сложность представления числа U в системе в остатках с модулем M', где $M' > M$, можно выразить следующими априорными оценками сложности без учета предвычислений:

$$C^{*}(n) = n^2 + 2n,\ C^{\div}(n) = n,\ C^{\langle\rangle}(n) = 2n,$$

где $C^{*}(n), C^{\div}(n)$ – количество однословных операций умножения и деления, $C^{\langle\rangle}(n)$ – число операций взятия по однословному модулю.

Доказательство. Модули M и M' составляют n однословных модулей. Вычислим некоторые величины, которые понадобятся далее для представления числа U в системе остатков m'_i, $i = \overline{0, n-1}$.

Матрица перехода W' содержит $w'_{ij} = \langle M_j \rangle_{m'_i}$, $M_i = M/m_i$, $i, j = \overline{0, n-1}$ и известно представление модуля M в остатках $m''_i = \langle M \rangle_{m'_i}$, $i = \overline{0, n-1}$.

Определим вектор F следующим образом:

$$f_i = \langle u_i h_i \rangle_{m_i}, \ \langle h_i \cdot M_i \rangle_{m_i} = 1, \ i = \overline{0, n-1}. \tag{3}$$

При восстановлении числа U по его остаткам, используя формулу (1), выражение (до взятия по модулю M) может превышать M ($\langle U \rangle_M < M$). Коэффициент превышения будет следующим:

$$k = \left\lfloor \frac{1}{2^{\omega}} \sum_{i=0}^{n-1} \left\lfloor \frac{f_i \cdot 2^{\omega}}{m_i} \right\rfloor \right\rfloor. \tag{4}$$

Очевидно, что k находится в диапазоне $0 \le k < n$.

С учетом коэффициента превышения k n-разрядное число U в системе остатков m'_i можно представить в виде:

$$u'_i = \left\langle \left\langle \sum_{j=0}^{n-1} (f_j w'_{ij}) \right\rangle_{m'_j} - \langle k m'' \rangle_{m'_j} \right\rangle_{m'_i} = \left\langle \left(\sum_{j=0}^{n-1} (f_j w'_{ij}) \right) - k m'' \right\rangle_{m'_i}, \ i, j = \overline{0, n-1}. \tag{5}$$

С учетом соотношений (3)–(5) находим, что число однословных умножений равно $n^2 + 2n$, число однословных делений равно n и необходимо выполнить $2n$ операций по однословному модулю, что и требовалось доказать.

Считаем, что использование выравнивающего коэффициента 2^{ω} в формуле (4) не увеличивает сложность, так как операции с этим коэффициентом реализуются линейными сдвигами. Выбор ω зависит от количества простых множителей n и, возможно, от величин самих простых множителей.

Заметим, что существуют методы, которые позволяют выполнить перевод менее чем за $n^2 + 2n$ операций однословного умножения.

Оптимизации метода Монтгомери за счет использования китайской теоремы об остатках. Предложенный метод реализует операцию $R = \langle U \cdot V \rangle_N$ умножения двух n-разрядных чисел U и V по n-разрядному модулю N.

Использование двух систем классов связано с тем, что алгоритм 1 на шагах 2 и 3 содержит две медленные операции в обычной арифметике – умножение по многоразрядному модулю M и деление на многоразрядное M нацело. Нахождение остатка по многоразрядному модулю M происходит в той же системе классов M. При этом остатком по сложному модулю M являются остатки в системе классов m_i, которые вычисляются автоматически после каждой операции в этой системе классов. Операцию деления на M в шаге 3 можно заменить операцией умножения в системе классов M'.

Многоразрядные числа M и M' не являются степенью двойки.
Необходимо выполнить следующие предвычисления.

С помощью расширенного алгоритма Евклида находим числа M^{-1} и N^{-1} такие, что $M \cdot M^{-1} - N \cdot N^{-1} = 1$.

Представляем N и N^{-1} в остатках $n_i = \langle N \rangle_{m_i}$, $n_i^{-1} = \langle N^{-1} \rangle_{m_i'}$, $n_i' = \langle N \rangle_{m_i'}$.

Находим числа h_i, h_i' такие, что $\langle h_i \cdot M_i \rangle_{m_i} = 1$, $M_i = M/m_i$, $(M_i, m_i) = 1$, $\langle h_i' \cdot M_i' \rangle_{m_i'} = 1$, $M_i' = M'/m_i'$, $(M_i', m_i') = 1$, $i = \overline{0, n-1}$.

Вычисляем матрицу W', где $w_{ij}' = \langle M_j \rangle_{m_i'}$, $i, j = \overline{0, n-1}$, используемую для перевода промежуточного результата из системы классов m_i в систему классов m_i'. Также находим $m_i'' = \langle M \rangle_{m_i'}$, $i = \overline{0, n-1}$.

Аналогично вычисляется матрица W, где $w_{ij} = \langle M_j' \rangle_{m_i}$, $i, j = \overline{0, n-1}$, используемая для обратного перевода промежуточного результата из системы классов m_i' в систему классов m_i. Для этого также необходимо найти $m_i''' = \langle M' \rangle_{m_i}$, $i = \overline{0, n-1}$, представляющий модуль M' в системе остатков m_i.

Операцию деления на число M в системе классов m_i' заменим умножением числа $M^{\div}$ такого, что $m_i'^{\div} = \langle M^{\div} \rangle_{m_i'}$, $\langle m_i'^{\div} \cdot m_i'' \rangle_{m_i'} = 1$, $m_i'' = \langle M \rangle_{m_i'}$, $i = \overline{0, n-1}$.

Найдем N -остатки чисел U и V

$$\overline{U} = \langle U \cdot M \rangle_N, \ \overline{V} = \langle V \cdot M \rangle_N;$$

$$u_i = \langle \overline{U} \rangle_{m_i}, \ u_i' = \langle \overline{U} \rangle_{m_i'}, \ v_i = \langle \overline{V} \rangle_{m_i}, \ v_i' = \langle \overline{V} \rangle_{m_i'}, \ i = \overline{0, n-1}.$$

Алгоритм 2. Одновременное вычисление в двух системах остатков.

Шаг 1. $t_i = \langle u_i v_i \rangle_{m_i}$, $t_i' = \langle u_i' v_i' \rangle_{m_i'}$, $i = \overline{0, n-1}$. $\Leftrightarrow \langle T = \overline{U} \cdot \overline{V} \rangle_{m_i}$.

Шаг 2. $s_i = \langle t_i n_i^{-1} \rangle_{m_i}$, $i = \overline{0, n-1}$. $\Leftrightarrow \langle S = T \cdot N^{-1} \rangle_{m_i}$.

Шаг 3. Переход в систему остатков M'.

Шаг 3, а. $f_i = \langle s_i h_i \rangle_{m_i}$, $i = \overline{0, n-1}$;

Шаг 3, б. $k = \left\lfloor \frac{1}{2^{\omega}} \sum_{i=0}^{n-1} \left\lfloor \frac{f_i \cdot 2^{\omega}}{m_i} \right\rfloor \right\rfloor$;

Шаг 3, в. $s_i' = \left\langle \sum_{j=0}^{n-1} w_{ij}' f_j \right\rangle_{m_i'}$, $s_i' = \langle s_i' - k m_i'' \rangle_{m_i'}$, $i = \overline{0, n-1}$. $\Leftrightarrow \langle S = T \cdot N^{-1} \rangle_{m_i'}$.

Шаг 4. $t_i' = \left\langle t_i' + n_i' s_i' \right\rangle_{m_i'}$, $i = \overline{0, n-1}$. $\Leftrightarrow \left\langle T = T + N \cdot S \right\rangle_{m_i'}$.

Шаг 5. $t_i' = \left\langle t_i' m_i'^{\div} \right\rangle_{m_i'}$. $\Leftrightarrow \left\langle T = (T + N \cdot S)/M \right\rangle_{m_i'}$.

Шаг 6. Возврат в исходную систему остатков M.

Шаг 6, а. $f_i' = \left\langle t_i' h_i' \right\rangle_{m_i'}, i = \overline{0, n-1}$;

Шаг 6, б. $k' = \left\lfloor \frac{1}{2^{\omega}} \sum_{i=0}^{n-1} \left\lfloor \frac{f_i' \cdot 2^{\omega}}{m_i'} \right\rfloor \right\rfloor$;

Шаг 6, в. $t_i = \left\langle \sum_{j=0}^{n-1} w_{ij} f_j' \right\rangle_{m_i}$, $t_i = \left\langle t_i - k' m_i''' \right\rangle_{m_i}, i = \overline{0, n-1}$.

$\Leftrightarrow \left\langle T = (T + N \cdot S)/M \right\rangle_{m_i}$.

Алгоритм 2 можно описать так. На шаге 1 вычисления происходят в двух системах сложных остатков M и M'. Шаг 2 выполняется только в классе остатков M. На этом шаге находится остаток по модулю M. Далее на шаге 3 результат переводится в систему классов M', чтобы выполнить вычисление только в системе классов M' на шагах 4 и 5, которые соответствуют шагу 3 алгоритма 1. На шаге 4 вычисляется числитель $T + N \cdot S$, причем операция $t_i = \left\langle t_i + n_i s_i \right\rangle_{m_i}$ отсутствует, так как в системе классов M числитель $T + N \cdot S$ делится нацело на M. Соответственно, $\left\langle T + N \cdot S \right\rangle_{m_i} = 0$. Деление на M нацело выполняется на шаге 5. Далее результат шага 5 переводится обратно в систему остатков M, после чего получаем представление результата в двух системах классов M и M'.

После шага 6 оба числа T и T' содержат $\overline{R}$ как результат умножения чисел $\overline{U}$ и $\overline{V}$ в системах классов M и M':

$$\overline{R} = \left\langle \overline{U} \cdot \overline{V} \cdot M^{-1} \right\rangle_N . \tag{6}$$

Для восстановления результата умножения чисел U и V по модулю N необходимо число $\overline{R}$ из формулы (6) умножить на M^{-1} по модулю N.

$$R = \left\langle \overline{R} \cdot M^{-1} \right\rangle_N = \left\langle U \cdot V \right\rangle_N , \tag{7}$$

$\left\langle \overline{R} \cdot M^{-1} \right\rangle_N = \langle 15813 \cdot 18169 \rangle_{34321} = 5306$, $\left\langle U \cdot V \right\rangle_N = \langle 13100 \cdot 2919 \rangle_{34321} = 5306$, что равносильно повторному выполнению шагов начиная со 2-го по 6-ой с последующим восстановлением R из t_i.

После шага 6 результаты в остатках классов M и M' могут быть использованы без дополнительных затрат для следующей итерации возведения в степень по модулю.

Теорема 3. Если числа $\overline{U}$ и $\overline{V}$ n-разрядные, то количество операций в алгоритме 2 можно выразить следующими априорными оценками:

$$C^{*}(n) = 2n^2 + 9n,\ C^{\div}(n) = 2n,\ C^{\langle\ \rangle}(n) = 9n,$$

где $C^{*}(n), C^{\div}(n)$ – количество однословных операций умножения и деления, $C^{\langle\ \rangle}(n)$ – число операций взятия по однословному модулю.

Доказательство. В алгоритме 2 на шаге 2 происходит переход в систему классов M', а на шаге 6 переход обратно в систему остатков M. Согласно теореме 2 каждый переход можно выразить следующими априорными оценками сложности: $C^{*}(n) = n^2 + 2n,\ C^{\div}(n) = n,\ C^{\langle\ \rangle}(n) = 2n$.

Шаги 1, 2, 4, 5 содержат дополнительно 5 однословных умножений по однословному модулю. Таким образом, общая оценка сложности составляет $C^{*}(n) = 2n^2 + 9n,\ C^{\div}(n) = 2n,\ C^{\langle\ \rangle}(n) = 9n$. Что и требовалось доказать.

Для предложенного алгоритма необходимо большее число предвычислений и подбор вначале $2n$ взаимно простых однословных модулей, которые составляют сложные модули M и M'. В конечном итоге, затраты на предвычисления окупаются, если использовать алгоритм для возведения в степень по модулю.

Приведем пошаговый пример умножения $U = 13100$ и $V = 2919$ по модулю $N = 34321$ в таблице, где $M = 36465$, $M' = 392863$, $N < M < M'$.

ТАБЛИЦА

$M = 36465 = 11 \cdot 13 \cdot 15 \cdot 17$ $(m_0, m_1, m_2, m_3) = (11,13,15,17)$ $(M_0, M_1, M_2, M_3) = (3315,2805,2431,2145)$ $(h_0, h_1, h_2, h_3) = (3,4,1,6), \langle h_i \cdot M_i \rangle_{m_i} = 1$	$M' = 392863 = 19 \cdot 23 \cdot 29 \cdot 31$ $(m'_0, m'_1, m'_2, m'_3) = (19,23,29,31)$ $(M'_0, M'_1, M'_2, M'_3) = (20677,17081,13547,1267$ $(h'_0, h'_1, h'_2, h'_3) = (4,20,22,5),$ $\langle h'_i \cdot M'_i \rangle_{m'_i} = 1$
$M \cdot M^{-1} - N \cdot N^{-1} = 36465 \cdot 18196 - 19304 \cdot 34321 = 1$	
$(n_0^{-1}, n_1^{-1}, n_2^{-1}, n_3^{-1}) = (10,12,14,9),$ $n_i^{-1} = \langle N^{-1} \rangle_{m_i}$	$(n'_0, n'_1, n'_2, n'_3) = (7,5,14,4),\ n'_i = \langle N \rangle_{m'_i}$
$\overline{U} = \langle U \cdot M \rangle_N = 11822,\ \overline{V} = \langle V \cdot M \rangle_N = 11914$	
$(u_0, u_1, u_2, u_3) = (8,5,2,7),\ u_i = \langle \overline{U} \rangle_{m_i},$ $(v_0, v_1, v_2, v_3) = (1,6,4,14),\ v_i = \langle \overline{V} \rangle_{m_i}$	$(u'_0, u'_1, u'_2, u_3) = (4,0,19,11),\ u'_i = \langle \overline{U} \rangle_{m'_i}$ $(v'_0, v'_1, v'_2, v'_3) = (1,0,24,10),\ v'_i = \langle \overline{V} \rangle_{m'_i}$
$W = \begin{bmatrix} 8 & 9 & 6 & 1 \\ 7 & 12 & 1 & 11 \\ 7 & 11 & 2 & 13 \\ 5 & 13 & 15 & 8 \end{bmatrix},\ w_{ij} = \langle M'_j \rangle_{m_i}$	$W' = \begin{bmatrix} 9 & 12 & 18 & 17 \\ 3 & 22 & 16 & 6 \\ 9 & 21 & 24 & 28 \\ 29 & 15 & 13 & 6 \end{bmatrix},\ w'_{ij} = \langle M_j \rangle_{m'_i}$

Окончание таблицы

$(m_0''', m_1''', m_2''', m_3''') = (9,3,13,10)$, $m_i''' = \langle M' \rangle_{m_i}$	$(m_0'', m_1'', m_2'', m_3'') = (4,10,12,9)$, $m_i'' = \langle M \rangle_{m_i'}$
$M^{\div} = 8315013$, $\langle M \cdot M^{\div} \rangle_{M'} = 1$, $(m_0'^{\div}, m_1'^{\div}, m_2'^{\div}, m_3'^{\div}) = (5,7,17,7)$, $m_i'^{\div} = \langle M^{\div} \rangle_{m_i'}$	
Шаг 1. $(t_0, t_1, t_2, t_3) = (8,4,8,13)$, $t_i = \langle u_i \cdot v_i \rangle_{m_i}$, $(t_0', t_1', t_2', t_3') = (4,0,21,17)$, $t_i' = \langle u_i' \cdot v_i' \rangle_{m_i'}$	
Шаг 2. $(s_0, s_1, s_2, s_3) = (3,9,7,15)$, $s_i = \langle t_i \cdot n_i^{-1} \rangle_{m_i}$	
Шаг 3, а. $(f_0, f_1, f_2, f_3) = (9,10,7,5)$, $f_i = \langle s_i \cdot h_i \rangle_{m_i}$	
Шаг 3, б. $k = \left\lfloor \frac{1}{2^{12}} \left(\begin{matrix} \lfloor 3351.272727 \rfloor + \lfloor 3150.769231 \rfloor + \\ + \lfloor 1911.466667 \rfloor + \lfloor 1204.705882 \rfloor \end{matrix} \right) \right\rfloor = \left\lfloor \frac{1}{2^{12}} \sum_{i=0}^{3} \frac{f_i \cdot 2^{12}}{m_i} \right\rfloor$ $k = 2 = \lfloor 2.34765625 \rfloor = \lfloor 9616 \div 4096 \rfloor$	
Шаг 3, в. $(s_0', s_1', s_2', s_3') = (13,21,19,5)$, $s_i' = \left\langle \sum_{j=0}^{3} w_{ij}' f_j \right\rangle_{m_i'}$ $(s_0', s_1', s_2', s_3') = (13,21,19,5) - 2(4,10,12,9) = (5,1,-5,-13) = (5,1,24,18)$ $(s_0', s_1', s_2', s_3') = (s_0', s_1', s_2', s_3') - k(m_0'', m_1'', m_2'', m_3'')$, $s_i' = \langle s_i' - k m_i'' \rangle_{m_i'}$	
Шаг 4. $(t_0', t_1', t_2', t_3') = (1,5,9,27)$, $t_i' = \langle t_i' + n_i' s_i' \rangle_{m_i'}$	
Шаг 5. $(t_0', t_1', t_2', t_3') = (5,12,8,3)$, $t_i' = \langle t_i' m_i'^{\div} \rangle_{m_i'}$	
Шаг 6, а. $(f_0', f_1', f_2', f_3') = (1,10,2,15)$, $f_i' = \langle t_i' h_i' \rangle_{m_i'}$	
Шаг 6, б. $k' = \left\lfloor \frac{1}{2^{12}} \left(\begin{matrix} \lfloor 215.5789474 \rfloor + \lfloor 1780.869565 \rfloor + \\ + \lfloor 282.4827586 \rfloor + \lfloor 1981.935484 \rfloor \end{matrix} \right) \right\rfloor = \left\lfloor \frac{1}{2^{12}} \sum_{i=0}^{3} \left\lfloor \frac{f_i' \cdot 2^{12}}{m_i'} \right\rfloor \right\rfloor$ $k' = 1 = \lfloor 1.039550781 \rfloor = \lfloor 4258 \div 4096 \rfloor$	
Шаг 6, в. $(t_0, t_1, t_2, t_3) = (4,8,1,13)$, $t_i = \left\langle \sum_{j=0}^{3} w_{ij} f_j' \right\rangle_{m_i}$ $(t_0, t_1, t_2, t_3) = (4,8,1,13) - 1(9,3,13,10) = (-5,5,-12,-3) = (6,5,3,3)$ $(t_0, t_1, t_2, t_3) = (t_0, t_1, t_2, t_3) - k'(m_0''', m_1''', m_2''', m_3''')$, $t_i = \langle t_i - k' m_i''' \rangle_{m_i}$	

Полученные значения $(t_0, t_1, t_2, t_3) = (6,5,3,3)$ представляют значение $\overline{R}$ в остатках m_i. При восстановлении $(6,5,3,3)$ дает 15813. Тот же результат можно получить, воспользовавшись формулой (6)

$$\overline{R} = \left\langle \overline{U} \cdot \overline{V} \cdot M^{-1} \right\rangle_N = \langle 11822 \cdot 11914 \cdot 18169 \rangle_{34321} = 15813.$$

Хотя пример содержит 4-х словные модули M и M', алгоритм 2 работает для более длинных модулей.

Число одноословных операций умножения в методе Монтгомери (лемма 1) и в предложенном методе (теорема 3) составляет $3n^2 + C_1$ и $2n^2 + C_2$ соответственно, где C_1 и C_2 – коэффициенты с линейной сложностью. Таким образом, предложенный метод на каждой итерации имеет на треть меньше операций однословных умножений без учета пред- и поствычислений.

Заключение. В данной работе описана оптимизация метода Монтгомери вычисления остатка умножения двух n-разрядных чисел по n-разрядному модулю за счет одновременного вычисления в двух системах сложных остатков M и M', которые сводятся к однословным умножениям по однословному модулю. Показан переход из одной системы остатков в другую систему остатков без промежуточного восстановления числа из остатков. Приведены пример в виде таблицы и пошаговый алгоритм вычисления остатка умножения двух n-разрядных чисел по n-разрядному модулю с использованием предложенного метода. Показано, что предложенный метод на каждой итерации имеет на одну треть меньше операций однословного умножения, чем метод Монтгомери.

А.М. Терещенко

ОПТИМІЗАЦІЯ МЕТОДУ МОНТГОМЕРІ ЗА РАХУНОК ВИКОРИСТАННЯ ОДНОСЛІВНИХ МНОЖЕНЬ ЗА ОДНОСЛІВНИМ МОДУЛЕМ

Розглядається оптимізація множення за модулем, від часу виконання якої в основному залежить швидкодія асиметричної криптографії. Пропонується оптимізація методу Монтгомері обчислення лишку. Показано, що множення *n*-розрядних чисел за *n*-розрядним модулем можна привести до однослівних множень за однослівним модулем.

A.N. Tereshchenko

OPTIMISATION OF MULTIPRECISION MONTGOMERY'S METHOD AT THE EXPENSE OF USING A SINGLE-PRECISION MULTIPLICATION MODULO SINGLE-PRECISION INTEGER

A modular multiplication that has a considerable influence on asymmetric cryptography performance is considered. An optimization of Montgomery's method of modulo calculation is proposed. It is shown that *n*-dimensional multiplication modulo *n* can yield a single precision multiplication by single precision module.

1. *Кнут Д.* Искусство программирования для ЭВМ: Пер. с англ. / Под ред. Бабенко. – М: Мир, 1977. – Т. 2. – 734 с.
2. *Задірака В., Олексюк О.* Комп'ютерна арифметика багаторозрядних чисел. – К., Наук. видання, 2003. – 263 с.
3. *Макклеллан Дж.Х., Рейдер Ч.М.* Применение теории чисел в цифровой обработке сигналов. – М.: Радио и связь, 1983. – С. 17–21.
4. *Montgomery P.L.* Modular Multiplication Without Trial Division // Math. Comp. – 1985. – Т. 44, N 170. – P. 519 – 521.

Получено 18.12.2009

Об авторе:

Терещенко Андрей Николаевич,
аспирант Института кибернетики имени В.М. Глушкова НАН Украины.
teramidi@ukr.net