УДК 681.301

А.А. Коляда, А.Ф. Чернявский

УМНОЖЕНИЕ ПО БОЛЬШИМ МОДУЛЯМ С ИСПОЛЬЗОВАНИЕМ МИНИМАЛЬНО ИЗБЫТОЧНОЙ МОДУЛЯРНОЙ СХЕМЫ МОНТГОМЕРИ

Предлагается новый быстрый алгоритм умножения по большому модулю p, реализующий минимально избыточную модулярную схему Монтгомери. Главной отличительной особенностью разработанной схемы является использование интервально-индексных характеристик и интервально-модулярной формы чисел в базовых процедурах расширения кода. Достигаемая за счет этого оптимизация синтезированного мультипликативного алгоритма обеспечивает (3,5-3,6)-кратное повышение производительности в сравнении с наиболее близким лучшим аналогом при выполнении на однопроцессорной ЭВМ. При этом необходимый объем табличной памяти в случае 1024- и 2462-битовых p не превышает соответственно 1,2 и 6,46 Гб. Если пороговые значения размера памяти таблиц для указанных p составляют 141 и 334 Мб, то получаемый выигрыш в быстродействии является двукратным.

Введение

В настоящее время арифметика модулярных систем счисления (МСС) – модулярная арифметика (МА) широко применяется в системах параллельной обработки для решения задач, требующих быстрых точных вычислений. Спектр таких задач охватывает, в частности, процедуры цифровой обработки сигналов, системы электронной цифровой подписи, базовые преобразования криптосистем с открытым ключом на основе схем RSA, Рабина и так далее [1−15].

Внутренний параллелизм модулярных вычислительных структур (МВС) обеспечивает им ряд существенных преимуществ над позиционными структурами при проведении расчетов в диапазонах больших чисел (ДБЧ). К таким преимуществам относятся:

- независимость времен выполнения модульных операций от числа оснований, а значит и длины кода МСС;
- приспособленность алгоритмов МА к конвейерному и табличному методам вычислений;
- простота организации на базе инструментальных платформ позиционного типа многомашинного и мультипроцессорного режимов обработки данных;
- гибкость табличного механизма реконфигурации МВС.

Исследования последних 10-15 лет по проблематике применения модулярной вычислительной технологии (МВТ) на ДБЧ в системах защиты информации (СЗИ) фактически базируются на стратегии, которая нацелена на реализацию как перечисленных выше, так и ряда других ключевых достоинств МВС в максимальной мере. Подавляющее большинство публикаций по криптографическим приложениям МВТ посвящено созданию высокоскоростных средств для выполнения операций умножения и возведения в степень по большим модулям, главным образом мультипликативных процедур, основанных на схеме Монтгомери [5−15]. В данной схеме используется операция деления нацело, а не операция общего деления. Поэтому модулярные версии алгоритма Монтгомери отличаются высокой производительностью. Вместе с тем, применение в известных МА-умножителях по модулю на ДБЧ оснований разрядностью 32 бита затрудняет внедрение табличного метода, что ведет к снижению быстродействия.

Эффективную альтернативу по отношению к разработанным мультипликативным МА-процедурам составляет класс процедур табличного типа, которые базируются на МСС с основаниями, допускающими широкое применение таблиц, как на аппаратном, так и программном уровнях.

Наиболее трудоемкую часть МА-алгоритмов Монтгомери составляют операции расширения модулярного кода (МК). Оптимизация процедур выполнения данных операций является важнейшим направлением развития МВТ на ДБЧ для криптографических приложений. Эффективной основой для решения обозначенной оптимизационной проблемы могут служить минимально избыточные МСС (МИМСС) [16].

Представляемая в настоящей статье разработка нацелена на реализацию ключевых преимуществ табличной версии компьютерной арифметики МИМСС – минимально избыточной МА (МИМА) на ДБЧ в части оптимизации базовой немодульной процедуры для алгоритмов умножения по модулю на основе схемы Монтгомери.

1. Компьютерно-арифметическая база модулярных мультипликативных процедур на основе схемы Монтгомери

На множестве **Z** целых чисел (ЦЧ) МСС определяется с помощью попарно простых оснований – натуральных модулей $m_1, m_2, \ldots, m_k$ ($k \geq 2$). Число $X \in \mathbf{Z}$ в данной МСС представляется набором $(\chi_1, \chi_2, \ldots, \chi_k)$ остатков $\chi_i = |X|_{m_i}$ от деления X на m_i ($i = \overline{1, k}$). Через $|a|_m$ будем обозначать элемент множества $\mathbf{Z}_m = \{0, 1, \ldots, m-1\}$, сравнимый с a (в общем случае рациональной величиной) по модулю $m>1$. В МСС с основаниями $m_1, m_2, \ldots, m_k$ может быть закодировано не более $M_k = \prod_{i=1}^{k} m_i$ ЦЧ. Обычно в качестве диапазона МСС используют множества $\mathbf{Z}_{M_k} = \{0, 1, \ldots, M_k - 1\}$ и $\mathbf{Z}^{-}_{M_k} = \{-\lfloor M_k/2 \rfloor, -\lfloor M_k/2 \rfloor + 1, \ldots, \lceil M_k/2 \rceil - 1\}$ (обозначения $\lfloor a \rfloor$ и $\lceil a \rceil$ употребляются для ближайших к a соответственно слева и справа ЦЧ).

Компьютерная реализация отображения $\Phi_{\text{МСС}}$: $\mathbf{Z}_{m_1} \times \mathbf{Z}_{m_2} \times \ldots \times \mathbf{Z}_{m_k} \to \mathbf{D}$ ($\mathbf{D} = \mathbf{Z}_{M_k}$ или $\mathbf{D} = \mathbf{Z}^{-}_{M_k}$), ставящее в соответствие МК $(\chi_1, \chi_2, \ldots, \chi_k)$ элемент X диапазона **D**, может быть осуществлена [16] с помощью соотношения

$$X = \sum_{i=1}^{k-1} M_{i,k-1} \left| M_{i,k-1}^{-1} \chi_i \right|_{m_i} + M_{k-1} I(X), \tag{1}$$

где $M_{i,\,k-1} = M_{k-1} / m_i$; $M_{k-1} = \prod_{j=1}^{k-1} m_j$; $I_k(X)$ – интегральная характеристика МК (ИХМК), называемая интервальным индексом (ИИ) числа X относительно модулей $m_1, m_2, \ldots, m_k$. Выражение (1) называется интервально-модулярной формой (ИМФ) ЦЧ X.

Справедливо [16; 17] следующее утверждение

Теорема 1. *Для ИИ* $I_l(X)$ ЦЧ $X \in \mathbf{Z}_{M_l}$ *в МСС с попарно простыми основаниями* $m_1, m_2, \ldots, m_{l-1}$, $m_l \geq l-2$ ($l \geq 2$) *имеет место формула*

$$I_l(X) = \hat{I}_l(X) - m_l \Theta_l(X), \tag{2}$$

где

$$\hat{I}_l(X) = |I_l(X)|_{m_l} = \left| \sum_{i=1}^{l} R_{i,l}(\chi_i) \right|_{m_l}; \tag{3}$$

$$R_{i,l}(\chi_i) = \left\lfloor \frac{m_l}{m_i} \left| M_{i,l}^{-1} \chi_i \right|_{m_i} \right\rfloor = \left| -m_i^{-1} \left| M_{i,l-1}^{-1} \chi_i \right|_{m_i} \right|_{m_l} \ (i \neq l), \quad R_{l,l}(\chi_l) = \left| \frac{\chi_l}{M_{l-1}} \right|_{m_l}; \tag{4}$$

$\Theta_l(X)$ – *двузначная величина, принимающая значения* 0 *или* 1.

Величина $\Theta_l(X)$ называется минимальной ИХМК ($l-1$)-го порядка, отвечающая ЦЧ X ($l = \overline{2, k}$).

Как следует из теоремы 1, в классической (неизбыточной) МСС вычисление интервально-индексной характеристики $I(X) = I_k(X)$ требует применения общего алгоритма формирования ИХМК [16; 17], который является довольно трудоемким. Арифметические свойства МСС удается значительно улучшить за счет избыточного кодирования элементов рабочего диапазона. Предложенное в [16] так называемое минимально-избыточное модулярное кодирование $\Phi_{\text{МИМСС}}$: $\mathbf{Z}_{m_1} \times \mathbf{Z}_{m_2} \times \ldots \times \mathbf{Z}_{m_k} \to \mathbf{D}$ предусматривает использование диапазона **D** с мощностью $|\mathbf{D}| < M_k$. Сущность реализуемого принципа раскрывает нижеследующее утверждение.

Теорема 2. *Для того, чтобы в МСС с попарно простыми основаниями* $m_1, m_2, \ldots, m_k$ *ИИ* $I(X) = I_k(X)$ *каждого элемента* X *диапазона* $\mathbf{D} = \mathbf{Z}^{-}_{2M} = \{-M, -M+1, \ldots, M-1\}$ ($M = m_0 M_{k-1}$; $M_{k-1} = \prod_{i=1}^{k-1} m_i$; m_0 – *вспомогательный модуль*) *однозначно определялся компьютерным ИИ – выче-*

том $\hat{I}_k(X) = \left|I(X)\right|_{m_k}$, *необходимо и достаточно, чтобы k-е основание МСС удовлетворяло условию* $m_k \geq 2m_0 + \rho_{k-1,\max}$ ($m_0 \geq \rho_{k-1,\max}$), *где* $\rho_{k-1,\max} = \max\limits_{Y \in \mathbf{Z}}\{\rho_{k-1}(Y)\}$, $\rho_{k-1}(Y)$ – *ранговая характеристика* ($k-1$)-*го порядка, определяемая равенством*

$$\left|Y\right|_{M_{k-1}} = \sum_{i=1}^{k-1} M_{i,k-1}\left|M_{i,k-1}^{-1}Y\right|_{m_i} - M_{k-1}\rho_{k-1}(Y). \tag{5}$$

При этом для I(X) верна формула

$$I(X) = \begin{cases} \hat{I}_k(X), & \text{если } \hat{I}_k(X) < m_0, \\ \hat{I}_k(X) - m_k, & \text{если } \hat{I}_k(X) \geq m_k - m_0 - \rho_{k-1,\max}. \end{cases} \tag{6}$$

Компьютерный ИИ $\hat{I}_k(X)$ *вычисляется согласно* (3), (4) *при* $l=k$.

Из (5) после деления на M_{k-1} и последующего перехода в обеих частях полученного равенства к антье следует, что

$$\rho_{k-1}(Y) = \left\lfloor \sum_{i=1}^{k-1} m_i^{-1}\left|M_{i,k-1}^{-1}Y\right|_{m_i} \right\rfloor \leq \rho_{k-1,\max} = \left\lfloor \sum_{i=1}^{k-1} \frac{m_i - 1}{m_i} \right\rfloor = k - 1 - \left\lceil \sum_{i=1}^{k-1} \frac{1}{m_i} \right\rceil \leq k - 2. \tag{7}$$

Для многих компьютерных приложений, включая СЗИ, достаточно в качестве базовой использовать версию МИМА, ориентированную на оперирование лишь с неотрицательными ЦЧ. В этом случае требуемая конфигурация МИМСС должна удовлетворять следующей теореме.

Теорема 3. *Для того, чтобы в МСС с попарно простыми основаниями* $m_1, m_2, \ldots, m_k$ *ИИ* $I(X)$ *каждого элемента* X *диапазона* $\mathbf{D}_+ = \mathbf{Z}_M = \{0, 1, \ldots, M-1\}$ *однозначно определялся компьютерным ИИ* $\hat{I}_k(X)$, *рассчитываемым по* (3), (4) *при* $l=k$, *необходимо и достаточно, чтобы k-й модуль МСС удовлетворял условию* $m_k \geq m_0 + \rho_{k-1,\max}$ ($m_0 \geq \rho_{k-1,\max}$ (см. (7)). *При этом для* $I(X)$ *справедлива формула*

$$I(X) = \begin{cases} \hat{I}_k(X), & \text{если } \hat{I}_k(X) < m_0, \\ \hat{I}_k(X) - m_k, & \text{если } \hat{I}_k(X) \geq m_k - \rho_{k-1,\max}. \end{cases}$$

Из теорем 2 и 3 видно, что при минимально избыточном модулярном кодировании интервально-индексные характеристики и ИМФ (1) позволяют достичь принципиально нового, в сравнении с традиционными конфигурациями МА, уровня оптимизации немодульных процедур по таким, в частности, показателям как быстродействие и объем реализационных затрат. Весьма показательными в этом отношении являются операции расширения МК, выполняемые в рамках МА-алгоритмов Монтгомери для умножения по большим модулям.

Расширение МК $(\chi_1, \chi_2, \ldots, \chi_l)$ на основание m_j МИМСС ($1<l<k$; $l<j\leq k$) сводится к вычислению ИИ $I_l(X)$ по формулам (3), (4), (6) и последующему применению ИМФ l-го порядка (см. (1)). Результирующее расчетное соотношение для рассматриваемой операции расширения кода имеет вид

$$\chi_j = \left|X\right|_{m_j} = \left|\sum_{i=1}^{l-1}\left|M_{i,l-1}\left|M_{i,l-1}^{-1}\chi_i\right|_{m_i}\right|_{m_j} + \left|M_{l-1}I_l(X)\right|_{m_j}\right|_{m_j}. \tag{8}$$

Предполагается, что число X, отвечающее МК $(\chi_1, \chi_2, \ldots, \chi_l)$, служит элементом диапазона $\mathbf{D}_l = \mathbf{Z}_{2M'}^{-} = \{-M', -M'+1, \ldots, M'-1\}$ ($M' = m_0' M_{l-1}$; m_0' – вспомогательный модуль, $m_0' \geq \rho_{l-1,\max}$ (см. (7)), который выбирается в соответствии с требованием минимальной избыточности МСС с основаниями $m_1, m_2, \ldots, m_l$, т.е. согласно теореме 2 – с учетом условия $m_l \geq 2m_0' + \rho_{l-1,\max}$. Это гарантирует корректность расчетного соотношения (8) операции расширения кода $(\chi_1, \chi_2, \ldots, \chi_l)$.

Что касается модульных операций над произвольными ЦЧ A и B, заданными своими минимально избыточными МК (МИМК):

$$A = (\alpha_1, \alpha_2, \ldots, \alpha_k),\ B = (\beta_1, \beta_2, \ldots, \beta_k)\ (\alpha_i = \left|A\right|_{m_i}, \beta_i = \left|B\right|_{m_i}\ (i = \overline{1,k})),$$

то в МИМСС они выполняются независимо по каждому из оснований, т.е. по правилу

$$A \circ B = (\alpha_1, \alpha_2, \ldots, \alpha_k) \circ (\beta_1, \beta_2, \ldots, \beta_k) =$$
$$(|\alpha_1 \circ \beta_1|_{m_1}, |\alpha_2 \circ \beta_2|_{m2}, \ldots, |\alpha_k \circ \beta_k|_{m_k})\ (\circ \in \{+, -, \cdot\}). \tag{9}$$

В свойстве (9) заключается главное фундаментальное преимущество МА над арифметикой позиционных систем счисления (ПСС).

2. Модулярная версия метода Монтгомери для умножения по простому модулю

Как известно продуктивность компьютерных приложений МВТ значительно возрастает с повышением уровня модульности целевых функций решаемых задач, причем на диапазонах больших чисел влияние данного фактора является особенно ощутимым. Весьма показательным в этом отношении примером служат МА-версии мультипликативных процедур на базе метода Монтгомери [9, 10, 12−14]. Дело в том, что в рамках метода Монтгомери для получения искомого произведения операндов $A=(\alpha_1, \alpha_1, \ldots, \alpha_k)$ и $B=(\beta_1, \beta_2, \ldots, \beta_k)$ в качестве базового используется выражение, значения которого кратны специально выбираемому вспомогательному модулю, а поскольку деление нацело (формальное деление) в МСС относится к разряду модульных операций, то МА-реализации применяемого подхода и обеспечивают высокую эффективность. Кроме того, переход от традиционной (неизбыточной) МА к МИМА, которая отличается более совершенными немодульными процедурами, включая процедуру расширения кода, позволяет достичь еще большего дополнительного повышения эффективности МСС-реализаций метода Монтгомери.

Остановимся подробнее на проблеме разработки минимально избыточной модулярной версии алгоритма Монтгомери умножения по простому модулю p.

Искомая вычислительная схема умножения ЦЧ A и B по простому модулю p методом Монтгомери [9; 18−20] конструируется на основе выражения вида

$$\widetilde{C} = C + |-Cp^{-1}|_{\widetilde{M}}\ p, \tag{10}$$

где $C=AB$; $\widetilde{M}$ − некоторый вспомогательный модуль, взаимно простой с p и удовлетворяющий условию $p<\widetilde{M}$. Сущность основополагающей идеи, выдвинутой Монтгомери [18] для построения требуемой мультипликативной схемы, состоит в обеспечении кратности значений выражения (10) модулю $\widetilde{M}$ при любых A и B. Указанное свойство базового выражения (10) вытекает из равенства:

$$|\widetilde{C}|_{\widetilde{M}} = |C + |-Cp^{-1}|_{\widetilde{M}}\ p|_{\widetilde{M}} = |C - Cp^{-1}p|_{\widetilde{M}} = 0 \tag{11}$$

Из (11) следует, что число

$$\widetilde{C}/\widetilde{M} = (C + p(|-Cp^{-1}|_{\widetilde{M}})/\widetilde{M})/\widetilde{M} \tag{12}$$

является целым. При этом переход в (12) к остаткам по модулю p дает

$$|\widetilde{C}/\widetilde{M}|_p = |C/\widetilde{M}|_p = |AB/\widetilde{M}|_p. \tag{13}$$

В соответствии с (13) в качестве искомого произведения операндов A и B принимается ЦЧ

$$\widetilde{\gamma} = |AB/\widetilde{M}|_p = (\widetilde{C}/\widetilde{M}) - Qp, \tag{14}$$

где Q – однозначно определяемый (для заданных операндов A и B) целочисленный коэффициент.

Пусть $A, B \in \mathbf{Z}_p$, тогда из (12) для $\widetilde{C}/\widetilde{M}$ вытекает оценка: $\widetilde{C}/\widetilde{M} < ((p-1)^2 + (\widetilde{M}-1)p)/\widetilde{M} = (p^2 - 2p + \widetilde{M}p)/\widetilde{M} = p + p(p-1)/\widetilde{M}$. Отсюда в виду $p<\widetilde{M}$ заключаем что, $\widetilde{C}/\widetilde{M} < 2p$. Следовательно, Q в (14) может принимать лишь два значения: 0 или 1. Таким образом, для определения коэффициента Q достаточно воспользоваться одной операцией сравнения ЦЧ, а именно чисел $\widetilde{\gamma} = \widetilde{C}/\widetilde{M}$ и p. Изложенное позволяет заключить, что базовая вычислительная схема для метода Монтгомери сводится к операционной последовательности:

$$\langle C=AB;\ D=|CF|_{\widetilde{M}}\ (F=|-p^{-1}|_{\widetilde{M}});\quad \widetilde{C} = C + Dp;\ \widetilde{\gamma} = (\widetilde{C}/\widetilde{M}) - Qp\ \ (Q \in \mathbf{Q},\ \mathbf{Q}=\{0, 1\})\rangle \tag{15}.$$

При выполнении умножения по модулю p в ПСС по схеме (15) вспомогательный модуль $\widetilde{M}$ удобно выбрать равным в двоичной экспоненте: $\widetilde{M} = 2^{\widetilde{b}}$ ($\widetilde{b}$ – разрядность $\widetilde{M}$ в битах). В этом случае операция $D=|CF|_{\widetilde{M}}$ существенно упрощается за счет тривиальности процедуры приведения ЦЧ

к остаткам по модулю $\widetilde{M}$, а деление нацело $\widetilde{C}$ на $\widetilde{M}$ сводится к сдвигу двоичного кода числа $\widetilde{C}$ на $\widetilde{b}$ разрядов вправо.

Предполагается, что мультипликативная инверсия $F=|-p^{-1}|_{\widetilde{M}}$ находится на этапе предварительных вычислений.

3. Синтез МИМА-алгоритма умножения по большим простым модулям на основе метода Монтгомери

Перейдем теперь к проблеме построения минимально избыточной модулярной вычислительной схемы типа (15) для метода Монтгомери и синтеза на ее основе МИМА-алгоритма умножения по модулю p.

Как видно из (15), трудоемкость ПСС-версий алгоритма Монтгомери определяется главным образом сложностью операций умножения больших чисел: AB, $|CF|_{\widetilde{M}}$ и Dp. Сказанное относится и к операции мультипликативной инверсии: $F=|-p^{-1}|_{\widetilde{M}}$. Однако, являясь параметром долговременного использования F может быть получена на этапе предварительных вычислений. Поэтому сложность операции определения F принципиального значения не имеет. В МСС все указанные операции, являясь модульными, реализуются значительно проще, чем в ПСС. Именно этим обстоятельством, обусловленным внутренним параллелизмом модулярных вычислительных структур, в первую очередь, и продиктована целесообразность применения МА в системах криптографической защиты информации.

Пусть операнды A, B и модуль p заданы в МИМСС с основаниями $m_1, m_2, ..., m_k$ и динамическим диапазоном $\mathbf{D}=\mathbf{Z}_{2M}^{-}$: $A=(\alpha_1, \alpha_2, \ldots, \alpha_k)$, $B=(\beta_1, \beta_2, \ldots, \beta_k)$, $p=(\pi_1, \pi_2, \ldots, \pi_k)$ и пусть $\widetilde{M} = M_l = \prod_{i=1}^{l} m_i$ $(1<l<k)$. Так как мультипликативная инверсия $F \in \mathbf{Z}_{M_l}=\{0, 1, \ldots, M_l-1\}$, то она однозначно определяется своим МК $(\varphi_1, \varphi_2, \ldots, \varphi_l)$, цифры которого находятся из сравнений $\pi_i\varphi_I \equiv -1(\mathrm{mod}\ m_i)$ $(\varphi_i \in \mathbf{Z}_{m_i}$; $i=\overline{1, l}$). Решение данной системы сравнений производится на стандартных компьютерных диапазонах, причем в ходе предварительных вычислений. Число D также определяется кодом МСС с основаниями $m_1, m_2, ..., m_l$:

$$D=|CF|_{M_l} = (\delta_1, \delta_2, \ldots, \delta_l)=(|\gamma_1\varphi_1|_{m_1}, |\gamma_2\varphi_2|_{m_2}, ..., |\gamma_l\varphi_l|_{m_l})$$
$$(\delta_i=|D|_{m_i}, \gamma_i=|C|_{m_i}=|\alpha_i\beta_i|_{m_i}\ (\ i=\overline{1, l}\ ). \qquad (16)$$

Поскольку ЦЧ D, полученное согласно (16) участвует в дальнейших вычислениях при получении $\widetilde{C}$ (см. (10)) по полной системе модулей МИМСС, то МК $(\delta_1, \delta_2, \ldots, \delta_l)$ должен быть расширен на остальные модули: $m_{l+1}, m_{l+2}, \ldots, m_{l+k}$. В рамках неизбыточного модулярного кодирования, а именно таковым является кодовое пространство МСС с основаниями $m_1, m_2, ..., m_l$ и диапазоном $\mathbf{D}_{M_l}$, данная операция требует использования сложно вычисляемых ИХМК, например, ранга [5; 6; 9]. В целях устранения отмеченного негативного фактора D заменим на число

$$\hat{D} = \sum_{i=1}^{l-1} M_{i,l-1}\,|\,M_{i,l-1}^{-1}\delta_i\,|_{m_i} + M_{l-1}\hat{I}_l(D), \qquad (17)$$

где $\hat{I}_l(D)=|I_l(D)|_{m_l}$ − компьютерный ИИ ЦЧ D и $\hat{D}$ относительно модулей $m_1, m_2, ..., m_l$, который согласно (3) вычисляется по формуле

$$\hat{I}_l(\hat{D}) = \hat{I}_l(D) = |\sum_{i=1}^{l} R_{i,l}(\delta_i)|_{m_l}\ ; \qquad (18)$$

вычеты $R_{i,l}(\delta_i)$ в (18) рассчитываются по правилу типа (4):

$$R_{i,l}(\delta_i)=\left\lfloor \frac{m_l}{m_i}\,|\,M_{i,l}^{-1}\delta_i\,|_{m_i} \right\rfloor = |-m_{i,l}^{-1}\,|\,M_{i,l-1}^{-1}\delta_i\,|_{m_i}|_{m_l}\ (i \neq l)\,,\ R_{l,l}(\delta_l)=\,|\,M_{l-1}^{-1}\delta_l\,|_{m_l}\ . \qquad (19)$$

Применяя (1) и (2) запишем $\hat{D}$ в виде

$$\hat{D} = \sum_{i=1}^{l-1} M_{i,l-1} \,|\, M_{i,l-1}^{-1} \delta_i \,|_{m_i} + M_{l-1}(\hat{I}_l(D) - m_l\theta_l(D) + m_l\theta_l(D)) =$$

$$= \sum_{i=1}^{l-1} M_{i,l-1} \,|\, M_{i,l-1}^{-1} \delta_i \,|_{m_i} + M_{l-1} I_l(D) + M_l\theta_l(D) = D + M_l\theta_l(D)\,, \qquad (20)$$

где $\theta_l(D)$ – двузначная минимальная ИХМК ($\theta_l(D) \in \{0, 1\}$). Из (20) вытекает равенство

$$|\, \hat{D} \,|_{M_l} = D\,. \qquad (21)$$

В соответствии с изложенным искомая модификация базового соотношения для синтеза МИМА-процедуры умножения Монтгомери имеет вид

$$\hat{C} = C + \hat{D}p = C + (\sum_{i=1}^{l-1} M_{i,l-1} \,|\, M_{i,l-1}^{-1} \delta_i \,|_{m_i} + M_{l-1}\hat{I}_l(D))\,p\,. \qquad (22)$$

С учетом (17), (21), а также (11) из (22) следует, что

$$|\, \hat{D} \,|_{M_l} = |\, C + |\, \hat{D}p \,|_{M_l} \,|_{M_l} = |\, C + |\, Dp \,|_{M_l} \,|_{M_l} = 0\,.$$

Таким образом, ЦЧ (22) нацело делится на M_l, то есть

$$\hat{\gamma} = \hat{C} / M_l = (C + (\sum_{i=1}^{l-1} M_{i,l-1} \,|\, M_{i,l-1}^{-1} \delta_i \,|_{m_i} + M_{l-1}\hat{I}_l(D))\,p) / M_l \qquad (23)$$

является целым.

Процесс реализации (23) в МИМСС с основаниями модулей m_1, m_2, ..., m_k, то есть получения МИМК ($\hat{\gamma}_1, \hat{\gamma}_2$, ..., $\hat{\gamma}_k$) ЦЧ $\hat{\gamma}$ ($\hat{\gamma}_i = |\, \hat{\gamma} \,|_{m_i}$ ($i = \overline{1, k}$)) состоит из двух шагов. На первом шаге $\hat{\gamma}$ вычисляется по набору модулей m_{l+1}, m_{l+2}, ..., m_k, а на втором шаге сформированный усеченный МК ($\hat{\gamma}_{l+1}, \hat{\gamma}_{l+2}, ..., \hat{\gamma}_k$) расширяется на модули m_1, m_2, ..., m_l. При этом данная операция, естественно должна выполняться по упрощенной минимально избыточной процедуре расширения:

$$\hat{\gamma}_j = |\, \sum_{i=l+1}^{k-1} |\, \frac{M_{k-1}}{M_l m_i} |\, \frac{M_l m_i}{M_{k-1}} \hat{\gamma}_i \,|_{m_i} \,|_{m_j} + |\, \frac{M_{k-1}}{M_l} I'_{k-l}(\hat{\gamma}) \,|_{m_j} \,|_{m_j} \;(j = \overline{1, l})\,, \qquad (24)$$

где $I'_{k-l}(\hat{\gamma})$ – ИИ числа $\hat{\gamma}$ относительно модулей m_{l+1}, m_{l+2}, ..., m_k. Так как базовая МСС с основаниями m_1, m_2, ..., m_k и динамическим диапазоном $\mathbf{D} = \mathbf{Z}_{2M}^-$ минимально избыточна, то согласно теореме 2 МСС с основаниями m_{l+1}, m_{l+2}, ..., m_k и диапазоном $\hat{\mathbf{D}} = \mathbf{Z}_{2M/M_l}^- = \{-M/M_l, M/M_l + 1, ..., M/M_l - 1\}$ также является минимально избыточной. Поэтому для ИИ $I'_{k-l}(\hat{\gamma})$ верна формула

$$I'_{k-l}(\hat{\gamma}) = \begin{cases} \hat{I}'_{k-l}(\hat{\gamma}), & \text{если } \hat{I}'_{k-l}(\hat{\gamma}) < m_0, \\ \hat{I}'_{k-l}(\hat{\gamma}) - m_k, & \text{если } \hat{I}'_{k-l}(\hat{\gamma}) \ge m_0. \end{cases} \qquad (25)$$

Компьютерный ИИ $\hat{I}'_{k-l}(\hat{\gamma}) = |\, I'_{k-l}(\hat{\gamma}) \,|_{m_k}$ определяется по расчетным соотношениям типа (3), (4):

$$\hat{I}'_{k-l}(\hat{\gamma}) = |\, \sum_{i=l+1}^{k} R'_{i,k}(\hat{\gamma}_i) \,|_{m_k}\,; \qquad (26)$$

$$R'_{i,k}(\hat{\gamma}_i) = \left\lfloor \frac{m_k}{m_i} |\, \frac{M_l m_i}{M_k} \hat{\gamma}_i \,|_{m_i} \right\rfloor = |\, -m_i^{-1} |\, \frac{M_l m_i}{M_{k-1}} \hat{\gamma}_i \,|_{m_i} \,|_{m_k} \;(i \neq k),$$

$$R'_{k,k}(\hat{\gamma}_k) = |\, \frac{M_l}{M_{k-1}} \hat{\gamma}_k \,|_{m_k}\,. \qquad (27)$$

Необходимым и достаточным условием корректности двухшагового процесса вычисления величины $\hat{\gamma}$, осуществляемого согласно (23)–(27) при $C = AB \in \mathbf{D} = \mathbf{Z}_{2M}^-$, служит принадлежность

чисел $\hat{D}$ и $\hat{\gamma}$ диапазону $\hat{\mathbf{D}} = \mathbf{Z}_{2M/M_l}^{-}$ МИМСС с основаниями m_{l+1}, m_{l+2}, ..., m_k. Отметим, что ЦЧ $\hat{C}$ (см. (22)) как промежуточный результат последовательности модульных операций, может и не быть элементом $\hat{\mathbf{D}}$. Найдем ограничения на модули m_1, m_2, ..., m_k и p, гарантирующее выполнение условия $\hat{D}$, $\hat{\gamma} \in \hat{\mathbf{D}}$.

Получим верхнюю оценку для $\hat{D}$, используя ранговую форму (l−1)-го порядка для ЦЧ (см. (5)), представим (17) в виде:

$$\hat{D} = \sum_{i=1}^{l-1} M_{i,l-1} \mid M_{i,l-1}^{-1} \delta_i \mid_{m_i} - M_{l-1}\rho_{l-1}(D) + M_{l-1}\rho_{l-1}(D) + M_{l-1}\hat{I}_l(D) =$$
$$= \mid D \mid_{M_{l-1}} + M_{l-1}(\rho_{l-1}(D) + \hat{I}_l(D))$$

Отсюда с учетом $\rho_{l-1}(D) \le \rho_{l-1,\max} \le l-2$ (см. (7)) имеем

$$\hat{D} \le M_{l-1} - 1 + M_{l-1}(l-2) + M_{l-1}(m_l - 1) = M_l + M_{l-1}(l-2) - 1. \quad (28)$$

Пусть $AB \in \mathbf{Z}_p = \{0, 1, \ldots, p-1\}$. Тогда с учетом (28) из (23) получим

$$\hat{\gamma} < (p^2 + (M_l + M_{l-1}(l-2) - 1)p)/M_l = p(1 + (p + M_{l-1}(l-2) - 1)/M_l). \quad (29)$$

Из (29) видно, что при выполнении неравенства $p + M_{l-1}(l-2) - 1 < M_l$ ЦЧ $\hat{\gamma} \in \mathbf{Z}_{2p}$ $\{0, 1, \ldots, 2p-1\}$. Такого же результата, то есть принадлежности $\hat{\gamma}$ множеству $\mathbf{Z}_{2p}$ можно достичь и в случае, когда A, $B \in \mathbf{Z}_{2p}$. Соответствующее ограничение на основания МИМСС и модуль p вытекает из неравенства $\hat{\gamma} < (4p^2 + (M_l + M_{l-1}(l-2) - 1)p)/M_l < 2p$ и имеет вид

$$4p + M_{l-1}(l-2) - 1 < M_l. \quad (30)$$

В рамках данного условия, обеспечивающего A, B, $\hat{\gamma} \in \mathbf{Z}_{2p}$, допускается режим многократного обращения к процедуре умножения по модулю p с использованием в качестве операндов результатов уже выполненных операций умножения. Это, в частности, необходимо для реализации в криптосистемах операций возведения в степень по модулю p.

Как уже отмечалось $\hat{D}$ и $\hat{\gamma}$ должны быть элементами диапазона $\hat{\mathbf{D}}$. С учетом (28) и $\hat{\gamma} < 2p$ это требование приводит к неравенствам:

$$M_l + M_{l-1}(l-2) - 1 < M/M_l \quad (31)$$

и

$$2p < M/M_l,$$

второе из которых ввиду (30) вытекает из первого.

Таким образом, приведенные оценочные выкладки позволяют заключить, что при A, B, $\hat{\gamma} \in \mathbf{Z}_{2p}$, корректность предлагаемой вычислительной МИМА-схемы Монтгомери (см. (22), (23)) обеспечивается в рамках условий (30) и (31). Заметим, что при выполнении (30) и (31) произведение $C = AB$ является элементом динамического диапазона $\hat{\mathbf{D}}$ базовой МИМСС.

Переход в (23) к остаткам по модулю p дает

$$\tilde{\gamma} = \mid \hat{\gamma} \mid_p = \mid \hat{C} M_l^{-1} \mid_p = \mid ABM_l^{-1} \mid_p. \quad (32)$$

Так как $\hat{\gamma} < 2p$, то аналогично (14) искомое произведение (32) ЦЧ A и B по модулю p можно получить по $\hat{\gamma}$ с использованием равенства

$$\tilde{\gamma} = \mid ABM_l^{-1} \mid_p = \hat{\gamma} - Qp \ (Q \in \{0,1\}) \quad (33)$$

Остановимся кратко на некоторых особенностях реализации в базовой МИМСС выражений (32) и (33). Расчет цифр МИМК ($\hat{\delta}_{l+1}, \hat{\delta}_{l+2}, \ldots, \hat{\delta}_k$) числа $\hat{D}$ по набору модулей m_{l+1}, m_{l+2}, ..., m_k. осу-

ществляется с помощью операции расширения, которая в соответствии с (17) выполняется по правилу

$$\hat{\delta}_j = |\sum_{i=1}^{l-1} |M_{i,l-1}|M_{i,l-1}^{-1}\delta_i|_{m_i}|_{m_j} + |M_{l-1}\hat{I}_l(D)|_{m_j}|_{m_j} \quad (j=\overline{l+1,k}) \tag{34}$$

с применением (18) и (19).

Обозначим МК числа $\hat{C}$ вида (22) через $(\gamma'_1,\gamma'_2,...,\gamma'_k)$. Поскольку $\hat{C}$ нацело делится на M_l, то $|\hat{C}|_{M_l}=0$ и следовательно $\gamma'_1=0, \gamma'_2=0, ..., \gamma'_l=0$. Поэтому после определения произведения $C=AB=(\gamma_1, \gamma_2, ..., \gamma_k)$ и ЦЧ $\hat{D}=(\hat{\delta}_1,\hat{\delta}_2, ..., \hat{\delta}_k)$ ($\hat{\delta}_i=|\hat{D}|_{m_i}$ $(i=\overline{1,k})$; $\hat{\delta}_1=\delta_1, \hat{\delta}_2=\delta_2, ..., \hat{\delta}_l=\delta_l$ (см. (21)) число $\hat{C}$ достаточно вычислить в соответствии с (22) в МСС с основаниями $m_{l+1}, m_{l+2}, ..., m_k$.:

$$(\gamma'_{l+1},\gamma'_{l+2},...,\gamma'_k) = (|\gamma_{l+1}+\hat{\delta}_{l+1}\pi_{l+1}|_{m_{l+1}}, |\gamma_{l+2}+\hat{\delta}_{l+2}\pi_{l+2}|_{m_{l+2}}, ..., |\gamma_k+\hat{\delta}_k\pi_k|_{m_k}). \tag{35}$$

Что касается операции модульного умножения ЦЧ $\hat{C}$ на константу M_l^{-1}, выполняемой в МСС с основаниями $m_{l+1}, m_{l+2}, ..., m_k$.:

$$\hat{\gamma} = \hat{C}M_l^{-1} = (\hat{\gamma}_{l+1},\hat{\gamma}_{l+2},...,\hat{\gamma}_k) = (|M_l^{-1}\gamma'_{l+1}|_{m_{l+1}}, |M_l^{-1}\gamma'_{l+2}|_{m_{l+2}},...,|M_l^{-1}\gamma'_k|_{m_k}), \tag{36}$$

то она может быть реализована не на умножителях по модулям $m_{l+1}, m_{l+2}, ..., m_k$, а с помощью простых таблиц.

Расширение МИМК (36) ЦЧ $\hat{\gamma}$ на основания $m_1, m_2, ..., m_l$ МИМСС производится по расчетным соотношениям (24)−(27).

Изложенное позволяет сформулировать нижеследующий алгоритм умножения по большим простым модулям.

Параметры алгоритма: определяющие основания базовой МИМСС – $m_0, m_1, ..., m_k$ и простой модуль $p=(\pi_1, \pi_2, \ldots, \pi_k)$ ($\pi_i=|p|_{m_i}$ $(i=\overline{1,k})$), которые удовлетворяют условиям (30) и (31).

Входные данные: операнды A и B ($A, B \in \mathbf{Z}_{2p}=\{0, 1, \ldots, 2p-1\}$), представленные в МИМСС − $A=(\alpha_1, \alpha_2, \ldots, \alpha_k)$, $B=(\beta_1, \beta_2, \ldots, \beta_k)$.

Выходные данные: МИМК $(\hat{\gamma}_1,\hat{\gamma}_2,...,\hat{\gamma}_k)$ аналога $\hat{\gamma}\in\mathbf{Z}_{2p}$ нормированного остатка $\tilde{\gamma}=|ABM_l^{-1}|_p$ $(M_l=\prod_{i=1}^{l} m_i;\ 1<l<k)$, отвечающего произведению AB по модулю p ($\hat{\gamma}\equiv\tilde{\gamma}\ (\mathrm{mod}\ p)$).

Предварительно вычисляемые данные:

- код $(\varphi_1, \varphi_2, \ldots, \varphi_l)$ противоположного значения $F=|-p^{-1}|_{M_l}$ мультипликативной инверсии $|p^{-1}|_{M_l}$ модуля p в МСС с основаниями $m_1, m_2, ..., m_l$, получаемый из системы сравнений $\pi_i\varphi_i\equiv -1\ (\mathrm{mod}\ m_i)$ ($\varphi_i\in\mathbf{Z}_{m_i}$; $(i=\overline{1,l})$;
- таблицы ИИ *TIIi* и *TII_i*, которые формируются согласно (19) и (27):

  $TIIi[\chi]=R_{i,l}(\chi)$ $(\chi\in\mathbf{Z}_{m_i}; i=\overline{1,l})$;

  $TII\_i[\chi]=R'_{i,k}(\chi)$ $(\chi\in\mathbf{Z}_{m_i}; i=\overline{l+1,k})$;
- таблицы расширения МК *TEi_j* и *TE_i_j*, генерируемые в соответствии с (34) и (24) по формулам

$$TEi\_j[\chi] = \begin{cases} |M_{i,l-1}|M_{i,l-1}^{-1}\chi|_{m_i}|_{m_j} \ (\chi\in\mathbf{Z}_{m_i}) \text{ при } i=\overline{1,l-1}, \\ |M_{l-1}\chi|_{m_j} \ (\chi\in\mathbf{Z}_{m_l}) \text{ при } i=l \end{cases}$$

$(j=\overline{l+1,k})$;

$$TE\_i\_j[\chi]=\begin{cases}|\dfrac{M_{k-1}}{M_l m_i}|\dfrac{M_l m_i}{M_{k-1}}\chi|_{m_i}|_{m_j}\ (\chi\in\mathbf{Z}_{m_i})\ \text{при}\ i=\overline{l+1,\,k-1},\\ |\dfrac{M_{k-1}}{M_l}\chi|_{m_j}\ (\chi\in\mathbf{Z}_{m_k})\ \text{при}\ i=k\ \text{и}\ \chi<m_0,\\ |\dfrac{M_{k-1}}{M_l}(\chi-m_k)|_{m_j}\ (\chi\in\mathbf{Z}_{m_k})\ \text{при}\ i=k\,\text{и}\,\chi\ge m_0\end{cases}$$
$$(j=\overline{1,\,l});$$

- таблицы *TMPli* умножения на константу M_l^{-1}, элементы которых рассчитываются по формуле:

$$TMPli[\chi]=|M_l^{-1}\chi|_{m_i}\ \ (\chi\in\mathbf{Z}_{2m_i-1};\ i=\overline{l+1,\,k}).$$

УМ.М1. В базовой МИМСС найти произведение $C=AB=(\gamma_1,\gamma_2,\ldots,\gamma_k)=(|\alpha_1\beta_1|_{m_1},|\alpha_2\beta_2|_{m_2},\ldots,|\alpha_k\beta_k|_{m_k})$.

УМ.М2. В МСС с основаниями $m_1, m_2, \ldots, m_l$ сформировать код числа $D=|\mathrm{CF}|_{M_l}$:

$$(\delta_1,\delta_2,\ldots,\delta_l)=(|\gamma_1\varphi_1|_{m_1},|\gamma_2\varphi_2|_{m_2},\ldots,|\gamma_l\varphi_l|_{m_l}).$$

УМ.М3. Вычислить интервально-индексную характеристику $\hat{I}_l(\hat{D})=\hat{I}_l(D)$ (см. (21)) числа

$$\hat{D}=\sum_{i=1}^{l-1}M_{i,\,l-1}|M_{i,\,l-1}^{-1}\delta_i|_{m_i}+M_{l-1}\hat{I}_l(D)$$

по расчетному соотношению

$$\hat{I}_l(\hat{D})=\eta_l=|\sum_{i=1}^{l}TIIi[\delta_i]|_{m_l}\ .$$

УМ.М4. Определить цифры МИМК $(\hat{\delta}_{l+1},\hat{\delta}_{l+2},\ldots,\hat{\delta}_k)$ ЦЧ $\hat{D}$ в МСС с основаниями $m_{l+1}, m_{l+2}, \ldots, m_k$. по правилу

$$\hat{\delta}_j=|\sum_{i=1}^{l-1}TEi\_j[\delta_i]+TEl\_j[\eta_l]|_{m_j}\ (j=\overline{l+1,k}).$$

УМ.М5. Получить код числа $\hat{C}=C+\hat{D}p$ в МИМСС с модулями $m_{l+1}, m_{l+2}, \ldots, m_k$.:

$$(\gamma'_{l+1},\gamma'_{l+2},\ldots,\gamma'_k)=(|\gamma_{l+1}+\hat{\delta}_{l+1}\pi_{l+1}|_{l+1},|\gamma_{l+2}+\hat{\delta}_{l+2}\pi_{l+2}|_{l+2},\ldots,|\gamma_k+\hat{\delta}_k\pi_k|_k\ .$$

УМ.М6. В МИСС с основаниями $m_{l+1}, m_{l+2}, \ldots, m_k$ сформировать код $(\hat{\gamma}_{l+1},\hat{\gamma}_{l+2},\ldots,\hat{\gamma}_k)$ ЦЧ $\hat{\gamma}=\hat{C}\hat{M}_l^{-1}$ по правилу $\hat{\gamma}_i=TMPli[\gamma'_i]$ $(i=\overline{l+1,\,k})$.

УМ.М7. Рассчитать интервально-индексную характеристику $\hat{I}'_{k-l}(\hat{\gamma})$ ЦЧ $\hat{\gamma}$ относительно модулей $m_{l+1}, m_{l+2}, \ldots, m_k$.:

$$\hat{I}'_{k-l}(\hat{\gamma})=\eta'_k=|\sum_{i=l+1}^{k}TII\_i[\hat{\gamma}_i]|_{m_k}\ .$$

УМ.М8. Расширить МИМК $(\hat{\gamma}_{l+1},\hat{\gamma}_{l+2},\ldots,\hat{\gamma}_k)$ на модули $m_1, m_2, \ldots, m_l$ с применением расчетного соотношения

$$\hat{\gamma}_j=|\sum_{i=l+1}^{k-1}TE\_i\_j\,[\hat{\gamma}_i]+TE\_k\_j[\eta'_k]|_{m_j}\ \ (j=\overline{1,\,l}).$$

УМ.М9. Число $\hat{\gamma}=\hat{C}M_l^{-1}=(\hat{\gamma}_1,\hat{\gamma}_2,\ldots,\hat{\gamma}_k)$ зафиксировать в качестве искомого аналога нормированного произведения $\tilde{\gamma}=|ABM_l^{-1}|_p$ операндов A и B по модулю p и завершить работу алгоритма.

Справедливо следующее утверждение.

Теорема 4. *Пусть подсистемы* $<m_1, m_2, \ldots, m_l>$ и $<m_{l+1}, m_{l+2}, \ldots, m_k>$ *набора оснований* $m_1, m_2, \ldots, m_k$ *базовой МИИМСС с динамическим диапазоном* $\mathbf{D}=\mathbf{Z}_{2M}^{-}=\{-M,\,-M+1,\,\ldots,\,M-1\}$

$(M=\prod_{j=0}^{k-1} m_j = m_0 M_{k-1};\ \ m_0 \ge \rho;\ m_k \ge 2m_0+\rho;\ \rho \le k-2;\ 1<l<k)$ *и простой модуль* p *удовлетворяют условию*

$$\begin{cases} 4p + M_{l-1}(l-2) - 1 < M_l, \\ M_l + M_{l-1}(l-2) - 1 < M / M_l \end{cases} \tag{37}$$

и пусть операнды A *и* B *мультипликативной операции* $\tilde{\gamma} = |ABM_l^{-1}|_p$ *принадлежат множеству* $\mathbf{Z}_{2p} = \{0, 1, ..., 2p-1\}$. *Тогда выходная величина* $\hat{\gamma}$ *МИМА-алгоритма* УМ.М1−УМ.М9 *умножения по модулю* p *на базе метода Монтгомери также является элементом множества* $\mathbf{Z}_{2p}$, *при этом* $\hat{\gamma} \equiv \tilde{\gamma} (\mathrm{mod}\ p)$ *и для* $\tilde{\gamma}$ *верна формула*

$$\tilde{\gamma} = \hat{\gamma} - p\overline{S}(\hat{\gamma} - p), \tag{38}$$

где через $\overline{S}(a)$ *обозначается знаковая функция вида*

$$\overline{S}(a) = \begin{cases} 1, & \text{если } a \ge 0; \\ 0, & \text{если } a < 0. \end{cases}$$

Алгоритм УМ.М1−УМ.М9 предназначен, в первую очередь, для вычисления степеней натуральных чисел по большим модулям p. Используемые при этом процедуры сводятся к последовательностям мультипликативных операций рассматриваемого вида. В условиях теоремы 4 значения операндов и результатов этих операций являются элементами множества $\mathbf{Z}_{2p}$. Поэтому ввиду равенства $|X|_p = ||X|_{2p}|_p$ (X – произвольное ЦЧ) после каждого умножения в процессе получения степеней корректирующую операцию (38) можно не выполнять. Достаточно осуществить коррекцию $\hat{\gamma}$ только для заключительной операции умножения соответствующей последовательности, причем лишь в случаях, когда необходим переход к расчетам в позиционном коде. В рамках производимого на данном этапе преобразования МК в код ПСС и надлежит выполнять корректирующую операцию (38). Для этого может быть применен предложенный в [17] алгоритм МП.1–МП.4., адаптированный к МИМСС с усеченным набором оснований: $< m_{l+1}, m_{l+2}, ..., m_k >$ и диапазоном $\mathbf{Z}_{2M/M_l}$.

На однопроцессорной ЭВМ необходимое преобразование МК в позиционный код (ПК), реализуемое по схеме побитового расслоения ИМФ, занимает время $t_{\text{МК}\to\text{ПК}} = (k-l)\lceil r/32 \rceil t_{\text{сл}}$, где r – разрядность модуля p; $t_{\text{сл}}$ − длительность операции сложения двух 32-битовых ЦЧ. При этом объем требуемой табличной памяти составляет $M_{\text{МК}\to\text{ПК}} = \lceil (k-l)/b_{\max} \rceil r\, 2^{b_{\max}}$ бит ($b_{\max}$ – разрядность оснований МИМСС). Аналогичные оценки быстродействия и затрат табличной памяти для преобразования ПК в МК определяются формулами : $t_{\text{ПК}\to\text{МК}} = 0{,}5k \lceil r/\, b_{\max} \rceil t_{\text{сл}}$ и $M_{\text{ПК}\to\text{МК}} = 0{,}5k\, b_{\max}\, 2^{b_{\max}} + 1$ бит.

Если, например, r=2462 и $b_{\max}$ = 16 бит, что требует применения МИМСС с l=157 и k=315, то при $t_{\text{сл}}$ = 2нс приведенные аналитические оценки дают: $t_{\text{МК}\to\text{ПК}} \approx 24332$ нс, $M_{\text{МК}\to\text{ПК}} \approx 192$ Мб, $t_{\text{ПК}\to\text{МК}} \approx 48000$ нс, $M_{\text{ПК}\to\text{МК}} \approx 39$ Мб.

Следует особо подчеркнуть, что необходимость в кодовых преобразованиях отпадает, если к входным данным системы удается применить принцип модулярной интерпретации. Суть его состоит в том, что блоки обрабатываемых данных рассматриваются как МК по используемому набору оснований.

4. Оценки быстродействия и характеристик сложности МИМА-процедуры Монтгомери для умножения по модулю

Предположим, что основания $m_1, m_2, ..., m_k$ МИМСС – простые числа, которые выбираются из отрезка вида $[\hat{m}; 2\hat{m}]$, где $\hat{m}$ − двоичная экспонента. В качестве базового метода умножения по модулям m_i ($i = \overline{1, k}$) примем индексный метод [21].

При реализации процедуры умножения УМ.М1−УМ.М9 на мультипроцессорной системе модулярной обработки информации (СМОИ) шаг УМ.М3 совмещается во времени с шагом УМ.М4. это относится и к шагам УМ.М7, УМ.М8. с учетом отмеченного обстоятельства временные затраты

в данном случае при использовании по каждому основанию МИМСС только одного сумматора ЦЧ составляют

$$t_{\text{УМ.М, СМОИ}} = 3t_{\text{МУ}} + (k+7)\mathrm{t}_{\text{сл}} + 11t_{\text{ч}}, \tag{39}$$

где $t_{\text{МУ}}$, $t_{\text{сл}}$, $t_{\text{ч}}$ – длительности операций умножения по модулям МИМСС (индексным методом), сложения двух ЦЧ и чтение из памяти элемента таблицы соответственно.

Выполнение алгоритма УМ.М1 − УМ.М9 на одиночной ПЭВМ занимает время

$$t_{\text{УМ.М, ПЭВМ}} = 2kt_{\text{МУ}} + ((2l+3)(k-l+1)-1)t_{\text{сл}} + (3k-l+4)t_{\text{ч}}. \tag{40}$$

Ключевая роль в представленной методологии синтеза мультипликативных МИМА-процедур на диапазонах больших чисел отводится табличным вычислениям. Так как при выбранном наборе оснований $m_1, m_2, ..., m_k$ и зафиксированном его разбиении на две группы: $< m_1, m_2, ..., m_l >$ и $< m_{l+1}, m_{l+2}, ..., m_k. >$ генерирование рабочего комплекта таблиц МИМА производится только один раз, причем вне вычислительного процесса, реализуемого в реальном времени, то временные затраты на выполнение данного процесса сводятся к минимуму.

Суммарное количество таблиц, необходимое для МИМА-алгоритма УМ.М1−УМ.М9 умножения по модулю p составляет

$$N_{\text{Т, УМ.М}} = 2(l+3)(k-l)+5l \tag{41}$$

на j-е основание приходится

$$N_{\text{Т, УМ.М},j} = \begin{cases} k-l+4 \ \text{ при } \ j=\overline{1,l-1}, \\ k+4 \ \text{ при } \ j=l, \\ l+5 \ \text{ при } \ j=\overline{l+1,\,k-1}, \\ k+5 \ \text{ при } \ j=k \end{cases}$$

таблиц, которые содержат

$$N_{\text{ЭТ, УМ.М},j} = 2^{b_{\text{сум},j}-\hat{b}_j} + 2^{\hat{b}_j} + 4m_j - 2 + \begin{cases} \sum\limits_{i=l+1}^{k} m_i \ \text{ при } \ j=\overline{1,l-1}, \\ \sum\limits_{i=1}^{k} m_i \ \text{ при } \ j=l, \\ (2m_j-1) + \sum\limits_{i=1}^{l} m_i \ \text{ при } \ j=\overline{l+1,\,k-1}, \\ (2m_k-1) + \sum\limits_{i=1}^{k} m_i \ \text{ при } \ j=k \end{cases} \tag{42}$$

элементов разрядностью $b_j \lceil \log_2 m_j \rceil$ бит. Фигурирующие в (42) параметры $b_{\text{сум},j}$ и $\hat{b}_j$ представляют собой соответственно максимально возможную разрядность накапливаемых согласно аккумулятивно-табличному методу [21] сумм вычетов из кольца $\mathbf{Z}_{m_j}$ и разрядность младших частей этих сумм ($\hat{b}_j \le b_j$).

Обращаясь к шагам УМ.М3, УМ.М4, УМ.М7, УМ.М8 алгоритма умножения заключаем, что

$$b_{\text{сум},j} = \begin{cases} \left\lceil \log_2((k-l)(m_j-1)) \right\rceil \ \text{ при } \ j=\overline{1,l-1}, \\ \max\{\lceil \log_2(l(m_l-1)) \rceil, \lceil \log_2((k-l)(m_l-1)) \rceil\} \ \text{ при } \ j=l, \\ \left\lceil \log_2(l(m_j-1)) \right\rceil \ \text{ при } \ j=\overline{l+1,\,k-1}, \\ \max\{\lceil \log_2(l(m_k-1)) \rceil, \lceil \log_2((k-l)(m_k-1)) \rceil\} \ \text{ при } \ j=k. \end{cases} \tag{43}$$

Из (42) следует, что рабочий комплект таблиц для разработанной МИМА-версии алгоритма умножения по Монтгомери в общей сложности занимает память объемом

$$M_{\text{Т, УМ.М}} = \frac{1}{8}\sum_{j=1}^{k} N_{\text{ЭТ,УМ.М},j}\, b_j = \frac{1}{8}\Big(\sum_{j=1}^{k}(2^{b_{\text{сум},j}-\hat{b}_j} + 2^{\hat{b}_j} + 4m_j - 2)b_j +$$

$$+ \sum_{j=l+1}^{k}(2m_j - 1)b_j + \Big(\sum_{i=l+1}^{k} m_i\Big)\Big(\sum_{j=1}^{l} b_j + b_k\Big) + \sum_{i=1}^{l} m_i \sum_{j=l}^{k} b_j\Big) \qquad (44)$$

байтов.

При практической реализации табличного подхода к построению МИМА-умножителей по большим простым модулям на определяющие основания m_0, m_1, m_2, ..., m_k МИМСС приходится налагать ограничения, диктуемые допустимыми размерами таблиц. В таблице приведены временные затраты на выполнение синтезированного алгоритма УМ.М1−УМ.М9 в мультипроцессорной СМОИ и одиночной ПЭВМ, рассчитанные согласно (39) и (40), а также полученные в соответствии с (41), (43), (44) оценки необходимой табличной памяти. Представленные данные получены в предположении, что $m_i \in [\hat{m};\, 2\hat{m}]$ ($i = \overline{1,\, k}$), причем для случая $\hat{m} = 2^{15}$. В качестве инструментальной платформы приняты процессоры типа INTEL PENTIUM 4 (3 ГГц), для которых $t_{\text{сл}} = 2$нс и $t_{\text{ч}} = 2{,}14$ нс. Это обеспечивает реализацию применяемого метода умножения по модулям МИМСС – индексного метода за время $t_{\text{МУ}} = 3t_{\text{ч}} + t_{\text{сл}} = 5{,}42$нс [21].

Указанные в таблице 1 объемы табличной памяти несколько завышены. Они представляют собой значения верхней оценки характеристики $M_{\text{Т, УМ.М}}$. Пусть $b_{\text{сум}} = \max\{b_{\text{сум, 1}},\, b_{\text{сум, 2}},\, \ldots,\, b_{\text{сум, }k}\}$. Тогда искомая оценка вытекает из (44) и (41) при $\hat{b}_1 = \hat{b}_2 = \ldots = \hat{b}_k = 15$:

$$M_{\text{Т, УМ.М}} < 2k(2^{b_{\text{сум}}-15} + 2^{15}) + 8k2^{16} + 4(k-l)2^{16} + 2(l+1)(k-l)2^{16} +$$
$$+ 2l(k-l+1)2^{16} = k(2^{b_{\text{сум}}-14} + 2^{16}) + (k-l+(4k+k-l+(l+1)(k-l)$$
$$+ l(k-l+1)))2^{17} = k(2^{b_{\text{сум}}-14} + 2^{16}) + (k-l+N_{\text{Т, УМ.М}})2^{17}.$$

Таблица 1 – Времена выполнения МИМА-алгоритма умножения по модулю на основе метода Монтгомери с использованием процессоров Intel Pentium 4 (3ГГц)

№	Параметры алгоритма и базовой МИМСС				Временные затраты на реализацию алгоритма (в нс)		Затраты табличной памяти	
	$\lceil \log_2 p \rceil$	l	k	$M_{\min}$	$t_{\text{УМ.М,ПЭВМ}}$	$t_{\text{УМ.М,СМОИ}}$	Число таблиц $N_{\text{Т,УМ.М}}$	Суммарный объем памяти (Мб) $M_{\text{Т,УМ.М}}$
1	64	5	10	$1{,}000183 \cdot 2^{132}$	295,46	62,8	105	14,375
2	128	9	18	$1{,}000427 \cdot 2^{260}$	668,98	78,8	261	34,875
3	256	17	34	$1{,}000916 \cdot 2^{516}$	1800,02	110,8	765	99,875
4	512	33	67	$1{,}001893 \cdot 2^{1028}$	5750,36	176,8	2613	335,0625
5	1024	66	133	$1{,}003910 \cdot 2^{2052}$	20183,9	308,8	9576	1213,6875
6	2462	157	315	$1{,}009480 \cdot 2^{4928}$	105121,48	672,8	51345	6457,87

В приводимых оценках значений параметров МИМСС и характеристик эффективности алгоритма УМ.М1−УМ.М9 в качестве основы принимается неравенство

$$4p < M_l < M/M_l < \frac{1}{2} M_k / M_l, \qquad (45)$$

которое вытекает из (37) и теоремы 2. Для определения границ изменения l и k достаточно применить следующую упрощенную версию (45):

$$4p < 2^{2r+2} < (\overline{m})^l < \frac{1}{2}(\overline{m}')^{k-l}, \qquad (46)$$

где $r = \lceil \log_2 p \rceil$ – разрядность модуля p; $\overline{m}$ и $\overline{m}'$ – усредненные значения оснований МИМСС по группам $< m_1, m_2, ..., m_l >$ и $< m_{l+1}, m_{l+2}, ..., m_k >$ соответственно ($\overline{m}$, $\overline{m}' \in [\hat{m}; 2\hat{m}]$).

Из (46) имеем

$$\begin{cases} r + 2 < l \log_2 \overline{m}, \\ l \log_2 \overline{m} < (k - l) \log_2 \overline{m}'. \end{cases} \tag{47}$$

Система (47) дает

$$\frac{r+2}{\log_2 \overline{m}} < l < \frac{k \log_2 \overline{m}' - 1}{\log_2 \overline{m} + \log_2 \overline{m}'}. \tag{48}$$

Отсюда для k вытекает оценка:

$$k > (l(\log_2 \overline{m} + \log_2 \overline{m}') + 1) / \log_2 \overline{m}'). \tag{49}$$

При $\hat{m} = 2^{15}$ величины $\log_2 \overline{m}$, $\log_2 \overline{m}' \in [15; 16]$. Данные, приведенные в таблице 1, рассчитаны при $\log_2 \overline{m} = \log_2 \overline{m}') = 15{,}75$. Согласно (48) и (49) для l и k в этом случае верны оценки: $l > (r+2)/15{,}75$ и $k > 2l$.

Что касается параметров M_l и M, а значит M_k, базовой МИМСС, то для них в рамках ограничений (37) могут быть получены более точные, чем используемые выше, нижние оценки.

Из первого неравенства системы (37) находим:

$$M_{l-1} > 4p/(m_l - l + 2). \tag{50}$$

Отсюда ввиду $m_l - l + 2 \le m_l - \rho_{l-1,\max}$ следует, что

$$M_l > 4p \frac{m_l}{m_l - l + 2} = 4p(1 + \frac{l-2}{m_l - l + 2}) > 4p(1 + \frac{l-2}{m_l - \rho_{l-1,\max}}). \tag{51}$$

Аналогично, с учетом (50) из второго неравенства системы (37) получаем

$$M > M_{l-1}^2 > (4p)^2 \frac{m_l(m_l + l - 2)}{(m_l - l + 2)^2} > (4p^2)(1 + \frac{l-2}{m_l - \rho_{l-1,\max}})(1 + \frac{2(l-2)}{m_l - \rho_{l-1,\max}}). \tag{52}$$

При выполнении условия $m_l - \rho_{l-1,\max} \ge 2m_0$ минимальной избыточности МСС с основаниями m_1, m_2, ..., m_l нижними порогами для M_l и M могут служить максимально возможные значения правых частей (51) и (52) относительно величины $m_l - \rho_{l-1,\max}$. В соответствии с этим в качестве искомых оценок можно принять:

$$M_l > 4p\ (1 + \frac{l-2}{2m_0})$$

и

$$M > (4p)^2 (1 + \frac{l-2}{2m_0})(1 + \frac{l-2}{m_0}). \tag{53}$$

В таблице 1 приведены значения нижнего порога $M_{\min}$ полумощности M динамического диапазона базовой МИМСС, определяемые согласно (53) для случая $m_0 = 3 \times 2^{13}$, и при замене p на 2^r, то есть значения, рассчитываемые по формуле

$$M_{\min} = 2^{2r+4}(1 + \frac{l-2}{2^{14}} + \frac{(l-2)^2}{9 \times 2^{27}}).$$

Как видно из таблицы 1 разработанная на базе метода Монтгомери МИМА-процедура умножения по большим простым модулям отличается высоким быстродействием. Временные затраты на ее выполнение в одиночной ПЭВМ и мультипроцессорной СМОИ, включающей лишь k позиционных сумматоров разрядностью, не превышающей 32 бита, находятся соответственно в микросекундном и наносекундном диапазонах. Это достигается за счет выделения из реализуемого в реальном времени вычислительного процесса трудоемких расчетов по формированию комплекта рабочих

таблиц в самостоятельный процесс, который осуществляется предварительно (независимо от модуля p), причем для фиксированного набора оснований МИМСС только один раз.

Предложенная табличная конфигурация МИМА-алгоритма умножения по модулю p позволяет свести все вычисления к операциям извлечения из табличной памяти наборов вычетов и их суммирования на позиционных сумматорах стандартной разрядности. Известные мультипликативные алгоритмы рассматриваемого класса используют 32-битовые основания, что затрудняет применение табличного метода. Это ограничивает возможности для повышения производительности. Согласно приведенным в [4, 22] оценкам реализация на однопроцессорной ЭВМ наиболее близкого МА-аналога процедуры УМ.М1–УМ.М9. В рамках принятой инструментальной платформы при $\lceil \log_2 p \rceil$ = 1024 и 2462 бита соответственно занимает 71858,16 и 363254,4 нс. Следовательно согласно данным, представленным в таблице 1, разработанный алгоритм УМ.М1–УМ.М9 в указанных случаях обеспечивает повышение быстродействия в 2,6–3,5раз.

Мультипроцессорная МА-версия алгоритма, описанного в [9], может быть выполнена за время 5244 нс. При использовании 2462-битовых p. В данном случае предлагаемое решение позволяет достичь (5244/672,8)≈7,8-кратного увеличения производительности.

При $\lceil \log_2 p \rceil$=1024 и 2462 бит объем необходимой табличной памяти для алгоритма УМ.М1–УМ.М9 не превышает соответственно 1,2 и 6,46 Гб. Эти максимальные требования можно значительно уменьшить за счет сужения базового комплекта рабочих таблиц. В частности, отказ от хранения в памяти таблиц расширения МИМК – ТЕi_j($i = \overline{1,l};\ j = \overline{l+1,k}$) и ТЕ_i_j ($i = \overline{l+1,k-1};\ \ j = \overline{1,l}$) ведет к снижению пороговых значений размера табличной памяти при рассматриваемых p соответственно до 141 и 334 Мб. При этом обеспечиваемое повышение производительности адекватной программной версией МИМА-алгоритма на основе схемы Монтгомери является двукратным.

Таким образом, благодаря внутреннему параллелизму и табличной природе МИМА, а также простоте применяемой базовой процедуры расширения кода предложенная технология синтеза мультипликативных алгоритмов для криптографических приложений позволяет существенно повысить производительность результирующих алгоритмов при приемлемых ограничительных требованиях на размер табличной памяти.

Заключение

Представленная в настоящей статье разработка по оптимизации модулярной схемы Монтгомери для умножения по большим модулям показывает, что принципиально новые возможности для решения данной проблемы обеспечивает табличная версия МИМА. Наиболее важные результаты выполненных исследований состоят в нижеследующем.

1. В целях упрощения немодульной составляющей МА-схемы Монтгомери на первом каскаде подлежащий расширению неизбыточный МК замещен интервально-модулярным кодом. Это сокращает реализационные затраты на вычисление базовой ИХМК в l/2 раз (l – число оснований соответствующей (первой) усеченной МСС).

2. Для мощностей диапазонов усеченных (первой и второй) МСС получены условия, устанавливаемые согласно разрядности модуля p мультипликативной схемы Монтгомери, которые гарантируют корректность режима многократного обращения к МИМА-процедуре умножения по модулю p и обеспечивают ((k−l)/2)-кратное уменьшение реализационных затрат на формирование ИХМК при расширении кода на втором каскаде схемы Монтгомери.

3. На базе минимально избыточной модулярной схемы Монтгомери синтезирован алгоритм умножения по модулю p, имеющий сумматорно-табличную конфигурацию. Для его выполнения требуются лишь операции извлечения наборов вычетов из таблиц и суммирования ЦЧ на позиционных сумматорах стандартной разрядности.

4. Для синтезированного алгоритма умножения по большим модулям даны оценки минимальных временных затрат на его реализацию как в мультипроцессорной СМОИ, так и в одиночной ЭВМ, а также суммарного объема необходимой табличной памяти. В случае выполнения на однопроцессорной ЭВМ разработанного алгоритма Монтгомери 1024- и 2462-битовых p теоретический минимум временных затрат достигается при пороговых значениях объема табличной памяти 1,2 и 6,5 Гб соответственно. При этом в сравнении с наиболее близким модулярным аналогом – разработкой фирмы Toshiba [9], обеспечивается не менее чем 3,6-кратное повышение производительности. При снижении предельных порогов для размера табличной памяти до 141 и 334 Мб повышение быстродействия для указанных p является двукратным.

Разработка выполнена в рамках ГКПНИ “ИНФОТЕХ” (задание “ИНФОТЕХ-19”).

Список литературы

1. Инютин, С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах / С.А. Инютин // Изв. вузов. Электрон. – 2001. – №6. – С. 65–73.

2. Инютин, С.А. Основы многоразрядной алгоритмики / С.А.Инютин. – Сургут: РИО, 2002. – 137с.

3. Четырехмодульная система модулярной обработки информации для высокоточных вычислений / А.А. Коляда [и др.] // Информатика. – 2008. – №1. – С. 18–30.

4. Чернявский, А.Ф. Умножение по большому модулю в минимально избыточной модулярной системе счисления с применением операций масштабирования / А.Ф. Чернявский [и др.] // Информатика. – 2009. – №4. – С. 49 – 65.

5. Posch, K.S. Modulo reduction in residue number system / K.S. Posch, R. Posch // IEEE Trans. on parallel and distributed syst. – 1995. – Vol 6, № 5. – P. 449–454.

6. Schwemmlein, J. RNS-modulo reduction upon a restricted base value set and its applicability to RSA cryptography // J. Schwemmlein, K.S. Posch, R. Posch // Comput. and security. – 1998. – Vol 17, №7. – P. 637–650

7. Bajart, J.-C. An RNS montgomery modular multiplication algorithm / J.-C. Bajart, L.-S. Didier, P. Kornerup // IEEE Trans. Comput. – 1998. – Vol. 47, № 7. P. 766–776.

8. Hiasat, A.A. New efficient structure for a modular multiplier for RNS / A.A. Hiasat // IEEE Trans. Comput.– 2000.– Vol. 49, № 2. P. 170–174.

9. Kawamura S. Cox-Rower architecture for fast parallel Montgomery multiplication / Shinichi Kawamura, Masanobu Koike, Fumihiko Sano, Atsushi Shimbo // Eurocrypt 2000, LNCS. – Vol. 1807. – Berlin, 2000. – P. 523–538.

10. Nozaki, H. Implementation of RSA Algorithm Based on RNS Montgomery Multiplication / H. Nozaki, M. Motoyama, A. Shimbo, S. Kawamura // Proc. Cryptographic Hardware and Embedded Systems (CHES 2001). – Sept., 2001. – P. 364–376.

11. Alia, G. Fast modular exponentiation of large number with large exponents / G. Alia, E. Martinelli // J. Syst. Archit. – 2002. – Vol. 47, № 14–15. – P. 1079–1088.

12. Lee, K.-J. Systolic multiplier for Montgomery's algorithm / K.-J. Lee, K.-J. Yoo // Integration. – 2002. – Vol. 32, № 1–2. – P. 99–109.

13. RSA speedup with residue number system immune against hardware fault cryptanalysis / S.-M. Yen [et al.] // Lect. Notes Comput. Sci. – 2002. – Vol. 2288. – P. 297–413.

14. Bajard, J.-C. A Full RNS Implementation of RSA / J.-C. Bajard, L. Imbert // IEEE Trans. Comp. – 2004. – Vol.53, № 6. – P. 769–774.

15. Lim, Z. An RNS-Enhanced microprocessor implementation of public key cryptography / Zhining Lim, B.J. Phillips // Signals, Systems and Computers. − 2007. − ACSSC 2007. Conf. Rec. of the forte-first Asilomar Conf. – 4–7 nov., 2007. – P. 1430–1434.

16. Коляда, А.А. Модулярные структуры конвейерной обработки цифровой информации / А.А. Коляда, И.Т. Пак. // Минск: Университетское. –1992. – 256 с.

17. Чернявский, А.Ф. Общая технология вычисления интегральных характеристик модулярного кода / А.Ф. Чернявский, А.А. Коляда // Доклады НАН Беларуси. – 2008. – Т.52, №4. – С. 38–44.

18. Montgomery, P.L. Modular multiplication without trial division / P.L. Montgomery // Mathematics of Computation.– 1985. – Vol. 170, №44. – P.519–521.

19. Харин, Ю.С. Компьютерный практикум по математическим методам защиты информации / Ю.С. Харин, С.В. Агиевич. // Мн.: БГУ, 2001. – 190 с.

20. Математические и компьютерные основы криптологии / Ю.С. Харин [и др.] – Мн.: Новое знание, 2003. – 382с.

21. Завгороднев, С.М. Функциональные особенности и общие принципы реализации модулярной вычислительной технологии на диапазонах большой мощности / С.М. Завгороднев [и др.] // Электроника инфо. – 2008. – №12. – С. 50–55.

22. Чернявский, А.Ф. Умножение по большим простым модулям на основе минимально избыточной модулярной схемы Барретта / А.Ф. Чернявский, А.А. Коляда // Доклады НАН Беларуси.− 2010. Т. 54, № 2 С. 40−53.

Поступила 06.05.10

Институт прикладных физических проблем
им. А.Н. Севченко БГУ,
Минск, Курчатова, 7
e-mail: shabinskaya@rambler.ru

A.A. Kolyada, A.F. Chernyavsky

УМНОЖЕНИЕ ПО БОЛЬШИМ МОДУЛЯМ С ИСПОЛЬЗОВАНИЕМ МИНИМАЛЬНО ИЗБЫТОЧНОЙ МОДУЛЯРНОЙ СХЕМЫ МОНТГОМЕРИ

MULTIPLICATION OF A LARGE MODULES WITH APPLICATION THE MINIMALLY REDUNDANT MODULAR MONTGOMERY'S SCHEME

The new prompt algorithm of multiplication on the big simple module *p,* realising the minimally redundant modular Montgomery's scheme is offered. The main distinctive feature of the developed scheme is application of intervalno-index performances and the intervalno-modular shape of numbers in base procedures of code extension. Optimisation of the synthesised multiplicative algorithm reached at the expense of it provides (3,5-3,6)-fold pinch of productivity in comparison with the closest best analogue at performance on the single-processor COMPUTER. Thus the necessary size of tabular storage in a case 1024 and 2462-bit *p* does not exceed accordingly 1,2 and 6,46 Gb. If threshold values of the size of storage of tables for specified *p* make 141 and 334 Mb the gained scoring in speed is double.

8029-182-91-15 Елена Владимировна