Использование метода Монтгомери в алгоритме быстрого возведения в степень

Яковлев Виктор Алексеевич
доктор технических наук, доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, viyak@bk.ru

Шемякин Сергей Николаевич
кандидат технических наук, доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А.Бонч-Бруевича, г. Санкт-Петербург, Россия, s4421764@ya.ru

Таров Евгений Викторович
студент Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, tarov25@mail.ru

АННОТАЦИЯ

Введение: стойкость современных криптографических систем напрямую зависит от размерности оперируемых чисел. Однако, увеличение значений данных чисел подразумевает не только большую стойкость, но и ведет к повышению вычислительных затрат для расчетов значений криптографических функций. Это объясняется необходимостью учета межразрядных переносов при реализации различных операций в полях классов вычетов. При этом наиболее распространенные операции в криптографических системах на основе полей являются умножение и возведение в степень. Причем довольно часто применяется алгоритм быстрого возведения в степень для сокращения вычислительных ресурсов. Этот алгоритм используется в таких криптосистемах, как например RSA, Диффи-Хеллмана, Эль-Гамаля. Однако вычисление значений криптографических функций в этих системах является достаточно громоздким процессом, даже с учетом вычислительной эффективности используемого в них алгоритма быстрого возведения в степень. Потому для уменьшения вычислительных ресурсов возможно применение алгоритма Монтгомери, который является одним из наиболее быстрым для реализации модульных вычислений. Общие вычислительные затраты при его использовании существенно уменьшаются из-за отсутствия необходимости в учете межразрядных переносов. **Цель работы:** в данной работе предлагается способ модернизации алгоритма быстрого возведения в степень в кольце классов вычетов по большому модулю с использованием идей Монтгомери в интересах уменьшения вычислительных затрат алгоритма. **Результаты:** предложен способ модернизации алгоритма быстрого возведения в степень. Проведено сравнение временных затрат при использовании стандартного алгоритма и при использовании методов Монтгомери. Оказалось, что скорость модульного возведения в степень с использованием предложенного алгоритма существенно выше, чем при использовании стандартных средств.

КЛЮЧЕВЫЕ СЛОВА: криптография; информационная безопасность; алгоритм Монтгомери; модульная арифметика; умножение по модулю; алгоритм быстрого возведения в степень по модулю.

Введение

В настоящее время стойкость значительной части криптосистем с общедоступным ключом основывается на использовании операций умножения и (или) возведения в степень. Они реализуются в поле классов вычетов по большому простому модулю, в поле Галуа или на точках некоторой эллиптической кривой. При этом подразумевается использование алгоритма быстрого возведения в степень [1, 2].

Данный алгоритм используется, например, в известной криптосистеме RSA. В ней сообщение m делится на части m_i, каждая из которых используется для формирования соответствующей криптограммы c_i по следующему правилу

$$c_i = m_i^e \bmod N,$$

где N и e – части открытого ключа A (см. [2, 3]).

Необходимость в этом алгоритме возникает также и при использовании криптосистемы Диффи-Хеллмана для вычисления общего ключа K по формуле:

$$K = \alpha^{yx} \bmod N,$$

где α – элемент с большим показателем в кольцах классов вычетов по модулю N (или же примитивный элемент поля, если N простое число); y и x – числа, которые формируют пользователи криптосистемы (см. [2, 3]).

Аналогично, в системе Эль-Гамаля используется алгоритм быстрого возведения в степень для вычисления части открытого ключа A по следующему правилу

$$A_i = \alpha^a \bmod N,$$

где a – случайное целое положительное число, не превосходящее значения $N - 2$, а α – примитивный элемент (см. [2, 3]).

И для формирования криптограммы C некоторой части сообщения M

$$\gamma = \alpha^k \bmod N,$$
$$\delta = M_i \alpha^{ak} \bmod N,$$

здесь γ и δ – части криптограммы C; M_i – некоторая часть изначального сообщения; k – случайное целое число, не превосходящее значения $N - 2$ (см. [2, 3]).

Необходимо отметить, что стойкость криптосистем зависит от размерности модуля N. Например, в системе Эль-Гамаля величина модуля должна составлять порядка 1024 бит для обеспечения достаточной стойкости. А в RSA более 2048 бит для высокой стойкости. Что касается систем на эллиптических кривых, в настоящее время принято, что достаточно стойкими являются системы с рекомендуемой размерностью модуля более 256 бит [4-9].

Вместе с тем необходимо указать, что вычисление значений криптографических функций в этих системах является достаточно громоздким процессом, даже с учетом вычисли-

тельной эффективности используемого в них алгоритма быстрого возведения в степень. И поэтому задача уменьшения вычислительных затрат при реализации операций в кольцах классов вычетов по большому модулю является актуальной.

Для ее решения возможно использование алгоритма Монтгомери. В соответствии с [10-16] данный алгоритм является одним из лучших для модульных вычислений. Это объясняется тем, что данный алгоритм заменяет стандартную операцию приведения по модулю на некоторую другую, очень просто реализуемую. Общие вычислительные затраты при этом существенно уменьшаются из-за отсутствия необходимости в учете межразрядных переносов.

В данной работе предлагается способ модернизации алгоритма быстрого возведения в степень в кольце классов вычетов по большому модулю с использованием идей Монтгомери для уменьшения вычислительных затрат.

Принцип работы алгоритма Монтгомери

Алгоритм Монтгомери представляет собой эффективный метод для выполнения операций с большими числами. Он применяется для ускорения умножения и возведения в степень в некотором кольце классов вычетов. Принцип работы заключается в замене умножения и деления операциями сдвига бит и сложения, что позволяет значительно повысить скорость модулярных вычислений. Это реализуется посредством замены вычислений по изначальному модулю N, который образует исходное кольцо классов вычетов Z_N, на работу с более удобным модулем R. Число R выбирается в соответствии со следующими условиями: $R > N$ и НОД(R, N) = 1, где НОД – наибольший общий делитель [17-19].

При этом данный модуль образует свое кольцо классов вычетов, которое называют кольцом Монтгомери. Обозначим его следующим образом: Z_R. R обычно выбирают как число, являющееся результатом возведения двойки в степень для возможности программной реализации сдвигов бит. Это позволяет упростить вычисления и сделать их более эффективными, обеспечивая оптимальную скорость выполнения модулярных операций.

Перед началом работы алгоритмов производят следующие предварительные вычисления [17-19]:

1. определяют модуль R;

2. при помощи расширенного алгоритма Евклида находят линейное разложение:

$$Rr - Nn = 1, \tag{1}$$

где $r = R^{-1} \bmod N$, $n = (-N)^{-1} \bmod R$;

3. вычисляют значение $R^2 \bmod N$ или $R^3 \bmod N$.

После чего производят отображение стартовых параметров из исходного кольца в кольцо Монтгомери (в дальнейшем контексте просто отображение) при помощи специальных методов. Обозначим их через следующие функции:

$$\varphi_R(x) = xr \bmod N; \tag{2}$$

$$\varphi_R(x, y) = xyr \bmod N. \tag{3}$$

Представим это в виде таблицы для лучшего понимания данного процесса – таблица 1.

Таб. 1. Сопоставление исходных параметров и их отображений

Параметры	Исходное кольцо	Кольцо Монтгомери
x	$x \bmod N$	$xr \bmod N$
xy	$xy \bmod N$	$xyr \bmod N$

Как видно из таблицы 1 – на выходе обеих функций к входным параметрам добавляется число r, получаемое в результате применения формулы (1). Это как раз и является свидетельством отображения.

Для того, чтобы вернуться в исходное кольцо (т.е. избавится от лишнего множителя r), образованное классом вычетов по модулю N, нужно применить функцию (3) следующим образом:

$$\varphi_R(\varphi_R(x), R^2) = \varphi_R(x)R^2 r \bmod N = xR^2r^2 \bmod N = x \bmod N;$$
$$\varphi_R(\varphi_R(x, y), R^2) = \varphi_R(x, y)R^2 r \bmod N = xyR^2r^2 \bmod N = xy \bmod N.$$

Рассмотрим подробнее, как устроены функции (2) и (3) «изнутри». Для этого обратимся к первой из них: $\varphi_R(x)$. Процесс отображения числа x при помощи данной функции может быть описан следующим алгоритмом [17-19]:

Алгоритм 1. Отображение $\varphi_R(x)$ одного числа x.

Пусть дано целое положительное число $x \in Z_N$. Необходимо вычислить $\varphi_R(x)$. Для этого нужно провести следующие вычисления:

1. вычислить значение $m = xn \bmod R$;
2. вычислить целое число $t = \dfrac{x + mN}{R}$;
3. положить, что $\varphi_R(x) = \begin{cases} t, t < N \\ t - N, t \geq N \end{cases}$.

Для случая, когда нужно отобразить произведения двух чисел, применяют функцию (3). Данный процесс можно описать при помощи следующего алгоритма [17-19]:

Алгоритм 2. Отображение $\varphi_R(x, y)$ произведения двух чисел x и y.

Пусть даны целые положительные числа x, $y \in Z_N$. Необходимо вычислить $\varphi_R(x, y)$. Алгоритм состоит в построении последовательности чисел z_0, z_1 ..., z_m, где $m = \lceil \log_\beta N \rceil$ (β – система счисления, в которой производятся вычисления). Здесь $z_0 = 0$, и по найденному z_i следующее число z_{i+1} вычисляется последовательно следующим образом:

1. вычислить значение $u = z_i + x_i y \bmod \beta$;
2. вычислить значение $v = un \bmod \beta$;
3. вычислить очередное значение последовательности $z_{i+1} = \dfrac{z_i + x_i y + vN}{\beta}$;
4. после того, как было вычислено значение z_m положить, что $\varphi_R(x, y) = \begin{cases} z_m, z_m < N \\ z_m - N, z_m \geq N \end{cases}$.

Анализ алгоритма быстрого возведения в степень

Алгоритм быстрого модульного возведения в степень предназначен для эффективного вычисления значения $x^k \bmod N$, где x, k и N – целые числа.

Основная идея заключается в использовании бинарного разложения показателя степени k. Алгоритм выполняет последовательные возведения в квадрат числа x и на каждом шаге проверяет биты показателя степени. Если бит равен 1, то текущий результат умножается на x и берется остаток от деления на N [20-21].

В результате применения этого подхода количество операции умножения и деления значительно сокращаются, что приводит к существенному ускорению вычислений. Вместо линейной сложности алгоритм работает за время, пропорциональное двоичному логарифму от показателя степени.

Однако даже в нем есть некоторые неоптимальные вычисления. Они связаны с нахождением остатка от деления, так как в стандартном алгоритме предполагается использование метода деления в столбик, что достаточно ресурсоемко при работе с большими числами (порядка 4000 бит). Потому для оптимизации этих вычислений необходимо применить метод Монтгомери. Для этого рассмотрим принцип работы алгоритма быстрого возведения в степень [20-21].

Алгоритм 3. Алгоритм быстрого возведения в степень по модулю.

Пусть дан целый положительный модуль N и два целых положительных числа x, $k \in Z_N$ и требуется найти $x^k \bmod N$. Для этого необходимо выполнить следующие вычисления:

1. показатель степени представить в двоичном виде:

$$k = \sum_{i=0}^{p-1} k_i 2^i \ . \tag{4}$$

где p – число бит, которое занимает двоичная запись k; в результате (4) будет получено следующее множество бит: $(k_0, k_1, ..., k_{p-1})$;

2. пройтись по битам показателя степени от младшего разряда k_0 к старшему k_{p-1} при этом сформировать либо одно целое число t_i (здесь i = 1, 2, …, p-1), если данный бит равен нулю $k_i = 0$, либо два – t_i и d_i, при условии, что $k_i = 1$.

Здесь число d изначально равно 1 ($d_0 = 1$) и варьируется в соответствии с битом k_i определяя свое значение согласно следующей формуле:

$$d_i = \begin{cases} d_{i-1}, k_i = 0 \\ d_{i-1} t_{i-1} \bmod N, k_i = 1 \end{cases}.$$

Число t изначально равно основанию степени ($t = x$) и в дальнейшем меняет свое значение независимо от множества $(k_0, k_1, ..., k_{p-1})$. Параметр t возводится в квадрат по модулю N и полученное значение присваивается уже как новое по следующей формуле:

$$t_i = t_{i-1}^{\ 2} \bmod N;$$

3. После того как все биты показателя степени были задействованы в предыдущем шаге, то положить, что результат модульного возведения в степень равен последнему значению параметра d_i:

$$d_{m-1} = x^k \bmod N .$$

Реализация быстрого возведения в степень с применением идей Монтгомери

На основании рассмотренного алгоритма быстрого возведения в степень можно сделать следующие выводы: во втором шаге алгоритма 3 для нахождения каждого d_i и t_i производятся умножения и возведения в степень по модулю N соответственно для каждой из величин. Наибольшее число операций для d_i будет равно $p-1$ при условии, что $k = 2^l - 1$, а минимальное – единице в том случае, если $k = 2^l$, где l – натуральное число. В свою очередь количество операций возведения в степень по модулю для чисел t_i всегда будет равно $p-1$ вне зависимости от значения показателя степени. Из этого следует, что при в процессе приведенных вычислений необходимо провести достаточно большое количество операций модульного умножения.

Потому алгоритм 3 можно эффективно оптимизировать при помощи методов Монтгомери следующим образом.

Алгоритм 4. Оптимизированный алгоритм быстрого возведения в степень с применением методов Монтгомери.

Пусть дан целый положительный модуль N и два целых положительных числа x, $k \in Z_N$ и требуется вычислить $x^k \bmod N$. Для этого необходимо выполнить следующие вычисления:

1. отобразить число x при помощи функции (3) и обозначить результат следующим образом:

$$z_R = \varphi_R(x, R^2 \bmod N) = xR \bmod N ;$$

2. получить множество бит показателя степени: $(k_0, k_1, ..., k_{p-1})$;

3. пройтись по множеству бит из 2 шага и сформировать числа t_i и d_i по следующим формулам:

$$t_i = \varphi_R(t_{i-1}, t_{i-1}) = t_{i-1}t_{i-1}R \bmod N ,$$

$$d_i = \begin{cases} d_{i-1}, k_i = 0 \\ \varphi_R(d_{i-1}, t_{i-1}), k_i = 1 \end{cases},$$

где $\varphi_R(d_{i-1}, t_{i-1}) = d_{i-1}t_{i-1}r \bmod N = d_{i-1}x^{2(i-1)}Rr \bmod N = d_{i-1}x^{2(i-1)} \bmod N$. При этом $d_0 = 1$, $t_0 = z_R$, $i = 1, 2, \ldots, p\text{-}1$;

4. положить, что результат модульного возведения в степень равен последнему значению d_{m-1}:

$$d_{m-1} = x^k \bmod N .$$

Сравнительный анализ обычного и модернизированного алгоритма быстрого возведения в степень

Упомянутые алгоритмы (стандартный – алгоритм 3 с использованием деления в столбик и модернизированный – алгоритм 4) были реализованные на языке программирования высокого уровня Python версии 3.10. Сравнительный анализ был проведен на процессоре Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz и на операционной системе Windows 11. Результат сравнения алгоритмов представлен на рисунке.

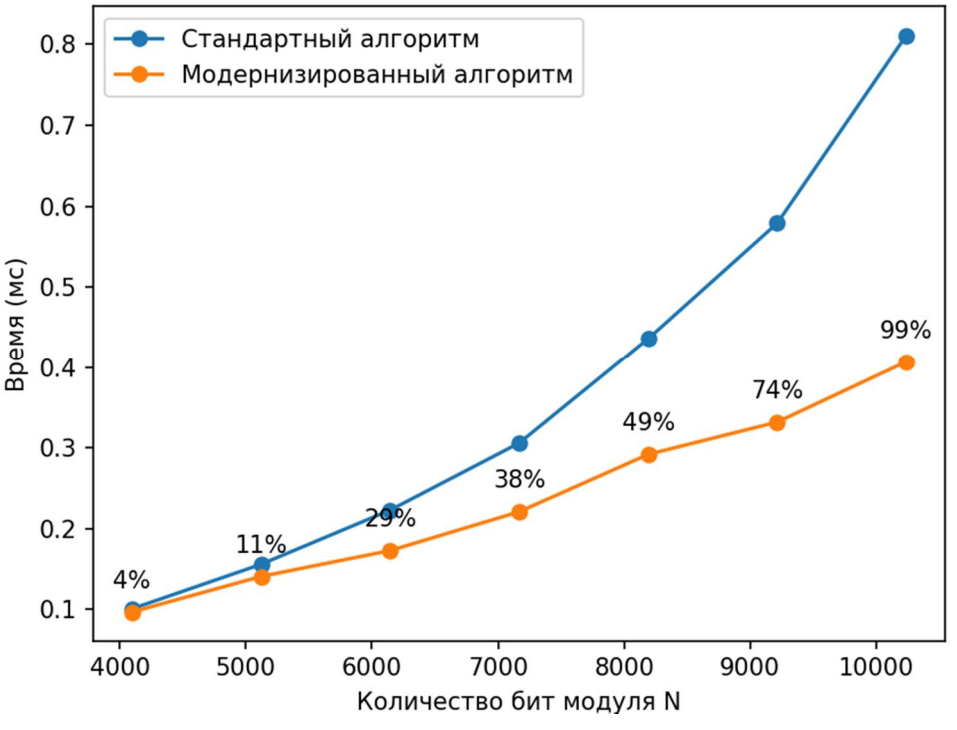

Рис. График времени работы алгоритмов

На рисунке представлен график времени работы алгоритмов. В соответствии с данными, представленными на нем, построим таблицу результатов – таблица 2.

Таблица 2. Результаты измерения

Количество бит модуля *N*	Время работы стандартного алгоритма, мс	Время работы оптимизированного алгоритма, мс	Процентное улучшение времени работы модернизированного алгоритма по сравнению с обычным
4096	0.101	0.097	4%
5120	0.156	0.141	11%
6144	0.223	0.173	29%
7168	0.306	0.221	38%
8192	0.436	0.292	49%
9216	0.579	0.332	74%
10240	0.811	0.407	99%

В соответствии с данными, представленными на рисунке и в таблице 2 можно проследить, что модернизация алгоритма быстрого возведения в степень по модулю при помощи методов Монтгомери позволила получить выигрыш во времени работы на 4% при числе разрядов 4096, а для 10240 бит данный выигрыш составил почти 100%, т.е. скорость выполнения оптимизированного алгоритма была в два раза быстрее в сравнении со скоростью вычислений стандартного. При всем этом данный выигрыш будет увеличиваться по мере роста разрядности оперируемых чисел.

Заключение

В данной работе рассмотрен принцип работы умножения и возведения в степень в поле классов вычетов по большому простому модулю с использованием идей Монтгомери. Описан модернизированный алгоритм быстрого возведения в степень по модулю с использованием данных идей. Проведен сравнительный анализ модернизированного и обычного алгоритма при помощи языка программирования Python, который показал, что применение идей Монтгомери позволило получить выигрыш во времени работы на 4% при числе разрядов 4096, а для 10240 бит – 99%. В дальнейшем необходимо провести исследования на основе данной работы и оценить выигрыш при использовании модернизированного алгоритма быстрого возведения в степень в криптосистемах RSA, Диффи-Хеллмана и Эль-Гамаля при работе с большими числами.

Литература

1. *Кнут Д.Э.* Искусство программирования. Том 1. Основные алгоритмы. М.: Мир, 1976. 735 с.
2. *Menezes A., Oorshot P., Vanstone S.* Handbook of applied cryptography. CRC, 1997. 815 с.
3. *Schneier B.* Applied Cryptography. W and S, 1994. 662 с.
4. *Тимофеева О.П., Лимаренко А.А., Усанова А.В., Фарафонова Н.А.* Применение системы RSA в задачах криптографии и исследование ее криптостойкости // Информационные системы и технологии ист-2021 : Сборник материалов XXVII Международной научно-технической конференции Нижегородский государственный технический университет им. Р.Е. Алексеева, Нижний Новгород, 23–24 апреля 2021 года / Нижегородский государственный технический университет им. Р.Е. Алексеева. Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2021. С. 532-538.
5. *Березина Е.О., Яковлев В.А.* Анализ атаки факторизации модуля в криптосистеме рша на основе моделирования алгоритма Шора с использованием квантовых // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019) : сборник научных статей VIII Международной научно-технической и научно-методической конференцияи : в 4 т., Санкт-Петербург, 27–28 февраля 2019 года. Том 1. Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2019. С. 143-148.
6. *Гаврилов К.С., Коржик В.И.* Реализация алгоритмов некоторых побочных атак на криптосистему рша // Актуальные проблемы инфотелекоммуникаций в науке и образовании : III Международная научно-техническая и научно-методическая конференция: сборник научных статей, Санкт-Петербург, 25–26 февраля 2014 года. Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2014. С. 235-241.
7. *Goyal R., Khurana M.* Cryptographic Security using Various Encryption and Decryption Method // International Journal of Mathematical Sciences and Computing. 017. Vol. 3. No 3. P. 1-11. DOI 10.5815/ijmsc.2017.03.01.
8. *Ахмеров А.С., Резяпов И.Н.* Сравнительный анализ алгоритмов асимметричного шифрования RSA и ECC // Мавлютовские чтения : материалы XV Всероссийской молодежной научной конференции: в 7 тома, Уфа, 26–28 октября 2021 года. Том 4. Уфа: Уфимский государственный авиационный технический университет, 2021. С. 326-330.

9. *Balasubramanian K., Rajakani M.* Problems in cryptography and cryptanalysis // Algorithmic Strategies for Solving Complex Problems in Cryptography, 2017. P. 23-38. DOI 10.4018/978-1-5225-2915-6.ch002.

10. *Ha J.C., Moon S.J.* A common-multiplicand method to the Montgomery algorithm for speeding up exponentiation // Information Processing Letters. 1998. Vol. 66. No 2. P. 105-107.

11. *Hong S.M., Oh S.Ye., Yoon H.* New Modular Multiplication Algorithms for Fast Modular Exponentiation // Lecture Notes in Computer Science. 1996. Vol. 1070. P. 0166.

12. *Каленик А.Н., Коляда А.А., Коляда Н.А., Чернявский А.Ф., Шабинская Е.В.* Умножение и возведение в степень по большим модулям с использованием минимально избыточной модулярной арифметики // Информационные технологии. 2012. № 4. С. 37-44.

13. *Tiountchik A.A.* Systolic modular exponentiation via Montgomery algorithm // Electronics Letters. 1998. Vol. 34. No 9. P. 874-875. DOI 10.1049/el:19980624.

14. *Trichina E., Tiountchik A.* Scalable algorithm for montgomery multiplication and its implementation on the coarse-grain reconfigurable chip // Lecture Notes in Computer Science. 2001. Vol. 2020. P. 235-249. DOI 10.1007/3-540-45353-9_18.

15. *Montgomery P.L.* Modular multiplication without trial division. // Mathematics of computation. 1985. Vol. 44, No. 170. P. 519-521.

16. *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. 2-е изд. М. : МЦНМО, 2007. 336 с.

17. *Лобес М.В., Червяков Н.И.* Повышение скорости выполнения операции модульного возведения в степень многоразрядных чисел // Инфокоммуникационные технологии. 2009. № 7. С. 8-12.

18. *Чернявский А.Ф., Коляда А.А., Коляда Н.А., Шабинская Е.В.* Умножение по большим модулям методом Монтгомери с применением минимально избыточной модулярной арифметики // Нейрокомпьютеры: разработка, применение. 2010. № 9. С. 3-8.

19. *Груздев С.В., Шилова Я.В., Ермаков К.Д.* Применение метода Монтгомери для инженерных приложений физико-математических наук // Информационные системы и технологии ист-2021 : сборник материалов XXVII Международной научно-технической конференции Нижегородский государственный технический университет им. Р.Е. Алексеева, Нижний Новгород, 23–24 апреля 2021 года / Нижегородский государственный технический университет им. Р.Е. Алексеева. Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2021. С. 513-520.

20. *Тюпко, И.Г.* Об одном методе возведения в степень в кольце классов вычетов по модулю n // Обучение фрактальной геометрии и информатике в вузе и школе в свете идей академика А. Н. Колмогорова : Материалы XVI Колмогоровских чтений: 3-й Международной научно-методической конференции, Кострома, 07–09 декабря 2021 года. Кострома: Костромской государственный университет, 2021. С. 218-222.

21. *Авдошин С. М., Набебин А.А.* Дискретная математика. Модулярная алгебра, криптография, кодирование. М.: ДМК Пресс, 2017. 352 с.

USING THE MONTGOMERY METHOD IN A FAST EXPONENTIATION ALGORITHM

VIKTOR A. YAKOVLEV
PhD, Associate Professor at the Department of secure communication systems
St. Petersburg, Russia, viyak@bk.ru

SERGEY N. SHEMYAKIN
PhD, Associate Professor at the Department of secure communication systems
St. Petersburg, Russia, s4421764@ya.ru

EVGENIY V. TAROV
St. Petersburg, Russia, tarov25@mail.ru

ABSTRACT

Introduction: the durability of modern cryptographic systems directly depends on the dimension of the numbers being operated. However, an increase in the values of these numbers implies not only greater durability, but also leads to an increase in computational costs for calculating the values of cryptographic functions. This is explained by the need to take into account inter-bit transfers when implementing various operations in the fields of deduction classes. At the same time, the most common operations in field-based cryptographic systems are multiplication and exponentiation. Moreover, a fast exponentiation algorithm is often used to reduce computing resources. This algorithm is used in such cryptosystems as, for example, RSA, Diffie-Hellman, El Gamal. However, calculating the values of cryptographic functions in these systems is a rather cumbersome process, even taking into account the computational efficiency of the fast exponentiation algorithm used in them. Therefore, to reduce computing resources, it is possible to use the Montgomeri algorithm, which is one of the fastest for implementing modular computing. The total computational costs when using it are significantly reduced due to the lack of need to account for inter-bit transfers. **Problem statement:** in this paper, we propose a way to modernize the algorithm for rapid exponentiation in the ring of residue classes by a large modulus using Montgomery's ideas in order to reduce the computational costs of the algorithm. **Results:** a method for upgrading the algorithm of rapid exponentiation is proposed. A comparison of the time spent using the standard algorithm and using Montgomery methods is carried out. It turned out that the speed of modular exponentiation using the proposed algorithm is significantly higher than when using standard tools.

Keywords: cryptography; information security; Montgomery algorithm; modular arithmetic; multiplication modulo; algorithm of rapid exponentiation modulo.

REFERENCES

1. Knut D.E. The art of programming. Volume 1. Basic algorithms Moscow: Mir, 1976. 735 p.
2. Menezes A., Oorshot P., Vanstone S. Handbook of applied cryptography. CRC, 1997. 815 p/
3. Schneier B. Applied Cryptography. W and S, 1994. 662 p.
4. Timofeeva O.P., Limarenko A.A., Usanova A.V., Farafonova N.A. Application of the RSA system in cryptography tasks and research of its cryptographic strength. Information systems and technologies ist-2021 : collection of materials of the XXVII International Scientific and Technical Conference Nizhny Novgorod State Technical University named after R.E. Alekseev, Nizhny Novgorod, April 23-24, 2021. Nizhny Novgorod State Technical University named after R.E. Alekseev. Nizhny Novgorod: Nizhny Novgorod State Technical University named after R.E. Alekseev, 2021. Pp. 532-538.
5. Berezina E.O., Yakovlev V.A. Analysis of the module factorization attack in the RSHA cryptosystem based on modeling of the Shore algorithm using quantum. Actual problems of infotelecommunications in science and education (APINO 2019) : collection of scientific articles of the VIII International Scientific-Technical and Scientific-methodological Conference : in 4 volumes, St. Petersburg, February 27-28, 2019. Volume 1. St. Petersburg: St. Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruevich, 2019. Pp. 143-148.
6. Gavrilov K.S., Korzhik V.I. Implementation of algorithms of some side attacks on the rsha cryptosystem. Actual problems of infotelecommunications in science and education : III International scientific-technical and scientific-methodical conference: collection of scientific articles, St. Petersburg, February 25-26, 2014. St. Petersburg: St. Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruevich, 2014. Pp. 235-241.
7. Goyal R., Khurana M. Cryptographic Security using Various Encryption and Decryption Method. International Journal of Mathematical Sciences and Computing. 2017. Vol. 3. No 3. Pp. 1-11. DOI 10.5815/ijmsc.2017.03.01.
8. Akhmerov A.S., Rezyapov I.N. Comparative analysis of RSA and ECC asymmetric encryption algorithms. Mavlyutov readings: Proceedings of the XV All-Russian Youth Scientific Conference: in Volume 7, Ufa, October 26-28, 2021. Volume 4. Ufa: Ufa State Aviation Technical University, 2021. Pp. 326-330.
9. Balasubramanian K., Rajakani M. Problems in cryptography and cryptanalysis. Algorithmic Strategies for Solving Complex Problems in Cryptography, 2017. Pp. 23-38. DOI 10.4018/978-1-5225-2915-6.ch002.
10. Ha J.C., Moon S.J. A common-multiplicand method to the Montgomery algorithm for speeding up exponentiation. Information Processing Letters. 1998. Vol. 66. No 2. Pp. 105-107.
11. Hong S.M., Oh S.Ye., Yoon H. New Modular Multiplication Algorithms for Fast Modular Exponentiation. Lecture Notes in Computer Science. 1996. Vol. 1070. P. 0166.
12. Kalenik A.N., Kolyada A.A., Kolyada N.A., Chernyavsky A.F., Shabinskaya E.V. Multiplication and exponentiation by large modules using minimally redundant modular arithmetic. Information technologies. 2012. No. 4. Pp. 37-44.
13. Tiountchik A.A. Systolic modular exponentiation via Montgomery algorithm. Electronics Letters. 1998. Vol. 34. No 9. Pp. 874-875. DOI 10.1049/el:19980624.
14. Trichina E., Tiountchik A. Scalable algorithm for montgomery multiplication and its implementation on the coarse-grain reconfigurable chip. Lecture Notes in Computer Science. 2001. Vol. 2020. Pp. 235-249. DOI 10.1007/3-540-45353-9_18.
15. Montgomery P.L. Modular multiplication without trial division. Mathematics of computation. 1985. Vol. 44, No. 170. Pp. 519-521.
16. Vasilenko O.N. Number-theoretic algorithms in cryptography. 2nd ed. M. : ICNMO, 2007. 336 p.
17. Lobes M.V., Chervyakov N.I. Increasing the speed of performing the operation of modular exponentiation of multi—digit numbers. Infocommunication technologies. 2009. No. 7. Pp. 8-12.
18. Chernyavsky A.F., Kolyada A.A., Kolyada N.A., Shabinskaya E.V. Multiplication by large modules by the Montgomery method using minimally redundant modular arithmetic. Neurocomputers: development, application. 2010. No. 9. Pp. 3-8.

19. Gruzdev S.V., Shilova Ya.V., Ermakov K.D. Application of the Montgomery method for engineering applications of physical and mathematical sciences. Information systems and technologies ist-2021 : collection of materials of the XXVII International Scientific and Technical Conference Nizhny Novgorod State Technical University named after R.E. Alekseev, Nizhny Novgorod, April 23-24, 2021. Nizhny Novgorod State Technical University named after R.E. Alekseev. Nizhny Novgorod: Nizhny Novgorod State Technical University named after R.E. Alekseev, 2021. Pp. 513-520.
20. Tyupko, I.G. On one method of exponentiation in the ring of residue classes modulo. Teaching fractal geometry and computer science at university and school in the light of the ideas of Academician A. N. Kolmogorov : materials of the XVI Kolmogorov readings: 3rd International Scientific and Methodological Conference, Kostroma, 07-09 December 2021. Kostroma: Kostroma State University, 2021. Pp. 218-222.
21. Avdoshin S. M., Nabebin A.A. Discrete mathematics. Modular algebra, cryptography, coding. M.: DMK Press, 2017. 352 p.