СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Ишмухаметов, Ш.Т. Методы факторизации натуральных чисел: учебное пособие / Ш.Т. Ишмухаметов. – Казань: Казан. ун., 2011. – 190 с.
2. Venturi, D. Lecture Notes on Algorithmic Number Theory / D. Venturi. – New-York, Berlin: Springer-Verlag, 2009. – 217 p.
3. Kozaczko, D. Vector Module Exponential in the Remaining Classes System / D.Kozaczko, M.Kasianchuk, I.Yakymenko, S.Ivasiev // Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS–2015). - Warsaw, Poland. – V.1, September – 2015. – P.161–163.
4. Kasianchuk, M. Efficient methods for modular multiplication through the use of Rademacher – Krestenson TNB/ M. Kasianchuk, I. Yakymenko, Ya. Nykolaychuk, S. Ivasiev // Modern Problems of RadioEngineering, Telecommunications and ComputerScience (TCSET–2014): proceedingsofthe XI–th International Conference – L'viv–Slavske. – 2014. – P. 93–94.

Материал поступил в редакцию 14.01.2018

IVASIEV S.V. Method of high-probabilistic determination of simple multi-discharge numbers based on vector-module multiplication

The test for checking on the simplicity of multidigit numbers with using the method of vector and modular multiplication, which is characterized by high performance and low computational complexity in comparison with known is designed in this article. The block diagram of algorithm operation and its gradually implementation is presented.

УДК 581.3

Касянчук М.Н.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ОПЕРАЦИИ УМНОЖЕНИЯ В ТРЁХМОДУЛЬНОЙ СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Введение. В настоящее время всё больше внимания уделяется разработке алгоритмов распараллеливания процессов выполнения арифметических операций [1], объёмы которых и значения соответствующих чисел растут значительными темпами. Особенно это касается асимметричной криптографии (криптосистемы RSA, Рабина, Эль-Гамаля, алгоритмов электронной цифровой подписи, шифрования на эллиптических кривых [2]), кодирования информации [3], обработки изображений, других задач теории чисел, дискретной и прикладной математики. Использующаяся на данный момент двоичная система исчисления имеет строго последовательную структуру, что ограничивает её возможности при параллельной обработке информации. Для этих целей целесообразно применять непозиционные системы исчисления, одной из которых является система остаточных классов (СОК) [4]. Хотя она тоже не лишена недостатков, главными из которых являются трудности при выполнении операций деления и сравнения, однако её успешно можно использовать для распараллеливания процессов при сложении, умножении и возведении в степень многоразрядных чисел.

Теоретические основы системы остаточных классов, её совершенной и модифицированной совершенной форм. Теоретической основой СОК является теория чисел [5]. Любое целое десятичное число N представляется в СОК в виде набора (b_1, b_2, … , b_s) наименьших положительных остатков от деления этого числа на фиксированные натуральные попарно взаимно простые числа (модули) p_1, p_2, … , p_s ($b_i=N \bmod p_i$), где s – количество модулей. При этом должно выполняться неравенство $0 \le N < P-1$, где $P=\prod_{i=1}^{s} p_i$ – число, которое определяет условие переполнения разрядности вычислений. Арифметические операции (сложение, умножение, возведение в степень) выполняются отдельно по каждому малоразрядному модулю, после чего полученные результаты преобразуются в десятичную систему исчисления с помощью китайской теоремы об остатках:

$$N=\left(\sum_{i=1}^{s} b_i B_i\right) \bmod P, \qquad (1)$$

где $B_i=M_i m_i$, $M_i=\dfrac{P}{p_i}$, $m_i=M_i^{-1} \bmod p_i$.

Нахождение обратных элементов по модулю характеризуется значительной вычислительной сложностью и в теории чисел реализуется полным перебором возможных вариантов, с помощью алгоритма Евклида или теоремы Эйлера [6]. В работе [7] описана совершенная форма (СФ) СОК, в которой выполняется условие $M_i \bmod p_i = 1$, что позволяет избежать процедуры поиска обратного элемента и умножения в (1) на базисные числа m_i. Выражение (1) в этом случае упрощается:

$$N=\left(\sum_{i=1}^{s} b_i M_i\right) \bmod P. \qquad (2)$$

Однако в этом случае $p_1=2$, $p_2=3$ и остальные значения p_i быстро увеличиваются, что неприемлемо при необходимости использования модулей приблизительно одинаковой разрядности.

В [8] предложена модифицированная совершенная форма (МСФ) СОК, в которой $M_i \bmod p_i = \pm 1$, что также исключает выполнение операции поиска обратного элемента. Вычисления (1) происходят согласно формуле

$$N=\left(\sum_{i=1}^{s} b_i m_i M_i\right) \bmod P, \qquad (3)$$

где $m_i=\pm 1$.

В [9] представлены теоретические основы построения трёхмодульной МСФ СОК. Однако в настоящее время отсутствуют экспериментальные исследования выполнения арифметических операций, в частности, умножения в СОК и её МСФ, что и составляет цель настоящей работы.

Экспериментальные исследования программной реализации системы остаточных классов и ее модифицированной совершенной формы. Для программной реализации операции умножения в СОК и МСФ СОК был выбран высокоуровневый язык программирования общего назначения Python, который ориентирован на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они, в свою очередь, могут быть объединены в пакеты). Пример ввода входных параметров представлен на рисунке 1.

Результаты размещаются в файл с расширением .csv, имя которого записано в последней строчке главного окна и включает в себя все входные параметры.

Касянчук Михаил Николаевич, *к.ф.-м.н., доцент, доцент кафедры компьютерной инженерии Тернопольского национального экономического университета, e-mail: kasyanchuk@ukr.net.*
Украина, ТНЭУ, 46000, г. Тернополь, ул. Львовская, 11.

Рисунок 1 – Главное окно программы

На рисунке 2 представлены временные характеристики выполнения операции умножения $N=p\cdot q$ в трёхмодульной СОК при фиксированном множителе p=65536 с двумя разными системами модулей (первый случай – модули мало отличаются друг от друга: p_1=1625=$\left[\sqrt[3]{65536^2}\right]$, p_2=1626, p_3=1627 – пунктирные линии; второй случай – модули отличаются сильно: p_1=163, p_2=1627, p_3=16381 – сплошные линии). Второй множитель q изменялся от значения 67 до p с шагом 1311. Последний определял количество полученных вычислений, которое равнялось 50. Произведение модулей обеих систем превышает 2^{32}.

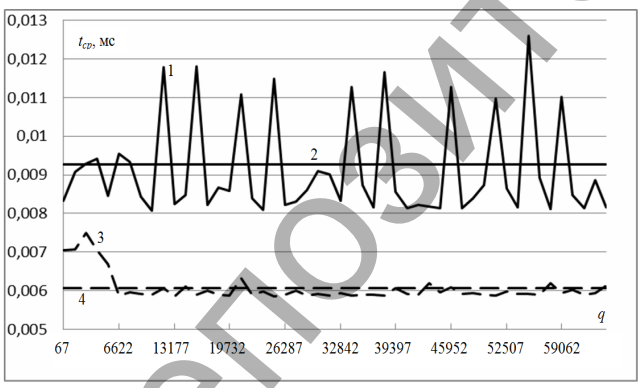

Рисунок 2 – Временные характеристики выполнения операции умножения в трёхмодульной СОК

Как видно из рисунка 2, график 1 носит осциллирующий характер. Среднее время выполнения операции умножения (линия 2) равняется 0,009259 мс. Во втором случае, кроме начальных значений q, время выполнения умножения (график 3) не испытывает существенных колебаний. Среднее время (линия 4) составляет 0,006066 мс, что в 1,53 раза меньше, чем в предыдущем случае. Поэтому для повышения быстродействия в СОК попарно взаимно простые модули необходимо выбирать так, чтобы они как можно меньше отличались друг от друга.

Для исследования МСФ СОК система модулей со значительной разницей между ними (p_1=651, p_2=691, p_3=11246) выбиралась по формуле, полученной в [8]:

$$p_3 = p_1 + \frac{p_1^2 \pm 1}{p_2 - p_1}. \qquad (4)$$

При построении трёхмодульной МСФ СОК по формуле (4) невозможно выбрать систему модулей одинаковой разрядности. Наименьшее отличие между модулями будет при таком условии:

$$p_{2,3} = 2p_1 \pm 1. \qquad (5)$$

Исходя из этого, были выбраны такие модули: p_1=1025, p_2=2049, p_3=2051. Опять же произведение модулей в обоих случаях превышает 2^{32}

Входящие параметры были те же, что и для обыкновенной СОК. Вычисления проводились согласно выражению для МСФ СОК, которое следует из (3):

$$N = \left(-b_1M_1 + b_2M_2 + b_3M_3\right) \bmod P. \qquad (6)$$

Полученные результаты представлены на рисунке 3. Сплошная линия показывает время выполнения умножения (кривая 1) и среднее время (линия 2) для 50 значений p при p_1=651, p_2=691, p_3=11246, пунктирная (графики 3, 4) – соответственно для p_1=1025, p_2=2049, p_3=2051.

Видно, что в обоих случаях при малых значениях q амплитуда колебаний большая, при увеличении q она уменьшается за исключением небольшого отрезка во второй половине диапазона изменений значения q. Среднее время для системы модулей p_1=651, p_2=691, p_3=11246 составляет 0,002293 мс (линия 2), а для p_1=1025, p_2=2049, p_3=2051 – 0,002169 мс (линия 4), что в 1,057 раза меньше, чем в предыдущем случае. Сравнение рисунков 2, 3 показывает существенное повышение быстродействия за счёт использовании МСФ СОК.

Дальнейшее исследование проводилось для чисел, разрядность n которых изменялась от 16 до 24 бит. Были рассмотрены четыре случая построения системы модулей:

1) модули СОК сильно отличаются друг от друга;

2) модулями являются три последовательных числа, первое и третье из которых нечётные: $p_1 \approx \left[\sqrt[3]{2^{2n}}\right]$, $p_2 = p_1+1$, $p_3 = p_1+2$;

3) модули вычисляются по формулам $p_2 = p_1+1$, $p_3 = p_1(p_1+1)-1$;

4) модули вычисляются по формулам $p_2 = 2p_1-1$, $p_3 = 2p_1+1$.

Таблица 1 – Наборы модулей и среднее время вычислений для чисел разных разрядностей

n		16	17	18	19	20	21	22	23	24
Случай 1	p_1	163	235	341	501	737	1093	1627	2429	3641
	p_2	1627	2587	4097	6503	10323	16387	26009	41287	65539
	p_3	16381	28413	49165	84541	144523	245807	416147	701881	1179703
$t_{ср}$, мкс		8,154	8,562	8.526	9,319	9,28	9,671	9,749	10,105	10,69
Случай 2	p_1	1625	2581	4095	6501	10321	16385	26007	41285	65537
	p_2	1626	2582	4096	6502	10322	16386	26008	41286	65538
	p_3	1627	2583	4097	6503	10323	16387	26009	41287	65539
$t_{ср}$, мкс		5,891	5,95	5,962	5,966	5,934	5,93	5,982	7,185	7,304
Случай 3	p_1	256	362	512	724	1024	1448	2048	2896	4096
	p_2	257	363	513	725	1025	1449	2049	2897	4097
	p_3	65791	131405	262655	524899	1049599	2098151	4196351	8389711	16781311
$t_{ср}$, мкс		5,461	5,98	5,618	5,689	5,65	5,684	5,57	5,732	5,593
Случай 4	p_1	1025	1626	2581	4097	6502	10322	16385	26008	41286
	p_2	2049	3251	5161	8193	13003	20643	32769	52015	82571
	p_3	2051	3253	5163	8195	13005	20645	32771	52017	82573
$t_{ср}$, мкс		5,551	5,351	5,33	5,364	5,365	5,44	5,956	5,959	6,679

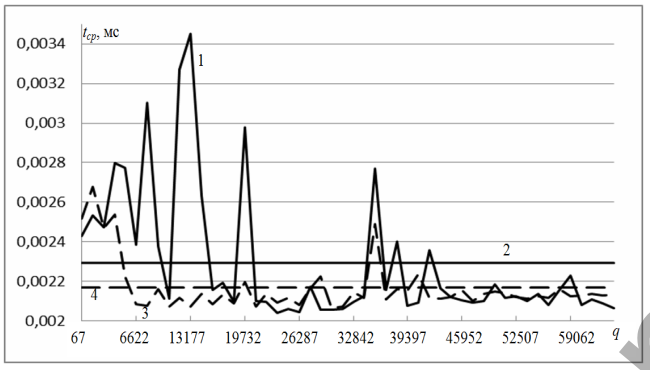

Рисунок 3 – Временные характеристики выполнения операции умножения в трёхмодульной МСФ СОК

Во всех случаях произведение модулей является минимальным, но превышающим 2^{2n}. В третьем и четвёртом случаях системы модулей образуют МСФ СОК. Первый множитель в произведении $N=p\cdot q$ был фиксированным: $p=2^n-1$, что соответствует максимальному числу заданной разрядности. Второй множитель q изменялся от начального значения $q = 2^n - \left[\frac{2^n}{1000}\right] + 1$ с шагом $\left[\frac{2^n}{1000}\right]$. Таким образом, получено 1000 различных значений числа q и соответственно время выполнения 1000 умножений $N=p\cdot q$ при фиксированном p и бегущем q. Далее для каждой разрядности определялось среднее время выполнения операции. Для нивелирования случайных влияний на работу компьютера все вычисления повторялись 100 раз. Используемые наборы модулей и среднее время вычислений для чисел разных разрядностей представлены в таблице 1.

На рисунке 4 представлены графики зависимости среднего времени выполнения операции умножения от разрядности n используемых чисел согласно таблице 1 (номер графика соответствует номеру случая в таблице 1).

Из рисунка 4 следует, что наибольшее время затрачивается для обыкновенной СОК в первом случае, когда модули сильно отличаются. Причём график растёт практически линейно с увеличением разрядности. Графики 2 и 4 при малых разрядностях почти линейные, временные скачки наблюдаются при n=22 и n=23 соответственно. И третий график практически линейный на всём рассматриваемом диапазоне. Анализ рисунка 4 свидетельствует о том, что использование модулей, которые или образуют МСФ СОК, или мало отличаются друг от друга, позволяет увеличить быстродействие вычислительных систем.

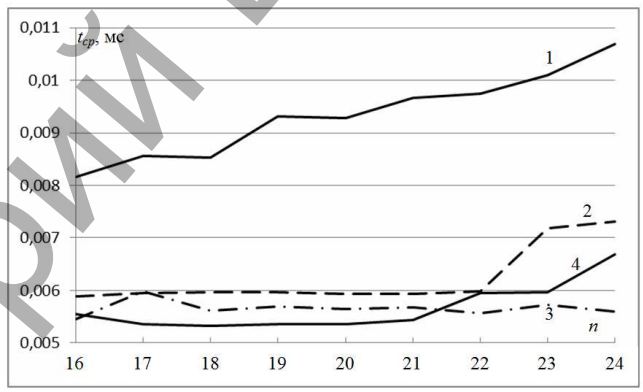

Рисунок 4 – Графики зависимости среднего времени выполнения операции умножения в СОК от разрядности n используемых чисел

На рисунке 5 представлены графики зависимости среднего времени выполнения операции умножения от разрядности n для третьего (кривая 1) и четвёртого (кривая 2) случая таблицы 1, модули в которых образуют МСФ СОК, при тех же входных параметрах с использованием формулы (6).

Анализ рисунков 4, 5 показывает, что среднее время вычислений в МСФ СОК уменьшается приблизительно в 2,5–3 раза в сравнении со случаем использования тех же систем модулей и формулы (1).

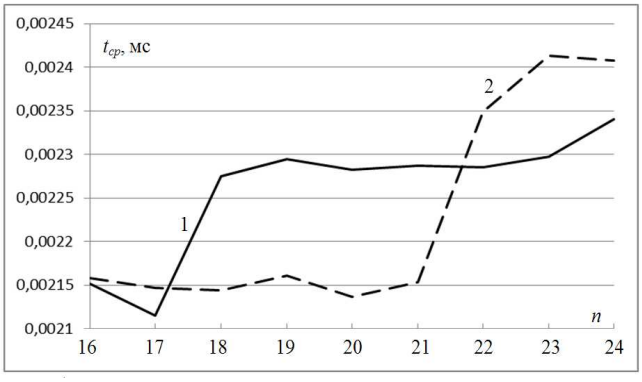

Рисунок 5 – Графики зависимости среднего времени выполнения операции умножения от разрядности n при использовании формулы (6)

Заключение. В данной работе проведено экспериментальное исследование временных характеристик программной реализации операции умножения в трёхмодульной системе остаточных классов и её модифицированной совершенной форме. Показано, что использование последней позволяет существенно уменьшить время процесса вычислений за счёт исключения выполнения операции поиска обратного элемента по модулю и умножения на него при переводе в десятичную систему исчислений. Представлены графические зависимости временных характеристик, которые подчёркивают преимущества использования модифицированной совершенной формы системы остаточных классов.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Николайчук, Я.Н. Теория источников информации / Я.Н. Николайчук. – Тернополь : ТНЭУ, 2008. – 536 с.
2. Задирака, В.К. Компьютерная криптология / В.К. Задирака, А.С. Олексюк. – Тернополь : ТАНХ, 2002. – 504 с.
3. Yatskiv, V. The Use of Modified Correction Code Based on Residue Number System in WSN / V. Yatskiv, N. Yatskiv, Su Jun, A. Sachenko, Hu Zhengbing // Proceedings of the 7-th2013 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2013). – Berlin, Germany. – 2013. – Volume 1. – P. 513–516.
4. Omondi, A. Residue number systems: theory and implementation / A. Omondi, B. Premkumar. – London: Imperial College Press, 2007. – 296 p.
5. Бухштаб, А.А. Теория чисел / А.А. Бухштаб. – Москва : Просвещение, 1966. – 384 с.
6. Вербицкий, О.В. Вступление в криптологию / О.В. Вербицкий. – Львов : Научно-техническая литература, 1998. – 248 с.
7. Kasianchuk, M. Algorithms of findings of perfect shape modules of remaining classes system / M.Kasianchuk, I.Yakymenko, I.Pazdriy, O.Zastavnyy // The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2015): proceedings of the XIII-th International Conference – Polyana-Svalyava (Zakarpattya), Ukraine. – 2015. – P. 168–171.
8. Nykolaychuk, Ya.M. Theoretical Foundations of the Modified Perfect Form of Residue Number System / Ya.M. Nykolaychuk, M.M. Kasianchuk, I.Z. Yakymenko // Cybernetics and Systems Analysis. – 2016. – Vol. 52, № 2. – P. 219–223.
9. Kasianchuk, M.N. Theory and Methods of Constructing of Modules System of the Perfect Modified Form of the System of Residual Classes / M.N. Kasianchuk, Ya.N. Nykolaychuk, I.Z. Yakymenko // Journal of Automation and Information Sciences. – 2016. – Vol. 48, № 8. – P. 56–63.

Материал поступил в редакцию 14.01.2018

KASIANCHUK M.N. The experimental research of software implementation of multiplication operation in the three-module system of residual classes

The present work is devoted to an experimental study of the time characteristics of the software implementation of the multiplication operation in the three-module system of residual classes and its modified perfect form. It has shown that the utilization of the modified perfect form allowed significantly reduce the time of the calculation process by excluding of the execution of the searching operation of the inverse element by the module and multiplying it by transition to the decimal system of calculation. Graphic dependencies of time characteristics were presented and they were emphasized the advantages of the modified perfect form of the residual class system.

УДК 004.056.53+004.492.3

Комар М.П.

ПОВЫШЕНИЕ БЕЗОПАСНОСТИ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Введение. На сегодняшний день для защиты компьютерных систем от вторжений, в основном, используется специализированное программное обеспечение (ПО) [1–9]. Однако такое ПО является уязвимым к самим вторжениям, из-за того, что могут перехватываться системные функции операционной системы, которая дает возможность активно противодействовать обнаружению и удалению их программными средствами. При этом вторжения способны блокировать запуск специализированного ПО, отслеживать его действия и возобновлять удаленные вредные процессы, изменять настройку в системном реестре и т. п.

В последнее время значительное развитие получили исследования в области коллективного (роевого) интеллекта (Swarm Intelligence) [10]. Данный научное направление возникло в рамках направления искусственного интеллекта и описывает коллективное поведение децентрализованной самоорганизующейся адаптивной системы. Системы коллективного интеллекта – это мультиагентные системы, состоящие из множества агентов, которые взаимодействуют между собой и с окружающей средой, образуя таким образом коллективный интеллект. В основу систем коллективного интеллекта положены биологические основы поведения животных и на основе наблюдений доказано, что групповой интеллект зачастую превосходит умственные способности одной особи.

В отличие от классического подхода искусственного интеллекта, по которому для определенной задачи создается одна интеллектуальная система, автором предложено использовать мультиагентный подход для обнаружения и классификации вторжений, где один агент имеет неполное представление о глобальной угрозе, поэтому создается множество агентов (детекторов атак) и обеспечивается эффективное взаимодействие между ними. Глобальное поведение всей системы рассматривается как результат взаимодействия множества простых агентов.

В данной работе в качестве агента (детектора атак) предложено использовать архитектуру нейронной сети [11, 12], которая разработана в [13–16]. Предложен подход интеграции нейросетевых агентов в искусственную иммунную систему. Реализация данного подхода осуществляется на основе базовых принципов и механизмов биологической иммунной системы: генерация и обучение иммунных агентов, отбор агентов, которые по определенным причинам генерируют ошибочные решения, функционирование агентов, активация агентов и формирования иммунной памяти; и на основе базовой схемы искусственной иммунной системы [17–19].

Итак, в интеллектуальной системе обнаружения вторжений применяется совокупность агентов [20–22], где каждый агент отвечает за обнаружение и классификацию вторжений определенного вида, а совокупность таких агентов выполняет защиту системы в целом. Однако программная реализация такого подхода имеет все недостатки программной защиты, которые указаны выше.

Предлагаемый подход к повышению безопасности системы обнаружения вторжений. Предлагается защищать компьютерные системы от вторжений аппаратными средствами. Аппаратное решение работает не в среде зараженной операционной системы, потому все действия вторжений будут безрезультатными, а вторжения будут

Комар Мирослав Петрович, *к.т.н., доцент кафедры информационно-вычислительных систем и управления Тернопольского национального экономического университета, e-mail:* mko@tneu.edu.ua.
Украина, ТНЭУ, 46000, г. Тернополь, ул. Львовская, 11.