УДК 004.41

КОНЦЕПТУАЛЬНАЯ ОРГАНИЗАЦИЯ ЗАЩИЩЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЫ С ВНУТРЕННИМ КОДИРОВАНИЕМ НА ОСНОВЕ НЕПОЗИЦИОННОЙ АРИФМЕТИКИ

Л. А. Васин

CONCEPTUAL ORGANIZATION OF PROTECTED COMPUTING STRUCTURE WITH INTERNAL CODING BASED ON OPPOSITE ARITHMETICS

L. A. Vasin

Аннотация. *Актуальность и цели.* Защита от потери данных путем получения несанкционированного доступа к вычислительным системам является актуальной проблемой, возникающей как при локальном применении, так и в составе систем передачи данных. Аппаратные и программные средства ограничения доступа не всегда позволяют обеспечить надежную защиту такой вычислительной структуры. Также существует возможность получения защищаемой информации путем съема и анализа потоков данных в вычислительной структуре, особенно распределенной архитектуры. В работе показан способ построения защищенной параллельной вычислительной системы с внутренним кодированием потоков данных, основанной на использовании непозиционной машинной арифметики. Приводится порядок функционирования такой вычислительной структуры. Целью работы являются разработка и исследование структурной организации вычислительных архитектур, специализированных операционных систем. *Материалы и методы.* Теоретическое исследование базируется на известных положениях организации вычислительного процесса и положений системы остаточных классов. *Результаты.* Применение непозиционной арифметики, в частности системы остаточных классов, как базового принципа кодирования данных в информационных системах на основе уникального ключа, в роли которого выступает выбранное основание. Выбор основания осуществляется оператором и выбирается случайным образом, смена которого полностью изменяет данные в вычислительном процессе. Дополнительное введение в вычислительную структуру блока потокового кодировщика или использование для этого дополнительных процессорных устройств позволяет осуществить необходимое потоковое преобразование входных и выходных данных из позиционной системы в непозиционную. *Выводы.* Предложенный подход к построению одного из вариантов защищенной информационной системы позволяет осуществить защиту вычислительного процесса от получения оригинальных данных и организовать его с большим параллелизмом, чем и обусловлено использование непозиционной арифметики.

Ключевые слова: архитектура вычислительной системы, организация вычислительного процесса, защита данных, непозиционная машинная арифметика, система остаточных классов, структурная организация операционной системы.

Abstract. *Background*. Protection from data loss by gaining unauthorized access to computer systems is an actual problem that arises in the local application, as well as in the structure of data transmission systems. Hardware and software access restrictions do not always provide reliable protection for such a computing structure. It is also possible to obtain protected information by removing and analyzing data flows into a computational

structure, especially a distributed architecture. The paper shows a method for constructing a protected parallel computing system with internal coding of data streams based on the use of non-position machine arithmetic. The order of functioning of such a computational structure is given. The aim of the work is to develop and study the structural organization of computing architectures, specialized operating systems. Materials and methods. Theoretical research is based on known provisions of the organization of the computational process and the positions of the residual class system. Results. The use of non-position arithmetic, in particular the residual class system, as the basic principle of data encoding in information systems based on a unique key, in whose role the chosen basis is. The choice of the base is carried out by the operator and is selected at random, the change of which completely changes the data in the computational process. An additional introduction to the computing structure of the stream encoder unit or the use of additional processor devices for this allows for the necessary streaming of the input and output data from the positional system to the non-positioning system. Conclusions. The proposed approach to the construction of one of the variants of the protected information system makes it possible to protect the computational process from obtaining original data and organize it with greater parallelism, which is the reason for using non-position arithmetic.

Key words: architecture of the computer system, organization of the computing process, data protection, non-position machine arithmetic, residual class system, structural organization of the operating system.

Введение

Важность защиты конфиденциальных данных в локальных и распределенных информационных системах является актуальной задачей, от решения которой зависит нормальное функционирование многих секторов современной экономики. Развитие систем защиты связано с технологическими достижениями в области математики, вычислительной техники. Существуют различные способы защиты данных, работающих по принципу кодирования хранящихся в вычислительной системе (ВС) данных. При этом процесс кодирования разделяется на зашифрование, выполняемое при передаче данных по каналам передачи данных или отправкой на хранение и расшифрование, обратный процесс восстановления данных к исходному состоянию. Вычислительный процесс (ВП) в таких системах использует исходные, незащифрованные данные. Существует теоретическая возможность получения доступа к данным, а также к используемым кодирующим ключам, например, путем прерывания и «заморозки» ВП и сохранения образа вычислительного процесса на устройства хранения, например, с использованием программного обеспечения (ПО) CRIU или CryoPID для ОС Linux. ПО CryoPID позволяет «заморозить» функционирующее приложение и скопировать его образ, находящийся в памяти, на внешний носитель с последующим запуском в другой ВС. ПО CRUID позволяет создавать образ работающей программы в произвольный момент времени и передачу его на выполнение в другую ВС, в том числе виртуальную. Такие программные способы позволяют организовать злоумышленнику возможность работы с конфиденциальными данными. Предлагается распространить процесс кодирования данных по ключевой информации на весь ВП с изменением позиционной машинной арифметики на непозиционную, в частности систему остаточных классов.

Постановка задачи

Данная статья исследовательского характера. В ходе изучения были рассмотрены публикации, касающиеся использования непозиционных систем арифметики для их применения при организации параллельного вычислительного процесса [1–3]. Ряд вопросов, касающихся изучения возможности создания систем защиты данных путем специализированных арифметических устройств, а также способов организации кодированного процесса передачи данных по каналам связи описаны в [4, 5].

Традиционная защита информации в ВС осуществляется на основе различных методов шифрования. Процесс шифрования и дешифрирования данных является требовательным к аппаратным средствам и осуществляется в течение некоторого временного интервала с использованием заранее изготовленного ключевого элемента, процесс его хранения может быть потенциально уязвимым. При этом процесс его замены является трудоемким и требует его передачи для всех вычислительных систем, обрабатывающих конфиденциальные данные.

В самой ВС хранение зашифрованной информации ограничивается областями (местами) их хранения и передачи через каналы связи, а выполнение вычислительных операций происходит уже с дешифрованными данными.

Существует возможность, при физическом доступе к вычислительной системе, заморозки ВП и получения образа с возможностью его сохранения, особенно он организовывается по принципам виртуализации. При этом возможно получение доступа к уникальному ключу и расшифровке данных путем анализа данных, находящихся в оперативной памяти, кэш-памяти и процессоре.

Один из выходов решения данной проблемы состоит в использовании специализированных вычислительных систем, которые реализованы на основе специализированных защищенных процессорных устройств (ПУ), осуществляющих шифрование всех этапов выполнения вычислений. Однако специализированные ПУ являются более дорогостоящими при их разработке и при создании вычислительной системы требуют больших финансовых затрат.

Второй подход заключается в распространении процесса кодирования на весь процесс выполнения арифметических операций при использовании универсальных ПУ, а также при применении в системах визуализации. При этом не проводится модификация ПУ или других элементов вычислительного устройства. В таких ПУ используется позиционная машинная арифметика. Для обеспечения кодирования по некоторой ключевой информации всего ВС можно применять систему непозиционного счисления, в которой значение цифры не изменяется в зависимости от его месторасположения. При этом хранение данных также может осуществляться в такой системе счисления.

Одним из вариантов такой арифметики является система остаточных классов или модулярная арифметика, которая получила распространение в системах криптографической защиты, цифровой обработки сигналов, а также в высокоточных вычислениях [3].

Система остаточных классов (СОК) базируется на представлении чисел на основе понятия вычета и китайской теореме об остатках [1]. Согласно тео-

реме для любой системы взаимно простых чисел $p_1, ..., p_n$ любое число X из диапазона [0; M), где $M = p_1 \cdot p_2 \cdot ... \cdot p_n$ может быть представимо в виде вектора $a_1 \cdot a_2, ..., a_n$ (a_1, a_2, …, a_n), где $a_1 = X\%\ p_i$.

Здесь «%» – операция взятия остатка от целочисленного деления X на p_i. $p_1, ..., p_n$ – модули (базисы) системы, a_1, a_2, …, a_n – остатки (вычеты) числа по заданной системе модулей. В качестве основания P_i используются простые числа.

Используемая китайская теорема об остатках позволяет однозначно представить числа из диапазона [0, M – 1]. Число X может быть также найдено следующим образом:

$$X = x_1 e_1 + ... + x_n e_n \left(\mathrm{mod}\, M\right), \tag{1}$$

где

$$e_i = M / P_i \left(\left(M / P_i \right) - 1 \mathrm{mod}\, P_i \right), \left(1 \le i \le n\right).$$

Кроме того, свойства системы остаточных классов к естественному параллелизму позволяют организовать более эффективный вычислительный процесс на ПУ с суперскалярной архитектурой, при котором ПУ имеет несколько конвейеров и вычислительных ядер.

Особенности организации вычислительного процесса

Вычислительный процесс с использованием СОК подразумевает введение дополнительных этапов, на котором происходит преобразование данных из ПСС в НСС, а также обратное их преобразование [6]. Преобразование может быть выполнено как с использованием CPU, так и при наличии дополнительных математических процессоров, например, Intel Xeon Phi или GPU. При этом можно выделить на каждое основание, которое входит в базис числа, свое микроядро CPU или GPU.

После преобразования все дальнейшие арифметические операции в ПУ, а также передача данных между ним и оперативной памятью осуществляются в СОК. Операции над числами выполняются с заранее заданным основанием, тем самым исключается возможность циркулирования исходной, некодируемой информации между вычислителями и оперативной памятью (ОП).

Дополнительное преимущество в использовании СОК в системе заключается в увеличении скорости работы с кодированными данными по сравнению с традиционными алгоритмами шифрования: используемые числа являются малоразрядными, что позволяет использовать ПУ с небольшой разрядностью в бортовых и мобильных информационных системах.

Алгоритм кодирования информации в вычислительной структуре в СОК показан на рис. 1 и осуществляется в следующим порядке:

1. Ввод чисел в позиционной системе счисления (ПСС).
2. Преобразование из ПСС в НСС.
3. Выполнение арифметических операций с основаниями.
4. Преобразование получившегося числа в НСС в ПСС.
5. Вывод результата.

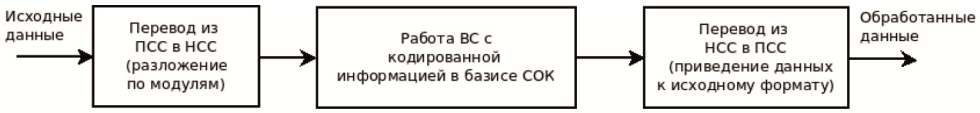

Рис. 1. Алгоритм кодирования информации
в вычислительной структуре на основе СОК

Функционирование защищенной вычислительной структуры

Работа с вычислительной системой начинается заданием основания, необходимого для модулярной арифметики. Первоначально производится выбор модулей, по которому будут производиться вычисления в СОК. Данные модули будут являться ключевой информацией для кодирования данных и задаются пользователем в двух вариантах:

1. При работе только с прикладным программным обеспечением, осуществляющим работу с конфиденциальной информацией. В этом случает данные вводятся в ручном режиме лично пользователем, обладающим необходимыми полномочиями. Такой режим может осуществляться при работе с универсальной ОС и может быть добавлен в существующие программные реализации ОС, такие как Linux, FreeBSD или Windows. Кроме этого, возможно применение в составе мобильной ОС Android.

2. При использовании специализированной операционной системы, которая ориентирована на работу с модулярной арифметикой. В этом случае ключевой набор модулей может быть задан как вручную, так и находиться в составе ВС, например, в виде аппаратного модуля, чтение из которого происходит один раз при старте системы, что дополнительно повышает защищенность ВС.

После осуществления преобразования исходных данных с выбранным основанием в СОК происходит выполнение всех необходимых наборов инструкций ПУ вычислительными и логическими командами, промежуточное хранение результатов вычислительных операций в ОП. Результаты вычислений хранятся как во внутренних регистрах, так и оперативной памяти, а также все передачи потоков данных по соответствующим шинам осуществляются в модулярном формате. После выполнения необходимых результатов данные осуществляют обратное преобразование из НСС в ППС и затем процесс повторяется. Кроме однократного указания выбранного основания, также возможен ввод нового набора через необходимый временной интервал или при начале работы с новой частью конфиденциальной информации.

В случае возникновения попытки снятия образа системы использование СОК не позволит получить исходные данные, представленные на входных портах до преобразования в НСС, поскольку хранение данных внутри памяти ВС осуществляется в виде набора модулей, $(X_1, X_2, X_3, ..., X_n)$, и без знания используемых в системе оснований не позволяет раскодировать данные.

Особенности применения аппаратных и программных ресурсов ВС

Наличие в ВС дополнительных процессорных элементов, таких как ПУ и графические ПУ, которые поддерживают массовый параллелизм и векторные операции, позволяет перенести часть вычислительного процесса, напри-

мер, перевод из ПСС в НСС и обратное преобразование с центрального ПУ, что в целом позволит повысить эффективность вычислительного процесса.

В такой ВС может использоваться как специализированная ОС, ориентированная на ограниченное число используемых приложений, так и универсальная, которая содержит различные системные и пользовательские приложения. Необходимым условием для их использования является выполнение преобразования входных и выходных данных из позиционной системы счисления в непозиционную. Для этого необходимо наличие специализированной библиотеки, которая позволяет использовать форматы данных, представленные в системе остаточных классов в существующих приложениях, например, их адаптация после перекомпиляции, и разработка программных приложений, ориентированных на работу с непозиционной арифметикой.

Перспективным направлением является использование специализированной ОС, реализованной на основе микроядерности. В такой системе создается специализированное ядро, обеспечивающее поддержку модулярной арифметики, а также процесса кодирования данных.

Заключение

В работе показан один из теоретических способов осуществления защиты конфиденциальной информации в ВС с использованием преимущества модулярной арифметики. Показано, что:

1. Использование СОК позволяет осуществить кодирование всего вычислительного процесса, происходящего в ВС.
2. Хранение данных в модулярном формате позволяет обеспечить естественное кодирование информации при выполнении ВП.
3. Получение образа ВП не позволяет получить доступ к исходной конфиденциальной информации.
4. Использование СОК позволяет более эффективно проводить обработку информации за счет естественного параллелизма.
5. Использование простых чисел СОК позволяет использовать малоразрядные ВС.

Библиографический список

1. Акушский, И. Д. Машинная арифметика в остаточных классах / И. Д. Акушский, Д. И. Юдицкий. – М. : Сов. радио, 1968.
2. Червяков, Н. И. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н. И. Червяков, С. А. Ряднов, П. А. Сахнюк, А. В. Шапошников. – М. : Физматлит, 2003. – 288 с.
3. Модулярная арифметика и ее приложения в инфокоммуникационных технологиях / Н. И. Червяков, И. Н. Лавриненко, А. В. Лавриненко, А. А. Коляда, П. А. Ляхов, М. Г. Бабенко. – М. : Физматлит, 2017. – 402 с.
4. Магомедов, Ш. Г. Вариант архитектуры защищенного микропроцессора на основе системы остаточных классов / Ш. Г. Магомедов // Прикаспийский журнал. Управление и высокие технологии. – 2013. – № 4. – С. 118–125.
5. Магомедов, Ш. Г. Использование системы остаточных классов для организации передачи данных морскими судами / Ш. Г. Магомедов // Вестник Астрахан. гос. техн. ун-та. Сер.: Морская техника и технология. – 2010. – № 2. – С. 44–46.
6. Исупов, К. С. Методика выполнения базовых немодульных операций в модулярной арифметике с применением интервальных позиционных характеристик / К. С. Исупов // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2013. – № 3 (27). – С. 26–39.

Васин Леонид Анатольевич
кандидат технических наук,
заведующий кафедрой
информационно-вычислительных систем,
Пензенский государственный
университет архитектуры и строительства
(Россия, г. Пенза, ул. Германа Титова, 28)
E-mail: leo.vasin@gmail.com

Vasin Leonid Anatolievich
candidate of technical sciences,
head of sub-department
of information-computing systems,
Penza State University of Architecture
and Construction
(28 Herman Titov street, Penza, Russia)

УДК 004.41
Васин, Л. А.
Концептуальная организация защищенной вычислительной структуры с внутренним кодированием на основе непозиционной арифметики / Л. А. Васин // Модели, системы, сети в экономике, технике, природе и обществе. – 2018. – № 2 (26). – С. 107–113.