Необходимо подчеркнуть, что фаза распространения атрибутов выполняется один раз. Метод позволяет на схеме, представленной в виде базы данных после фазы распространения атрибутов, находить всех кандидатов для любого числа запросов без повторного распространения атрибутов. Такой подход гарантирует существенное уменьшение суммарного времени на поиск. С помощью данного метода решаются следующие задачи: определение глобальных сигналов синхронизации, асинхронных управляющих сигналов (сброс/установка), определение и анализ доменов синхронизации, проверка правил проектирования, проверка правил тестопригодного проектирования, поиск циклов с обратными связями.

ВЫВОДЫ

Задача поиска по заданным параметрам в некоторой внутренней модели решается на разных этапах проектирования аппаратуры. Особенно заметна роль таких задач на этапе логического синтеза. С помощью поиска по шаблону можно находить некорректно спроектированные цепи сброса или синхронизации, «случайные» триггеры-защелки, использование комбинационной логики для управления синхронизацией триггеров и т. д. При древовидной либо графовидной структуре поиск выполняется последовательно, начиная от корневой вершины [4]. Применение табличного представления позволяет свести поиск к реализации запросов по системе таблиц.

СПИСОК ЛИТЕРАТУРЫ

1. *Yadav M.* Hardware Architecture of a Parallel Pattern Matching Engine / Yadav M., Venkatachaliah A., Franzon P. // Proceedings of the ISCAS 2007. – Pp. 1369–1372.
2. *Tarau P.* Exact Combinational Logic Synthesis and Non-Standard Circuit Design / Tarau P., Luderman B. // Proceedings of the CF'08, May 5–7, 2008, Ischia, Italy. – Pp.15–24.
3. *Tarau P.* Revisiting Exact Combinational Circuit Synthesis / Tarau P., Luderman B. // Proceedings of the SAC'08, Fortaleza, Cear´a, Brazil. – Pp. 1620–1621.
4. *Clifford R.* From coding theory to efficient pattern matching / Clifford R., Efremenko K., E. Porat, Rothschild A. // Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, 2009, New York, USA. – Pp. 778–784.

Надійшла 15.03.2010

Кривуля Г. Ф., Сиревич Є. Ю., Карасьов А. Л.

АНАЛІЗ СПИСКУ З'ЄДНАНЬ У СИСТЕМАХ ЛОГІЧНОГО СИНТЕЗУ

Розглянуто проблему аналізу результатів логічного синтезу для їх подальшого аналізу на відповідність правилам синтезу. Запропоновано табличну модель, що складається із словника атрибутів, отриманого після фази поширення. Визначено етапи пошуку за заданими параметрами при використанні розробленої моделі.

Ключові слова: логічний синтез, HDL, атрибут лінії, макрос, пошук за параметрами.

Krivulya G. F., Syrevich Yev. Yu., Karasyov A. L.

NETLIST ANALYSIS IN LOGICAL SYNTHESIS SYSTEMS

The problem of logical synthesis results analysis for their further analysis for accordance to the rules of synthesis is considered. A tabular model is proposed, consisting of an attributes dictionary obtained after the distribution phase. Stages of search by the preset parameters are determined when using the developed model.

Key words: logical synthesis, HDL, line attribute, macro, pattern matching.

УДК 003.26+004.272.4+004.415.2+004.051

Неласая А. В.[1], Верещак М. И.[2]

[1]Старший преподаватель Запорожского национального технического университета

[2]Студент Запорожского национального технического университета

ОЦЕНКА ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕК ДЛИННОЙ АРИФМЕТИКИ В КРИПТОГРАФИЧЕСКИХ ПРИЛОЖЕНИЯХ

В статье проводится анализ эффективности целочисленных операций в современных библиотеках длинной арифметики. Выбрана программно-аппаратная платформа, допускающая выполнение параллельных алгоритмов, для построения эффективных процедур определения параметров криптографических систем.

Ключевые слова: криптографическая система, асимметричная криптография, эллиптическая кривая, порядок группы, библиотека длинной арифметики, анализ производительности, параллельные вычисления, графический процессор.

ВВЕДЕНИЕ

Потребность широкого применения электронных способов передачи и обработки информации выводит на качественно новый уровень проблемы защиты электронной информации. Большинство угроз целостности и конфиденциальности информации, цирку-

лирующей в компьютерных системах, может быть перекрыто на основе использования механизмов безопасности, реализуемых с помощью криптографических методов защиты.

Традиционно криптографические преобразования делятся на две большие ветви – симметричные и асимметричные. Симметричные криптопреобразования являются более быстрыми и используются, собственно, для шифрования файлов. Появление асимметричной криптографии позволило решить задачу аутентификации электронной информации в условиях, когда обменивающиеся информацией стороны не доверяют друг другу. Эту задачу невозможно было бы решить с использованием только лишь симметричных алгоритмов.

Асимметричные криптографические преобразования используются в системах направленного шифрования и цифровой подписи для обеспечения целостности, неопровержимости и аутентичности электронных ресурсов. В частности, задача аутентификации документов решается с помощью использования механизма электронной цифровой подписи, который реализуется методами криптографии с открытым ключом.

Современные стандарты цифровой подписи (в том числе Украинский ДСТУ 4145-2002) и направленного шифрования основаны на использовании операций в группах точек эллиптических кривых. Для обеспечения достаточной стойкости длина секретного ключа должна превышать 160 бит. Размер элементов основного поля, над которым определяется кривая, имеет такой же порядок. Отсюда ясно, что для программной реализации алгоритмов криптографических преобразований на эллиптических кривых не обойтись без длинной арифметики. Естественным обобщением эллиптических кривых являются гиперэллиптические кривые – кривые более высокого рода [1, 2], которые позволяют решать аналогичные задачи.

Стойкость алгоритмов асимметричной криптографии обеспечивается использованием алгебраических структур больших порядков. Наиболее важным параметром, с точки зрения обеспечения стойкости, является порядок группы точек эллиптической кривой и, соответственно, порядок якобиана гиперэллиптической кривой. Построенная криптосистема будет достаточно надежной при условии соответствия этого параметра ряду ограничений [3]. Однако, задача определения порядка в общем случае не является тривиальной и требует значительных затрат временных и технических ресурсов.

Цель данной работы состоит в проведении сравнительного анализа эффективности использования библиотек длинной арифметики для целочисленных вычислений и выборе программно-аппаратного решения для повышения производительности процедуры определения порядка кривой.

1. КРАТКАЯ ХАРАКТЕРИСТИКА СОВРЕМЕННЫХ БИБЛИОТЕК ДЛИННОЙ АРИФМЕТИКИ

В вычислительной технике под длинной арифметикой понимается выполнение операций над числами, разрядность которых превышает длину машинного слова данной вычислительной машины. Частный случай – арифметика произвольной точности – относится к арифметике, в которой длина чисел ограничена только объемом доступной памяти. Именно арифметика произвольной точности находит широкое применение в расчетах, связанных с криптографией. В языках программирования, как правило, нет встроенных средств для выполнения расчетов с произвольной точностью. Для этого необходимо использовать сторонние библиотеки либо создавать собственные решения. На сегодняшний день существует множество различных готовых библиотек в разных языках программирования для выполнения таких расчетов. При этом возникает проблема выбора инструментария, наиболее подходящего для решения конкретных прикладных задач в каждом случае. В данной статье проводится анализ производительности с точки зрения удобства использования в алгоритмах асимметричной криптографии.

Прежде чем остановить свой выбор на какой-либо из библиотек, либо прийти к выводу о необходимости создания собственного программного продукта, необходимо проанализировать и сравнить существующие решения.

Все множество предлагаемых библиотек можно классифицировать по нескольким параметрам. Первый – лицензия, под которой распространяется программный продукт. Ее необходимо учитывать при публикации своих исследований, проведенных с использованием той или иной библиотеки. Различают коммерческие (IMSL), бесплатные для некоммерческого использования (LiDIA, MIRACL) и бесплатные (GMP, NTL, CLN, MPI, Imath) программные продукты. Следует отметить, что распространяемых под различными разновидностями свободных лицензий программных продуктов сейчас значительно больше, нежели коммерческих. Причиной тому, на наш взгляд, является то, что большая часть таких библиотек рождалась в стенах университетов в приложении к различным научным проектам. Впоследствии эти наработки публиковались, и за их доработку

и совершенствование бралось множество программистов по всему миру.

Вторым важным критерием, который нужно учесть при выборе библиотеки, является перечень поддерживаемых типов и структур данных, от которых напрямую зависит эффективность производимых расчетов. Это:

– целые числа произвольной длины (знаковые и беззнаковые);

– рациональные дроби;

– числа с плавающей точкой произвольной точности;

– комплексные числа;

– векторы;

– матрицы;

– полиномы.

В задачах, с которыми сталкивается современная криптография, наиболее востребованными являются расчеты целочисленных значений произвольной длины (присутствуют во всех библиотеках) и полиномиальная арифметика (реализация которой встречается значительно реже). Поэтому результаты сравнения библиотек, приведенные в данной статье, могут оказаться не вполне подходящими для иных предметных областей. В табл. 1 приведены основные характеристики современных библиотек длинной арифметики [4].

GMP – это свободная библиотека длинной арифметики, поддерживающая работу со знаковыми целыми, рациональными числами и числами с плавающей запятой [5]. Преимущества GMP:

– практически не имеет ограничений точности вычислений, кроме ограниченного объема доступной памяти;

– имеет богатый набор функций и удобный интерфейс;

– поддерживает большинство современных платформ: Unix-подобные операционные системы, такие как GNU/Linux, Solaris, HP-UX, Mac OS X/Darwin, BSD, AIX; Windows. Существуют 32-разрядная и 64-разрядная версии GMP;

– допускает использование наиболее эффективных алгоритмов и оптимизированного под различные современные процессорные системы ассемблерного кода во всех внутренних циклах.

GMP является одной из самых быстрых на сегодняшний день библиотек длинной арифметики. Основные области применения библиотеки – это криптографические системы и исследования, обеспечение безопасности межсетевых взаимодействий, алгебраические пакеты. Библиотека GMP является частью проекта GNU.

CLN – это библиотека для расчетов с использованием всех существующих числовых типов [6]. В этой библиотеке реализованы классы для таких типов данных, как: целые числа, рациональные дроби, числа с плавающей точкой, комплексные числа, универиантные полиномы, вычисления по модулю. CLN использует GMP в качестве вычислительного ядра и незначительно уступает ей в скорости расчетов. Богатый набор поддерживаемых типов, удобный программный интерфейс, прозрачные механизмы взаимодействия и преобразования друг в друга различных структур данных вместе с достаточно высокой скоростью вычислений обеспечивают широкую распространенность данной библиотеки. CLN лежит в основе многих программных продуктов, связанных с научными исследованиями и математическими расчетами: Scilab, Octave (свободные аналоги MatLAB), maxima (пакет для алгебраических вычислений, свободный аналог MAPLE) и др.

NTL – это высокопроизводительная библиотека C++, предоставляющая структуры данных и алгоритмы для работы с целыми произвольной длины, векторами, матрицами, полиномами, числами с плавающей точкой произвольной точности [7]. Полиномиальная арифметика в NTL – одна из самых быстрых на сегодняшний день. Она не раз использовалась при становлении «мировых рекордов» в скорости алгоритмов факторизации и определении порядка эллиптических кривых. Все алгоритмы NTL реализованы на C++, что обеспечивает высокую мобильность данной библиотеки. Несмотря на наличие собственного вычислительного ядра, NTL позволяет использовать вместо него библиотеку GMP.

MIRACL – единственная из перечисленных библиотек, распространяемая под коммерческой лицензией [8]. Однако авторами разрешается использование MIRACL для некоммерческих нужд. Основные преимущества данной библиотеки:

– наличие реализованных C/C++ интерфейсов «из коробки»;

– богатая библиотека специализированных функций для вычислений в области эллиптической криптографии;

– наличие различных алгоритмов решения одних и тех же задач с возможностью выбора оптимального варианта для текущих нужд.

2. АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ

Анализ производительности исследуемых библиотек проводился с применением простых, аналогичных по структуре программ, выполняющих различные арифметические операции. Всего было проведено

Таблица 1. Основные характеристики библиотек длинной арифметики

Пакет / название библиотеки	Поддерживаемые типы данных	Языки программирования	Лицензия
apfloat	Decimal floats, integers, rationals, and complex	Java and C++	LGPL and Freeware
ARPREC and MPFUN	Integers, binary floats, complex binary floats	C++ with C++ and Fortran bindings	BSD
Base One Number Class	Decimal floats	C++	Proprietary
bbnum library	Integers and floats	Assembler and C++	New BSD
phpseclib	Decimal floats	PHP	LGPL
BigDigits	Naturals	C	Freeware [1]
BigFloat	Binary Floats	C++	GPL
BigNum	Binary Integers, Floats (with math functions)	C# /. NET	Freeware
C++ Big Integer Library	Integers	C++	Public domain
CLN, a Class Library for Numbers	Integers, rationals, and complex	C and C++	GPL
Computable Real Numbers	Reals	Common Lisp	
IMSL		C	Commercial
decNumber	Decimals	C	ICU licence (MIT licence) [2]
FMLIB	Floats	Fortran	
GNU Multi-Precision Library (and MPFR)	Integers, rationals and floats	C and C++ with bindings (GMPY,…)	LGPL
GNU Multi-Precision Library for. NET	Integers	C# /. NET	LGPL
Eiffel Arbitrary Precision Mathematics Library	Integers	Eiffel	LGPL
HugeCalc	Integers	C++ and Assembler	Proprietary
IMath	Integers and rationals	C	Freeware
IntX	Integers	C# /. NET	New BSD
JScience LargeInteger	Integers	Java	
libmpdec	Decimals	C and C++	Simplified BSD
LibTomMath	Integers	C and C++	Public domain
LiDIA	Integers, floats, complex floats and rationals	C and C++	Free for non-commercial use
MAPM	Integers and decimal floats	C (bindings for C++ and Lua)	Freeware
MIRACL	Integers and rationals	C and C++	Free for non-commercial use
MPI	Integers	C	LGPL
MPArith	Integers, floats, and rationals	Pascal / Delphi	zlib
mpmath	Floats, complex floats	Python	New BSD
NTL	Integers, floats	C and C++	GPL
bigInteger (and bigRational)	Integers and rationals	C and Seed7	LGPL
TTMath library	Integers and binary floats	Assembler and C++	New BSD
vecLib. framework	Integers	C	Proprietary
W3b. Sine	Decimal floats	C# /. NET	New BSD

4 теста на компьютере следующей конфигурации: CPU Intel Celeron M 1,73 GHz, RAM DDR2 2 Gb.

Тест 1. Работа с целыми числами.

Тесты 2–3. Полиномиальная арифметика.

Тест 4. Вычисление точек эллиптической кривой.

Целочисленные вычисления. Этот тест (табл. 2) представляет собой вычисление факториала числа без применения каких-либо оптимизирующих алгоритмов или функций. При этом в GMP использован C-интерфейс, а во всех остальных библиотеках – C++-интерфейс.

Таблица 2. Время вычисления факториала N!

N	GMP, с	NTL, с	CLN, с	MIRACL, с
1000	<0,01	<0,01	<0,01	0,01
5000	0,01	0,01	0,02	0,16
10000	0,05	0,05	0,08	0,65
20000	0,24	0,23	0,34	2,71
50000	1,67	1,66	2,46	! Overflow!
100000	7,22	7,19	12,06	! Overflow!

Полиномиальные расчеты. Второй тест (табл. 3) представляет собой последовательное умножение полинома первой степени на самого себя. Результирующее значение на каждой итерации – полином, степень которого линейно зависит от номера текущей итерации. Этот эксперимент по своей сути соответствует основной операции метода [9] определения порядка кривой.

Таблица 3. Время последовательного умножения полинома первой степени на самого себя

Количество итераций	NTL, с		CLN, с	
	Длина коэффициентов полинома			
	16 бит	32 бит	16 бит	32 бит
50	0,03	0,04	0,04	0,05
100	0,15	0,15	0,17	0,18
200	0,6	0,59	0,67	0,73
500	3,78	3,78	4,33	4,66
1000	15,84	15,16	17,24	18,29

Третий тест (табл. 4) представляет собой последовательное возведение полинома в квадрат. Длина коэффициентов – 32 бит. При этом вычислительная сложность и объем используемой программой памяти растет очень быстро. Такая операция чаще всего встречается в полиномиальной арифметике, применяемой в криптографии.

Таблица 4. Время возведения полинома в квадрат

Количество итераций	Степень полинома	NTL, с	CLN, с
4	8	<0,01	<0,01
6	32	<0,01	0,01
8	128	0,01	0,2
12	2048	0,17	50,16
16	32768	3,86	inf
18	131072	17,08	inf
20	524288	! overflow!	inf

Скалярное умножение на эллиптической кривой. Последний тест (табл. 5) был проведен для оценки производительности встроенных функций библиотеки MIRACL для операций на эллиптических кривых. По результатам предыдущих тестов ясно, что CLN сильно отстает в скорости вычислений по сравнению с NTL, поэтому в таблице приведены данные сравнительного анализа производительности только NTL и MIRACL. В данном тесте определялось время умножения точки кривой, определенной над простым полем с размером модуля 192 бита, на длинное число размером 256 бит. Эта операция является основой для процедур вычисления открытого ключа, формирования и проверки цифровых подписей. Для чистоты эксперимента за один запуск производится 1000 операций умножения. Тест был повторен 3 раза.

Таблица 5. Скалярное умножение на эллиптической кривой

Номер запуска	NTL		MIRACL	
	Общее время, мс	Одна операция, мс	Общее время, мс	Одна операция, мс
1	6820	6,82	2845	2,84
2	5678	5,68	3048	3,05
3	7532	7,53	3209	3,21

Этот эксперимент демонстрирует, что, не смотря на то, что MIRACL проигрывает в производительности целочисленных вычислений библиотеке NTL, производительность специализированных функций превосходит аналогичные по реализации в NTL более чем в 2 раза.

По результатам анализа можно сделать некоторые выводы относительно исследованных библиотек и дать им краткие характеристики:

– Как было сказано выше, библиотека GMP является частью проекта GNU, что в значительной мере определят идеологию ее разработки: каждый отдельно взятый продукт данного проекта выполняет определенную возложенную на него задачу и только ее. В случае с GMP этот принцип выливается в ограниченный набор поддерживаемых типов данных – в GMP реализованы лишь базовые операции над скалярными величинами. Упор в этой библиотеке делается на оптимизацию вычислительного ядра под каждую из популярных моделей процессоров и своевременные обновления алгоритмов с появлением новых платформ, и

с этой задачей разработчики справляются в полной мере. Таким образом, у GMP нет определенной направленности, что делает возможным и желательным использование ее в качестве базы для разработки более узконаправленных продуктов.

– Библиотека CLN, пожалуй, имеет наиболее широкий спектр применений в области математических исследований. Однако универсальность этой библиотеки имеет и свою обратную сторону – как показывают тесты, ее скорость в сравнении с NTL и GMP падает в 1,5–2 раза. Эта библиотека прекрасно подходит для программных продуктов, где существует необходимость в поддержке значительного количества различных типов данных и скорость вычислений не является критичной, например математических пакетов общего назначения (наподобие MAPLE, MathCAD и пр.).

– Библиотека NTL имеет более узкую направленность. Здесь основной упор сделан на целочисленную арифметику и полиномиальные расчеты. Применяемые в ней алгоритмы полиномиальных вычислений на сегодняшний день обладают рекордными показателями скорости. Следует также заметить, что в полной мере потенциал NTL раскрывается в сочетании с GMP. Таким образом, NTL доказывает целесообразность использования GMP в качестве основы для создания более узконаправленных библиотек.

– Библиотека MIRACL может наиболее эффективно использоваться для решения специализированных задач области эллиптической криптографии.

3. ИСПОЛЬЗОВАНИЕ СОВРЕМЕННЫХ АППАРАТНЫХ РЕШЕНИЙ

Традиционно для решения объемных вычислительных задач, к которым относится и задача определения порядка кривой, используются параллельные методы вычислений, требующие наличия специализированной высокопроизводительной техники. Параллельные вычисления в крупных масштабах (десятки и сотни вычислительных узлов) требуют очень серьезных капиталовложений на приобретение и обеспечение аппаратной базы. Первый вариант – это кластеры, состоящие из нескольких десятков вычислительных узлов, управляемых одной операционной системой. При этом каждый вычислительный узел способен функционировать автономно, как самостоятельная вычислительная система. Подобные системы очень громоздки, имеют большой расход электроэнергии и требуют применения сложных и высоконадежных систем охлаждения наряду с системой обеспечения безопасности. Другой вариант – это технология облачных вычислений (cloud computing) и GRID-системы. Их преимуществом перед кластерами является то, что подобные системы достаточно легко могут быть развернуты практически на любой имеющейся в наличии аппаратной базе. При этом они легко масштабируемы и практически не имеют ограничений по наращиванию мощностей. Недостатком облачных вычислений является то, что в качестве вычислительных узлов выступают отдельные рабочие станции, объединенные локальной сетью, вследствие чего они опять-таки требовательны к ресурсам. К тому же передача данных по локальной сети происходит сравнительно медленно, поэтому для получения высокой эффективности в такой системе каждую задачу необходимо разбивать на крупные блоки, каждый их которых будет выполняться на своем узле, что значительно затрудняет процесс разработки и подходит не для каждой задачи.

Альтернативное решение пришло с довольно неожиданной стороны. По мере развития игровой компьютерной индустрии в течение многих лет велись разработки по наращиванию мощности графических адаптеров, так как на них возлагалась задача быстрого пересчета в режиме реального времени массивов данных, связанных с моделированием трехмерного виртуального пространства. Для решения этой задачи в графические адаптеры было внедрено множество отдельных процессоров с усеченным по сравнению с CPU набором команд, которые способны, работая параллельно, выполнять различные арифметические операции. Когда количество и мощность этих процессоров достигли определенного уровня, оказалось, что они могут быть эффективно использованы не только для расчетов координат трехмерного пространства, но и для любых других математических расчетов. Так родилась идея использования возможностей графических адаптеров для организации параллельных вычислений, превратившаяся впоследствии в отдельный проект.

На сегодняшний день компанией NVIDIA разработан и запущен проект под названием CUDA [10, 11]. Это программно-аппаратная архитектура, позволяющая производить вычисления с использованием графических процессоров NVIDIA, поддерживающих технологию GPGPU. CUDA SDK позволяет программистам реализовывать на специальном упрощенном диалекте языка программирования Си алгоритмы, выполнимые на графических процессорах NVIDIA, и включать специальные функции в текст программы на Си. CUDA дает разработчику возможность по своему усмотрению организовывать доступ к набору инструкций графического ускорителя и управлять его памятью, организовывать на нем сложные параллель-

ные вычисления. Все современные графические адаптеры NVIDIA поддерживают технологию CUDA. Кроме этого, NVIDIA выпустила семейство вычислительных систем Tesla на основе графических процессоров с архитектурой CUDA, которые могут быть использованы для научных и технических вычислений общего назначения. Технические решения на базе данной технологии значительно дешевле, чем аналогичные им по мощности кластерные системы, и обладают значительно меньшими показателями энергопотребления и тепловыделения в процессе работы, за счет чего снижаются затраты на их обеспечение.

Технология CUDA является достаточно перспективной для решения поставленных в данной работе задач. На данный момент на CUDA разработана библиотека вычислений произвольной точности с действительными числами для применения в задачах моделирования сложных процессов, описываемых дифференциальными уравнениями [12]. Однако в криптографических приложениях основная масса вычислений выполняется над длинными целыми числами. Программного инструментария, реализующего подобную функциональность для GPU, пока не существует. Исходя из этого, для обеспечения возможности эффективной реализации алгоритмов определения порядков кривых на графических ускорителях первоочередной задачей является разработка библиотеки длинной целочисленной (в том числе модульной) арифметики на CUDA. В этом направлении авторами планируется проводить дальнейшие исследования.

ЗАКЛЮЧЕНИЕ

Проведенный анализ современных программных и аппаратных средств позволил определить особенности использования и эффективность для конкретных операций распространенных библиотек длинной арифметики GMP, CLN, NTL и MIRACL. Эту информацию могут использовать разработчики программного обеспечения при выборе библиотеки для своих нужд.

Анализ аппаратных решений позволил выбрать достаточно мощную и недорогую технологию параллельных вычислений, которую планируется использовать для повышения производительности процедур определения параметров криптосистем на эллиптических и гиперэллиптических кривых.

СПИСОК ЛИТЕРАТУРЫ

1. *Koblitz N.* Hyperelliptic cryptosystem / N. Koblitz // Journal of Crypto. – 1989. – № 1. – P. 139–150.
2. *Menezes A.* An Elementary Introduction to Hyperelliptic Curves [Электронный ресурс] : published as Technical Report CORR 96-19 Department of C&O University of Waterloo/ Menezes A., Wu Y., Zuccherato R. – Электрон. дан. – Ontario, Canada, 1996. – P. 1–35. – Режим доступа: www.cacr.math.uwaterloo.ca/techreports/1997/corr96-19.ps, свободный. – Загл. с экрана.
3. *Неласа Г. В.* Удосконалення методів перетворень в якобіанах гіпереліптичних кривих для криптографічних додатків : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.21 «Системи захисту інформації» / Г. В. Неласа. – Харків, 2010. – 22 с.
4. Arbitrary-precision arithmetic [Электронный ресурс]. – Электрон. дан. – Режим доступа: http://en. wikipedia. org/wiki/Arbitrary-precision_arithmetic, свободный. – Название с титул. экрана.
5. GMP «Arithmetic without limitation» The GNU Multiple Precision Arithmetic Library [Электронный ресурс]. – Электрон. дан. – Режим доступа: http://gmplib.org/, свободный. – Название с титул. экрана.
6. CLN-Class Library for Numbers [Электронный ресурс]. – Электрон. дан. – Режим доступа: http://www.ginac.de/CLN/, свободный. – Название с титул. экрана.
7. NTL: A Library for doing Number Theory [Электронный ресурс]. – Электрон. дан. – Режим доступа: http://www.shoup.net/ntl/, свободный. – Название с титул. экрана.
8. Multiprecision Integer and Rational Arithmetic C/C++ Library [Электронный ресурс]. – Электрон. дан. – Режим доступа: http://www.shamus.ie/, свободный. – Название с титул. экрана.
9. *Bostan A.* Linear recurrences with polynomial coefficients and application to integer factorization and Cartier-Manin operator / Bostan A., Gaudry P., Schost É. // Proceedings of Fq7, Lecture Notes in Comput. Sci. Vol. 2948. – Berlin : Springer-Verlag. 2004. – P. 40–58.
10. CUDA Toolkit 3.0 (March 2010) : Download Quick Links [Электронный ресурс] / NVIDIA. – Электрон. дан. – Режим доступа: http://developer.nvidia.com/object/cuda_3_0_downloads.html, свободный. – Загл. с экрана.
11. *Борсеков А. В.* Основы работы с технологией CUDA / А. В. Борсеков, А. А. Харламов. – М. : ДМК Пресс, 2010. – 232 с.
12. Научно-образовательный центр «Параллельные вычисления» [Электронный ресурс]. – Электрон. дан. – Режим доступа: www.parallel-compute.com, свободный. – Название с титул. экрана.

Надійшла 15.02.2010

Неласа Г. В., Верещак М. І.

ОЦІНКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ БІБЛІОТЕК ДОВГОЇ АРИФМЕТИКИ В КРИПТОГРАФІЧНИХ ДОДАТКАХ

В статті проводиться аналіз ефективності операцій з цілими числами в сучасних бібліотеках довгої арифметики. Вибрано програмно-апаратну платформу, що допускає виконання паралельних алгоритмів, для побудови ефективних процедур визначення параметрів криптографічних систем.

Ключові слова: криптографічна система, асиметрична криптографія, еліптична крива, порядок групи, бібліотека довгої арифметики, аналіз ефективності, паралельні обчислення, графічний процесор.

Nelasa G. V., Vereschak M. I.

EFFICIENCY EVALUATION OF LONG NUMBER LIBRARIES IN CRYPTOGRAPHY APPLICATIONS

Research of integer arithmetic efficiency in modern long number libraries is carried out. The soft hardware with parallel commands for construction of efficient procedures of cryptosystem parameters determination is chosen.

Key words: cryptosystem, open key cryptography, elliptic curve, order of group, long number library, efficiency evaluation, parallel computing, graphic processor unit.