# ИНФОРМАТИКА

УДК 681.3.06

## ПРИМЕНЕНИЕ ТРАНСЦЕНДЕНТНЫХ ПРЕОБРАЗОВАНИЙ, ЧУВСТВИТЕЛЬНЫХ К ПОГРЕШНОСТИ ВЫЧИСЛЕНИЯ В СИСТЕМАХ ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ

***канд. техн. наук, доц. Д.О. ГЛУХОВ***
***(Полоцкий государственный университет)***

*Активная государственная политика Республики Беларусь и стран СНГ в направлении формирования рынка интеллектуальной собственности создает основу для развития современных средств защиты программного обеспечения. Решается задача защиты научного программного обеспечения электронным ключом от несанкционированного использования применением математических методов, сверхчувствительных к погрешности вычислительных машин. Такой подход позволяет обеспечить высокую сложность обратных преобразований, большое количество и разнообразие фрагментов кода, выполняющего верификацию электронного ключа в защищаемом программном обеспечении.*

В настоящее время активно совершенствуется государственная политика Республики Беларусь и стран СНГ по защите авторских прав на программы для ЭВМ и базы данных, принимаются жесткие меры по борьбе с пиратством, преступлениями в информационных сетях. Такая политика позволяет всерьез говорить о том, что формируется рынок интеллектуальной собственности, в частности программного обеспечения (ПО) и компонентов программного обеспечения.

Эти процессы требуют соответствующих шагов и от разработчиков программного обеспечения. Программа становится товаром, со всеми присущими для него характеристиками. Отрабатываются различные схемы продаж программного обеспечения. Разрабатываются новые способы защиты программного обеспечения от несанкционированного копирования и использования.

Несмотря на все усилия различных организаций во главе с BSA в последние годы продолжается рост компьютерного пиратства. В среднем доля пиратского ПО в глобальном масштабе составляет 40 %, т.е. каждые четыре из десяти копий программы оказываются в каком-то смысле украденными у производителя и лишают его прибыли. По расчетам BSA, в 2002 году убытки софтверной отрасли от пиратства составили порядка 13 миллиардов долларов [2].

Рассмотрим классификацию основных современных криптографических методов [5 – 7]:

1. Симметричные (с секретным, единым ключом, одноключевые, single-key).
   1.1. Потоковые (шифрование потока данных):
   - с одноразовым или бесконечным ключом (infinite-key cipher);
   - с конечным ключом (система Вернама – Vernam);
   - на основе генератора псевдослучайных чисел (ПСЧ).

   1.2. Блочные (шифрование данных поблочно):
   1.2.1. Шифры перестановки (permutation, P-блоки).
   1.2.2. Шифры замены (подстановки, substitution, S-блоки):
   - моноалфавитные (код Цезаря);
   - полиалфавитные (шифр Видженера, цилиндр Джефферсона, диск Уэтстоуна, Enigma).

   1.2.3. Составные:
   - Lucipher (фирма IBM, США);
   - DES (Data Encryption Standard, США);
   - FEAL-1 (Fast Enciphering Algoritm, Япония);
   - IDEA/IPES (International Data Encryption Algorithm/Improved Proposed Encryption Standard, фирма Ascom-Tech AG, Швейцария);
   - B-Crypt (фирма British Telecom, Великобритания);
   - ГОСТ 28147-89 (СССР); Skipjack (США).
2. Асимметричные (с открытым ключом, public-key):
   - Диффи-Хеллман DH (Diffie, Hellman);
   - Райвест-Шамир-Адлеман RSA (Rivest, Shamir, Adleman);
   - Эль-Гамаль ElGamal.

Исходя из этой классификации можно выделить **основные тенденции** разработки средств защиты, которые по аналогии начинают применяться и для решения задачи построения средств защиты программного обеспечения:

1. *Шифрование кода или данных симметричным или ассиметричным (с открытым ключом) методом.* Шифрование выполняется с применением аппаратных или электронных (серийные номера) ключей:

- для ассиметричных систем наблюдается усложнение варианта процедуры получения открытого ключа с применением процедуры активации на сервере активации, применение многокомпонентных ключей, применение передачи открытого ключа на электронном носителе (аппаратные ключи);
- разновидностью ассиметричных систем защиты являются системы, обеспечивающие привязку к аппаратным особенностям носителя, защищаемого программного обеспечения;
- для симметричных систем, построенных в основном на алгоритмах над большими целыми простыми числами, наблюдается тенденция увеличения разрядности ключа.

2. *Запутывание и усложнение кода или обфускация.* Выполняется на стадии компиляции и значительно усложняет анализ кода при деассемблировании и отладке в процессе взлома [8].

3. *Мутация кода или данных при каждом запуске программы*, не нарушающая работоспособности и корректности функционирования программного обеспечения. Разновидностью являются одноразовые системы ключей, мутации шифров [1].

5. *Методы затруднения деассемблирования* – используются различные приемы, направленные на предотвращение деассемблирования в пакетном режиме.

6. *Методы затруднения отладки* – используются различные приемы, направленные на усложнение отладки программы.

Обычно методы защиты комбинируются с целью получения более эффективного варианта защиты.

Таким образом, современные алгоритмы шифрования, гаммирования, обфускации и мутации все чаще применяются в коммерческих программных продуктах. Если еще 5...10 лет назад в условиях ограниченности ресурсов вычислительных машин в подходе к построению компиляторов доминировал принцип необходимости оптимизации кода, достижения максимального быстродействия, компактности кода, то сегодня в условиях некоторой избыточности быстроразвивающихся вычислительных мощностей, даже к компиляторам начинают применяться требования поддержки систем защиты кода. Возможно, в ближайшем будущем мы увидим появление промышленных стандартов в этой области.

Однако необходимо учитывать, что защищаемое программное обеспечение неодинаково. Например, нельзя подходить к защите графического пакета точно так же, как к защите научного ПО. Учет специфики программного обеспечения может помочь построить защиту более эффективным способом.

Категория программного обеспечения «Научное программное обеспечение», рассматриваемого в данной статье, характеризуется рядом важных особенностей:

1) научные алгоритмы построены на основе проведенного численного анализа [11] и поэтому не допускают применения алгоритмов запутывания, нарушающих последовательность потока вычислительных операций устройства вычислений с плавающей точкой (изменение флагов регистра, управляющих округлением, точностью; влияние предыдущих состояний регистров стека сопроцессора на результат операций);

2) применение сложных вычислений в алгоритмах защиты над данными, используемыми в основном защищаемом алгоритме, также может вносить дополнительные погрешности вычислений, что недопустимо;

3) применение методов мутации и других вариантов защиты времени выполнения существенно снижает быстродействие алгоритмов расчета и, соответственно, конкурентоспособность ПО. Поскольку научное ПО обычно требует значительного ресурса процессорного времени (часы, иногда сутки) при решении серьезных инженерных и научных задач, то потеря производительности является критичным параметром;

4) наиболее важными областями кода являются именно математические алгоритмы, представляющие собой объекты интеллектуальной собственности, раскрытие которых или запуск в работоспособном состоянии и является целью пиратов, поэтому защита должна фокусироваться в первую очередь на этих фрагментах кода и данных.

Также, когда речь идет о защите ПО как особого способа представления данных и знаний о механизмах манипулирования ими (алгоритмах), то необходимо отметить общие особенности ПО как объекта защиты:

1) необходимо учитывать, что защита выполняется в интерпретации того либо иного микропроцессорного устройства, а значит, оперирует системой команд и учитывает особенности аппаратной платформы (например, система защиты методом обфускации для RISC процессоров не подходит для защиты ПО, построенного на системе команд CISC архитектуры);

2) необходимо учитывать отличия устройств вычислений с плавающей точкой различных производителей (Intel, AMD, MAC, SPARC) и возможные ошибки в различных моделях и архитектурах микропроцессоров.

## Формализация

Для формального определения рассмотрим функции, заданные на дискретном множестве представимых в машинном формате вещественных чисел $\boldsymbol{D}$. Будем говорить, что функция $f$ двух переменных замкнута относительно $\boldsymbol{D}$, если $f(u, k) \in \boldsymbol{D}$ при $u, k \in \boldsymbol{D}$.

Функцию $f$ будем называть обратимой, если существует обратная функция $g$, такая, что для всех $u \in \boldsymbol{D}$

$$G(f(u, k), k) = u. \quad (1)$$

**Определение 1. Квазиобратимая функция представимых вещественных чисел**

*Функция $f$ квазиобратима, если существует такая обратная функция $g$, что для всех $u \in \boldsymbol{D}$ выполняется неравенство:*

$$u - \Delta u < g(f(u, k), k) < u + \Delta u, \quad (2)$$

где $\Delta u$ – абсолютная погрешность обращения: $\Delta u = u\delta$ ($\delta$ – относительная погрешность обращения).

Теоретические основы влияния шумоподобных сигналов на процессы шифрования и дешифрования активно изучаются в теории кодирования телевизионных сигналов. Источником погрешности при кодировании таких сигналов являются погрешности АЦП- и ЦАП-преобразователей, электромагнитные шумы с известной плотностью распределения. В нашем случае источником погрешности выступает вычислительное устройство, причем характеристики «зашумленности» шифруемых данных при определенном выборе квазиобратимых функций будут также приближаться к случайным последовательностям. Псевдослучайная погрешность вычисления, однако, обладает одной отличительной чертой – она повторима в точности при одинаковых начальных условиях. Ею можно управлять, *ее можно использовать*.

**Определение 2. Необратимая функция представимых вещественных чисел**

Функция $f$ необратима, если не существует такая обратная функция $g$, что для всех $u \in \boldsymbol{D}$ выполняется неравенство (2) при заданной константе $\delta$.

Необратимость функции отражает тот факт, что функция крайне чувствительна к погрешности вычисления. Поиск решения обратной задачи (вычисления обратной функции) затруднен ее плохой обусловленностью и малейшие изменения аргументов приводят к значительным отличиям результатов вычислений. Малые невязки отражают значительные ошибки при вычислении неизвестных.

**Определение 3. Функции шифрования и дешифрования представимых вещественных чисел**

Если мы имеем две функции $f(u_1, k_1)$ и $g(u_2, k_2)$, зависящие от $u_1$ и $u_2$ и ключей $k_1$ и $k_2$, такие, что $f(g(u, k_2), k_1) = u$ для некоторой пары ключей $k_1$ и $k_2$ и любого $u$, тогда $g(.)$ и $f(.)$ – функции шифрования и дешифрования соответственно. Результат $s = g(u, k_1)$ называется криптограммой (или шифрограммой).

К функциям $g(.)$ и $f(.)$ предъявляются следующие требования:

1) должны быть легко вычислимы для любых $u$, если известны $k_1$ и $k_2$;

2) вычисление $f(u, k_1)$ – трудная задача при неизвестном ключе $k_1$ (т.е., не зная ключа дешифрования, мы не можем вычислить исходное сообщение по криптограмме);

3) вычисление ключей $k_1$ и $k_2$ – нелегкая задача при наличии некоторого набора пар $<u, s>$ (имея набор криптограмм и исходных сообщений, мы не можем вычислить ключи);

4) вычисление $k_2$ (в случае $k_2$, отличного от $k_1$) при известном $k_1$ также должно быть трудной задачей (это требование относится к асимметричным шифрам. Для цифровой подписи $k_2$ и $k_1$ меняются ролями).

Надежность шифра определяется именно по его соответствию вышеперечисленным требованиям, при этом считается, что алгоритмы вычисления $g(.)$ и $f(.)$ известны всем (есть еще вариант так называемой Security by obscurity, т.е. защиты из-за неизвестности алгоритма, но он при оценке стойкости шифров не рассматривается) [13].

Если $k_1 = k_2$, то шифр называется симметричным, в противном случае – асимметричным. Примеры известных симметричных шифров: DES, IDEA, Blowfish, ГОСТ; асимметричных – RSA, ECC [13].

**Определение 4. Хеш-функции представимых вещественных чисел**

Основное предназначение криптографических хешей – контроль подлинности данных путем вычисления от них некоторой функции $h(.)$, дающей результат фиксированной (и обычно небольшой длины). Функция $h(.)$ должна удовлетворять следующим требованиям:

1) для любых сообщений $u$, $s = h(u)$ легко вычисляема;

2) задача нахождения такого $u_2$, отличного от $u$, чтобы $h(u_2) = s$, должна являться трудной при неизвестном $u$;

3) задача нахождения такого $u_2$, что $h(u) = h(u_2)$, является трудной при известном $u$.

Большинство популярных хеш-функций генерируют хеш длиной 128 бит и более. Примерами наиболее распространенных хеш-функций являются MD5 и SHA. Значения хеш-функций часто используются в системах электронной цифровой подписи для генерации дайджеста сообщения, который затем и подписывается тем или иным алгоритмом. Также хеш-функции применяются в системах аутентификации для проверки паролей – открытый пароль пользователя не должен храниться в системе, вместо него хранится его хеш, который затем и сравнивается с хешем от пароля, вводимого пользователем при входе в систему [13].

### Векторные функции представимых вещественных чисел

Поскольку рассмотренные квазиобратимые и необратимые функции представимых вещественных чисел замкнуты относительно универсума $\boldsymbol{D}$, то и в векторной постановке они сохраняют свои свойства квазиобратимости и необратимости. Верхней оценкой отклонения для квазиобращения в относительных величинах будет рассматриваться $\delta_u$, определяющая отклонение некоторой векторной нормы по вектору $\boldsymbol{\delta}$.

Именно векторная постановка задачи построения шифрующих, дешифрующих и хеш-функций, определенных на дискретном множестве представимых в машинном формате вещественных чисел, и является ключевой в данной работе.

**Новизна** состоит в предложении избирательного криптографического преобразования, в рамках которого шифрующая и дешифрующая функции сохраняют свои функции обратимости только для некоторого конечного подмножества шифруемых сообщений, для остальных, строго говоря, обладают свойством квазиобратимости или даже необратимости.

### Особенности множества представимых вещественных чисел D

Дискретное конечное множество представимых в машинном формате вещественных чисел $\boldsymbol{D}$ формируется из бесконечного множества вещественных чисел $\boldsymbol{R}$ заданием формата числа (разрядности мантиссы и порядка, а также правил интерпретации выбранного представления).

Вещественные числа из множества $\boldsymbol{D}$ расположены в метрическом пространстве $\boldsymbol{R}$ с заданной оценкой расстояния (метрикой) (–) неравномерно. Блоки чисел, образованные полным перебором всех вариантов мантиссы, следуют с двукратным расширением/сжатием при увеличении/уменьшении порядка.

Это множество обладает симметрией относительно числа 0. Оно при рассмотрении подмножества положительных чисел ограничено сверху и снизу максимальным и минимальным числом.

### Особенности квазиобратимых функций представимых вещественных чисел

Класс рассматриваемых функций очень широк. Практически все известные нам трансцендентные функции обладают свойством квазиобратимости или необратимости, если они рассматриваются как функции, замкнутые относительно $\boldsymbol{D}$. Иными словами, компьютерные реализации трансцендентных функций вносят погрешности, которые превращают их в квазиобратимые или необратимые функции.

Приведем простейший пример функции, заданной произведением аргумента на число, и обратной ей функции деления (умножения на обратное число):

$$\begin{aligned} f(x,v,k) &= \frac{v+k}{x}; \\ g(x,f,k) &= fx-k, \end{aligned} \tag{3}$$

где $x \in \boldsymbol{D}$; $v$ – вектор эталонов; $k$ – дополнительный ключ.

Отклонение значений восстанавливаемого аргумента в результате такого функционального преобразования иллюстрирует рисунок 1.

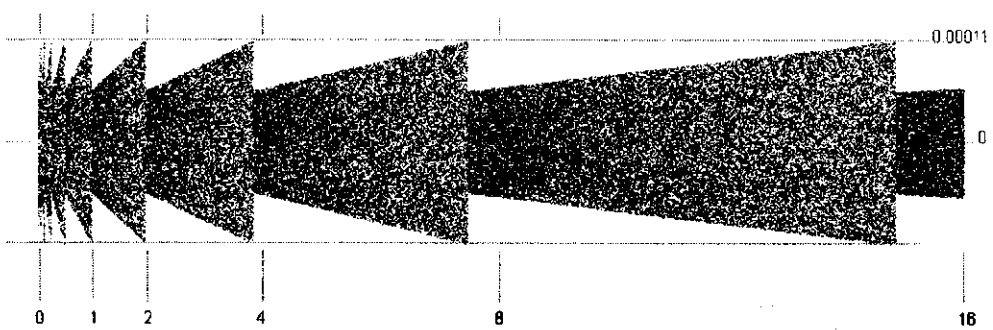


Рис. 1. Распределение погрешности квазиобращения
для операций деления и умножения для чисел в интервале от 0 до 16

Введение параметра $k$ позволяет дополнительно увеличить погрешность на заданное количество разрядов, в частности, в рассматриваемом примере погрешность колебалась в пределах ±0,00011.

Погрешность квазиобращения будет определяться по формуле:

$$\Delta_v = \left\| g(x,f,k) - v \right\|. \tag{4}$$

Распределение погрешности носит случайный характер[1], однако четко прослеживается скачкообразное изменение пределов разброса восстанавливаемых значений, связанное с переходом степени числа два. Причем уникальные штампы разброса формируют полные переборы мантиссы, а это 4 503,599 627 370 496 триллиона комбинаций для мантиссы 52 бита (без учета знака) и 9 007,199 254 740 992 триллиона комбинаций для мантиссы 53 бита.

Изменение степени числа два, по сути, смещает уникальный штамп, пропорционально расширяя (в 2 раза) или сужая его.

Так, штамп, полученный прострелом 1000 точек интервала от 1 до 2, идентичен штампу по интервалу от 2 до 4, что иллюстрирует рисунок 2, также наблюдается зеркальная симметрия относительно 0.

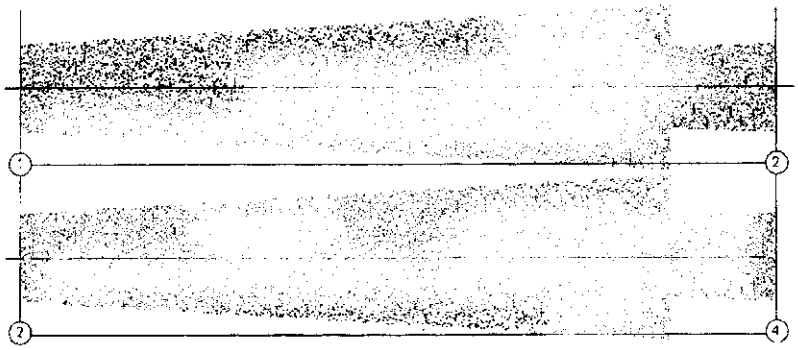


Рис. 2. Распределение погрешности квазиобращения для операций деления и умножения в интервале от 1 до 2 и в интервале от 2 до 4 – идентичны

При постановке задачи в виде (3) будем говорить, что требуется определить такой вектор ключей $x$, что эталонный вектор $v$ восстанавливается без погрешности. Для вектора $v$, элементы которого не превосходят 3-х порядков и ключа $k$ 12-го порядка, происходит потеря 9 порядков, это для чисел двойной точности означает, что требование точного восстановления эталонного вектора обеспечивает одно число на $10^8$ чисел. Оценка приблизительная и требует уточнения для каждого конкретного $k$.

В данном случае функциональное преобразование является простым и допускает анализ, позволяющий вычислить ключи, минимизируя перебор чисел. Но оно наглядно иллюстрирует разрабатываемую идею.

Более сложные функциональные преобразования, как будет показано далее, устраняют повторяемость штампов, симметрию и даже обеспечивают невозможность обратного преобразования, в силу того, что погрешность вычисления превышает результат. Рассмотрим экспоненциально-логарифмическое и логарифмически-синусоидальное преобразование.

Зададим ключи $P1 := 1{,}08 \cdot 10^8$ $P2 := 1 \cdot 10^{-6}$.

Зададим экспоненциальную функцию

$$y(x) := exp(x \cdot P2) \tag{5}$$

и обратную ей функцию логарифма

$$x2(x) := \frac{\ln(y(x))}{P2} - x. \tag{6}$$

Более сложную топологию погрешности квазиобращения иллюстрирует рисунок 3.

Логарифмически-синусоидальная функция обладает свойством необратимости:

$$y(x) := \sin(\ln(x)) \cdot P1). \tag{7}$$

Особенностью данного преобразования является то, что функция логарифма нелинейно сжимает пространство с 308 порядка до верхней границы в 709, так что исходное пространство оказывается разбито на классы эквивалентности, в пределах которых значение логарифма не изменяется. Умножение на значительный параметр P1 приводит к тому, что малейшее изменение мантиссы логарифма изменяет синус на величину $10^{-6}$. Причем характер распределения результатов вычисления функции не изменяется в зависимости от масштаба анализа, что показано на рисунках 4 – 6.

[1] Этот тезис требует доказательства, но в рамках данной статьи результаты исследования различных распределений погрешности на соответствие равномерному закону распределения мы не приводим. Отметим только, что характер распределения таков, что позволяет говорить о распределениях погрешности как о псевдослучайных последовательностях, соответствующих равномерному закону распределения вероятности в некотором диапазоне значений.

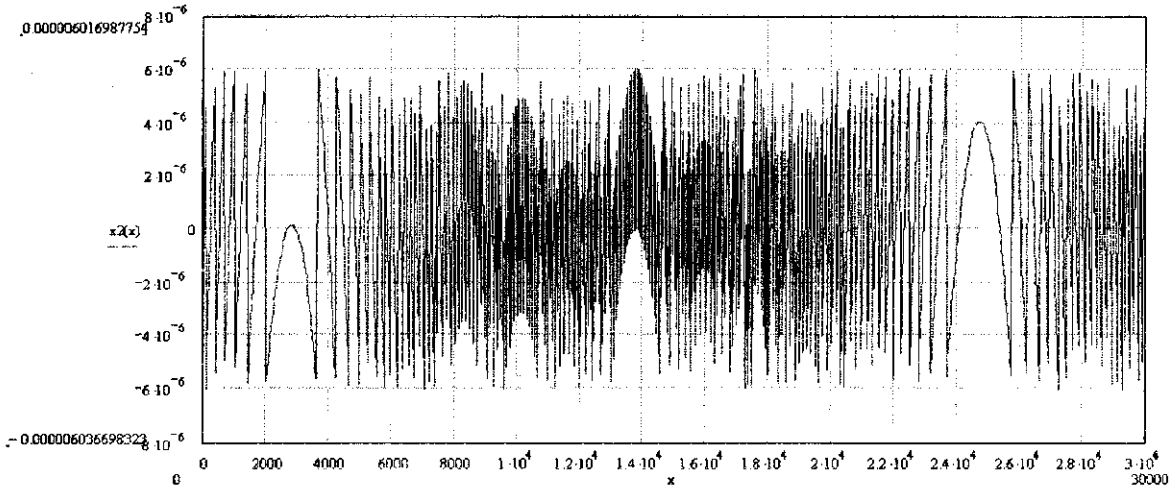


Рис. 3. Распределение погрешности квазиобращения
экспоненциально-логарифмической функции на интервале до 30 000

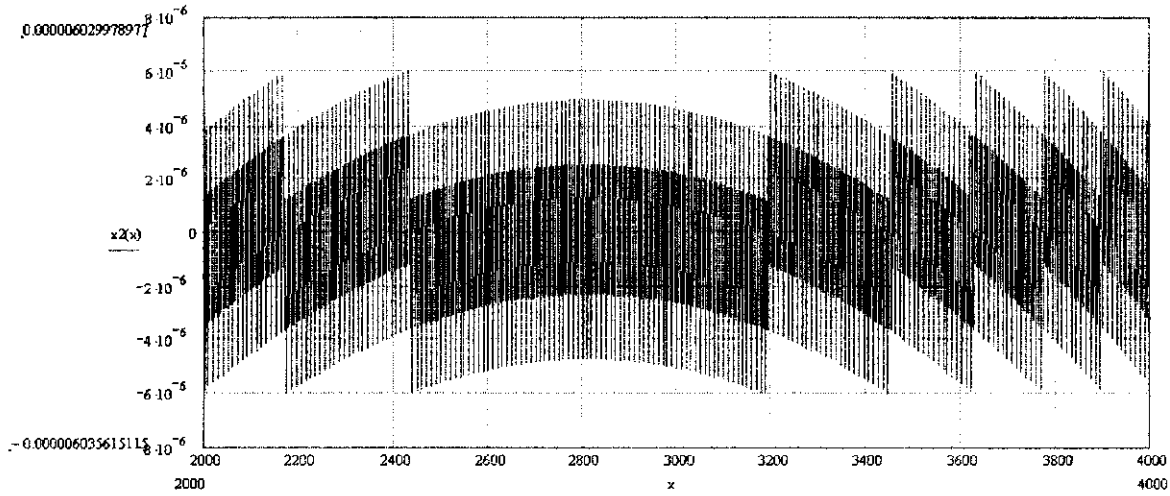


Рис. 4. Распределение погрешности квазиобращения
экспоненциально-логарифмической функции на интервале от 2000 до 4000

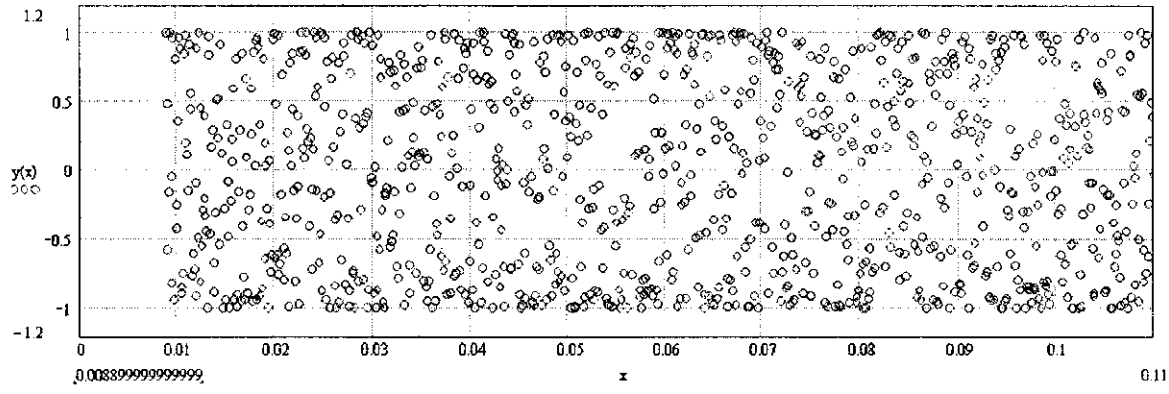


Рис. 5. Логарифмически-синусоидальное преобразование
для чисел в интервале 0…0,11

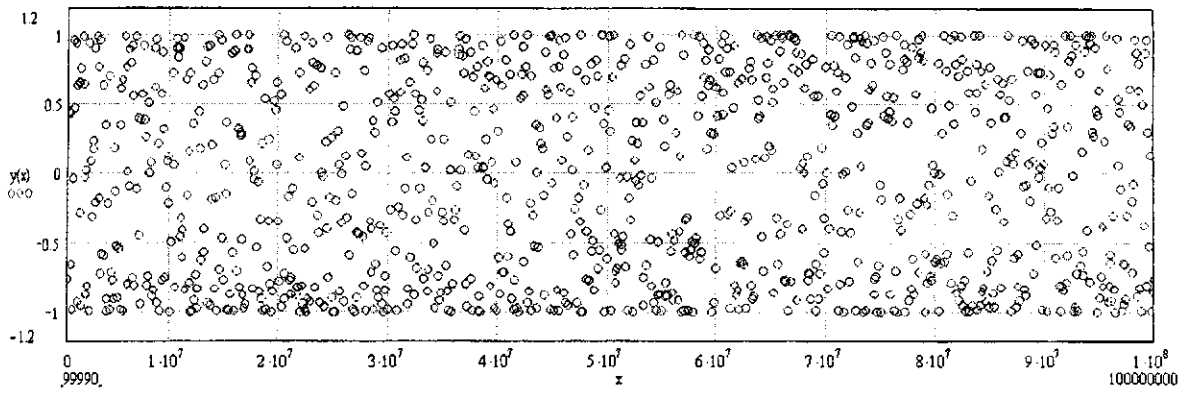


Рис. 6. Логарифмически-синусоидальное преобразование для чисел в интервале $0...10^8$

Интересной задачей в случае рассмотрения необратимой функции является задача поиска таких $x$, которые не изменяют число 1, как показано (8), или не изменяют заданный эталонный вектор:

$$x2(x) := 1 + \frac{P2}{\frac{1}{y(x)}}. \qquad (8)$$

Иными словами, например, такие $x$, значения необратимой функции от которых лежат в узком коридоре $\pm 10^{-6}$.

Варьируя ширину коридора, можно обеспечить управление степенью криптозащиты.

Недостатком необратимых функций является ресурсоемкая процедура поиска валидных ключей даже для разработчика защиты.

Рассмотренные примеры позволяют говорить о большом разнообразии вариантов защиты научного ПО, основанного на квазиобратимых и необратимых функциях представимых вещественных чисел. Нами разработан программный комплекс защиты ПО электронными ключами, предоставляющий разработчику научного ПО полный комплекс услуг от включения средств защиты в код приложения до создания дистрибутива и сайта технической поддержки лицензионных пользователей защищаемого ПО.

**Выводы**

В данной работе предложен ряд **новых** положений:

1. Предложен подход к построению систем защиты ПО от несанкционированного использования, основанный на применении методов криптографии, чувствительных к погрешности компьютерных вычислений.

2. Рассмотрены особенности дискретного пространства представимых в машинном формате вещественных чисел двойной точности, функций, замкнутых относительно данного пространства, а также свойства обратимости, квазиобратимости и необратимости таких функций.

3. Рассмотрена векторная постановка задачи обратимости для квазиобратимых функций представимых вещественных чисел, обеспечивающая избирательный характер действия криптографического преобразования.

4. Предложены реализации шифрующих, дешифрующих и хеш-функций избирательного действия на основе квазиобратимых и необратимых функций представимых вещественных чисел.

## ЛИТЕРАТУРА

1. Середа С.А. Оценка эффективности систем защиты программного обеспечения // IPSIT'99: Материалы междунар. конф., 1999.
2. Новичков А., Сардарян Р. Анализ рынка средств защиты от копирования и взлома программных средств // Pc Week. – 2004. – № 6.
3. Воробьев А.А. Адаптивное управление процессами защиты информации от несанкционированного доступа в автоматизированных системах: Автореф. дис.... д-ра техн. наук. – Люберцы, 2004. – 34 с.

4. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа: Наука и техника. – СПб., 2004
5. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
6. Ковалевский В., Максимов В. Криптографические методы // КомпьютерПресс. – 1993. – № 5. – С. 31 – 34.
7. Водолазский В. Коммерческие системы шифрования: основные алгоритмы и их реализация. Ч. 1 // Монитор. – 1992. – № 6 – 7. – С. 14 – 19.
8. Чернов А.В. Интегрированная среда для исследования «обфускации» программ: Докл. на конф., посв. 90-летию со дня рожд. А.А.Ляпунова. Россия, Новосибирск, 8 – 11 октября 2001 года. – [Electronic resource]. – Mode of access: http://www.ict.nsc.ru/ws/Lyap2001/2350/
9. [Electronic resource]. – Mode of access: http://www.x86.org/secrets/Dan0411.html
10. Руководство программиста по архитектуре Intel. Т. 2. (Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference Manual)
11. Глухов Д.О. Численный анализ расчетной модели нормального сечения железобетонной конструкции проекта СНБ 5.03.01-98 // Вклад вузовской науки в развитие приоритетных направлений производственно-хозяйственной деятельности, разработку экономичных и экологически чистых технологий и прогрессивных методов обучения: Материалы 54-й междунар. науч.-техн. конф. – Ч. 7. – Мн.: БГПА, 2000.
12. Каханер Д., Моулер К., Нэш С. Численные методы и математическое обеспечение: Пер с англ. – М.: Мир, 1998. – 575 с.
13. Остапенко А. Хеширование, шифрование и цифровая подпись с использованием CryptoAPI и .NET // RSDN Magazine – 2005. – № 1.
14. Петраков А.В., Лагутин В.С. Телеохрана. – М.: Энергоатомиздат, 1998. – С. 245 – 257.