



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 7/50 (2019.02); G06F 7/523 (2019.02); G06F 7/72 (2019.02); H04L 9/0618 (2019.02); H04L 2209/16 (2019.02)

(21)(22) Заявка: 2017114868, 30.09.2015

(24) Дата начала отсчета срока действия патента:
30.09.2015Дата регистрации:
30.09.2019

Приоритет(ы):

(30) Конвенционный приоритет:
30.09.2014 EP 14186951.1

(43) Дата публикации заявки: 07.11.2018 Бюл. № 31

(45) Опубликовано: 30.09.2019 Бюл. № 28

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 02.05.2017(86) Заявка РСТ:
EP 2015/072635 (30.09.2015)(87) Публикация заявки РСТ:
WO 2016/050884 (07.04.2016)Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, стр. 3, ООО
"Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

МАРИН Леандро (NL),
БРЮКЕРС Альфонс Антониус Мария
Ламбертус (NL),
ГОРИССЕН Паулус Матхиас Хюбертус
Мехтилдис Антониус (NL)

(73) Патентообладатель(и):

КОНИНКЛЕЙКЕ ФИЛИПС Н.В. (NL)

(56) Список документов, цитированных в отчете
о поиске: US 5270956 A, 14.12.1993. WO 2004/
112307 A2, 23.12.2004. WO 2009/076669 A1,
18.06.2009. WO 2014/016795 A2, 30.01.2014. RU
2467389 C1, 20.11.2012. SU 1667059 A2,
30.07.1991.(54) ЭЛЕКТРОННОЕ ВЫЧИСЛИТЕЛЬНОЕ УСТРОЙСТВО ДЛЯ ВЫПОЛНЕНИЯ АРИФМЕТИКИ
С ОБФУСКАЦИЕЙ

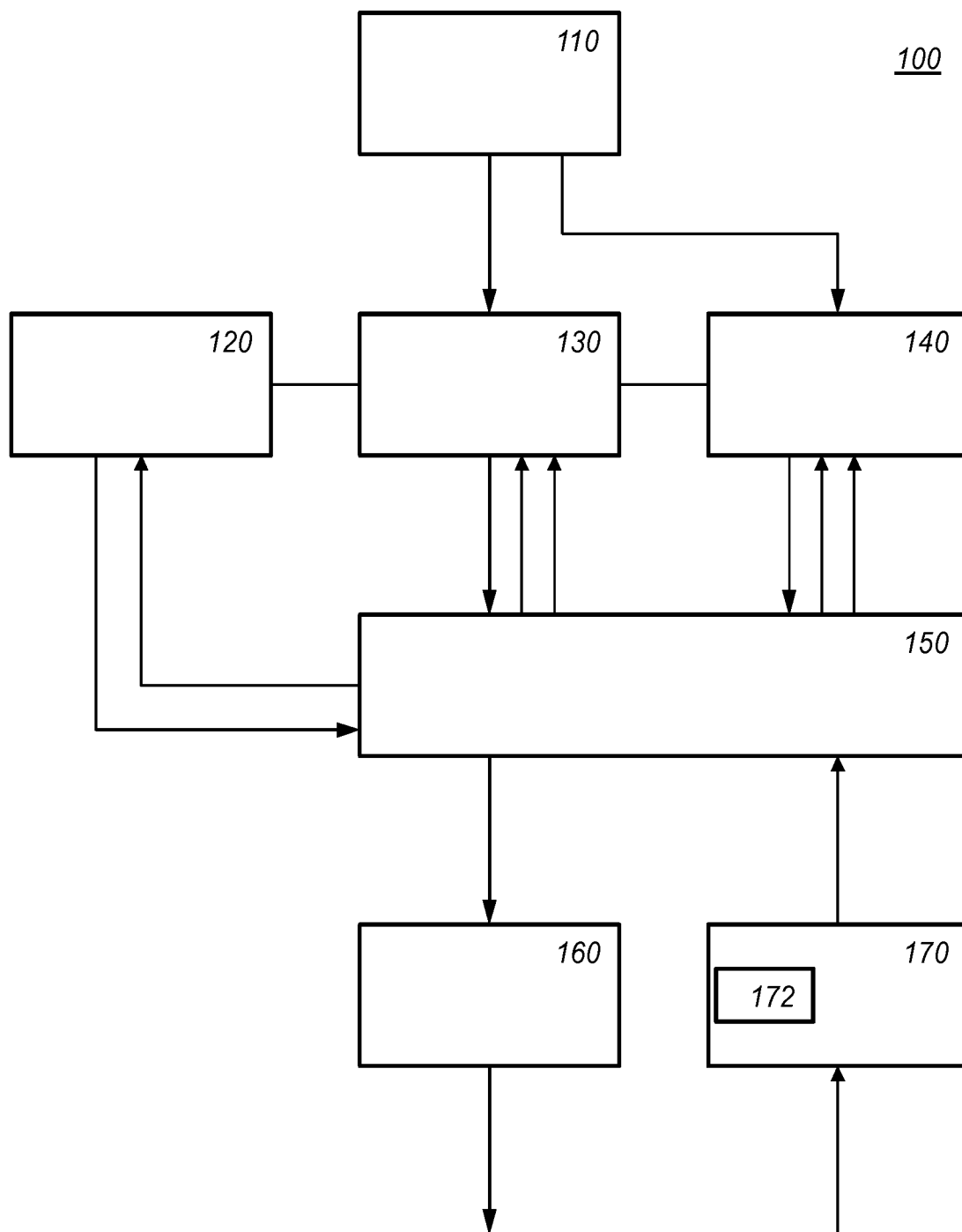
(57) Реферат:

Группа изобретений относится к области вычислительной техники и может быть использована для выполнения арифметики с обфускацией в коммутативном кольце. Техническим результатом является повышение защищенности. Устройство содержит хранилище, выполненное с возможностью хранения таблицы приращений (T), определенной для приращения кольцевого элемента $(1; u^t)$, причем таблица приращений отображает входной кольцевой

элемент $(k = u^{k_1} - u^{k_2})$ в выходной целочисленный список $(T((k_1, k_2)) = (l_1, l_2))$, кодирующий выходной кольцевой элемент $(l = u^{l_1} - u^{l_2})$, так, что выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом $(l = k + 1)$. С использованием таблицы приращений блок кольцевого сложения

складывает первый входной для сложения целочисленный список $((a_1, a_2))$, кодирующий первый входной для сложения кольцевой элемент, и второй входной для сложения целочисленный

список $((b_1, b_2))$, кодирующий второй входной для сложения кольцевой элемент. Устройство может содержать блок кольцевого умножения, также использующий таблицу приращений. 6 н. и 11 з.п. ф-лы, 10 ил., 1 табл.



ФИГ. 1А



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 7/50 (2019.02); *G06F 7/523* (2019.02); *G06F 7/72* (2019.02); *H04L 9/0618* (2019.02); *H04L 2209/16* (2019.02)

(21)(22) Application: **2017114868, 30.09.2015**(24) Effective date for property rights:
30.09.2015Registration date:
30.09.2019

Priority:

(30) Convention priority:
30.09.2014 EP 14186951.1(43) Application published: **07.11.2018 Bull. № 31**(45) Date of publication: **30.09.2019 Bull. № 28**(85) Commencement of national phase: **02.05.2017**(86) PCT application:
EP 2015/072635 (30.09.2015)(87) PCT publication:
WO 2016/050884 (07.04.2016)

Mail address:

**129090, Moskva, ul. B. Spasskaya, 25, str. 3, OOO
"Yuridicheskaya firma Gorodisskij i Partnery"**

(72) Inventor(s):

**MARIN Leandro (NL),
BRUEKERS Alphons Antonius Maria
Lambertus (NL),
GORISSEN Paulus Mathias Hubertus Mechtildis
Antonius (NL)**

(73) Proprietor(s):

Koninklijke Philips N.V. (NL)

(54) **ELECTRONIC COMPUTER FOR PERFORMING ARITHMETIC WITH OBFUSCATION**

(57) Abstract:

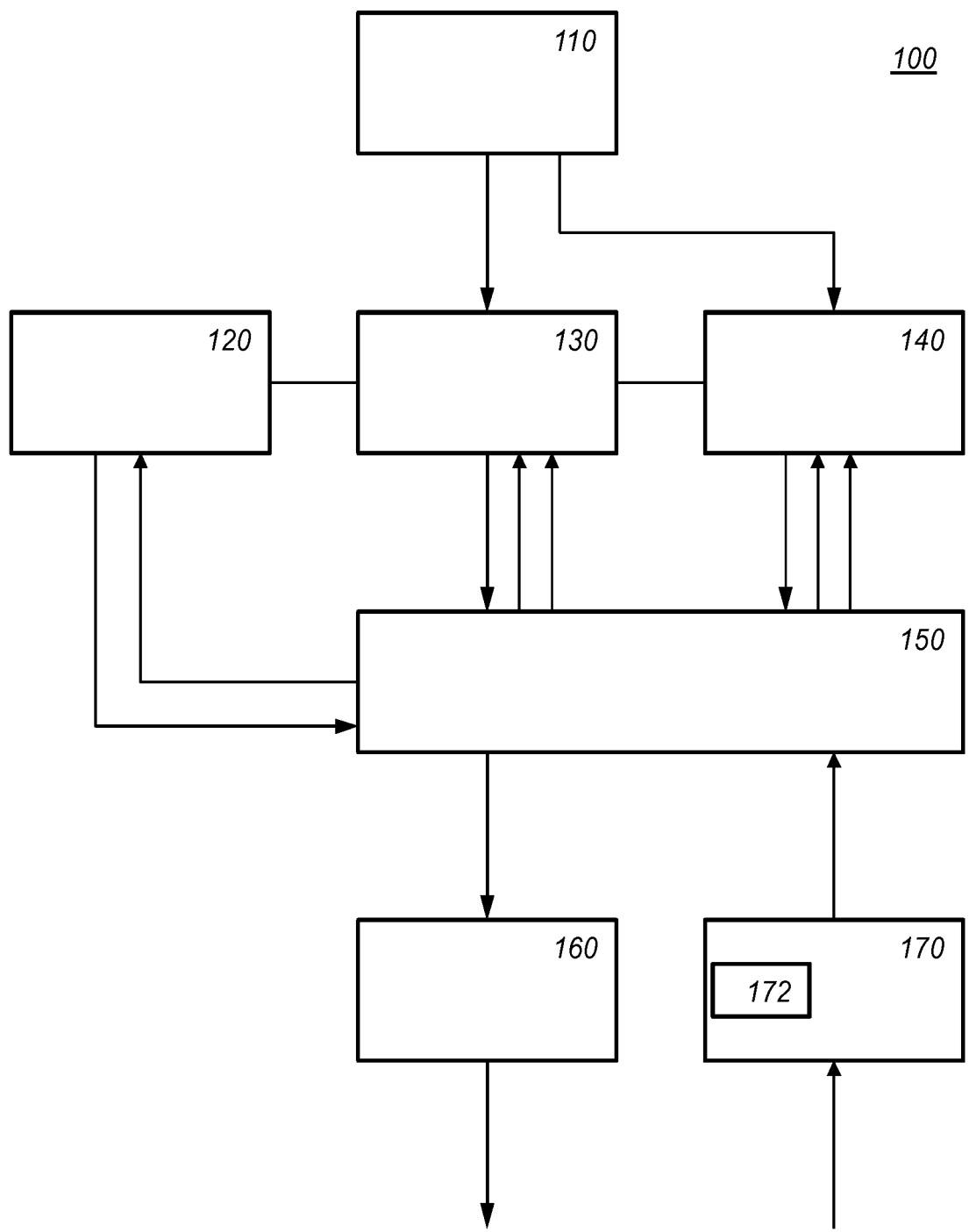
FIELD: calculating; counting.

SUBSTANCE: group of inventions relates to computer engineering and can be used to perform obfuscation arithmetic in a commutative ring. Device comprises a storage configured to store a table of increments (I), determined for increment of annular element ($1; u^l$), wherein table of increments displays input circular element ($k = u^k_I - u^k_2$) to output integer list ($T((k_1, k_2) = (I_1, I_2))$ encoding output circular element ($I = u^l_I - u^l_2$), so, that outlet annular element is equal to

annular increment element, annularly folded with inlet annular element ($I = k + I$). Using a table of increments, the ring addition unit adds the first input integer list ((a_1, a_2)), encoding a first input adder for addition of a circular element, and a second input for addition of an integer list ((b_1, b_2)) encoding the second input element for addition of a ring element. Device may contain an annular multiplying unit which also uses a table of increments.

EFFECT: technical result is improved security.

17 cl, 10 dwg, 1 tbl



ФИГ. 1А

ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ

Изобретение относится к электронному вычислительному устройству, устройству кольцевого кодирования, устройству кольцевого декодирования, устройству вычисления таблицы, способу электронного вычисления, компьютерной программе и

5 машиночитаемому носителю.

УРОВЕНЬ ТЕХНИКИ

В криптографии типа "белый ящик" и, в общем, в программной обфускации вычисления часто выполняются над закодированными значениями вместо простых значений. Обратное проектирование программных средств с обфускацией сложнее,

10 если вычисления выполняются над закодированными значениями вместо самих открытых значений.

После кодирования обычные операции, такие как сложение или умножение, больше не могут выполняться с использованием встроенных примитивов компьютера. Прямое сложение закодированных значений в обычном случае не дает в результате кодирования

15 сложения значений. То же самое относится к умножению. В виде формулы:

$E(x) + E(y) \neq E(x + y)$ для большинства x и y ; E обозначает функцию кодирования.

Решение этой проблемы состоит во введении таблиц сложения (A) и умножения (M). Таблицы получают два закодированных значения в качестве входных данных и производят закодированное значение в качестве выходных данных, которое

20 соответствует кодированию операции сложения или умножения. Таблицы могут быть определены следующим образом: $A(E(x), E(y)) = E(x + y)$; $M(E(x), E(y)) = E(xy)$. Эти таблицы обеспечивают возможность арифметике выполняться непосредственно над закодированными значениями.

Сложение и умножение с обфускацией с использованием таблиц страдает по меньшей

25 мере от двух недостатков. Во-первых, таблицы могут стать довольно большими. Если x и y представлены как l бит, каждой таблице требуется 2^{2l} бит.

Во-вторых, такие большие таблицы могут быть легко найдены в программных средствах. Что еще хуже, таблицы все еще могут быть идентифицированы как операции

30 сложения или умножения, даже несмотря на то, что они закодированы; например, через свойства этих функций, которые сохраняются в кодировании. Например, таблица умножения удовлетворяет $M(E(0), E(x)) = E(0)$. Злоумышленник может использовать это и подобные свойства, чтобы догадаться, какую операцию таблицы представляют.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Наличие улучшенного способа для выполнения арифметики с обфускацией обеспечит

35 преимущества. Обеспечено вычислительное устройство, определенное в формуле изобретения.

Изобретатели обнаружили, что в некоторых случаях умножение и сложение над закодированными значениями может выполняться с использованием одной таблицы

40 без необходимости кодирования множества значений в одно закодированное значение. Поскольку одна и та же таблица используется для сложения и умножения, будет сложно увидеть во время обратного проектирования, сложение или же умножение выполняется.

Поскольку сложение и умножение выглядят как одна и та же операция при взгляде извне, изобретатели назвали этот способ термином "гомогенная обфускация". Даже если злоумышленнику удалось найти таблицу, которая используется, и даже если ему

45 удалось каким-то образом понять ее функцию как таблицы приращений, ему все еще будет неизвестно, выполняется ли операция сложения или же умножения. То, как таблица действует на элемент целочисленного списка, будет отличаться для сложения и умножения, однако это может быть легко скрыто с использованием традиционной

обфускации, такой как обфускация кода, осуществление "белого ящика" и т. д.

Дополнительно, одна таблица, которая используется, также меньше рассмотренной в уровне техники: необходимо приблизительно 2^l бит. Даже если используется только сложение, таблица, необходимая для сложения с обфускацией, меньше таблицы, предлагаемой в уровне техники.

Изобретение применяется к множеству различных коммутативных колец R , хотя не все без исключения кольца обеспечивают возможность кодирования в виде целочисленных списков. Коммутативные кольца являются математической концепцией, которая включает в себя множество различных известных математических структур, например целые по модулю некоторого числа ($\mathbb{Z}_n$) или многочлены по модулю некоторого числа и многочлена ($\mathbb{Z}_n[x]/f(x)$). Поля являются частным случаем коммутативных колец. Как будет описано здесь, специалист может удостовериться, обеспечивает ли некоторое заданное кольцо возможность обфускации.

Например, кольцевой элемент может быть закодирован в виде двух целых (a, b) . Арифметика может выполняться непосредственно над кодированием с использованием таблицы приращений, которая отображает закодированный кольцевой элемент в закодированный кольцевой элемент плюс значение приращения. Например, таблица может отображать (a, b) в (c, d) , если $u^c - u^d = u^a - u^b + 1$. И сложение, и умножение выполняются путем многократных применений таблицы приращений.

Как будет здесь рассмотрено более полно, существует множество других возможностей и вариантов. Обычно неизвестно злоумышленнику, какой из многих вариантов был выбран в любом конкретном осуществлении.

Вычислительное устройство является электронным устройством и может быть мобильным электронным устройством, например мобильным телефоном, ресивером цифрового телевидения, компьютером, интеллектуальной картой и т. д.

Арифметика с обфускацией, описанная здесь, может применяться в широком диапазоне практических приложений. Такие практические приложения включают в себя безопасные приложения, запущенные на частных аппаратных средствах, например банковские приложения и т. д., причем обратное проектирование должно быть предотвращено. Другие приложения включать в себя приложения, в которых случайная утечка данных должна быть предотвращена. Если программу обманом заставляют раскрыть приватные данные, эта проблема не такая большая, если просочившиеся данные закодированы. Арифметика с обфускацией может также применяться к серверам, на которых запущены приложения. Приватность увеличивается, если пользователи посылают и принимают данные в закодированной форме.

Способ согласно изобретению может осуществляться на компьютере в качестве компьютерно-реализованного способа или в специализированных аппаратных средствах или в комбинации того и другого. Исполняемый код или его части для способа согласно изобретению могут сохраняться в компьютерном программном продукте. Примеры компьютерных программных продуктов включают в себя устройства памяти, оптические устройства хранения, интегральные цепи, серверы, сетевые программные средства и т. д. Предпочтительно, компьютерный программный продукт содержит некратковременные средства программного кода, сохраненные на машиночитаемом носителе, для выполнения способа согласно изобретению, когда упомянутый программный продукт исполняется на компьютере.

В предпочтительном варианте осуществления компьютерная программа содержит средства компьютерного программного кода, выполненные с возможностью

выполнения всех этапов способа согласно изобретению, когда компьютерная программа запущена на компьютере. Предпочтительно, компьютерная программа осуществляется на машиночитаемом носителе.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

- 5 Эти и другие аспекты изобретения очевидны из и будут освещены со ссылками на варианты осуществления, описанные далее. На чертежах
- фиг.1a схематически изображает пример варианта осуществления вычислительного устройства 100,
- 10 фиг.1b схематически изображает пример варианта осуществления блока 130 кольцевого сложения,
- фиг.1c схематически изображает пример варианта осуществления блока 140 кольцевого умножения,
- фиг.2 схематически изображает пример варианта осуществления вычислительного устройства 101,
- 15 фиг.3 схематически изображает пример варианта осуществления устройства 200 вычисления таблицы для вычисления таблицы приращений для использования в вычислительном устройстве,
- фиг.4 схематически изображает пример варианта осуществления способа 300 вычисления для выполнения арифметики с обфускацией,
- 20 фиг.5 схематически изображает пример варианта осуществления способа 400 сложения,
- фиг.6 схематически изображает пример варианта осуществления способа 500 умножения,
- фиг.7a изображает машиночитаемый носитель, имеющий перезаписываемую часть,
- 25 содержащую компьютерную программу согласно одному варианту осуществления,
- фиг.7b изображает схематическое представление процессорной системы согласно одному варианту осуществления.

Элементы, которые имеют одни и те же ссылочные позиции на различных чертежах, имеют одни и те же структурные признаки и одни и те же функции или являются одними

30 и теми же сигналами. Если функция и/или структура такого элемента уже была объяснена, нет необходимости повторять ее объяснение в подробном описании.

ПОДРОБНОЕ ОПИСАНИЕ

Хотя настоящее изобретение допускает осуществление во множестве различных форм, на чертежах показаны и будут подробно описаны здесь один или несколько

35 конкретных вариантов осуществления с пониманием, что настоящее раскрытие должно рассматриваться в качестве примера принципов изобретения и не предназначено для ограничения изобретения конкретными вариантами осуществления, показанными и описанными.

Далее в целях понимания описаны элементы вариантов осуществления в действии.

40 Однако будет очевидно, что соответственные элементы выполнены с возможностью выполнения функций, описанных как выполняемые ими.

Электронное вычислительное устройство выполняет эффективную арифметику с использованием неожиданно маленьких таблиц. Кроме того, в области техники арифметики с обфускацией считается преимуществом, если операция может выполняться

45 посредством таблицы, поскольку для таких операций может быть легко осуществлена дополнительная обфускация, например с использованием традиционных методик "белого ящика" (см., например, Чоу и др., "Криптография типа "белый ящик" и осуществление AES" ("White-box cryptography and an AES implementation")). Существует,

таким образом, необходимость выразить арифметические операции с использованием таблиц. Варианты осуществления осуществляют сложение с использованием меньшей таблицы, чем в предшествующем уровне техники. Даже без дополнительной обфускации, такой как криптография типа "белый ящик", электронное вычислительное устройство

5 участвует в обфускации. Как показано здесь, существует множество способов, которыми таблица кодирования и приращений может осуществляться. То, какое кодирование используется в любом конкретном варианте осуществления, неизвестно злоумышленнику, и, таким образом, наблюдаемое вычисление становится сложнее интерпретировать.

10 Варианты осуществления обеспечивают возможность операций умножения и сложения, которые должны выполняться с использованием одной и той же таблицы. Это дополнительно способствует обфускации, поскольку из того факта, что таблица приращений используется, уже нельзя определить, какая операция выполняется. Ниже сначала рассматривается некоторое количество возможных архитектур вариантов

15 осуществления вычислительных устройств. Далее рассматривается некоторое количество альтернативных способов для выполнения арифметики с обфускацией.

Фиг.1 схематически изображает пример варианта осуществления вычислительного устройства 100. Вычислительное устройство 100 является электронным устройством для выполнения арифметики с обфускацией в конечном коммутативном кольце. Известно

20 множество примеров коммутативных колец. Ниже даны примеры для двух таких колец: целые по модулю некоторого числа ($\mathbb{Z}_n$) и многочлены по модулю некоторого числа и многочлена ($\mathbb{Z}_n[x]/f(x)$). Другой вариант осуществления может использовать другие коммутативные кольца.

Элементы кольца называются кольцевыми элементами. Над кольцевыми элементами

25 определены сложение и умножение, последние называются кольцевым сложением и кольцевым умножением.

Кольцевые элементы могут быть представлены в любой подходящей форме, если это требуется. Например, элементы $\mathbb{Z}_n$ могут быть представлены как целые; элементы

30 $\mathbb{Z}_n[x]/f(x)$ - как многочлены. Однако в вычислительном устройстве 100 кольцевые элементы представляются как целочисленные списки. Например, кольцевой элемент a может быть представлен в вычислительном устройстве 100 списком (a_1, a_2) . Последнее верно даже для нецелочисленных колец, например колец многочленов. Целочисленный список кодирует кольцевой элемент согласно некоторому отображению между

35 кольцевыми элементами и целочисленным списком; при заданном любом кольцевом элементе существует по меньшей мере один целочисленный список, который представляет кольцевой элемент, и при заданном любом целочисленном списке существует ровно один кольцевой элемент, который его представляет. В вариантах осуществления любой кольцевой элемент может быть представлен в виде целочисленного

40 списка.

Целочисленные списки имеют по меньшей мере два элемента. Как оказывается, операции сложения и умножения требуют меньшего количества этапов, если целочисленный список короче. Соответственно, в одном варианте осуществления целочисленные списки всегда имеют два элемента. В основном описании мы будем

45 предполагать, что целочисленные списки являются целочисленными парами, однако примеры целочисленных списков, имеющих более двух элементов, обеспечены. В качестве примера, (a_1, a_2) может отображаться в кольцевой элемент $u^{a_1} - u^{a_2}$, причем u является особым кольцевым элементом, называемым кольцевым элементом основания.

Ниже рассматривается множество вариантов, включающих в себя использование множества элементов основания. Однако в основном рассмотрении мы будем предполагать в качестве "примерного кодирования", что некоторый заданный целочисленный список (a_1, a_2) отображается в кольцевой элемент $(u^{a_1} - u^{a_2})$.

В одном варианте осуществления целые в целочисленном списке неотрицательны. Это упрощает вычисление, но не является необходимым. Кроме того, в одном варианте осуществления целые в целочисленном списке берутся по модулю порядка элемента основания. Порядком элемента основания u является наименьшее целое k такое, что $u^k = 1$. Удобно хранить значения в целочисленном списке в диапазоне $[0, k - 1]$, например путем выполнения операций по модулю k .

Вычислительное устройство 100 может содержать хранилище 150 операндов. Операнды сохраняются в виде целочисленных списков в хранилище 150 операндов. Арифметика может выполняться над операндами, сохраненными в хранилище 150 операндов. Результаты упомянутой арифметики могут сохраняться в хранилище 150 операндов, где они могут быть использованы в новых операциях, или могут выводиться к другому устройству и т. д.

Вычислительное устройство 100 содержит хранилище 110, выполненное с возможностью хранить таблицу приращений T , определенную для приращения кольцевого элемента. Таблица приращений отображает входной кольцевой элемент в выходной целочисленный список, кодирующий выходной кольцевой элемент, так, что выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом. В одном варианте осуществления входной кольцевой элемент представлен в виде целочисленного списка. Таким образом, таблица T отображает целочисленные списки в целочисленные списки; и те, и другие соответствуют одному и тому же кодированию, например одному и тому же отображению. Однако существуют варианты осуществления, в которых входной кольцевой элемент представлен как целочисленный список в альтернативном кодировании. В любом случае входной кольцевой элемент представлен в цифровой форме, обеспечивая возможность таблице отображать входной кольцевой элемент в выходной кольцевой элемент.

Таблица может перечислять входные кольцевые элементы в некотором формате вместе с ассоциированным выходным целочисленным списком. Таблица может также быть представлена в хранилище не включая входное кольцо и перечисляя только выходные целочисленные списки. Например, это может быть сделано, если входное кольцо представлено в каноническом формате.

Например, предполагая примерное кодирование, входной кольцевой элемент $k = u^{k_1} - u^{k_2}$ может быть отображен таблицей T в выходной целочисленный список. В этом случае входной кольцевой элемент может быть представлен в виде целочисленного списка так, что мы можем иметь $((k_1, k_2)) = (l_1, l_2)$. Последнее кодирует выходной кольцевой элемент $l = u^{l_1} - u^{l_2}$. Выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом. Например, если кольцевой элемент приращения равен 1, то $l = k + 1$. В варианте осуществления элемент приращения может быть равен 1, однако это не необходимо. Например, с использованием примерного кодирования элемент приращения может быть выбран как u^t для некоторого значения t , например любого значения $0 \leq t < \text{порядок}(u)$.

Таблица приращений гораздо меньше таблиц, описанных в уровне техники. Последние таблицы принимают два элемента входных данных, например два закодированных числа, для производства закодированных выходных данных. Однако таблица T получает только один закодированный элемент входных данных для производства одного закодированного элемента выходных данных; кольцевой элемент приращения фиксирован. Предполагая, что кодирования занимают схожее количество места, место для входа T уменьшается примерно до квадратного корня. Это существенное улучшение размера.

Вычислительное устройство 100 содержит блок 130 кольцевого сложения и блок 140 кольцевого умножения. Вычислительное устройство 100 может также содержать блок 120 кольцевого отрицания. В одном варианте осуществления блок 140 кольцевого умножения может использовать блок 130 сложения для выполнения сложения; блок 130 сложения может использовать блок 120 отрицания. Это было указано на фиг. 1 линиями между блоками 120, 130 и 140. Однако блоки могут дублироваться; например, блок 130 сложения может осуществлять собственное отрицание, и умножение 140 может осуществлять собственное сложение. Отрицание также называется "изменением знака".

Блок 120 отрицания может принимать входной для отрицания целочисленный список (a_1, a_2) , кодирующий входной для отрицания кольцевой элемент a . Блок 120 отрицания выполнен с возможностью определять выходной для отрицания целочисленный список (b_1, b_2) , кодирующий выходной для отрицания кольцевой элемент b . Выходной для отрицания кольцевой элемент является величиной противоположного знака для входного для отрицания кольцевого элемента, например выходной для отрицания кольцевой элемент равен нейтральному кольцевому элементу для сложения (0) минус входной для отрицания кольцевой элемент. Таким образом, $b = -a$.

В одном варианте осуществления блок отрицания может вычислять выходной целочисленный список путем перестановки входного для отрицания целочисленного списка. С использованием примерного кодирования $a = u^{a_1} - u^{a_2}$, выходной целочисленный список может быть (a_2, a_1) . Отрицание путем перестановки может эффективно осуществляться в коде путем изменения адреса, из которого элемент считывается, и не обязательно изменять фактический порядок в памяти.

В одном варианте осуществления блок отрицания может вычислять выходной целочисленный список путем прибавления константы к каждому целому из целочисленного списка. Например, в примерном кодировании с использованием целого m такого, что $u^m = -1$; например, выходной целочисленный список может быть $(a_1 + m, a_2 + m)$.

Блок 130 кольцевого сложения выполнен с возможностью принимать первый входной для сложения целочисленный список (a_1, a_2) , кодирующий первый входной для сложения кольцевой элемент, и второй входной для сложения целочисленный список (b_1, b_2) , кодирующий второй входной для сложения кольцевой элемент. Например, блок 130 кольцевого сложения может принимать два операнда из хранилища 150 операндов. Блок 130 кольцевого сложения выполнен с возможностью определять выходной для сложения целочисленный список, кодирующий выходной для сложения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для сложения целочисленных списков, причем выходной для сложения кольцевой элемент равен кольцевому сложению первого входного для сложения кольцевого элемента и второго входного для сложения кольцевого элемента.

В одном варианте осуществления отображение целочисленного списка в конкретный кольцевой элемент содержит множество подотображений, причем каждое подотображение ассоциировано с целым из целочисленного списка, причем подотображение отображает целое в кольцевой элемент. Отображение является линейной комбинацией, например суммой, подотображений, применяемых к ассоциированному целому. Подотображение может быть возведением элемента основания в степень, определенную ассоциированным целым. Например, в примерном кодировании (a_1, a_2) может считаться суммой подотображений u^{a_1} и $-u^{a_2}$.

Фиг.1b изображает вариант осуществления блока 130 сложения. Блок 130 сложения принимает первый входной для сложения целочисленный список 131 и второй входной для сложения целочисленный список 132. Блок 130 сложения содержит промежуточный блок 134 сложения, выполненный с возможностью итерационно прибавлять кольцевой элемент, полученный из целого из второго входного для сложения целочисленного списка 132, к первому входному для сложения кольцевому элементу. Например, промежуточный блок 134 сложения может прибавлять к промежуточной сумме 133, которая инициализируется для первого элемента целочисленного списка. Сложение включает в себя применение таблицы приращений из хранилища 110.

Блок 140 кольцевого умножения выполнен с возможностью принимать первый входной для умножения целочисленный список (r_1, r_2) , кодирующий первый входной для умножения кольцевой элемент, и второй входной для умножения целочисленный список (s_1, s_2) , кодирующий второй входной для умножения кольцевой элемент. Например, блок 140 умножения может принимать два операнда из хранилища 150 операндов. Блок 140 кольцевого умножения выполнен с возможностью определять выходной для умножения целочисленный список, кодирующий выходной для умножения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для умножения целочисленных списков, причем выходной для умножения кольцевой элемент равен кольцевому умножению первого входного для умножения кольцевого элемента и второго входного для умножения кольцевого элемента.

Фиг.1c изображает возможный вариант осуществления блока 140 умножения. Блок 140 умножения принимает первые входные для умножения целочисленные списки 141 и вторые входные для умножения целочисленные списки 142. Блок 140 умножения содержит промежуточный блок 144 умножения, выполненный с возможностью определения из первого и второго входных для умножения целочисленных списков 141, 142 первого промежуточного целочисленного списка 145 умножения (t_1, t_2) и второго промежуточного целочисленного списка 146 умножения (u_1, u_2) , кодирующих первый и второй промежуточный кольцевой элемент умножения соответственно. Блок 140 умножения выполнен с возможностью сложения первого 145 и второго промежуточного целочисленного списка 146 умножения посредством блока 130 кольцевого сложения. Определение промежуточного целочисленного списка может включать в себя арифметические операции над целыми в целочисленном списке, но не требует таблицы приращений.

Вычислительное устройство 100 опционально содержит блок 170 кольцевого кодирования для кодирования кольцевого элемента коммутативного кольца в виде целочисленного списка и блок 160 кольцевого декодирования для декодирования целочисленного списка (a, b) в кольцевой элемент (x) коммутативного кольца. Блок 170 кодирования и/или блок 160 декодирования может отсутствовать, например, когда

вычислительное устройство 100 принимает закодированные входные данные и/или обеспечивает отчет в закодированных выходных данных. Блок 170 кодирования и/или блок 160 декодирования может осуществляться как отдельный блок, например как устройство кодирования и/или устройство 160 декодирования.

5 Блок 170 кольцевого кодирования может содержать хранилище 172, выполненное с возможностью хранить таблицу кодирования, определенную для одного или нескольких кольцевых элементов основания (u), причем таблица кодирования отображает кольцевой элемент (x) в целочисленный список $((a,b))$ так, что кольцевой элемент равен линейной комбинации степеней одного или нескольких кольцевых

10 элементов основания ($x = u^a - u^b$), причем степени имеют показатели, определенные целочисленным списком. Блок 170 кодирования может сохранять закодированный кольцевой элемент в хранилище 150 операторов. Блок 170 кодирования обеспечивает возможность системе работать с простой информацией.

15 Блок 160 кольцевого декодирования выполнен с возможностью определять для одного или нескольких кольцевых элементов основания (u) кольцевой элемент (x) такой, что кольцевой элемент равен линейной комбинации степеней одного или нескольких кольцевых элементов основания ($x = u^a - u^b$), причем степени имеют показатели, определенные целочисленным списком. Например, блок 160 декодирования может

20 содержать хранилище, хранящее таблицу декодирования, отображающую целочисленные списки в кольцевые элементы. Например, блок 160 декодирования может содержать блок вычисления для вычисления степеней и их линейной комбинации.

Во множестве интересных вариантов осуществления опускается один или оба из блоков 160 и 170 кодирования и декодирования. Например, вычислительное устройство

25 100 может быть сконфигурировано для приема закодированной информации по компьютерной сети, например Интернет. Владелец системы, в которой устройство 100 вычисления с обфускацией работает, например компьютер, исполняющий программные средства вычисления с обфускацией, может не знать кодирование, используемое для входной информации, а также для информации, выводимой системой 100, например

30 передаваемой обратно по компьютерной сети. Соответственно, даже несмотря на то, что вычисления выполняются в облаке, хозяин информации имеет некоторую гарантию, что его информация защищена. Оперирование над информацией в закодированной форме обычно невозможно с использованием криптографии, например шифрования. Даже если табличная система используется, как очерчено в уровне техники, это требует

35 двойных таблиц.

Обычно вычислительное устройство 100 содержит микропроцессор (не показан), который исполняет надлежащие программные средства, сохраненные в устройстве 100; например, эти программные средства могли быть загружены и/или сохранены в соответствующую память, например энергозависимую память, такую как RAM, или

40 энергонезависимую память, такую как флэш-память (не показана). В качестве альтернативы, устройство 100 может, в целом или частично, осуществляться в программируемой логике, например в качестве программируемой пользователем вентильной матрицы (FPGA). Устройство 100 может осуществляться, в целом или частично, в качестве так называемой специализированной интегральной цепи (ASIC), т. е. интегральной цепи (IC), изготовленной специально для ее конкретного

45 использования.

В одном варианте осуществления электронное вычислительное устройство содержит цепь кольцевого сложения и цепь кольцевого умножения, выполненную с возможностью исполнения функции соответствующего блока. Вычислительное устройство может

также содержать цепь отрицания. Цепь может быть интегральными цепями, такими как CMOS, например полученными путем описания функций на языке описания аппаратных средств, таком как Verilog и VHDL. Цепи могут быть процессорной цепью и запоминающей цепью, причем процессорная цепь исполняет инструкции, представленные электронно в запоминающих цепях. Цепи могут также быть FPGA, ASIC или подобным.

Хранилище 110 таблиц и хранилище 150 операндов могут осуществляться в качестве электронного хранилища, например памяти. Оба хранилища могут входить в состав одной и той же памяти, но они могут быть отдельными средствами памяти. Хранилище 110 таблиц может быть энергонезависимой, непerezаписываемой памятью, например ROM, или памятью с однократной записью и многократным считыванием (WORM). Хранилище 150 операндов может быть энергозависимой или энергонезависимой перезаписываемой памятью, например флэш-памятью или RAM.

Фиг.2 схематически изображает пример варианта осуществления вычислительного устройства 101. Вычислительное устройство 101 является усовершенствованием вычислительного устройства 100. В одном варианте осуществления вычислительное устройство 101 содержит множество блоков кольцевого сложения, множество блоков кольцевого умножения и, опционально, множество блоков отрицания. Например, фиг.2 изображает три блока 1401.1, 140.2 и 140.3 умножения и два блока 130.1 и 130.2 сложения. Эти блоки могут иметь то же самое проектирование, что и блоки 140 и 130, соответственно. Блоки умножения и сложения занимают относительно мало места, например, при осуществлении в программных средствах эти блоки не требуют больше чем несколько сотен компьютерных инструкций нижнего уровня. В частности, копия блока сложения и/или умножения может быть использована для каждого умножения или сложения, которое требуется в компьютерной программе. Это обеспечивает возможность традиционных методик обфускации. В качестве примера, фиг.2 изображает, как многочлен $ax^2 + bx + c$ может быть вычислен с использованием арифметики с обфускацией.

Операции множества арифметических блоков, например, сложение, умножение, отрицание, могут быть упорядочены в любом порядке, который допускают их зависимости от данных. Например, операция 140.3 может быть внесена в порядок 140.1, 140.2., 130.1 и 130.2 в любом месте перед 130.1. Кроме того, порядок последующих умножений или сложений может быть обращен. Таким образом, схема, такая как схема 2, может быть преобразована в линейный порядок для компьютерной программы множеством способов. Не является необходимым, чтобы блоки были строго раздельными; инструкции для первого блока могут перемежаться с инструкциями для другого блока.

Фиг.3 схематически изображает пример варианта осуществления устройства 200 вычисления таблицы для вычисления таблицы приращений для использования в вычислительном устройстве. Таблица приращений может быть использована в устройстве, таком как вычислительное устройство 100. Таблица приращений может сохраняться на непереходном устройстве хранения, например жестком диске, энергонезависимом кристалле памяти и т.д.

Устройство 200 вычисления таблицы содержит блок 210 создания таблицы, выполненный с возможностью строить таблицу приращений. Например, блок создания таблицы может быть выполнен с возможностью

- многократно выбирать входной кольцевой элемент, например x ,
- определять выходной кольцевой элемент, который равен кольцевому элементу

приращения, кольцевым образом сложенному с входным кольцевым элементом.

Например, $y = x + 1$, если значение приращения равно 1.

- определять выходной целочисленный список, кодирующий выходной кольцевой элемент. Например, устройство 200 вычисления таблицы может содержать блок

5 кодирования, такой как блок 170 кодирования.

- добавлять запись в таблицу приращений, отображающую входной кольцевой элемент в выходной целочисленный список.

Эти этапы могут выполняться, пока не все кольцевые элементы были отображены в целочисленный список. В некоторых вариантах осуществления элементы могут

10 пропускаться для построения частичной таблицы приращений; например, может быть известно из контекста, что конкретные кольцевые элементы не будут возникать.

При заданном кольце R , потенциальном кольцевом элементе основания u , кодировании, например примерном кодировании, и длине целочисленного списка, например 2, таблица декодирования может генерироваться, как обеспечено ниже. Пусть

15 k имеет порядок u .

- генерировать все целочисленные списки, например путем генерирования всех целочисленных списков длины целочисленного списка и допуская для каждой позиции в списке все целые от 0 до k не включительно. Например, генерировать: (0,0), (0,1), (1,0), (1,1), (0,2), (1,2), (2,2) (2,0), (2,1), (0,3),... и т. д.,

20 - для каждого сгенерированного целочисленного списка вычислять кольцевой элемент, закодированный целочисленным списком, и добавлять запись в таблицу декодирования, ассоциирующую целочисленный список с декодированием.

Хотя декодирование может использовать или не использовать таблицу декодирования, такая таблица также полезна, поскольку таблица кодирования может генерироваться

25 из таблицы декодирования, например, путем сортировки таблицы для кольцевых элементов. Может случиться так, что кольцевой элемент имеет множество кодирований. Например, кольцевой элемент 0 (нейтральный элемент для сложения) может быть представлен как (a, a) в примерном кодировании для любого a . Такое множество кодирований может удаляться из таблицы, например путем удаления всех кроме одной

30 из множества записей для некоторого заданного кольцевого элемента; или путем оставления множества кодирований в таблице и использования таблицы кодирования для кодирования в случайную запись из множества записей.

Построение таблицы декодирования или кодирования может также быть использовано, чтобы выяснить, является ли кольцевой элемент u кольцевым элементом

35 основания. Если построение таблицы кодирования терпит неудачу, поскольку оказывается, что некоторые кольцевые элементы не имеют кодирования, то u не является кольцевым элементом основания.

Ниже представлено некоторое количество вариантов осуществления кодирований, таблиц приращения, способов кольцевого сложения и способов кольцевого умножения.

40 Блоки отрицания, сложения и умножения вычислительного устройства 100 могут быть сконфигурированы для любого из этих вариантов осуществления. Все примеры применяются к любому коммутативному кольцу, в частности $\mathbb{Z}_n$ и $\mathbb{Z}_n[x]/f(x)$. Здесь n является положительным целым. Кроме того, очень предпочтительно, чтобы любой элемент коммутативного кольца мог быть представлен в выбранном кодировании. Не

45 все коммутативные кольца обеспечивают возможность всем элементам быть представленными в некотором заданном кодировании, например, в виде некоторого заданного типа представления целочисленного списка. При заданном коммутативном кольце R мы будем считать, что оно обеспечивает возможность полной гомогенной

обфускации, если любой элемент в R может быть представлен в виде целочисленного списка с использованием некоторого заданного типа кодирования. Специалист в данной области техники может подтвердить, что если некоторое заданное коммутативное кольцо обеспечивает возможность полной гомогенной обфускации при заданном кодировании, например путем генерирования всех допустимых кодирований и подтверждения, что вместе они представляют все элементы некоторого заданного кольца. Для некоторых приложений может быть обеспечена возможность того, что кодирование имеет некоторые пропуски. Это может иметь следствием то, что арифметика не может выполняться над этими пропусками, по меньшей мере не с использованием кодирования целочисленного списка с обфускацией. Конкретные примеры коммутативных колец, обеспечивающих возможность конкретных типов кодирований, дополнительно представляются ниже.

Ниже сначала дается описание примерного кодирования. Существует множество типов кодирований, которые имеют сходство в том, что кольцевые элементы могут быть представлены в виде списков целых чисел. Эти целые не являются кольцевыми элементами, например, даже если кольцо не является целочисленным кольцом, например кольцом многочленов, то тем не менее элементы могут быть представлены в виде целочисленных списков. В используемом здесь кодировании то, как некоторый заданный целочисленный список отображается в кольцевой элемент, называется кодированием. Обычно целочисленные списки будут всегда одной и той же длины, однако это не необходимо. В общем случае, поскольку кодирование обеспечивает возможность большего количества типов целочисленных списков, например более длинные списки, становится более вероятно, что некоторый заданный кольцевой элемент может быть закодирован в виде целочисленного списка различными способами.

При заданном коммутативном кольце R с примерным кодированием существует особый кольцевой элемент u такой, что любой элемент a из R может быть записан как $u^{a_1} - u^{a_2}$ для некоторых целых a_1 и a_2 . Мы называем такой особый кольцевой элемент кольцевым элементом основания. Не все коммутативные кольца могут быть закодированы таким образом, но достаточно многие из них, чтобы кодирование было полезным. Целые a_1 и a_2 сами по себе не являются кольцевыми элементами кольца R ; они являются целыми, над которыми выполняются операции по модулю порядка элемента основания. Следует заметить, что кольцевой элемент a равен линейной комбинации степеней элемента основания u , а именно u^{a_1} и u^{a_2} ; в этом случае линейная комбинация получается путем умножения степеней на +1 или -1 и их суммирования, в частности, путем вычитания второй степени из первой степени. Вычислительное устройство оперирует над кольцевыми элементами, закодированными вышеупомянутым образом. Блоки сложения, отрицания и умножения могут оперировать над кольцевыми элементами в этом кодировании.

Таблица приращений T играет центральную роль в операции как сложения, так и умножения. Таблица приращений отображает входной кольцевой элемент, в этом случае входной кольцевой элемент может быть представлен в виде целочисленного списка. Например, при заданном входном целочисленном списке (k_1, k_2) , представляющем входной кольцевой элемент $k = u^{k_1} - u^{k_2}$, таблица T отображает его в выходной целочисленный список, например $T((k_1, k_2)) = (l_1, l_2)$, кодирующий выходной кольцевой элемент $l = u^{l_1} - u^{l_2}$. Выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом. В этом примере

элемент приращения может быть принят за 1, т. е. кольцевым элементом, который является единичным элементом для кольцевого умножения; в этом случае $l = k + 1$. Удобно то, что таблица может применяться непосредственно к кольцевым элементам, которые используют то же самое кодирование, и, таким образом, может применяться к кольцевым элементам, имеющим представление целочисленного списка. Тем не менее, существуют варианты осуществления, в которых таблица применяется к кольцевым элементам в альтернативном кодировании. Альтернативное кодирование может также быть целочисленным списком, но альтернативного типа. Кроме того, кольцевой элемент приращения не обязательно должен быть 1.

Ниже описаны операции отрицание, сложение и умножение.

Отрицание. При заданном входном для отрицания целочисленном списке (a_1, a_2) , представляющем входной для отрицания кольцевой элемент $a = u^{a_1} - u^{a_2}$, выходной для отрицания целочисленный список может быть получен путем перестановки целочисленного списка, в этом случае путем обращения порядка. Выходным для отрицания целочисленным списком может быть (a_2, a_1) . Предполагая, что существует m такое, что $u^m = -1$, что случается для многих колец R , отрицание может в качестве альтернативы быть получено путем прибавления константы, например m , к каждому целому из целочисленного списка. В последнем случае выходным для отрицания целочисленным списком может быть $(a_1 + m, a_2 + m)$. Это срабатывает, поскольку $-a = u^{a_2} - u^{a_1} = u^{a_1+m} - u^{a_2+m}$. Арифметика в целочисленном списке предпочтительно выполняется по модулю порядка элемента основания. Здесь целое из целочисленных списков соответствует показателю элемента основания, так что целые, которые являются одними и теми же по модулю порядка элемента основания, кодируют один и тот же кольцевой элемент.

Сложение. Для сложения принятого первого входного для сложения целочисленного списка (a_1, a_2) , кодирующего первый входной для сложения кольцевой элемент $a = u^{a_1} - u^{a_2}$, и второго входного для сложения целочисленного списка (b_1, b_2) , кодирующего второй входной для сложения кольцевой элемент $b = u^{b_1} - u^{b_2}$, первый промежуточный целочисленный список сложения $((c_1, c_2))$, кодирующий промежуточный кольцевой элемент сложения c , определяется.

Кольцевой элемент c может быть первым входным для сложения кольцевым элементом a плюс элемент основания u в степени, определенной из второго входного для сложения целочисленного списка, в частности первого целого из второго входного для сложения целочисленного списка. В этом примере мы можем иметь $c = u^{a_1} - u^{a_2} + u^{b_1}$.

Чтобы вычислить последнее, мы делаем наблюдение, что

$c = u^{a_1} - u^{a_2} + u^{b_1} = (u^{a_1-b_1} - u^{a_2-b_1} + 1)u^{b_1}$. Выражение в скобках может быть переписано в кодировании с использованием таблицы приращений. Посредством первого применения таблицы приращений к кольцевому элементу $u^{a_1-b_1} - u^{a_2-b_1}$ получается элемент $u^{d_1} - u^{d_2} = u^{a_1-b_1} - u^{a_2-b_1} + 1$. Например, посредством $T((a_1 - b_1, a_2 - b_1)) = (d_1, d_2)$. Тогда мы имеем, что $c_1 = d_1 + b_1$ и $c_2 = d_2 + b_1$, таким образом, определение промежуточного целочисленного списка сложения $((c_1, c_2))$ может дополнительно содержать сложение целого, определенного из вторых входных для сложения целочисленных списков, с целыми в целочисленном списке, возникшем в результате первого применения. Прибавление u^{b_1}

к кольцевому элементу в представлении целочисленного списка, в этом случае к a , иногда называется этапом положительного приведения.

Таким образом, блок сложения получил промежуточный кольцевой элемент сложения $c = u^{a_1} - u^{a_2} + u^{b_1} = u^{c_1} - u^{c_2}$ в виде целочисленного списка (c_1, c_2) . Промежуточный кольцевой элемент сложения, таким образом, является линейной комбинацией степеней одного или нескольких элементов основания, причем степени определяются из первого и второго входных для сложения целочисленных списков. В этом случае таблица приращений применяется к кольцевому элементу $u^{a_1-b_1} - u^{a_2-b_1}$, формируемому одним или несколькими кольцевыми элементами основания (u) , возведенными в степень первого целого из первого целочисленного списка (a_1) минус первое целое из второго целочисленного списка (b_1) , минус кольцевой элемент основания (u) , возведенный в степень второго целого из первого целочисленного списка (a_2) минус первое целое из второго целочисленного списка (b_1) .

В этом примере, выходной для сложения целочисленный список может быть определен посредством второго применения таблицы приращений к кольцевым элементам, определенным из промежуточного целочисленного списка сложения и второго входного для сложения целочисленного списка. Это может содержать вычисление суммы промежуточного кольцевого элемента сложения c минус элемент основания, возведенный в степень, определенную из второго входного для сложения целочисленного списка, например второго целого из второго входного для сложения целочисленного списка b_2 : $c - u^{b_2} = u^{c_1} - u^{c_2} - u^{b_2}$. Это может быть выполнено путем отрицания промежуточного кольцевого элемента сложения, представленного промежуточным целочисленным списком сложения, перед вторым применением таблицы приращений. Отрицание c может быть выполнено, как указано выше. В качестве примера мы используем перестановку, но та же самая операция может выполняться путем добавления константы к показателю. После отрицания сумма может использовать прибавление (вместо вычитания) элемента основания, возведенного в степень, определенную из второго входного для сложения целочисленного списка: $-c + u^{b_2} = u^{c_2} - u^{c_1} + u^{b_2}$.

Последняя операция имеет тот же тип, что и выше, и может выполняться посредством применения таблицы тем же самым образом, что и прибавление u^{b_1} . После этого для результата снова выполняется отрицание. Полное сложение может использовать два отрицания и два применения таблицы одной и той же таблицы приращений T .

Вычитание $-u^{b_2}$ из кольцевого элемента в представлении целочисленного списка, в этом случае из c , иногда называется этапом отрицательного приведения. Этап отрицательного приведения может выполняться путем отрицания, выполнения этапа положительного приведения и снова отрицания.

Умножение. Для умножения принятого первого входного для умножения целочисленного списка (r_1, r_2) , кодирующего первый входной для умножения кольцевой элемент $r = u^{r_1} - u^{r_2}$, и второго входного для умножения целочисленного списка $((s_1, s_2))$, кодирующего второй входной для умножения кольцевой элемент $s = u^{s_1} - u^{s_2}$, первый промежуточный целочисленный список умножения (t_1, t_2) и второй промежуточный целочисленный список умножения (u_1, u_2) определяются. Выходной для умножения целочисленный список, кодирующий выходной для умножения кольцевой элемент, определяется из первого и второго промежуточного элемента. В других вариантах

осуществления может быть более двух промежуточных целочисленных списков умножения. Мы имеем, что

$$\begin{aligned} r \cdot s &= (u^{r_1} - u^{r_2})(u^{s_1} - u^{s_2}) = u^{r_1+s_1} - u^{r_1+s_2} + u^{r_2+s_2} - u^{r_2+s_1} = (u^{r_1+s_1} - u^{r_1+s_2}) + (u^{r_2+s_2} - u^{r_2+s_1}) = \\ &= t + u. \end{aligned}$$

Разбиение выражения в разложенных произведениях по двум выражениям t и u может быть сделано различными способами, например как $(u^{r_1+s_1} - u^{r_2+s_1}) + (u^{r_2+s_2} - u^{r_1+s_2})$.

Таким образом, для умножения двух кольцевых элементов, представленных в виде целочисленных списков, они могут быть преобразованы в два новых целочисленных списка, которые могут быть сложены для получения ответа для умножения. Сложение может выполняться, как описано выше. Например, блок умножения может вычислять промежуточные целочисленные списки и посылать их блоку умножения.

Например, первое целое t_1 из первого промежуточного целочисленного списка умножения может содержать первое целое r_1 из первого входного для умножения целочисленного списка плюс первое целое s_1 из второго входного для умножения целочисленного списка, и второе целое t_2 из первого промежуточного целочисленного списка умножения может содержать первое целое r_1 из первого входного для умножения целочисленного списка плюс второе целое s_2 из второго входного для умножения целочисленного списка, $t_1 = r_1 + s_1, t_2 = r_1 + s_2$; первое целое u_1 из второго промежуточного целочисленного списка умножения может содержать второе целое r_2 из первого входного для умножения целочисленного списка плюс второе целое s_2 из второго входного для умножения целочисленного списка, и второе целое u_2 из второго промежуточного целочисленного списка умножения может содержать второе целое r_2 из первого входного для умножения целочисленного списка плюс первое целое s_1 из второго входного для умножения целочисленного списка, $u_1 = r_2 + s_2, u_2 = r_2 + s_1$.

В одном варианте осуществления, например в только что раскрытом примере, арифметика выполняется над целочисленными списками, кольцевые элементы не имеют необходимости вычисляться в качестве кольцевых элементов в некотором естественном представлении. Теперь будет рассмотрено некоторое количество вариантов. Многие из вариантов независимы, например вариант для кодирования может комбинироваться с вариантом для выполнения сложения.

Посредством арифметики с обфускацией, когда вычисления выполняются в целочисленном списке, соответствующем, например, $u^{a_1-b_1}$ и т. д., значение может быть приведено по модулю порядка u . Например, если порядок u равен 30, все вычисления могут выполняться по модулю 30.

Значение приращения. Значение приращения не обязательно должно быть 1.

Существует по меньшей мере два способа для использования другого значения приращения. Во-первых, уравнение $c = u^{a_1} - u^{a_2} + u^{b_1} = (u^{a_1-b_1} - u^{a_2-b_1} + 1)u^{b_1}$ может быть модифицировано в $c = u^{a_1} - u^{a_2} + u^{b_1} = (u^{a_1-b_1+t} - u^{a_2-b_1+t} + u^t)u^{b_1-t}$. Это означает, что может быть построена таблица приращений, которая прибавляет значение u^t . Эта таблица приращений применяется к тем же самым целочисленным спискам, за исключением того что целое t прибавляется. После первого применения таблицы приращений число $b_1 - t$ прибавляется вместо b_1 .

Другой способ для изменения значения приращения состоит в том, чтобы взять два

элемента g и p из R такие, что многократное прибавление g в кольце дает p . Например, существует целое h такое, что $h \cdot g = g + \dots + g = p$. Предположим, что существует таблица приращений T_p со значением приращения p , например $p = 1$ или $p = u^t$. Таблица приращений T_g может быть построена для g в качестве значения приращения. Таблица T_g может применяться h раз для получения того же самого эффекта, что и применение T_p непосредственно. Использование различных таблиц приращений с различными значениями приращения может даже комбинироваться в один вариант осуществления, например для увеличения обфускации. Последнее построение имеет преимущество в том, что множество значений приращения может комбинироваться без изменения последующего вычисления сложения.

Построение таблицы приращений может также быть подтверждено. Например, возвращаясь к уравнению для промежуточного кольцевого элемента сложения, но вместо разложения $c = u^{a_1} - u^{a_2} + u^{b_1} = (u^{a_1-b_1} - u^{a_2-b_1} + 1)u^{b_1}$ делается следующее наблюдение:

$c = u^{a_1} - u^{a_2} + u^{b_1} = (u^{a_1-a_2} + u^{b_1-a_2} - 1)u^{a_2}$. С использованием этой формулы таблица приращений может быть построена для значения приращения -1. Этот тип таблицы приращений применяется к кольцевому элементу $u^{a_1-a_2} + u^{b_1-a_2}$. Этот кольцевой элемент не имеет примерного кодирования. Кольцевой элемент может, тем не менее, быть представлен в виде целочисленного списка, например в виде $(a_1 - a_2, b_1 - a_2)$, так, что эта таблица приращений получает целочисленный список в качестве входных данных и производит целочисленный список в качестве выходных данных. Однако в отличие от предыдущего примера входной целочисленный список имеет кодирование, отличное от выходного кодирования. Кроме того, хотя очень предпочтительно, чтобы кодирование, используемое на входе в блок сложения, не имело пропусков, т. е. чтобы любой кольцевой элемент мог быть представлен в этом кодировании, нет необходимости, чтобы это альтернативное входное кодирование этой таблицы приращений не имело пропусков; все элементы, которые должны быть представлены в качестве входных данных таблицы, могут быть представлены путем построения.

После применения таблицы приращений к кольцевому элементу $u^{a_1-a_2} + u^{b_1-a_2}$, например представленному в виде целочисленного списка $(a_1 - a_2, b_1 - a_2)$, целое a_2 прибавляется к обоим элементам выходных данных таблицы приращений. Результатом является промежуточное значение c , определенное выше. Для выполнения применения второй таблицы то же самое построение может быть использовано, что и выше: отрицание, прибавление u^{b_2} с использованием этой альтернативной таблицы приращений, снова отрицание. С использованием построения, указанного выше, значение приращения может варьироваться от -1 до других значений.

Применение таблицы приращений к кольцевому элементу $u^{a_1-a_2} + u^{b_1-a_2}$ имеет существенное преимущество, выражение является симметричным, таким образом, $T(a_1 - a_2, b_1 - a_2) = T(b_1 - a_2, a_1 - a_2)$ с использованием выражения целочисленного списка в качестве входного значения. Это в свою очередь обеспечивает возможность хранить таблицу приращений в сжатой форме, около половины таблицы не обязательно сохранять. Например, можно сохранить только $T(x, y)$, если $x \leq y$. Небольшой потенциальный недостаток этого способа состоит в том, что промежуточный целочисленный список использует другое кодирование.

В качестве дополнительного варианта таблица приращений может также применяться

$$\text{к } u^{a_2 - a_1} - u^{b_1 - a_1}.$$

Принципы, изображенные для примерного кодирования, могут применяться к множеству альтернативных кодирований. Первое альтернативное кодирование предназначено для кодирования кольцевого элемента a в качестве целочисленного списка (a_1, a_2) с использованием кодирования $a = u^{a_1} + u^{a_2}$. Кольцо, которое имеет кольцевой элемент основания u такой, что любой кольцевой элемент может быть закодирован таким образом, считается обеспечивающим возможность положительной арифметики с обфускацией. Примерное кодирование будет называться отрицательной арифметикой с обфускацией. Может быть доказано математически, что для любого кольца, которое обеспечивает возможность положительной арифметики с обфускацией с кольцевым элементом основания u , существует целое m такое, что $u^m = -1$. Кроме того, кольцо, которое обеспечивает возможность отрицательной арифметики с обфускацией, обеспечивает возможность положительной арифметики с обфускацией тогда и только тогда, когда такое значение m существует. Любое кольцо, которое обеспечивает возможность положительной арифметики с обфускацией, также обеспечивает возможность отрицательной арифметики с обфускацией, хотя обратное неверно.

Положительная арифметика с обфускацией следует в основном тем же самым линиям, что и для отрицательной арифметики с обфускацией, обрисованной выше. Кратко, изменение знака целочисленного списка может быть выполнено путем добавления значения m ко всем целым в целочисленном списке. При заданных входных для сложения $a = u^{a_1} + u^{a_2}$ и $b = u^{b_1} + u^{b_2}$ сложение может выполняться путем вычисления промежуточного $c = u^{a_1} + u^{a_2} + u^{b_1}$, например через $(u^{a_1 - b_1} + u^{a_2 - b_1} + 1)u^{b_1}$. Таблица приращений применяется к $u^{a_1 - b_1} + u^{a_2 - b_1}$ со значением приращения 1. Положительное приведение может применяться дважды, для обоих из u^{b_1} и u^{b_2} , никакое отрицательное приведение не необходимо. Это упрощает сложение. Построение таблицы приращений может варьироваться, как указано выше, путем вынесения за скобку другой степени u . Значение приращения может варьироваться, как указано выше. Положительная арифметика с обфускацией имеет преимущество в том, что таблица приращений всегда симметрична и может сохраняться в сжатой форме. Недостаток положительной обфускации состоит в том, что меньше колец обеспечивает возможность этого типа кодирования.

Кодирования, данные до сих пор, могут быть опционально умножены на постоянный кольцевой элемент $w = u^v$ для некоторого v . Таким образом, целочисленный список (a_1, a_2) может представлять кольцевой элемент $a = w(u^{a_1} - u^{a_2})$. Этап отрицания неизменен. Этап положительного приведения принимает вид $c = wu^{a_1} - wu^{a_2} + wu^{b_1} = (wu^{a_1 - b_1} - wu^{a_2 - b_1} + w)u^{b_1}$. Таблица приращений может использовать в качестве значения приращения w и применяется к $wu^{a_1 - b_1} - wu^{a_2 - b_1}$, которое имеет тот же самый тип кодирования. Этап отрицательного приведения может быть найден из этапа положительного приведения, как указано выше. Умножение может перемножать $r = wu^{r_1} - wu^{r_2}$ и $s = wu^{s_1} - wu^{s_2}$, представленные в виде целочисленных списков (r_1, r_2) и целочисленных списков (s_1, s_2) , с использованием

$$r \cdot s = (wu^{r_1} - wu^{r_2})(wu^{s_1} - wu^{s_2}) = (wu^{r_1 + s_1 + v} - wu^{r_1 + s_2 + v}) + (wu^{r_2 + s_2 + v} - wu^{r_2 + s_1 + v}) = t + u.$$

Дополнительное альтернативное кодирование дается посредством $a = u^{a_1}(u^{a_2} - u^{-a_2})$

или умноженным на константу $1/2$, то есть $u^{a_1}(\frac{u^{a_2} - u^{-a_2}}{2})$. Можно доказать, что для кольца,

которое обеспечивает возможность отрицательной арифметики с обфускацией, с
кольцевым элементом основания u , который имеет нечетный порядок, что любой

кольцевой элемент x может быть записан в виде $u^{a_1}(\frac{u^{a_2} - u^{-a_2}}{2})$. Это изменяет кодирование,

например отображение из целочисленного списка в кольцевой элемент. Если кольцо
имеет отрицательную обфускацию, оно также обеспечивает возможность этого

представления, при условии что кольцевой элемент основания имеет нечетный порядок.

Этап сложения и умножения может быть выполнен с возможностью различных
кодирований, соответственно. Например, при заданном числе в закодированной форме
 $a = u^{a_1}(u^{a_2} - u^{-a_2})$ можно вычислить a'_1 и a'_2 в a_1 и a_2 такие, что $a = u^{a'_1} - u^{a'_2}$, например путем
вычисления $a_1 + a_2$ по модулю порядка u и $a_1 - a_2$ по модулю порядка u . С использованием
последних целых, сложение и умножения, как выше, могут быть использованы.

То, что мы сделали для получения гиперболического представления, может быть
обобщено до любого вида линейного преобразования, и новое представление
эквивалентно исходному, если преобразование может быть обращено.

Допустим, мы имеем представление $a = u^{a_1} - u^{a_2}$ и соответствие, записанное в матричной
форме:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} m & n \\ o & p \end{pmatrix} \begin{pmatrix} a_3 \\ a_4 \end{pmatrix}$$

Представление в a_3 и a_4 эквивалентно другому, если преобразование имеет
определитель $mp - no$, который является единицей в кольце $\mathbb{Z}_k$; k является порядком u в
кольце R . Это истинно тогда и только тогда, когда $\gcd(mp - no, k) = 1$. Гиперболическое
представление является примером (включающим в себя умножение на $1/2$) и требует,
чтобы k было нечетным, поскольку в таком случае определитель преобразования равен
2 (или -2).

Мы объясним способ посредством другого примера. Рассмотрим кольцо $\mathbb{Z}_{79}$ и возьмем
 $u = 8$. Этот элемент имеет порядок $k = 13$, и мы знаем, что все элементы в $\mathbb{Z}_{79}$ могут
быть записаны в виде разницы $u^{a_1} - u^{a_2}$ для некоторых показателей. Рассмотрим

преобразование $M = \begin{pmatrix} 10 & 8 \\ 6 & 4 \end{pmatrix}$. Определитель равен 5 по модулю 13, так что матрица имеет
обратную; которой является $M^{-1} = \begin{pmatrix} 6 & 1 \\ 4 & 2 \end{pmatrix}$.

Мы знаем, что для каждого x в $\mathbb{Z}_{79}$ мы можем найти α и β такие, что $x = 8^\alpha - 8^\beta$, но с
использованием этого преобразования мы немедленно выводим, что для всех x мы
можем найти значения γ и δ такие, что $x = 8^{10\gamma + 8\delta} - 8^{6\gamma + 4\delta}$.

Это показывает, что большой класс представлений эквивалентен. Линейные
преобразования могут быть обобщены до аффинных преобразований, если мы включаем
туда две аддитивные постоянные r, s такие, что

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} m & n \\ o & p \end{pmatrix} \begin{pmatrix} a_3 \\ a_4 \end{pmatrix} + \begin{pmatrix} r \\ s \end{pmatrix}$$

Это преобразование может быть обращено, если линейное преобразование M может
быть обращено.

Количество целых в целочисленном списке. В примере, рассматриваемом до сих пор, количество элементов в целочисленном списке всегда было равно двум. Это количество имеет преимущества, т. е. оно уменьшает количество этапов вычисления. С другой стороны, обеспечение возможности большего количества элементов в целочисленном списке расширяет количество колец, которые обеспечивают возможность обфускации. Пример ниже рассматривает три целых на каждый список, но большее количество возможно и работает аналогично.

Рассмотрим первый целочисленный список (a_1, a_2, a_3) и второй целочисленный список (b_1, b_2, b_3) , кодирующие элементы $u^{a_1} + u^{a_2} + u^{a_3}$ и $u^{b_1} + u^{b_2} + u^{b_3}$ соответственно. Отрицание может быть выполнено путем прибавления константы m к целым в списках. Прибавление может быть выполнено путем применений таблицы приращений для каждого целого во втором целочисленном списке, в этом случае три раза. Первый промежуточный целочисленный список сложения может быть вычислен из

$u^{a_1} + u^{a_2} + u^{a_3} + u^{b_3} = (u^{a_1-b_3} + u^{a_2-b_3} + u^{a_3-b_3} + 1) u^{b_3}$. В этом случае значение приращения равно 1, и таблица приращений применяется к $u^{a_1-b_3} + u^{a_2-b_3} + u^{a_3-b_3}$. Для умножения делается то же самое количество промежуточных целочисленных списков умножения, что и во втором целочисленном списке, например: $(a_1 + b_1, a_2 + b_1, a_3 + b_1)$,

$(a_1 + b_2, a_2 + b_2, a_3 + b_2)$, $(a_1 + b_3, a_2 + b_3, a_3 + b_3)$.

Множество различных кольцевых элементов основания. Рассмотрим два элемента основания u и v с показателями такими, что $u^s = v$ и $v^t = u$. Целочисленный список (a_1, a_2) кодирует кольцевой элемент $a = u^{a_1} - v^{a_2}$; аналогично для (b_1, b_2) . Отрицание получается путем отображения (a_1, a_2) в (sa_2, ta_1) . Этап положительного приведения $u^{a_1} - v^{a_2} + u^{b_1} = (u^{a_1-b_1} - v^{a_2-tb_1} + 1)u^{b_1}$. Значение приращения равно 1, и таблица применяется к целочисленному списку $(a_1 - b_1, a_2 - tb_1)$. Отрицательное приведение может быть приведено к положительному приведению с использованием отрицания. Умножение может быть приведено к сложению.

Ниже обеспечиваются примеры для колец, обеспечивающих возможность отрицательной и/или положительной обфускации.

Кольцо R может быть целочисленным кольцом $\mathbb{Z}_n$ для модуля n .

Например, n может быть равно 13 с кольцевым элементом основания $u = 4$. Этот элемент имеет порядок 6. Ниже все кольцевые элементы 0-6 кодируются в виде целочисленного списка с использованием примерного кодирования. Следует заметить, что здесь все элементы имеют множество кодирований. Для первого перечисленного кодирования был дан пример отображения, который демонстрирует, как для заданного целочисленного списка соответствующий кольцевой элемент может быть найден. Кольцевые элементы 7-12 могут быть найдены путем отрицания кольцевых элементов 1-6.

Кольцевой элемент	Целочисленный список	Пример отображения
0	(x, x) для любого $0 \leq x < 6$	$4^x \cdot 4^x$
1	$(1, 2), (5, 4)$	$4^1 \cdot 4^2$
2	$(0, 3), (2, 0), (3, 5)$	$4^0 \cdot 4^3$
3	$(1, 0), (3, 4)$	$4^1 \cdot 4^0$
4	$(0, 5), (2, 3)$	$4^0 \cdot 4^5$

5	(0,4), (1,3), (4,1)	$4^0 \cdot 4^4$
6	(2,5), (4,2), (5,1)	$4^2 \cdot 4^5$

Этот пример также обеспечивает возможность положительной обфускации, поскольку $4^3 = -1$ в этом кольце.

Другие значения для n и u , которые обеспечивают возможность отрицательной обфускации: $n=151, u=2$; $n=87, u=20$; $n=79, u=8$ и т. д.

Изобретатели обнаружили большое количество примеров колец, которые обеспечивают возможность отрицательных и/или положительных кодирований. Следует заметить, что многие варианты выводимы из некоторого заданного отрицательного и/или положительного кодирования, как описано здесь.

Кольцо R может быть кольцом многочленов $\mathbb{Z}_n[x]/f(x)$ для многочлена f и модуля n . Многочлен не обязательно должен быть неприводимым. Если f неприводим, мы получаем коммутативное кольцо, которое не является полем. Оказывается, что любое коммутативное кольцо многочленов R обеспечивает возможность обфускации.

Например, дано некоторое количество полей

Поле $F(2^6)$

Это поле изоморфно $F_2[x]/(x^6+x^4+x^3+x+1)$. Основание $u=x^3$ имеет порядок 21.

Поле $F(2^8)$

Это поле изоморфно $F_2[x]/(x^8+x^4+x^3+x^2+1)$.

Основание $u=x^3$ имеет порядок 85.

Основание $u=x+1$ имеет порядок 51.

Поле $F(2^{10})$

Это поле изоморфно $F_2[x]/(x^{10}+x^6+x^5+x^3+x^2+x+1)$.

Основание $u=x^3$ имеет порядок 341.

Основание $u=x^7+x^6+x^4+x^3+x^2+x$ имеет порядок 93.

Поле $F(2^{12})$

Это поле изоморфно $F_2[x]/(x^{12}+x^7+x^6+x^5+x^3+x+1)$.

Основание $u=x^3$ имеет порядок 1365.

Основание $u=x^5$ имеет порядок 819.

Основание $u=x^7$ имеет порядок 585.

Основание $u=x^9$ имеет порядок 455.

Основание $u=x^8+x^7+x^6+x^4+x^2+x$ имеет порядок 315.

Основание $u=x^{10}+x^9+x^8+x^6+x^4+x^3$ имеет порядок 273.

Основание $u=x^{11}+x^{10}+x^7+x^5+x^3+x^2+x+1$ имеет порядок 195.

Фиг.4 схематически изображает пример варианта осуществления способа 300 вычисления для выполнения арифметики с обфускацией в коммутативном кольце (например $\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$), причем кольцо имеет конечное количество кольцевых элементов, кольцевое сложение и кольцевое умножение определены над кольцевыми элементами, способ вычисления оперирует над целочисленными списками $((a_1, a_2))$, кодирующими кольцевые элементы $(u^{a_1} - u^{a_2})$, целочисленные списки содержат по меньшей мере два целых. Способ вычисления содержит этапы, на которых

- сохраняют таблицу приращений (T) , определенную для приращения кольцевого элемента $(1; u^t)$, причем таблица приращений отображает входной кольцевой элемент $(k = u^{k_1} - u^{k_2})$ в выходной целочисленный список $(T((k_1, k_2)) = (l_1, l_2))$, кодирующий выходной кольцевой элемент $(l = u^{l_1} - u^{l_2})$, так, что выходной кольцевой элемент равен кольцевому

элементу приращения, кольцевым образом сложенному с входным кольцевым элементом ($l = k + 1$),

- осуществляют кольцевое сложение, причем кольцевое сложение содержит этапы, на которых

- принимают 310 первый входной для сложения целочисленный список $((a_1, a_2))$, кодирующий первый входной для сложения кольцевой элемент, и второй входной для сложения целочисленный список $((b_1, b_2))$, кодирующий второй входной для сложения кольцевой элемент,

- определяют 320 выходной для сложения целочисленный список, кодирующий выходной для сложения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для сложения целочисленных списков, причем выходной для сложения кольцевой элемент равен кольцевому сложению первого входного для сложения кольцевого элемента и второго входного для сложения кольцевого элемента,

- осуществляют кольцевое умножение, причем кольцевое умножение содержит этапы, на которых

- принимают 330 первый входной для умножения целочисленный список $((r_1, r_2))$, кодирующий первый входной для умножения кольцевой элемент, и второй входной для умножения целочисленный список $((s_1, s_2))$, кодирующий второй входной для умножения кольцевой элемент,

- определяют 340 выходной для умножения целочисленный список, кодирующий выходной для умножения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для умножения целочисленных списков, причем выходной для умножения кольцевой элемент равен кольцевому умножению первого входного для умножения кольцевого элемента и второго входного для умножения кольцевого элемента.

Фиг.5 схематически изображает пример варианта осуществления способа 400 сложения, который может быть использован в устройстве 100 или в способе 300 и т. д. Этот пример использует примерное кодирование. Способ может быть выполнен с возможностью других кодирований. Все варианты, описанные здесь, могут применяться; этот пример использует значение приращения 1, и таблица приращений строится путем вынесения за скобку u^{b_1} .

Способ 400 содержит прием операндов сложения 410. Это может содержать прием 410 первого входного для сложения целочисленного списка, например (a_1, a_2) , и прием 420 второго входного для сложения целочисленного списка, например (b_1, b_2) .

Способ 400 дополнительно содержит определение 420 промежуточного целочисленного списка сложения, например (c_1, c_2) . Например, это может содержать применение таблицы приращений к кольцевому элементу, определенному из первого и второго входных для сложения целочисленных списков. В частности, таблица приращений может применяться к целочисленному списку, причем элементы в целочисленном списке находятся из элементов во входных целочисленных списках.

Например, определение 420 может содержать применение 422 таблицы приращений к $(a_1 - b_1, a_2 - b_1)$, получая, например, (d_1, d_2) ; и прибавление 424 целого b_1 , определенного из вторых входных для сложения целочисленных списков, к целым в целочисленном списке, получающемся в результате первого применения, например $(c_1, c_2) = (d_1 + b_1, d_2 + b_1)$.

Способ 400 дополнительно содержит этап, на котором определяют 430 выходной

для сложения целочисленный список посредством второго применения таблицы приращений к кольцевому элементу, определенному из промежуточного целочисленного списка сложения и второго входного для сложения целочисленного списка. Для более длинных целочисленных списков это может включать в себя дополнительные применения таблицы приращений. Например, это может содержать отрицание 431 промежуточного целочисленного списка сложения, например перестановку в (c_2, c_1) . Применение 432 таблицы приращений и сложение 434 являются теми же самыми, что и применение 422 и сложение 424, за исключением того, что входные для сложения целочисленные списки (a_1, a_2) заменяются на промежуточный целочисленный список (c_2, c_1) , а b_1 - на b_2 . Наконец, для результата 434 выполняется отрицание 453 для получения результата сложения с обфускацией.

Если вместо отрицательной обфускации, как здесь, используется положительная обфускация, то отрицание 431, 435 может опускаться.

Фиг.6 схематически изображает пример варианта осуществления способа 500 умножения, который может быть использован в устройстве 100 или в способе 300 и т. д. Этот пример использует те же самые кодирования и таблицы приращений, что и способ 400.

Способ 500 содержит прием 510 операндов умножения. Это может содержать прием 510 первого входного для умножения целочисленного списка, например (r_1, r_2) , и прием 514 второго входного для умножения целочисленного списка (s_1, s_2) .

Способ 500 дополнительно содержит определение 520 первого и второго промежуточного целочисленного списка умножения. Например, 520 может содержать определение 522 первого промежуточного целочисленного списка умножения и определение 524 второго промежуточного целочисленного списка умножения. Они могут, например, быть выбраны как $(s_1 + r_1, s_1 + r_2)$ и $(s_2 + r_2, s_2 + r_1)$, соответственно, хотя существуют и другие варианты выбора. Умножение продолжается путем сложения этих чисел в способе 400 сложения.

Следует заметить, что таблица используется только в применении 422 и применении 432 и больше нигде в способах 400 и 500. И сложение, и умножение используют одну и ту же таблицу, причем и то, и другое использует таблицу одно и то же количество раз (2). Другие операции содержат маленькие арифметические операции над целыми в целочисленном списке, например по модулю порядка кольцевого элемента основания.

Множество различных вариантов исполнения способов возможно, как будет очевидно специалисту в данной области техники. Например, порядок этапов может варьироваться или некоторые этапы могут выполняться параллельно. Кроме того, между этапами могут быть вставлены другие этапы способа. Вставленные этапы могут представлять усовершенствования способа, такие как описанные здесь, или могут не относиться к способу. Кроме того, некоторый заданный этап может быть не полностью завершен перед тем, как начинается следующий этап.

Способ согласно одному варианту осуществления может выполняться с использованием программных средств, которые содержат инструкции для побуждения процессорной системы выполнять любой из способов 300, 400 и 500. Программные средства могут включать в себя только этапы, взятые на себя конкретным подобъектом системы. Программные средства могут сохраняться в подходящем носителе данных, таком как жесткий диск, гибкий диск, память и т. д. Программные средства могут быть посланы в виде сигнала по проводу, или беспроводным образом, или с использованием сети данных, например Интернет. Программные средства могут делаться доступными

для скачивания и/или для удаленного использования на сервере. Способ может исполняться с использованием битового потока, выполненного с возможностью конфигурировать программируемую логику, например программируемую пользователем вентильную матрицу (FPGA), для выполнения способа.

5 Следует понимать, что вариант осуществления также распространяется на компьютерные программы, в частности компьютерные программы на или в носителе, выполненном с возможностью применять вариант осуществления на практике. Программа может иметь форму исходного кода, объектного кода, промежуточного источника кода и объектного кода, например в частично скомпилированной форме, или
10 в любой другой форме, подходящей для использования в осуществлении способа согласно одному варианту осуществления. Вариант осуществления, относящийся к компьютерному программному продукту, содержит машиноисполняемые инструкции, соответствующие каждому из этапов обработки по меньшей мере одного из изложенных способов. Эти инструкции могут подразделяться на подпрограммы и/или сохраняться
15 в одном или нескольких файлах, на которые может быть сделана ссылка, статически или динамически. Другой вариант осуществления, относящийся к компьютерному программному продукту, содержит машиноисполняемые инструкции, соответствующие каждому из средств по меньшей мере одной из изложенных систем и/или продуктов.

Фиг.7а изображает машиночитаемый носитель 1000, имеющий перезаписываемую
20 часть 1010, содержащую компьютерную программу 1020, причем компьютерная программа 1020 содержит инструкции, чтобы побуждать процессорную систему выполнять способ вычисления для выполнения арифметики с обфускацией согласно одному варианту осуществления. Перезаписываемая часть может быть выполнена с возможностью множественной перезаписи или только единственной перезаписи.
25 Компьютерная программа 1020 может осуществляться на машиночитаемом носителе 1000 в виде физических меток или посредством намагничивания машиночитаемого носителя 1000. Однако любой другой подходящий вариант осуществления также мыслим. Кроме того, следует понимать, что, хотя машиночитаемый носитель 1000 показан здесь как оптический диск, машиночитаемый носитель 1000 может быть любым подходящим
30 машиночитаемым носителем, таким как жесткий диск, твердотельная память, флэш-память и т. д., и может быть неперезаписываемым или перезаписываемым. Компьютерная программа 1020 содержит инструкции, чтобы побуждать процессорную систему выполнять упомянутый способ вычисления для выполнения арифметики с обфускацией.

35 Машиночитаемый носитель, например машиночитаемый носитель 1000, может содержать таблицу приращений, и/или таблицу декодирования, и/или таблицу кодирования.

Фиг.7b изображает схематическое представление процессорной системы 1100 согласно одному варианту осуществления. Процессорная система содержит одну или несколько
40 интегральных цепей 1110. Архитектура одной или нескольких интегральных цепей 1110 схематически изображена на фиг.7b. Цепь 1110 содержит обрабатывающий блок 1120, например CPU, для запуска компьютерных программных компонентов, чтобы исполнять способ согласно одному варианту осуществления и/или осуществлять его модули или блоки. Цепь 1110 содержит память 1122 для хранения программного кода, данных и
45 т. д. Часть памяти 1122 может быть доступной только для чтения. Цепь 1110 может содержать элемент 1126 связи, например антенну, средства подключения или то и другое, и т. п. Цепь 1110 может содержать специализированную интегральную цепь 1124 для выполнения части или всей обработки, определенной в способе. Процессор

1120, память 1122, специализированная IC 1124 и элемент 1126 связи могут быть соединены друг с другом через средство 1130 взаимного соединения, например шину. Процессорная система 1110 может быть выполнена с возможностью контактной и/или бесконтактной связи с использованием антенны и/или средств подключения,

5 соответственно.

Следует заметить, что вышеупомянутые варианты осуществления иллюстрируют, а не ограничивают изобретение, и что специалисты в данной области техники смогут спроектировать множество альтернативных вариантов осуществления.

В формуле изобретения любые позиционные обозначения, помещенные в скобках, не должны толковаться как ограничивающие пункт формулы. Использование глагола "содержат" и его спряжений не исключает наличия элементов или этапов помимо упомянутых в пункте формулы. Элементы, упомянутые в единственном числе, не исключают наличие множества таких элементов. Изобретение может осуществляться посредством аппаратных средств, содержащих несколько отдельных элементов, и посредством подходящим образом запрограммированного компьютера. В пункте формулы об устройстве, где перечисляется несколько средств, несколько из этих средств может осуществляться одним и тем же элементом аппаратных средств. Сам факт того, что некоторые конкретные меры перечислены во взаимно различных зависимых пунктах формулы изобретения, не указывает, что комбинация этих мер не может быть

10

15

20

использована выгодным образом.

В формуле изобретения ссылки в скобках ссылаются на позиционные обозначения на чертежах вариантов осуществления или на формулы вариантов осуществления, таким образом повышая понятность пункта формулы. Эти ссылки не являются исчерпывающими и не должны толковаться как ограничивающие пункт формулы.

25 СПИСОК ССЫЛОЧНЫХ ПОЗИЦИЙ НА ФИГ.1:

100	вычислительное устройство
110	хранилище, выполненное с возможностью хранить таблицу приращений
120	блок кольцевого отрицания
130	блок кольцевого сложения
140	блок кольцевого умножения
150	хранилище операндов
160	блок декодирования
170	блок кодирования
172	хранилище, выполненное с возможностью хранить таблицу кодирования

35

(57) Формула изобретения

1. Электронное вычислительное устройство (100) для выполнения арифметики с обфускацией в коммутативном кольце $(\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x))$, причем кольцо имеет конечное количество кольцевых элементов, кольцевое сложение и кольцевое умножение

40 определены над кольцевыми элементами, вычислительное устройство оперирует на целочисленных списках $((a_1, a_2))$, кодирующих кольцевые элементы $(u^{a_1} - u^{a_2})$, целочисленные списки содержат по меньшей мере два целых числа, целочисленный список $((a_1, a_2))$ кодирует кольцевой элемент (a) так, что кольцевой элемент равен

45 линейной комбинации степеней $(a = u^{a_1} - u^{a_2}; a = u^{a_1} - v^{a_2})$ одного или нескольких кольцевых элементов основания $(u; u, v)$, степени имеют показатели, определенные целочисленным списком, причем вычислительное устройство содержит

- хранилище (110), выполненное с возможностью хранить таблицу приращений (T),

определенную для фиксированного приращения кольцевого элемента $(1; u^t)$,

- таблицу приращений, отображающую входной кольцевой элемент $(k = u^{k_1} - u^{k_2})$ в выходной целочисленный список $(T((k_1, k_2)) = (l_1, l_2))$, кодирующий выходной кольцевой элемент $(l = u^{l_1} - u^{l_2})$ так, что выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом ($l = k + 1$),

- блок (130) кольцевого сложения, выполненный с возможностью

- принимать первый входной для сложения целочисленный список $((a_1, a_2))$, кодирующий первый входной для сложения кольцевой элемент, и второй входной для сложения целочисленный список $((b_1, b_2))$, кодирующий второй входной для сложения кольцевой элемент, причем кольцевой элемент приращения не зависит от первого и второго входного для сложения кольцевого элемента,

- определять выходной для сложения целочисленный список, кодирующий выходной для сложения кольцевой элемент путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для сложения целочисленных списков, причем входной для сложения кольцевой элемент равен кольцевому сложению первого входного для сложения кольцевого элемента и второго входного для сложения кольцевого элемента, причем определение выходного для сложения целочисленного списка содержит этапы, на которых

- определяют промежуточный целочисленный список сложения $((c_1, c_2))$, кодирующий промежуточный кольцевой элемент сложения, путем первого применения таблицы приращений к кольцевому элементу $(u^{a_1 - b_1} - u^{a_2 - b_1})$, который является линейной комбинацией степеней одного или нескольких элементов основания, причем степени определяются из первого и второго входных для сложения целочисленных списков, ($u^{c_1} - u^{c_2} = u^{a_1} - u^{a_2} + u^{b_1}$),

- определяют выходной для сложения целочисленный список, содержащий второе применение таблицы приращений к кольцевым элементам, определенным из промежуточного целочисленного списка сложения и определенным из второго входного для сложения целочисленного списка.

2. Вычислительное устройство по п.1, содержащее

- блок (140) кольцевого умножения, выполненный с возможностью

- принимать первый входной для умножения целочисленный список $((r_1, r_2))$, кодирующий первый входной для умножения кольцевой элемент, и второй входной для умножения целочисленный список $((s_1, s_2))$, кодирующий второй входной для умножения кольцевой элемент,

- определять выходной для умножения целочисленный список, кодирующий выходной для умножения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для умножения целочисленных списков, причем выходной для умножения кольцевой элемент равен кольцевому умножению первого входного для умножения кольцевого элемента и второго входного для умножения кольцевого элемента.

3. Вычислительное устройство по п.1 или 2, в котором целочисленный список $((a_1, a_2))$ кодирует кольцевой элемент (a) так, что

- кольцевой элемент равен элементу основания, возведенному в степень, определенную первым целым из целочисленного списка, минус элемент основания, возведенный в

степень, определенную вторым целым из целочисленного списка ($a = u^{a_1} - u^{a_2}$),

опционально умноженному на константу ($a = t(u^{a_1} - u^{a_2})$), или

- кольцевой элемент равен элементу основания, возведенному в степень, определенную первым целым из целочисленного списка, плюс элемент основания, возведенный в степень, определенную вторым целым из целочисленного списка ($a = u^{a_1} + u^{a_2}$),

опционально умноженному на константу, или

- кольцевой элемент равен элементу основания, возведенному в степень, определенную первым целым из целочисленного списка, умноженному на результат элемента основания, возведенного в степень, определенную вторым целым из целочисленного списка, минус элемент основания, возведенный в степень, определенную

противоположным числом второго целого из целочисленного списка ($a = u^{a_1}(u^{a_2} - u^{-a_2})$),

опционально умноженному на константу, ($a = u^{a_1}(\frac{u^{a_2} - u^{-a_2}}{2})$), или

- кольцевой элемент равен элементу основания, возведенному в степень, которая является первой линейной комбинацией первого целого и второго целого из целочисленного списка, плюс или минус элемент основания, возведенный в степень, которая является второй линейной комбинацией первого целого и второго целого из целочисленного списка, ($x = u^{b_1} + u^{b_2}$ или $x = u^{b_1} - u^{b_2}$ при заданной матрице M такой, что

$M \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$), опционально умноженному на константу.

$$(c_1, c_2) \quad u^{a_1 - b_1} - u^{a_2 - b_1} \quad u^{c_1} - u^{c_2} = u^{a_1} - u^{a_2} + u^{b_1}$$

4. Вычислительное устройство по п.1, в котором

- определение промежуточного целочисленного списка сложения ((c_1, c_2))

дополнительно содержит этап, на котором прибавляют целое, определенное из первого и второго входных для сложения целочисленных списков, к целым в целочисленном списке, получаемом из первого применения.

5. Вычислительное устройство по п. 1, в котором

- таблица приращений применяется к кольцевому элементу ($u^{a_1 - b_1} - u^{a_2 - b_1}; u^{a_1 - b_1} + u^{a_2 - b_1}$), формируемому одним или несколькими кольцевыми элементами

основания (u), возведенными в степень первого целого из первого целочисленного списка (a_1) минус первое целое из второго целочисленного списка (b_1), плюс или минус кольцевой элемент основания (u), возведенный в степень второго целого из первого целочисленного списка (a_2) минус первое целое из второго целочисленного списка (b_1); или

- таблица приращений применяется к кольцевому элементу ($u^{a_1 - a_2} + u^{b_1 - a_2}; u^{a_1 - a_2} + u^{b_1 - a_2}$), формируемому одним или несколькими кольцевыми элементами

основания (u), возведенными в степень первого целого из первого целочисленного списка (a_1) минус второе целое из первого целочисленного списка (a_2), плюс или минус кольцевой элемент основания (u), возведенный в степень первого целого из второго целочисленного списка (b_1) минус второе целое из первого целочисленного списка (a_2); или

- таблица приращений применяется к кольцевому элементу ($u^{b_1 - a_1} - u^{a_2 - a_1};$
 $u^{a_2 - a_1} + u^{b_1 - a_1}$), формируемому одним или несколькими кольцевыми элементами

основания (u), возведенными в степень второго целого из первого целочисленного списка (a_2) минус первое целое из первого целочисленного списка (a_1) плюс или минус
 5 кольцевой элемент основания (u), возведенный в степень первого целого из второго целочисленного списка (b_1) минус первое целое из первого целочисленного списка (a_1).

6. Вычислительное устройство по п. 1, в котором

10 - для промежуточного кольцевого элемента сложения, представленного промежуточным целочисленным списком сложения, выполняется отрицание перед вторым применением таблицы приращений.

7. Вычислительное устройство по п. 1, в котором

15 - для кольцевого элемента, представленного целочисленным списком, выполняется отрицание путем перестановки целочисленного списка, и/или

- для кольцевого элемента, представленного целочисленным списком, выполняется отрицание путем прибавления константы к каждому целому из целочисленного списка, и/или

20 - для кольцевого элемента, представленного целочисленным списком ($((a_1, a_2))$), выполняется отрицание путем перестановки целочисленного списка и умножения одного или нескольких целых из целочисленного списка на константу ($((sa_2, ta_1))$).

8. Вычислительное устройство по п. 1, в котором таблица приращений получает в качестве входных данных входной целочисленный список ($((k_1, k_2))$), представляющий
 25 входной кольцевой элемент ($k = u^{k_1} - u^{k_2}$).

9. Вычислительное устройство по п.2, в котором определение выходного для умножения целочисленного списка содержит этапы, на которых

30 - определяют из первого и второго входных для умножения целочисленных списков первый промежуточный целочисленный список умножения ($((t_1, t_2))$) и второй

промежуточный целочисленный список умножения ($((u_1, u_2))$), кодирующие первый и второй промежуточный кольцевой элемент умножения соответственно,

- складывают первый и второй промежуточный целочисленный список умножения посредством блока кольцевого сложения.

10. Вычислительное устройство по п.9, в котором

35 - первое целое (t_1) из первого промежуточного целочисленного списка умножения содержит

первое целое (r_1) из первого входного для умножения целочисленного списка плюс первое целое (s_1) из второго входного для умножения целочисленного списка, и

40 - второе целое (t_2) из первого промежуточного целочисленного списка умножения содержит

первое целое (r_1) из первого входного для умножения целочисленного списка плюс второе целое (s_2) из второго входного для умножения целочисленного списка (

$t_1 = r_1 + s_1, t_2 = r_1 + s_2$), и

45 - первое целое (u_1) из второго промежуточного целочисленного списка умножения содержит

второе целое (r_2) из первого входного для умножения целочисленного списка плюс

второе целое (s_2) из второго входного для умножения целочисленного списка, и
 - второе целое (u_2) из второго промежуточного целочисленного списка умножения
 содержит

второе целое (r_2) из первого входного для умножения целочисленного списка плюс
 первое целое (s_1) из второго входного для умножения целочисленного списка ($u_1 = r_2 + s_2, u_2 = r_2 + s_1$).

11. Вычислительное устройство по п. 1, в котором

- коммутативное кольцо является кольцом, формируемым целыми по модулю
 целочисленного модуля ($\mathbb{Z}_n$), или

- коммутативное кольцо является кольцом, формируемым целочисленными
 многочленами по модулю целочисленного многочленного модуля ($\mathbb{Z}_n[x]/f(x)$).

12. Устройство (170) кольцевого кодирования $\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$, причем устройство
 кольцевого кодирования содержит

- хранилище (172), выполненное с возможностью хранить таблицу кодирования,
 определенную для одного или нескольких кольцевых элементов основания (u), причем
 таблица кодирования отображает кольцевой элемент (x) в целочисленный список $((a, b))$
 так, что кольцевой элемент равен линейной комбинации степеней одного или нескольких
 кольцевых элементов основания ($x = u^a - u^b$), причем степени имеют показатели,
 определенные целочисленным списком, причем устройство (170) кольцевого кодирования
 сконфигурировано с возможностью

- кодировать кольцевой элемент коммутативного кольца ($\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$) в качестве
 целочисленного списка для использования с вычислительным устройством по п.1,
 сконфигурированным с возможностью принимать закодированную информацию по
 компьютерной сети.

13. Устройство (160) кольцевого декодирования $(a, b)x \mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$, причем устройство
 кольцевого декодирования выполнено с возможностью

- декодировать целочисленный список $((a, b))$ в кольцевой элемент (x) коммутативного
 кольца ($\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$) для использования с вычислительным устройством по п.1,
 сконфигурированным с возможностью принимать закодированную информацию по
 компьютерной сети, причем декодирование содержит этап, на котором

- определяют для одного или нескольких кольцевых элементов основания (u)

кольцевой элемент (x) так, что кольцевой элемент равен линейной комбинации степеней
 одного или нескольких кольцевых элементов основания ($x = u^a - u^b$), причем степени
 имеют показатели, определенные целочисленным списком.

14. Устройство (200) вычисления таблицы для вычисления таблицы приращений для
 использования в вычислительном устройстве для выполнения арифметики с обфускацией
 в коммутативном кольце ($\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x)$), причем кольцо имеет конечное количество
 кольцевых элементов, кольцевое сложение и кольцевое умножение определены над
 кольцевыми элементами, вычислительное устройство оперирует над целочисленными
 списками $((a_1, a_2))$, кодирующими кольцевые элементы ($u^{a_1} - u^{a_2}$), целочисленные списки

содержат по меньшей мере два целых, причем устройство вычисления таблицы содержит

- блок (210) создания таблицы, выполненный с возможностью строить таблицу
 приращений, причем блок создания таблицы выполнен с возможностью

- многократно выбирать входной кольцевой элемент,

- определять выходной кольцевой элемент, который равен фиксированному кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом,

- определять выходной целочисленный список, кодирующий выходной кольцевой элемент,

- добавлять в таблицу приращений запись, отображающую входной кольцевой элемент в выходной целочисленный список, причем устройство (200) вычисления таблицы выполнено с возможностью хранить построенную таблицу приращений в вычислительном устройстве.

15 15. Способ электронного вычисления для выполнения арифметики с обфускацией в коммутативном кольце $(\mathbb{Z}_n; \mathbb{Z}_n[x]/f(x))$, причем кольцо имеет конечное количество кольцевых элементов, кольцевое сложение и кольцевое умножение определены над кольцевыми элементами, способ вычисления оперирует над целочисленными списками $((a_1, a_2))$, кодирующими кольцевые элементы $(u^{a_1} - u^{a_2})$, целочисленные списки содержат по меньшей мере два целых, целочисленный список $((a_1, a_2))$ кодирует кольцевой элемент (a) так, что кольцевой элемент равен линейной комбинации степеней $(a = u^{a_1} - u^{a_2};$

$a = u^{a_1} - v^{a_2})$ одного или нескольких кольцевых элементов основания $(u; u, v)$, причем степени имеют показатели, определенные целочисленным списком, причем способ вычисления содержит этапы, на которых

- сохраняют таблицу приращений (T) , определенную для фиксированного приращения кольцевого элемента $(1; u^t)$, причем таблица приращений отображает входной кольцевой элемент $(k = u^{k_1} - u^{k_2})$ в выходной целочисленный список $(T((k_1, k_2)) = (l_1, l_2))$, кодирующий выходной кольцевой элемент $(l = u^{l_1} - u^{l_2})$, так, что выходной кольцевой элемент равен кольцевому элементу приращения, кольцевым образом сложенному с входным кольцевым элементом $(l = k + 1)$,

- осуществляют кольцевое сложение, причем кольцевое сложение содержит этапы, на которых

- принимают первый входной для сложения целочисленный список $((a_1, a_2))$, кодирующий первый входной для сложения кольцевой элемент, и второй входной для сложения целочисленный список $((b_1, b_2))$, кодирующий второй входной для сложения кольцевой элемент,

- определяют выходной для сложения целочисленный список, кодирующий выходной для сложения кольцевой элемент, путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для сложения целочисленных списков, причем выходной для сложения кольцевой элемент равен кольцевому сложению первого входного для сложения кольцевого элемента и второго входного для сложения кольцевого элемента, причем определение выходного для сложения целочисленного списка содержит этапы, на которых

- определяют промежуточный целочисленный список сложения $((c_1, c_2))$, кодирующий промежуточный кольцевой элемент сложения, путем первого применения таблицы приращений к кольцевому элементу $(u^{a_1 - b_1} - u^{a_2 - b_1})$, который является линейной комбинацией степеней одного или нескольких элементов основания, причем степени определяются из первого и второго входных для сложения целочисленных списков, (

$$u^{c_1} - u^{c_2} = u^{a_1} - u^{a_2} + u^{b_1}),$$

- определяют выходной для сложения целочисленный список, содержащий второе применение таблицы приращений к кольцевым элементам, определенным из промежуточного целочисленного списка сложения и определенным из второго входного для сложения целочисленного списка.

16. Способ электронного вычисления по п.15, содержащий этап, на котором

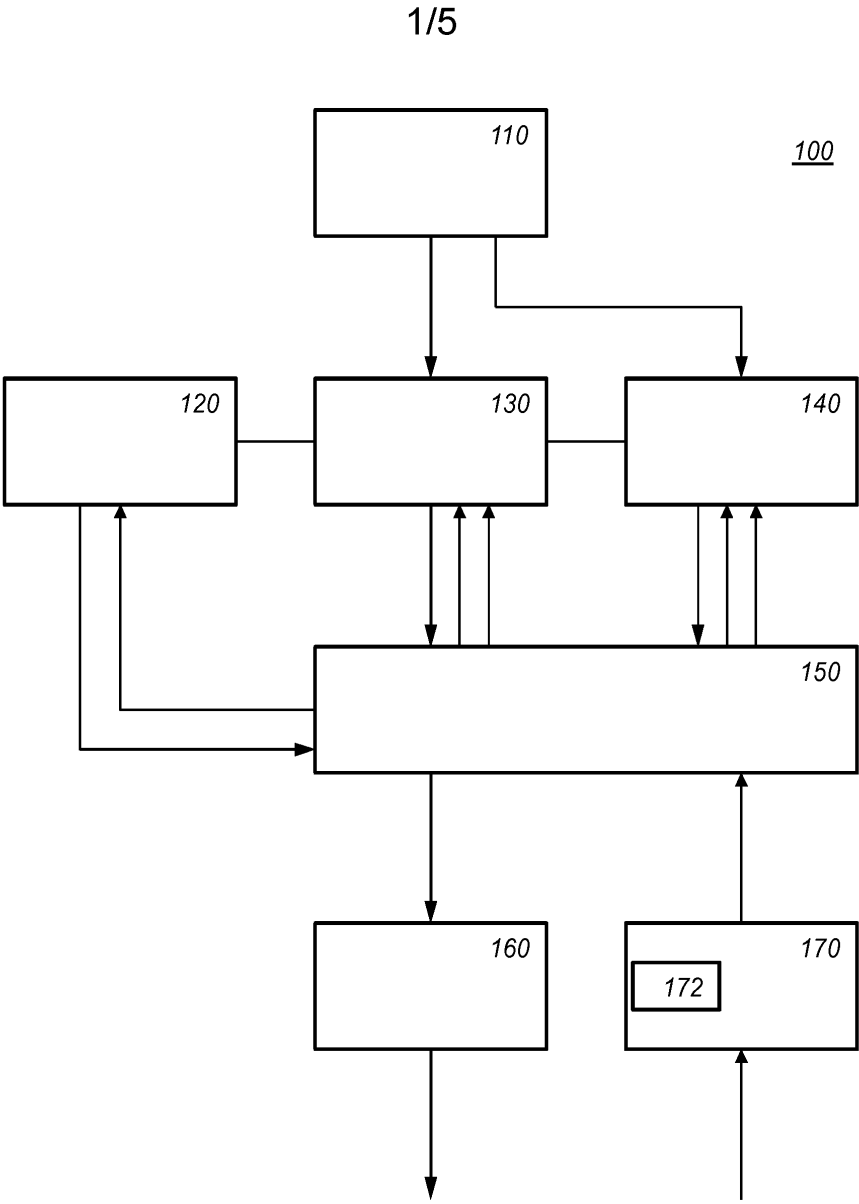
- осуществляют кольцевое умножение, причем кольцевое умножение содержит этапы, на которых

- принимают первый входной для умножения целочисленный список $((r_1, r_2))$, кодирующий первый входной для умножения кольцевой элемент, и второй входной для умножения целочисленный список $((s_1, s_2))$, кодирующий второй входной для умножения кольцевой элемент,

- определяют выходной для умножения целочисленный список, кодирующий выходной для умножения кольцевой элемент путем применения таблицы приращений к кольцевым элементам, определенным из первого и второго входных для умножения целочисленных списков, причем выходной для умножения кольцевой элемент равен кольцевому умножению первого входного для умножения кольцевого элемента и второго входного для умножения кольцевого элемента.

17. Машиночитаемый носитель, содержащий компьютерную программу, содержащую компьютерные программные инструкции, выполненные с возможностью выполнять способ по п.15 или 16, когда компьютерная программа запущена на программируемом устройстве.

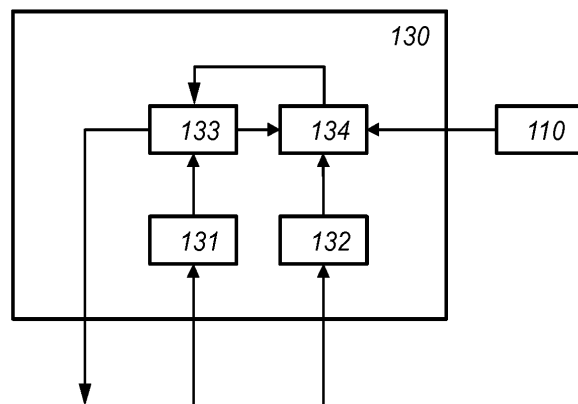
1



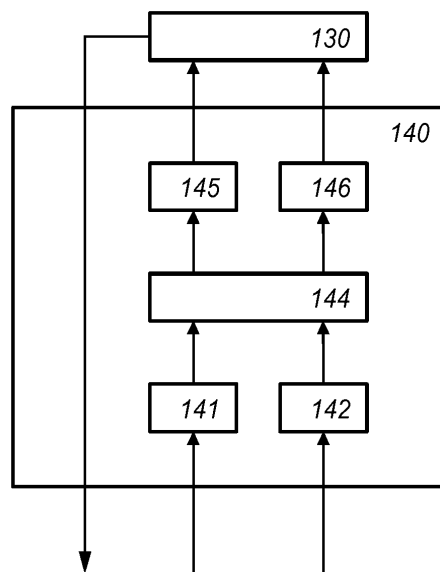
ФИГ. 1А

2

2/5

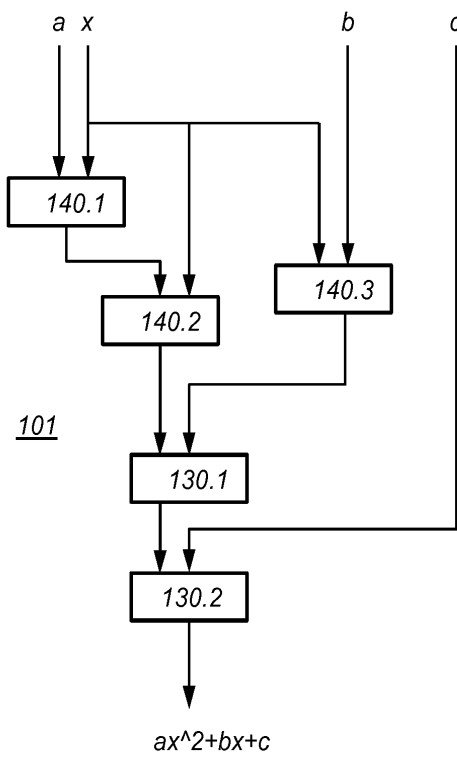


ФИГ. 1В

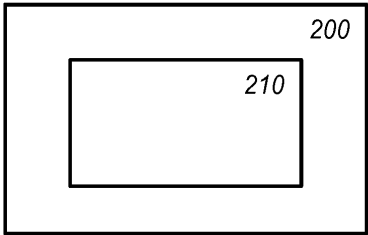


ФИГ. 1С

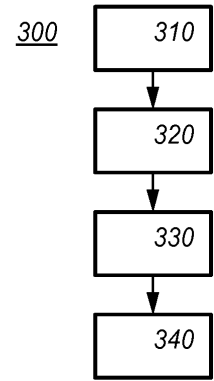
3/5



ФИГ. 2

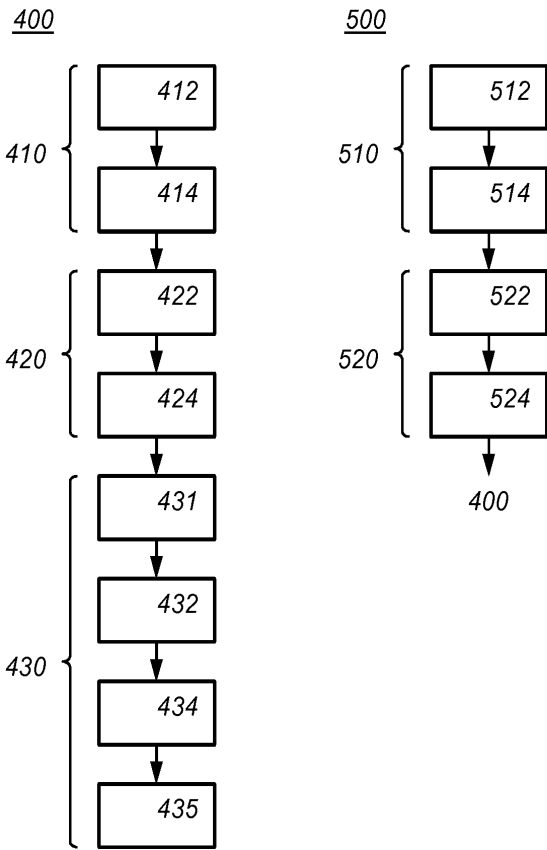


ФИГ. 3



ФИГ. 4

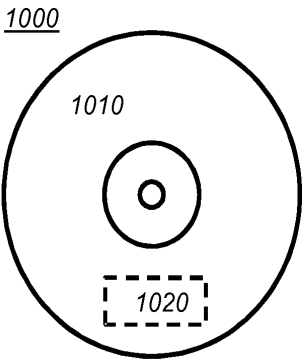
4/5



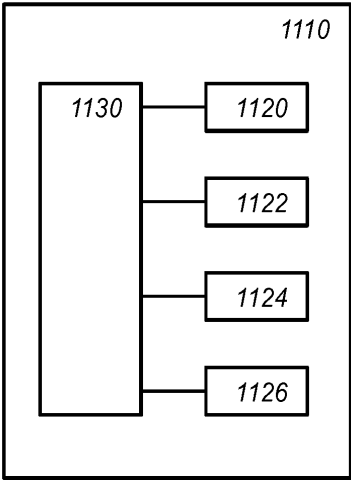
ФИГ. 5

ФИГ. 6

5/5



ФИГ. 7А



1140

ФИГ. 7В