РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 449 482** (13) **C2**

(51) МПК
H04L 9/28 (2006.01)
G09C 1/00 (2006.01)
G09C 1/04 (2006.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2009107223/08, 29.08.2007**

(24) Дата начала отсчета срока действия патента:
29.08.2007

Приоритет(ы):
(30) Конвенционный приоритет:
01.09.2006 JP 2006-238225

(43) Дата публикации заявки: **10.09.2010** Бюл. № 25

(45) Опубликовано: **27.04.2012** Бюл. № 12

(56) Список документов, цитированных в отчете о поиске: **US 3962539 A, 08.06.1976. Ralph C. Merkle «Fast Software Encryption Functions», опубл. 1998, [найдено 19.07.2011]. Найдено в Интернет по адресу URL: http: //www.springerlink.com/content/1756732848224r62/, с.481, строки 6-16, с.484, строка 25 - с.488, строка 9. КОРОТЫГИН С. Шифрующая файловая система (EFS), опубл. 26.01.2001, [найдено** (см. прод.)

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **27.02.2009**

(86) Заявка PCT:
JP 2007/066730 (29.08.2007)

(87) Публикация заявки PCT:
WO 2008/026622 (06.03.2008)

Адрес для переписки:
109012, Москва, ул. Ильинка, 5/2, ООО "Союзпатент", С.В.Истомину

(72) Автор(ы):
СИРАИ Таизо (JP),
СИБУТАНИ Кёдзи (JP),
АКИСИТО Тору (JP),
МОРИАИ Сихо (JP)

(73) Патентообладатель(и):
СОНИ КОРПОРЕЙШН (JP)

(54) УСТРОЙСТВО ОБРАБОТКИ ШИФРОВАНИЯ, СПОСОБ ОБРАБОТКИ ШИФРОВАНИЯ И КОМПЬЮТЕРНАЯ ПРОГРАММА

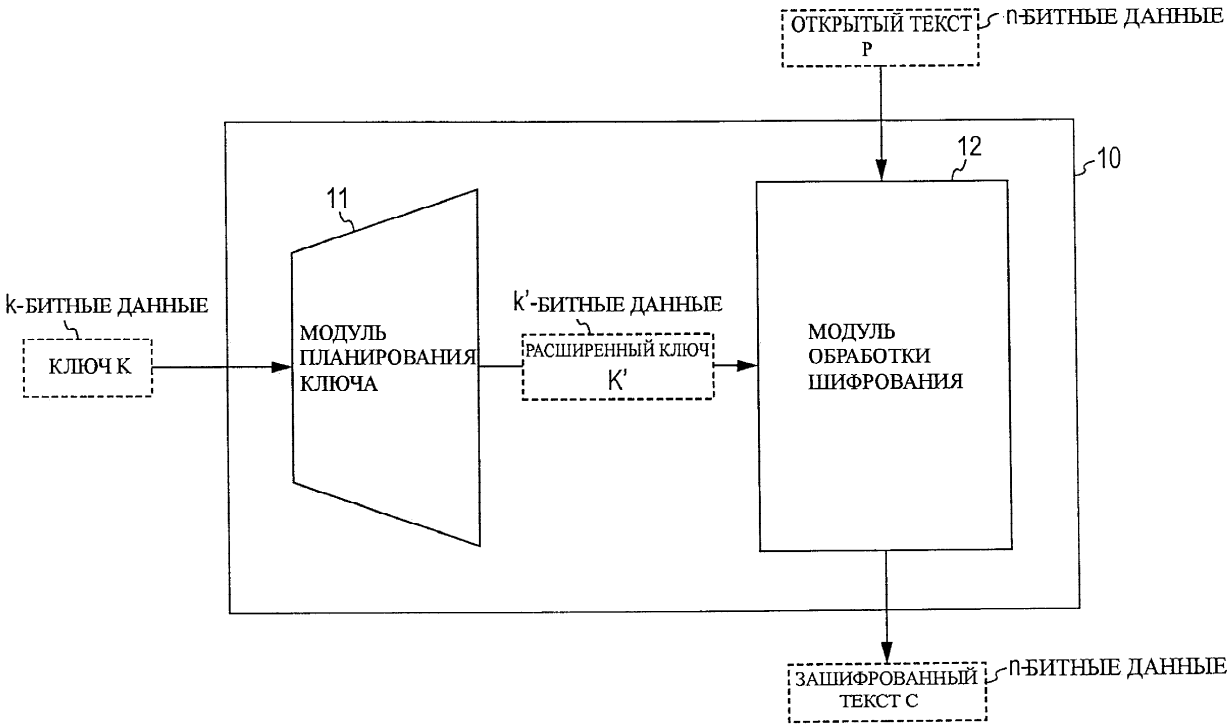
(57) Реферат:

Изобретение относится к устройствам обработки шифрования. Техническим результатом является повышение трудности криптоанализа и воплощение в высокой степени защищенного алгоритма блочного шифра с общим ключом. Реализована конфигурация обработки блочного шифра с общим ключом, с улучшенным иммунитетом против атак, таких как атаки способом насыщения и алгебраические атаки (атаки РЯС). В устройстве обработки шифрования, которое выполняет обработку блочного шифра с общим ключом, S-блоки, использующиеся как модули обработки нелинейного преобразования в раундовой

RU 2449482 C2

RU 2449482 C2

функции, установленные в модулях выполнения раундовых функций, выполнены с возможностью использования S-блоков, по меньшей мере, двух разных типов. С такой конфигурацией можно улучшить иммунитет против атак способом насыщения. Кроме того, типы S-блоков представляют собой смесь различных типов. 2 н. и 12 з.п. ф-лы, 19 ил.



Фиг. 2

(56) (продолжение):
19.07.2011]. Найдено в Интернет по адресу URL: http://www.ixbt.com/storage/efs.html, с.3 строки 7-60, с.4, строки 1-21. RU 2188513 C2, 27.08.2002.

RU 2449482 C2

RU 2449482 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 449 482** (13) **C2**

(51) Int. Cl.
H04L 9/28 (2006.01)
G09C 1/00 (2006.01)
G09C 1/04 (2006.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2009107223/08, 29.08.2007**

(24) Effective date for property rights:
29.08.2007

Priority:
(30) Convention priority:
01.09.2006 JP 2006-238225

(43) Application published: **10.09.2010** Bull. 25

(45) Date of publication: **27.04.2012 Bull. 12**

(85) Commencement of national phase: **27.02.2009**

(86) PCT application:
JP 2007/066730 (29.08.2007)

(87) PCT publication:
WO 2008/026622 (06.03.2008)

Mail address:
109012, Moskva, ul. Il'inka, 5/2, OOO "Sojuzpatent", S.V.Istominu

(72) Inventor(s):
SIRAI Taizo (JP),
SIBUTANI Kedzi (JP),
AKISITO Toru (JP),
MORIAI Sikho (JP)

(73) Proprietor(s):
SONI KORPOREJShN (JP)

RU 2449482 C2

(54) ENCRYPTION PROCESSING DEVICE, ENCRYPTION PROCESSING METHOD AND COMPUTER PROGRAMME

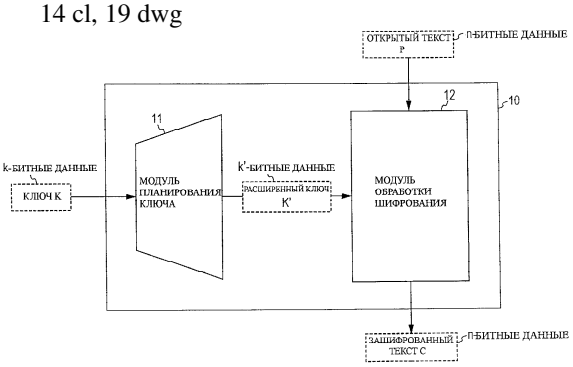
(57) Abstract:

FIELD: information technology.

SUBSTANCE: block cipher with common key processing configuration is implemented with improved immunity against such attacks as saturation attacks and algebraic attacks ("РЯС" attack). In the encryption processing device which executes processing of block cipher with common key, S-blocks used as modules of nonlinear transformation processing in round function and installed in round functions execution modules are made capable to use S-blocks of at least two different types. With such configuration, immunity against saturation attacks can be improved. Additionally, types of S-blocks represent mixture of various types.

EFFECT: increased difficulty of cryptanalysis and implementation of highly protected algorithm of block cipher with common key.

14 cl, 19 dwg



Фиг. 2

RU 2449482 C2

Область техники, к которой относится изобретение

Настоящее изобретение относится к устройствам обработки шифрования, способам обработки шифрования и компьютерным программам. Более конкретно, настоящее изобретение относится к устройству обработки шифрования, способу обработки шифрования и компьютерной программе для выполнения обработки блочного шифра с общим ключом.

Уровень техники

В последние годы, по мере развития сетевой связи и электронной торговли, обеспечение безопасности при передаче данных стало важной проблемой. Один из способов обеспечения безопасности представляет собой криптографическая технология. В настоящее время передача данных фактически осуществляется с использованием различных шифров.

Например, была переведена в практическое использование система, в которой модуль обработки шифрования встроен в малое устройство, такое как карта с микросхемой, и передачу/прием данных выполняют между картой с микросхемой и устройством считывания/записи, используемым как устройство считывания/записи данных, воплощая, таким образом, обработку аутентификации или шифрования/дешифрования передаваемых/принимаемых данных.

Доступны различные алгоритмы обработки шифрования. Эти алгоритмы шифрования можно крупно классифицировать на криптографию с открытым ключом, в которой ключ шифрования и ключ дешифрования установлены как разные ключи, такие как открытый ключ и секретный ключ, и криптографию с общим ключом, в которой ключ шифрования и ключ дешифрования установлены как общий ключ.

Существуют различные алгоритмы для криптографии с общим ключом. Один алгоритм включает в себя генерирование множества ключей на основе общего ключа с многократным выполнением обработки преобразования данных с последовательным приращением блока (например, 64 бита или 128 бит) с применением генерируемых ключей. Типичный алгоритм, в котором применяется такая схема генерирования ключей и обработка преобразования данных, представляет собой блочный шифр с общим ключом.

Известно, что в качестве, например, типичных алгоритмов блочного шифра с общим ключом в прошлом использовали алгоритм DES (СШД, Стандарт шифрования данных), который в прошлом представлял собой стандартный шифр для Соединенных Штатов Америки и алгоритм AES (УСШ, Усовершенствованный стандарт шифрования), который представляет собой стандартный шифр для Соединенных Штатов Америки в настоящее время.

Эти алгоритмы блочного шифра с общим ключом, в основном, состоят из части обработки шифрования, включающей в себя части выполнения раундовой функции, которые многократно выполняют преобразование входных данных, и часть планирования ключа, которая генерирует раундовые ключи, применяемые для соответствующих раундов частей раундовой функции. Часть планирования ключа генерирует расширенный ключ на основе главного ключа (основного ключа), который представляет собой секретный ключ, путем увеличения количества битов, и на основе сгенерированного расширенного ключа генерирует раундовые ключи (дополнительные ключи), применяемые в соответствующих частях раундовой функции части обработки шифрования.

В качестве специфичной структуры для выполнения такого алгоритма известна структура, которая многократно выполняет раундовую функцию, включающую в

себя часть линейного преобразования и часть нелинейного преобразования. Например, структура Фейстеля представляет собой одну типичную структуру. Структура Фейстеля имеет структуру, которая преобразует открытый текст в зашифрованный текст путем простого повторения раундовой функции (F-функции), используемой как функция преобразования данных. В раундовой функции (F-функции) выполняют обработку линейного преобразования и обработку нелинейного преобразования. В качестве документов, описывающих обработку шифрования с применением структуры Фейстеля, можно отметить, например, Непатентный документ 1 и Непатентный документ 2.

Однако при использовании блочного шифра с общим ключом существует проблема утечки ключей в результате криптоанализа. Тот факт, что ключи могут быть легко проанализированы с помощью криптоанализа, означает, что шифр имеет низкую защищенность, в результате чего возникает серьезная проблема при применении.

Непатентный документ 1: K.Nyberg, "Generalized Feistel networks", ASIACRYPT '96, Springer Verlag, 1996, pp.91-104.

Непатентный Документ 2: Yuliang Zheng, Tsutomu Matsumoto, Hideki Imai: On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. CRYPTO 1989: 461-480.

Сущность изобретения

Техническая задача

Настоящее изобретение было выполнено с учетом описанных выше проблем, и цель настоящего изобретения состоит в том, чтобы предоставить устройство обработки шифрования, способ обработки шифрования и компьютерную программу для повышения трудности криптоанализа и воплощения в высокой степени защищенного алгоритма блочного шифра с общим ключом.

Техническое решение

Первый аспект настоящего изобретения направлен на:

устройство обработки шифрования, которое выполняет обработку блочного шифра с общим ключом, отличающееся тем, что включает в себя:

модуль обработки шифрования, который выполняет обработку преобразования данных, в которой раундовую функцию повторяют для множества раундов,

в котором модуль обработки шифрования выполнен с возможностью выполнения обработки нелинейного преобразования с использованием S-блоков при выполнении обработки раундовых функций в соответствующих раундах, и

в котором модуль обработки шифрования выполнен с возможностью выполнения обработки с использованием, по меньшей мере, двух S-блоков разного типа в качестве S-блоков, применяемых при обработке нелинейного преобразования.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования, с использованием структуры Фейстеля, в которой количество строк данных (количество разделений) равно двум, или обобщенной структуры Фейстеля, в которой количество строк данных (количество разделений) равно двум или больше. Модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования с применением F-функций, применяемых как модули выполнения раундовых функций, которые имеют одинаковую строку входных данных и строку выходных данных и которые вертикально расположены рядом друг с другом, в которых обработка нелинейного преобразования, выполняемая F-функциями, установлена как разные S-

блоки, которые выполняют разные типы обработки нелинейного преобразования.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением устройство отличается тем, что каждая из F-функций, используемых как модули выполнения раундовых функций, включает в себя множество S-блоков, которые выполняют обработку нелинейного преобразования соответствующих частей данных, на которые разделены данные, предназначенные для обработки. Множество S-блоков включает в себя, по меньшей мере, два разных типа S-блоков.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что каждая из F-функций, используемых как модули выполнения раундовых функций, включает в себя множество S-блоков, которые выполняют обработку нелинейного преобразования соответствующих частей данных, на которые разделены данные, предназначенные для обработки. Обработка нелинейного преобразования, выполняемая с последовательными приращениями частей данных в F-функциях, используемых как модули выполнения раундовых функций, которые имеют одинаковую строку входных данных и строку выходных данных и которые расположены вертикально рядом друг с другом, выполнена с возможностью установки в качестве других S-блоков, которые выполняют другие типы обработки нелинейного преобразования.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что типы S-блоков и количество отдельных S-блоков, включенных в каждую из F-функций, используемых как модули выполнения раундовых функций, имеют одинаковую установку среди отдельных F-функций.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования выполнен с возможностью использования в качестве разных s-битовых входных/выходных S-блоков, предназначенных для применения при обработке нелинейного преобразования,

(1) типа 1: S-блок, использующий инверсную карту: $Y=X^{-1}$ или показательную функцию $Y=X^q$ по полю расширения GF (2^s);

(2) типа 2: S-блок, генерируемый путем комбинирования множества малых t-битных S-блоков, где $t<s$; и

(3) типа 3: S-блок, выбранный случайным образом,

по меньшей мере, двух разных типов S-блоков среди описанных выше трех типов S-блоков (1)-(3).

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования имеет, в отношении S-блоков, применяемых для выполнения раундовых функций,

(a) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 2;

(b) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 3;

(c) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 2, и остальные S-блоки представляют собой S-блоки типа 3; и

(d) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, некоторые из остальных S-блоков представляют собой S-блоки типа 2, и

остальные из S-блоков представляют собой S-блоки типа 3,

любую одну из описанных выше конфигураций (a)-(d).

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования включает в себя, в модулях выполнения раундовых функций, множество S-блоков, которые выполняют обработку нелинейного преобразования для соответствующих частей данных, на которые разделены данные, предназначенные для обработки. Модуль обработки шифрования выполнен с возможностью выполнения обработки с использованием S-блоков одного типа в одном раунде и S-блоков других типов на основе от раунда к раунду.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования включает в себя, в модулях выполнения раундовых функций, множество S-блоков, которые выполняют обработку нелинейного преобразования соответствующих частей данных, на которые разделены данные, предназначенные для обработки. Модуль обработки шифрования выполнен с возможностью использования различных типов S-блоков в одном раунде.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что типы S-блоков и количество отдельных S-блоков, включенных в каждый из модулей выполнения раундовых функций, имеют одинаковые установки среди отдельных F-функций.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования, в соответствии с криптографией с общим ключом.

Кроме того, в варианте воплощения устройства обработки шифрования в соответствии с настоящим изобретением оно отличается тем, что модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования в соответствии с криптографией блочного шифра с общим ключом.

Кроме того, второй аспект настоящего изобретения направлен на:

способ обработки шифрования, состоящий в выполнении обработки блочного шифра с общим ключом в устройстве обработки шифрования, отличающийся тем, что включает в себя:

этап обработки шифрования, выполняемый в части обработки шифрования, обработки преобразования данных, в которой раундовую функцию повторяют для множества раундов,

в котором этап обработки шифрования представляет собой этап выполнения обработки нелинейного преобразования, с применением S-блоков при обработке существующих раундовых функций в соответствующих раундах, и выполнения обработки с использованием, по меньшей мере, двух разных типов S-блоков в качестве S-блоков, применяемых при обработке нелинейного преобразования.

Кроме того, в варианте воплощения способа обработки шифрования в соответствии с настоящим изобретением он отличается тем, что на этапе обработки шифрования выполняют обработку шифрования в соответствии с криптографией с общим ключом или криптографией блочного шифра с общим ключом.

Кроме того, в третьем аспекте настоящее изобретение относится к:

компьютерной программе, которая обеспечивает выполнение устройством обработки шифрования обработки блочного шифра с общим ключом, отличающейся

тем, что включает в себя:

этап обработки шифрования, обеспечивающий возможность выполнения частью обработки шифрования обработки преобразования данных, в которой раундовую функцию повторяют для множества раундов,

в котором этап обработки шифрования представляет собой этап, обеспечивающий выполнение обработки нелинейного преобразования с применением S-блоков при обработке выполнения раундовых функций в соответствующих раундах, и обеспечения выполнения обработки с использованием, по меньшей мере, двух различных типов S-блоков, в качестве S-блоков, применяемых при обработке нелинейного преобразования, предназначенной для выполнения.

Кроме того, в варианте воплощения компьютерной программы в соответствии с настоящим изобретением, она отличается тем, что этап обработки шифрования представляет собой этап обеспечения выполнения обработки шифрования в соответствии с криптографией с общим ключом или криптографией блочного шифра с общим ключом.

Следует отметить, что компьютерная программа в соответствии с настоящим изобретением представляет компьютерную программу, которая может быть предоставлена через носители записи, такие как носители записи, включающие в себя CD (КД, компакт-диск), FD (ГД, гибкий диск) и MO (МО, магнитооптический диск), или среды передачи данных, такие как сеть, которая позволяет предоставлять программу в формате, считываемом компьютером, например, для компьютерной системы, позволяющей выполнять различные программные коды. Путем предоставления такой программы в считываемом компьютером формате обработка в соответствии с программой может быть выполнена в компьютерной системе.

Другие цели, свойства и предпочтительные эффекты настоящего изобретения будут понятны из следующего подробного описания вариантов выполнения настоящего изобретения и чертежей, приложенных к нему. Следует отметить, что система в настоящем описании относится к логическому узлу из множества устройств и не ограничивается узлом, в котором устройства, имеющие отдельную конфигурацию, содержатся в одном корпусе.

Предпочтительные эффекты

В соответствии с конфигурацией варианта воплощения настоящего изобретения в устройстве обработки шифрования, которое выполняет обработку блочного шифра с общим ключом, оно выполнено с возможностью использования, по меньшей мере, двух разных типов S-блоков в качестве S-блоков, используемых как модуль обработки нелинейного преобразования в модуле выполнения раундовой функции. При использовании такой конфигурации можно улучшить защиту от атак способом насыщения. Кроме того, в соответствии с конфигурацией варианта воплощения настоящего изобретения, в котором типы S блоков представляют смесь разных типов, можно улучшить устойчивость против алгебраических атак (атак XSL (РЯС, расширяемый язык стилей)), реализуя, таким образом, высокозащищенное устройство обработки шифрования.

Краткое описание чертежей

На фиг.1 показана схема, представляющая основную конфигурацию алгоритма блочного шифра с общим ключом.

На фиг.2 показана схема, описывающая внутреннюю конфигурацию модуля E10 обработки блочного шифра с общим ключом, иллюстрируемой на фиг.1.

На фиг.3 показана схема, описывающая подробную конфигурацию модуля 12

обработки шифрования, иллюстрируемую на фиг.2.

На фиг.4 показана схема, описывающая раундовую функцию SPN-структуры, используемую как пример конфигурации модуля выполнения раундовой функции.

На фиг.5 показана схема, описывающая структуру Фейстеля, используемую как пример конфигурации модуля выполнения раундовой функции.

На фиг.6 показана схема, описывающая обобщенную структуру Фейстеля, используемую как пример конфигурации модуля выполнения раундовой функции.

На фиг.7 показана схема, описывающая конкретный пример модуля обработки нелинейного преобразования.

На фиг.8 показана схема, описывающая конкретный пример модуля обработки линейного преобразования.

На фиг.9 показана схема, описывающая пример общей конфигурации структуры Фейстеля или обобщенной структуры Фейстеля.

На фиг.10 показана схема, описывающая пример конфигурации структуры Фейстеля или обобщенной структуры Фейстеля, в которой скомпонованы различные S-блоки.

На фиг.11 показана схема, описывающая пример конфигурации, в которой различные S-блоки установлены для улучшения иммунитета против атак способом насыщения.

На фиг.12 показана схема, описывающая пример конфигурации, в которой различные S-блоки установлены для улучшения иммунитета против атак способом насыщения.

На фиг.13 показана схема, описывающая пример конфигурации, в которой различные S-блоки установлены для улучшения иммунитета против атак способом насыщения.

На фиг.14 показана схема, описывающая пример конфигурации, в которой различные типы S-блоков установлены для улучшения иммунитета против алгебраических атак (атак РЯС).

На фиг.15 показана схема, описывающая пример конфигурации, в которой различные типы S-блоков размещены так, что они улучшают иммунитет против алгебраических атак (атак РЯС).

На фиг.16 показана схема, описывающая пример конфигурации, в которой различные типы S-блоков размещены для улучшения иммунитета против алгебраических атак (атак РЯС).

На фиг.17 показана схема, описывающая пример конфигурации, в которой размещены различные типы S-блоков для улучшения иммунитета против алгебраических атак (атак РЯС).

На фиг.18 показана схема, описывающая пример конфигурации, в которой различные типы S-блоков размещены для улучшения иммунитета против алгебраических атак (атак РЯС).

На фиг.19 показан схема, описывающая пример конфигурации модуля с микросхемой, используемого в качестве устройства обработки шифрования, которое выполняет обработку шифрования в соответствии с настоящим изобретением.

Подробное описание изобретения

Устройство обработки шифрования, способ обработки шифрования и компьютерная программа в соответствии с настоящим изобретением будут подробно описаны ниже. Описание будет приведено в соответствии со следующими разделами:

1. Схематичное представление блочного шифра с общим ключом.

2. Конфигурация, в которой улучшен иммунитет путем размещения множества различных S-блоков.

(2A) Конфигурация, в которой иммунитет против атак способом насыщения улучшают путем размещения двух или больше различных типов S-блоков в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков.

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшают путем смешения двух или больше S-блоков различных типов в блочном шифре, с использованием S-блоков.

(2C) Конфигурация, в которой описанные выше подходы (2A) и (2B) одновременно реализуют в шифре Фейстеля или в шифре Фейстеля обобщенного типа, с использованием S-блоков.

3. Пример конфигурации устройства обработки шифрования.

[1. Схематичное представление блочного шифра с общим ключом]

Вначале будет описана схема блочных шифров с общим ключом, которые можно применять в настоящем изобретении. В этом описании блочные шифры с общим ключом (ниже называемые блочными шифрами) представляют собой блочные шифры, определенные ниже.

В блочный шифр поступает открытый текст P и ключ K, как входные данные, и блочный шифр выводит зашифрованный текст C. Длина битов открытого текста и зашифрованного текста называется размером блока, который обозначен здесь с помощью буквы n. Хотя n может представлять собой любое целое число, в общем случае, n представляет собой заданное значение для каждого алгоритма блочного шифра. Блочный шифр, длина блока которого равна n, может называться n-битным блочным шифром.

Длина в битах ключа обозначена как k. Ключ может представлять собой любое целое число. Алгоритм блочного шифра с общим ключом предназначен для работы с одним или с множеством размеров ключа. Например, один алгоритм A блочного шифра имеет размер блока n=128 и может быть выполнен с возможностью обработки различных размеров, то есть, длины в битах k=128, k=192 или k=256 ключа.

Индивидуальные размеры битов открытого текста [P], зашифрованного текста [C] и ключа [K] обозначают следующим образом:

открытый текст P: n битов;

зашифрованный текст C: n битов;

ключ K: k битов.

Алгоритм E блочного шифра с общим ключом длиной n-битов, позволяющий обрабатывать ключ длиной k-битов, представлен на фиг.1. Как показано на фиг.1, в модуль E10 обработки блочного шифра с общим ключом подают n-битный открытый текст P и k-битный ключ K, он выполняет заданный алгоритм шифрования и выводит n-битный зашифрованный текст C. Следует отметить, что, хотя обработка шифрования, состоящая в генерировании зашифрованного текста из открытого текста, показана на фиг.1, при обработке дешифрования, состоящей в генерировании открытого текста из зашифрованного текста, обычно используют функцию обратную E10. Следует отметить, что, в зависимости от структуры модуля E10 обработки шифрования, аналогичная модуль E10 обработки блочного шифра с общим ключом также применим для обработки дешифрования, и обработка дешифрования позволяет изменять последовательность, такую как порядок вводимых ключей или тому подобное.

Со ссылкой на фиг.2 будет описана внутренняя конфигурация модуля E10

обработки блочного шифра с общим ключом, иллюстрируемой на фиг.1. Блочный шифр можно рассматривать как имеющий два отдельных модуля. Один модуль представляет собой модуль 11 планирования ключа, которая принимает ключ K как входные данные, расширяет длину битов, вводимого ключа K, выполняя заданные этапы, и выводит расширенный ключ K' (длина k' в битах), и другой модуль представляет собой модуль 12 обработки шифрования, который выполняет преобразование данных для генерирования зашифрованного текста C, принимая расширенный ключ K', вводимый из модуля 11 планирования ключа, получая открытый текст P в качестве входных данных, и выполняя обработку шифрования, применяя расширенный ключ K'. Следует отметить, что, как было описано выше, в зависимости от структуры модуля 12 обработки шифрования, модуль 12 обработки шифрования применим к обработке дешифрования данных, состоящей в преобразовании зашифрованного текста обратно в открытый текст.

Далее, со ссылкой на фиг.3, будет описана подробная конфигурация модуля 12 обработки шифрования, иллюстрируемой на фиг.2. Как показано на фиг.3, модуль 12 обработки шифрования выполнен с возможностью многократного выполнения преобразования данных, применяя модуля 20 выполнения раундовой функции. Таким образом, модуль 12 обработки шифрования может быть разделен на модули для обработки, которые представляют собой модуля 20 выполнения раундовой функции. Каждый модуль 20 выполнения раундовой функции принимает две части данных, как входные данные, выход X_i модуля выполнения раундовой функции на предыдущем этапе и раундовый ключ PK_i, сгенерированный на основе расширенного ключа, выполняет в них обработку преобразования данных и выводит выходные данные X_{i+1} в следующий модуль выполнения раундовой функции. Следует отметить, что вход в первый раунд представляет собой открытый текст или данные обработки инициализации для открытого текста. Кроме того, выход из последнего раунда представляет собой зашифрованный текст.

В примере, иллюстрируемом на фиг.3, модуль 12 обработки шифрования имеет r модулей 20 выполнения раундовой функции и выполнен с возможностью многократного выполнения преобразования данных r раз в модулях выполнения раундовой функции для генерирования зашифрованного текста. Количество раз выполнения раундовой функции называется количеством раундов. В представленном примере количество раундов равно r.

Входные данные X_i каждого модуля выполнения раундовой функции, представляют собой n-битные данные, подвергаемые шифрованию. Выход X_{i+1} раундовой функции в определенном раунде подают как вход в следующий раунд. В качестве других входных данных в каждый модуль выполнения раундовой функции используют данные, основанные на расширенном ключе K', выводимом из модуля планирования ключа. Ключ, вводимый в каждый модуль выполнения раундовой функции и применяемый для выполнения раундовой функции, называется раундовым ключом. На схеме раундовый ключ, применяемый в i-ом раунде, обозначен как RK_i. Расширенный ключ K' выполнен как, например, объединенные данные раундовых ключей RK_1-RK_r для r раундов.

Конфигурация, представленная на фиг.3, представляет собой конфигурацию модуля 12 обработки шифрования, в которой входные данные в первый раунд, когда их рассматривают со стороны ввода модуля 12 обработки шифрования, обозначены как X_0, данные, выводимые из i-ой раундовой функции, обозначены как X_i, и раундовый ключ обозначен как RK_i. Следует отметить, что в зависимости от

структуры модуля 12 обработки шифрования, например, путем установки последовательности применения для применяемых раундовых ключей так, чтобы они были противоположны ключам при обработке шифрования, и при вводе зашифрованного текста в модуль 12 обработки шифрования модуль 12 обработки шифрования может быть выполнен с возможностью вывода открытого текста.

Модули 20 выполнения раундовой функции модуля 12 обработки шифрования, показанной на фиг.3, могут иметь различные формы. Раундовые функции могут быть классифицированы в соответствии со структурами, принятыми соответствующими алгоритмами шифрования. Представительные структуры включают в себя следующие:

(a) структура SPN (СПП, Сеть с использованием подстановок-перестановок);

(b) Структура Фейстеля; и

(c) Обобщенная структура Фейстеля.

Эти конкретные структуры будут описаны ниже со ссылкой на фиг.4-6.

(a) Раундовая функция структуры СПП

Прежде всего, со ссылкой на фиг.4, будет описана раундовая функция структуры СПП, используемая в качестве примера конфигурации модуля 20 выполнения раундовой функции. Модуль 20a исполнения раундовой функции структуры СПП имеет так называемую структуру типа SP (ПП, подстановок-перестановок), в которой соединены нелинейный уровень преобразования (уровень S) и линейный уровень преобразования (уровень P). Как показано на фиг.4, модуль 20a исполнения раундовой функции структуры СПП построен из модуля 21 расчета "исключающее ИЛИ", который выполняет операцию "исключающее ИЛИ" (EXOR, ИИЛИ) для всех n-битных входных данных и раундового ключа, модуля 22 обработки нелинейного преобразования, который принимает результат операции, полученный модулем 21 расчета "исключающее ИЛИ" в качестве входных данных, и выполняет нелинейное преобразование этих входных данных, модуля 23 обработки линейного преобразования, который принимает результат обработки нелинейного преобразования, полученный модулем 22 обработки нелинейного преобразования в качестве входных данных, и выполняет обработку линейного преобразования входных данных, и т.п. Результат обработки линейного преобразования, полученный модулем 23 обработки линейного преобразования, выводят в следующий раунд. Выход последнего раунда представляет собой зашифрованный текст. Следует отметить, что, хотя последовательность обработки модуля 21 расчета "исключающее ИЛИ", модуля 22 обработки нелинейного преобразования и модуля 23 обработки линейного преобразования иллюстрируется в примере, показанном на фиг.4, последовательность модулей обработки не ограничивается этим, и обработка может выполняться в других последовательностях.

(b) Структура Фейстеля

Далее, со ссылкой на фиг.5, будет описана структура Фейстеля, используемая в качестве примера конфигурации модуля 20 выполнения раундовой функции. Структура Фейстеля выполняет, как показано на фиг.5, обработку путем разделения n-битных входных данных из предыдущего раунда (входной текст в первом раунде) на два равных модуля данных длиной n/2-бит и расширяет эти два модуля данных друг для друга на основании от раунда к раунду.

При обработке с применением модуля 20b выполнения раундовой функции, со структурой Фейстеля, как показано на чертеже, один модуль n/2-битных данных и раундовый ключ вводят в часть 30 F-функции. Часть 30 F-функции имеет, как и описанная выше структура СПП так называемую структуру ПП-типа, в которой

уровень нелинейного преобразования (S уровень) и уровень линейного преобразования (P уровень) соединены друг с другом.

Одну часть n/2-битных данных из предыдущего раунда и раундовый ключ вводят в модуль 31 расчета "исключающее ИЛИ" части 30 F-функции и выполняют обработку "исключающее ИЛИ" (ИИЛИ). Кроме того, эти полученные в результате данные вводят в модуль 32 обработки нелинейного преобразования для нелинейного преобразования. Кроме того, результат нелинейного преобразования вводят в модуль 33 линейного преобразования для линейного преобразования. Результат линейного преобразования выводят как получаемые в результате данные обработки F-функции.

Кроме того, выход F-функции и другой модуль n/2-битных данных, вводимых из предыдущего раунда, подают в модуль 34 расчета "исключающее ИЛИ" и выполняют операцию "исключающее ИЛИ" (ИИЛИ). Результат выполнения установлен как вход для F-функции следующего раунда. Следует отметить, что n/2-битов, установленные как входные данные для F-функции i-ого раунда, представленной на схеме, применяют в операции "исключающее ИЛИ" с выводом F-функции следующего раунда. Таким образом, структура Фейстеля выполняет обработку преобразования данных, применяя F функции, при обмене входами друг с другом на основе от раунда к раунду.

(c) Обобщенная структура Фейстеля

Далее, со ссылкой на фиг.6, будет описана обобщенная структура Фейстеля, используемая в качестве примера конфигурации модуля 20 выполнения раундовой функции. Структура Фейстеля, которая была описана выше со ссылкой на фиг.5, выполняет обработку путем разделения n-битового открытого текста на две равные части, имеющие по n/2 бит. Таким образом, количество разделений d равно двум при обработке. Следует отметить, что количество разделений также может называться количеством строк данных.

Обобщенная структура Фейстеля устанавливает количество строк d данных (количество разделений) равным любому целому числу, большему или равному двум. Различные обобщенные структуры Фейстеля могут быть определены в соответствии со значением количества d строк данных (количества разделений). В примере, показанном на фиг.6, количество строк d данных (количество разделений) равно четырем, и n/4-битовые данные вводят в каждую строку данных. В каждом раунде выполняют одну или больше F-функций, используемых как раундовые функции. Иллюстрируемый пример представляет собой пример конфигурации выполнения раундовых операций с использованием двух модулей F-функций в каждом раунде.

Конфигурация модулей 41 и 42 F-функции аналогична конфигурации модуля 30 F-функции, описанной выше со ссылкой на фиг.5. Модули 41 и 42 F-функции выполнены с возможностью выполнения операции "исключающее ИЛИ" по раундовому ключу и входному значению, обработки нелинейного преобразования и обработки линейного преобразования. Следует отметить, что раундовый ключ, вводимый в каждый из модулей F-функции, регулируют таким образом, что количество битов раундового ключа совпадает с количеством битов входных битов. В представленном примере количество битов раундовых ключей, вводимых в соответствующие модули 41 и 42 F-функции, равно n/4 бита. Эти ключи генерируют путем сегментирования битов каждого из раундовых ключей, составляющих расширенный ключ. Следует отметить, что, пусть d представляет собой количество строк данных (количество разделений), тогда данные, вводимые в каждую строку, составляют n/d битов, и количество битов ключа, вводимого в каждую F-функцию, регулируют так, чтобы оно составляло n/d

битов.

Следует отметить, что обобщенная структура Фейстеля, представленная на фиг.6, представляет собой пример конфигурации, в которой пусть d будет равно количеству строк данных (количество разделений), тогда d/2 F-функции выполняют параллельно друг с другом в каждом раунде. Обобщенная структура Фейстеля может быть выполнена с возможностью выполнения, по меньшей мере, одной и меньше чем или равного d/2 количества F-функций в каждом раунде.

Как было описано со ссылкой на фиг.4-6, модуль 20 выполнения раундовой функции модуля 12 обработки шифрования в блочном шифре с общим ключом может иметь одну из следующих структур:

(a) СПП (структура сети с использованием подстановок-перестановок).

(b) Структура Фейстеля.

(c) Обобщенная структура Фейстеля.

Каждый из этих модулей выполнения раундовой функции имеет так называемую структуру типа ПП, в которой соединены уровень нелинейного преобразования (уровень S) и уровень линейного преобразования (уровень P). Таким образом, каждый исполнительный модуль раундовой функции имеет модуль обработки нелинейного преобразования, который выполняет обработку нелинейного преобразования, и модуль обработки линейного преобразования, который выполняет обработку линейного преобразования. Такие конфигурации обработки преобразования будут описаны ниже.

(Модуль обработки нелинейного преобразования)

Со ссылкой на фиг.7 будет описан конкретный пример модуля обработки нелинейного преобразования. Как показано на фиг.7, модуль 50 обработки нелинейного преобразования включает в себя, в частности, массив из m таблиц нелинейного преобразования, называемых S-блоками 51, каждый из которых занимает s битов, в качестве входных данных и генерирует s битов в качестве выходных данных, в которых ms-битные входные данные разделены на равные части s-битных данных, и части этих данных вводят в соответствующие S-блоки 51 и преобразуют. Каждый из S-блоков 51 выполняет обработку нелинейного преобразования путем применения, например, таблицы преобразования.

Существует тенденция, состоящая в том, что по мере того, как размер входных данных увеличивается, также повышается стоимость воплощения. Для исключения этого, во многих случаях, как показано на фиг.7, используется конфигурация разделения данных X, предназначенных для обработки, на множество частей и выполнения нелинейного преобразования каждой части. Например, если входной размер равен ms битов, входные данные разделяют на m частей данных размером s битов, и m частей s данных размером s битов вводят в соответствующие S-блоки 51 для нелинейного преобразования путем применения, например, таблицы преобразования, и m частей выходных s-битовых данных комбинируют для получения ms-битного нелинейного результата преобразования.

(Модуль обработки линейного преобразования)

Со ссылкой на фиг.8 будет описан конкретный пример модуля обработки линейного преобразования. Модуль обработки линейного преобразования принимает, как входные данные, входное значение, такое как выходное значение размером ms битов, которое представляет собой выходные данные, из S-блоков, в качестве входного значения X, применяет линейное преобразование к этим входным данным и выводит результат размером ms битов. Обработка линейного преобразования

выполняет обработку линейного преобразования, такую как обработка перестановки положений входных битов, и выводит выходное значение Y размером ms битов. Обработка линейного преобразования применяет, например, матрицу линейного преобразования к входным данным и позволяет выполнить обработку перестановки положений входных битов. Пример матрицы представляет собой матрицу линейного преобразования, показанную на фиг.8.

Элементы матрицы линейного преобразования, применяемого к модулю обработки линейного преобразования, могут быть, в общем, выполнены как различные представления, применимые к матрице, такие как элементы в полях расширения $GF(2^8)$ или элементы в поле GF(2). На фиг.8 иллюстрируется пример конфигурации модуля обработки линейного преобразования, в которую поступают ms-битные входные данные, в которой генерируют ms-битные выходные данные и которая определена по матрице m×m, полученной через $GF(2^s)$.

[2. Конфигурация, в которой улучшен иммунитет путем размещения множества различных S-блоков]

Как было описано выше, блочный шифр с общим ключом выполнен с возможностью выполнения обработки шифрования путем многократного выполнения раундовой функции. Обработка блочного шифра с общим ключом имеет проблему утечки ключей в результате криптоанализа. Тот факт, что ключи могут быть легко проанализированы с помощью криптоанализа, означает, что шифр обладает низкой безопасностью, что приводит к серьезной проблеме во время применения. В дальнейшем будет описана конфигурация обработки шифрования, в которой иммунитет улучшен путем размещения множества различных S блоков.

Как было описано со ссылкой на фиг.7, модуль обработки нелинейного преобразования, включенный в каждый модуль выполнения раундовой функции, включает в себя множество S-блоков, которые выполняют обработку нелинейного преобразования. В случае необходимости таблицу обработки общего нелинейного преобразования применяют ко всем S-блокам, и S-блоки выполнены так, чтобы они выполняли общую нелинейную обработку преобразования.

В настоящем изобретении предложена конфигурация, в которой уделяется внимание уязвимости, связанной с таким подобием S-блоков, то есть восприимчивости к атакам, а именно криптоанализу, такому как анализ ключей, и иммунитет повышают путем установки множества разных S-блоков.

Ниже, как варианты воплощения настоящего изобретения, будут последовательно описаны следующие три варианта воплощения.

(2A) Конфигурация, в которой иммунитет против атак способом насыщения улучшают путем размещения двух или больше различных типов S-блоков в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшают путем смешения двух или больше S-блоков различных типов в блочном шифре с использованием S-блоков

(2C) Конфигурация, в которой описанные выше подходы (2A) и (2B) одновременно реализуют в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков

(2A) Конфигурация, в которой иммунитет против атак способом насыщения улучшают путем размещения двух или больше различных типов S-блоков в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков

Прежде всего, будет описана конфигурация, в которой улучшен иммунитет против

атак способом насыщения, благодаря размещению двух или больше различных типов S-блоков в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков.

(2A-1. Краткое описание атаки способом насыщения)

Прежде всего, будут описаны атаки способом насыщения, известные как атаки, направленные против блочных шифров. Существует множество типов атак способом насыщения. Первый тип представляет собой процесс атаки, в котором используется характеристика, состоящая в том, что, если 256 типов значений вводят одновременно в определенном положении данных открытого текста после того, как обработка преобразования в раунде будет выполнена для множества раундов, все 256 типов значений появляются в определенном положении байта выходного значения.

Кроме того, в качестве другого типа атаки способом насыщения существует технология атаки, в которой используется характеристика, состоящая в том, что сумма значений, появляющихся в определенном положении байта после того, как преобразование в раунде будет выполнено для множества раундов, всегда равно нулю.

Например, в качестве 256 типов открытого текста P_0-P_{255}, которые вводят в устройство обработки блочного шифра с общим ключом, выполняющего раундовые функции, последовательно вводят данные типы открытого текста P_0-P_{255}

$P_0 = (0, 0, 0, 0, 0, 0, 0, 0)$

$P_1 = (0, 0, 0, 0, 0, 0, 0, 1)$

...

$P_{255} = (0, 0, 0, 0, 0, 0, 0, 255)$

Следует отметить, что, в представленном выше представлении, каждый [0] обозначает 1 байт данных 0.

В случае, когда последовательно вводят эти типы открытого текста P_0-P_{255}, выходные значения, полученные после того, как обработка преобразования данных будет выполнена для определенных раундов, обозначены как C_0-C_{255} следующим образом:

$C_0 = (c_0, ?, ?, ?, ?, ?, ?, ?)$

$C_1 = (c_1, ?, ?, ?, ?, ?, ?, ?)$

...

$C_{255} = (c_{255}, ?, ?, ?, ?, ?, ?, ?)$

В описанных выше выходных значениях [?] может представлять собой любое значение бита.

Эти выходные значения C_0-C255 имеют, как было описано выше, характеристику, состоящую в том, что все 256 типов значений C_0-C_{255} появляются в определенном положении байта (первое положение байта в представленном выше примере). Если заранее известно, как отмечено выше, когда значения от 0 до 255 появляются однажды без различия порядка появления, атака может быть выполнена с использованием этой характеристики. Известно, что раундовые ключи можно оценивать путем анализа выходных значений, полученных в результате последовательного изменения входного значения.

Кроме того, в случае, когда сумма (ИИЛИ) значений c_0-c_{255} в определенном положении байта, включенном в выходные данные C_0-C_{255}, равна нулю, можно выполнить атаку (криптоанализ), используя эту характеристику. Таким образом, ключи можно оценивать путем последовательного ввода 256 типов открытого текста P_0-P_{255} и анализируя выходные данные в определенном положении байта.

Когда результаты преобразования модулей раундовой функции имеют выходные данные с определенной регулярностью, такой, как описано выше, то есть, появляются все 256 типов значений c_0-c_{255}, или

сумма (ИИЛИ) значений c_0-c_{255} в определенном положении байта равна нулю,

при выводе с проявлением такой регулярности, атака способом насыщения представляет собой технологию атаки (анализ), выполняемую на основе этой регулярности.

Поэтому, для того чтобы получить шифр, защищенный от атак способом насыщения, на этапе конструирования шифра эффективно выполнить конфигурацию шифра таким образом, чтобы не генерировать такие специфичные выходные данные, как выходные данные модулей раундовой функции. Следует отметить, что атаки способом насыщения не ограничиваются анализом на основе от-байта-к-байту (8 битов); атаки, использующие аналогичную характеристику, могут быть выполнены по произвольной длине битов.

(2A-2) Проблемы при обработке шифрования с применением структуры Фейстеля или обобщенной структуры Фейстеля

Далее будут описаны проблемы, возникающие при обработке шифрования, с применением структуры Фейстеля или обобщенной структуры Фейстеля.

Что касается структуры Фейстеля или обобщенной структуры Фейстеля, обе они выполнены так, чтобы повторять раундовую операцию с применением модуля F-функции ПП - типа, включающего в себя модуль обработки нелинейного преобразования и модуль обработки линейного преобразования, как было описано выше со ссылкой на фиг.5 и фиг.6. В структуре Фейстеля количество строк данных (количество разделений) ограничено двумя; однако в обобщенной структуре Фейстеля различие состоит в том, что количество строк данных (количество разделений) установлено равным любому числу, большему или равному двум.

В следующем описании предполагается конфигурация, в которой при обработке шифрования с применением структуры Фейстеля или обобщенной структуры Фейстеля используют S-блоки в части обработки нелинейного преобразования, в каждой F-функции, которая представляет собой модуль выполнения раундовой функции. Как было описано выше со ссылкой на фиг.7, S-блоки, соответственно, выполняют, например, путем применения таблицы нелинейного преобразования, обработки нелинейного преобразования m частей s-битовых данных, на которые разделяют данные размером ms битов, который вводят в модуль обработки нелинейного преобразования.

Как было описано выше, что касается F-функций применяемых для выполнения раундовых функций в обычном блочном шифре, одну и ту же F-функцию многократно используют в каждом раунде. Структура Фейстеля или обобщенная структура Фейстеля, в которой одна и также F-функция установлена в каждом раунде, в большей степени подвержена описанным выше атакам способом насыщения. Причина этого будет описана со ссылкой на фиг.9.

На фиг.9 показана схема, представляющая конфигурацию вырезанного участка структуры Фейстеля или обобщенной структуры Фейстеля. Таким образом, два модуля, выполняющих раундовую функцию, а именно F-функции 101 и 102, включенные в шифр со структурой Фейстеля или с обобщенной структурой Фейстеля, показаны на фиг.9. Эти две F-функции 101 и 102 представляют собой F-функции, которые имеют одинаковую строку (x) входных данных и строку (y) выходных данных, и вертикально расположены рядом друг с другом.

Две F-функции 101 и 102 включают в себя модули расчета "исключающее ИЛИ", которые рассчитывают функцию "исключающее ИЛИ" с раундовым ключом, модули обработки нелинейного преобразования и модули обработки линейного преобразования. В этом примере обработки F-функции 101 и 102 выполнены с возможностью выполнения 32-битной входной и выходной обработки. Каждый модуль обработки нелинейного преобразования включают в себя четыре S-блока, и каждый из S-блоков принимает 8-битные входные данные и формирует 8-битные выходные данные.

Позиции от A до J, показанные на фиг.9, обозначают различные типы данных, то есть обозначены следующие типы данных.

A: вход в предыдущую F-функцию 101;

B: выход из предыдущей F-функции 101;

C: вход в последующую F-функцию 102;

D: выход из последующей F-функции 102;

E: данные операции "исключающее ИЛИ" с выходом B из предыдущей F-функции 101;

F: данные операции "исключающее ИЛИ" с данными A;

G: результат операции "исключающее ИЛИ" по данным B и данным E;

H: результат операции "исключающее ИЛИ" по данным D и данным G;

I: ввод раундового ключа в предыдущую F-функцию 101; и

J: ввод раундового ключа в последующую F-функцию 102

В следующем описании, в случае, когда 32-битные данные, предназначенные для обработки каждой из F-функций 101 и 102, представлены в виде приращений 1 байт (восемь битов), например, если данные A представляют собой 32-битные данные, они представляют собой объединенные данные 1 байта (8 бит) данных [0], [1], [2] и [3], при этом данные A будут представлены, как указано выше:

A=[0]||[1]||[2]||[3].

Здесь предполагается, что в качестве, например, 256 типов данных используется открытый текст, предназначенный для ввода в конфигурацию обработки шифрования, показанную на фиг.9

$P_0 = (0, 0, 0, 0)$

$P_1 = (1, 0, 0, 0)$

…

$P_{255} = (255, 0, 0, 0)$

эти типы открытого текста P_0-P_{255} последовательно вводят. Следует отметить, что в описанном выше представлении каждые из [0], [1], … [255] обозначают данные размером 1 байт.

Предполагается, что эти входные значения используются как входные данные A для предыдущей F-функции 101, показанной на фиг.9. Данные A представляют собой такие данные, что, как было описано выше, в случае, когда наблюдают 256 типов данных, предполагается, что все эти 256 типов значений от 0 до 255 появляются в первом байте A[0], и остальные положения байтов являются фиксированными с одним и тем же значением (это предполагается в связи с тем, что атакующий, пытающийся запустить атаку способом насыщения, может сгенерировать такую ситуацию, управляя вводом открытого текста).

Кроме того, предполагая, что значение данных F для операции "исключающее ИЛИ" с данными A остается фиксированным все время при обработке

последовательного ввода описанных выше 256 типов данных A, обеспечивается, что все 256 типов значений от 0 до 255 появляются в первом байте C[0] входных данных C последующей F-функции 102, и что остальные положения байтов являются фиксированными с одинаковым значением.

В таком случае, в зависимости от комбинации следующих значений элементов данных:

I: раундовый ключ, вводимый в предыдущую F-функцию 101;

J: раундовый ключ, вводимый в последующую F-функцию 102; и

F: данные для операции "исключающее ИЛИ" с данными A,

всегда выполняется следующее уравнение:

A [0] (ИИЛИ) I [0]=C [0] (ИИЛИ) J [0]

Описанное выше уравнение может быть справедливым.

Следует отметить, что (ИИЛИ) обозначает операцию "исключающее ИЛИ", и что A[0] (ИИЛИ) I [0] обозначает операцию "исключающее ИЛИ" для данных [0] и данных I [0], и

C [0] (ИИЛИ) J [0] обозначает операцию "исключающее ИЛИ" для данных C [0] и данных J [0].

Уравнение: A [0] (ИИЛИ) I [0]=C [0] (ИИЛИ) J [0]

Это уравнение означает, что одно и то же значение всегда вводят в два S-блока в двух F-функциях 101 и 102. Эти S-блоки выполняют одинаковую обработку нелинейного преобразования, и выводят одинаковое выходное значение для одинакового входного значения. Поэтому эти два S-блока двух F-функций 101 и 102 всегда имеют одинаковые выходы. Одинаковые выходы S-блоков линейно преобразуют с помощью матриц модулей обработки линейного преобразования отдельных F-функций 101 и 102 и результаты выводят в модули расчета "исключающее ИЛИ" по строке (y) данных с правой стороны. Эти модули расчета "исключающее ИЛИ" представляют собой модули 111 и 112, показанные на схеме.

Значения B и D, выводимые из этих двух F-функций 101 и 102 модулей 111 и 112 расчета "исключающее ИЛИ", соответственно, имеют специфичное дифференциальное значение Δ. Таким образом,

B (ИИЛИ) Δ=D.

В этом случае модуль 111 расчета "исключающее ИЛИ" рассчитывает данные G, путем расчета

G=B (ИИЛИ) E,

и модуль 112 расчета "исключающее ИЛИ" рассчитывает

H=G (ИИЛИ) D.

Поскольку G=B (ИИЛИ) E и B (ИИЛИ) A=D, описанное выше уравнение H=G (ИИЛИ) D представляет собой:

H=B (ИИЛИ) E (ИИЛИ) B (ИИЛИ) Δ

=E (ИИЛИ) Δ.

Таким образом, результат выполнения операции "исключающее ИЛИ" по значениям, имеющим фиксированное дифференциальное значение, представляет собой фиксированное значение Δ, и, в результате

H=B (ИИЛИ) E (ИИЛИ) B (ИИЛИ) Δ

=Δ (ИИЛИ) E

=E (ИИЛИ) Δ.

Таким образом, выход H модуля 112 расчета "исключающее ИЛИ" представляет собой результат операции "исключающее ИЛИ" по данным E и фиксированному

значению Δ. Хотя раундовую функцию (F-функцию) выполняют в два этапа, в результате этих данных не выполняют перестановку. Используя такую характеристику, может быть легко получена оценка раундового ключа последующего раунда. Таким образом, если существует последующий раунд, временно установленный ключ используют в этом раунде для дешифрования данных, вплоть до данных H, и проверяют, можно или нет наблюдать эту характеристику, пробабилистичным способом определяя, таким образом, является ли временно используемый ключ правильным или нет. Другими словами, можно оценивать раундовый ключ, и можно выполнять анализ, используя атаку способом насыщения.

Для того чтобы исключить такие ситуации, участок матрицы, применяемый при обработке линейного преобразования, может быть изменен в зависимости от положения каждой F-функции. В случае, когда S-блоки отдельных F-функций одинаковы, если возникает условие, аналогичное описанному выше, в зависимости от взаимосвязи между элементами матрицы линейного преобразования, некоторые из байтов могут смещать друг друга в момент, когда выходные данные D последней F-функции 102 подвергают обработке " исключающее ИЛИ" с данными G, в результате чего возникает благоприятная ситуация для атакующего.

В соответствии с этим, в случае, когда обработку нелинейного преобразования с одинаковой конфигурацией применяют, по меньшей мере, для множества F-функций, которые выводят данные в одной строке, можно выполнять оценку ключей при осуществлении атаки способом насыщения. Кроме того, в зависимости от S-блоков, как результат их операций (ИИЛИ), то есть

S (A [0] (ИИЛИ) I [0] (ИИЛИ) S (C [0] (ИИЛИ) J [0])

в качестве результата их результатов, нельзя назвать желательным случай, когда появляются все 256 типов значений от 0 до 255. В нормальных условиях, даже в случае, когда оба вывода A[0] и C[0] выводят 256 разных типов значений, результаты операций (ИИЛИ) по A[0] и C[0] не обязательно могут занимать все 256 типов выходных значений. Однако такая ситуация может возникать в зависимости от S-блоков. Если возникает такая неожиданная ситуация, информацию, которую можно использовать для выполнения атаки (информация, обозначающая, что все значения являются разными), сохраняют для следующего этапа, в результате чего возникает благоприятная ситуация для атакующего.

(2A-3) Способ улучшения иммунитета с использованием множества типов S-блоков

Пример конфигурации повышения трудности оценки ключей при выполнении атаки способом насыщения будет описан ниже. Таким образом, даже в случае, когда удовлетворяются описанные выше условия, модули обработки нелинейного преобразования отдельных F-функций, а именно S-блоки, выполнены таким образом, что данные перед и данные после выполнения раундовой функции не становятся эквивалентными в результате смещения данных.

Конкретный пример этого будет описан ниже со ссылкой на фиг.10. Конфигурация, иллюстрируемая на фиг.10, представляет, как и на фиг.9, конфигурацию вырезанного участка структуры Фейстеля или обобщенной структуры Фейстеля. На фиг.10 иллюстрируются F-функции 201 и 202, которые имеют одну и ту же входную строку (x) данных и выходную строку (y) данных и которые расположены вертикально рядом друг с другом.

Эти две F-функции 201 и 202 включают в себя модули расчета "исключающее ИЛИ", которые рассчитывают функцию "исключающее ИЛИ" с раундовыми ключами, модули обработки нелинейного преобразования и модули обработки линейного

преобразования. F-функции 201 и 202 выполнены с возможностью выполнения обработки 32-битных входных и выходных данных. Каждый модуль обработки нелинейного преобразования включает в себя четыре S-блока, и каждый из S-блоков принимает 8-битные входные данных и формирует 8-битные выходные данные.

Так же, как и на фиг.9, позиции A-J, представленные на фиг.10, обозначают следующие типы данных:

A: вход в предыдущую F-функцию 201;

B: выход из предыдущей F-функции 201;

C: вход в последующую F-функцию 202;

D: выход из последующей F-функции 202;

E: данные для операции "исключающее ИЛИ" с выходом B из предыдущей F-функции 201;

F: данные для операции "исключающее ИЛИ" с данными A;

G: результат операции "исключающее ИЛИ" по данным B и данным E;

H: результат операции "исключающее ИЛИ" по данным D и данным G;

I: раундовый ключ, вводимый в предыдущую F-функцию 201; и

J: раундовый ключ, вводимый в последующую F-функцию 202

В конфигурации, представленной на фиг.10, S-блоки модулей нелинейного преобразования, установленные в предыдущей F-функции 201 и в последующей F-функции 202, соответственно, выполнены с возможностью использования различных S-блоков [S1] и [S2].

Таким образом, S-блоки [S1], выполняющие обработку нелинейного преобразования в предыдущей F-функции 201, и S-блоки [S2], выполняющие обработку нелинейного преобразования в последующей F-функции 202, выполняют различные типы обработки нелинейного преобразования. В частности, S-блоки [S1] и [S2] выполняют обработку нелинейного преобразования, используя, например, различные таблицы нелинейного преобразования. S-блоки [S1] и [S2], могут не иметь одинаковый выход для одинаковых входных данных.

Здесь предполагается, что индивидуальные S-блоки S1 и S2 представляют собой два разных S-блока, удовлетворяющих следующим условиям.

Предполагая, что отдельные S-блоки S1 и S2 представляют собой S-блоки, которые выполняют обработку нелинейного преобразования с n-битными входными данными и n-битными выходными данными, удовлетворяются следующие условия:

(Условие 1)

Если все части S-битных данных, а именно 2^s данных x, последовательно вводят в любые из S-битных данных c,

выход S1 (x) первого S-блока [S1] для входных данных [x] и

выход S2 (x (ИИЛИ) c) S-блока [S2] для входных данных [x (ИИЛИ) c]

имеют, по меньшей мере, одно отличающееся значение.

Таким образом,

S1 (x) (ИИЛИ) S2 (x (ИИЛИ) c)

описанное выше уравнение не приводит к получению фиксированного значения.

Кроме того,

(Условие 2)

если все части S-битных данных, а именно 2^s данных x, последовательно вводят в любые S-битные данные c,

выход S1 (x) первого S-блока [S1] для входных данных [x] и

выход S2 (x (ИИЛИ) c) для S-блока [S2] для входных данных [x (ИИЛИ) c] имеет,

по меньшей мере, одно значение-дубликат. Таким образом,

S1 (x) (ИИЛИ) S2 (x (ИИЛИ) c)

в описанном выше уравнении никогда не появляются все 2^s.

Это представляет условие, состоящее в том, что если на фиг.10 предположить, что

данные A представляют собой [x], и

данные F представляют собой [c],

выход S1 (x) S-блока [S1] предыдущей F-функции 201 и

выход S2 (x (ИИЛИ) c) S-блока [S2] последующей F-функции 202

не будут одинаковыми, или не все результаты операции "исключающее ИЛИ" по выходам будут представлять собой разные значения.

Два S-блока [S1] и [S2], удовлетворяющие условиям, установлены, как представлено на фиг.10.

Таким образом, определенная F-функция использует модуль обработки нелинейного преобразования с использованием только S-блоков [S1], и следующая F-функция использует модуль обработки нелинейного преобразования, с использованием только S-блоков [S2]. Если имеются дополнительные раунды после этого, S-блоки [S1] и [S2] аналогично устанавливают в этом порядке в модулях обработки нелинейного преобразования отдельных F-функций.

При конфигурировании обработки нелинейного преобразования так, чтобы она была различной, то есть при установке множества различных S-блоков в вертикально соседних F-функциях с одинаковой строкой входных данных и строкой выходных данных, можно значительно уменьшить вероятность появления данных в выходной строке так, что они имеют сильную корреляцию с тем, что появляется в той же выходной строке перед выполнением раундовой функции.

Таким образом, использование S-блоков, удовлетворяющих описанному выше (условию 1), обеспечивает то, что даже в случае, когда входы в два S-блока имеют фиксированное различие, результаты выполнения операции "исключающее ИЛИ" по их выходам имеют разные значения, по меньшей мере, однажды, что гарантирует, таким образом, что выходы не будут полностью смещены друг от друга.

Кроме того, использование S-блоков, удовлетворяющих описанному выше (условию 2), обеспечивает то, что даже в случае, когда входы в эти два S-блока имеют фиксированное различие, результаты выполнения операции "исключающее ИЛИ" по их выходам имеют значения - дубликаты, по меньшей мере, однажды, что, таким образом, ухудшает характеристику, которую можно использовать при организации атак. Поэтому при размещении двух S-блоков описанным выше образом, можно свести к минимуму благоприятные условия для атакующих, которые организуют атаки способом насыщения. Таким образом, можно ожидать улучшения иммунитета против атак.

Таким образом, даже в случае, когда значения, вводимые в S-блоки отдельных F-функций 201 и 202, равны на фиг.10, то есть даже если

A [0] (ИИЛИ) I [0]=C [0] (ИИЛИ) J [0],

значения, выводимые из S-блоков в отдельных F-функциях, а именно

S1([0](ИИЛИ)1[0]) и

S2 (C [0] (ИИЛИ) J [0])

не будут одинаковыми во всех случаях. В результате выходные данные B и D F-функции отдельных F-функций 201 и 202 не будут полностью одинаковыми. При этом не возникает такая ситуация, как описанная со ссылкой на фиг.9, когда справедливо

E=H (ИИЛИ) Δ,

и можно устранить ситуацию, когда вероятность данных в одной строке данных до и после выполнения раундовой функции (F-функции), имеет фиксированное различие.

При размещении множества различных S-блоков, которые выполняют разные типы обработки нелинейного преобразования в вертикально соседних F-функциях, имеющих одинаковую строку входных данных и строку выходных данных, трудность организации атак способом насыщения можно значительно повысить, и можно улучшить иммунитет против атак.

(Система 1 разработки)

В описанной выше со ссылкой на фиг.10 конфигурации принимают во внимание только взаимосвязь между двумя F-функциями и получают условие, состоящее в том, что различные S-блоки устанавливают в этих двух F-функциях. Аналогичная идея применима к трем или больше F-функциям. Например, можно ожидать улучшение иммунитета против атак способом насыщения при размещении множества различных S-блоков в F-функциях, как показано на фиг.11.

На фиг.11 иллюстрируется конфигурация вырезанного участка структуры Фейстеля или обобщенной структуры Фейстеля. На фиг.11 иллюстрируются три F-функции 211-213, которые имеют одинаковую строку (x) входных данных и строку (y) выходных данных и которые вертикально расположены рядом друг с другом.

S-блоки [S1] установлены в модуле обработки нелинейного преобразования F-функции 211;

S-блоки [S2] установлены в модуле обработки нелинейного преобразования в F-функции 212; и

S-блоки [S3] установлены в модуле обработки нелинейного преобразования в F-функции 213.

Следует отметить, что S1≠S2≠S3.

Таким образом, условия, требуемые для множества S-блоков, должны представлять собой:

(Условие 1)

Заданы наборы S1, S2, …, Sk для k (k>2) S-блоков и пары двух разных S-блоков S_i и S_j (i≠j). Если все возможные 2s данных x будут заданы как входные данные для любого c,

S_i(x) и

S_j (x (ИИЛИ) c)

выходные данные этих S-блоков не полностью противоречат друг другу, и S-блоки выводят разные значения, по меньшей мере, один раз. Таким образом,

результаты выполнения операции "исключающее ИЛИ" по S_i (x) и S_j (x (ИИЛИ) c) не приводят к получению фиксированного значения.

Кроме того,

(Условие 2)

заданы наборы S1, S2, …, Sk для k (k>2) S-блоков, и пары двух различньгх S-блоков S_i и S_j (i≠j). Если все возможные 2^n данных x будут заданы как входы для любого c,

Si (x)и

S_j (x (ИИЛИ) c)

выходы этих S-блоков не имеют все 2^n значений, появляющихся однажды. Таким образом, выходы S-блоков имеют, по меньшей мере, одно значение-дубликат.

При установке наборов S1, S2, …, Sk S-блоков, удовлетворяющих этим условиям и при размещении этих F-функций во множестве F-функций, которые имеют одинаковые

строки (x) входных данных и строки (y) выходных данных и которые расположены последовательно вертикально рядом друг с другом, вероятность противоречия данных, появляющихся в выходной строке данным, появляющимся в той же выходной строке перед выполнением раундовой функции, может быть значительно уменьшена. В результате, можно существенно повысить трудность атак способом насыщения, и можно улучшить иммунитет против атак.

(Система 2 разработки)

Учитывая практические варианты воплощения, даже в случае, когда множество типов S-блоков включены в отдельные F-функции, может быть желательным, чтобы одинаковая комбинация S-блоков была включена в каждую F-функцию.

Таким образом, в случае, когда преобразование данных, соответствующее F-функциям, выполняют, используя, например, аппаратные или программные средства, если та же комбинация S-блоков будет включена в каждую F-функцию, аппаратные или программные средства, используемые как F-функции, могут быть выполнены как одинаковые аппаратные или программные средства, и преобразование данных на основе F-функции можно выполнять в каждом раунде только при изменении входных и выходных данных в каждом раунде, в соответствии с необходимостью.

Конкретный пример будет описан со ссылкой на фиг.12. Как и на фиг.10, на фиг.12 иллюстрируется конфигурация вырезанного участка структуры Фейстеля или обобщенной структуры Фейстеля. На фиг.12 иллюстрируются F-функции 221 и 222, которые имеют одинаковые строку (x) входных данных и строку (y) выходных данных и которые расположены вертикально рядом друг с другом.

Четыре S-блока, включенные в предыдущую F-функцию 221, расположены в порядке S1, S2, S1 и S2 сверху вниз, и S-блоки, включенные в последующую F-функцию 222 в следующем раунде, расположены в порядке S2, SI, S2 и S1 сверху вниз.

Следует отметить, что $S1 \neq S2$.

При такой установке, если воплощена конфигурация, позволяющая выполнять два S1 и два S2 параллельно друг другу, F-функции 221 и 222 могут быть выполнены с использованием такой конфигурации. В соответствии с этим, можно снизить стоимость воплощения, и устройство может быть выполнено более компактным.

Также в конфигурации, представленной на фиг.12, обработка нелинейного преобразования применяется для соответствующих строк битов в отдельных F-функциях 221 и 222 в следующем порядке:

от S1 до S2, или

от S2 до S1,

и обработка соответствующих битовых данных (например, в каждом модуле размером байт) представляет собой обработку, аналогичную обработке, описанной со ссылкой на фиг.10. В результате аналогичный эффект может быть достигнут, то есть вероятность противоречия данных, появляющихся в выходной строке с данными, появляющимися в той же выходной строке, перед выполнением раундовой функции, может быть значительно уменьшена. В результате трудность организации атак способом насыщения может быть значительно повышена, и можно улучшить иммунитет против этих атак.

Другой конкретный пример представлен на фиг.13. Как и на фиг.11, на фиг.13 иллюстрируется конфигурация вырезанного участка структуры Фейстеля или обобщенной структуры Фейстеля. На фиг.13 иллюстрируются три F-функции 231-233, которые имеют одинаковую строку (x) входных данных и строку (y) входных данных и которые вертикально расположены рядом друг с другом.

Четыре S-блока, включенные в начальную F-функцию 231, расположены в порядке S1, S2, S3 и S4 сверху вниз. Четыре S-блока, включенные в среднюю F-функцию 232 в следующем раунде, расположены в порядке S2, S3, S4 и S1 сверху вниз. Затем четыре S-блока, включенные в среднюю F-функцию 233 в следующем раунде, расположены в порядке S3, S4, S1 и S2 сверху вниз.

Следует отметить, что S1≠S2≠S3≠S4.

При такой установке, если будет воплощена конфигурация, позволяющая выполнять S1-S4 параллельно друг другу, все F-функции 231 и 233 могут быть выполнены с использованием этой конфигурации. В соответствии с этим стоимость воплощения может быть уменьшена, и устройство может быть выполнено более компактным.

Также в конфигурации, представленной на фиг.13, обработка нелинейного преобразования, применяемая для соответствующих битовых строк в отдельных F-функциях 231-233, представлена в следующем порядке:

S1, S2, S3, S4, S1, S2, …,

и обработка соответствующих битовых данных (например, каждого модуля размером байт) представляет собой обработку, аналогичную обработке, описанной со ссылкой на фиг.10 или фиг.11. В результате, может быть достигнут аналогичный эффект, то есть вероятность появления противоречащих данных в выходной строке с данными, появляющимися в той же выходной строке, перед выполнением раундовой функции может быть значительно уменьшена. В результате можно существенно повысить трудность организации атак способом насыщения и можно улучшить иммунитет против атак.

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшают путем смешения двух или больше S-блоков различных типов в блочном шифре с использованием S-блоков

Далее будет описана конфигурация, в которой улучшен иммунитет против алгебраических атак (атак РЯС) путем смешения различных типов S-блоков в блочном шифре с использованием S-блоков.

(2B-1) Краткое описание алгебраических атак (атак РЯС)

Вначале будут описаны алгебраические атаки (атаки РЯС), которые известны как атаки на блочные шифры. Алгебраические атаки (атаки РЯС) на блочные шифры представляют собой атаки, в которых используется алгебраическое представление S-блоков. Когда входные и выходные данные S-блоков представлены как алгебраические выражения, можно вывести множество выражений. Сложность расчетов для организации атаки меняется в зависимости от максимального порядка выражений и количества членов, включенных в выражение.

В качестве одного примера алгебраической атаки (атаки РЯС) существует способ с использованием Булевых выражений. Например, заданный блочный шифр включает в себя множество S-блоков, каждый из которых принимает 8-битные входные данные и формирует 8-битные выходные данные, и пусть входные биты и выходные биты каждого из S-блока с 8-битным входом/выходом будут выражены следующим образом:

вход X: (x1, x2, x3, x4, x5, x6, x7, x8), и

выход Y: (y1, y2, y3, y4, y5, y6, y7, y8),

затем оценивают количество выражений, которые выражают с использованием квадратных или более низкого порядка Булевых выражений.

Более конкретно, оценивают количество многочленов, включающих в себя

квадратные или более низкого порядка члены, такие как

(1, xi, yi, xixj, yiyj, xiyj),

которые можно получить путем выражения описанных выше входных X и выходных Y данных, как Булевых выражений.

В случае, когда выражения более низкого порядка, такие как выражения, в которых максимальный порядок представляет собой второй порядок, выбирают из всех Булевых выражений, представленных таким образом, если выбирают большее количество независимых выражений, и если количество членов мало, ситуация становится преимущественной для атакующего. Таким образом, если выбирают большее количество независимых выражений, в которых максимальный порядок ограничен вторым порядком или тому подобное, и если количество членов мало, ситуация является преимущественной для атакующего и представляет плохой иммунитет против атак.

Кроме того, помимо Булевых выражений, если алгебраические выражения низкого порядка можно вывести в пределах поля определения, такого как поле GF (2^8) расширения, аналогичную методику можно использовать для того, чтобы легко организовать алгебраическую атаку (атаку РЯС), что означает плохой иммунитет против атак.

(2B-2) Проблема использования S-блоков одного типа

Далее будет описана проблема конфигурации, в которой используют S-блоки только одного типа в блочном шифре с использованием S-блоков, то есть проблема, состоящая в том, что повышается вероятность осуществления на практике алгебраической атаки (атаки РЯС).

Существуют три следующих представительных типа s-битных S-блоков, которые выполняют нелинейное преобразование с использованием n-битного входа и с получением n-битного выхода:

тип 1: S-блок с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ в пределах поля расширения GF (2s);

тип 2: S-блоки, генерируемые путем комбинирования множества S-блоков, входные и выходные биты которых меньше, чем s битов, например четыре бита; и

тип 3: S-блок, выбираемый случайно.

Эти три типа являются репрезентативными.

В частности, тип 1 и тип 2 представляют собой S-блоки, которые часто используют из-за низкой стоимости аппаратных средств (H/W, AC).

Ниже, для каждого из описанных выше типов 1-3, будет описана проблема, связанная с конфигурацией, в которой используются S-блоки только одного типа, то есть проблема, состоящая в том, что повышается вероятность осуществления алгебраической атаки (атаки РЯС).

<Проблема типа 1>

Проблема типа 1, то есть проблема S-блоков, с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ для GF (2s), будет описана ниже.

Например, в случае, когда S-блок с использованием инверсной карты по GF (2^8) представлен как Булево выражение, известно, что представление включает в себя приблизительно двадцать независимых квадратных выражений и приблизительно восемьдесят членов. Аналогичное простое соответствие может быть найдено в случае показательной функции. Кроме того, аналогичное соответствие, как ожидается, должно быть действительным для S-блоков, определенных не только по GF (2^8), но

также и по GF(2^S).

Используя такие представления в виде многочленов, можно оценить сложность вычислений для алгебраической атаки (атаки РЯС). Во время разработки шифра необходимо использовать достаточное количество S-блоков для того, чтобы обеспечить достаточно высокую сложность расчетов для обеспечения безопасности. Кроме того, для S-блока, с использованием инверсной карты по GF (2^S) может быть получено алгебраическое представление, такое как XY=1 по GF (2^S), и могут быть выведены многочлены низкого порядка. Известно, что существуют способы атаки с использованием этих характеристик. Аналогичный результат может быть применимым к показательной функции.

Поскольку два типа алгебраических характеристик используются в шифре с применением S-блоков с использованием инверсной карты или показательной функции по GF (2^S), конструкция шифра должна учитывать эти два типа алгебраических характеристик.

Следует отметить, что то же относится к S-блокам, генерируемым, путем добавления аффинного преобразования перед/после инверсной карты и показательной функции.

<Проблема типа 2>

Далее будет описана проблема типа 2, то есть проблема S-блока, генерируемого путем комбинирования множества меньших (например, 4-битных) S-блоков.

Рассмотрим 8-битный S-блок, генерируемый путем комбинирования множества малых S-блоков, с учетом, например, 4-битных входных данных и формирования 4-битных выходных данных. Известно, что 8-битный S-блок может быть выполнен с использованием от трех до пяти 4-битных S-блоков. Для организации алгебраической атаки (атаки РЯС), квадратные или более низкого порядка Булевы многочлены выводят из входных и выходных битов 4-битных S-блоков. Поскольку сумма входных и выходных битов составляет восемь, известно, что существует приблизительно двадцать или что-то около этого независимых выражений, представленных такими многочленами низкого порядка. Поэтому используя такую характеристику, можно организовать атаку. Такая тенденция применима к случаю, когда для того, чтобы сделать S-блок с большим размером входных/выходных данных, конфигурируют более крупный S-блок с использованием меньших S-блоков.

Однако в качестве преимущества такого способа, поскольку вероятность простого алгебраического соотношения, существующего в поле GF (2^S), например, как в случае использования S-блоков с использованием инверсной карты по GF (2^8), значительно уменьшается, известно, что сложность вычислений для организации атаки увеличивается. Это означает, что по сравнению с предыдущими S-блоками, существуют как преимущества, так и недостатки в отношении алгебраических атак (атак РЯС).

<Проблема типа 3>

Далее будет описана проблема типа 3, то есть проблема случайно выбранного S-блока. Ожидается, что S-блоки, выбранные случайно, не имеют алгебраически слабых характеристик, как было описано выше, и, следовательно, эти S-блоки, как ожидается, обеспечивают высокую защиту против алгебраических атак (атак РЯС). Однако стоимость аппаратного воплощения очень высока. Поэтому предпочтительно не делать все S-блоки S-блоками, выбранными случайно.

Конфигурация (2B-3), в которой иммунитет повышается в результате

использования множества типов S-блоков, имеющих различные алгебраические характеристики

С учетом описанных выше проблем ниже приведено описание конфигурации, в которой иммунитет против как алгебраических атак (атак РЯС) с использованием Булевых многочленов, так и против алгебраических атак (атак РЯС) с использованием поля GF (2s) может быть улучшен благодаря применению двух или больше типов S-блоков, имеющих различные алгебраические характеристики, и в которых эффективность воплощения в виде аппаратных средств (A/C) улучшена даже в большей степени, чем в случае, когда все S-блоки представляют собой S-блоки, выбранные случайно.

Как было описано выше, существуют следующие три репрезентативных типа s-битных S-блоков, которые выполняют нелинейное преобразование, учитывая s-битные входные данные, и формируют s-битные выходные данные:

тип 1: S-блок с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^{p}$ в поле GF (2s) расширения;

тип 2: S-блок, генерируемый путем комбинирования множества малых t-битных S-блоков (где t<s); и

тип 3: S-блок, выбираемый случайно.

Эти три типа представляют собой представительные типы.

В настоящем варианте воплощения смешанное использование этих разных типов S-блоков позволяет реализовать конфигурацию, в которой улучшается иммунитет против алгебраических атак (атак РЯС) и повышается эффективность воплощения в виде аппаратных средств (A/C). Таким образом, смешанное использование двух или больше S-блоков разных типов в блочном шифре с использованием S-блоков позволяет реализовать конфигурацию, в которой улучшается иммунитет против алгебраических атак (атак РЯС). Следует отметить, что необходимо только, чтобы конфигурация обработки шифрования, в которой применим настоящий вариант воплощения, должна представлять собой конфигурацию обработки шифрования, имеющую S-блоки, которые выполняют обработку нелинейного преобразования. Например, конфигурация обработки шифрования применима к любой следующей конфигурации обработки шифрования, которые были описаны выше, то есть:

(a) структура СПП (Сеть подстановок-перестановок);

(b) Структура Фейстеля; и

(c) обобщенная структура Фейстеля.

В этом примере обработки S-блоки, используемые как модули обработки нелинейного преобразования, включенные в раундовые функции для выполнения обработки преобразования данных, представляют собой любую одну из следующих параметров установок (a)-(d):

(a) конфигурация, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 2;

(b) конфигурация, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 3;

(c) конфигурация, в которой некоторые из S-блоков представляют собой S-блоки типа 2, и остальные S-блоки представляют собой S-блоки типа 3; и

(d) конфигурация, в которой некоторые из S-блоков представляют собой S-блоки типа 1, некоторые из остальных S-блоков представляют собой S-блоки типа 2, и остальные S-блоки представляют собой S-блоки типа 3.

Например, в случае описанной выше установки (a),

необходимо рассмотреть теоретический шифр, в котором половина S-блоков, используемая как модули обработки нелинейного преобразования, включена в раундовые функции для выполнения обработки преобразования данных типа 1, то есть S-блоки с использованием инверсной карты по GF (2^8), и остальные S-блоки исключены. Оценивают сложность расчета при организации алгебраической атаки (атаки РЯС), используя Булевы выражения по теоретическому шифру. Если будет оценена достаточная вычислительная сложность, остальную часть S-блоков конфигурируют как S-блоки типа 2, то есть 8-битные S-блоки, генерируемые путем комбинирования множества малых 4-битных S-блоков.

При использовании конфигурации обработки шифрования, имеющей смесь типа 1 и типа 2, как в описанной выше позиции (a), если достаточный иммунитет может быть обеспечен на основе оценки вычислительной сложности по GF (2^8), блочный шифр, в котором общий иммунитет улучшен, может быть сгенерирован по сравнению со случаем, в котором используют по отдельности S-блоки каждого типа.

Этот эффект не ограничивается описанной выше установкой. Аналогично в любом из описанных выше случаев (a)-(d) конфигурация обработки шифрования установлена таким образом, что даже ограниченное количество S-блоков позволяет обеспечить достаточно сильный иммунитет против алгебраических атак (атак РЯС), и остальные S-блоки могут быть определены с учетом эффективности воплощения или тому подобное.

Конкретные примеры конфигурации обработки шифрования, включающие в себя компоновки различных типов S-блоков, как и в описанных выше случаях от (a) до (d), будут описаны со ссылкой на фиг.14-18. Каждый из примеров, показанных на фиг.14-18, представляет конфигурацию обработки шифрования, имеющую модули выполнения раундовой функции в шести раундах. Каждый модуль выполнения раундовой функции включает в себя модуль обработки нелинейного преобразования, включающий в себя множество S-блоков, и модуль обработки линейного преобразования.

На фиг.14 иллюстрируется примерный блочный шифр СПП с шестью раундами, и каждый раунд включает в себя десять S-блоков. Блочный шифр СПП выполняет преобразование данных, включающее в себя уровень нелинейного преобразования (S уровень) и уровень линейного преобразования (P уровень) в каждом раунде. Десять S-блоков, включенных в каждый раунд, занимают соответствующие части входных данных, на которые разделяют входные данные, в качестве входных данных выполняют обработку нелинейного преобразования и выводят части данных после нелинейного преобразования в уровень линейного преобразования (P уровень). Данные после линейного преобразования выводят в модуль выполнения следующей раундовой функции. Выходные данные модуля выполнения раундовой функции на последнем этапе представляют собой зашифрованный текст.

В отдельных модулях 301-306 выполнения раундовой функции, показанных на схеме, $[S_1]$ и $[S_2]$ обозначают S-блок типа 1 и S-блок типа 2, соответственно, которые представляют собой S-блоки, используемые как различные типы модулей обработки нелинейного преобразования, как было описано выше.

В примере, показанном на фиг.14, представлен пример конфигурации, в которой,

тип 1: S-блок с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ через поле GF (2^S) расширения,

S-блоки этого типа 1 размещены в модулях 301-303 выполнения раундовой функции, в предыдущих трех раундах, и

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S блоки

S-блоки этого типа 2 расположены в модулях 301-303 раундовой функции, в последних трех раундах.

В конфигурации, показанной на фиг.14, обработку нелинейного преобразования в предыдущих раундах выполняют как обработку с применением S-блоков типа 1, и обработку нелинейного преобразования в последних раундах выполняют как обработку с применением S-блоков типа 2. Алгебраическую атаку (атаку РЯС) обычно организуют на основе предположения, что все S-блоки представляют собой блоки одного типа. В случае, когда S-блоки разных типов смешаны так, как описано выше, атака, а именно анализ, становится трудной. В результате, реализуется конфигурация обработки шифрования с сильным иммунитетом против криптоанализа, такого как алгебраические атаки (атаки РЯС).

На фиг.15 представлен примерный блочный шифр СПП с шестью раундами, и каждый раунд включает в себя десять S-блоков, как на фиг.14.

В примере, показанном на фиг.15, представлен пример конфигурации, в которой,

тип 1: S-блок, в котором используется инверсная карта: $Y=X^{-1}$ или показательная функция $Y=X^p$ по полю GF (2s) расширения,

S-блоки этого типа 1 размещены в модулях 321, 323 и 325 выполнения раундовой функции, в первом, третьем и пятом раундах с нечетными номерами, и

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S-блоки.

S-блоки такого типа 2 размещены в модулях 322, 324 и 326 выполнения раундовой функции, во втором, четвертом и шестом раундах или в раундах с четными номерами.

В конфигурации, показанной на фиг.15, обработку нелинейного преобразования в раундах с нечетными номерами выполняют как обработку применения S-блоков типа 1, и обработку нелинейного преобразования в раундах с четными номерами выполняют как обработку с применением S-блоков типа 2. Как и в конфигурации, показанной на фиг.14, смесь разных типов S-блоков установлена в настоящей конфигурации. В соответствии с этим реализуется конфигурация обработки шифрования с сильным иммунитетом против криптоанализа, такого как алгебраические атаки (атаки РЯС).

На фиг.16 иллюстрируется примерный блочный шифр СПП с шестью раундами, и каждый раунд включает в себя десять S-блоков, как показано на фиг.14 и фиг.15.

В примере, показанном на фиг.16, представлен пример конфигурации, в которой

тип 1: S-блок, с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ по полю GF (2^S) расширения,

S-блоки этого типа 1 размещены как половина S-блоков в модулях 341-346 выполнения раундовой функции во всех раундах, и

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S-блоки.

S-блоки такого типа 2 размещены, как остальная половина S-блоков. Таким образом, пять S-блоков типа 1 [S_1] и пять S-блоков типа 2 [S_2] включены в каждый из модулей 341-346 выполнения раундовой функции.

Данные, вводимые в каждый из модулей 341-346 выполнения раундовой функции, разделяют на десять равных частей, и эти десять частей вводят в соответствующие S-блоки. Среди десяти равных частей d_1-d_{10} данных первую половину частей данных d_1-d_5 вводят в S-блоки типа 1 и выполняют обработку нелинейного преобразования с

применением S-блоков типа 1; и вторую половину частей данных d_6-d_{10} вводят в S-блоки типа 2, и обработку нелинейного преобразования выполняют с использованием S-блоков типа 2.

Как и в конфигурации, показанной на фиг.14 и фиг.15, смесь различных типов S-блоков также установлена в конфигурации, показанной на фиг.16. В соответствии с этим, реализуется конфигурация обработки шифрования с сильным иммунитетом против криптоанализа, такого как алгебраические атаки (атаки РЯС).

На фиг.17 иллюстрируется примерный блочный шифр СПП с шестью раундами, и каждый раунд включает в себя десять S-блоков, как показано на фиг.14-16.

Как и в примере, показанном на фиг.16, пример, показанный на фиг.17, представляет собой пример конфигурации, в которой,

тип 1: S-блок с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ по полю GF (2^S) расширения,

S-блоки этого типа 1 размещены как половина S-блоков в модулях 361-366 выполнения раундовой функции, во всех раундах, и

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S-блоки.

S-блоки такого типа 2 размещены как остальная половина S-блоков. Таким образом, пять S-блоков типа 1 [S_1] и пять S-блоков типа 2 [S_2] включены в каждую из частей 361-366 выполнения раундовой функции

Данные, вводимые в каждый из модулей 361-366 выполнения раундовой функции, разделяют на десять равных частей, и эти десять частей вводят в соответствующие S-блоки. Среди десяти равных частей d_1-d_{10} данных части с нечетными номерами данных d_1, d_3, d_5, d_7 и d_9 вводят в S-блоки типа 1, и выполняют обработку нелинейного преобразования с применением S-блоков типа 1; и части с четными номерами данных d_2, d_4, d_6, d_8 и d_{10} вводят в S-блоки типа 2, и выполняют обработку нелинейного преобразования, с применением S-блоков типа 2.

Как и в конфигурации, показанной на фиг.14-16, смесь различных типов S-блоков также устанавливают в конфигурации по фиг.17. В соответствии с этим, реализуют конфигурацию обработки шифрования с сильным иммунитетом против криптоанализа, такого как алгебраические атаки (атаки РЯС).

В конфигурациях, показанных на фиг.16 и фиг.17, S-блоки, предназначенные для выполнения параллельно друг другу в каждом раунде, включают в себя пять S-блоков типа 1 и пять S-блоков типа 2. Это свойство является общим для всех раундов. Поэтому, если будет воплощена конфигурация, позволяющая выполнять пять S-блоков типа 1 и пять S-блоков типа 2 параллельно друг другу, такую конфигурацию можно многократно применять для выполнения раундовых функций во всех раундах, в результате чего получают преимущество, состоящее в снижении стоимости воплощения и уменьшении размера.

Пример, в котором S-блоки разных типов размещены в отдельных модулях 381-386 выполнения раундовых функций в структуре Фейстеля, показан на фиг.18.

Пример, иллюстрируемый на фиг.18, представляет собой пример конфигурации, в которой,

тип 1: S-блок, с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^p$ по полю GF (2s) расширения,

S-блоки этого типа 1 размещены как половина S-блоков в модулях 381-386 выполнения раундовой функции, во всех раундах, и

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S-блоки

S-блоки такого типа 2 размещены в остальной половине S-блоков. Таким образом, два S-блока типа 1 [S_1] и два S-блока типа 2 [S2] включены в каждый из модулей 381-386 выполнения раундовой функции.

Данные, вводимые в каждый из модулей 381-386 выполнения раундовой функции, разделяют на четыре равные части, и эти четыре части подают в соответствующие S-блоки. Из четырех равных частей d_1-d_4 данных части с нечетными номерами данных d_1 и d_3 вводят в S-блоки типа 1 и выполняют обработку нелинейного преобразования с применением S-блоков типа 1; и части с четными номерами данных d_2 и d_4 вводят в S-блоки типа 2 и выполняют обработку нелинейного преобразования с применением S-блоков типа 2.

Как и в конфигурации, показанной на фиг.14-17, смесь различных типов S-блоков также установлена в конфигурации, показанной на фиг.18. В соответствии с этим реализуется конфигурация обработки шифрования с сильным иммунитетом против криптоанализа, такого как алгебраические атаки (атаки РЯС).

Следует отметить, что в примерах, показанных на фиг.14-18, иллюстрируются примеры конфигурации, с использованием смеси двух различных типов S-блоков, S-блоков типа 1 и типа 2. В качестве конфигурации со смесью различных типов S-блоков возможны конфигурации со следующими различными типами смесей, как было описано выше:

(a) конфигурация, в которой некоторые из S-блоков представляют собой тип 1, и остальные S-блоки представляют собой тип 2;

(b) конфигурация, в которой некоторые из S-блоков представляют собой тип 1, и остальные S-блоки представляют собой тип 3;

(c) конфигурация, в которой некоторые из S-блоков представляют собой тип 2, и остальные S-блоки представляют собой тип 3; и

(d) конфигурация, в которой некоторые из S-блоков представляют собой тип 1, некоторые из остальных S-блоков представляют собой тип 2, и остальные S-блоки представляют собой тип 3.

В любом случае реализуется улучшение иммунитета против алгебраических атак (атак РЯС).

(2C) Конфигурация, в которой описанные выше подходы (2A) и (2B) одновременно реализуют в шифре Фейстеля или в шифре Фейстеля обобщенного типа, с использованием S-блоков

Ниже будет приведено описание примера конфигурации для одновременной реализации описанных выше случаев (2A) и (2B) в шифре Фейстеля или обобщенном шифре Фейстеля, с использованием S-блоков, то есть:

(2A) Конфигурация, в которой иммунитет против атак способом насыщения улучшен, благодаря размещению двух или больше S-блоков разного типа в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков; и

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшен путем смешивания двух или больше различных типа S-блоков в блочном шифре, с использованием S-блоков.

Конфигурация в описанном выше случае (2A) выполнена для улучшения иммунитета против атак способом насыщения, путем применения двух или больше типов S-блоков в структуре Фейстеля или обобщенной структуре Фейстеля. Конфигурация в описанном выше случае (2 B) выполнена для улучшения иммунитета

против алгебраических атак (атак РЯС), с использованием двух или больше типов S-блоков в любом блочном шифре, с использованием S-блоков.

Эти конфигурации в случаях (2A) и (2B) могут быть скомбинированы и реализованы как одна конфигурация. Таким образом, становится возможным сконфигурировать блочный шифр, имеющий структуру Фейстеля или обобщенную структуру Фейстеля, с использованием двух или больше типов S-блоков, удовлетворяющих характеристикам, необходимым в случаях (2A) и (2B), таким образом, одновременно улучшают иммунитет против обоих типов атак.

В частности, для каждого из S-блоков [S1], [S2], [S3], [S4], … которые выполняют различные типы обработки нелинейного преобразования, применяемых в каждой из конфигураций по фиг.10-13, описанных в разделе

Конфигурация, в которой иммунитет против атак способом насыщения улучшают путем размещения двух или больше различных типов S-блоков в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков,

различные типы S-блоков, описанные в разделе

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшают путем смешения двух или больше S-блоков различных типов в блочном шифре, с использованием S-блоков

то есть,

тип 1: S-блок, с использованием инверсной карты: $Y=X^{-1}$ или показательной функции $Y=X^{p}$ по полю GF (2^{s}) расширения;

тип 2: S-блок, генерируемый путем комбинирования множества малых S-блоков, таких как 4-битные S-блоки; и

тип 3: S-блок, выбранный случайно,

эти три типа установлены во взаимной связи.

Например, в конфигурации, показанной на фиг.10,

путем установки S-блоков [S1] и S-блоков [S2] в качестве S-блоков различных типов, описанных в разделе (2B),

реализуется конфигурация с сильным иммунитетом против атак способом насыщения и алгебраических атак (атак РЯС).

То же относится к конфигурациям, представленным на фиг.11-13.

Путем установки S-блоков [S1], [S2], … как разных типов S-блоков, описанных в разделе (2B),

реализуется конфигурация с сильным иммунитетом против атак способом насыщения и алгебраических атак (атаки РЯС).

[3. Пример конфигурации устройства обработки шифрования]

Наконец, пример конфигурации модуля 700 с микросхемой, используемого в качестве устройства обработки шифрования, которое выполняет обработку шифрования в соответствии с описанными выше вариантами воплощения, показан на фиг.19. Описанная выше обработка может выполняться различными устройствами обработки информации, такими как PC (ПК, персональный компьютер), карта с микросхемой, блок считывания-записи и т.п. Модуль 700 с микросхемой, показанный на фиг.19, может быть выполнен как любое из этих устройств.

ЦПУ (Центральное процессорное устройство) 701, показанное на фиг.19, представляет собой процессор, который управляет началом и прекращением обработки шифрования, передачей/приемом данных и передачей данных среди отдельных элементов, и выполняет различные другие программы. Запоминающее устройство 702 включает в себя, например, ROM (ПЗУ, постоянное запоминающее

устройство), в котором содержится программа, выполняемая CPU (ЦПУ) 701, или фиксированные данные, такие как операционные параметры, и RAM (ОЗУ, Оперативное запоминающее устройство), используемое как область сохранения или рабочая область для программы, выполняемой при обработке, выполняемой ЦПУ 701, и параметров, изменяющихся, по мере необходимости, в ходе обработки выполняемой программы. Кроме того, запоминающее устройство 702 можно использовать как область накопления, например, для данных ключей, необходимых для обработки шифрования, таблицы преобразования (таблицы перестановки), применяемой при обработке шифрования, и данных, подаваемых в матрицу преобразования. Следует отметить, что область сохранения данных, предпочтительно, выполнена как запоминающее устройство со структурой, защищенной от несанкционированного вмешательства.

Процессор 703 шифрования выполняет обработку шифрования и обработку дешифрования, в соответствии с алгоритмом обработки блочного шифра с общим ключом, с применением одной из следующих структур в соответствующих конфигурациях, то есть, например, описанных выше различных конфигурациях обработки шифрования:

(a) структура СПП (Сеть подстановок-перестановок);

(b) Структура Фейстеля; и

(c) Обобщенная структура Фейстеля.

Кроме того, процессор 703 шифрования включает в себя S-блоки, используемые как модули обработки нелинейного преобразования, имеющие конфигурации, соответствующие описанным выше вариантам воплощения, то есть конфигурации, соответствующие любой одной из следующих конфигураций:

(2A) Конфигурация, в которой два или больше разных типов S-блоков размещены в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков.

(2B) Конфигурация, в которой два или больше разных типов S-блоков смешаны в блочном шифре с использованием S-блоков.

(2C) Конфигурация, в которой описанные выше случаи (2A) и (2B) одновременно реализованы в шифре Фейстеля или в шифре Фейстеля обобщенного типа с использованием S-блоков.

Следует отметить, что хотя пример, в котором средство обработки шифрования представляет собой отдельный модуль, был описан выше, вместо предоставления такого независимого модуля обработки шифрования, например, программа обработки шифрования может быть сохранена в ПЗУ, и ЦПУ 701 может быть выполнен с возможностью считывания и выполнения этой программы, сохраненной в ПЗУ.

Генератор 704 случайных чисел выполняет обработку генерирования случайных чисел, которые необходимы, например, для генерирования ключей, используемых для обработки шифрования.

Передатчик/приемник 705 представляет собой процессор передачи данных, который выполняет обмен данными с внешними устройствами. Например, передатчик/приемник 705 выполняет обмен данными с модулем с микросхемой, таким как блок считывания/записи, и выполняет вывод зашифрованного текста, сгенерированного в модуле с микросхемой, или принимает данные из устройства, такого как внешний блок считывания/записи, в качестве входных данных.

Модуль 700 с микросхемой имеет компоновку S-блоков, используемых как модули обработки нелинейного преобразования, в соответствии с описанными выше

вариантами воплощения. В результате, модуль 700 с микросхемой имеет одну из этих конфигураций:

(2A) Конфигурация, в которой иммунитет против атак способом насыщения улучшен путем размещения двух или больше разных типов S-блоков в шифре Фейстеля или шифре Фейстеля обобщенного типа с использованием S-блоков.

(2B) Конфигурация, в которой иммунитет против алгебраических атак (атак РЯС) улучшен путем смешения двух или больше разных типов S-блоков в блочном шифре с использованием S-блоков.

(2C) Конфигурация, в которой описанные выше случаи (2A) и (2B) одновременно реализованы в шифре Фейстеля или в шифре Фейстеля обобщенного типа, с использованием S-блоков.

В соответствии с этим, модуль 700 с микросхемой имеет конфигурацию, в которой улучшен иммунитет против атак способом насыщения и алгебраических атак (атак РЯС).

Настоящее изобретение было подробно описано выше со ссылкой на конкретные варианты воплощения. Следует, однако, понимать, что модификации или альтернативы вариантов воплощения могут быть выполнены специалистом в данной области техники без выхода за пределы объема настоящего изобретения. Таким образом, настоящее изобретение было раскрыто на примерах, и его раскрытие не следует рассматривать как ограничительное. Для определения объема настоящего изобретения следует обращаться к приложенной формуле изобретения.

Следует отметить, что последовательность обработки, описанная в описании, может быть выполнена с помощью аппаратных средств, программных средств или их комбинаций. В случае, когда последовательность обработки выполняется с помощью программных средств, программа, в которой записана последовательность обработки, может быть установлена в запоминающем устройстве, в компьютере, который встроен в специализированные аппаратные средства, и может быть выполнена. В качестве альтернативы, программа может быть установлена в компьютере общего назначения, который позволяет выполнять различную обработку, и может быть в нем выполнена.

Например, программа может быть заранее записана на жестком диске или в ПЗУ (Постоянное запоминающее устройство), используемом как носитель записи. В качестве альтернативы, программа может быть сохранена (записана) временно или постоянно на съемном носителе записи, таком как гибкий диск, CD-ROM (ПЗУ-К, постоянное запоминающее устройство на компакт-диске), на диске MO (магнитооптический диск), DVD (ЦУД, цифровой универсальный диск), магнитный диск или полупроводниковое запоминающее устройство. Такой съемный носитель записи может поставляться в виде так называемого пакета программного обеспечения.

Следует отметить, что, помимо установки программы с описанного выше съемного носителя записи в компьютер, программа может быть передана по беспроводному каналу передачи данных с сайта загрузки в компьютер, или может быть передана по проводам в компьютер через сеть, такую как LAN (ЛВС - локальная вычислительная сеть) или сеть Интернет, обеспечивающая возможность для компьютера принимать программу, передаваемую описанным выше способом, и устанавливать эту программу на внутренний носитель записи, такой как жесткий диск.

Следует отметить, что различная обработка, описанная в описании, не обязательно должна быть выполнена последовательно в описанном порядке, и может быть выполнена параллельно или индивидуально в соответствии с характеристиками или

потребностями обработки устройства, которое выполняет эту обработку. Кроме того, система в настоящем описании относится к логической компоновке множества устройств и не ограничена компоновкой, в которой устройства, имеющие индивидуальную конфигурацию, содержатся в одном корпусе.

Промышленная применимость

Как было описано выше, в соответствии с конфигурацией варианта воплощения настоящего изобретения, в устройстве обработки шифрования, которое выполняет обработку блочного шифра с общим ключом, S-блоки, используемые как модули обработки нелинейного преобразования, установленные в модулях выполнения раундовой функции, выполнены как, по меньшей мере, два разных типа S-блоков. При использовании такой конфигурации может быть улучшен иммунитет против атак способом насыщения. Кроме того, в соответствии с конфигурацией варианта воплощения настоящего изобретения, в котором типы S-блоков представляют собой смесь различных типов, иммунитет против алгебраических атак (атак РЯС) может быть улучшен, в результате чего реализуется устройство обработки шифрования с высокой надежностью.

Формула изобретения

1. Устройство обработки шифрования, отличающееся тем, что содержит:

модуль обработки шифрования, который выполняет обработку преобразования данных с использованием F-функции, включающей в себя S-блоки, в качестве раундовой функции по отдельным строкам данных, полученным путем разделения входных данных на число, большее чем или равное трем,

в котором модуль обработки шифрования выполнен так, что он включает в себя различные типы S-блоков в F-функциях, которые имеют одинаковую строку входных данных и строку выходных данных, и которые вертикально расположены рядом друг с другом.

2. Устройство обработки шифрования по п.1, отличающееся тем, что:

модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования с использованием структуры Фейстеля, в которой количество строк данных (количество разделений) равно двум, или обобщенной структуры Фейстеля, в которой количество строк данных (количество разделений) равно двум или больше, и

модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования с использованием F-функций, применяемых как модули выполнения раундовых функций, которые имеют одинаковую строку входных данных и строку выходных данных и которые вертикально расположены рядом друг с другом, в которых обработка нелинейного преобразования, выполняемая F-функциями, установлена как разные S-блоки, которые выполняют разные типы обработки нелинейного преобразования.

3. Устройство обработки шифрования по п.2, отличающееся тем, что каждая из F-функций, используемых как модули выполнения раундовых функций, включает в себя множество S-блоков, которые выполняют обработку нелинейного преобразования соответствующих частей данных, на которые разделены данные, предназначенные для обработки, и

множество S-блоков включает в себя, по меньшей мере, два разных типа S-блоков.

4. Устройство обработки шифрования по п.2, отличающееся тем, что каждая из F-функций, используемых как модули выполнения раундовых функций, включает в себя множество S-блоков, которые выполняют обработку нелинейного преобразования

соответствующих частей данных, на которые разделены данные, предназначенные для обработки, и

обработку нелинейного преобразования, выполняемую с последовательным приращением частей данных в F-функциях, используемых как модули выполнения раундовых функций, которые имеют одинаковую строку входных данных и строку выходных данных и которые расположены вертикально рядом друг с другом, выполненную с возможностью установки в качестве других S-блоков, которые выполняют другие типы обработки нелинейного преобразования.

5. Устройство обработки шифрования по п.2, отличающееся тем, что типы S-блоков и количество отдельных S-блоков, включенных в каждую из F-функций, используемых как модули выполнения раундовых функций, имеют одинаковую установку среди отдельных F-функций.

6. Устройство обработки шифрования по п.1, отличающееся тем, что модуль обработки шифрования выполнен с возможностью использования

в качестве разных s-битовых входных/выходных S-блоков, предназначенных для применения при обработке нелинейного преобразования,

(1) типа 1: S-блок, использующий инверсную карту: $Y=X^{-1}$ или показательную функцию $Y=X^q$ по полю GF (2s) расширения;

(2) типа 2: S-блок, генерируемый путем комбинирования множества малых t-битных S-блоков, где $t<s$; и

(3) типа 3: S-блок, выбранный случайным образом,

по меньшей мере, двух разных типов S-блоков среди описанных выше трех типов S-блоков (1)-(3).

7. Устройство обработки шифрования по п.6, отличающееся тем, что модуль обработки шифрования имеет

в отношении S-блоков, применяемых для выполнения раундовой функции,

(a) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 2;

(b) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, и остальные S-блоки представляют собой S-блоки типа 3;

(c) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 2, и остальные S-блоки представляют собой S-блоки типа 3; и

(d) конфигурацию, в которой некоторые из S-блоков представляют собой S-блоки типа 1, некоторые из остальных S-блоков представляют собой S-блоки типа 2, и остальные из S-блоков представляют собой S-блоки типа 3,

любую одну из описанных выше конфигураций (a)-(d).

8. Устройство обработки шифрования по п.6, отличающееся тем, что:

модуль обработки шифрования включает в себя в модулях выполнения раундовых функций множество S-блоков, которые выполняют обработку нелинейного преобразования для соответствующих частей данных, на которые разделены данные, предназначенные для обработки, и

модуль обработки шифрования выполнен с возможностью выполнения обработки с использованием S-блоков одного типа в одном раунде и S-блоков других типов на основе от раунда к раунду.

9. Устройство обработки шифрования по п.6, отличающееся тем, что:

модуль обработки шифрования включает в себя в модулях выполнения раундовых функций множество S-блоков, которые выполняют обработку нелинейного преобразования соответствующих частей данных, на которые разделены данные,

предназначенные для обработки, и

модуль обработки шифрования выполнен с возможностью использования различных типов S-блоков в одном раунде.

10. Устройство обработки шифрования по п.9, отличающееся тем, что типы S-блоков и количество отдельных S-блоков, включенных в каждую из частей выполнения раундовых функций, имеют одинаковые установки среди отдельных F-функций.

11. Устройство обработки шифрования по любому из пп.1-10, отличающееся тем, что:

модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования в соответствии с криптографией с общим ключом.

12. Устройство обработки шифрования по любому из пп.1-10, отличающееся тем, что:

модуль обработки шифрования выполнен с возможностью выполнения обработки шифрования в соответствии с криптографией блочного шифра с общим ключом.

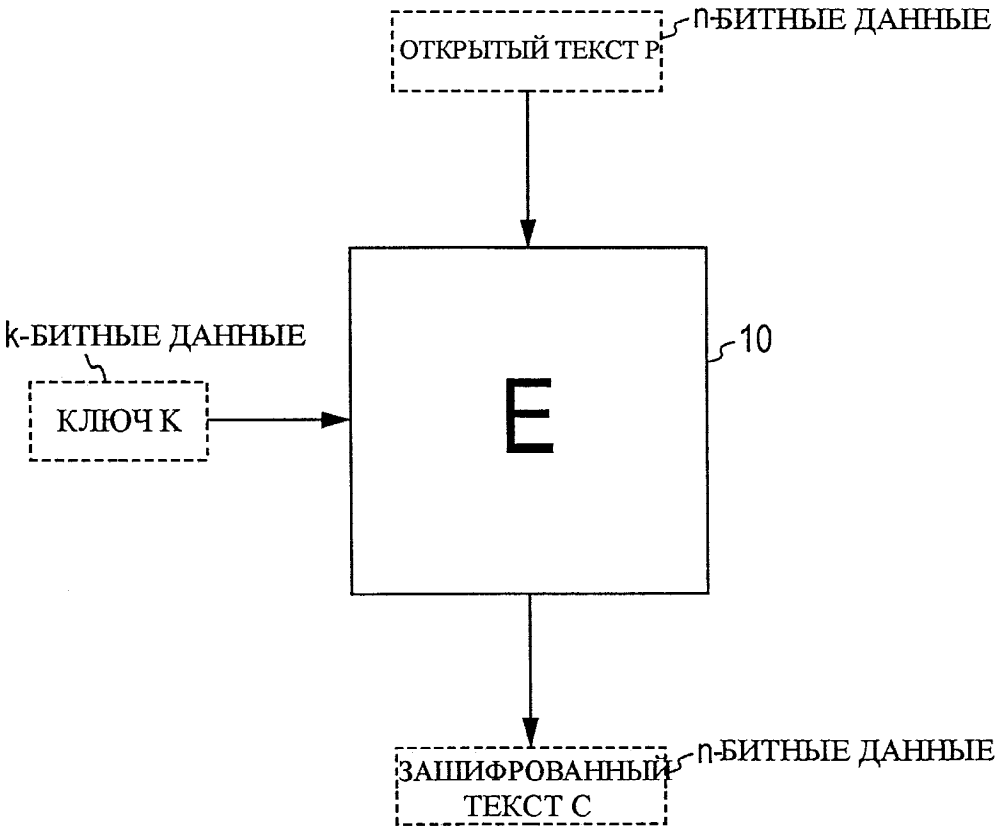
13. Способ обработки шифрования, состоящий в выполнении обработки шифрования в устройстве обработки шифрования, отличающийся тем, что содержит:

этап обработки шифрования, состоящий в выполнении в части обработки шифрования обработки преобразования данных с использованием F-функции, включающей в себя S-блоки, в качестве раундовой функции по отдельным строкам данных, полученным путем разделения входных данных на число, большее чем или равное трем,

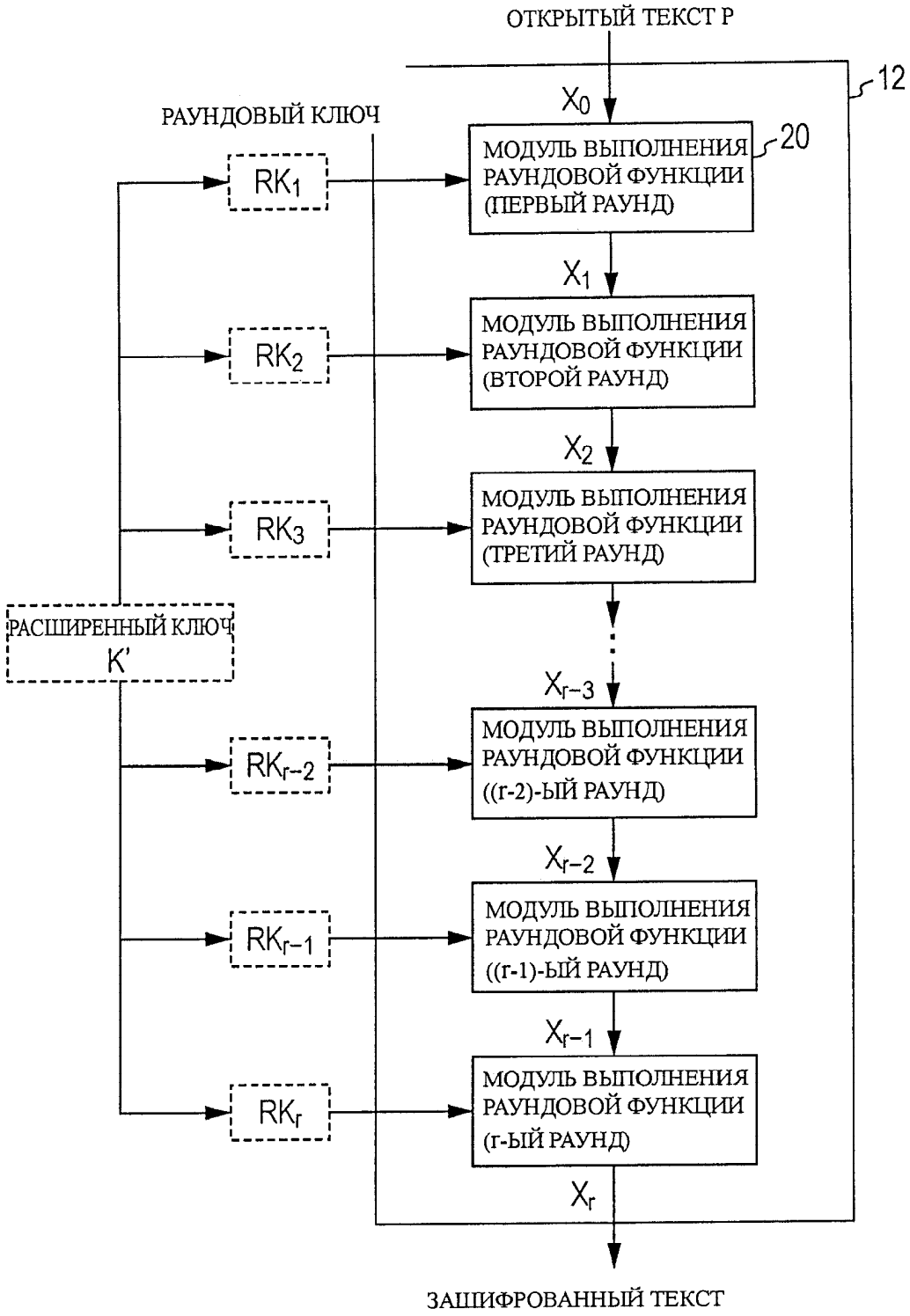
в котором этап обработки шифрования представляет собой этап выполнения обработки преобразования данных с использованием разных типов S-блоков в F-функциях, которые имеют одинаковую строку входных данных и строку выходных данных и которые вертикально расположены рядом друг с другом.

14. Способ обработки шифрования по п.13, отличающийся тем, что:

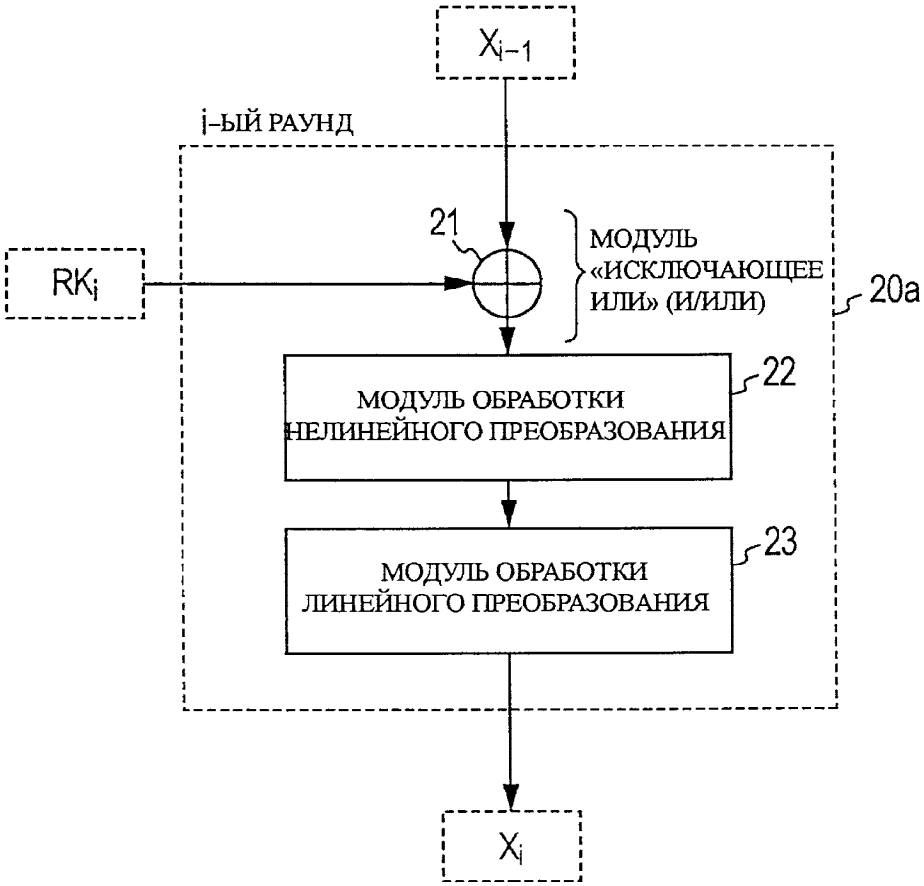
на этапе обработки шифрования выполняют обработку шифрования в соответствии с криптографией с общим ключом или криптографией блочного шифра с общим ключом.


ОТКРЫТЫЙ ТЕКСТ P
n-БИТНЫЕ ДАННЫЕ
k-БИТНЫЕ ДАННЫЕ
КЛЮЧ K
E
10
ЗАШИФРОВАННЫЙ ТЕКСТ C
n-БИТНЫЕ ДАННЫЕ

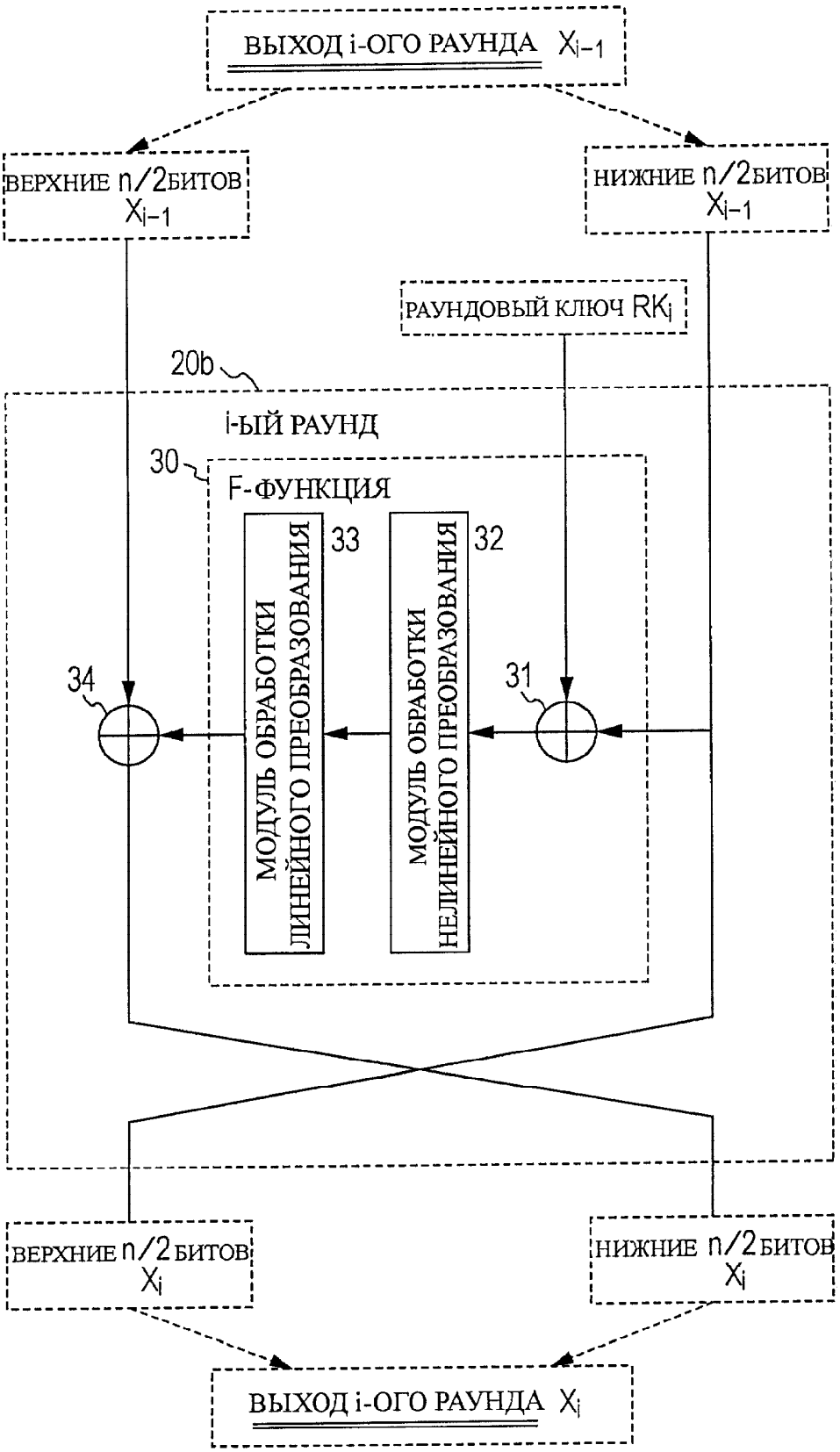
Фиг. 1


ОТКРЫТЫЙ ТЕКСТ P
12
X_0
РАУНДОВЫЙ КЛЮЧ
RK_1
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ (ПЕРВЫЙ РАУНД)
20
X_1
RK_2
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ (ВТОРОЙ РАУНД)
X_2
RK_3
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ (ТРЕТИЙ РАУНД)
РАСШИРЕННЫЙ КЛЮЧ K'
X_{r-3}
RK_{r-2}
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ ((r-2)-ЫЙ РАУНД)
X_{r-2}
RK_{r-1}
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ ((r-1)-ЫЙ РАУНД)
X_{r-1}
RK_r
МОДУЛЬ ВЫПОЛНЕНИЯ РАУНДОВОЙ ФУНКЦИИ (r-ЫЙ РАУНД)
X_r
ЗАШИФРОВАННЫЙ ТЕКСТ

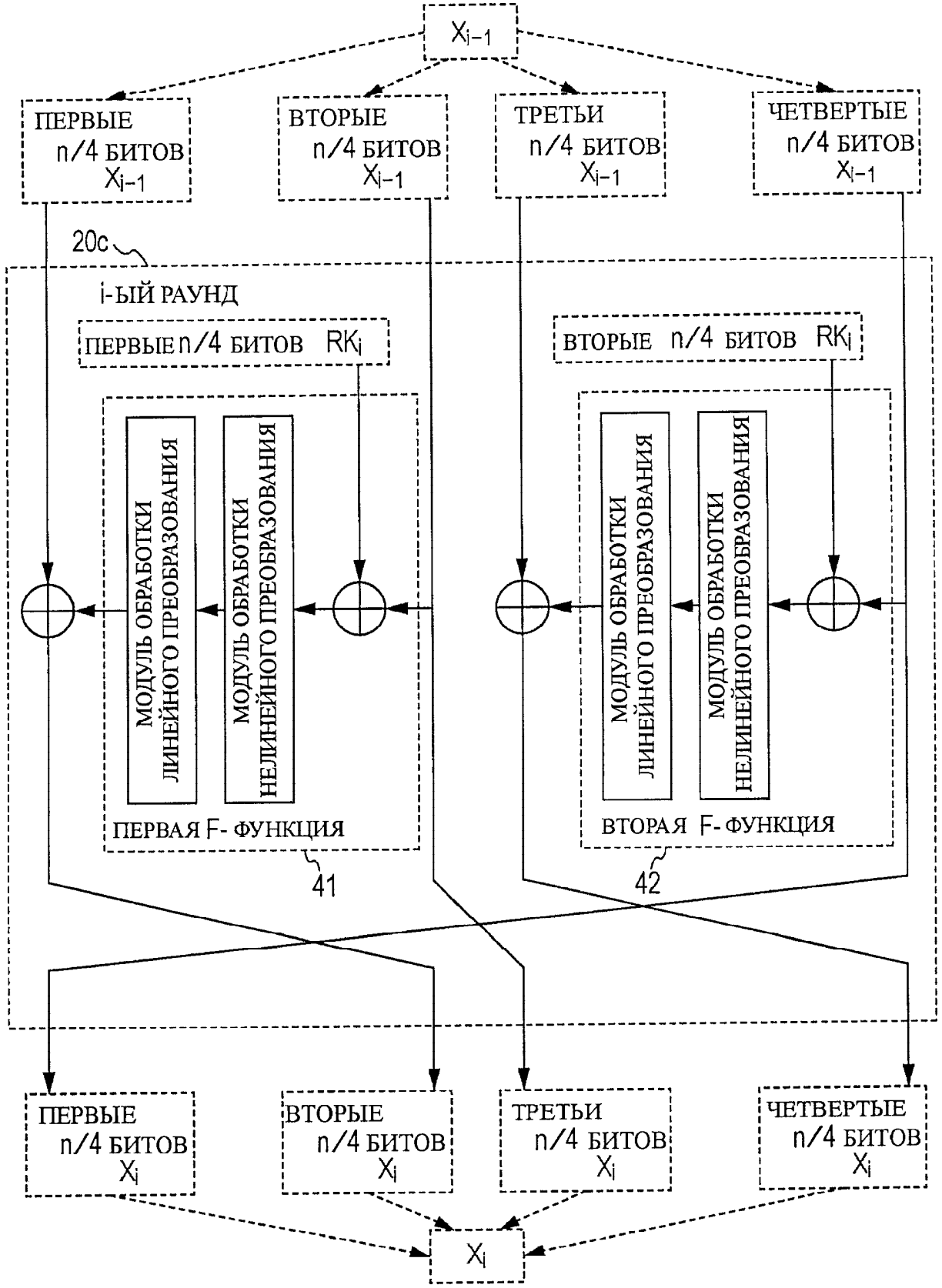
Фиг. 3


X_{i-1}
i-ЫЙ РАУНД
21
МОДУЛЬ «ИСКЛЮЧАЮЩЕЕ ИЛИ» (И/ИЛИ)
RK_i
20a
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
22
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
23
X_i

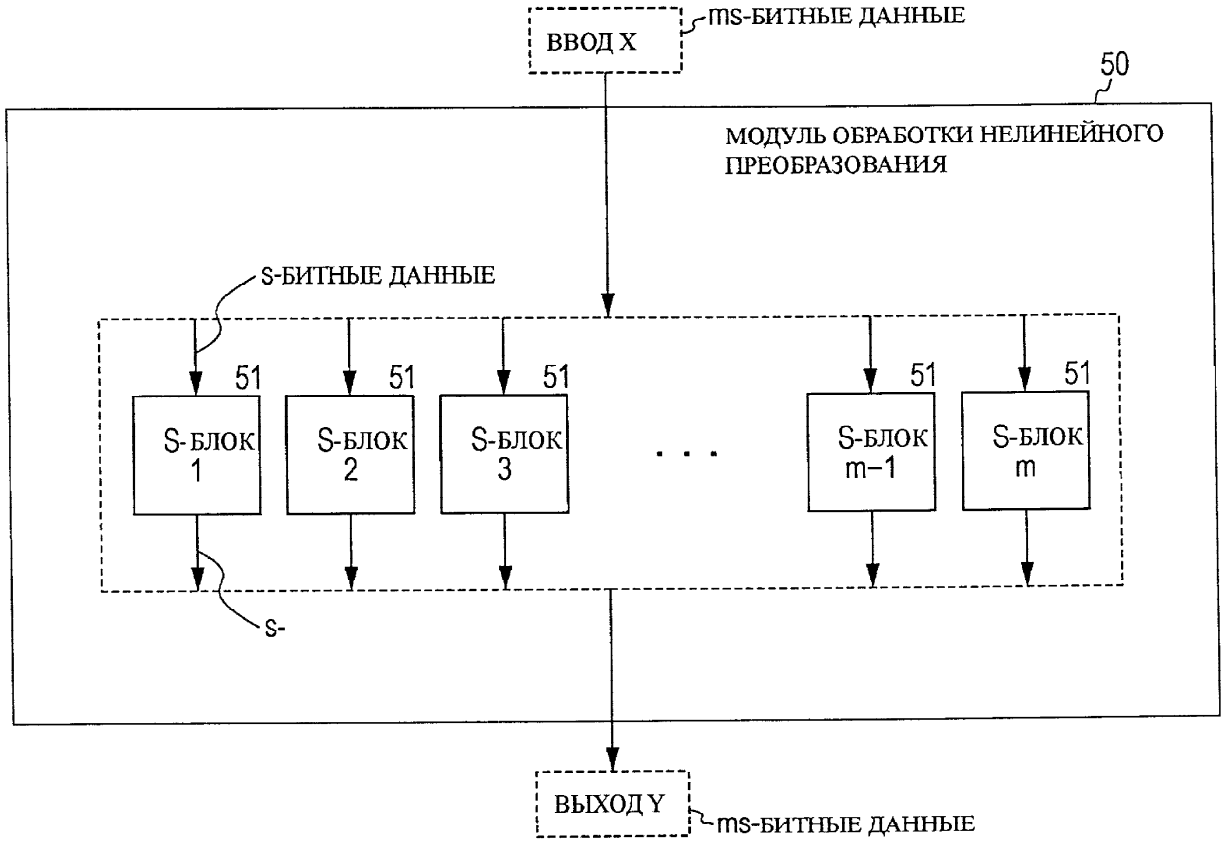
Фиг. 4


ВЫХОД i-ОГО РАУНДА X_{i-1}
ВЕРХНИЕ n/2БИТОВ X_{i-1}
НИЖНИЕ n/2БИТОВ X_{i-1}
РАУНДОВЫЙ КЛЮЧ RK_i
20b
i-ЫЙ РАУНД
30
F-ФУНКЦИЯ
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
33
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
32
34
31
ВЕРХНИЕ n/2 БИТОВ X_i
НИЖНИЕ n/2 БИТОВ X_i
ВЫХОД i-ОГО РАУНДА X_i

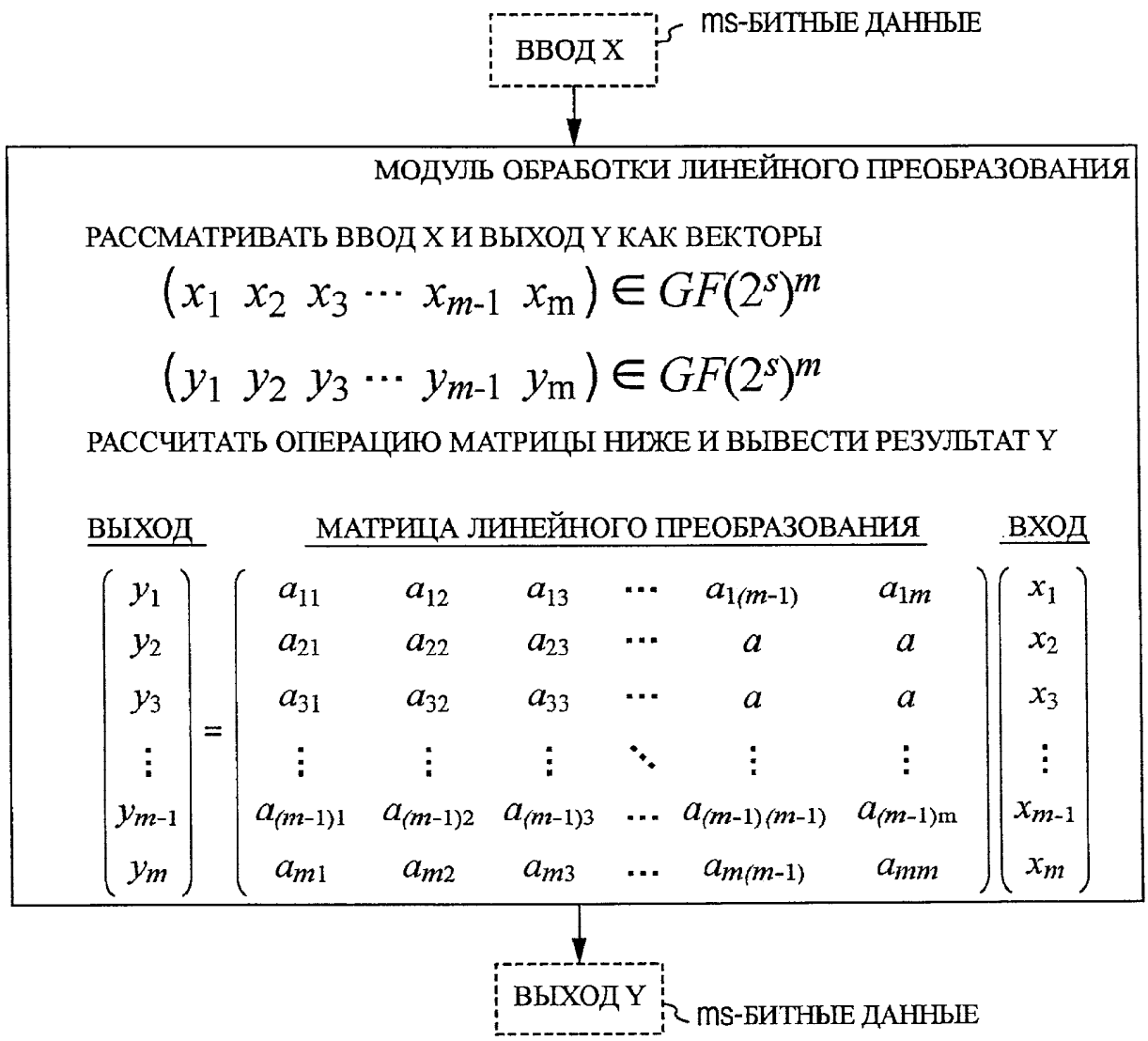
Фиг. 5


Xi-1
ПЕРВЫЕ n/4 БИТОВ Xi-1
ВТОРЫЕ n/4 БИТОВ Xi-1
ТРЕТЬИ n/4 БИТОВ Xi-1
ЧЕТВЕРТЫЕ n/4 БИТОВ Xi-1
20c
i-ЫЙ РАУНД
ПЕРВЫЕ n/4 БИТОВ RKi
ВТОРЫЕ n/4 БИТОВ RKi
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
ПЕРВАЯ F- ФУНКЦИЯ
41
ВТОРАЯ F- ФУНКЦИЯ
42
ПЕРВЫЕ n/4 БИТОВ Xi
ВТОРЫЕ n/4 БИТОВ Xi
ТРЕТЬИ n/4 БИТОВ Xi
ЧЕТВЕРТЫЕ n/4 БИТОВ Xi
Xi

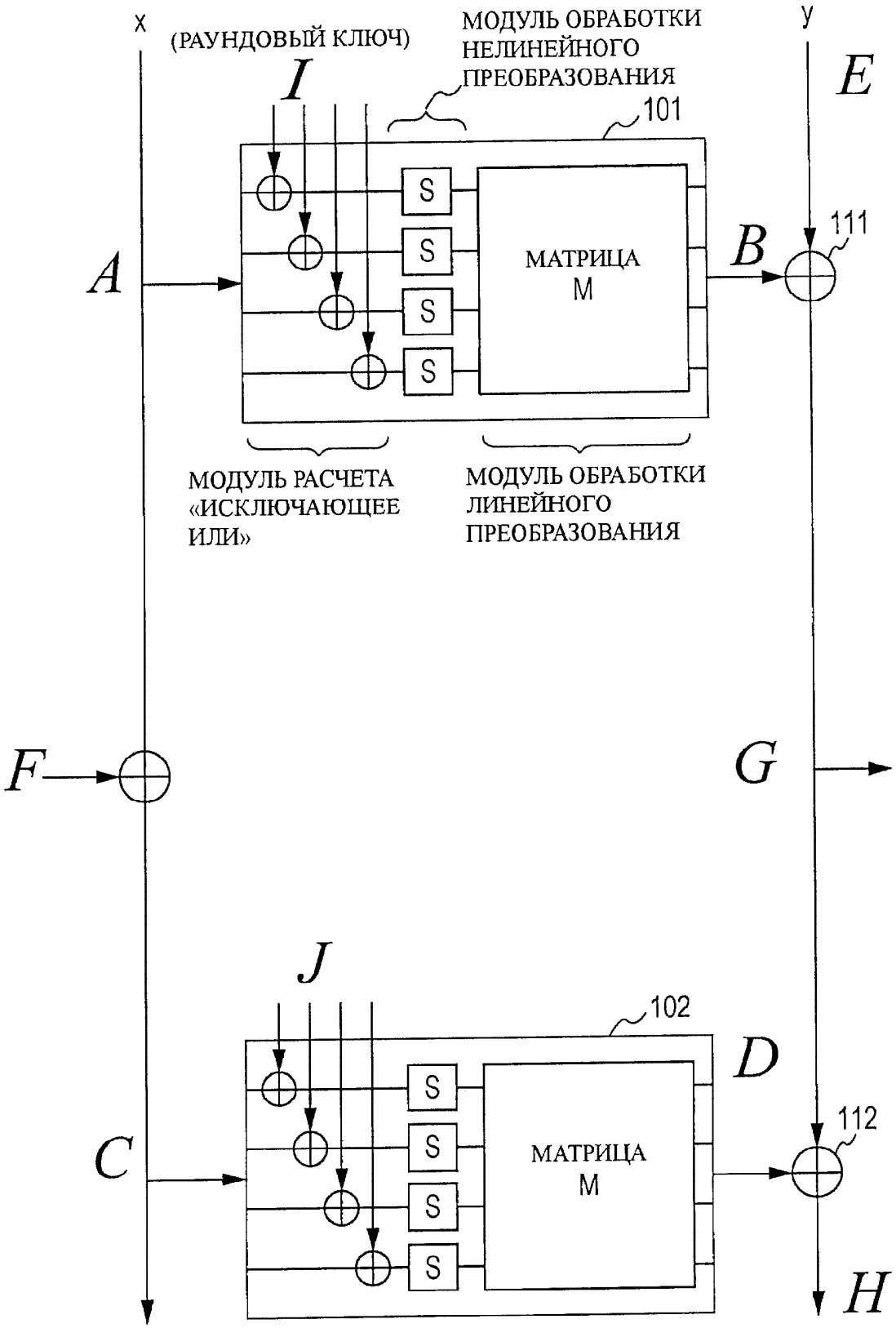
Фиг. 6


ВВОД X
ms-БИТНЫЕ ДАННЫЕ
50
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
s-БИТНЫЕ ДАННЫЕ
51
S-БЛОК 1
S-БЛОК 2
S-БЛОК 3
. . .
S-БЛОК m−1
S-БЛОК m
s-
ВЫХОД Y
ms-БИТНЫЕ ДАННЫЕ

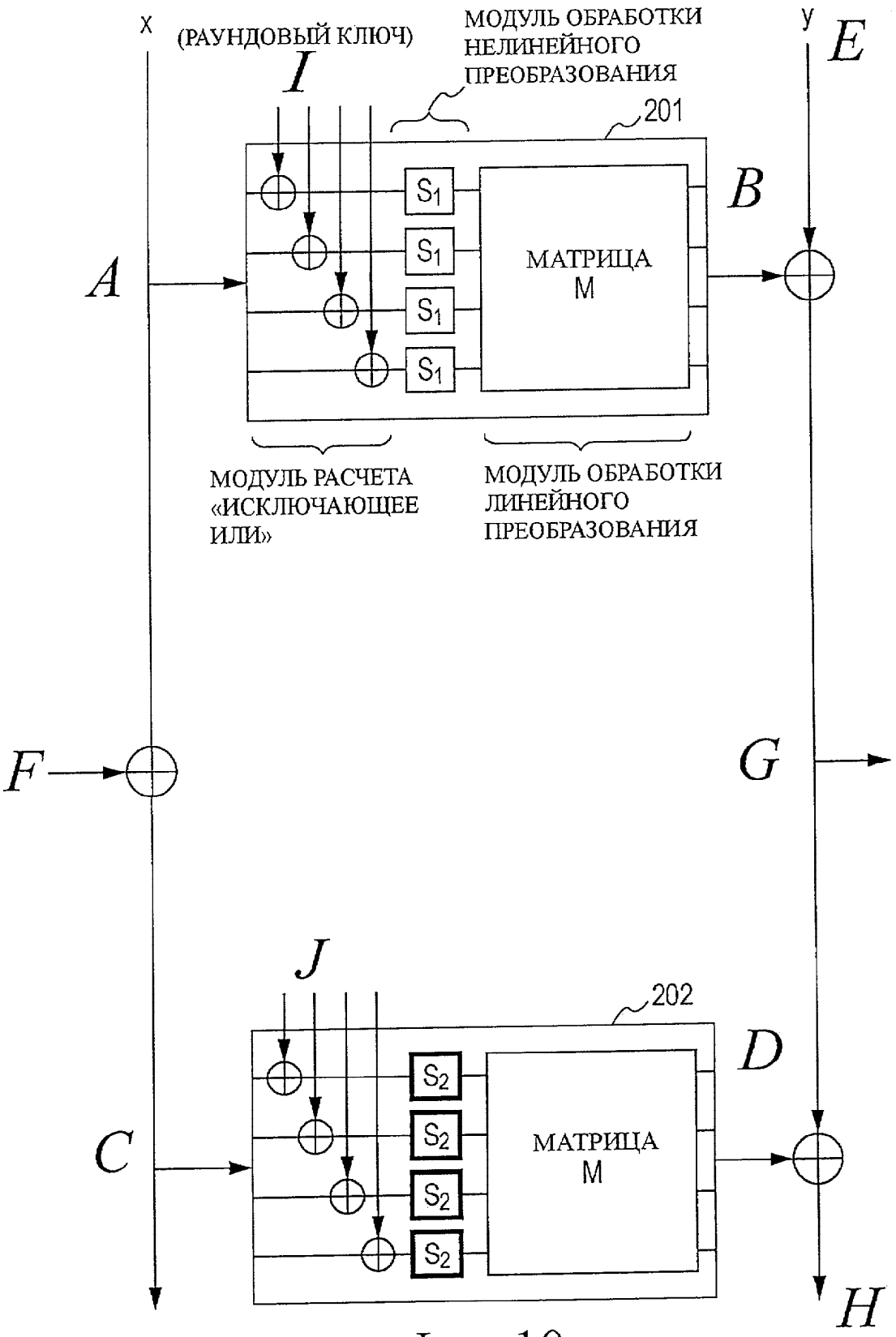
Фиг. 7


ВВОД X
ms-БИТНЫЕ ДАННЫЕ
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
РАССМАТРИВАТЬ ВВОД X И ВЫХОД Y КАК ВЕКТОРЫ
$(x_1\ x_2\ x_3 \cdots x_{m\text{-}1}\ x_m) \in GF(2^s)^m$
$(y_1\ y_2\ y_3 \cdots y_{m\text{-}1}\ y_m) \in GF(2^s)^m$
РАССЧИТАТЬ ОПЕРАЦИЮ МАТРИЦЫ НИЖЕ И ВЫВЕСТИ РЕЗУЛЬТАТ Y
ВЫХОД
МАТРИЦА ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
ВХОД
$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{m\text{-}1} \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1(m\text{-}1)} & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a & a \\ a_{31} & a_{32} & a_{33} & \cdots & a & a \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{(m\text{-}1)1} & a_{(m\text{-}1)2} & a_{(m\text{-}1)3} & \cdots & a_{(m\text{-}1)(m\text{-}1)} & a_{(m\text{-}1)m} \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{m(m\text{-}1)} & a_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{m\text{-}1} \\ x_m \end{pmatrix}$
ВЫХОД Y
ms-БИТНЫЕ ДАННЫЕ

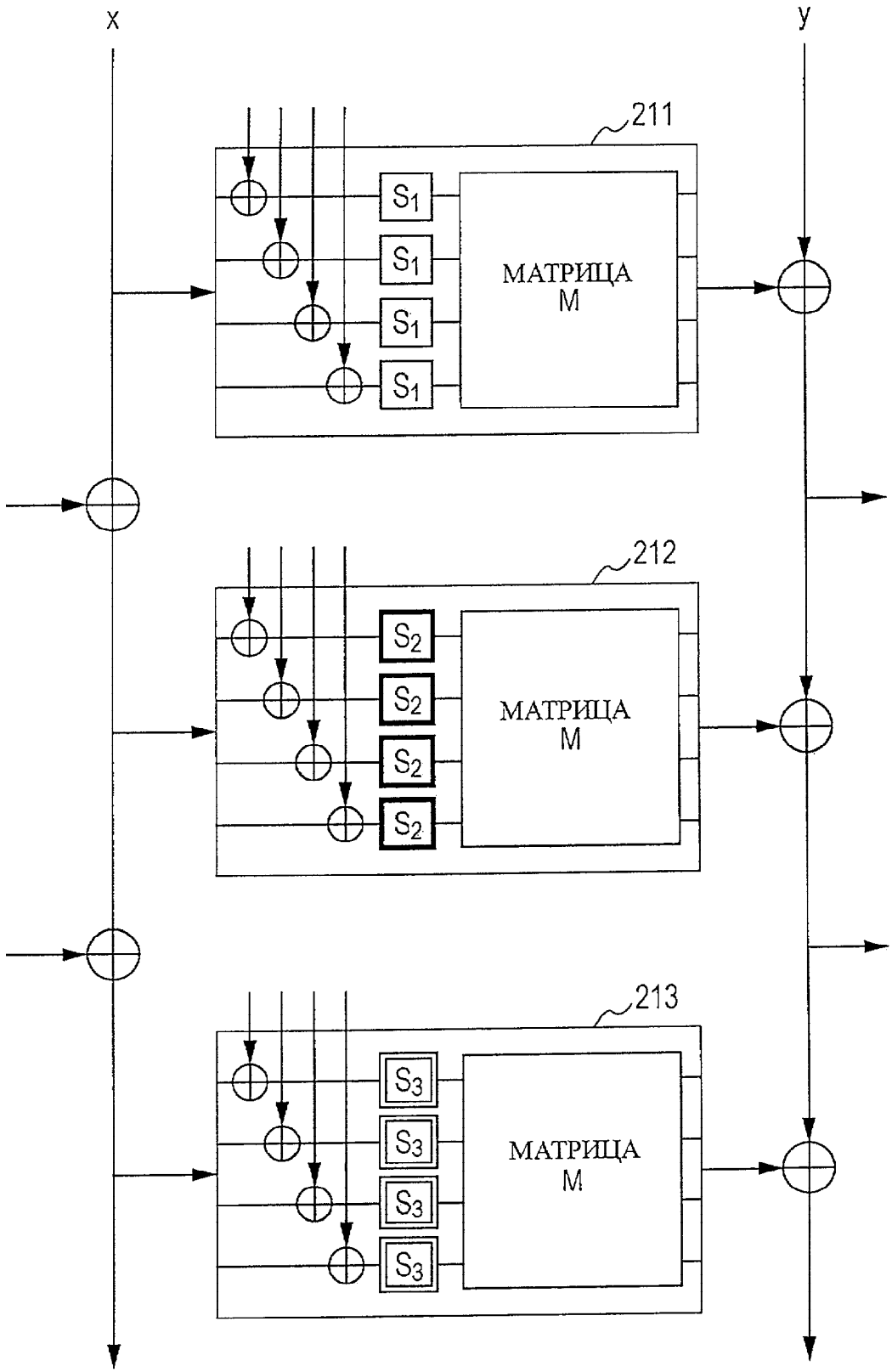
Фиг. 8


x
(РАУНДОВЫЙ КЛЮЧ)
I
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
101
y
E
S
S
S
S
МАТРИЦА M
A
B
111
МОДУЛЬ РАСЧЕТА «ИСКЛЮЧАЮЩЕЕ ИЛИ»
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
F
G
J
102
D
S
S
S
S
МАТРИЦА M
C
112
H

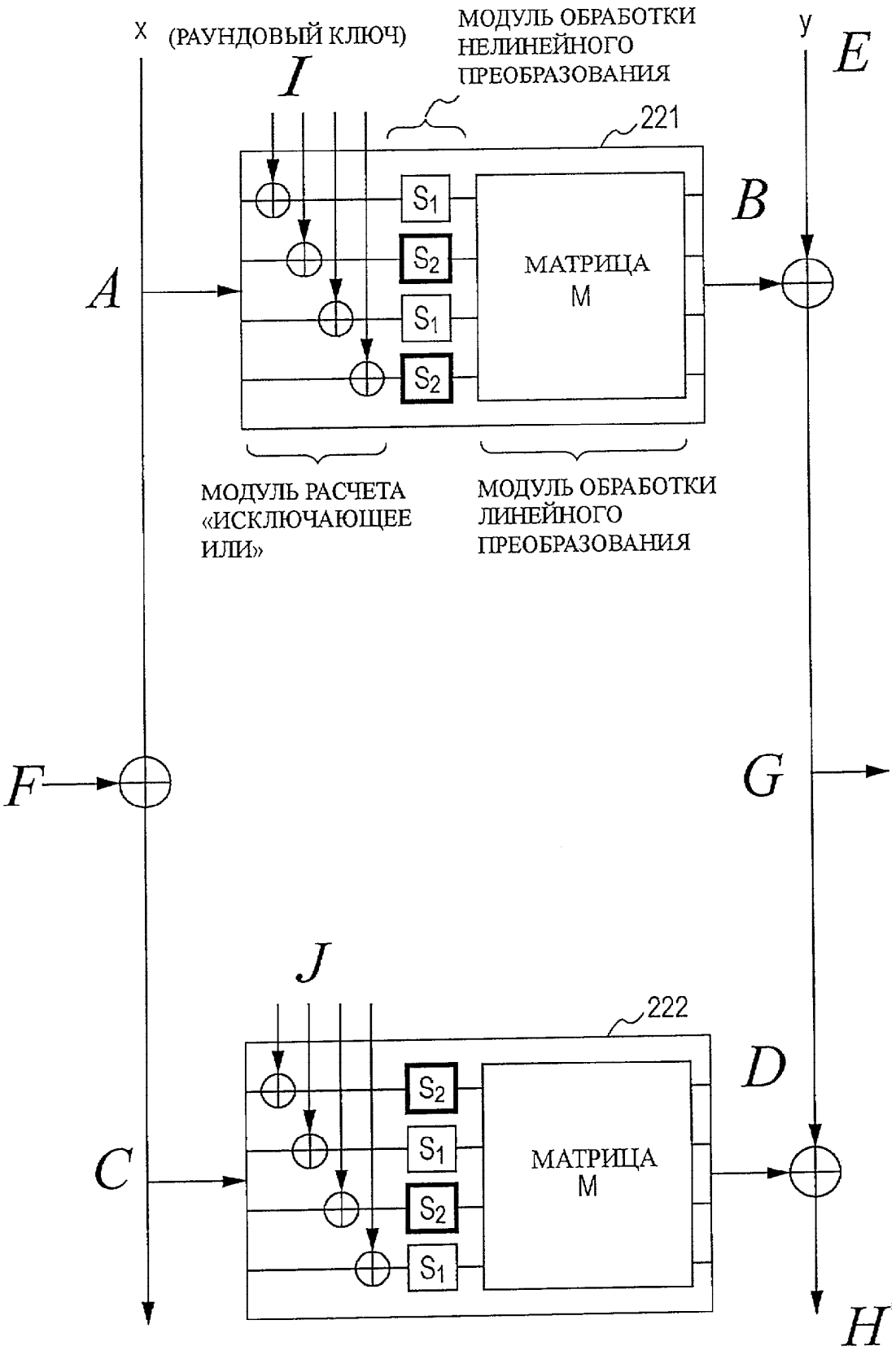
Фиг. 9


x
(РАУНДОВЫЙ КЛЮЧ)
I
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
201
y
E
S1
S1
S1
S1
МАТРИЦА M
A
B
МОДУЛЬ РАСЧЕТА «ИСКЛЮЧАЮЩЕЕ ИЛИ»
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
F
G
J
202
D
S2
S2
S2
S2
МАТРИЦА M
C
H

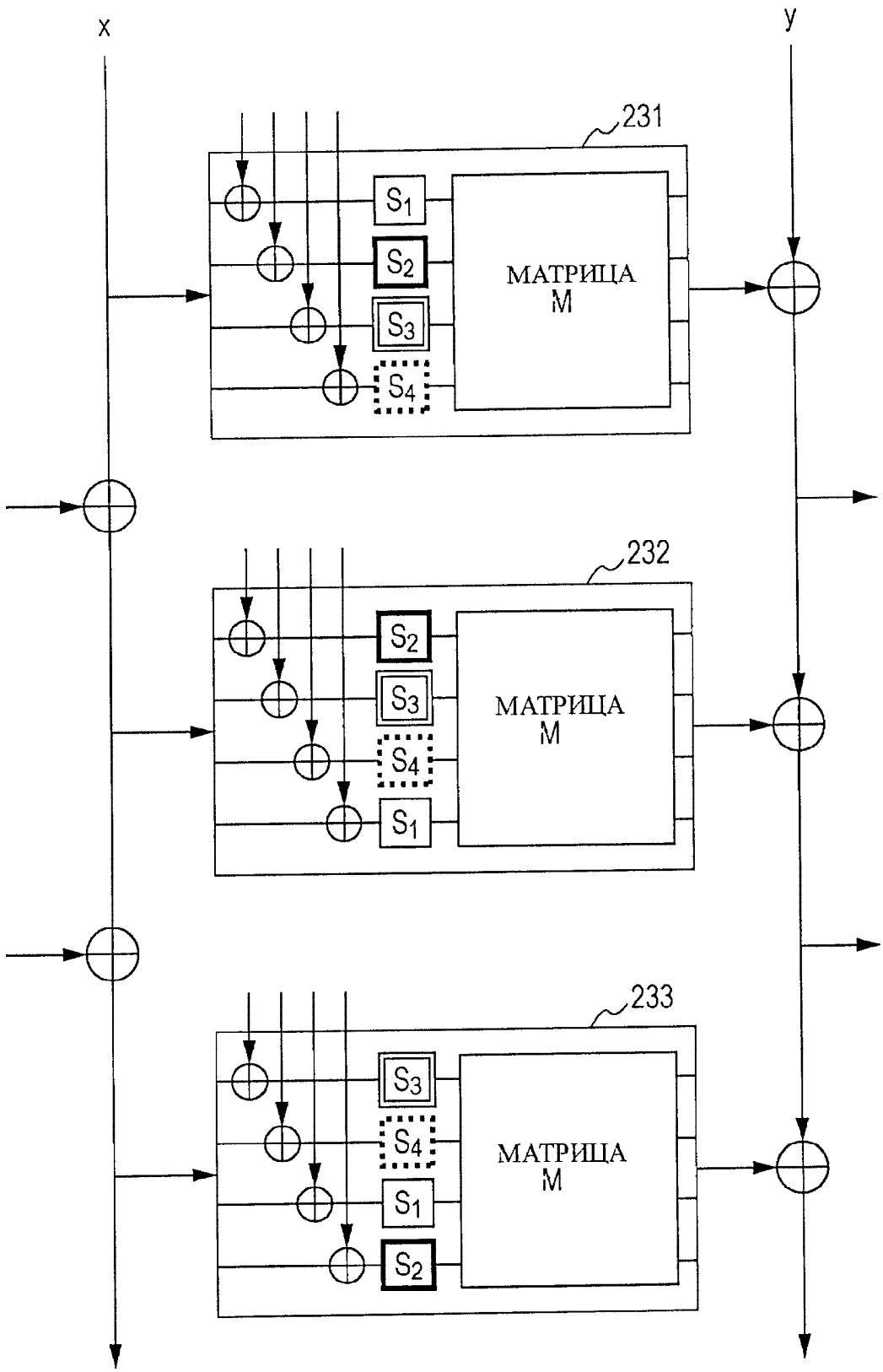
Фиг. 10


x
y
211
S1
S1
S1
S1
МАТРИЦА
M
212
S2
S2
S2
S2
МАТРИЦА
M
213
S3
S3
S3
S3
МАТРИЦА
M

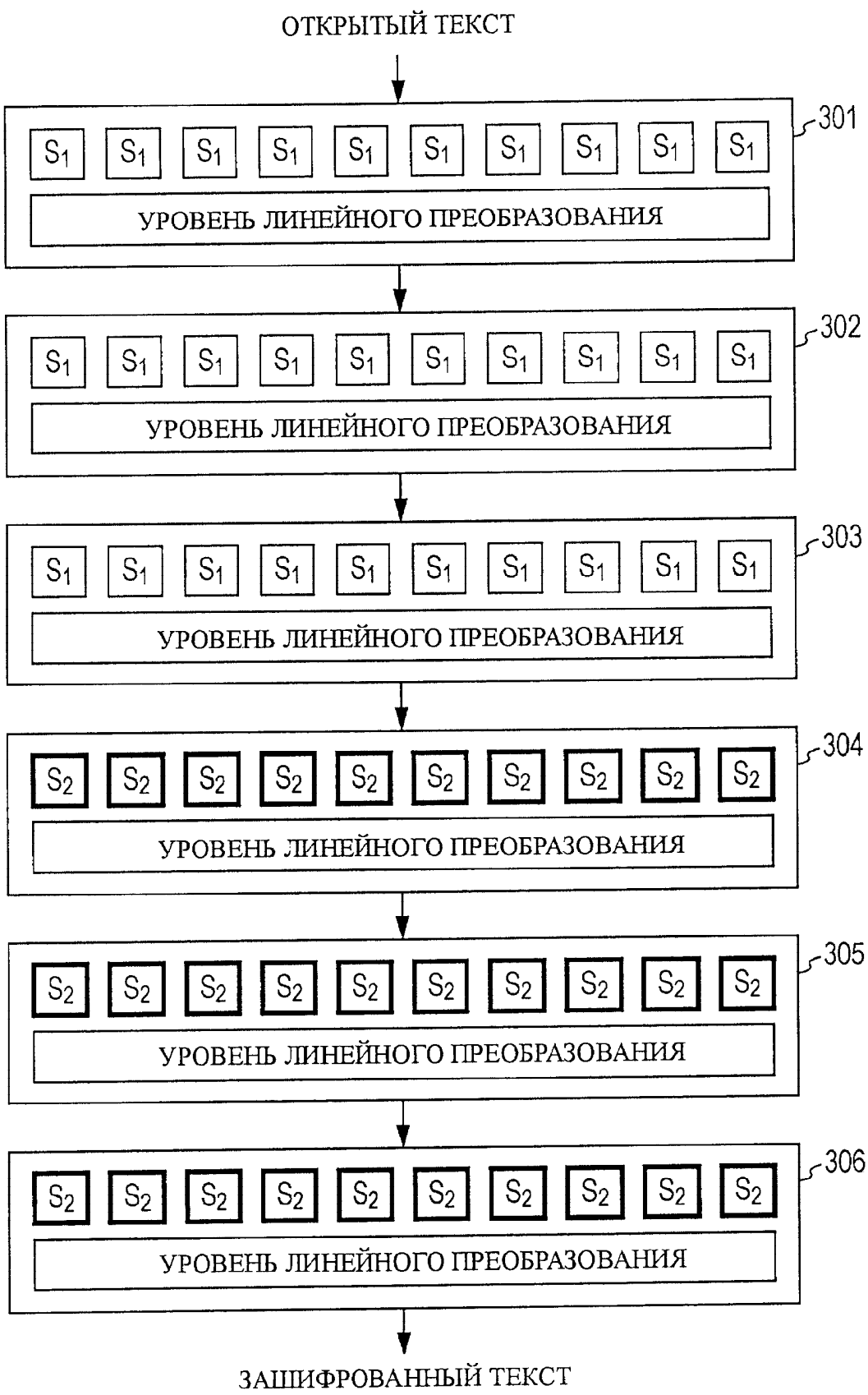
Фиг. 11


X
(РАУНДОВЫЙ КЛЮЧ)
I
МОДУЛЬ ОБРАБОТКИ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
221
y
E
B
A
S1
S2
S1
S2
МАТРИЦА M
МОДУЛЬ РАСЧЕТА «ИСКЛЮЧАЮЩЕЕ ИЛИ»
МОДУЛЬ ОБРАБОТКИ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
F
G
J
222
D
C
S2
S1
S2
S1
МАТРИЦА M
H

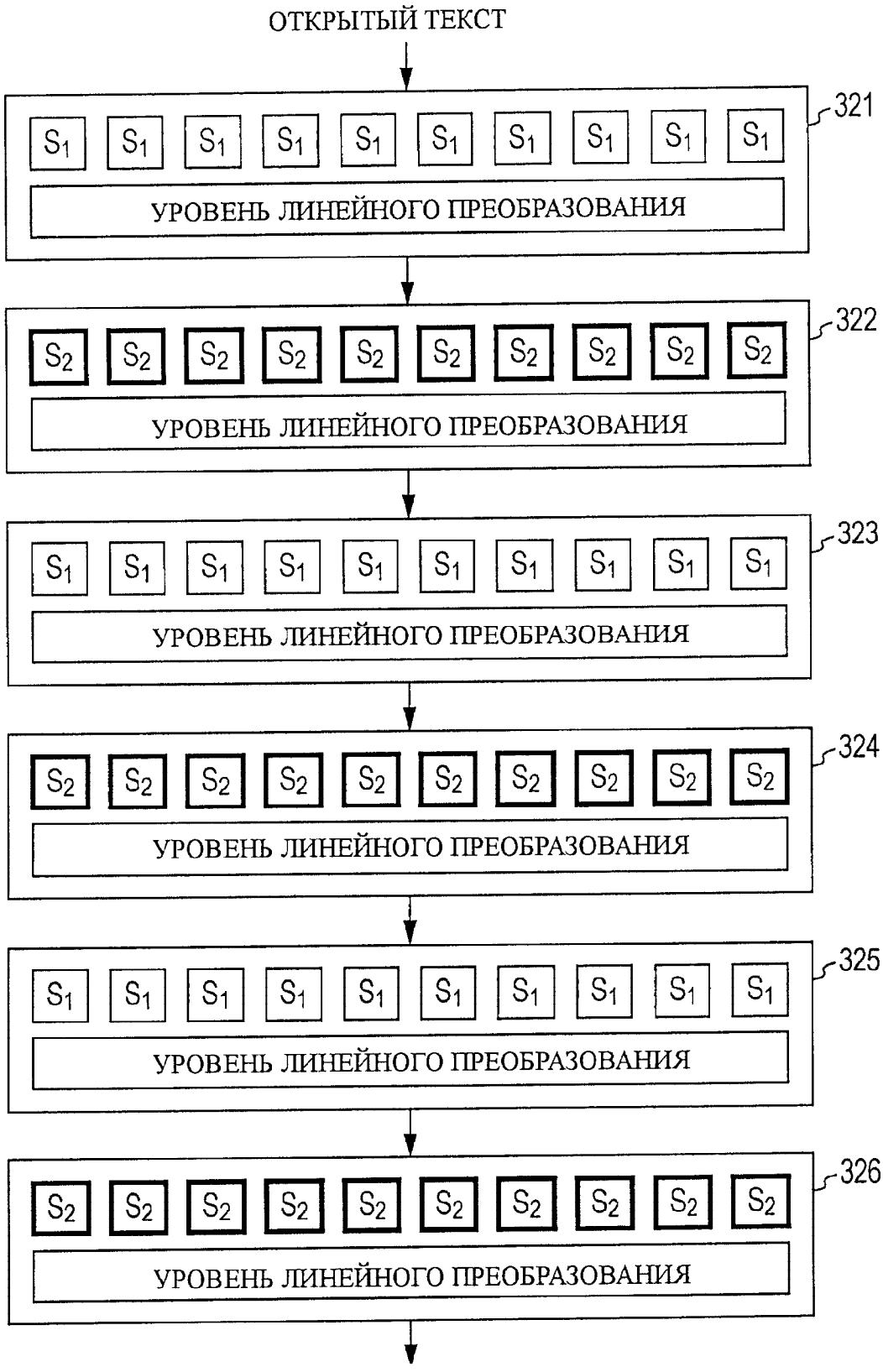
Фиг. 12


x
y
231
S1
S2
S3
S4
МАТРИЦА
M
232
S2
S3
S4
S1
МАТРИЦА
M
233
S3
S4
S1
S2
МАТРИЦА
M

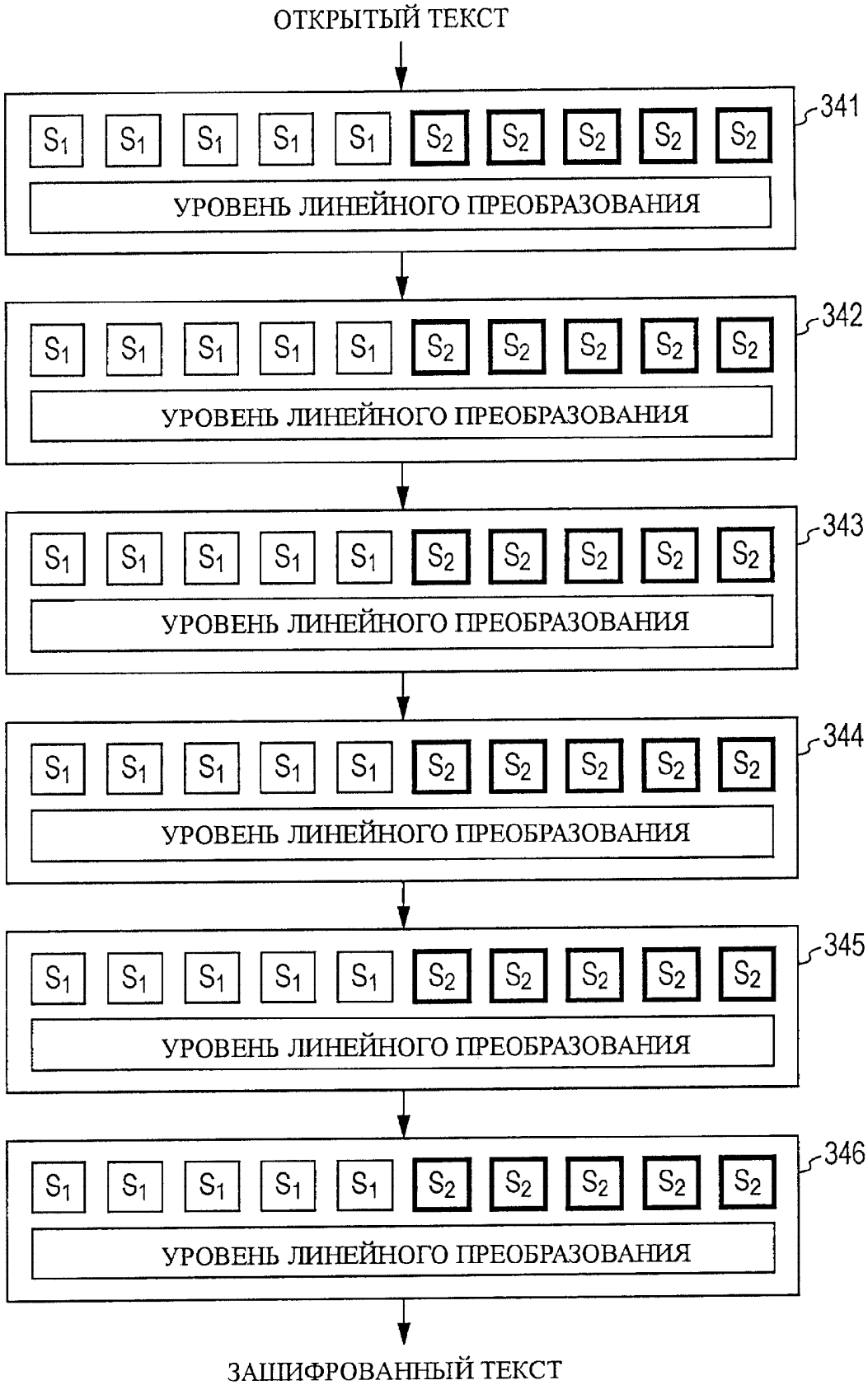
Фиг. 13


ОТКРЫТЫЙ ТЕКСТ
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
301
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
302
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
303
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
304
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
305
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
306
ЗАШИФРОВАННЫЙ ТЕКСТ

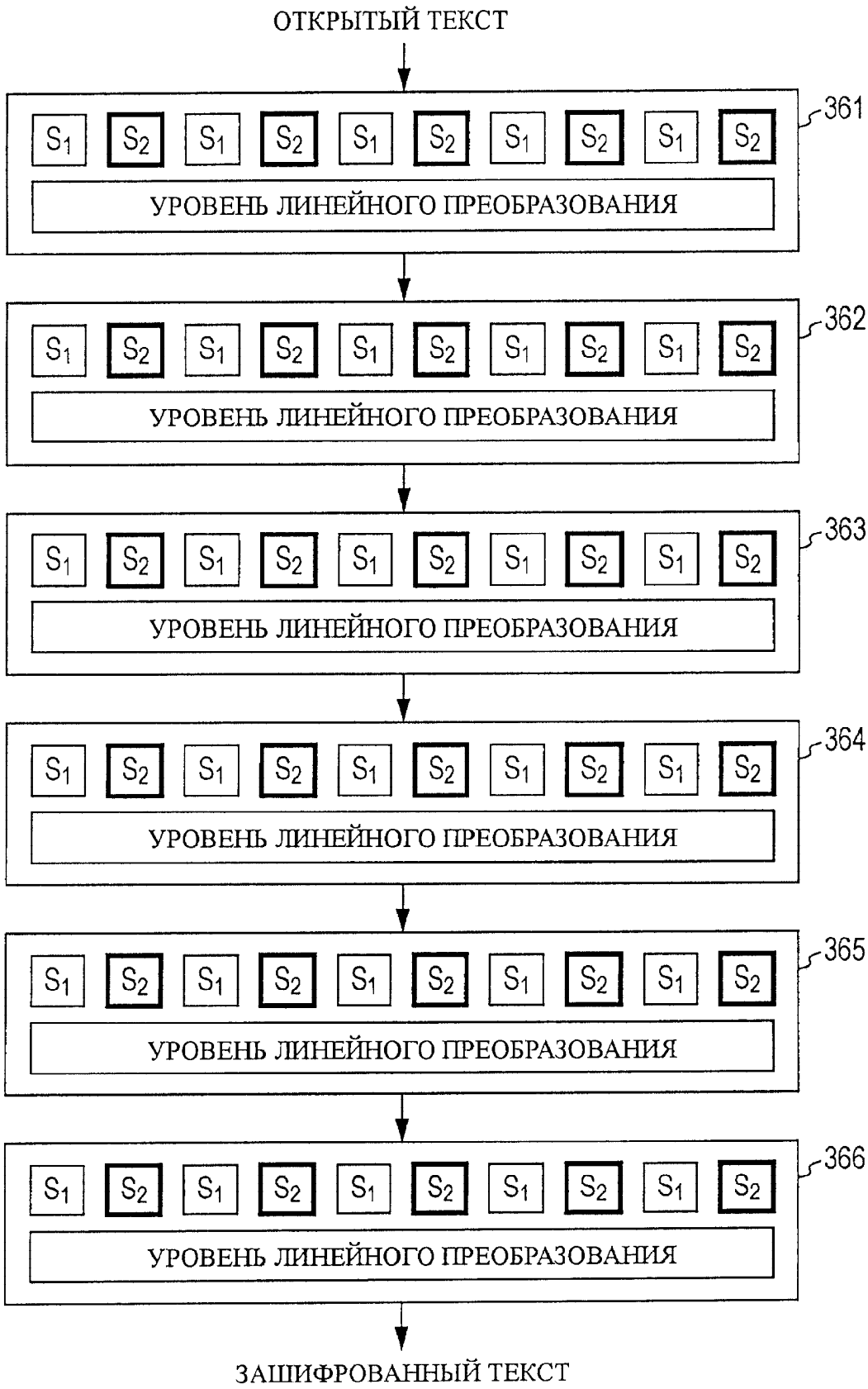
Фиг. 14


ОТКРЫТЫЙ ТЕКСТ
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
321
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
322
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
323
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
324
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
S1 S1 S1 S1 S1 S1 S1 S1 S1 S1
325
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
S2 S2 S2 S2 S2 S2 S2 S2 S2 S2
326
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
ЗАШИФРОВАННЫЙ ТЕКСТ

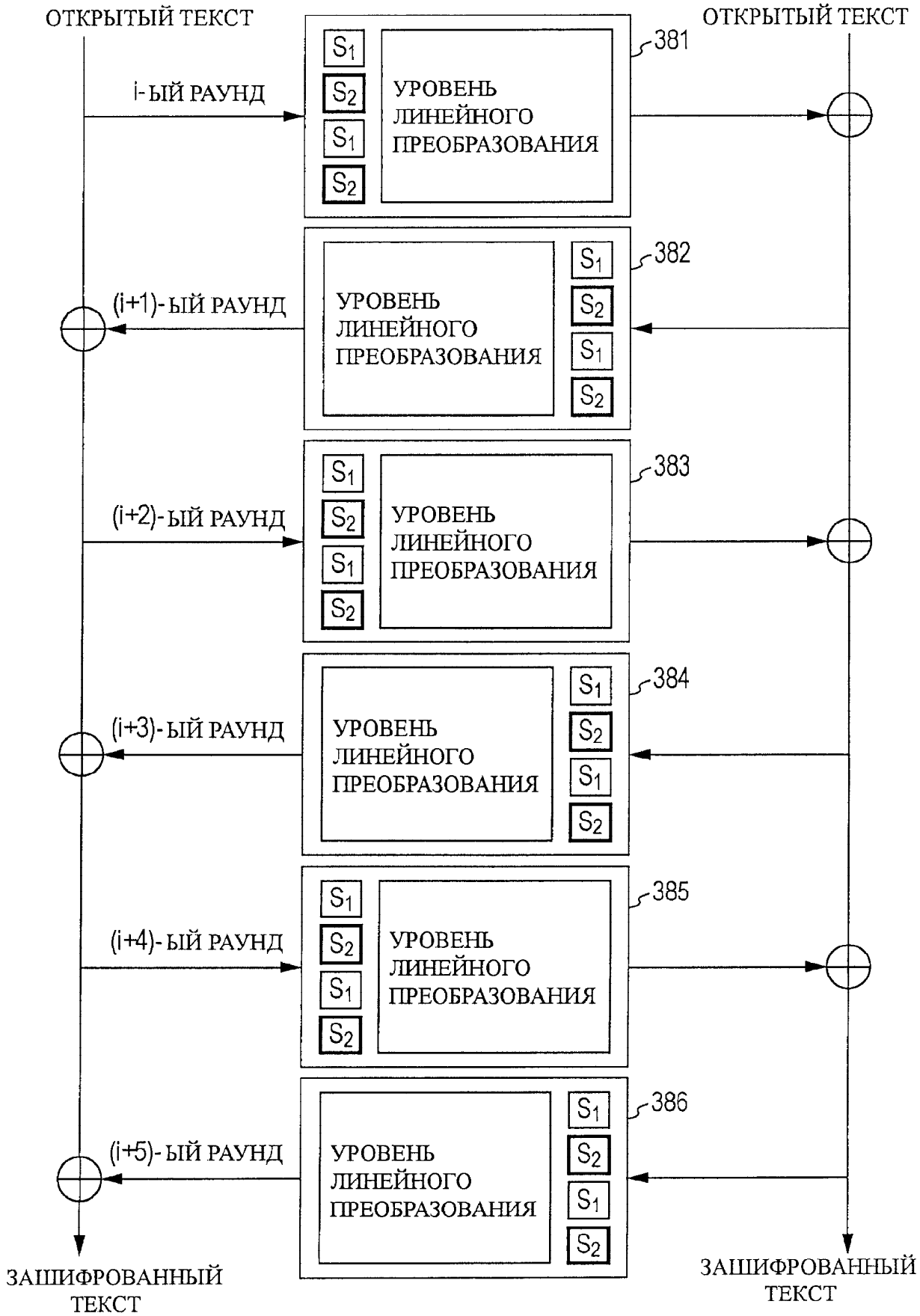
Фиг. 15


ОТКРЫТЫЙ ТЕКСТ
S1 S1 S1 S1 S1 S2 S2 S2 S2 S2
341
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
342
343
344
345
346
ЗАШИФРОВАННЫЙ ТЕКСТ

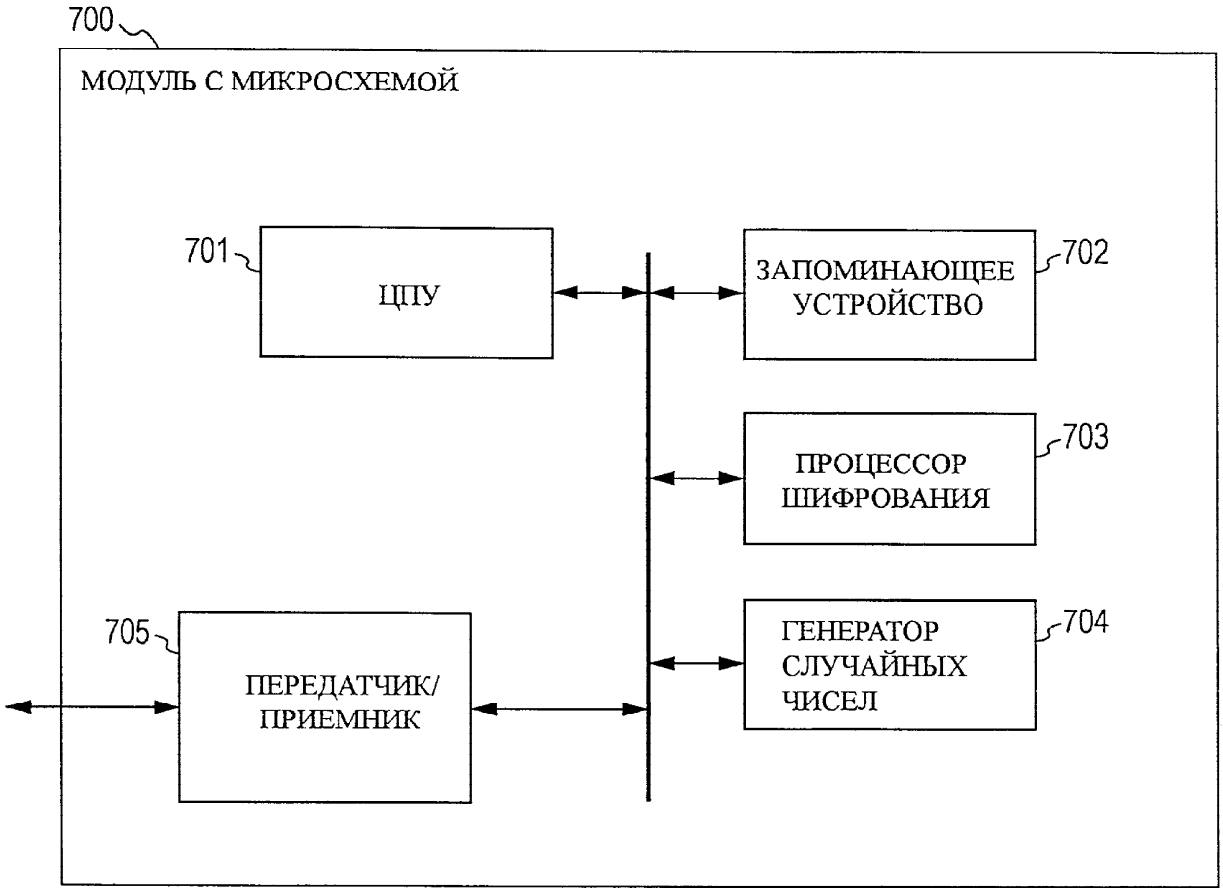
Фиг. 16


ОТКРЫТЫЙ ТЕКСТ
S1 S2 S1 S2 S1 S2 S1 S2 S1 S2
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
361
362
363
364
365
366
ЗАШИФРОВАННЫЙ ТЕКСТ

Фиг. 17


ОТКРЫТЫЙ ТЕКСТ
ОТКРЫТЫЙ ТЕКСТ
i-ЫЙ РАУНД
(i+1)-ЫЙ РАУНД
(i+2)-ЫЙ РАУНД
(i+3)-ЫЙ РАУНД
(i+4)-ЫЙ РАУНД
(i+5)-ЫЙ РАУНД
S1
S2
S1
S2
УРОВЕНЬ ЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ
381
382
383
384
385
386
ЗАШИФРОВАННЫЙ ТЕКСТ
ЗАШИФРОВАННЫЙ ТЕКСТ

Фиг. 18


700
МОДУЛЬ С МИКРОСХЕМОЙ
701
ЦПУ
ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО
702
ПРОЦЕССОР ШИФРОВАНИЯ
703
705
ПЕРЕДАТЧИК/ ПРИЕМНИК
ГЕНЕРАТОР СЛУЧАЙНЫХ ЧИСЕЛ
704

Фиг. 19