РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 621 182**(13) **C1**

(51) МПК
H04L 9/08 (2006.01)
G06F 21/62 (2013.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2014147044, 28.03.2013**

(24) Дата начала отсчета срока действия патента:
28.03.2013

Дата регистрации:
31.05.2017

Приоритет(ы):
(30) Конвенционный приоритет:
21.05.2012 US 61/649,464;
21.05.2012 EP 12168710.7;
12.06.2012 US 61/658,475

(45) Опубликовано: **31.05.2017** Бюл. № **16**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **22.12.2014**

(86) Заявка PCT:
EP 2013/056730 (28.03.2013)

(87) Публикация заявки PCT:
WO 2013/174554 (28.11.2013)

Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, строение 3, ООО "Юридическая фирма Городисский и Партнеры"

(72) Автор(ы):
ГАРСИЯ МОРЧОН Оскар (NL),
ТОЛХЭЙЗЕН Людовикус Маринус Герардус Мария (NL),
ГУТЬЕРРЕС Хайме (NL),
КУМАР Сандип Шанкаран (NL),
ГОМЕС Доминго (NL)

(73) Патентообладатель(и):
КОНИНКЛЕЙКЕ ФИЛИПС Н.В. (NL)

(56) Список документов, цитированных в отчете о поиске: **RU 2009101908 A, 27.07.2010. WO 95/05712 A2, 23.02.1995. WO 2010/032161 A1, 25.03.2010. WO 2010/106496 A1, 23.09.2010. WO 2007/149850 A2, 27.12.2007..**

(54) УСТРОЙСТВО СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ КЛЮЧА И СИСТЕМА ДЛЯ ЕГО КОНФИГУРАЦИИ

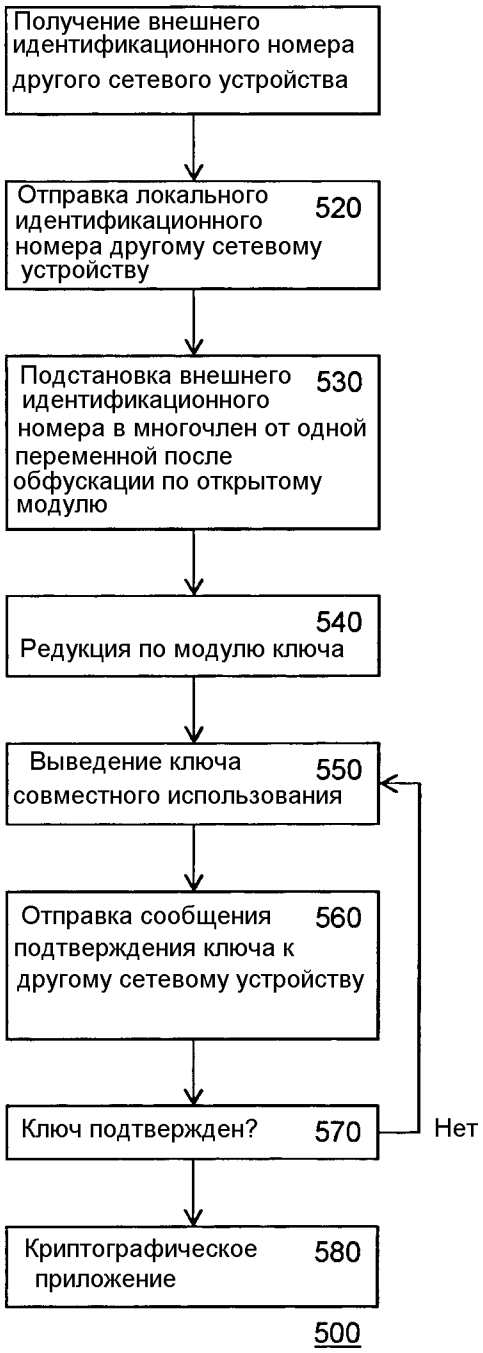
(57) Реферат:

Изобретение относится к области сетевой связи. Технический результат – обеспечение безопасности между двумя сетевыми устройствами за счет ключа совместного использования. Система для конфигурирования сетевого устройства для совместного использования ключа, содержащая: средство получения материала ключей для получения в электронной форме личного модуля (122, p_1), открытого модуля (110, N) и симметрического многочлена (124, f_1) от двух переменных, имеющего целочисленные коэффициенты, причем двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины (b) ключа, генератор (200) для генерирования локального материала ключей для сетевого устройства, содержащий средство (250) управления сетевыми устройствами для получения в электронной форме идентификационного номера (A) для сетевого устройства и для электронного сохранения генерируемого локального материала ключей в сетевом устройстве и сохранения открытого модуля в сетевом устройстве, и устройство (240) манипуляции многочленами для определения многочлена от одной переменной из многочлена от двух переменных путем подстановки

RU 2621182 C1

RU 2621182 C1

идентификационного номера в многочлен от двух переменных, редукции по личному модулю результата подстановки. 5 н. и 9 з.п. ф-лы, 6 ил.



ФИГ.5

RU 2621182 C1

RU 2621182 C1

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 621 182**(13) **C1**

(51) Int. Cl.
H04L 9/08 (2006.01)
G06F 21/62 (2013.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2014147044, 28.03.2013**

(24) Effective date for property rights:
28.03.2013

Registration date:
31.05.2017

Priority:
(30) Convention priority:
21.05.2012 US 61/649,464;
21.05.2012 EP 12168710.7;
12.06.2012 US 61/658,475

(45) Date of publication: **31.05.2017** Bull. № **16**

(85) Commencement of national phase: **22.12.2014**

(86) PCT application:
EP 2013/056730 (28.03.2013)

(87) PCT publication:
WO 2013/174554 (28.11.2013)

Mail address:
129090, Moskva, ul. B. Spasskaya, 25, stroenie 3,
OOO "Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):
GARSIYA MORCHON Oskar (NL),
TOLKHEJZEN Lyudovikus Marinus Gerardus Mariya (NL),
GUTERRES Khajme (NL),
KUMAR Sandip Shankaran (NL),
GOMES Domingo (NL)

(73) Proprietor(s):
KONINKLEJKE FILIPS N.V. (NL)

RU 2621182 C1

RU 2621182 C1

(54) KEY JOINT USAGE DEVICE AND THE SYSTEM FOR ITS CONFIGURATION

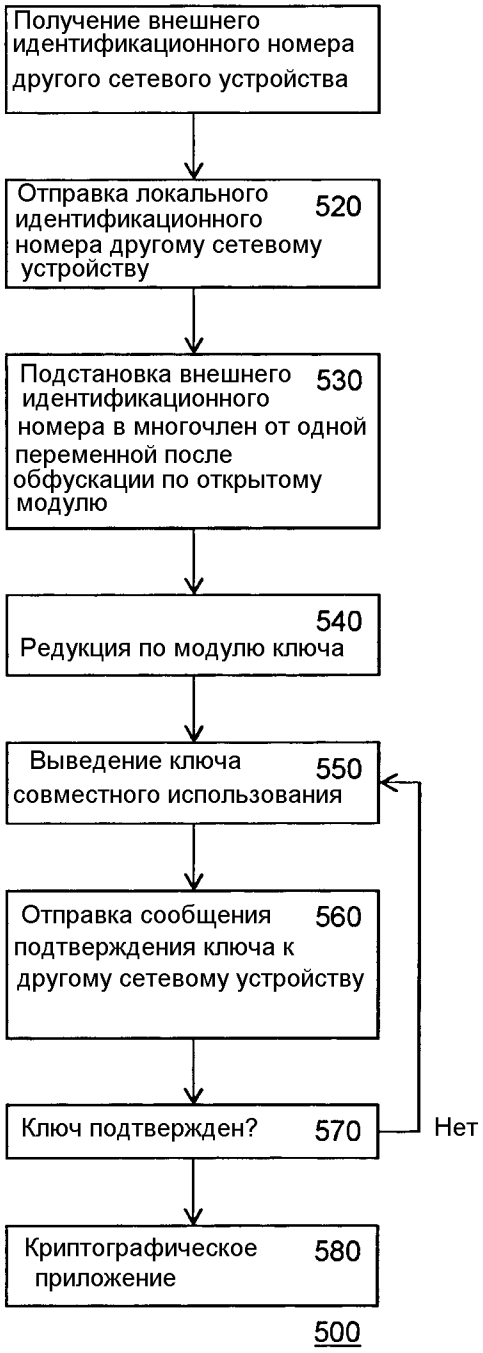
(57) Abstract:

FIELD: radio engineering, communication.

SUBSTANCE: system for the network device configuring for the key joint usage, containg: means for obtaining the key material for getting personal module in electronic form (122, p_1), the open module (110, N) and the symmetric polynomial (124, f_1) from two variables, having integral coefficients, at that the binary representation of the open module and the binary representation of the individual module are identical in at least in consecutive bits of the key length (b), generator (200) for generating the local key material for the network device, containing: network devices control means (250) to obtain the identification number (A) in electronic form for the network device and for the electronic storing of the generated local key material in the network device and storing the open module in the network device, and the polynomial manipulation device (240) to determe the polynomial from one variable from the polynomial from two variables by substituting the identification number in the polynomial from two variables, reduction of the substitution result according to the personal module.

EFFECT: security provision between two network devices by key joint usage.

14 cl, 6 dwg



ФИГ.5

RU 2621182 C1

RU 2621182 C1

Область техники, к которой относится изобретение

Изобретение относится к способу конфигурирования сетевого устройства для совместного использования ключа, причем способ содержит генерирование локального материала ключей для сетевого устройства, содержащее получение в электронной форме идентификационного номера для сетевого устройства, определение с использованием устройства манипуляции многочленами многочлена от одной переменной из многочлена от двух переменных путем подстановки идентификационного номера в многочлен от двух переменных и электронное сохранение генерируемого локального материала ключей в сетевом устройстве.

Изобретение дополнительно относится к способу для первого сетевого устройства для определения ключа совместного использования, причем ключ является криптографическим ключом, причем способ содержит получение локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен от одной переменной, получение идентификационного номера для второго сетевого устройства, причем второе сетевое устройство отличается от первого сетевого устройства, подстановку идентификационного номера второго сетевого устройства в многочлен от одной переменной и выведение ключа совместного использования из него.

Изобретение дополнительно относится к системе для конфигурирования сетевого устройства для совместного использования ключа и к сетевому устройству, сконфигурированному для определения ключа совместного использования.

Уровень техники

В сети связи, содержащей множество сетевых устройств, представляет проблему установление защищенных соединений между парами таких сетевых устройств. Один способ достижения этого описан в работе C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro и M. Yung "Perfectly-Secure Key distribution for Dynamic Conferences" ("Идеально защищенное распределение ключей для динамических конференций") в журнале "Lecture Notes in Mathematics" издательства Springer, том 740, стр. 471-486, 1993 г. (упоминаемой как "Blundo").

Он предполагает центральную администрацию, также называемую администрацией сети или доверенной третьей стороной (TTP), которая генерирует симметрический многочлен f(x, y) от двух переменных с коэффициентами в конечном поле F с p элементами, где p - простое число или степень простого числа. Каждое устройство имеет идентификационный номер в F и обеспечено локальным материалом ключей от TTP. Для устройства с идентификатором η локальным материалом ключей являются коэффициенты многочлена f(η, y).

Если устройству η нужно связаться с устройством η', оно использует свой материал ключей для генерирования ключа K(η, η')=f(η, η'). Поскольку f является симметрическим, генерируется один и тот же ключ.

Проблема этой схемы совместного использования ключа возникает, если злоумышленнику известен материал ключей t+1 или более устройств, где t является степенью многочлена от двух переменных. Тогда злоумышленник может восстановить многочлен f(x, y). В этот момент безопасность системы полностью нарушается. Благодаря идентификационным номерам любых двух устройств, злоумышленник может восстановить ключ, совместно используемый этой парой устройств.

Делается ссылка на работу "A Permutation-Based Multi-Polynomial Scheme for Pairwise Key Establishment in Sensor Networks" ("Схема с множеством многочленов на основе перестановок для установления парного ключа в сенсорных сетях") авторов Song Guo,

Victor Leung и Zhuzhong Qian с Международной конференции по связи IEEE, 2010 г. Она представляет схему с множеством многочленов на основе перестановок для установления парного ключа в беспроводных сенсорных сетях. В отличие от Blundo, схема, представленная в Song, дает каждому узлу не всего одну долю симметрического многочлена, а группу долей после перестановки.

Сущность изобретения

Будут обеспечены преимущества, если иметь улучшенный способ для установления ключа совместного использования между двумя сетевыми устройствами. Обеспечиваются способ конфигурации сетевого устройства для совместного использования ключа и способ для сетевого устройства для определения ключа совместного использования.

Способ конфигурации сетевого устройства для совместного использования ключа содержит получение в электронной форме личного модуля, открытого модуля и многочлена от двух переменных, имеющего целочисленные коэффициенты, причем двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины ключа, генерирование локального материала ключей для сетевого устройства, содержащее получение в электронной форме идентификационного номера для сетевого устройства, определение с использованием устройства манипуляции многочленами многочлена от одной переменной из многочлена от двух переменных путем подстановки идентификационного номера в многочлен от двух переменных, редукции по личному модулю результата подстановки и электронное сохранение генерируемого локального материала ключей в сетевом устройстве. В одном варианте осуществления, генерирование локального материала ключей для сетевого устройства содержит генерирование осуществляющего обфускацию числа, например посредством использования электронного генератора случайных чисел, и прибавление, с использованием устройства манипуляции многочленами, осуществляющего обфускацию числа к некоторому коэффициенту многочлена от одной переменной для получения многочлена от одной переменной после обфускации, причем генерируемый локальный материал ключей содержит многочлен от одной переменной после обфускации. Более одного коэффициента могут быть подвергнуты обфускации, предпочтительно различные коэффициенты подвергаются обфускации различным образом. В одном варианте осуществления генерирование локального материала ключей для сетевого устройства содержит генерирование множественных осуществляющих обфускацию чисел, например посредством электронного генератора случайных чисел, и прибавление, с использованием устройства манипуляции многочленами, каждого осуществляющего обфускацию числа из множественных осуществляющих обфускацию чисел к соответственному из коэффициентов многочлена от одной переменной для получения многочлена от одной переменной после обфускации. В одном варианте осуществления к каждому коэффициенту многочлена от одной переменной прибавляется число осуществляемой обфускации.

Многочлен от двух переменных может или может не быть симметрическим. Если многочлен или многочлены от двух переменных являются симметрическими, любые два сетевых устройства могут выводить ключ совместного использования. Интересно, что использование асимметрических многочленов от двух переменных или одного или нескольких асимметрических многочленов от двух переменных среди множественных многочленов от двух переменных в качестве корневого ключевого материала обеспечивает возможность предусмотреть создание двух групп устройств, таких как

устройства; два устройства, принадлежащих к одной и той же группе, не могут генерировать общий ключ, но два устройства в различных группах могут.

Добавление обфускации является необязательным этапом. Без обфускации защита против атак все равно достигается, поскольку выведение локального материала ключей использует личный модуль, который отличен от открытого модуля; математическая зависимость, которая имела бы место при работе, например, в одном конечном поле, нарушается. Это означает, что обычные математические инструменты для анализа многочленов, например конечномерная алгебра, более неприменимы. С другой стороны, поскольку личный и открытый модули перекрываются в некотором количестве последовательных бит, два сетевых устройства, которые имеют локальный материал ключей, с большой вероятностью будут иметь возможность вывести один и тот же ключ совместного использования. Безопасность может быть увеличена путем прибавления одного или нескольких осуществляющих обфускацию чисел к коэффициентам многочлена от одной переменной для получения многочлена от одной переменной после обфускации. Этап прибавления осуществляющих обфускацию чисел, однако, является необязательным и может опускаться. Добавлять или не добавлять обфускацию является компромиссом между вероятностью верного выведения ключа совместного использования и дополнительной безопасностью.

Открытый модуль предназначен для использования в сетевом устройстве. Способ конфигурирования сетевого устройства для совместного использования ключа может содержать обеспечение открытого модуля доступным сетевому устройству, например сохранение открытого модуля вместе с локальным материалом ключей.

Способ определения ключа совместного использования для первого сетевого устройства, где ключ является криптографическим ключом, содержит получение локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен от одной переменной, возможно после обфускации, получение идентификационного номера для второго сетевого устройства, причем второе сетевое устройство отличается от первого сетевого устройства, подстановку идентификационного номера второго сетевого устройства в многочлен от одной переменной после обфускации, редукцию результата подстановки по открытому модулю, с последующей редукцией по модулю ключа, и выведение ключа совместного использования из результата редукции по модулю ключа. В одном варианте осуществления, например, способ содержит редукцию результата подстановки по открытому модулю, деление результата на степень двух, и редукцию по модулю ключа.

Любая пара двух сетевых устройств из множественных сетевых устройств, каждое из которых имеет идентификационный номер и локальный материал ключей, генерируемый для идентификационного номера, имеет возможность согласовывать ключ совместного использования с небольшим количеством ресурсов. Двум сетевым устройствам нужно только обменяться их идентификационными номерами, которые не обязательно хранить в тайне, и выполнить вычисления с многочленами. Тип необходимых вычислений не требует больших вычислительных ресурсов, что означает, что этот способ подходит для низкозатратного крупномасштабного типа приложений.

Если локальный материал ключей был получен из симметрического многочлена, это обеспечивает возможность обоим сетевым устройствам в паре сетевых устройств получить один и тот же ключ совместного использования. Если осуществляющее обфускацию число было прибавлено к локальному материалу ключей, соотношение между локальным материалом ключей и корневым материалом ключей было нарушено. Соотношение, которое имело бы место между многочленом от одной переменной без

обфускации и симметрическим многочленом от двух переменных, больше не имеет место. Это означает, что прямая атака в такой схеме более не сработает.

Даже если никакая обфускация не была использована, сложность атаки остается, поскольку открытый модуль и личный модуль (или модули) неравны. Редукция по открытому модулю увеличивает вероятность выведения одного и того же ключа совместного использования, даже без обфускации.

В одном варианте осуществления двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины (b) ключа. Следует заметить, что могут быть использованы множественные личные модули; они могут быть выбраны так, чтобы двоичное представление любого из множественных личных модулей открытого модуля и двоичное представление личного модуля были одинаковыми в по меньшей мере последовательных битах длины (b) ключа. Для каждого личного модуля из множественных личных модулей, необязательно симметрический, многочлен от двух переменных, имеющий целочисленные коэффициенты, выбирается для получения множественных, необязательно симметрических, многочленов от двух переменных.

Поскольку выведение локального материала ключей использует личный модуль, который отличен от открытого модуля, математические зависимости, которые присутствовали бы при работе, например, в единственном конечном поле, искажены. Это означает, что обычные математические инструменты для анализа многочленов, например конечномерная алгебра, более неприменимы. В лучшем случае злоумышленник может использовать гораздо менее эффективные структуры, такие как решетки. Также при выведении ключа совместного использования комбинируются две операции по модулю, которые несовместимы в обычном математическом смысле; так что математическая структура избегается в двух местах. Способ обеспечивает возможность непосредственного попарного генерирования ключей и является устойчивым к захвату очень большого числа, например порядка 10^5 или даже выше, сетевых устройств. С другой стороны, поскольку личный и открытый модуль перекрываются в некотором количестве последовательных бит, два сетевых устройства, которые имеют локальный материал ключей, с большой вероятностью будут иметь возможность вывести один и тот же ключ совместного использования.

Конкретным озарением изобретателя было то, что открытый модуль не обязательно должен быть простым числом. В одном варианте осуществления открытый модуль является составным. Также нет причин для того, почему открытый модуль должен быть числом бит "с одними единицами", например числом, которое состоит только из битов 1 в своем двоичном представлении. В одном варианте осуществления открытый модуль не является степенью двух минус 1. В одном варианте осуществления двоичное представление открытого модуля содержит по меньшей мере один нулевой бит (не считая начальных нулей, т.е. двоичное представление открытого модуля содержит по меньшей мере один нулевой бит, менее значащий чем старший значащий бит открытого модуля). В одном варианте осуществления открытый модуль является степенью двух минус 1 и составным.

В одном варианте осуществления открытый модуль больше одного или нескольких личных модулей.

В одном варианте осуществления по меньшей мере последовательные биты длины ключа двоичного представления открытого модуля минус личный модуль все являются нулевыми битами. Эта разница должна быть оценена с использованием представления в виде числа со знаком открытого модуля минус личный модуль, не представления в

виде двух дополнений. В качестве альтернативы можно потребовать, чтобы по меньшей мере последовательные биты длины ключа двоичного представления абсолютного значения открытого модуля минус личный модуль все были нулевыми битами. Существует набор последовательных позиций длины (b) ключа, в котором двоичное представление открытого модуля согласуется с двоичным представлением всех личных модулей.

Последовательные битовые позиции, в которых открытый модуль согласуется с личными модулями, могут быть младшими значащими битами. В одном варианте осуществления младшие значащие биты длины ключа двоичного представления открытого модуля минус личный модуль все являются нулевыми битами; это обеспечивает преимущество в том, что деление на степень двух не необходимо при выведении ключа совместного использования.

Позволительно, чтобы личный модуль из множественных личных модулей был равен открытому модулю; однако, если только один личный модуль используется, то это нежелательно.

Желательно, чтобы личные модули предлагали достаточную нелинейность. В одном варианте осуществления существует набор последовательных битовых позиций, в котором открытый модуль отличается от каждого из личных модулей. Кроме того, также может быть наложено условие, что личные модули отличаются между собой; попарное сравнение двоичного представления личного модуля может также отличаться в по меньшей мере одном бите в наборе последовательных бит, например по меньшей мере длины ключа, причем набор равен для всех личных модулей и возможно также является тем же самым для открытого модуля.

Сетевое устройство может быть электронным устройством, оборудованным электронной связью и вычислительным средством. Сетевое устройство может быть присоединено, например, в форме RFID-метки, к любому неэлектронному объекту. Например, этот способ будет подходить для "Интернета вещей". Например, объекты, в частности объекты низкой стоимости, могут быть оборудованы радио-метками, через которые они могут осуществлять связь, например, могут быть идентифицированы. Может производиться инвентаризация таких объектов, например, посредством электронных средств, таких как компьютер. Украденные или сломанные единицы легко отследить и обнаружить. Одним особенно перспективным приложением является лампа, содержащая сетевое устройство, сконфигурированное для определения ключа совместного использования. Такая лампа может безопасно передавать по связи свое состояние; такой лампой может осуществляться безопасное управление, например включение и/или выключение. Сетевое устройство может быть одним из множественных сетевых устройств, каждое из которых содержит электронное средство связи для отправки и приема идентификационного номера и для отправки сообщения электронного состояния, и каждое из которых содержит интегральную цепь, сконфигурированную для выведения ключа совместного использования следуя способу согласно изобретению.

В одном варианте осуществления способ в изобретении может быть использован как криптографический способ для протоколов безопасности, таких как IPSec, (D)TLS, HIP или ZigBee. В частности, устройство, использующее один из этих протоколов, ассоциировано с идентификатором. Второе устройство, желающее осуществить связь с первым устройством, может генерировать общий парный ключ с первым устройством с учетом его идентификатора, и парный ключ (или ключ, найденный из этого посредством, например, функции выведения ключа) может быть использован в способе вышеупомянутых протоколов на основе предварительно выданного ключа. В частности,

идентификатором устройства, как определено в настоящем изобретении, может быть сетевой адрес, такой как короткий адрес ZigBee, IP-адрес или идентификатор хоста. Идентификатором также может быть IEEE-адрес устройства или собственная битовая строка, ассоциированная с устройством, так, чтобы устройство принимало некоторый локальный ключевой материал, ассоциированный с IEEE-адресом во время изготовления.

Выведение ключа совместного использования может быть использовано для многих применений. Обычно ключ совместного использования будет криптографическим симметричным ключом. Симметричный ключ может быть использован для конфиденциальности, например исходящие или входящие сообщения могут быть зашифрованы посредством симметричного ключа. Только устройство с доступом к обоим идентификационным номерам и одному из двух локальных материалов ключей (или доступом к корневому материалу ключей) будет иметь возможность дешифровать сообщения. Симметричный ключ может быть использован для аутентификации, например исходящие или входящие сообщения могут аутентифицироваться посредством симметричного ключа. Таким образом, источник сообщения может быть проверен. Только устройство с доступом к обоим идентификационным номерам и одному из двух локальных материалов ключей (или доступом к корневому материалу ключей) будет иметь возможность создавать аутентифицированные сообщения.

Способ конфигурирования сетевого устройства для совместного использования ключа обычно будет исполняться администрацией сети, например доверенной третьей стороной. Администрация сети может получать необходимый материал, например корневой материал ключей, от другого источника, но может также генерировать его сама. Например, может генерироваться открытый модуль. Например, может генерироваться личный модуль, даже если открытый модуль является системным параметром и принимается.

В одном варианте осуществления открытый модуль N выбирается так, чтобы он удовлетворял $2^{(a+2)b-1} \leq N \leq 2^{(a+2)b}-1$, где a представляет степень многочлена от двух переменных, a b представляет длину ключа. Например, в одном варианте осуществления $N=2^{(a+2)b}-1$. Операция по модулю для последнего упомянутого выбора может осуществляться особенно эффективно.

Наличие фиксированного открытого модуля имеет преимущество в том, что его нет необходимости передавать сетевым устройствам, но он может быть интегрирован, например, в их системные программные средства. В частности, открытый модуль может быть выбран с использованием генератора случайных чисел.

Открытый и личный модули могут быть представлены в качестве битовой строки. Они могут также быть сокращены с использованием конкретной математической структуры для каждого. Например, вместо сохранения личного модуля можно также сохранить его разницу с открытым модулем, которая гораздо короче.

Если личный модуль выбран таким образом, что число "длины ключа" младших значащих битов двоичного представления открытого модуля минус личный модуль все являются нулевыми битами, увеличивается вероятность того, что ключ совместного использования в первом сетевом устройстве пары сетевых устройств близок к ключу совместного использования, выведенному на втором сетевом устройстве из пары сетевых устройств; то есть двоичное представление личного модуля имеет те же самые биты в младших значащих позициях "длины ключа", что и двоичное представление открытого модуля. Например, если длина ключа равна 64, личный модуль может быть выбран путем вычитания величины, кратной 2^64, из открытого модуля. В одном варианте осуществления открытый модуль минус личный модуль, деленный на два в

степени длины ключа, меньше двух в степени длины ключа.

В одном варианте осуществления множественные личные модули получаются или генерируются в электронной форме, для каждого личного модуля из множественных личных модулей симметрический многочлен от двух переменных, имеющий целочисленные коэффициенты, выбирается для получения множественных симметрических многочленов от двух переменных так, чтобы каждому личному модулю соответствовал симметрический многочлен от двух переменных. Определение многочлена от одной переменной содержит подстановку идентификационного номера в каждый из множественных симметрических многочленов от двух переменных, редукцию по личному модулю из множественных личных модулей, соответствующих одному симметрическому многочлену от двух переменных, и сложение множества результатов множественных редукций вместе. Наличие множественных симметрических многочленов от двух переменных для различных модулей увеличивает безопасность, поскольку несовместимые структуры дополнительно смешаны. Обычно личные модули отдельны. Наличие множественных личных модулей дополнительно усложняет анализ еще больше, если соответствующие алгебраические структуры сильно различаются; например, если выбрать их взаимно простыми числами, в частности попарно взаимно простыми, или даже если выбрать их различными простыми числами.

Наличие другого личного модуля и, в частности, множественных личных модулей, усложнит анализ. Для дополнительного увеличения безопасности возможно дополнительное управление коэффициентами. В одном варианте осуществления администрация, складывающая множественные получающиеся в результате многочлены от одной переменной множественных редукций вместе, проверяет, является ли значение каждого из получающихся в результате коэффициентов либо слишком малым, либо слишком большим, например меньше минимального порога или больше максимального порога. Это улучшает безопасность еще больше, поскольку в любом из двух случаев злоумышленник может выяснить компоненты множественных редукций, если они слишком велики или слишком малы. Например, если значение коэффициента, получающегося в результате после сложения, равно 1 и есть только два многочлена от одной переменной, то злоумышленник знает, что либо соответствующий коэффициент, ассоциированный с первым многочленом, равен 1, а ассоциированный со вторым многочленом равен 0, или наоборот. В частности, администрация, генерирующая локальный материал ключей для устройства, может проверять, является ли значение каждого из получающихся в результате коэффициентов локального ключевого материала по меньшей мере "минимальным значением" и по большей мере "максимальным значением". Эта проверка может быть опущена, в частности, если открытый модуль относительно близок ко всем личным модулям и все элементы материала ключей находятся между 0 и N-1. Если TTP имеет возможность назначить идентификационные номера, она также может назначить другой идентификационный номер устройству, если TTP обнаруживает малые или большие коэффициенты.

В одном варианте осуществления каждый конкретный личный модуль таков, что младшие значащие биты длины (b) ключа двоичного представления открытого модуля минус конкретный личный модуль все являются нулевыми битами.

Открытый модуль может быть как больше, так и меньше личного модуля. В одном варианте осуществления двоичное представление открытого модуля минус личный модуль имеет по меньшей мере все нулевые биты длины ключа. Нулевые биты из по меньшей мере нулевых бит длины ключа являются последовательными и могут быть представлены в любой точке в двоичном представлении. Наличие строки из нулевых

битов в разнице между открытым модулем и личным модулем избегает того, что обфускация заходит слишком далеко. В одном варианте осуществления существует целочисленный параметр "s" такой, чтобы младшие значащие биты длины ключа открытого модуля минус личный модуль, деленные на два в степени s, были все нулевыми. Параметр "s" является одним и тем же для всех личных модулей.

Например, можно определить делитель строки нулевых битов, который является степенью двух, так, чтобы каждый конкретный личный модуль был такой, чтобы биты длины (b) ключа двоичного представления открытого модуля минус конкретный личный модуль, деленные на делитель строки нулевых битов, все были нулевыми битами. Если младшие значащие биты являются нулевыми, делитель строки нулевых битов может быть взят равным 1. В одном варианте осуществления делитель строки нулевых битов больше 1. Деление на степень двух следует интерпретировать как целочисленное деление, дающее тот же самый результат, что и смещение битов в направлении младших значащих битов. Любой остаток от деления игнорируется.

Для генерирования ключа совместного использования из бит длины ключа сетевые устройства сначала применяют этап дополнительного деления. Первое сетевое устройство оценивает ключевой материал для идентификационного номера второго устройства по открытым модулям, деля на 2^s и осуществляя редукцию по модулю два в степени длины ключа. Следует заметить, что это равносильно применению сначала модуля 2^(s+длина ключа) после открытого модуля и затем делению на 2^s. Здесь "деление" включает в себя округление в меньшую сторону.

В одном варианте осуществления личный модуль генерируется с использованием генератора случайных чисел. В одном варианте осуществления множественные личные модули генерируются так, чтобы они были попарно взаимно простыми. Например, множественные личные модули могут генерироваться итерационно с проверкой для каждого нового личного модуля, что они все еще попарно взаимно простые, и если нет, последний сгенерированный личный модуль отбрасывается. Вариант осуществления содержит итерационное генерирование модуля-кандидата с использованием генератора случайных чисел так, чтобы последовательные биты длины (b) ключа двоичного представления открытого модуля минус модуль-кандидат все были нулевыми битами, например младшими значащими битами длины ключа, пока модуль-кандидат не удовлетворяет тесту простоты с использованием устройства теста простоты, причем таким образом полученный модуль-кандидат, удовлетворяющий тесту простоты, используется в качестве личного модуля. Тестом простоты может быть, например, тест простоты Миллера-Рабина или тест простоты Соловея-Штрассена.

Симметрический многочлен от двух переменных с переменными x и y степени a имеет только одночлены формы x^iy^j, где $i \le a$, $j \le a$. Кроме того, коэффициент, соответствующий x^iy^j, является тем же самым, что и коэффициент x^jy^i. Это может быть использовано для уменьшения количества сохраненных коэффициентов примерно вдвое. Следует заметить, что используется более мягкое определение степени. Мы определяем степень одночлена как максимальную степень переменных в одночлене. То есть степень x^jy^i равна max(i,j), где $i \le a$, $j \le a$. Так что, например, то, что мы называем многочленом степени 1, имеет общую форму a+bx+cy+dxy, (следует заметить, что, поскольку считаются только симметрические многочлены, мы имеем то, что b=c). Следует заметить, что при желании можно наложить дополнительные ограничения на многочлен от двух переменных, в том числе, например, то, что только одночлены с $i+j \le a$ используются, но в этом нет необходимости.

В одном варианте осуществления симметрический многочлен от двух переменных генерируется администрацией сети. Например, симметрический многочлен от двух переменных может быть случайным симметрическим многочленом от двух переменных. Например, коэффициенты могут быть выбраны в виде случайных чисел с использованием генератора случайных чисел.

Хотя используемая обфускация сильно увеличивает устойчивость против атак, в частности против атак по коллизии, когда комбинируется множество локальных материалов ключей, оно имеет потенциальный недостаток. Иногда ключ совместного использования, выведенный первым сетевым устройством, не во всех битах идентичен ключу совместного использования, выведенному вторым сетевым устройством. Это происходит главным образом ввиду расхождения в битах переноса после прибавления осуществляющих обфускацию коэффициентов. Другая причина состоит в отсутствии эффекта модульных эффектов каждого из личных модулей в течение генерирования ключа, что влияет на генерируемые биты переноса. Хотя он и является помехой, этот недостаток может быть решен различными способами. Путем выбора обфускации с большей осторожностью вероятность различия и, в частности, вероятность большого различия может быть существенно уменьшена. Кроме того, было найдено, что различия, если таковые присутствуют, с большой вероятностью расположены в младших значащих битах генерируемых ключей. Так что путем удаления одного или нескольких из младших значащих битов вероятность идентичного ключа совместного использования может быть увеличена. Например, в одном варианте осуществления способа определение ключа совместного использования содержит определение, вывели ли первое сетевое устройство и второе сетевое устройство один и тот же ключ совместного использования, и если не вывели, выведение дополнительного ключа совместного использования из результата редукции по модулю ключа. Кроме того, ключи совместного использования могут выводиться, пока не найден такой, который равен с обеих сторон. Если в ключе совместного использования остается меньше порогового количества бит, способ может быть прерван. Для некоторых приложений может быть просто принято, что некоторый процент сетевых устройств не имеет возможности осуществлять связь. Например, в специализированных беспроводных сетях, где сообщение может маршрутизироваться через различные маршруты, нет потерь способности подключения, если некоторые из сетевых устройств не имеют возможности осуществлять связь.

Следует заметить, что без обфускации может также случиться, что ключ совместного использования, выведенный первым сетевым устройством, не во всех битах идентичен ключу совместного использования, выведенному вторым сетевым устройством, хотя вероятность этого меньше, чем в случае с обфускацией.

В одном варианте осуществления некоторое количество младших значащих битов ключа совместного использования удаляются; например, количество удаленных бит может быть 1, 2 или более, 4 или более, 8 или более, 16 или более, 32 или более, 64 или более. Путем удаления большего количества младших значащих битов вероятность получения ключей, которые не равны, уменьшается; в частности, она может быть уменьшена до любого желаемого порога. Вероятность, что ключи совместного использования равны, может быть вычислена путем следования математическим зависимостям, она может также быть определена опытным путем.

Также может осуществляться управление выбором осуществляющих обфускацию чисел, в одном варианте осуществления диапазон, из которого выбирается осуществляющее обфускацию число, уменьшен для коэффициентов, соответствующих одночленам более высоких степеней. В частности, можно потребовать, чтобы $|\in_{A,i}|<2^{(a+1-}$

$^{i)b}$, где $\in_{A,i}$ обозначает осуществляющее обфускацию число для i-го одночлена, i обозначает степень одночлена, соответствующего коэффициенту, а представляет степень многочлена от двух переменных, и b представляет длину ключа. А представляет сетевое устройство, для которого генерируется локальный материал ключей. В одном варианте осуществления осуществляющее обфускацию число генерируется для каждого коэффициента, например с использованием вышеприведенной формулы. Другая обфускация может применяться для других сетевых устройств. Например, даже если присутствует 3 или более сетевых устройств, то для каждого сетевого устройства могут генерироваться различные осуществляющие обфускацию числа.

Следует заметить, что осуществляющее обфускацию число может быть ограничено положительными числами, но в этом нет необходимости, осуществляющие обфускацию числа могут быть отрицательными. В одном варианте осуществления числа осуществляемой обфускации генерируются с использованием генератора случайных чисел. Множественные осуществляющие обфускацию числа могут генерироваться и прибавляться к коэффициентам многочлена от одной переменной для получения многочлена от одной переменной после обфускации. Обфускация одного или нескольких, предпочтительно даже всех коэффициентов многочлена от одной переменной может быть осуществлена таким образом.

Количество бит в идентификационном номере для сетевого устройства обычно выбирается как меньшее или равное длине ключа. Идентификационный номер может быть битовой строкой, например 32 или 64, или более длинной, битовой строкой. Длина ключа может быть 32 или более, 48 или более, 64 или более, 96 или более, 128 или более, 256 или более. Длиной ключа может быть выбрано некоторое более высокое количество бит для того, чтобы уменьшить соответствующее количество младших значащих бит определяемого ключа совместного использования. С другой стороны, в одном варианте осуществления длина идентификационного номера длиннее, чем длина ключа. В этом случае эффект модулярных операций может приводить к большему эффекту на младших значащих битах из битов длины ключа генерируемого ключа так, чтобы эти биты не могли быть равными для пары устройств, желающих сгенерировать общий ключ. Наличие большей длины для идентификатора может иметь, однако, положительный эффект в безопасности, поскольку больше битов смешано вместе, когда производятся соответствующие вычисления.

Устройство манипуляции многочленами может осуществляться в программных средствах, запущенных на компьютере, например в интегральной цепи. Устройство манипуляции многочленами может очень эффективно осуществляться в аппаратных средствах. Комбинация также возможна. Например, устройство манипуляции многочленами может осуществляться путем манипулирования матрицами коэффициентов, представляющих многочлены.

Электронное сохранение генерируемого локального материала ключей в сетевом устройстве может осуществляться путем электронной отправки генерируемого локального материала ключей сетевому устройству, например с использованием проводного соединения или с использованием беспроводного соединения, и наличия генерируемого локального материала ключей, сохраненного в сетевом устройстве. Это может осуществляться во время изготовления или установки, например во время испытания, интегральной цепи в сетевом устройстве. Оборудование испытаний может содержать или быть подключенным к администрации сети. Это может также происходить после успешного присоединения устройства к операционной сети (т.е. после сетевого доступа или начальной загрузки). В частности, локальный материал ключей может

распространяться в составе параметров операционной сети.

Получение локального материала ключей для первого сетевого устройства в электронной форме может осуществляться путем электронного приема локального материала ключей от системы для конфигурирования сетевого устройства для совместного использования ключа, например устройства администрации сети. Получение локального материала ключей может также осуществляться путем извлечения локального материала ключей из локального хранилища, например памяти, такой как флэш-память.

Получение идентификационного номера для второго сетевого устройства может осуществляться путем приема идентификационного номера от второго сетевого устройства, например непосредственно от второго сетевого устройства, например приема беспроводным образом от второго сетевого устройства.

Открытый модуль и модуль ключа могут сохраняться в сетевом устройстве. Они могут также приниматься от администрации сети. Они могут также предполагаться в программных средствах сетевого устройства. Например, в одном варианте осуществления модуль ключа является степенью двух. Редукция по такому модулю ключа может быть выполнена путем отбрасывания всех битов, кроме младших значащих битов длины ключа. Сначала осуществляется редукция результата подстановки по открытому модулю, который затем дополнительно редуцируется по модулю ключа.

Хотя это не требуется, открытый модуль и модуль ключа могут быть взаимно простыми. Это может достигаться путем наличия открытого модуля в виде нечетного числа и модуля ключа в виде степени 2. В любом случае, избегается то, что модуль ключа делит открытый модуль, поскольку затем редукция по открытому модулю может быть опущена.

Способ для согласования ключей между двумя устройствами может использовать в качестве корневого ключевого материала некоторое количество многочленов от двух переменных. Можно использовать способ для согласования ключей для x-согласования между x сторонами посредством многочленов от x переменных в качестве корневого ключевого материала. В этом расширении, доверенная третья сторона оценивает многочлены от x переменных с переменной в соответствующем кольце, получающиеся многочлены от x-1 переменных затем складываются над целыми числами, генерируя локальный материал ключей, сохраненный на устройстве. Когда x устройствам необходимо согласовать ключ, устройство оценивает свой локальный материал ключей в идентификаторах других x-1 устройств. Например, можно использовать многочлены от множества переменных в способе конфигурирования сетевого устройства для совместного использования ключа, причем способ содержит получение в электронной форме личного модуля (p_1), открытого модуля (N) и многочлена от множества переменных (f_1), имеющего целочисленные коэффициенты, причем двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины (b) ключа, генерирование локального материала ключей для сетевого устройства, содержащее получение в электронной форме идентификационного номера (A) для сетевого устройства, определение с использованием устройства манипуляции многочленами многочлена из многочлена от множества переменных путем подстановки идентификационного номера в многочлен от множества переменных, редукции по личному модулю результата подстановки и электронное сохранение генерируемого локального материала ключей в сетевом устройстве. Многочлен, полученный устройством манипуляции многочленами, имеет на одну переменную меньше. Для совместного использования ключа удобно, если

многочлен от множества переменных является симметрическим по всем переменным. Соответствующий способ для первого сетевого устройства для определения ключа совместного использования, причем ключ является криптографическим ключом, причем способ содержит получение локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен, необязательно, после обфускации, получение идентификационного номера для множественных других сетевых устройств, причем второе сетевое устройство отличается от первого сетевого устройства, подстановку идентификационного номера других сетевых устройств в упомянутый многочлен, необязательно после обфускации, редукцию результата подстановки по открытому модулю и редукцию по модулю ключа, и выведение ключа совместного использования из результата редукции по модулю ключа. Следует заметить, что после подстановки всех, кроме одного из других идентификационных номеров, способ уменьшается до ситуации, в которой используется многочлен от одной переменной.

В одном варианте осуществления первое сетевое устройство принимает множественные (n) локальные материалы ключей, ассоциированные с идентификатором устройства. Ключ, генерируемый между этим первым устройством и вторым устройством, получается в виде комбинации (например, конкатенации) множественных (n) ключей, полученных путем оценки каждого из множественных (n) локальных материалов ключей первого устройства в идентификаторе второго устройства. Это обеспечивает возможность использования способа параллельно.

Использование асимметрических многочленов от двух переменных в качестве корневого ключевого материала, т.е. f(x, y)!=f(y, x), обеспечивает возможность предусмотреть создание двух групп устройств, например устройства в первой группе принимают KM(Id, y), а устройства во второй группе принимают KM(x, iD), где KM - локальный материал ключей, сохраненный на устройстве. Два устройства, принадлежащие к одной и той же группе, не могут генерировать общий ключ, но два устройства в различных группах могут. См. дополнительно Blundo.

Идентификационный номер сетевого устройства может быть вычислен как односторонняя функция от битовой строки, содержащей информацию, ассоциированную с устройством. Односторонней функцией может быть криптографическая хэш-функция, такая как SHA2 или SHA3. Выходные данные односторонней функции могут быть сокращены так, чтобы удовлетворять размеру идентификатора. В качестве альтернативы размер односторонней функции меньше максимального размера идентификатора.

В одном варианте осуществления симметрические многочлены включают в себя один одночлен в форме $<ax^iy^i>_{pj}$, где $<>_p$ представляет модульную операцию. В этом случае элементы находятся внутри конечной группы, и операция является умножением. Открытый модуль может быть больше личного модуля или меньше; если существуют множественные личные модули, некоторые могут быть больше личного модуля и некоторые могут быть меньше.

В одном варианте осуществления способа конфигурирования сетевого устройства для совместного использования ключа способ содержит получение в электронной форме множественных личных модулей (p_i) и множественных симметрических многочленов (f_i) от двух переменных, имеющих целочисленные коэффициенты, так что существует набор последовательных позиций длины (b) ключа, в котором двоичное представление открытого модуля является тем же самым, что и двоичное представление всех личных модулей, генерирование локального материала ключей для сетевого

устройства, содержащее получение в электронной форме идентификационного номера (A) для сетевого устройства, определение с использованием устройства манипуляции многочленами многочлена от одной переменной из множественных многочленов от двух переменных путем подстановки идентификационного номера в каждый из множественных многочленов от двух переменных, редукцию по личному модулю из множественных личных модулей, соответствующих одному симметрическому многочлену от двух переменных, и сложение множества результатов множественных редукций, и генерирование осуществляющего обфускацию числа и прибавление с использованием устройства манипуляции многочленами, осуществляющего обфускацию числа к коэффициенту многочлена от одной переменной для получения многочлена от одной переменной после обфускации, причем генерируемый локальный материал ключей содержит многочлен от одной переменной после обфускации, и электронное сохранение генерируемого локального материала ключей в сетевом устройстве. Многочлены от двух переменных из множественных (f_i) многочленов от двух переменных могут быть представлены как имеющие коэффициенты по соответствующему личному модулю (p_i).

В более общем случае корневой материал ключей может оцениваться над любым кольцом. Существует возможность использовать многочлены из одного одночлена, такого как Ax^*a*, в случае чего может быть использована группа.

Аспект изобретения касается системы для конфигурирования сетевого устройства для совместного использования ключа, например администрации сети, причем система содержит средство получения материала ключей для получения в электронной форме личного модуля, открытого модуля, который может или может не быть больше личного модуля, и симметрического многочлена от двух переменных, имеющего целочисленные коэффициенты, биты длины ключа двоичного представления открытого модуля минус личный модуль все являются нулевыми битами, возможно младшими значащими битами длины ключа, генератор для генерирования локального материала ключей для сетевого устройства, содержащий средство управления сетевыми устройствами для получения в электронной форме идентификационного номера для сетевого устройства и для электронного сохранения генерируемого локального материала ключей в сетевом устройстве, и устройство манипуляции многочленами для определения многочлена от одной переменной из многочлена от двух переменных путем подстановки идентификационного номера в многочлен от двух переменных, редукции по личному модулю результата подстановки.

Вариант осуществления системы содержит генератор осуществляющего обфускацию числа, например генератор случайных чисел, для генерирования осуществляющего обфускацию числа, устройство манипуляции многочленами сконфигурировано для прибавления осуществляющего обфускацию числа к коэффициенту многочлена от одной переменной для получения многочлена от одной переменной после обфускации, причем генерируемый локальный материал ключей содержит многочлен от одной переменной после обфускации.

Аспект изобретения касается первого сетевого устройства, сконфигурированного для определения ключа совместного использования, причем ключ является криптографическим ключом, причем первое сетевое устройство содержит средство получения локального материала ключей для получения локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен от одной переменной после обфускации, приемник для получения идентификационного номера для второго сетевого устройства, причем

второе сетевое устройство отличается от первого сетевого устройства, устройство манипуляции многочленами для подстановки идентификационного номера второго сетевого устройства в многочлен от одной переменной после обфускации и редукции результата подстановки по открытому модулю, с последующей редукцией по модулю ключа, причем открытый модуль и модуль ключа являются взаимно простыми, устройство выведения ключей для выведения ключа совместного использования из результата редукции по модулю ключа.

Устройство выведения ключей может осуществляться в качестве компьютера, например интегральной цепи, на которой запускаются программные средства, в аппаратных средствах, в их комбинации и т.п., сконфигурированных для выведения ключа совместного использования из результата редукции по модулю ключа.

Выведение ключа совместного использования из результата редукции по модулю ключа может включать в себя применение функции выведения, например функции KDF, определенной в спецификации OMA DRM Открытого мобильного альянса (OMA-TS-DRM-DRM-V2_0_2-20080723-A, раздел 7.1.2 KDF) и подобных функций. Выведение ключа совместного использования может включать в себя отбрасывание одного или нескольких младших значащих битов (перед применением функции выведения ключа). Выведение ключа совместного использования может включать в себя сложение, вычитание или конкатенацию целого числа (перед применением функции выведения ключа).

Множественные сетевые устройства, каждое из которых имеет идентификационный номер и соответствующий локальный материал ключей, могут вместе формировать сеть связи, сконфигурированную для безопасной, например конфиденциальной и/или аутентифицируемой, связи между парами сетевых устройств.

Генерирование ключа основано на ID и обеспечивает возможность генерирования парных ключей между парами устройств. Первое устройство A может полагаться на алгоритм, который получает ключ из локального материала ключей и идентификационного номера.

В одном варианте осуществления первое сетевое устройство отправляет сообщение подтверждения ключа второму сетевому устройству. Например, сообщение подтверждения может содержать шифрование сообщения, и необязательно само сообщение. Второе сетевое устройство может проверять шифрование сообщения. Сообщение может быть фиксированным и присутствовать во втором устройстве, чтобы избежать необходимости в его отправке. Сообщение может быть случайным или временным и т.д., в случае чего оно может быть отправлено вместе с шифрованием. Второе устройство может отвечать сообщением, которое содержит указание, согласуются ли ключи. Второе устройство может также отвечать собственным сообщением подтверждения ключа. Если первое и/или второе устройство обнаруживает, что ключи неравны, они могут начать процесс выравнивания ключей, например, путем стирания младших значащих битов и т.д.

Сетевые устройства и система могут быть электронными устройствами. Сетевые устройства могут быть мобильными сетевыми устройствами.

Способ согласно изобретению может осуществляться на компьютере в качестве компьютерно-реализованного способа или в специализированных аппаратных средствах или в комбинации того и другого. Исполняемый код для способа согласно изобретению может сохраняться в компьютерном программном продукте. Примеры компьютерных программных продуктов включают в себя устройства памяти, оптические устройства хранения, интегрированные цепи, серверы, сетевые программные средства и т.д.

Предпочтительно компьютерный программный продукт содержит невременные средства программного кода, сохраненные на считываемом компьютером носителе, для выполнения способа согласно изобретению, когда упомянутый программный продукт исполняется на компьютере.

В предпочтительном варианте осуществления компьютерная программа содержит средства компьютерного программного кода, выполненные с возможностью выполнения всех этапов способа согласно изобретению, когда компьютерная программа работает на компьютере. Предпочтительно, компьютерная программа осуществлена на считываемом компьютером носителе.

Для полноты международная заявка WO 2010032161, озаглавленная "A method for secure communication in a network, a communication device, a network and a computer program therefor" ("Способ для безопасной связи в сети, устройство связи, сеть и компьютерная программа для них"), упоминается, которая относится к способу для безопасной связи в сетях связи.

Существует некоторое количество различий с этой заявкой, включающих в себя: использование модулярных операций, в частности объединение модулярных операций с другим открытым и личным модулем, повторные модулярные операции, например редукция по открытому модулю, с последующей редукцией по модулю ключа, использование обфускации, использование целых многочленов.

Обеспечены способ конфигурирования сетевого устройства для совместного использования ключа и способ для первого сетевого устройства для определения ключа совместного использования. Способ конфигурирования использует личный модуль (p_1), открытый модуль (N) и многочлен (f_1) от двух переменных, имеющий целочисленные коэффициенты, двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины (b) ключа. Локальный материал ключей для сетевого устройства генерируется путем подстановки идентификационного номера в многочлен от двух переменных и редукции по личному модулю результата подстановки для получения многочлена от одной переменной. Безопасность может быть увеличена путем прибавления (440) одного или нескольких осуществляющих обфускацию чисел к коэффициентам многочлена от одной переменной для получения многочлена от одной переменной после обфускации. В фазе использования сетевое устройство определяет совместно используемый криптографический ключ путем подстановки (530) идентификационного номера другого сетевого устройства в многочлен от одной переменной и редукции по открытому модулю и редукции по модулю ключа.

Краткое описание чертежей

Эти и другие аспекты изобретения очевидны из и будут освещены со ссылками на варианты осуществления, описанные далее. На чертежах

фиг. 1 изображает схематичную структурную схему, иллюстрирующую генератор корневого материала ключей,

фиг. 2 изображает схематичную структурную схему, иллюстрирующую генератор локального материала ключей,

фиг. 3 изображает схематичную структурную схему, иллюстрирующую сеть связи,

фиг. 4 изображает схематическую блок-схему последовательности операций, иллюстрирующую генерирование локального материала ключей,

фиг. 5 изображает схематическую блок-схему последовательности операций, иллюстрирующую генерирование ключа совместного использования,

фиг. 6 изображает схематическую схему последовательности, иллюстрирующую

генерирование ключа совместного использования.

Следует заметить, что элементы, которые имеют одни и те же ссылочные позиции на различных чертежах, имеют одни и те же структурные признаки и одни и те же функции или являются одними и теми же сигналами. Если функция и/или структура такого элемента уже была объяснена, нет необходимости для повторного ее объяснения в подробном описании.

Список ссылочных позиций:

100 средство получения корневого материала ключей

110 элемент открытого модуля

112 элемент степени многочлена

114 элемент длины ключа

116 элемент количества многочленов

122 средство управления личным модулем

124 средство управления симметрическим многочленом от двух переменных

200 генератор локального материала ключей

210 элемент открытого материала

220 элемент личного материала

240 устройство манипуляции многочленами

250 средство управления сетевыми устройствами

260 генератор осуществляющего обфускацию числа

300 сеть связи

310 первое сетевое устройство

320 второе сетевое устройство

330 приемопередатчик

342 устройство манипуляции многочленами

344 средство получения локального материала ключей

346 устройство выведения ключей

348 средство выравнивания ключей

350 криптографический элемент

Описание вариантов осуществления

В то время как настоящее изобретение допускает вариант осуществления во многих различных формах, на чертежах показаны и будут подробно здесь описаны один или несколько конкретных вариантов осуществления с пониманием, что настоящее раскрытие следует расценивать в качестве примера принципов изобретения, не предназначенного для ограничения изобретения конкретными вариантами осуществления, показанными и описанными.

Ниже описан вариант осуществления способа совместного использования ключа. Способ имеет фазу установки и фазу использования. Фаза установки может включать в себя этапы инициации и этапы регистрации. Этапы инициации не задействуют сетевых устройств.

Этапы инициации выбирают системные параметры. Этапы инициации могут выполняться доверенной третьей стороной (TTP). Однако системные параметры могут также расцениваться как заданные в качестве входных данных. В таком случае доверенной третьей стороне нет необходимости их генерировать, и этапы инициации могут быть пропущены. Например, доверенная третья сторона может принимать системные параметры от изготовителя устройства. Изготовитель устройства может выполнить этапы инициации для получения системных параметров. Для удобности пояснения мы будем ссылаться на доверенную третью сторону как на выполняющую

этапы инициации, принимая во внимание, что это не необходимо.

Этапы инициации

Выбирается желаемая длина ключа для ключа, который будет совместно использоваться между устройствами в фазе использования; эта длина ключа обозначается "b". Типичное значение для приложения с низкой безопасностью может быть 64 или 80. Типичное значение для уровня безопасности потребителя может быть 128. В очень секретных приложениях может предпочитаться 256 или даже более высокие значения.

Выбирается желаемая степень; степень управляет степенью некоторых многочленов. Степень будет обозначаться как "*a*", она равна по меньшей мере 1. Частным выбором для a является 2. Более безопасное приложение может использовать более высокое значение a, например 3 или 4 или даже выше. Для простого приложения также возможно a=1. Случай a=1 относится к так называемой "задаче скрытых чисел"; более высокие значения "*a*" относятся к расширенной задаче скрытых чисел, подтверждающей, что эти случаи сложно взломать.

Выбирается количество многочленов. Количество многочленов будет обозначаться "m". Частным выбором для m является 2. Более безопасное приложение может использовать более высокое значение m, например 3 или 4 или даже выше. Следует заметить, что приложение низкой сложности, например, для устройств с ограниченными ресурсами, может использовать m=1.

Более высокие значения параметров безопасности *a* и m увеличивают сложность системы и соответственно увеличивают ее трудность обработки. Более сложные системы сложнее анализировать, и, таким образом, они более устойчивы к криптоанализу.

В одном варианте осуществления открытый модуль N выбирается как удовлетворяющий $2^{(a+2)b-1} \leq N$, и также наиболее предпочтительно $N \leq 2^{(a+2)b}-1$. Эти границы не обязательно необходимы; система может также использовать более малое/большое значение N, хотя это не считается лучшим вариантом.

Часто длина ключа, степень и количество многочленов будут предварительно определенны, например системным проектировщиком, и обеспечены доверенной стороне в виде входных данных. В качестве конкретного выбора можно взять $N=2^{(a+2)b}-1$. Например, если a=1, b=64, то N может быть $N=2^{192}-1$. Например, если a=2, b=128, то N может быть $N=2^{512}-1$. Выбор для N верхней или нижней границы из вышеупомянутого интервала имеет преимущество простого вычисления. Для увеличения сложности можно выбрать случайное число внутри диапазона для N.

Выбирается некоторое количество m личных модулей $p_1, p_2, \ldots, p_m$. Модули являются положительными целыми числами. В течение этапов регистрации каждое устройство будет ассоциировано с идентификационным номером. Каждый выбранный личный модуль больше наибольшего используемого идентификационного номера. Например, можно связать идентификационные номера тем требованием, что они меньше или равны 2^b-1 и что выбранные личные модули больше 2^b-1. Каждое выбранное число удовлетворяет следующей зависимости: $p_j=N+\gamma_j \cdot 2^b$. Где γ_j - целые числа такие, что $|\gamma_j|<2^b$. Один конкретный способ выбора чисел, которые удовлетворяют этому требованию, состоит в выборе набора m случайных целых γ_j так, чтобы $-2^b+1 \leq \gamma_j \leq 2^b-1$, и вычислении выбранных личных модулей из зависимости $p_j=N+\gamma_j \cdot 2^b$. Наличие $|\gamma_j|$, большего на бит, может допускаться, однако может возникать проблема в том, что модулярная операция

заходит так далеко, что ключи совместного использования могут не быть равны.

Для m>1 система более сложна и, таким образом, более безопасна, поскольку операции по модулю для различных модулей комбинируются, даже несмотря на то, что такие операции несовместимы в обычном математическом смысле. По этой причине обеспечивает преимущества выбор выбранных личных модулей в виде попарно различных.

Генерируется некоторое количество m симметрических многочленов от двух переменных $f_1, f_2, \ldots, f_m$ степеней a_j. Все степени удовлетворяют $a_j \leq a$, наиболее предпочтительно $a=MAX\{a_1, \ldots, a_m\}$. Практический выбор состоит в том, чтобы взять каждый многочлен степени a. Многочлен от двух переменных является многочленом с двумя переменными. Симметрический многочлен f удовлетворяет f(x, y)=f(y, x). Каждый многочлен f_j оценивается в конечном кольце, сформированном целыми числами по модулю p_j, полученными путем вычисления по модулю p_j. Целые числа по модулю p_j формируют конечное кольцо с p_j элементами. В одном варианте осуществления многочлен f_j представлен с коэффициентами от 0 до p_j-1. Многочлены от двух переменных могут быть выбраны случайным образом, например путем выбора случайных коэффициентов внутри этих границ. Следует заметить, что некоторые или все из многочленов от двух переменных могут генерироваться асимметричным образом, что приводит к системе с двумя группами. Мы будем предполагать для простоты, что все выбранные многочлены являются симметрическими.

Безопасность совместного использования ключа зависит от этих многочленов от двух переменных, поскольку они являются корневым ключевым материалом системы; так что предпочтительно сильные меры принимаются для их защиты, например процедуры управления, устройства, защищенные от несанкционированного использования, и т.п. Предпочтительно выбранные целые $p_1, p_2, \ldots, p_m$ также хранятся в тайне, включая значение γ_j, соответствующее p_j, хотя это менее критично. Мы будем ссылаться на многочлены от двух переменных также в следующей форме: для j=1, 2, …, m мы записываем $f_j(x, y)=\Sigma_{i=0}^{a} f_{i,j}(x)y^i$.

Вышеупомянутый вариант осуществления может варьироваться некоторым количеством способов. Ограничения на открытые и личные модули может быть выбрано множеством различных способов так, чтобы обфускация многочлена от одной переменной была возможной, но чтобы ключи совместного использования, получаемые в сетевых устройствах, оставались достаточно близки друг к другу достаточно часто. Как объяснено, то, чего будет достаточно, будет зависеть от применения, требуемого уровня безопасности и вычислительных ресурсов, доступных в сетевых устройствах. Вышеупомянутый вариант осуществления комбинирует положительные целые числа так, чтобы модулярные операции, которые осуществляются при генерировании долей многочленов, комбинировались нелинейным образом, когда они складываются над целыми, создавая нелинейную структуру для локального материала ключей, сохраненного на сетевом устройстве. Вышеупомянутый выбор для N и p_j имеет то свойство, что: (i) размер N фиксирован для всех сетевых устройств и привязан к *a*; (ii) нелинейный эффект возникает на старших значащих битах коэффициентов, формирующих материал ключей, сохраненный на устройстве. Ввиду этой конкретной формы ключ совместного использования может генерироваться путем редукции по модулю 2^b после редукции по модулю N.

Эти концепции проектирования могут быть применены более общим образом для улучшения над аспектами (i) и (ii), упомянутыми в предыдущем абзаце. Ниже обеспечиваются другие, общие конструкции для выбора открытых и личных модулей. Для решения первого пункта (i) эта структура для N и p_j удовлетворяет более общему выражению, где мы записываем $p_j=2^X+\gamma_j 2^{Y_j}-1$ так, чтобы для каждого j выполнялось $Y_j+ba_j=X$ и $|\gamma_j|<2^b$. Это выражение обеспечивает возможность для более переменной формы p_j, при этом обеспечивая максимальный эффект при добавлении нелинейных эффектов. Следует заметить, что можно также сделать $Y_j+ba_j\approx X$, где разница между левой и правой сторонами является частью длины ключа.

Для решения второго пункта вышеупомянутые формы для N и p_j удовлетворяют даже более общему выражению, в котором $p_j=\beta 2^X+\gamma_j 2^{Y_j}+\zeta_j 2^{Z_j}$. Путем установления, например, $\zeta_j=-1$, $\beta=1$ и $Z_j=0\ \forall\ j$ мы получаем предыдущее выражение, в котором различные значения γ_j представляют нелинейный эффект в старших значащих битах коэффициентов материала ключей, сохраненного на сетевом устройстве. В этом случае постоянный открытый модуль (N) будет $N=2^X-1$, в то время как личная переменная часть, используемая при генерировании различных положительных целых, участвующих в модулярных операциях, будет $\gamma_j 2^{\gamma j}$. В качестве альтернативы мы можем установить $\gamma_j=1$, $\beta=1$, $Z_j=0$, $Y_j=(a_j+1)b$, $X=(a_j+2)b\ \forall\ j$, в то время как ζ_j различны для различных j так, что $|\zeta_j|<2^b$. В этом случае разницы в ζ_j обеспечивают возможность представить нелинейный эффект в младших значащих битах коэффициентов локального материала ключей, сохраненного на узле. Конструкция открытой части в этом случае также другая и равна $N=\beta_j 2^{Xj}+\gamma_j 2^{Yj}=2^X+2^{b(aj+1)}$, т.е. части, которые остаются постоянными. Следует заметить, что в этом случае нелинейным эффектом является самая низкая часть, и ввиду условия для максимального эффекта смешения, упомянутого ранее, разница между $Y_j-Z_j-\log_2(\zeta_j)$ должна быть $a_j b$. Подобным образом другие конструкции могут быть определены, следуя той же концепции.

Этапы регистрации

На этапе регистрации каждому сетевому устройству назначается ключевой материал (KM). Сетевое устройство ассоциировано с идентификационным номером. Идентификационный номер может быть назначен по требованию, например от TTP, или может уже храниться в устройстве, например, быть сохранен в устройстве при изготовлении, и т.д.

TTP генерирует набор ключевого материала для устройства A следующим образом:

$$KM^A(X)=\sum_{j=1}^{m}<f_j(x,A)>_{p_j}+2^b\sum_{i=0}^{a}\epsilon_{A,i}X^i=\sum_i C_i^A x^i$$

где $KM^A(X)$ - ключевой материал устройства с идентификационным номером A; X - формальная переменная. Следует заметить, что ключевой материал является нелинейным. Обозначение <…>p_j обозначает редукцию по модулю p_j каждого коэффициента многочлена между скобками. Обозначение "$\epsilon_{A,i}$" обозначает случайное целое, которое является примером осуществляющего обфускацию числа, такого, что $|\epsilon_{A,i}|<2^{(a+1-i)b}$. Следует заметить, что любое из случайных целых чисел может быть положительным или отрицательным. Случайные числа ϵ генерируются повторно для

каждого устройства. Обозначение $\sum_{i=0}^{a} \epsilon_{A,i} X^i$, таким образом, представляет многочлен от X степени *a*, у которого длина коэффициентов укорачивается с увеличением степени. В качестве альтернативы более общее, но более сложное условие состоит в том, что $\sum_{i=0}^{a} |\epsilon_{A,i}| \cdot 2^{b+i}$ мало, например <2*a*. Следует заметить, что этап добавления обфускации является необязательным и может опускаться, но предпочтителен для получения более высокого уровня безопасности. Мы будем предполагать, что обфускация используется.

Все другие прибавления могут либо использовать натуральную целую арифметику, либо (предпочтительно) они используют сложение по модулю N. Так что каждая оценка многочленов от одной переменной $\sum_{j=1}^{m} < f_j(x, A) >_{p_j}$ отдельно выполняется по меньшему модулю p_j, но суммирование самих этих редуцированных многочленов от одной переменной предпочтительно выполняется по модулю N. Также прибавление осуществляющего обфускацию многочлена $2^b \sum_{i=0}^{a} \epsilon_{A,i} X^i$ может выполняться с использованием натуральной целой арифметики или, предпочтительно, по модулю N. Ключевой материал содержит коэффициенты C_i^A, где i=0, …, *a*. Ключевой материал может быть представлен в виде вышеприведенного многочлена. На практике ключевой материал может сохраняться в виде списка, например матрицы, целых чисел C_i^A. Устройство A также принимает числа N и b. Манипуляция многочленами может осуществляться, например, как манипуляция матрицами, содержащими коэффициенты, например перечисляющими все коэффициенты в предварительно определенном порядке. Следует заметить, что многочлены могут осуществляться в других структурах данных, например в виде ассоциативной матрицы (также называемой "картой"), содержащей набор пар (степень, коэффициент), предпочтительно так, чтобы каждый коэффициент возникал максимум один раз в наборе. Коэффициенты C_i^A, которые обеспечены устройству, предпочтительно находятся в диапазоне 0, 1, …, N-1.

В случае, когда используется общая конструкция для N и целых чисел p_j, осуществляющий обфускацию многочлен необходимо отрегулировать так, чтобы случайные числа ∈ влияли на различные части коэффициентов. Например, если нелинейный эффект представлен в младших значащих битах коэффициентов материала ключей, сохраненного на сетевых устройствах, то случайные числа должны влиять только на наивысшую часть коэффициентов и переменное количество битов в наинизшей части коэффициентов. Это непосредственное расширение способа, описанного выше, и другие расширения возможны.

Фаза использования

Когда два устройства A и B получили идентификационный номер и приняли свой материал ключей от TTP, они могут использовать свой ключевой материал для получения ключа совместного использования. Устройство A может выполнять следующие этапы для получения своего ключа совместного использования. Сначала устройство A получает идентификационный номер B устройства B, затем A генерирует ключ совместного использования путем вычисления следующего:

$$K_{AB} = << KM^A(x)|_{x=B} >_N >_{2^b} = \ll \sum_i C_i^A B^i >_N >_{2^b}$$

То есть A оценивает свой ключевой материал, имеющий вид целочисленного многочлена, для значения B; результатом оценки ключевого материала является целое число. Далее устройство A делит результат оценки сначала по отрытому модулю N и затем по модулю ключа 2^b. Результат будет называться ключом совместного

использования A, он является целым числом в диапазоне от 0 до 2^b-1. Со своей стороны устройство B может генерировать ключ совместного использования B' путем оценки своего материала ключей для идентификации A и редукции результата по модулю N и затем по модулю 2^b.

В соответствии с вышеприведенным описанием, если используется более общее выражение N и положительных целых p_j, то способ для получения b-битного ключа требует некоторого регулирования. В частности, если нелинейный эффект представлен в самых низких битах коэффициентов материала ключей, сохраненного на сетевых устройствах, в то время как вторым членом в выражении для N является Y_j, то ключ генерируется следующим образом:

$$K_{AB} = < \frac{< KM^A(x)|_{x=B} >_N}{2^{Y_j}} >_{2^b}$$

Поскольку многочлены от двух переменных в корневом материале ключей симметрические, ключ совместного использования A и ключ совместного использования B часто, хотя не обязательно всегда, равны. Конкретные требования на целые числа $p_1, p_2, \ldots, p_m$ и на (необязательные) случайные числа $\in$ таковы, чтобы ключи были часто равны и почти всегда близки друг к другу по модулю два в степени длины ключа. Если A и B получили один и тот же ключ совместного использования, то они могут использовать его в качестве симметричного ключа, который совместно используется между A и B; например, он может быть использован для множества различных криптографических приложений, например они могут обмениваются одним или несколькими сообщениями, зашифрованными и/или аутентифицированными с использованием ключа совместного использования. Предпочтительно алгоритм выведения ключа применяется к ключу совместного использования для дополнительной защиты первичного ключа, например, может применяться хэш-функция.

Если A и B не получили один и тот же ключ совместного использования, то почти обязательно эти ключи близки друг к другу, путем удаления некоторого количества младших значащих битов ключей генерируемые ключи могут почти всегда быть сделаны одинаковыми. A и B могут проверять, равны ли их ключи совместного использования, путем выполнения подтверждения ключей, например A может отправлять к B сообщение, содержащее пару (m, E(m)), где m - сообщение, например фиксированная строка или случайное число, а E(m) - шифрование с использованием ключа совместного использования A.

Путем дешифрования E(m) с использованием ключа совместного использования B, B может проверять, равны ли ключи. Если да, B может отвечать A, информируя его о ситуации.

Если ключи неравны, A и B могут вступать в протокол выравнивания ключей. Например, они могут воспользоваться тем фактом, что два ключа арифметически близки друг к другу. Например, сетевые устройства A и B могут итерационно удалять младший значащий бит и отправлять сообщение подтверждения ключа, пока ключи не станут равны. После получения равных ключей A и B могут выполнять алгоритм выведения ключа для восстановления ключа обычной длины ключа.

Выбранные m личных модулей, $p_1, p_2, \ldots, p_m$, предпочтительно попарно взаимно простые. Если эти числа попарно взаимно простые, увеличивается отсутствие совместимости между операциями по модулю. Получение попарно взаимно простых чисел может быть получено путем выбора целых чисел по порядку, проверки для

каждого нового целого числа, являются ли все пары различных чисел все еще взаимно простыми, если нет, то только что выбранное число удаляется из набора. Эта процедура продолжается, пока все m чисел не будут выбраны.

Сложность увеличивается еще сильнее посредством требования, чтобы выбранные m личных модулей, $p_1, p_2, \ldots, p_m$, были различными простыми числами. В таком случае от каждого простого числа может требоваться иметь форму $p_j=N+\gamma_j \cdot 2^b$. Где γ_j - целые числа такие, что $|\gamma_j|<2^b$. Эксперименты подтвердили, что эти простые числа легко доступны. Например, можно многократно выбирать случайное γ_j и проверять получающееся в результате p_j, пока не будет найдено простое. То же самое применимо, если более общее выражение, как описано выше, применяется. Действительно, из теоремы о распределении простых чисел для арифметических прогрессий следует, что до тех пор, пока *a* примерно того же порядка величины, что и b, в частности для $a<b$, такие простые числа имеются в избытке. В частности, для любой комбинации длины ключа в группе 64, 128, 196, 256 и степени в группе 2, 3 мы подтвердили экспериментально, что множество простых чисел этой формы могут генерироваться с использованием вышеупомянутого алгоритма в конкретных временных пределах. При использовании простых чисел каждый многочлен f_j, таким образом, берется в конечном поле с p_j элементами.

Множество вариантов имеют возможность выбирать различные параметры, используемые в течение фазы использования и регистрации. Например, в упрощенном варианте осуществления личные модули меньше, чем открытый модуль, и удовлетворяют зависимости $p_j=N-\beta_j \cdot 2^b$. Где β_j - положительные целые числа такие, что $\beta_j<2^b$. Один конкретный способ выбора чисел, которые удовлетворяют этому требованию, состоит в выборе набора m случайных положительных целых β_j таких, что $\beta_j<2^b$, и вычислении выбранных личных модулей из зависимости $p_j=N-\beta_j \cdot 2^b$.

Как отмечено, разница между $Y_j-Z_j-\log_2(\zeta_j)$ может быть $a_j b$. Подобным образом могут быть определены другие конструкции, следуя той же концепции. В частности, мы можем записать $p_j=\beta 2^X+\gamma_j 2^{Yj}+\delta 2^W+\zeta_j 2^{Zj}$ для личных модулей и $N=\beta 2^X+\delta 2^W$ для открытого модуля. Частным случаем этой конструкции является $p_j=2^{2(a+1)b}+\gamma_j 2^{(a+1)b}+2^{ab}+\zeta_j$ и $N=2^{2(a+1)b}+2^{ab}$. В этом случае абсолютное значение членов γ_j и β_j меньше 2^b, и они отвечают за создание нелинейного эффекта на MSB и LSB коэффициентов локального сохраненного материала ключей на устройстве. Следует заметить, что, поскольку идентификаторы устройства около b бит в длину, γ_j (β_j) влияет на MSB (LSB) коэффициентов разделяемого многочлена, оцениваемого в кольце целых чисел по модулю p_j. После этого в течение генерирования локального материала ключей для устройства коэффициенты долей многочленов в различных кольцах складываются над целыми так, чтобы источник компонентов был скрыт.

Ключ может генерироваться следующим образом: $K_{AB}=<\frac{<KM^A(x)|_{x=B}>^N}{2^{Y_j}}>_{2^b}$, , но если даже более общее выражение p_j и N используется, которое обеспечивает возможность представления нелинейного эффекта для обоих MSB и LSB, то деление после редукции

по модулю N будет на 2 в степени W, где 2^W - наивысшая целая степень 2, для которой N является кратным. Другие конструкции N и p_j могут требовать деление на другую степень двух. Поскольку многочлены от двух переменных в корневом материале ключей являются симметрическими, ключ совместного использования A и ключ совместного использования B часто, хотя не обязательно всегда, равны.

Подтверждение ключа

Может быть желательно для одного из A и B отправлять сообщение подтверждения ключа другой стороне. Так называемое сообщение подтверждения ключа (KC) обеспечивает возможность адресату сообщения подтверждения ключа проверить, что он вычислил тот же самый ключ, что и отправитель сообщения подтверждения ключа. В частности, в схеме совместного использования ключа, для которой известно, что ключ, установленный обеими сторонами, может отличаться, сообщение подтверждения ключа может быть использовано как в качестве подтверждения, что оба установили один и тот же ключ, так и, если нет, для определения равного ключа совместного использования. Например, в общем случае MAC (код аутентификации сообщений) на основе установленного ключа может выполнять функцию сообщения подтверждения, например HMAC на основе SHA2 или SHA3, или CMAC на основе AES и т.п. Также криптографически сильная хэш-функция может быть использована, например, хэш установленного ключа может быть использован в качестве сообщения подтверждения ключа. Хэш может быть вычислен над самим ключом. MAC может быть вычислен над данными, которые известны B или включены в сообщение подтверждения ключа, например временные, и т.д.

Однако общие криптографически сильные сообщения подтверждения ключа требуют некоторых ресурсов, возможно большего количества ресурсов, чем алгоритм совместного использования ключа согласно вышеописанным принципам. Схемы совместного использования ключа, предоставленные выше, обеспечивают возможность для более простых функций, которые требуют гораздо меньше вычислительных ресурсов, чем универсальные схемы подтверждения ключа.

Устройства A и B вычисляют ключи $K_A(B)$ и $K_B(A)$. Может быть показано следующими математическими зависимостями, что существует целое Δ, зависящее от параметров проектирования, такое, чтобы:

$$K_A(B) \in \{ < K_B(A) + j >_{2^b} \mid -\Delta \leq j \leq \Delta \},$$

Снова $<x>_m$ обозначает целое между 0 и m-1 такое, что $x-<x>_m$ является величиной, кратной m. Определяется функция следующим образом: $h(x)=<x>_{2^r}$, где r - предварительно определенное целое такое, что $2^r \geq 2\Delta+1$. По сравнению с общим вариантом осуществления нет необходимости, чтобы устройства вычисляли возможно сложные хэш-функции; недостаток состоит в том, что некоторая информация о ключе, который используется, отправляется по наблюдаемому каналу связи. Обычно предпочтительно, чтобы сообщение подтверждения ключа не давало утечек ни для какой, или для незначительного количества, информации о ключе, для которого оно вычисляется. Этому недостатку можно противостоять путем деления установленного ключа на 2^r после того, как был найден ключ, который является одним и тем же для обоих A и B. В более общем случае во втором варианте осуществления $h(x)=<x>_v$, где $v \geq 2\Delta+1$ такое, чтобы либо 2^b было величиной, кратной v, либо $<2^b>_v \geq 2\Delta+1$. В обоих

случаях $h(K_A(B))$ может быть использовано A в качестве сообщения подтверждения ключа.

Помимо отправки сообщения подтверждения ключа можно уменьшить разницу между $K_A(B)$ и $K_B(A)$ путем деления обоих ключей на степень 2. $K_A(B)$ и $K_B(A)$ являются b-битными ключами, тогда удаление l младших значащих битов b-битных генерируемых ключей такое, что b-l-битный ключ, который соответствует b-l старшим значащим битам генерируемого ключа, используется для обеспечения безопасности связи. Если b относительно велико (скажем, 100) и l также велико (скажем, 50), вероятность того, что b-l старших значащих битов будут равны, очень высока, т.е. около $1-2\Delta/2^{b-l}$. Этот подход не требует обмена какой-либо информацией, l бит исходного генерируемого ключа удаляются, и получившийся в результате ключ может быть использован для связи. Однако это имеет недостаток, поскольку размер ключа уменьшается, потенциально существенным образом, чтобы обеспечить то, что все устройства в сети будут совместно использовать общий b-l-битный ключ с очень высокой вероятностью.

Следует заметить, что удаление младших значащих битов может комбинироваться с сообщением подтверждения ключа. Например, после удаления l бит сообщение подтверждения ключа вычисляется и отправляется другой стороне. Это подход имеет преимущество в том, что, даже если удаление младших значащих битов не было достаточным для установления общего ключа, станет проще найти такой общий ключ.

В другом подходе проблема потенциально различных ключей, установленных сторонами A и B, состоит в следующем: центральная администрация имеет всю информацию для вычисления заранее, если любые два устройства могут вывести различные ключи. Например, центральная администрация может начать с одного идентификатора A и ключевого материала, вычисленного для A. Устройства добавляются в пул устройств итерационно. Когда новое устройство B' должно быть добавлено в систему, TTP вычисляет ключевой материал для B'. TTP проверяет для каждой комбинации B' и устройств, уже находящихся во множестве, сошлись ли они на одном и том же общем ключе. Например, TTP может проверять, что они нашли один и тот же ключ, непосредственно. TTP может также проверять, что B' и любое другое устройство сойдутся на общем ключе, путем участия в подходящем протоколе согласования ключей для исправления возможного различного ключа; например, путем деления на степень 2 и/или путем отправки одного или нескольких сообщений подтверждения ключа. Ввиду вышеупомянутого вероятностного подхода, очень вероятно, что случайный выбор для B' делает {A,B'} действительным для всех A, если количество устройств A относительно мало.

Если окажется, что B' не придет к общему ключу с каким-то из устройств, уже находящихся во множестве устройств, TTP назначает новый идентификатор для B' или вычисляет новый ключевой материал, но с другими случайными выборами. Хотя проверка этого условия представляет довольно большие непроизводительные издержки, существует возможность относительно малых сетей (скажем, $\sim O(10^4)$ или $O(10^5)$ устройств).

Родственный подход может также применяться в группе устройств. В частности, в некоторых установках не всем устройствам может быть необходимо переговариваться друг с другом, например, если устройства статические и размещены группами (например, в здании). В этом случае проверка, выполняемая посредством TTP, когда добавляется новое устройство B', ограничена проверкой для устройств, принадлежащих к группе, к которой B' будет добавлено. Например, TTP может проверять, генерируют ли все

устройства в некоторой заданной группе ключ, если l LSB ключа удаляются. Следует заметить, что этот способ также обеспечивает возможность для проектирования более улучшенных иерархических схем так, чтобы все устройства принадлежали к основной группе на первом уровне, устройства разделялись на некоторое количество групп на втором уровне, устройства в группе на втором уровне дополнительно разделялись на некоторое количество подгрупп. В такой иерархической организации TTP может проверять, генерируют ли все устройства в некоторой заданной группе на уровне w общий ключ после удаления l_w бит. В такой системе группы на более глубоком уровне могут требовать удаление меньшего количества бит, в то время как группы на высоких уровнях могут требовать удаление большего количества бит для обеспечения генерирования общих ключей.

TTP может выполнять эти проверки всегда, когда новое устройство добавляется, но оно может также предварительно создавать множество идентификаторов устройств и ключевой материал так, чтобы каждая пара идентификаторов из этого множества давала действительный общий ключ.

Например, TTP может накладывать ограничение на пары действительных устройств {A,B}, где пара действительна, если:

$$\left\lfloor \frac{K_B(A)}{2^l} \right\rfloor = \left\lfloor \frac{K_A(B)}{2^l} \right\rfloor ,$$

где l обозначает l бит, соответствующих l младших значащих бит из $K_A(B)$ и $K_B(A)$. Это условие, в общем случае, показывает способ проверки, что ключи, которые действительно будут использованы, равны. Другое условие состоит в том, что новое B принимается, если и только если для всех A, l младших значащих бит из $K_A(B)$ и $K_B(A)$ соответствуют числу в $[\Delta, 2^l - 1 - \Delta]$.

Фиг. 1 изображает схематичную структурную схему, иллюстрирующую генератор 100 корневого материала ключей. Средство получения материала ключей сконфигурировано для обеспечения входных данных, кроме идентификационного номера, необходимого генератору локального материала ключей для генерирования локального материала ключей. Генератор ключей является примером средства получения материала ключей. Вместо генерирования всех или части входных данных некоторые параметры могут также быть получены генератором корневого материала ключей путем их приема; например, средство получения ключей может содержать электронный приемник для приема входных данных, например открытого и личного модуля. Средство получения материала ключей получает все необходимые параметры, кроме идентификационных номеров, от внешнего источника. В одном варианте осуществления *a*, b и m предварительно определяются, например, принимаются, и открытый модуль и личные модули и соответствующие симметрические многочлены от двух переменных генерируются. В одном варианте осуществления также открытый модуль предварительно определяется, например, принимается.

Генератор 100 корневых ключей содержит элемент 112 степени многочлена, элемент 114 длины ключа и элемент 116 количества многочленов, сконфигурированные для обеспечения степени многочлена, длины ключа и количества многочленов, т.е. *a*, b и m соответственно. Хотя эти элементы могут генерироваться, например, в зависимости от обстоятельств, обычно эти параметры выбираются проектировщиком системы. Например, элементы могут проектироваться как энергонезависимые средства памяти, или как приемники для приема элемента значений, или как энергозависимые средства памяти, соединенные с приемником, и т.д. Подходящий выбор включает в себя *a*=2, b=

128, m=2. Любое из чисел может быть увеличено или уменьшено для получения более или менее безопасной системы.

Генератор 100 корневых ключей содержит элемент 110 открытого модуля, сконфигурированный для обеспечения открытого модуля N. Открытый модуль может быть или не быть выбран системным проектировщиком. Например, открытый модуль может быть установлен удобным числом, обеспечивающим возможность быстрой редукции (близкой или равной степени двух). Открытый модуль выбирается внутри диапазона, определенного элементами 112 и 114.

Генератор 100 корневых ключей содержит средство 122 управления личным модулем, сконфигурированное для обеспечения личного модуля p или множественных личных модулей $p_1, \ldots, p_m$. Например, они выбираются случайным образом внутри надлежащих границ.

Генератор 100 корневых ключей содержит средство 124 управления симметрическим многочленом от двух переменных, сконфигурированное для обеспечения симметрического многочлена от двух переменных f или множественных симметрических многочленов от двух переменных $f_1, \ldots, f_m$. Каждый симметрический многочлен от двух переменных выбирается со случайными коэффициентами по соответствующему личному модулю, т.е. личный модуль, имеющий тот же самый индекс. Коэффициенты могут быть выбраны внутри диапазона от 0 до p-1, и могут быть выбраны случайным образом.

Личные модули могут быть выбраны путем прибавления или вычитания величины, кратной двум в степени длины ключа, к открытому модулю. Это даст в результате личные модули так, чтобы разница с открытым модулем в результате становилась последовательностью последовательных нулей. Можно также выбрать открытый модуль и один или несколько личных модулей так, чтобы последовательность последовательных нулей длины ключа происходила не в конце, а в другой позиции, например в позиции "s", если отсчитывать от младшего значащего бита.

Фиг. 2 изображает схематичную структурную схему, иллюстрирующую генератор 200 локального материала ключей. Генератор 100 материала ключей и генератор 200 локального материала ключей вместе формируют систему для конфигурирования сетевого устройства для совместного использования ключа.

Генератор 200 локального материала ключей содержит устройство 240 манипуляции многочленами. Генератор 200 локального материала ключей содержит элемент 210 открытого материала для обеспечения открытых параметров a, N устройству 240 манипуляции многочленами. Генератор 200 локального материала ключей содержит элемент 220 личного материала для обеспечения личных параметров p_i, f_i и m устройству 240 манипуляции многочленами. Элементы 210 и 220 могут осуществляться посредством соответствующих элементов генератора 100 материала ключей; эти элементы могут также быть средствами памяти или шинами для подключения к генератору 100 материала ключей.

Генератор 200 локального материала ключей содержит генератор 260 осуществляющего обфускацию числа для обеспечения осуществляющего обфускацию числа "$\epsilon_{A,i}$" устройству 240 манипуляции многочленами. Осуществляющее обфускацию число может быть случайным числом, например генерируемым посредством генератора случайных чисел. Генератор 260 осуществляющего обфускацию числа может генерировать множественные осуществляющие обфускацию числа для множественных коэффициентов многочлена от одной переменной. В одном варианте осуществления осуществляющее обфускацию число определяется для каждого коэффициента

многочлена от одной переменной.

Генератор 200 локального материала ключей содержит средство управления сетевыми устройствами 250, сконфигурированное для приема идентификационного номера, для которого локальный материал ключей должен генерироваться, например, от сетевого устройства, и сконфигурирован для отправки локального материала ключей к сетевому устройству, соответствующему идентификационному номеру. Вместо приема идентификационного номера он может также генерироваться, например, как случайное, порядковое или временное число. В последнем случае идентификационный номер отправляется вместе с локальным материалом ключей к сетевому устройству.

Устройство 240 манипуляции многочленами получает, возможно множественные, многочлены от одной переменной путем подстановки идентификационного номера от средства 250 управления в каждый из многочленов от двух переменных и редукции каждого по соответствующему личному модулю. Получающиеся в результате множественные редуцированные многочлены от одной переменной складываются покоэффициентно посредством натурального арифметического сложения. Также прибавляются одно или несколько осуществляющих обфускацию чисел. Предпочтительно результат редуцируется, снова покоэффициентно, по открытому модулю; коэффициенты последнего могут быть представлены в диапазоне от 0 до N-1.

Многочлен от одной переменной после обфускации входит в состав локального материала ключей, соответствующего идентификационному номеру. Если необходимо, открытый модуль, степень и длина ключа также отправляются сетевому устройству.

Фиг. 3 изображает схематичную структурную схему, иллюстрирующую сеть 300 связи, содержащую множественные сетевые устройства; показаны первое сетевое 310 устройство и второе 320 сетевое устройство. Мы будем иллюстрировать первое сетевое 310 устройство. Второе сетевое 320 устройство может быть аналогичным или работать по аналогичным принципам.

Сетевое устройство 310 содержит приемопередатчик 330, объединяющий отправитель и приемник для отправки и приема сообщений в электронном, например цифровом, формате, проводным или беспроводным образом от и к второму 320 сетевому устройству. Возможно, приемопередатчик 330 также используется для приема локального материала ключей от администрации 200 сети. Через приемопередатчик 330 принимается идентификационный номер другого сетевого устройства; на чертеже второго сетевого устройства 320.

Сетевое устройство 310 содержит средство 344 получения локального материала ключей. Средство 344 получения локального материала ключей может осуществляться в виде локальной памяти, например энергонезависимой памяти, такой как флэш-память, для хранения локального материала ключей. Средство 344 получения локального материала ключей может также быть сконфигурировано для получения локального материала ключей от генератора 200, например, посредством приемопередатчика 330. Средство получения локального материала ключей 344 сконфигурировано для обеспечения устройства манипуляции многочленами с необходимыми параметрами.

Сетевое устройство 310 содержит устройство 342 манипуляции многочленами, сконфигурированное для подстановки идентификационного номера второго сетевого устройства в многочлен от одной переменной после обфускации и для выполнения двух редукций над результатом: первой редукции результата подстановки по открытому модулю и второй редукции по модулю ключа. Следует заметить, что даже если множественные личные модули были использованы, потребуется только один открытый

модуль. Следует заметить, что для некоторых комбинаций N и личного модуля деление на степень 2 требуется перед тем, как результат редуцируется по модулю ключа.

Сетевое устройство 310 содержит устройство 346 выведения ключей для выведения ключа совместного использования из результата редукции по модулю ключа. Например, устройство 346 выведения ключей может удалять один или несколько младших значащих бит. Устройство 346 выведения ключей может также применять функцию выведения ключей. Также возможно использовать результат второй редукции без дополнительной обработки.

Сетевое устройство 310 содержит необязательное средство 348 выравнивания ключей. Следует заметить, что может случится так, что ключ совместного использования, выведенный в первом сетевом устройстве, неравен ключу, выведенному во втором сетевом устройстве (на основе идентификационного номера первого сетевого устройства). Если это считается нежелательным, можно следовать протоколу выравнивания ключей.

Сетевое устройство 310 содержит криптографический элемент 350, сконфигурированный для использования ключа совместного использования для криптографического приложения. Например, криптографический элемент 350 может шифровать или аутентифицировать сообщение первого сетевого устройства с ключом совместного использования перед его отправкой второму сетевому устройству, например сообщение состояния. Например, криптографический элемент 350 может дешифровать или проверять подлинность сообщения, принятого от второго сетевого устройства.

Обычно и система 200 для конфигурирования сетевого устройства для совместного использования ключа, и первое сетевое устройство 310, сконфигурированное для определения ключа совместного использования, содержат микропроцессор (не показан), который исполняет надлежащие программные средства, сохраненные на соответственных устройствах, например, программные средства, которые могли быть загружены и сохранены в соответствующей памяти, например RAM (не показана).

Интересный вариант осуществления получается для $a=1$, в особенности в комбинации с более высокими значениями m, например выше 1, 2 или выше, 4 или выше. Требуемая манипуляция над многочленом уменьшается до одного умножения и редукции, обеспечивая в особенности простую реализацию. Однако даже для этого простого случая извлечение исходных многочленов от двух переменных не прямолинейно и становится все более сложным при более высоких значениях m. Хотя никакие реализуемые атаки не известны даже для $a=1$, линейная структура может быть исходной точкой для будущего анализа, так что по этой причине может быть смысл наложить ограничение $a>1$.

Фиг. 4 изображает схематическую блок-схему последовательности операций, иллюстрирующую способ генерирования локального материала ключей 400. Способ содержит получение 410 открытого и личного модуля и симметрического многочлена от двух переменных, получение 420 идентификационного номера сетевого устройства, подстановку 430 идентификационного номера в многочлен от двух переменных по личному модулю, прибавление 440 осуществляющего обфускацию числа к коэффициенту и сохранение 450 многочлена от одной переменной после обфускации в сетевом устройстве.

Фиг. 5 изображает схематическую блок-схему последовательности операций, иллюстрирующую способ генерирования ключа совместного использования 500. Способ содержит получение 510 внешнего идентификационного номера другого сетевого устройства, отправку 520 локального идентификационного номера другому сетевому

устройству, подстановку 530 внешнего идентификационного номера в многочлен от одной переменной после обфускации по открытому модулю, редуцирование 540 по модулю ключа, выведение 550 ключа совместного использования, отправку 560 сообщения подтверждения ключа к другому сетевому устройству, определение 570, подтверждается ли ключ, и криптографическое приложение 580. Если ключ не подтверждается на этапе 570, то способ продолжается на этапе 550 с выведения нового ключа. Например, этап 550 может удалять один дополнительный младший значащий бит каждый раз, когда ключ не подтверждается.

Этапы 550, 560 и 570 вместе формируют протокол выравнивания ключа. Например, на этапе 560 временная величина (nonce) и шифрование временной величины под ключом совместного использования, выводимого на этапе 550, могут быть отправлены второму устройству. На этапе 560 принимается сообщение от второго устройства. Принятое сообщение может просто говорить, что принятое сообщение подтверждения ключа показало, что ключи неравны. Принятое сообщение может также содержать сообщение подтверждения ключа. В последнем случае первое сетевое устройство проверяет сообщение подтверждения ключа и устанавливает, равны ли ключи. Если нет, выводится новый ключ, например путем удаления младшего значащего бита.

Множество различных способов исполнения способа возможны, как будет очевидны специалисту в данной области техники. Например, порядок этапов может варьироваться или некоторые этапы могут исполняться параллельно. Кроме того, между этапами могут быть введены другие этапы способа. Введенные этапы могут представлять уточнения способа, такого как описан здесь, или могут не относиться к способу. Например, этапы 410 и 420 или 510 и 520 могут исполняться, по меньшей мере частично, параллельно. Кроме того, некоторый заданный этап может не быть закончен полностью, прежде чем следующий этап начинается.

Способ согласно изобретению может исполняться с использованием программных средств, которые содержат инструкции для побуждения процессорной системы выполнять способ 400 или 500. Программные средства могут включать в себя только те этапы, которые осуществляются конкретной подсущностью системы. Программные средства могут сохраняться на подходящем носителе данных, таком как жесткий диск, гибкий диск, память и т.д. Программные средства могут быть отправлены в качестве сигнала по проводу, или беспроводным образом, или с использованием сети данных, например Интернета. Программные средства могут быть сделаны доступными для загрузки и/или для удаленного использования на сервере.

Фиг. 6 изображает в схематической форме возможную последовательность сообщений между двумя сетевыми устройствами, устройством A и B, в то время как они генерируют ключ совместного использования. Время течет сверху вниз. На этапе 610 сетевое устройство A отправляет свой идентификационный номер устройству B. На этапе 620 устройство B отправляет свой идентификационный номер и сообщение подтверждения ключа для ключа совместного использования (K1), который оно вывело на основе идентификационного номера A и своего локального материала ключей. На этапе 630 устройство A обнаруживает, что они не сгенерировали один и тот же ключ. Устройство A удаляет один младший значащий бит (например, целое, деленное на 2) для получения ключа K2. На этапе 630 устройство A отправляет новое сообщение подтверждения ключа. Таким образом A и B обмениваются сообщениями подтверждения ключа 640, пока они не сходятся на одном и том же ключе на этапе 650. На этапе 650 устройство A отправляет сообщение подтверждения ключа устройству B. Устройство B смогло проверить, что они сошлись на одном и том же ключе. На этапе 660 оно отправляет

подтверждение этого, это может быть аутентифицированным сообщением или сообщением подтверждения ключа и т.д. На этапе 670 устройство A отправляет сообщение M1, которое зашифровано (например, с использованием AES) и/или аутентифицировано (например, с использованием HMAC) с использованием теперь уже равного ключа совместного использования.

Алгоритм ниже обеспечивает возможное осуществление этого подхода, т.е. протокола для взаимного согласования ключей и выведения ключей сеанса, запускаемого устройством A и устройством B.

Установить l=L
Установить продолжить=ИСТИНА
Установить Длина=b-l
Генерировать b-битный ключ K
Пока(продолжить И (Длина>МИНИМАЛЬНАЯ_ДЛИНА)){
K=K>>l
Выполнить взаимное подтверждение аутентификации с B на основе K
Если подтверждение успешно, то{
продолжить=ЛОЖЬ
}иначе{
Длина=b-l
}

Протокол удаляет некоторое количество бит битовой строки, генерируемой алгоритмом совместного использования ключа, таким как описан здесь, и выполняет подтверждение аутентификации, например, ответ на вопрос. Подтверждение аутентификации может содержать сообщение подтверждения ключа. Если это безуспешно, несколько дополнительных бит удаляются, и так далее, пока подтверждение не выполнится успешно или ключ не станет слишком коротким. Протокол может быть модифицирован некоторым количеством способов, например путем удаления переменного числа бит, зависящего от итерации, или требования всегда фиксированного количества этапов так, чтобы перехватчик, наблюдающий за исполнением протокола, не получил какой-либо информации о длине совместно используемого общего ключа между A и B. Этот подход имеет преимущество в том, что он удостоверяется, что ключи совместного использования настолько длины, насколько это возможно; однако он имеет потенциальный недостаток в том, что он требует некоторого количества обменов для согласования об общем ключе. С другой стороны, для большинства приложений это не будет большой проблемой, поскольку для большинства пар устройств ключи будут равны или отличаться только в нескольких битах, и только редкие пары устройств будут приходить к ключам с относительно высоким количеством различных младших значащих бит. Это следует из свойств генерируемых ключей.

Существуют другие способы прийти к одному и тому же ключу для обоих устройств. Снова мы предполагаем, что устройства A и B вычисляют ключи $K_A(B)$ и $K_B(A)$. Протоколы ниже применяются для любой схемы совместного использования ключа, для которой существует целое Δ, зависящее от проектировочных параметров, такое, чтобы:

$$K_A(B) \in \{< K_B(A) + j >_{2^b} \mid -\Delta \leq j \leq \Delta\}.$$

Например, схемы совместного использования ключа, описанные здесь, имеют это свойство. Генерируемые ключи представляются как b-битные целые. Так что ключи могут расцениваться как элементы из множества $\{0,1,2,\ldots,2^b-1\}$. Например, если Δ=2 и

$K_B(A)=1$, то $K_A(B)$ находится в $\{1,2,3,0,2^b-1\}$ (следует заметить, что $<1-2>_2^b=2^b-1$). Для надлежащим образом выбранных параметров системного проектирования, Δ относительно мало. Изобретение обеспечивает, чтобы один и тот же ключ генерировался всегда, поскольку неудача генерирования общего ключа может быть исправлена.

Согласно этому способу устройство A отправляет устройству B значение функции $h(K_A(B))$. Здесь h - подходящая хэш-функция, например криптографическая хэш-функция. Устройство B вычисляет h(i) для всех i из $\{<K_B(A)+j>_2^b \mid -\Delta\leq j\leq\Delta\}$ и использует для будущей связи целое i, для которого h(i) соответствует принятому значению $h(K_A(B))$. Если Δ слишком велико, устройства A и B могут сначала выводить свои ключи по степени 2 для уменьшения размера Δ.

Следует понимать, что изобретение также распространяется на компьютерные программы, в частности компьютерные программы на или в носителе, выполненном с возможностью применения изобретения на практике. Программа может иметь форму исходного кода, объектного кода, промежуточного ресурса кода и объектного кода, такого как частично скомпилированная форма, или в любой другой форме, подходящей для использования в осуществлении способа согласно изобретению. Вариант осуществления, относящийся к компьютерному программному продукту, содержит считываемые компьютером инструкции, соответствующие каждому из этапов обработки по меньшей мере одного из изложенных способов. Эти инструкции могут подразделяться на подпрограммы и/или сохраняться в одном или нескольких файлах, которые могут быть связаны статически или динамически. Другой вариант осуществления, относящийся к компьютерному программному продукту, содержит считываемые компьютером инструкции, соответствующие каждому из средств по меньшей мере одной из систем и/или продуктов, изложенных здесь.

Следует заметить, что вышеупомянутые варианты осуществления иллюстрируют, а не ограничивают изобретение, и что специалисты в данной области техники смогут спроектировать множество альтернативных вариантов осуществления. В формуле изобретения любые позиционные обозначения, помещенные в скобки, не должны толковаться как ограничивающие пункт формулы. Использование глагола "содержать" и его спряжений не исключает возможности наличия элементов или этапов помимо перечисленных в пункте формулы. Упоминание элемента в единственном числе не исключает возможности наличия множества таких элементов. Изобретение может осуществляться посредством аппаратных средств, содержащих несколько отдельных элементов, и посредством подходящим образом запрограммированного компьютера. В пункте формулы устройства, перечисляющем несколько средств, несколько из этих средств могут осуществляться одним и тем же элементом аппаратных средств. Сам факт того, что конкретные меры перечислены во взаимно различных зависимых пунктах формулы изобретения не указывает на то, что комбинация этих мер не может быть использована для достижения преимущества.

(57) Формула изобретения

1. Способ конфигурирования сетевого устройства для совместного использования ключа, причем способ содержит:

получение (410) в электронной форме личного модуля (p_1), открытого модуля (N) и многочлена (f_1) от двух переменных, имеющего целочисленные коэффициенты, причем двоичное представление открытого модуля и двоичное представление личного модуля

одинаковы в по меньшей мере последовательных битах длины (b) ключа,

генерирование локального материала ключей для сетевого устройства, причем этап генерирования содержит получение (420) в электронной форме идентификационного номера (A) для сетевого устройства и определение с использованием устройства манипуляции многочленами многочлена от одной переменной из многочлена от двух переменных путем подстановки (430) идентификационного номера в многочлен от двух переменных, редукции по личному модулю результата подстановки, и

сохранение (450) электронным образом генерируемого локального материала ключей в сетевом устройстве и сохранение открытого модуля в сетевом устройстве.

2. Способ по п. 1, в котором генерирование локального материала ключей для сетевого устройства содержит генерирование осуществляющего обфускацию числа и прибавление (440), с использованием устройства манипуляции многочленами, осуществляющего обфускацию числа к некоторому коэффициенту многочлена от одной переменной для получения многочлена от одной переменной после обфускации, причем генерируемый локальный материал ключей содержит многочлен от одной переменной после обфускации.

3. Способ по п. 1 или 2, в котором многочлен (f_1) от двух переменных является симметрическим многочленом.

4. Способ по п. 1, в котором младшие значащие биты длины (b) ключа двоичного представления открытого модуля являются такими же, как младшие значащие биты длины (b) ключа личного модуля.

5. Способ по п. 1, дополнительно содержащий

генерирование личного модуля (p_1) с использованием электронного генератора случайных чисел, и/или

генерирование многочлена от двух переменных с использованием электронного генератора случайных чисел путем генерирования одного или нескольких случайных коэффициентов для многочлена от двух переменных.

6. Способ по п. 1, в котором открытый модуль удовлетворяет $2^{(a+2)b-1} \leq N$, где N представляет открытый модуль, а представляет степень многочлена от двух переменных, и b представляет длину ключа.

7. Способ по п. 1, содержащий получение в электронной форме множественных личных модулей (p_i) и множественных многочленов (f_i) от двух переменных, имеющих коэффициенты по модулю p_i, так что существует набор последовательных позиций длины (b) ключа, в котором двоичное представление открытого модуля совпадает с двоичным представлением всех личных модулей,

определение многочлена от одной переменной содержит подстановку идентификационного номера в каждый из множественных многочленов (f_i) от двух переменных, редукцию по личному модулю из множественных личных модулей, соответствующих одному симметрическому многочлену от двух переменных, и сложение множественных результатов множественных редукций.

8. Способ по п. 1, в котором осуществляющее обфускацию число генерируется так, что

$$|\epsilon_{A,i}| < 2^{(a+1-i)b} ,$$

где $\epsilon_{A,i}$ обозначает осуществляющее обфускацию число, i обозначает степень одночлена, соответствующего упомянутому коэффициенту, а представляет степень многочлена от двух переменных, и b представляет длину ключа.

9. Способ для первого сетевого устройства, сконфигурированного способом конфигурирования сетевого устройства для совместного использования ключа по п. 1, для определения ключа совместного использования, причем ключ является криптографическим ключом, причем способ содержит:

получение локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен от одной переменной необязательно после обфускации,

получение (510) идентификационного номера для второго сетевого устройства, причем второе сетевое устройство отличается от первого сетевого устройства,

подстановку (530) идентификационного номера второго сетевого устройства в упомянутый многочлен от одной переменной необязательно после обфускации,

редукцию результата подстановки по открытому модулю и редукцию (540) по модулю ключа, и

вывод (550) ключа совместного использования из результата редукции по модулю ключа.

10. Способ по п. 9, дополнительно содержащий

определение (560, 570), вывели ли первое сетевое устройство и второе сетевое устройство один и тот же ключ совместного использования, и если не вывели, то вывод дополнительного ключа совместного использования из результата редукции по модулю ключа.

11. Способ по п. 9 или 10, дополнительно содержащий деление результата подстановки по открытому модулю на делитель строки нулевых битов, который является степенью двойки, причем делитель строки нулевых битов больше 1.

12. Система для конфигурирования сетевого устройства для совместного использования ключа, причем система содержит:

средство (100) получения материала ключей для получения в электронной форме личного модуля (122, p_1), открытого модуля (110, N) и симметрического многочлена (124, f_1) от двух переменных, имеющего целочисленные коэффициенты, причем двоичное представление открытого модуля и двоичное представление личного модуля одинаковы в по меньшей мере последовательных битах длины (b) ключа,

генератор (200) для генерирования локального материала ключей для сетевого устройства, содержащий

средство (250) управления сетевыми устройствами для получения в электронной форме идентификационного номера (A) для

сетевого устройства и для электронного сохранения генерируемого локального материала ключей в сетевом устройстве и сохранения открытого модуля в сетевом устройстве, и

устройство (240) манипуляции многочленами для определения многочлена от одной переменной из многочлена от двух переменных путем подстановки идентификационного номера в многочлен от двух переменных, редукции по личному модулю результата подстановки.

13. Первое сетевое устройство (310), сконфигурированное для определения ключа совместного использования, как в п. 1, причем ключ является криптографическим ключом, причем первое сетевое устройство содержит:

средство (344) получения локального материала ключей для получения локального материала ключей для первого сетевого устройства в электронной форме, причем локальный материал ключей содержит многочлен от одной переменной необязательно после обфускации,

приемник (330) для получения идентификационного номера для второго сетевого устройства, причем второе сетевое устройство отличается от первого сетевого устройства,

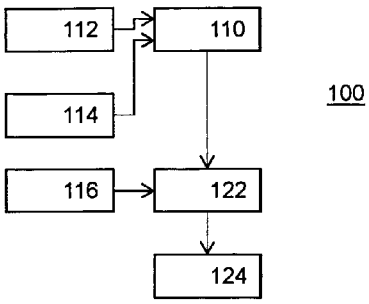
устройство (342) манипуляции многочленами для подстановки идентификационного номера второго сетевого устройства в упомянутый многочлен от одной переменной необязательно после обфускации и редукции результата подстановки по открытому модулю с последующей редукцией по модулю ключа, и

устройство (346) вывода ключа для вывода ключа совместного использования из результата редукции по модулю ключа.

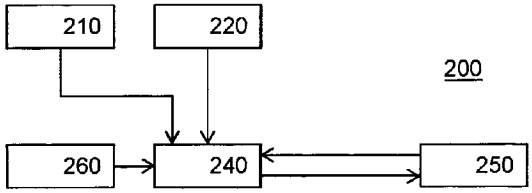
14. Считываемый компьютером носитель, хранящий компьютерную программу, содержащую средство компьютерного программного кода, выполненное с возможностью выполнения всех этапов по любому из пп. 1-11, когда компьютерная программа работает на компьютере.

1

519261

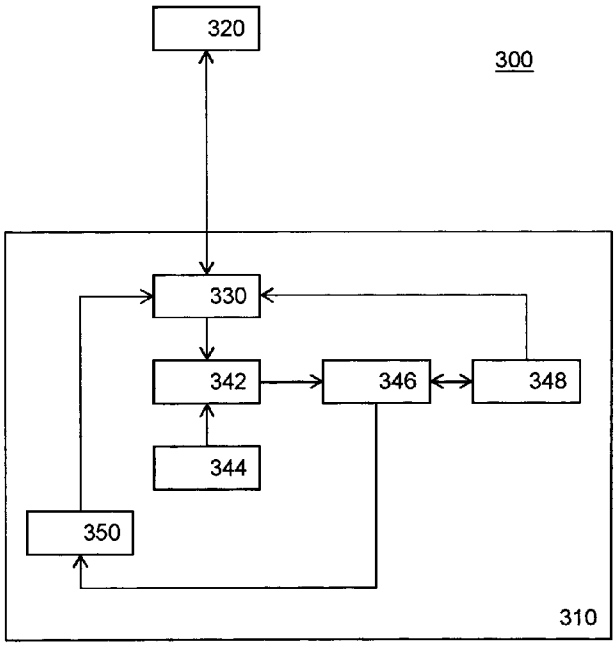


ФИГ.1



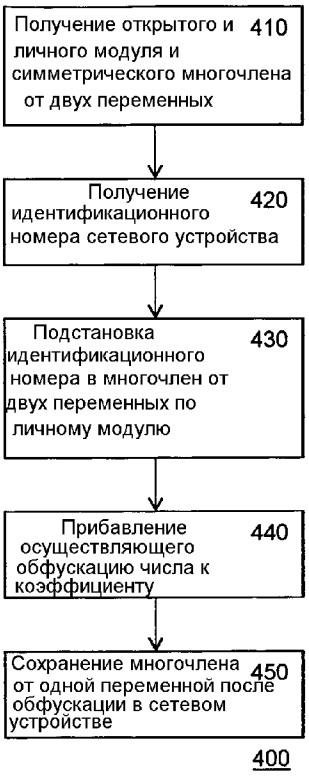
ФИГ.2

2

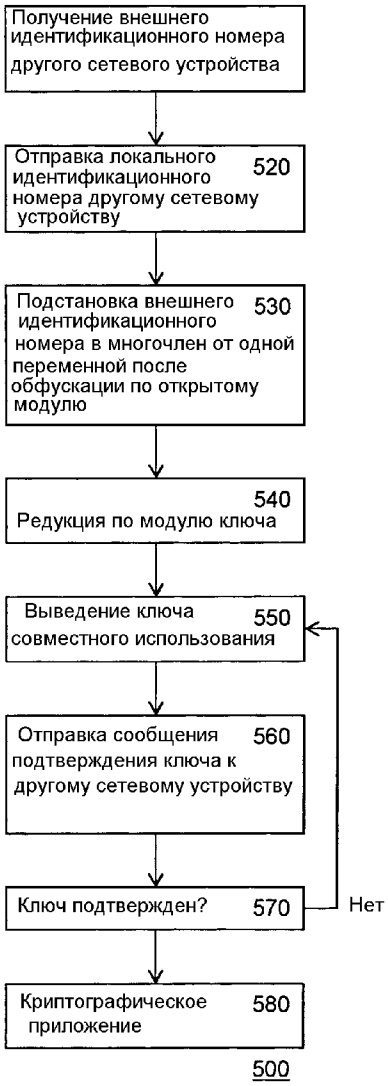


ФИГ.3

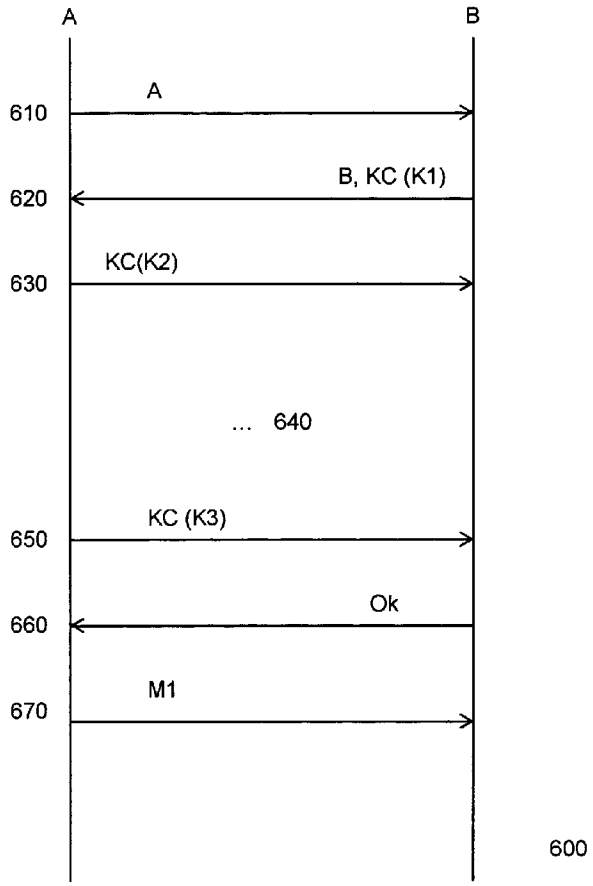
3


Получение открытого и личного модуля и симметрического многочлена от двух переменных 410
Получение идентификационного номера сетевого устройства 420
Подстановка идентификационного номера в многочлен от двух переменных по личному модулю 430
Прибавление осуществляющего обфускацию числа к коэффициенту 440
Сохранение многочлена от одной переменной после обфускации в сетевом устройстве 450
400

ФИГ.4


Получение внешнего идентификационного номера другого сетевого устройства
Отправка локального идентификационного номера другому сетевому устройству 520
Подстановка внешнего идентификационного номера в многочлен от одной переменной после обфускации по открытому модулю 530
Редукция по модулю ключа 540
Выведение ключа совместного использования 550
Отправка сообщения подтверждения ключа к другому сетевому устройству 560
Ключ подтвержден? 570
Нет
Криптографическое приложение 580
500

ФИГ.5



ФИГ.6