РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 320 008** (13) **C2**

(51) МПК
G06F 21/20 (2006.01)
H04L 9/32 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2003112059/09, 24.04.2003**

(24) Дата начала отсчета срока действия патента:
24.04.2003

(30) Конвенционный приоритет:
29.04.2002 (пп.1-19) US 10/134,780

(43) Дата публикации заявки: **10.12.2004**

(45) Опубликовано: **20.03.2008 Бюл. № 8**

(56) Список документов, цитированных в отчете о поиске: **RU 2179738 A2, 20.02.2002. US 5633933 A, 27.05.1997. US 6249873 B1, 19.06.2001. GB 2355886 A, 02.05.2001.**

Адрес для переписки:
129010, Москва, ул. Б. Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(72) Автор(ы):
ГУПТА Рохит (US),
ГАВРИЛЕСКУ Александру (US),
МИЛЛЕР Джон Л. (US),
УИЛЕР Грэхэм А. (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ КОРПОРЕЙШН (US)

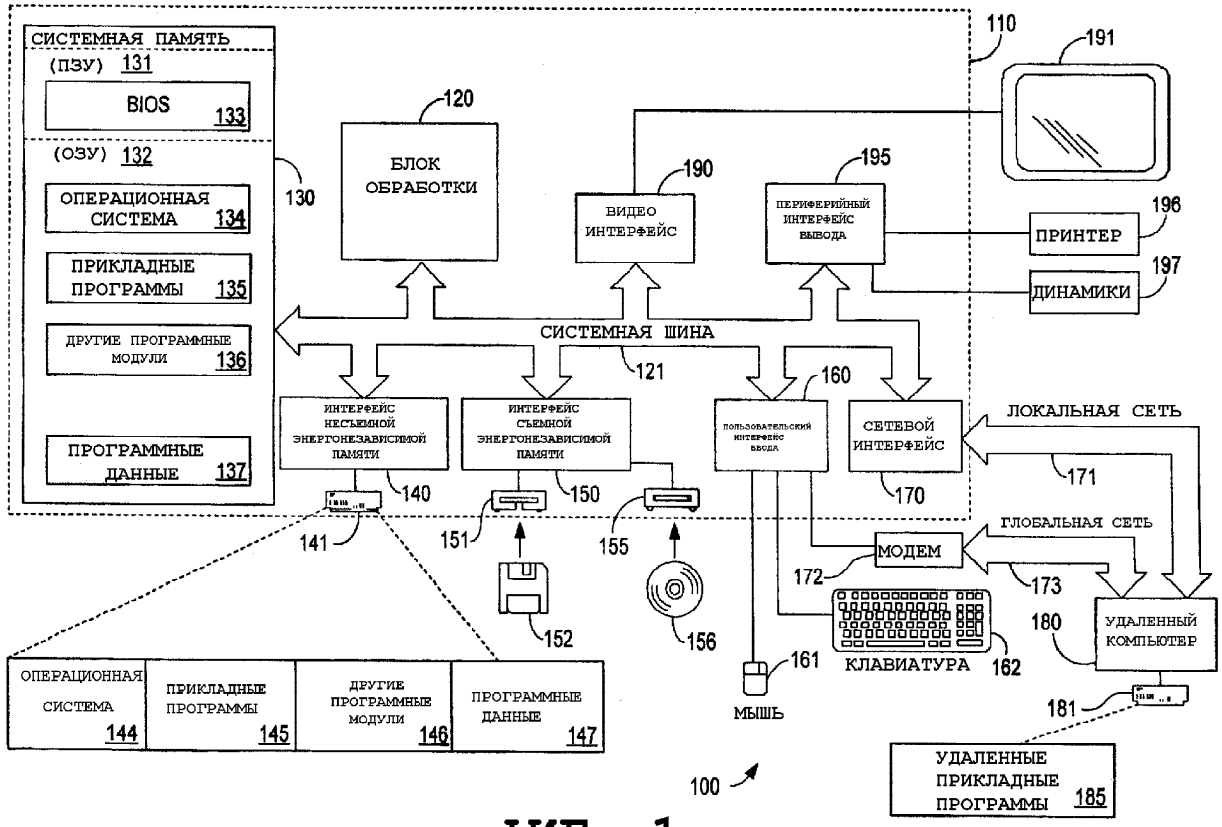
(54) ЗАЩИТНАЯ ИНФРАСТРУКТУРА И СПОСОБ ДЛЯ ПРОТОКОЛА РАЗРЕШЕНИЯ РАВНОПРАВНЫХ ИМЕН (PNRP)

(57) Реферат:

Изобретение относится к протоколам взаимодействия равноправных объектов сетевой структуры и, в частности, касается защитных инфраструктур для протоколов взаимодействия равноправных объектов. Техническим результатом является создание инфраструктуры защиты для системы с одноранговой сетевой структурой. Представлены способы, которые подавляют способность злонамеренного узла нарушать нормальную работу одноранговой сети. Способы изобретения позволяют узлам использовать как защищенные, так и незащищенные данные об идентичности, обеспечивая их самопроверку. Когда это необходимо или удобно, проверяется принадлежность ID путем «вкладывания» процедуры проверки достоверности в существующие сообщения. Вероятность подсоединения к злонамеренному узлу изначально уменьшается благодаря случайному выбору узла, с которым устанавливается соединение. Кроме того, идентифицируется информация от злонамеренных узлов, которая может быть отброшена путем поддержания информации о предыдущих соединениях, которые потребуют ответа в будущем. 4 н. и 15 з.п. ф-лы, 6 ил.

RU 2320008 C2

RU 2320008 C2



ФИГ. 1

RU 2320008 C2

RU 2320008 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 320 008** (13) **C2**

(51) Int. Cl.
G06F 21/20 (2006.01)
H04L 9/32 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2003112059/09, 24.04.2003**

(24) Effective date for property rights: **24.04.2003**

(30) Priority:
29.04.2002 (cl.1-19) US 10/134,780

(43) Application published: **10.12.2004**

(45) Date of publication: **20.03.2008 Bull. 8**

Mail address:
129010, Moskva, ul. B. Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595

(72) Inventor(s):
GUPTA Rokhit (US),
GAVRILESKU Aleksandru (US),
MILLER Dzhon L. (US),
UILER Grehkhehm A. (US)

(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

(54) PROTECTIVE INFRASTRUCTURE AND METHOD FOR PEER NAME RESOLUTION PROTOCOL (PNRP)

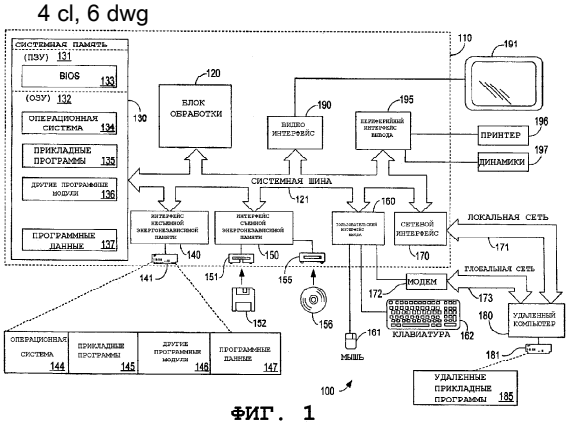
(57) Abstract:

FIELD: protocols for interaction of peer entities of network structure and, in particular, concerns protective infrastructures for protocols of interaction of peer entities.

SUBSTANCE: methods are provided, which suppress capability of malicious node to disrupt normal operation of peer-to-peer network. Claimed methods allow nodes to use both protected and unprotected data about identity, ensuring self-check thereof. Then necessary or comfortable, association of ID is checked by "enclosing" a trustworthiness checking procedure into appropriate messages. Probability of connection to malicious node is initially reduced due to random selection of node with which connection is established. Also, information from malicious nodes is identified and may be discarded by recording information about previous connections, which will require a response in the future.

EFFECT: creation of protection infrastructure for a system with peer-to-peer network structure.

4 cl, 6 dwg



ФИГ. 1

RU 2320008 C2

RU 2320008 C2

Область техники, к которой относится изобретение

Настоящее изобретение относится в целом к протоколам взаимодействия равноправных объектов и, в частности, касается защитных инфраструктур для протоколов взаимодействия равноправных объектов.

Уровень техники

Одноранговая связь, а в действительности все типы связи, зависит от возможности установления правильных соединений между выбранными объектами. Однако объекты могут иметь один или несколько адресов, которые могут изменяться по причине перемещения объектов в сети, из-за изменения топологии либо потому, что не может быть возобновлена аренда адреса. Классическое архитектурное решение проблемы адресации заключается, таким образом, в присвоении каждому объекту постоянного имени и «превращении» (resolve, разрешении) этого имени в текущий адрес, когда необходимо соединение. Это имя для преобразования адреса должно быть очень надежным; также необходимо иметь возможность его легкого и быстрого обновления.

Для повышения вероятности нахождения адреса объекта теми, кто ищет с ним соединения, имеется множество протоколов взаимодействия равноправных объектов, позволяющих объектам сообщать свой адрес, используя различные механизмы. Некоторые протоколы позволяют также клиенту получать информацию об адресах других объектов посредством обработки запросов от других объектов в сети. В действительности получение объектами информации об адресах и позволяет обеспечить успешную работу таких одноранговых сетей. То есть, чем качественнее информация о других равноправных объектах в сети, тем больше вероятность того, что поиск конкретного ресурса окажется суженным.

Однако, если нет надежной защитной инфраструктуры, лежащей в основе протокола взаимодействия равноправных систем, то злонамеренные объекты смогут без труда подорвать способность указанных одноранговых систем обеспечивать суженный поиск. Указанные нарушения могут быть вызваны, например, объектом, который занимается кражей данных об идентичности. При такой атаке на одноранговую сеть, имеющей своей целью кражу данных об идентичности, злонамеренный узел сообщает (публикует) информацию об адресах для идентификаторов (ID), с которыми он не имеет санкционированных отношений, то есть не является ни их владельцем, ни групповым членом и т.п. Злонамеренный объект может также перехватить упомянутые данные и/или, представившись таким образом «добропорядочным» узлом, ответить первым до того, как среагирует действительно добропорядочный узел.

Злонамеренный объект может также затруднить разрешение согласно протоколу PNRP, заполнив сеть вредной информацией, в результате чего другие объекты в сети будут пытаться направлять запросы на несуществующие узлы (что негативно скажется на сходимости результатов поисков), либо на узлы, управляемые атакующим узлом. К аналогичному результату может также привести модификация пакета RESOLVE (превращение, разрешение), используемого для обнаружения ресурсов, перед направлением его дальше, либо посылка неправильного пакета RESPONSE (ответ) обратно запрашивающему объекту, который сформировал пакет RESOLVE. Злонамеренный объект может также попробовать нарушить работу одноранговой сети, попытавшись обеспечить, чтобы результаты поисков не сходились, например, таким образом: вместо того чтобы направить поиск к узлу в его кэш-памяти, который ближе к идентификатору ID, что способствует сходимости поиска, направляет поиск к узлу, который находится дальше от запрашиваемого ID. В альтернативном варианте злонамеренный объект может просто вообще не реагировать на запрос поиска. Разрешение согласно протоколу PNRP может быть дополнительно затруднено злонамеренным узлом, посылающим неправильное сообщение BYE (До свидания!) от имени правильного ID. В результате другие узлы в скоплении узлов удалят этот правильный ID из своей кэш-памяти, уменьшив количество хранящихся в ней достоверных узлов.

Хотя проверка сертификата адреса может снять проблему, связанную с кражей данных об идентичности, указанная проверка не эффективна против атаки второго типа, которая затрудняет превращение по протоколу PNRP. Атакующий узел может продолжать формирование проверяемых сертификатов адресов (либо сформировать их заранее) и заполнять скопление равноправных объектов соответствующими ID. Если любой из этих узлов попытается проверить принадлежность ID, атакующий узел сможет убедить его, что он является владельцем поступающих ID, поскольку в действительности это так и есть. Однако, если атакующий узел контролирует формирование достаточного количества ID, он сможет направить большинство одноранговых поисков к одному из узлов, находящихся под его управлением. В этом случае атакующий узел может достаточно успешно контролировать сеть и управлять ее работой.

Если по протоколу взаимодействия равноправных систем требуется, чтобы вся информация о новых адресах проверялась с целью предотвращения кражи данных об идентичности, обсужденной выше, у злонамеренных объектов появляется возможность атаки третьего типа. Подобная атака, к которой чувствительны одноранговые сети указанных типов, имеет вид атаки типа «отказ от обслуживания (DoS)». Если все узлы, которые узнают о новых записях, попытаются выполнить проверку принадлежности ID, то возникнет всплеск сетевой активности, направленный на владельца, объявившего ID. Используя эту слабость, атакующий объект может поддерживать атаку типа IP DoS, направленную на конкретную цель, сделав эту цель особенно популярной. Например, если злонамеренный объект объявляет в качестве IP идентификаторов ID IP-адрес Web Microsoft, то все узлы в одноранговой сети, которые получают этот объявленный IP, попытаются соединиться с этим IP (IP Web-сервера Microsoft) для проверки подлинности записи. Конечно, сервер Microsoft не сможет проверить принадлежность ID, поскольку эту информацию создал атакующий узел. Однако ущерб уже нанесен. То есть атакующему узлу просто удалось заставить значительное количество участников одноранговой связи атаковать Microsoft.

Другая атака типа DoS, которая подавляет узел или скопление узлов путем истощения одного или нескольких ресурсов, совершается злонамеренным узлом, который посылает большой объем недействительных/подлинных сертификатов равноправных адресов (PAC) на один узел, например, используя пакеты FLOOD/RESOLVE/SOLICIT (наполнение/превращение/требование). Узел, который принимает эти сертификаты PAC, будет расходовать все ресурсы центрального процессора, пытаясь проверить все сертификаты PAC. Аналогично, посылая недействительные пакеты FLOOD/RESOLVE, злонамеренный узел добьется размножения пакетов в скоплении узлов. Иными словами, злонамеренный узел может расходовать полосу пропускания сети для скопления узлов, действующих по протоколу PNRP, используя небольшое количество указанных пакетов, поскольку узел, на который посылаются эти пакеты, будет выдавать ответы, высылая дополнительные пакеты. Злонамеренный узел может также добиться увеличения загрузки полосы пропускания сети, посылая ложные сообщения REQUEST, на которые добропорядочные узлы будут реагировать, заполняя сеть сертификатами PAC, которые имеют больший размер, чем сообщения REQUEST.

Злонамеренный узел может также совершить атаку в скоплении узлов, действующих согласно протоколу PNRP, затрудняя синхронизацию начального узла. То есть для соединения со скоплением узлов, действующих согласно протоколу PNRP, узел пытается подсоединиться к одному из тех узлов, которые уже находятся в скоплении узлов, действующих согласно протоколу PNRP. Если узел попытается соединиться с злонамеренным узлом, он может оказаться под глобальным контролем этого злонамеренного узла. Кроме того, злонамеренный узел может посылать недействительные пакеты REQUEST, когда два добропорядочных узла находятся в процессе синхронизации. Такая атака относится к типу DoS, при которой затрудняется синхронизация, поскольку недействительные пакеты REQUEST будут инициировать в ответ формирование сообщений FLOOD.

Таким образом, имеется потребность в защитных механизмах, которые будут обеспечивать целостность скопления узлов P2P путем предотвращения или ослабления воздействия указанных атак.

Сущность изобретения

Раскрытые в этой заявке концепции изобретения включают в себя новый и улучшенный способ подавления способности злонамеренного узла нарушать нормальную работу одноранговой сети. В частности, настоящее изобретение представляет способы, направленные на подавление атак различных типов, которые могут быть запущены злонамеренным узлом, в том числе: атаки, связанные с кражей данных об идентичности; атаки, связанные с отказом от обслуживания; атаки, которые просто пытаются затруднить превращение (разрешение) адреса в одноранговой сети; а также атаки, которые пытаются затруднить способность нового узла присоединиться к одноранговой сети и участвовать в ее работе.

Представленные защитная инфраструктура и способы разрешают применение как защищенных, так и незащищенных данных об идентичности, используемых узлами, путем их самопроверки. Когда это необходимо либо удобно, принадлежность ID проверяется путем совмещения проверки с существующими сообщениями или, если это необходимо, путем посылки небольшого сообщения с запросом. Вероятность исходного соединения с злонамеренным узлом уменьшается путем рандомизированного выбора узла, с которым выполняется соединение. Кроме того, идентифицируют информацию от злонамеренных узлов, которую можно игнорировать, поддерживая информацию о предыдущих соединениях, на которую потребуется реагировать в будущем. Атаки типа отказа от обслуживания подавляют, давая возможность узлу игнорировать запросы, когда использование его ресурсов превышает заранее установленный предел. Способность злонамеренного узла устранять правильно действующий узел уменьшают благодаря обязательному требованию присвоения удаляемому узлу сертификатов аннулирования.

Согласно одному варианту настоящего изобретения предлагается способ формирования самопроверяемого незащищенного сертификата равноправного адреса (PAC), который не позволяет злонамеренному узлу сообщать защищенные идентифицирующие данные о другом узле в незащищенном PAC в одноранговой сети. Данный способ содержит этапы формирования незащищенного PAC для ресурса, поддающегося обнаружению в равноправной сети. Этот ресурс имеет идентификатор (ID) для взаимодействия с равноправными объектами. Кроме того, способ содержит этап включения унифицированного идентификатора ресурса (URI) в незащищенный PAC, из которого получают ID для взаимодействия с равноправными объектами. Предпочтительно, чтобы URI был представлен в формате «p2p://URI». Идентификатор для взаимодействия с равноправными объектами также может быть незащищенным.

В другом варианте предлагается способ удобной проверки сертификата равноправного адреса в первом узле в равноправной сети. В первом узле используется многоуровневая кэш-память для хранения сертификатов равноправных адресов, а способ содержит этапы заявленного приема сертификата равноправного адреса (PAC) от второго узла и определение того, на каком уровне многоуровневой кэш-памяти должен быть запомнен PAC. Если PAC необходимо запомнить на одном из двух самых низких уровней кэш-памяти, то согласно данному способу PAC размещают в отдельном списке, формируют сообщение INQUIRE (запрос), содержащее ID сертификата PAC, подлежащего проверке, и передают сообщение INQUIRE на второй узел. Если PAC должен храниться на более высоком уровне кэш-памяти, отличном от одного из двух самых низких уровней кэш-памяти, то согласно данному способу PAC запоминают на более высоком уровне кэш-памяти с пометкой «не проверен на действительность». В этом случае PAC будет проверен на действительность первый раз, когда он используется. Согласно данному способу для PAC можно также запросить последовательность сертификатов.

В предпочтительном варианте формирование сообщения INQUIRE содержит этап формирования ID транзакции, который необходимо включить в сообщение INQUIRE. Когда

от второго узла в ответ на сообщение INQUIRE получено сообщение AUTHORITY (полномочия), PAC удаляют из отдельного списка и запоминают на одном из двух самых низких уровней кэш-памяти. Если была запрошена последовательность сертификатов, то проверяют сообщение AUTHORITY, чтобы определить, имеется ли последовательность сертификатов и является ли она действительной. Если это так, то PAC запоминают на одном из двух самых низких уровней кэш-памяти, а если нет, то PAC удаляется. ID транзакции можно также использовать в варианте изобретения для обеспечения сообщения AUTHORITY в ответ на предыдущую передачу.

В другом варианте настоящего изобретения предлагается способ обнаружения узла в одноранговой сети, который уменьшает вероятность подсоединения к злонамеренному узлу. Этот способ содержит этапы: вещание сообщения обнаружения в одноранговой сети без включения каких-либо локально зарегистрированных идентификаторов ID; прием ответа от узла в одноранговой сети; и установление равноправных отношений с этим узлом. В одном варианте прием ответа от узла содержит этап приема ответа по меньшей мере от двух узлов в равноправной сети. В этом случае этап установления равноправных отношений с узлом включает этапы рандомизированного выбора одного по меньшей мере из двух узлов и установление равноправных отношений с узлом, случайно выбранным по меньшей мере из упомянутых двух узлов.

Еще в одном варианте настоящего изобретения предлагается способ подавления атаки типа «отказа от обслуживания» на основе процесса синхронизации в равноправной сети. Этот способ содержит этапы приема сообщения SOLICIT, запрашивающего синхронизацию кэш-памяти у первого узла, имеющего сертификат равноправного адреса (PAC); проверку PAC с целью определения его действительности; и сброс пакета SOLICIT, когда на этапе проверки PAC определено, что PAC недействителен. Предпочтительно, чтобы, если на этапе проверки PAC определено, что PAC действителен, способ дополнительно включал: формирование «временных данных» (данных времени, nonce); шифрование «временных данных» с использованием открытого ключа первого узла; формирование сообщения ADVERTISE (извещение), содержащего зашифрованные «временные данные»; и посылку сообщения ADVERTISE на первый узел. Когда от первого узла получено сообщение REQUEST, то согласно способу проверяют сообщение REQUEST с целью определить, мог ли первый узел расшифровать зашифрованные «временные данные», и обрабатывают сообщение REQUEST, если первый узел был способен расшифровать зашифрованные «временные данные».

Предпочтительно, чтобы этот способ, кроме того, содержал этапы поддержания информации о соединении, конкретно идентифицирующем связь с первым узлом; проверку сообщения REQUEST для подтверждения того, что оно конкретно относится к сообщению ADVERTISE; и отбрасывание сообщения REQUEST, если оно конкретно не относится к сообщению ADVERTISE. В одном варианте изобретения этап поддержания информации о соединении, конкретно идентифицирующий связь с первым узлом, включает этапы вычисления первой позиции двоичного разряда в качестве хэш-функции «временных данных» и данных по идентичности первого узла; и установку бита на первой битовой позиции в векторе битов. Когда это выполнено, этап проверки сообщения REQUEST включает этапы: выделение «временных данных» и данных об идентичности первого узла из сообщения REQUEST; вычисление второй битовой позиции в качестве хэш-функции «временных данных» и данных об идентичности первого узла; проверку вектора битов с целью определения того, имеется ли бит, установленный в соответствии со второй битовой позицией; и указание о том, что сообщение REQUEST конкретно не связано с сообщением ADVERTISE, если проверка вектора битов не обнаружила бит, установленный в соответствии со второй битовой позицией. В альтернативном варианте «временные данные» можно использовать непосредственно в качестве битовой позиции. В этом случае при приеме сообщения REQUEST проверяется битовая позиция, соответствующая вложенным «временным данным». Если она установлена, то сообщение REQUEST считается действительным, и битовая позиция очищается. В противном случае сообщение

REQUEST недействительно, либо имеет место ответная атака, и сообщение REQUEST отбрасывается.

Еще в одном варианте настоящего изобретения способ подавления атаки типа отказа от обслуживания на основе процесса синхронизации в равноправной сети включает этапы: заявленный прием сообщения REQUEST от первого узла; определение того, является ли сообщение REQUEST ответом на предыдущую связь с первым узлом; и отбрасывание (отклонение) сообщения REQUEST, когда сообщение REQUEST не является ответом на предыдущую связь с первым узлом. Предпочтительно, чтобы этап определения того, является ли сообщение REQUEST ответом на предыдущую связь, включал этапы: заявленное выделение «временных данных» и данных об идентичности первого узла из сообщения REQUEST; вычисление битовой позиции в качестве хэш-функции «временных данных» и данных об идентичности; проверку вектора битов, чтобы определить, имеется ли бит, установленный в соответствии с указанной битовой позицией; и указание о том, что сообщение REQUEST не является ответом на предыдущую связь с первым узлом, когда отсутствует бит, установленный в соответствии с указанной битовой позицией. Также предлагается способ подавления атак типа отказа от обслуживания на основе потребления ресурсов в узле в одноранговой сети. Способ включает этапы: прием сообщения от узла в одноранговой сети; проверку текущего использования ресурсов; и отказ от обработки сообщения, когда текущее использование ресурсов превысит заранее установленный предел. При приеме сообщения RESOLVE этап отказа от обработки сообщения включает этап посылки сообщения AUTHORITY на первый узел. Это сообщение AUTHORITY содержит указание о том, что сообщение RESOLVE не будет обработано, поскольку текущее использование ресурсов слишком велико. Когда получено сообщение FLOOD, содержащее сертификат равноправного адреса (PAC), и согласно данному способу определено, что PAC необходимо запомнить на одном и двух самых низких уровней кэш-памяти, то этап отказа от обработки упомянутого сообщения включает этап помещения PAC в отдельнный список для последующей обработки. Если согласно способу определено, что PAC следует запомнить на уровне кэш-памяти, расположенном выше двух самых низких уровней кэш-памяти, то этап отказа от обработки сообщения включает этап отбрасывания сообщения FLOOD.

В другом варианте настоящего изобретения предлагается способ подавления атак типа отказа от обслуживания на основе использования полосы пропускания узла в равноправной сети. Этот способ включает в себя этапы: прием запроса на синхронизацию кэш-памяти от узла в равноправной сети; проверку показателя, указывающего на количество синхронизаций кэш-памяти, выполненных в прошлом; и отказ от обработки запроса на синхронизацию кэш-памяти, когда количество синхронизаций кэш-памяти, выполненных в прошлом, превышает заранее установленный максимум. В еще одном варианте согласно данному способу проверяется показатель, определяющий количество синхронизаций кэш-памяти, выполненных в течение заранее установленного интервала времени в прошлом. В этом варианте этап отказа от обработки запроса включает этап отклонения обработки запроса на синхронизацию кэш-памяти, когда количество синхронизаций кэш-памяти, выполненное за предшествующий период времени, превышает заранее установленный максимум.

Еще в одном варианте настоящего изобретения способ подавления поиска на основе атаки типа отказа от обслуживания в равноправной сети включает этапы: проверку записей известных сертификатов равноправных адресов в кэш-памяти для определения соответствующих узлов, на которые посылается запрос разрешения; случайный выбор одного из подходящих узлов; и посылку запроса на разрешение на случайно выбранный узел. В одном варианте этап случайного выбора одного из соответствующих узлов включает этап вычисления взвешенной вероятности для каждого из подходящих узлов на основе расстояния ID протокола PNRP от целевого ID. Затем определяют вероятность выбора следующего конкретного перехода как величину, обратно пропорциональную расстоянию ID между данным узлом и целевым узлом.

В следующем варианте настоящего изобретения способ подавления поиска на основе атаки типа отказа от обслуживания в одноранговой сети включает этапы: прием сообщения RESPONSE; определение того, является ли сообщение RESPONSE ответом на предыдущее сообщение RESOLVE, и отбрасывание сообщения RESPONSE, когда сообщение RESPONSE не является ответом на предыдущее сообщение RESOLVE. Предпочтительно, чтобы шаг определения того, является ли сообщение RESPONSE ответом на предыдущее сообщение RESOLVE, включал этапы: вычисление битовой позиции в качестве значения хэш-функции информации в сообщении RESPONSE; и проверку вектора битов, чтобы определить, установлен ли в нем бит, соответствующий указанной битовой позиции.

В одном варианте, где сообщение RESPONSE содержит список адресов, способ дополнительно включает этапы: определение того, было ли модифицировано сообщение RESPONSE при попытке помешать разрешению (имени); и отбрасывание сообщения RESPONSE, когда это сообщение RESPONSE было модифицировано при попытке помешать разрешению. Предпочтительно, чтобы этап определения того, было ли модифицировано сообщение RESPONSE при попытке помешать разрешению, включал: вычисление битовой позиции в качестве значения хэш-функции списка адресов в сообщении RESPONSE; и проверку вектора битов, чтобы определить, установлен ли в нем бит, соответствующий указанной битовой позиции.

В другом варианте настоящего изобретения способ, предотвращающий попытку злонамеренного узла устранить правильный узел из одноранговой сети, включает этапы: заявленный (намеренный) прием сертификата аннулирования от правильного узла, имеющего сертификат равноправного адреса (PAC), хранящийся в кэш-памяти; и проверку того, что сертификат аннулирования подписан правильным узлом.

Краткое описание чертежей

Включенные в данное описание сопроводительные чертежи, которые являются его частью, иллюстрируют ряд аспектов настоящего изобретения и вместе с описанием служат для объяснения принципов изобретения. На чертежах:

фиг.1 - блок-схема, иллюстрирующая в целом типовую компьютерную систему, в которой реализовано настоящее изобретение;

фиг.2 - упрощенная блок-схема алгоритма, иллюстрирующая защитные аспекты обработки пакета AUTHORITY согласно варианту настоящего изобретения;

фиг.3 - упрощенная блок-схема алгоритма обработки связи, иллюстрирующая защитные аспекты фазы обнаружения P2P синхронизации согласно варианту настоящего изобретения;

фиг.4 - упрощенная блок-схема алгоритма, иллюстрирующая защитные аспекты обработки пакета RESOLVE согласно варианту настоящего изобретения;

фиг.5 - упрощенная блок-схема алгоритма, иллюстрирующая защитные аспекты обработки пакета FLOOD согласно варианту настоящего изобретения; и

фиг.6 - упрощенная блок-схема алгоритма, иллюстрирующая защитные аспекты обработки пакета RESPONSE согласно варианту настоящего изобретения.

Хотя изобретение описано в связи с конкретными предпочтительными вариантами его осуществления, не следует полагать, что оно ограничено этими вариантами. Наоборот, предполагается, что изобретение распространяется на все альтернативные варианты, модификации и эквивалентные варианты, не выходящие за рамки сущности и объема изобретения, определенных прилагаемой формулой изобретения.

Подробное описание изобретения

Чертежи, на которых одинаковые ссылочные позиции относятся к одинаковым элементам, иллюстрируют изобретение, показанное реализованным в подходящей для него вычислительной среде. Хотя это не является обязательным, изобретение описывается в общем контексте команд, выполняемых компьютером, таких как программные модули, выполняемые персональным компьютером. Обычно программные модули включают подпрограммы, программы, объекты, компоненты, структуры данных и

т.д., которые выполняют частные задачи или реализуют особые абстрактные типы данных. Кроме того, специалистам в данной области техники очевидно, что изобретение можно практически реализовать с помощью других конфигураций компьютерной системы, включая: переносные устройства, многопроцессорные системы, бытовую электронную аппаратуру на базе микропроцессора или программируемую электронную аппаратуру, сетевые персональные компьютеры, миникомпьютеры, универсальные компьютеры и т.п. Изобретение также можно практически реализовать в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки, которые связаны через сеть передачи данных. В распределенной вычислительной среде программные модули могут быть расположены как в местных, так и в удаленных запоминающих устройствах.

На фиг.1 представлен пример подходящей вычислительной системной среды 100, в которой может быть реализовано изобретение. Вычислительная системная среда 100 является лишь одним из примеров подходящей вычислительной среды, который не претендует на какое-либо ограничение, связанное с объемом использования изобретения или его функциональных возможностей. Ни в коем случае не следует считать, что вычислительная среда 100 каким-то образом зависит либо связана какими-либо требованиями, имеющими отношение к любой компоненте или любой комбинации компонент, показанных в типовой рабочей среде 100.

Изобретение может работать вместе с большим числом других вычислительных системных сред или конфигураций общего или специального назначения. Примерами известных вычислительных систем, сред и/или конфигураций, которые могут подойти для использования вместе с изобретением, являются, но не ограничиваются ими: персональные компьютеры, компьютеры-серверы, переносные устройства или «лэптопы», многопроцессорные системы, системы на базе микропроцессоров, компьютерные приставки к телевизорам, программируемая бытовая электронная аппаратура, сетевые компьютеры, миникомпьютеры, универсальные компьютеры, распределенные вычислительные среды, которые содержат любую из вышеуказанных систем или устройств, и т.п.

Изобретение может быть описано в общем контексте команд, выполняемых компьютером, таких как программные модули, выполняемые компьютером. Обычно программные модули включают подпрограммы, программы, объекты, компоненты, структуры данных и т.д., которые выполняют частные задачи или реализуют особые абстрактные типы данных. Изобретение также можно практически реализовать в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки, которые связаны через сеть передачи данных. В распределенной вычислительной среде программные модули могут быть расположены как в местной, так и в удаленной компьютерной запоминающей среде, включающей запоминающие устройства.

Обратимся к фиг.1, где типовая система для реализации изобретения включает вычислительное устройство общего назначения в виде компьютера 110. Компоненты компьютера 110 могут включать, но не ограничиваются ими, блок обработки 120, системную память 130 и системную шину 121, которая связывает различные системные компоненты, включая системную память, с блоком обработки 120. Системная шина 121 может относится к любому из нескольких типов шинных структур, в том числе представляя собой шину памяти или контроллер памяти, периферийную шину и локальную шину, с использованием любой из множества различных шинных архитектур. В качестве примера, но не ограничения, указанные архитектуры включают шину ISA (архитектура шины промышленного стандарта), шину MCA (микроканальная архитектура шины), шину EISA (расширенная архитектура ISA), локальную шину VESA (стандарт высокоскоростной локальной видеошины для персональных компьютеров) и шину PCI (шина для соединения периферийных компонентов), известную также как шина Mezzanine.

Компьютер 110 обычно содержит разнообразную машинно-считываемую среду. Машинно-считываемая среда может представлять собой любые имеющиеся в наличии

носители, которые могут быть доступны компьютеру 110, причем эти носители включают как энергозависимые, так и энергонезависимые носители, съемные и несъемные (стационарные) носители. Например, но не как ограничение, машинно-считваемая среда может включать компьютерную среду для запоминания и среду для связи. Компьютерная запоминающая среда включает в себя энергозависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или с помощью технологии для запоминания информации, такой как машинно-считываемые команды, структуры данных, программные модули или другие данные. Компьютерная запоминающая среда включает, но не только, ОЗУ, ПЗУ, ЭСППЗУ (электрически стираемое программируемое постоянное запоминающее устройство), флэш-память, либо другие технологии памяти, ПЗУ на компакт-дисках (CD ROM), цифровые многофункциональные диски (DVD), либо другие оптические запоминающие устройства на дисках, магнитные кассеты, магнитную ленту, магнитные дисковые запоминающие устройства, либо другие магнитные запоминающие устройства, или любые другие носители, которые можно использовать для запоминания требуемой информации и которые могут быть доступны компьютеру 110. Среда для связи обычно содержит машинно-считываемые команды, структуры данных, программные модули либо другие данные в модулированном информационном сигнале, таком как сигнал несущей, либо другой механизм транспортировки, и содержит любую среду для доставки информации. Термин «модулированный информационный сигнал» означает сигнал, который имеет одну или несколько характеристик, настраиваемых или изменяемых таким образом, чтобы закодировать информацию в сигнале. К примеру, но не только, среда для связи включает проводную среду, такую как проводная сеть или непосредственное проводное соединение, и беспроводную среду, такую как акустическая, радиочастотная, инфракрасная и другая беспроводная среда. В перечень машинно-считываемых сред следует также включить любые комбинации из вышеупомянутых сред.

Системная память 130 включает в себя компьютерную запоминающую среду в виде энергозависимой и/или энергонезависимой памяти, такой как память только для считывания (ПЗУ) 131 и память с произвольным доступом (ОЗУ) 132. Базовая система ввода/вывода 133 (BIOS), содержащая базовые подпрограммы, помогающие пересылать информацию между элементами в компьютере 110, к примеру, во время запуска, обычно хранится в ПЗУ 131. ОЗУ 132 обычно содержит данные и/или программные модули, которые непосредственно доступны блоку обработки данных 120 и/или обрабатываются им в данный момент. Например, но не как ограничение, на фиг.1 показана операционная система 134, прикладные программы 135, другие программные модули 136 и программные данные 137.

Компьютер 110 также может включать другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные запоминающие носители. На фиг.1 только в качестве примера показан накопитель на жестких дисках 141, который считывает или записывает данные на несъемный, энергонезависимый магнитный носитель; накопитель на магнитном диске 151, который выполняет считывание или запись на съемный энергонезависимый магнитный диск 152, и накопитель на оптическом диске 155, который выполняет считывание или запись на съемный, энергонезависимый оптический диск 156, такой как ПЗУ на компакт-диске либо другой оптический носитель. Другие съемные/несъемные, энергозависимые/энергонезависимые запоминающие носители, которые можно использовать в приведенной в качестве примера рабочей среде, включают, но не только, кассеты с магнитной лентой, платы флэш-памяти, цифровые многофункциональные диски, цифровую видеоленту, твердотельное ОЗУ, твердотельное ПЗУ и т.п. Накопитель на жестких дисках 141 обычно подсоединен к системной шине 121 через несъемный интерфейс памяти, такой как интерфейс 140, а накопитель на магнитном диске 151 и накопитель на оптическом диске 155 обычно подсоединены к системной шине 121 через съемный интерфейс памяти, такой как интерфейс 150. Накопители и соответствующие компьютерные запоминающие носители, обсужденные выше и показанные на фиг.1, обеспечивают сохранение машинно-считываемых команд, структур

данных, программных модулей и других данных для компьютера 110. К примеру, на фиг.1 показано, что накопитель на жестких дисках 141 хранит операционную систему 144, прикладные программы 145, другие программные модули 146 и программные данные 147. Заметим, что эти компоненты либо могут совпадать, либо отличаться от операционной системы 134, прикладных программ 135, других программных модулей 136 и программных данных 137. Операционная система 144, прикладные программы 145, другие программные модули 146 и программные данные 147 представлены здесь под другими номерами, указывающими на то, что они являются, как минимум, другими копиями. Пользователь может ввести команды и информацию в компьютер 110 через устройства ввода, такие как клавиатура 162 и указывающее устройство 161, обычно известное как «мышь», шаровой манипулятор или сенсорная панель. Другие устройства ввода (не показаны) могут включать микрофон, джойстик, игровую панель, спутниковую тарелку, сканер и т.п. Эти и другие устройства ввода часто подсоединяют к блоку обработки 120 через пользовательский интерфейс ввода 160, который связан с системной шиной, но может быть подсоединен через другие интерфейсные и шинные структуры, такие как параллельный порт, игровой порт или универсальную последовательную шину (USB). К системной шине 121 через интерфейс, к примеру видеоинтерфейс 190, также подсоединен монитор 191 либо устройство отображения другого типа. Вдобавок к монитору компьютеры также могут иметь и другие периферийные устройства вывода, такие как динамики 197 и принтер 196, которые могут быть подсоединены через периферийный интерфейс вывода 195.

Компьютер 110 может работать в сетевой среде, используя логические соединения с одним или несколькими удаленными компьютерами, таким как удаленный компьютер 180. Удаленный компьютер 180 может быть другим персональным компьютером, сервером, маршрутизатором, сетевым персональным компьютером, равноправным устройством либо другим общим сетевым узлом, причем этот удаленный компьютер обычно включает многие либо все элементы, описанные выше в связи с персональным компьютером 110, хотя на фиг.1 показано только запоминающее устройство 181. Логические соединения, показанные на фиг.1, включают локальную сеть (LAN) 171 и глобальную сеть (WAN) 173, но могут также включать и другие сети. Указанные сетевые среды - обычное явление для офисов, компьютерных сетей предприятий, интрасетей (сети Intranet) и сети Интернет. При использовании сетевой среды LAN персональный компьютер 110 подсоединяют к LAN 171 через сетевой интерфейс или адаптер 170. При использовании в сетевой среде WAN компьютер 110 обычно включает модем 172, либо другое средство для установления соединений через WAN 173, такую как Интернет. Модем 172, который может быть встроенным или внешним, можно подсоединить к системной шине 121 через пользовательский интерфейс ввода 160 либо другое соответствующее устройство. В сетевой среде программные модули, относящиеся к персональному компьютеру 110 или их части, могут храниться в удаленном запоминающем устройстве. Как пример, но не как ограничение, на фиг.1 показаны удаленные прикладные программы 185, размещенные в запоминающем устройстве 181. Очевидно, что показанные сетевые соединения являются лишь примерами и что можно использовать другие средства установления линии связи между компьютерами.

Далее изобретение описывается со ссылками на действия и символические представления операций, которые выполняет один или несколько компьютеров, если не указано иное. Очевидно, что указанные действия и операции, которые иногда называют машинно-выполняемыми, включают работу с электрическими сигналами, представляющими данные в структурированном виде, причем эту работу выполняет компьютерный блок обработки. В результате такой обработки происходит преобразование данных либо их поддержание в ячейках системы памяти компьютера, которая реконфигурирует или, иначе, изменяет работу компьютера способом, хорошо понятным специалистам в данной области техники. Структуры данных, где поддерживаются данные, представляют собой физические ячейки памяти, которые имеют конкретные

характеристики, определяемые форматом данных. Однако, хотя изобретение описывается в вышеупомянутом контексте, это не означает, что оно этим ограничивается; специалистам в данной области техники очевидно, что описанные далее разного рода действия и операции также можно реализовать аппаратными средствами.

Как было указано выше, успешная работа протокола взаимодействия между равноправными системами (P2P) зависит от способности протокола устанавливать действительные соединения между выбранными объектами. Поскольку отдельный пользователь может подсоединиться к сети разными способами и в различных местах, имеющих разные адреса, предпочтительным подходом является присвоение пользователю уникальной идентификационной информации с последующим преобразованием (разрешением) этих данных об идентификационной информации в конкретный адрес с использованием указанного протокола. Такой протокол преобразования равноправных имен (PNRP), согласно которому защитная инфраструктура согласно настоящему изобретению находит конкретное применение, описан в одновременно рассматриваемой заявке №09/942164 «Peer-To-Peer Name Resolution Protocol (PNRP) And Multilevel Cache For Use Therewith», поданной 29 августа 2001 года, сущность и содержание которой целиком включено сюда по ссылке. Однако специалистам в данной области техники из последующего изложения станет очевидным, что защитная инфраструктура и способы настоящего изобретения не ограничиваются конкретным протоколом взаимодействия равноправных объектов, описанным в этой совместно рассматриваемой заявке, но с равным успехом могут быть применены с другими протоколами.

Как раскрыто в упомянутой одновременно рассматриваемой заявке, содержание которой целиком включено сюда по ссылке, протокол преобразования равноправных имен (PNRP) является протоколом преобразования имени в адрес на основе принципа равноправности. Имена являются 256-разрядными числами, называемыми идентификаторами ID протокола PNRP. Адреса состоят из адреса IPv4 (Интернет-протокола версии 4) или IPv6 (Интернет-протокола версии 6), порта и номера протокола. При преобразовании ID протокола PNRP в адрес выдается сертификат равноправного адреса (PAC). Этот сертификат включает целевой ID протокола PNRP, текущий IP адрес, открытый ключ и множество других полей. Объект протокола PNRP называют узлом. Узел может иметь один или несколько локально зарегистрированных идентификаторов ID протокола PNRP. Узел выполняет отображение ID в адрес, обнаруживаемый в протоколе PNRP посредством регистрации. Каждая регистрация включает в себя локально сформированный равноправный сертификат и требует кэш-память PNRP соответствующего вида. Хост-компьютеры, не являющиеся узлами PNRP, могут преобразовывать идентификаторы ID протокола PNRP в IP адреса через шлюз DNS (сервер доменных имен) протокола PNRP. Шлюз DNS PNRP принимает запросы DNS'A' и 'AAAA', выполняет поиск по протоколу PNRP для заданного поднабора имен хост-компьютеров и выдает полученные результаты в виде ответа на запрос DNS.

Как было указано выше, протокол PNRP обеспечивает на равноправной основе механизм, связывающий идентификаторы P2P и ID PNRP с сертификатами (PAC) равноправных адресов. Идентификатор ID P2P представляет собой устойчивый 128-разрядный идентификатор. Идентификаторы ID P2P создаются путем хэширования корректно отформатированного имени P2P. Имеется два типа идентификаторов P2P - защищенный и незащищенный. Защищенный ID P2P представляет собой идентификатор с проверяемой взаимосвязью с открытым ключом. Незащищенный ID P2P - это любой идентификатор, который не имеет защиты. Заданный ID P2P может быть опубликован множеством различных узлов. В протоколе PNRP используется суффикс «место обслуживания» для обеспечения каждого опубликованного объекта уникальным ID PNRP. «Место обслуживания» - это 128-разрядное число, соответствующее уникальной конечной точке сетевого обслуживания. Места обслуживания имеют несколько распознаваемых элементов, но должны рассматриваться как непроницаемые для клиентов PNRP. Место обслуживания имеет два важных свойства. В любой момент только один сокет (технология связи компьютеров в сети) в скоплении объектов соответствует данному месту

обслуживания. При сравнении двух мест обслуживания длина общего префикса для каждого из них является подходящей мерой их близости в сети. Два места обслуживания, которые начинаются с четырех одинаковых битов, находятся друг от друга не дальше, чем два места обслуживания, которые начинаются с трех одинаковых битов.

ID P2P уникально определятся его сцеплением с местом обслуживания. Результирующий 256-разрядный (32 байта) идентификатор называется идентификатором (ID) протокола PNRP. Узлы PNRP регистрируют ID протокола PNRP путем вызова услуг PNRP с помощью имени P2P, полномочий и ряда других параметров. Затем услуги PNRP создают и поддерживают сертификат равноправного адреса (PAC), содержащий представленные данные. Сертификаты PAC, как минимум, включают ID протокола PNRP, период достоверности сертификата, адрес услуги PNRP, открытый ключ и криптографическую подпись, созданную по выбранному содержимому PAC.

Создание и регистрация идентификаторов ID протокола PNRP - это лишь одна часть процесса обслуживания по протоколу PNRP. Выполнение услуг PNRP может быть разделено на четыре фазы. Сначала обнаруживается скопление объектов PNRP. Во время этой фазы новый узел должен найти существующий узел в том скоплении узлов, к которому он хочет присоединиться. Это скопление может быть глобальным скоплением PNRP, локальным скоплением (на уровне предприятия) либо скоплением на локальной линии связи (link local cloud). Как только такой узел найден, наступает вторая фаза присоединения к скоплению PNRP. Как только новый узел нашел существующий узел, он выполняет процедуру SYNCHRONIZE (синхронизация), чтобы получить копию верхнего уровня кэш-памяти существующих узлов. Единый уровень кэш-памяти обеспечивает достаточную основу для того, чтобы новый узел стал участником данного скопления узлов. Как только синхронизация достигнута, может быть начата следующая фаза, а именно активное участие в функционировании скопления узлов. После завершения инициализации узел может принять участие в регистрации и преобразовании ID PNRP. Во время этой фазы равноправный объект также постоянно поддерживает кэш-память. Когда операции с данным узлом завершены, наступает четвертая фаза, а именно его уход из скопления. Узел больше не регистрирует локально зарегистрированные ID протокола PNRP, после чего его работа прекращается.

Протокол PNRP состоит из пакетов девяти различных типов, некоторые из которых были раскрыты выше. Однако следует заметить, что в данной заявке имена пакетов используются только для облегчения понимания их функционального назначения, а не как ограничение вида или формата пакета или самого сообщения. Пакет RESOLVE запрашивает преобразование целевого ID PNRP в PAC. Пакет RESPONSE является результатом завершенного запроса RESOLVE. Пакет FLOOD содержит PAC, предназначенный для кэш-памяти PNRP приемника. Пакет SOLICIT используют для того, чтобы попросить узел PNRP объявить (ADVERTISE) данные о его кэш-памяти высокого уровня. Запрашиваемый пакет ADVERTISE содержит список идентификаторов ID протокола PNRP для сертификатов PAC кэш-памяти верхнего уровня узла. Пакет REQUEST используется для того, чтобы попросить узел заполнить поднабор объявленных (ADVERTISED) сертификатов PAC. Пакет INQUIRE используется для незащищенного запроса узла о том, зарегистрирован ли определенный ID PNRP в этом узле. Для подтверждения местной регистрации ID PNRP используют пакет AUTHORITY. Этот пакет факультативно обеспечивает последовательность сертификатов, помогающую проверить PAC для данного ID. Пакет ACK (подтверждение) подтверждает прием и/или успешную обработку конкретных сообщений. Наконец, пакет REPAIR (восстановление) используют для того, чтобы попытаться объединить скопления, которые могли оказаться расщепленными. Как только узел полностью инициализирован, он может стать участником скопления узлов PNRP, выполнив операции пяти типов. Во-первых, узел может зарегистрировать или не зарегистрировать идентификаторы ID протокола PNRP. Когда ID PNRP зарегистрирован, услуга PNRP создает сертификат равноправного адреса (PAC), связанный с ID протокола PNRP, порт и протокол адреса обслуживания, порт и протокол

адреса PNRP и открытый ключ. Этот PAC вводится в локальную кэш-память, и инициируется пакет RESOLVE с использованием в качестве источника нового PAC, а также [ID PNRP+1] в качестве цели (назначения). Этот пакет RESOLVE обрабатывается несколькими узлами, имеющими идентификаторы ID PNRP, очень похожие на зарегистрированный ID. Каждый приемник пакета RESOLVE добавляет в свою кэш-память сертификат PAC нового узла, объявляя тем самым о новом ID PNRP в скоплении узлов. Если ID протокола PNRP не регистрируется, то создается обновленный PAC с установкой флага «отмена полномочий». Обновленный PAC поступает во все записи на самом нижнем уровне локальной кэш-памяти. Каждый приемник пакета FLOOD проверяет свою кэш-память на предмет старой версии сертификата PAC. Если она найдена, то приемник удаляет этот PAC из своей кэш-памяти. Если PAC удален из самого нижнего уровня кэш-памяти, то приемник, в свою очередь, направляет сообщение об отмене полномочий на те узлы PNRP, которые представлены всеми другими сертификатами PAC на самом нижнем уровне его кэш-памяти.

Узел PNRP может также участвовать в преобразовании ID протокола PNRP. Как обсуждалось в вышеуказанной заявке, содержание которой целиком включено сюда по ссылке, идентификаторы ID протокола PNRP преобразуются в сертификаты PAC путем последовательного направления сообщений RESOLVE ближе к целевому ID PNRP. Когда узел принимает сообщение RESOLVE, он может отклонить это сообщение RESOLVE обратно на предыдущий отрезок сети, ответить предыдущему отрезку сети сообщением RESPONSE либо направить сообщение RESOLVE в узел, чей ID протокола PNRP ближе к целевому ID, чем собственный ID у данного узла. Узел также принимает и направляет пакеты RESPONSE в качестве части операции преобразования. Узел PNRP может также инициировать сообщения RESOLVE от имени локального клиента. Услуга PNRP обеспечивает интерфейс API (интерфейс прикладного программирования), разрешая асинхронные запросы на операцию преобразования. Локальный узел выдает пакеты RESOLVE и, в конце концов, принимает соответствующее сообщение RESPONSE.

Узел PNRP также удовлетворяет запросы на синхронизацию кэш-памяти. После приема пакета SOLICIT узел формирует ответ в виде пакета ADVERTISE, где перечислены идентификаторы ID протокола PNRP на самом верхнем уровне его кэш-памяти. Затем узел-запросчик посылает REQUEST, где перечислены идентификаторы ID протокола PNRP для объявленных сертификатов PAC, которые ему требуются. Затем каждая запрошенная запись кэш-памяти вносится в запрашивающий объект. Наконец, как более подробно обсуждается ниже, PNRP выполняет, кроме того, проверку идентификационной информации. Проверка идентификационной информации представляет собой механизм (устройство) ослабления угрозы, используемый для проверки сертификатов PAC. Проверка идентичности имеет в основном две цели. Во-первых, проверка идентичности гарантирует, что узел PNRP, определенный в сертификате PAC, будет иметь ID протокола PNRP из данного локально зарегистрированного сертификата PAC. Во-вторых, для защищенных идентификаторов ID PNRP (которые описаны ниже) проверка идентичности гарантирует, что сертификат PAC был подписан с использованием ключа с криптографически проверяемой связью с полномочиями в ID протокола PNRP.

Имея новую подтвержденную рабочую информацию о системе PNRP, для которой особенно уместен вариант защитной инфраструктуры настоящего изобретения, ниже рассматриваются защитные механизмы, обеспечиваемые защитной инфраструктурой настоящего изобретения. Эти механизмы предлагаются системой согласно настоящему изобретению для того, чтобы исключить или, как минимум, ослабить последствия различных атак, которые может предпринять злонамеренный узел в скоплении P2P, как обсуждалось выше. Протокол PNRP не имеет ни какого-либо механизма для предотвращения этих атак, ни единого решения по отводу всех этих угроз. Однако защитная инфраструктура по настоящему изобретению минимизирует нарушения, которые могут быть вызваны атакой злонамеренного узла, и может быть включена в протокол PNRP.

Как в случае со многими успешно действующими протоколами P2P можно организовать публикацию информации об объектах с целью их легкого обнаружения. Однако, чтобы обеспечить защиту и целостность для протокола P2P, любая информация об идентичности (идентификационной информации) предпочтительно должна включать в себя прикрепленный сертификат идентичности. Однако устойчивая архитектура защиты должна иметь возможность обрабатывать как защищенные, так и незащищенные объекты. Согласно варианту настоящего изобретения устойчивость обеспечивается благодаря использованию самопроверяющихся сертификатов PAC.

Защищенный сертификат PAC «самопроверяется» путем обеспечения отображения между ID и открытым ключом. В результате никому не позволяется публиковать защищенный сертификат PAC без секретного ключа для подписи данного PAC, что предотвращает множество атак, имеющих своей целью кражу данных об идентичности. Держатель секретного ключа ID использует сертификат для присоединения дополнительной информации к ID, такой как IP адрес, дружественное имя и т.д. Предпочтительно, чтобы каждый узел формировал свою собственную пару «секретный ключ - открытый ключ», хотя это может быть обеспечено уполномоченным поставщиком. Затем открытый ключ вводится как часть идентификатора узла. Только тот узел, который создал пару ключей, имеет секретный ключ, с помощью которого он может доказать, что именно он создал данные об идентичности узла. Таким путем может быть раскрыта, а следовательно, предотвращена кража идентичности.

Общий формат для указанных сертификатов может быть представлен в виде [версия, ID, <информация, относящаяся к ID>, достоверность, алгоритмы, $P_{ISSUER}]K_{ISSUER}$. Действительно, имя P2P/URL является частью базового формата сертификата, показывающей, является ли ID защищенным или незащищенным. При используемом здесь представлении сертификата «версия» - это версия сертификата, ID - идентификатор, подлежащий опубликованию, <информация, относящаяся к ID> представляет информацию, «привязываемую» к ID, «достоверность» представляет интервал достоверности, который указывается с помощью пары дат «от - до», выраженных в единицах всемирного времени (или GMT (среднее время по Гринвичу)). «Алгоритмы» относятся к алгоритмам, используемым для формирования пар ключей и подписей, а P_{ISSUER} - открытый ключ создателя сертификата. Если создатель сертификата является тем же, что и владелец ID, то тогда это будет P_{ID}, или открытый ключ владельца ID. Обозначение K_{ISSUER} означает секретный ключ, соответствующий P_{ISSUER}. Если создатель сертификата является владельцем ID, то тогда это будет K_{ID}, то есть секретный ключ владельца ID.

В предпочтительном варианте <информация, относящаяся к ID> включает в себя адресный кортеж, где можно найти этот ID, и адресный кортеж для PNRP-обслуживания создателя сертификата. В данном варианте сертификат адреса получается следующим [версия, ID, $<адрес>_{ID}$ $<адрес>_{PNRP}$, достоверность, флаг аннулирования, алгоритмы, $P_{ISSUER}]K_{ISSUER}$. В этом расширенном представлении ID - это идентификатор, подлежащий опубликованию, который может представлять собой групповой ID или равноправный ID. <Адрес> - кортеж из адреса Ipv6, порта и протокола. $<Адрес>_{ID}$ - адресный кортеж, привязываемый к ID. $<Адрес>_{PNRP}$ - адресный кортеж PNRP обслуживания (или другой услуги P2P) в механизме создания сертификата. Предпочтительно, чтобы это был адрес PNRP создателя сертификата. Затем он используется другими узлами PNRP для проверки подлинности сертификата. «Достоверность» - это интервал достоверности, выраженный парой дат «От - До». Флаг аннулирования, если он установлен, помечает сертификат аннулирования. P_{ISSUER} - открытый ключ создателя сертификата, а K_{ISSUER} - это секретный ключ, соответствующий P_{ISSUER}. Если создатель сертификата является владельцем ID, то тогда это будет K_{ID}, то есть секретный ключ ID.

В предпочтительном варианте настоящего изобретения для сертификата, являющегося достоверным, должны выполняться следующие условия. Подпись на сертификате должна быть подлинной, и срок действия сертификата не должен закончиться. То есть текущая

дата, выраженная в единицах всемирного времени, должна находиться в диапазоне, заданным полем достоверности. Также значение хэш-функции открытого ключа должно совпадать с ID. Если создатель сертификата является владельцем ID, то тогда необходимо проверить хэширование открытого ключа создателя сертификата в ID (то есть ID=хэш(P_{ID})). Если P_{ISSUER} отличается от P_{ID}, то тогда должна существовать последовательность сертификатов, ведущая к сертификату, помеченному K_{ID}. Такая последовательность удостоверяет связь между создателем сертификата и владельцем ID. Кроме того, в случае, когда для этого класса идентификаторов ID опубликован список аннулированных сертификатов (CRL), и список CRL является доступным, то тогда аутентификатор может удостовериться, что ни один из сертификатов в данной последовательности не вошел в CRL.

Защитная инфраструктура согласно настоящему изобретению также обрабатывает незащищенные сертификаты PAC. Согласно настоящему изобретению незащищенный PAC сам себя проверяет путем введения универсального идентификатора ресурса (URI), из которого получен ID. В действительности, и защищенные, и незащищенные идентификаторы ID содержат URI в сертификате PAC. URI имеет формат «p2p://URI». Это не позволяет злонамеренному узлу опубликовать защищенный ID другого узла с незащищенным PAC.

Защищенная инфраструктура согласно настоящему изобретению также позволяет использовать незащищенные идентификаторы ID. Проблема с указанным незащищенным ID заключается в том, что такие ID очень легко подделать. Злонамеренный узел может опубликовать незащищенный ID любого другого узла. Незащищенные идентификаторы ID также вскрывают слабые места в системе безопасности, где возможны трудности при обнаружении добропорядочного узла. Однако в результате использования URI согласно настоящему изобретению незащищенные ID ни коим образом не могут оказывать влияние на защищенные ID. Кроме того, инфраструктура настоящего изобретения требует, чтобы сертификаты PAC, содержащие незащищенный ID, имели бы тот же формат, что и защищенные PAC, то есть чтобы они содержали открытый ключ и секретные ключи. В результате принудительного использования одной и той же структуры как в незащищенных, так и в защищенных PAC, барьер для формирования сертификатов PAC не снижается. Кроме того, благодаря включению URI в сертификат PAC невозможно вычислительным путем сформировать URI, преобразующийся в конкретный защищенный ID.

Один из вопросов, который при этом возникает, состоит в следующем: когда следует проверять сертификаты PAC исходя из компромисса между повышением уровня защиты скопления объектов P2P и увеличением непроизводительных издержек. Однако сертификат PAC, находящийся в разных пакетах, как обсуждалось выше, должен проверяться одновременно. Такая проверка PAC включает выяснение того, является ли подпись ID подлинной, и проверку того, соответствует ли ID открытому ключу для защищенных ID. Для обеспечения сбалансированного разрешения проблем непроизводительных расходов и уровня безопасности в одном варианте настоящего изобретения сертификаты PAC проверяют до какой бы то ни было обработки пакетов. Это гарантирует, что неправильные данные никогда не будут обрабатываться. Однако имея в виду, что проверка PAC может замедлить обработку пакетов, что может оказаться не приемлемым для пакетов некоторых классов, например пакетов RESOLVE, в альтернативном варианте настоящего изобретения сертификат PAC в таких пакетах не проверяется.

Вдобавок к проверке PAC защитная инфраструктура настоящего изобретения также выполняет проверку принадлежности ID для проверки PAC. Как описано выше, кража данных об идентичности может быть выявлена путем простой проверки сертификата адреса перед использованием этого адреса в PNRP или других протоколах P2P. Такая проверка может повлечь за собой простую проверку того, что ID является значением хэш-функции открытого ключа, включенного в сертификат. Проверка принадлежности может также повлечь за собой выдачу пакета INQUIRE по адресу в данном PAC. Пакет INQUIRE

будет содержать ID, подлежащий проверке, и ID транзакции. Если в этом адресе присутствует ID, то узел должен подтвердить данный запрос INQUIRE. Если ID в адресе отсутствует, то узел не должен подтверждать этот INQUIRE. Если требуется проверить идентичность последовательности сертификатов, узел возвращает полную цепочку сертификатов. Хотя проверка подписи и преобразования ID->URL достаточно сложна и требует значительных ресурсов, когда проверяется цепочка доверительных отношений в поступившей цепочке сертификатов, система по настоящему изобретению избегает использования какой-либо сортировки по протоколу вызов/ответ, что вносит дополнительную сложность для проверки PAC. Кроме того, включение ID транзакции не дает возможности злонамеренному узлу заранее формировать ответ на запросы INQUIRE. Вдобавок этот механизм не требует, чтобы PAC содержал полную последовательность сертификатов.

Проверка принадлежности ID в системе по настоящему изобретению также упрощается путем модификации стандартного пакета RESOLVE таким образом, чтобы он мог к тому же выполнять проверку принадлежности ID. Такой модифицированный пакет RESOLVE содержит идентификатор ID адреса, по которому направляется пакет RESOLVE. Если ID по этому адресу имеется, то он посылает подтверждение ACK, в противном случае посылают ответ NACK (нет подтверждения). Если ID не обрабатывает RESOLVE или если получен ответ NACK, то ID удаляют из кэш-памяти. Таким образом, PAC проверяется без какой-либо сортировки по протоколу вызов/ответ и без посылки какого-либо специального пакета INQUIRE путем фактического совмещения передачи сообщения INQUIRE с RESOLVE. Такой процесс вложения описан ниже со ссылками на фиг.2. Эта процедура помогает избавиться от недействительных или просроченных сертификатов PAC.

Указанная проверка достоверности данных об идентичности происходит в два разных момента времени. Первый раз - когда узел собирается добавить PAC на один из двух самых низких уровней кэш-памяти. Достоверность PAC на двух самых нижних уровнях кэш-памяти является критичной для способности PNRP выполнять преобразование идентификаторов ID PNRP. Выполнение проверки идентичности перед занесением PAC на любой из этих двух уровней сдерживает ряд атак. Принадлежность ID не подтверждается, если PAC должен быть введен в кэш-память любого более высокого уровня из-за оборачиваемости в этих более высоких уровнях. Было определено, что почти 85% всех записей PAC на более высоких уровнях кэш-памяти заменяется, либо их срок действия истекает, прежде чем они когда-либо были использованы. Раз так, то вероятность проявления неблагоприятных последствий от недействительных сертификатов PAC, находящихся на этих более высоких уровнях, достаточно низка, чтобы не требовать выполнение проверки ID при их вводе.

Когда определено, что запись принадлежит одному из двух самых низких уровней кэш-памяти, PAC помещают в отдельный список, пока не сможет быть подтверждена его идентичность. При проверке идентичности первого типа используют сообщение INQUIRE. Такая проверка идентичности подтверждает, что PAC еще является действительным (зарегистрирован) в его исходном узле, и запрашивает информацию, чтобы помочь проверить полномочия исходного узла для опубликования этого PAC. Для поля «флаги» определяется один флаг в сообщении INQUIRE, то есть RF_SEND_CHAIN, который запрашивает приемник послать в ответе AUTHORITY цепочку сертификатов (если она существует). Если приемник сообщения INQUIRE не имеет полномочий публиковать PAC либо если PAC больше в этом месте не зарегистрирован, то приемник просто отбрасывает сообщение INQUIRE. Поскольку локальный узел не принимает соответствующий ответ посредством сообщения AUTHORITY, неправильный PAC никогда не будет введен в его кэш-память, и следовательно, не может оказать неблагоприятное воздействие на работу данного узла в скоплении объектов P2P.

Если приемник сообщения INQUIRE имеет полномочия выдавать PAC, и если он все еще локально зарегистрирован, то тогда этот узел ответит (200) на сообщение INQUIRE сообщением AUTHORITY, как показано на фиг.2. Хотя на фиг.2 это не показано,

принимающий узел в варианте настоящего изобретения проверяет, говорит ли сообщение AUTHORITY о том, что ID еще зарегистрирован в узле, который послал сообщение AUTHORITY. Как только локальный узел определяет 202, что данное сообщение AUTHORITY является ответом на сообщение INQUIRE, он удаляет 204 PAC из отдельного списка. Если был запрос 206 на последовательность (цепочку) сертификатов, то проверяют сообщение AUTHORITY, чтобы установить, присутствует ли последовательность сертификатов и является ли она действительной (208). Если последовательность сертификатов присутствует и является действительной, то тогда PAC вводится в кэш-память и отмечается как действительный 210. В противном случае PAC удаляется 212. Если не было запроса на последовательность сертификатов 206, то тогда PAC просто вводится в кэш-память и отмечается как действительный 210.

Как теперь очевидно, указанное сообщение AUTHORITY используют для подтверждения или отрицания того, что ID протокола PNRP все еще зарегистрирован в локальном узле, и факультативно предоставляется последовательность сертификатов, чтобы дать возможность получателю AUTHORITY проверить право узла опубликовать PAC, соответствующий целевому ID. Вдобавок к сообщению INQUIRE сообщение AUTHORITY может представлять собой правильный ответ на сообщение RESOLVE, как описано ниже. Сообщение AUTHORITY включает в себя различные флаги, которые может установить принимающий узел для указания отрицательного ответа. Одним таким флагом является флаг AF_REJECT_TOO_BUSY, который является единственно правильным в ответ на сообщение RESOLVE. Этот флаг указывает на то, что хост-компьютер слишком занят, чтобы принимать сообщение RESOLVE, и говорит отправителю, что сообщение RESOLVE следует направить куда-либо еще для обработки. Хотя он не способствует проверке идентичности, имеется другой защитный механизм по настоящему изобретению для предотвращения атаки типа DoS, который подробно описан ниже. Флаг AF_INVALID_SOURCE, который является единственно действительным в ответ на сообщение RESOLVE, указывает, что PAC источника в сообщении RESOLVE является недействительным. Флаг AF_INVALID_BEST_MATCH, который также является единственно правильным в ответ на сообщение RESOLVE, указывает на то, что «наиболее совпадающий» PAC в сообщении RESOLVE не является действительным. Флаг AF_UNKNOWN_ID указывает, что заданный «проверенный» ID PNRP не зарегистрирован на этом хост-компьютере. Другие флаги в сообщении AUTHORITY указывают принимающему узлу о том, что запрошенная информация внесена. Флаг AF_CERT_CHAIN указывает, что поступила последовательность (цепочка) сертификатов, которая позволит проверить связь между «проверенным» ID PNRP и открытым ключом, использованным для подписи его PAC. Сообщение AUTHORITY только посылают в качестве подтверждения/ответа либо на сообщение INQUIRE, либо на сообщение RESOLVE. Если в какой-либо момент принято сообщение AUTHORITY вне этого контекста, то оно отбрасывается.

Второй раз проверку идентичности стоит удобно выполнить во время процесса RESOLVE. Как обсуждалось выше, кэш-память PNRP обладает высокой скоростью обновления. Следовательно, большинство записей кэш-памяти переписываются до того, как они будут когда-либо использованы. Поэтому защитная инфраструктура настоящего изобретения не проверяет сертификаты PAC до тех пор, пока и если они в действительности не используются. При использовании PAC для маршрутизации пути RESOLVE система по настоящему изобретению «накладывает» проверку идентичности на верхнюю часть пакета RESOLVE, как было указано выше. RESOLVE содержит ID «следующего перехода», который интерпретируется так же как «целевой ID» в пакете INQUIRE. Затем этот пакет RESOLVE подтверждается с помощью пакета AUTHORITY, так же как это ожидается для пакета INQUIRE, рассмотренного выше. Если идентичность при проверке не подтвердилась, приемник пакета RESOLVE является не тем узлом, в котором уверен отправитель. Тогда пакет RESOLVE направляется куда-нибудь еще, и недействительный PAC удаляется из кэш-памяти.

Этот процесс также показан на фиг.2. Когда узел P протокола PNRP принимает пакет AUTHORITY 200 с полем типа «сообщения заголовка», установленным для RESOLVE 202, принимающий узел проверяет флаги AUTHORITY, чтобы определить, является ли флаг AUTHORITY отрицательным 214, как описано выше. Если в сообщении AUTHORITY установлен хотя бы один отрицательный ответный флаг, то PAC удаляют 216 из кэш-памяти, а пакет RESOLVE направляют куда-нибудь еще. Адрес, на который был послан пакет RESOLVE, добавляется к маршруту RESOLVE и отмечается как REJECTED (отброшен). Затем пакет RESOLVE направляют новому адресату. Если флаг AUTHORITY не является отрицательным, и если была запрошена 218 последовательность сертификатов, то тогда проверяют флаг AF_CERT_CHAIN сообщения AUTHORITY, чтобы узнать, имеется ли последовательность сертификатов. Если она имеется, то принимающий узел должен выполнить операцию проверки последовательности по записанному в кэш-памяти сертификату PAC для ID протокола PNRP, заданного при проверке. Эта последовательность должна быть проверена для того, чтобы убедиться в подлинности всех сертификатов и правильности связи между корневым и листовым элементами последовательности. Значение хэш-функции открытого ключа для корневого элемента последовательности необходимо сравнить, как минимум, с полномочиями в имени P2P сертификатов PAC, чтобы убедиться в их совпадении. Открытый ключ для листового элемента последовательности необходимо сравнить с ключом, используемым для подписи PAC, чтобы убедиться в их совпадении. Если любая из этих проверок даст отрицательный результат, либо если при запросе последовательность сертификатов отсутствует 220, то PAC должен быть удален из кэш-памяти 222, а пакет RESOLVE повторно обработан. Если запрошенная последовательность сертификатов имеется и оказалась подлинной 220, то PAC, соответствующий проверенному ID протокола PNRP, следует отметить как полностью проверенный 224. Если это необходимо, то ID протокола PNRP, адрес обслуживания PNRP и время проверки могут быть извлечены из PAC, а сам PAC удален из кэш-памяти для экономии памяти.

Для иллюстрации проверки идентичности предположим, что узел P запрашивает проверку идентичности для ID PNRP 'T'. Узел N принимает запрос на проверку идентичности. Это может случиться, если узел P принимает пакет INQUIRE с целевым ID= T, либо пакет RESOLVE со следующим переходом (отрезком сети), равным T. Узел N проверяет свой список локально зарегистрированных ID PNRP. Если T нет в этом списке, то тогда проверяется тип принятого пакета. Если он был в пакете INQUIRE, то узел N отбрасывает запрос INQUIRE. По окончании обычных попыток повторной передачи узел P отбросит PAC как недействительный, и будет выполнена обработка. Если это был пакет RESOLVE, то узел N выдает ответ с пакетом AUTHORITY, указывающим на то, что идентификатор ID T локально не зарегистрирован. Затем узел P посылает пакет RESOLVE куда-нибудь еще. Если T имеется в списке ID PNRP узла N, то узел N формирует пакет AUTHORITY и устанавливает значение целевого ID равным T. Если T является незащищенным ID, то тогда узел N посылает пакет AUTHORITY в узел P. Если T является защищенным ID и полномочия для защищенного ID представлены ключом, используемым для подписи PAC, то тогда узел N посылает пакет AUTHORITY в узел P. Если ни одно из этих условий не выполняется, и если установлен флаг RF_SEND_CHAIN, то тогда узел N извлекает последовательность (цепочку) сертификатов, относящуюся к ключу, используемому для подписи PAC, для полномочий для ID PNRP T. Последовательность сертификатов вводится в пакет AUTHORITY, а затем узел N посылает пакет AUTHORITY в узел P. В этот момент, если T является незащищенным ID, то обработка заканчивается. В противном случае узел P проверяет связь между ключом для подписи PAC и полномочиями, используемыми для формирования ID PNRP T. Если проверка дала отрицательный результат, то PAC отбрасывается. Если проверка дала отрицательный результат, а инициирующим сообщением был пакет RESOLVE, то узел P направляет этот пакет RESOLVE куда-либо еще.

Как теперь очевидно из этих двух случаев, когда выполняется проверка принадлежности

данных об идентичности, либо посредством пакета INQUIRE, либо посредством модифицированного пакета RESOLVE, недействительный PAC не сможет распространиться по всему скоплению объектов P2P с использованием FLOOD, и поиски не будут направлены на несуществующие или недействительные идентификаторы ID. Проверка PAC необходима для FLOOD, поскольку, если разрешено распространение пакета FLOOD в сети без проверки, это может привести к атаке типа DoS. «Заселенный» (populated) узел не будет подвергаться проверке принадлежности ID посредством этих механизмов, поскольку его ID будет принадлежать двум самым низким уровням кэш-памяти только у очень небольшого количества узлов.

Как более подробно описано в вышеупомянутой совместно рассматриваемой заявке, PNRP-узел N узнает о новом ID одним из четырех способов. Он может узнать о новом ID через начальное заполнение кэш-памяти соседнего узла. В частности, когда появлется узел P2P, он вступает в контакт с другим узлом скопления P2P и инициирует последовательность синхронизации кэш-памяти. О новом ID соседний узел может также узнать в результате введения новой записи в самую нижнюю часть его кэш-памяти. Например, предположим, что узел N появляется в качестве записи на самом низком уровне кэш-памяти узла M. Когда M узнает о новом идентификаторе ID, если этот ID уместился на самом низком уровне кэш-памяти, он внесет его в другие записи на этом уровне кэш-памяти, относящиеся к узлу N. Узел может также узнать о новом ID в результате запроса на поиск. Источник запроса на поиск вставляет в запрос свой сертификат адреса, и PAC для «наилучшего совпадения» с запросом на поиск пока что также вставляет свой PAC в этот запрос. Таким образом все узлы вместе с путем запроса на поиск обновят свою кэш-память с помощью адреса источника запроса и адреса наилучшего совпадения. Аналогично, узел может узнать о новом ID в результате ответа на поиск. Результат ответа на поиск проходит по поднабору пути запроса в обратном порядке. Узлы вдоль этого пути обновляют свою кэш-память по результату поиска.

Согласно протоколу PNRP при первом своем появлении узел находит соседа. Однако, как описано выше, если первым обнаруженным узлом оказался злонамеренный узел, то новый узел может попасть под контроль этого злонамеренного узла. Для предотвращения или минимизации вероятности такого случая защитная структура настоящего изобретения предлагает два механизма для обеспечения защищенной загрузки узла. Первый механизм состоит в рандомизированном нахождении узлов. Когда узел пытается обнаружить другой узел, который может позволить ему соединиться со скоплением PNRP, в последнем выборе с целью обнаружения другого узла используется групповое вещание/широкое вещание, поскольку это наиболее незащищенный способ обнаружения PNRP. Благодаря природе обнаружения очень трудно отличить нормальный узел от злонамеренного. Следовательно, когда требуется использовать указанный метод группового вещания/широкого вещания, защитная структура согласно настоящему изобретению заставляет узел случайным образом выбирать один из тех узлов, который среагировал на сообщение для широковещательного обнаружения (протокол MARCOPOLO или существующий протокол обнаружения на основе группового вещания, например, SSDP). Выбирая случайный узел, система по настоящему изобретению минимизирует вероятность выбора ненормального узла. Система, предложенная в настоящем изобретении, также выполняет указанное обнаружение узлов без использования каких-либо идентификаторов ID. Неиспользование идентификаторов во время обнаружения узлов в системе настоящего изобретения лишает злонамеренный узел возможностей выбрать в качестве цели своей атаки определенный ID.

Второй механизм начальной загрузки защищенного узла обеспечивается модифицированной фазой синхронизации, в течение которой узел поддерживает вектор битов. Этот механизм на основе модифицированной фазы синхронизации можно лучше всего понять на примере, показанном на упрощенной блок-схеме по фиг.3. Предположим, что Алиса 226 посылает Бобу 230 пакет SOLICIT 228 с вложенным в него своим сертификатом PAC. Если PAC Алисы недействителен 232, то Боб 230 просто отбросит

SOLICIT 234. Если PAC является действительным, то Боб 230 будет поддерживать вектор битов для запоминания состояния этого соединения. При получении указанного пакета SOLICIT Боб 230 формирует 236 «временные данные» и хэширует их с ID PNRP Алисы. Результирующее число будет использовано в качестве индекса в векторе битов, который установит Боб. Затем Боб 230 посылает 238 Алисе 226 ответ в виде сообщения ADVERTISE. Это сообщение ADVERTISE будет содержать PAC Боба и «временные данные», зашифрованные с помощью открытого ключа Алисы, отдельно от другой информации, и будут подписаны Бобом 230. Когда Алиса 226 получает сообщение ADVERTISE, она проверяет подпись и PAC Боба. Если результат проверки отрицательный, то сообщение отбрасывается 241. При подтверждении Алиса 226 расшифровывает 242 «временные данные». Затем Алиса 226 формирует 244 запрос REQUEST, который будет содержать эти «временные данные» и ID PNRP Алисы. Боб 230 обработает 246 этот REQUEST путем хэширования ID PNRP Алисы с «временными данными», посланными в пакете REQUEST. Если 248 в векторе битов, имеющем в качестве индекса указанные хэшированные результаты, установлен бит, то тогда Боб устанавливает биты в исходное состояние и начинает обработку REQUEST 250. В противном случае Боб проигнорирует REQUEST 252, так как возможно, что он представляет собой ответную атаку.

Это делает процесс начального запуска узла защищенным, поскольку данная последовательность не может быть воспроизведена. Это требует минимальных непроизводительных расходов с точки зрения потребляемых ресурсов, в том числе центрального процессора (CPU), сетевых портов и сетевого трафика. Для поддержания информации о состоянии не требуются таймеры, а данные будут посылаться только тем ID, который инициировал синхронизацию. В действительности это модифицированная фаза синхронизации является асинхронной, что позволяет узлу одновременно обрабатывать множество пакетов SOLICIT.

Многие из описанных выше процессов обработки могут быть минимизированы путем управления скоростью обработки пакетов, то есть ограничения потребления ресурсов узла. В основе этого лежит идея, что узел не должен потреблять 100% ресурса своего CPU при попытке обработать пакеты PNRP. Поэтому согласно варианту настоящего изобретения узел может отказаться от обработки некоторых сообщений, когда он поймет, что такая обработка снизит его способность эффективно выполнять свои функции.

Одним из таких сообщений, относительно которых узел может принять решение отказаться от его обработки, является сообщение RESOLVE, полученное от другого узла. Этот процесс показан в упрощенном виде на фиг.4. Как только принимается 254 сообщение RESOLVE, узел проверяет 256, не превышает ли текущая производительность CPU заранее установленный предел. Если CPU слишком загружен обработкой сообщения RESOLVE, то он посылает 258 сообщение AUTHORITY с флагом AF_REJECT_TOO_BUSY, указывающим на отказ обработки запроса, поскольку он слишком загружен. Если CPU не слишком загружен 256, то узел определяет 260, все ли сертификаты PAC в сообщении RESOLVE являются действительными, и отбрасывает 162 сообщение, если обнаружится хотя бы один недействительный сертификат. Если все сертификаты PAC являются подлинными 260, то узел будет обрабатывать 264 сообщение RESOLVE.

Если узел может ответить 266 на сообщение RESOLVE, то он преобразовывает 268 RESOLVE в RESPONSE и посылает его в узел, из которого было получено сообщение RESOLVE. Однако, если целевой ID локально не зарегистрирован, то узел вычисляет 270 битовую позицию в виде значения хэш-функции полей в сообщении RESOLVE и устанавливает соответствующие битовые позиции в векторе битов. Как коротко описано выше, этот вектор битов используют как защитный механизм для предотвращения обработки ошибочных ответных сообщений, когда узел не посылал каких-либо сообщений, на которые ожидается ответ. Узел находит следующий отрезок сети, по которому направляется RESOLVE, с соответствующими модификациями для подтверждения обработки сообщения. Если (272) узел, на который должно быть направлено сообщение RESOLVE, уже был проверен, этот узел просто направляет 276 сообщение RESOLVE на

следующий переход. Если (272) этот выбранный следующий переход еще не был проверен, то узел накладывает 274 запрос на принадлежность ID в пакет RESOLVE и направляет 276 его в этот узел. В ответ на запрос о принадлежности с наложенным ID ожидается, что узел примет описанное выше сообщение AUTHORITY, процесс обработки которого показан на фиг.2. Как показано на фиг.2, если проверка AUTHORITY на этапе 214 не принята, то сертификат PAC узла, на который было направлено сообщение RESOLVE, удаляется 216 из кэш-памяти, и RESOLVE обрабатывается вновь, начиная с этапа 254 (фиг.4.) Другим сообщением, на основе которого узел может изменить решение не обрабатывать его из-за слишком большой загрузки CPU, является сообщение FLOOD. В этом процессе, показанном в упрощенном виде на фиг.5, если (278) новая информация, присутствующая в сообщении FLOOD, поступает на любой из двух самых низких уровней кэш-памяти, то проверяется сертификат PAC, чтобы определить его достоверность 280. Если PAC недействителен, то FLOOD отбрасывается 284. Однако, если PAC действителен 280, то он помещается в отдельный список 282. Записи в отдельном списке выбираются со случайными интервалами и обрабатываются тогда, когда CPU не слишком загружен. Поскольку эти записи собираются вводить на два самых низких уровня кэш-памяти, выполняется как проверка идентификатора ID, так и проверка принадлежности, как обсуждалось выше. Если 278 новая информация, присутствующая в сообщении FLOOD, поступает на высокие уровни кэш-памяти, а CPU слишком загружен, чтобы их обрабатывать 286, то их отбрасывают 288. Если у узла имеются достаточные ресурсы CPU для обработки 286, то проверяется сертификат PAC, чтобы определить его достоверность 290. Если он подлинный, то тогда PAC добавляется в флэш-память 292, в противном случае FLOOD отбрасывается 294.

Запуск узла (синхронизация) является еще одним процессом, который потребляет значительные ресурсы узла, в том числе, но не только, возможности обработки данных CPU, а также полосу пропускания сети. Однако процесс синхронизации необходим для того, чтобы дать возможность новому узлу принять полноценное участие в функционировании скопления узлов P2P. Таким образом, узел будет реагировать на запрос от другого узла для запуска, если он имеет в наличии достаточно ресурсов в данный момент времени. То есть как в случае с только что описанными двумя сообщениями узел может отказаться участвовать в запуске, если степень использования его CPU слишком велика. Однако, поскольку этот процесс потребляет столько ресурсов, злонамеренный узел может все же использовать это путем подачи большого количества указанных последовательностей. Поэтому вариант защитной инфраструктуры настоящего изобретения ограничивает количество синхронизаций узла, которые могут быть выполнены данным узлом, для предотвращения атак такого рода. Это ограничение может быть дополнительно ограничено во времени, так что злонамеренный узел не сможет запретить узлу выполнить указанную синхронизацию повторно в будущем.

Выше обсуждалось множество атак на основе поисков, которые были инициированы или вызваны злонамеренным узлом. Для исключения или минимизации воздействия указанных атак на основе поисков в системе по настоящему изобретению предлагается два механизма. Первый из них - это рандомизация. То есть, когда узел ищет соответствующий следующий переход (отрезок сети, двухточечное соединение) для направления запроса на поиск (RESOLVE), он определяет количество возможных узлов-кандидатов, а затем случайным образом выбирает один идентификатор ID из указанных ID-кандидатов, по которому будет направлено сообщение RESOLVE. В одном варианте для рандомизированного выбора идентифицируют три узла-кандидата. Идентификаторы ID можно выбирать на основе взвешенной вероятности в качестве альтернативы глобальной рандомизации. Один такой способ вычисления взвешенной вероятности, где ID принадлежит незлонамеренному узлу, основан на расстоянии ID PNRP от целевого ID. Затем определяют вероятность как величину, обратно пропорциональную расстоянию ID между данным узлом и целевым узлом. В любом случае такая рандомизация уменьшает вероятность посылки запроса RESOLVE в злонамеренный узел.

Второй защитный механизм, который эффективен против атак на основе поисков, использует вектор битов, обсужденный выше, для поддержания информации о состоянии. То есть узел поддерживает информацию, идентифицирующую все сообщения RESOLVE, которые он обработал и для которых еще не получен ответ. Поля, которые используются для поддержания информации о состоянии, представляют собой целевой ID и список адресов в пакете RESOLVE. Второе поле используют для того, чтобы предотвратить модификацию списка адресов злонамеренным узлом при попытке вмешаться в процесс поиска. Как обсуждалось выше в связи с другими вариантами использования вектора битов, узел формирует значение хэш-функции этих полей из сообщения RESOLVE и устанавливает соответствующие битовые позиции в векторе битов для поддержания «истории» обработки этого сообщения RESOLVE.

Как показано на упрощенной блок-схеме по фиг.6, при приеме 296 сообщения от другого узла поля сообщения RESPONSE хэшируются 298 для вычисления битовой позиции. Затем узел вычисляет 300 вектор битов, чтобы узнать, установлена ли битовая позиция. Если бит не установлен, а значит, что это сообщение RESPONSE не относится к ранее обработанному сообщению RESOLVE, то тогда данный пакет отбрасывается 302. Если битовая позиция установлена, а значит, что данное сообщение RESPONSE относится к ранее обработанному сообщению RESOLVE, то битовая позиция сбрасывается 304. Сбрасывая битовую позицию, узел далее будет игнорировать идентичные сообщения RESPONSE, которые могли быть посланы как часть атаки считывания со стороны злонамеренного узла. Затем узел выполняет проверку, чтобы удостовериться в подлинности 306 всех сертификатов PAC в сообщении RESPONSE, прежде чем обработать сообщение RESPONSE и направить его на следующий переход. Если какие-то сертификаты PAC оказались недействительными 306, то узел отбросит 310 пакет.

Процесс RESOLVE подразумевает преобразование запроса RESOLVE в RESPONSE. Обработка пакета RESPONSE, как описано выше, включает в себя подтверждение соответствия RESPONSE перед этим принятому пакету RESOLVE и направление RESPONSE на следующий заданный переход (отрезок сети). Для примера предположим, что узел P принимает пакет RESPONSE S, содержащий целевой ID протокола PNRP, PAC наилучшего совпадения и путь с адресами всех узлов, которые обрабатывали исходный пакет RESOLVE перед данным узлом, заканчивая адресом PNRP этих узлов. Узел P подтверждает прием RESPONSE сообщением ACK. Узел P проверяет путь RESPONSE для своего собственного адреса. Этот адрес должен быть последней записью в адресном списке для того, чтобы этот пакет был действительным. Узел P также проверяет свой принятый вектор битов, чтобы убедиться в совпадении RESPONSE с полученным перед этим RESOLVE. Если RESPONSE не соответствует полю в принятом векторе битов или если адрес узла P не является последним адресом в списке пути, то RESPONSE отбрасывается и обработка прекращается. Узел P проверяет PAC наилучшего совпадения и добавляет его в свою локальную кэш-память. Если наилучшее совпадение не подтверждается, то RESPONSE отбрасывается и обработка прекращается. Узел P удаляет свой адрес из конца пути RESPONSE. Он продолжает удалять записи с конца пути RESPONSE, пока самая последняя запись не будет иметь установленный флаг, указывающий на узел, который принят как соответствующий запросу RESOLVE. Если путь не пустой, то соответствующий запрос RESOLVE порожден локально. PNRP выполняет проверку идентичности по наилучшему совпадению. Если проверка идентичности дает положительный результат, то сертификат с наилучшим совпадением поступает менеджеру запроса, в противном случае проходит индикация об отказе. Если путь оказывается пустым, то обработка завершается. Если путь не пустой, то узел направляет пакет RESPONSE в самую последнюю запись в списке пути.

Потребность в аннулировании сертификата адреса PNRP существует каждый раз, когда опубликованный сертификат адреса становится недействительным до истечения срока действия сертификата (достоверность/поле «До»). Примерами такого рода событий являются случаи, когда узел постепенно отсоединяется от сети P2P, либо когда узел

покидает группу и т.п. Механизм аннулирования согласно настоящему изобретению использует публикацию сертификата аннулирования. Сертификат аннулирования имеет установленный флаг аннулирования и дату «От» поля достоверности, установленную по текущему времени (или времени, когда сертификат должен быть аннулирован и поле «До», установленное равным тому же значению, что и ранее объявленные сертификаты. Все сертификаты, для которых удовлетворяются все следующие условия, считаются аннулированными: сертификат подписан той же запрашивающей стороной; ID совпадает с ID в сертификате аннулирования; поля адреса совпадают с полями в сертификате аннулирования; дата «До» поля достоверности такая же, как дата «До» поля достоверности в сертификате аннулирования; и дата «От» поля достоверности перекрывает дату «От» поля достоверности в сертификате аннулирования. Поскольку сертификат аннулирования подписан, он гарантирует, что злонамеренный узел не сможет отсоединить какой-либо узел от данного скопления узлов.

Приведенное выше описание различных вариантов изобретения было представлено в целях иллюстрации и описания. Оно не предполагает, что изобретение исчерпывается или ограничивается раскрытыми здесь конкретными вариантами. В свете вышеизложенных принципов возможны многочисленные модификации или варианты изобретения. Описанные варианты были выбраны и описаны для наилучшей иллюстрации принципов изобретения и его практического применения, чтобы тем самым дать возможность специалистам в данной области техники использовать изобретение в различных вариантах и с различными модификациями, подходящими для конкретного применения. Все указанные модификации и видоизменения лежат в рамках объема изобретения, определенного прилагаемой формулой изобретения, и должны интерпретироваться в соответствии с объемом прав, на который она объективно, законно и справедливо претендует.

Формула изобретения

1. Способ формирования самопроверяемого незащищенного сертификата равноправного адреса для того, чтобы не дать возможность злонамеренному узлу опубликовать защищенные данные об идентификации другого узла в незащищенном сертификате равноправного адреса в одноранговой сети, причем способ включает

формирование узлом одноранговой сети незащищенного сертификата равноправного адреса (PAC) для ресурса, обнаруживаемого в одноранговой сети, при этом ресурс имеет одноранговый идентификатор (ID); и включение унифицированного идентификатора ресурса (URI) в незащищенный PAC, из которого получен одноранговый ID.

2. Способ по п.1, в котором этап включения URI в незащищенный PAC, из которого получен одноранговый ID, содержит этап включения URI в формат «p2p2://URI».

3. Способ по п.1, в котором одноранговый ID является незащищенным.

4. Способ удобной проверки сертификата равноправного адреса в первом узле в одноранговой сети, причем упомянутый первый узел использует многоуровневую кэш-память для сохранения сертификатов равноправных адресов, причем способ содержит этапы:

прием сертификата равноправного адреса (PAC) от заданного второго узла;

определение того, на каком уровне многоуровневой кэш-памяти должен сохраняться PAC;

когда PAC должен сохраняться на одном из двух самых низких уровней кэш-памяти, то

(a) помещение PAC в отдельный список;

(b) формирование сообщения INQUIRE (запрос), содержащего идентификатор ID сертификата PAC, подлежащего проверке на действительность,

(c) передачу сообщения INQUIRE на второй узел; и

когда PAC должен сохраняться на верхнем уровне кэш-памяти, отличном от одного из двух самых низких уровней кэш-памяти, то сохранение PAC на этом верхнем уровне кэш-памяти с пометкой «не проверен на подлинность» с целью проверки на действительность

при первом использовании.

5. Способ по п.4, в котором передача сообщения INQUIRE включает в себя этап запроса цепочки сертификатов для PAC.

6. Способ по п.4, в котором формирование сообщения INQUIRE содержит этап формирования идентификатора ID транзакции, подлежащего включению в сообщение INQUIRE.

7. Способ по п.4, дополнительно содержащий этапы:

прием сообщения AUTHORITY (полномочия) от второго узла в ответ на сообщение INQUIRE;

удаление PAC из упомянутого отдельного списка; и

сохранение PAC на одном из двух самых низких уровней кэш-памяти.

8. Способ по п.5, дополнительно содержащий этапы:

прием сообщения AUTHORITY от второго узла в ответ на сообщение INQUIRE;

удаление PAC из отдельного списка;

анализ сообщения AUTHORITY с целью определения того, имеется ли последовательность сертификатов и является ли она действительной;

сохранение PAC на одном из двух нижних уровней кэш-памяти, когда последовательность сертификатов присутствует и является действительной;

и удаление PAC, когда последовательность сертификатов отсутствует и не является действительной.

9. Способ по п.6, дополнительно содержащий этапы:

прием сообщения AUTHORITY от второго узла в ответ на сообщение INQUIRE;

удаление PAC из отдельного списка;

анализ сообщения AUTHORITY с целью определения того, присутствует ли идентификатор ID транзакции;

сохранение PAC на одном из двух самых низких уровней кэш-памяти, когда ID транзакции присутствует; и

удаление PAC, когда ID транзакции отсутствует.

10. Способ по п.4, дополнительно содержащий этапы:

выбор PAC, сохраненного на верхнем уровне кэш-памяти, отличном от одного из двух самых низких уровней кэш-памяти, для маршрутизации пакета RESOLVE;

передачу пакета RESOLVE на второй узел, причем пакет RESOLVE имеет наложенную на него информацию о действительности принадлежности ID; и пометка PAC как действительного, когда второй узел подтверждает принадлежность ID.

11. Способ по п.10, дополнительно содержащий этапы:

удаление PAC из верхнего уровня кэш-памяти, отличного от одного из двух самых низких уровней кэш-памяти, когда второй узел не в состоянии подтвердить принадлежность ID; и повторную обработку пакета RESOLVE с другим PAC.

12. Способ по п.11, в котором этап удаления PAC из верхнего уровня кэш-памяти, отличного от одного из двух самых низких уровней кэш-памяти, когда второй узел не способен подтвердить принадлежность ID, содержит этап ожидания в течение заранее установленного периода времени подтверждения принадлежности ID вторым узлом перед удалением PAC.

13. Способ по п.11, в котором удаление PAC из верхнего уровня кэш-памяти, отличного от одного из двух самых низких уровней кэш-памяти, когда второй узел не способен подтвердить принадлежность ID, содержит этапы:

прием сообщения AUTHORITY от второго узла;

анализ сообщения AUTHORITY с целью определения того, смог ли второй узел подтвердить принадлежность ID;

определение того, что второй узел не смог подтвердить принадлежность ID;

и удаление PAC.

14. Способ по п.13, в котором этап определения того, что второй узел не смог подтвердить принадлежность ID, содержит этап определения того, что второй узел

установил флаг в сообщении AUTHORITY, указывающий на то, что он не смог подтвердить принадлежность ID сертификата PAC.

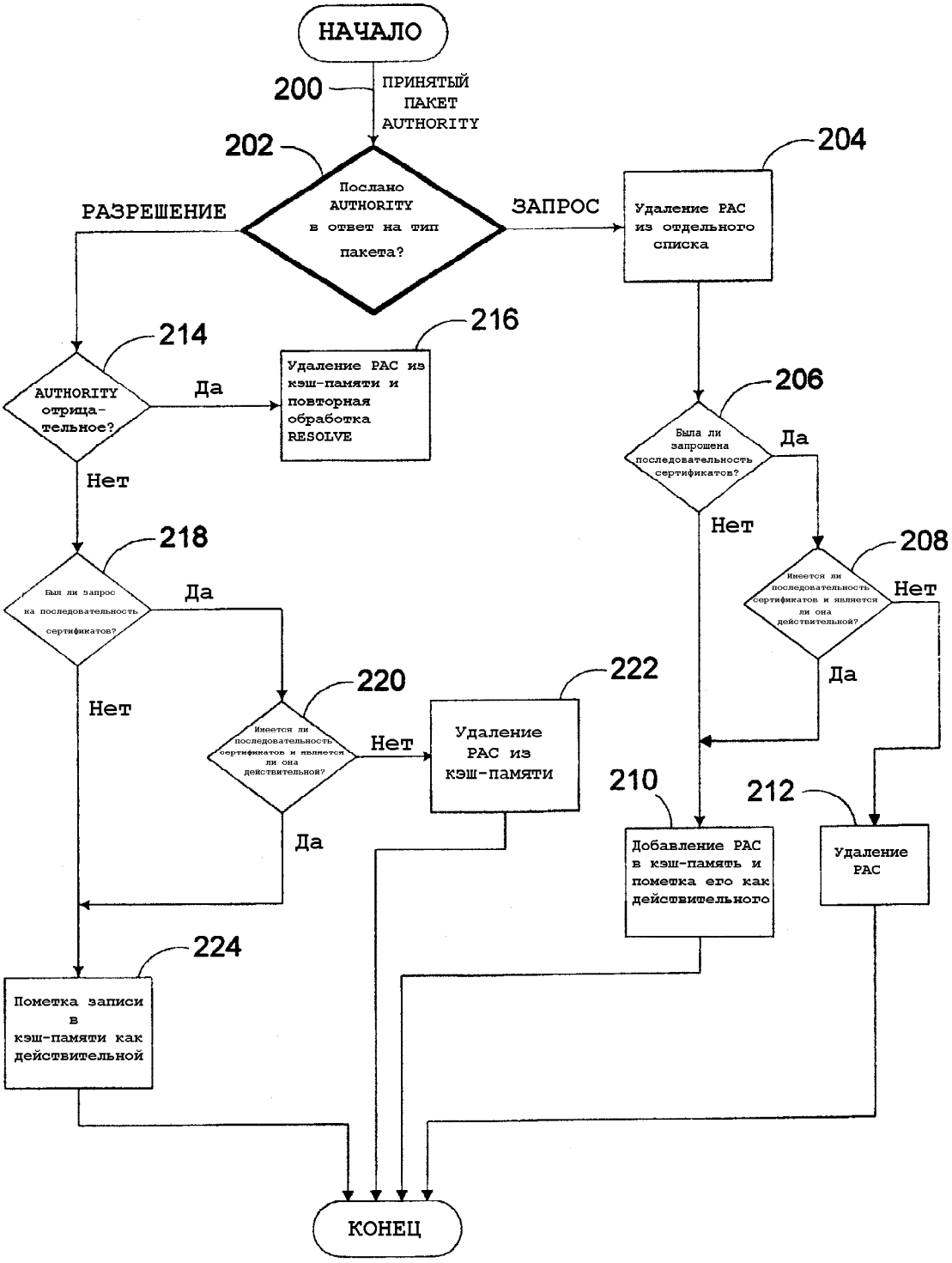
15. Способ по 13, в котором определение того, что второй узел не смог подтвердить принадлежность ID, содержит этап анализа сообщения AUTHORITY с целью определения того, что последовательность сертификатов отсутствует и не действительна.

16. Способ по п.10, в котором пометка PAC как действительного, когда второй узел подтверждает принадлежность ID, содержит этап приема сообщения AUTHORITY от второго узла, подтверждающего принадлежность ID.

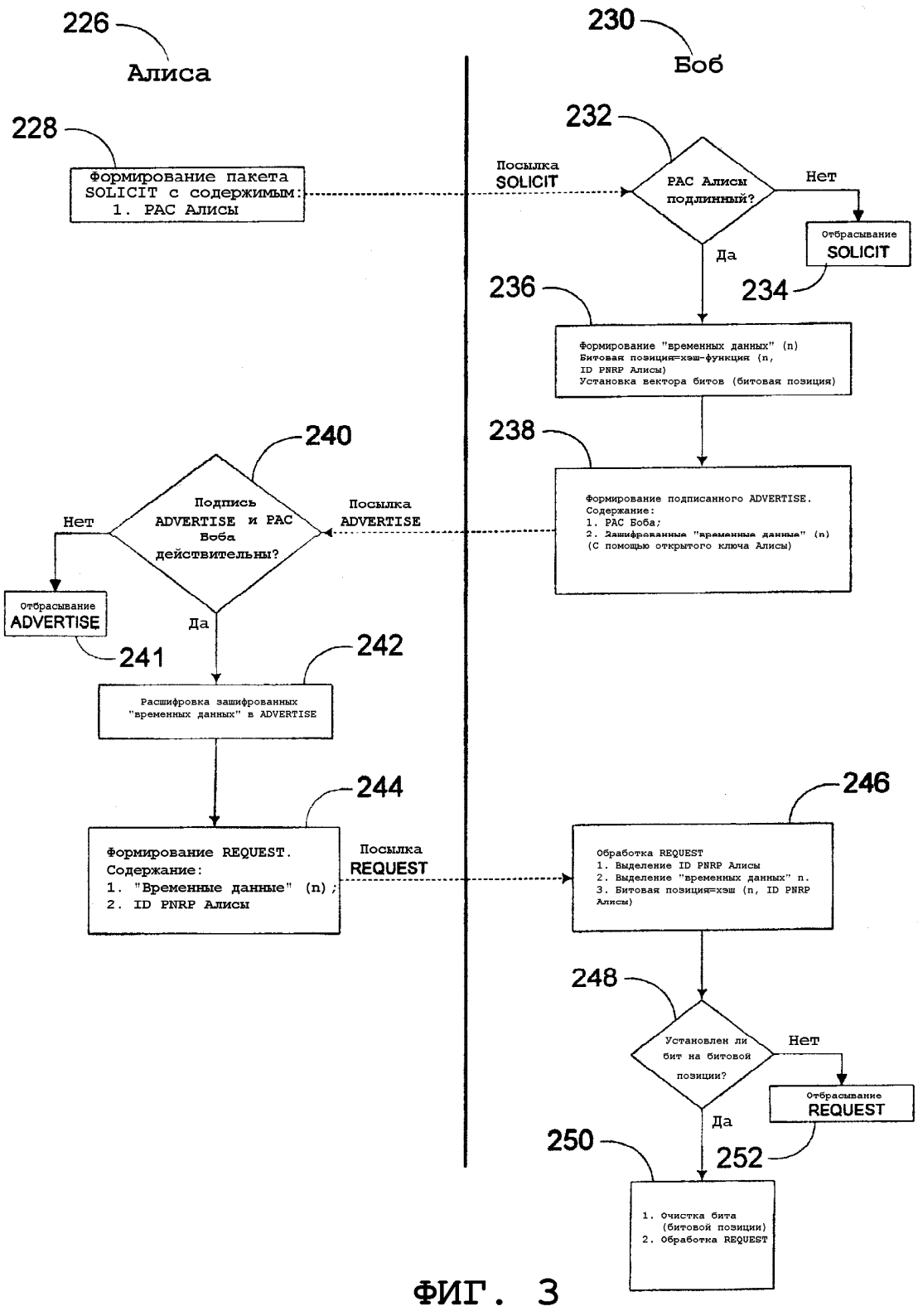
17. Способ по п.10, в котором этап пометки PAC как действительного, когда второй узел подтверждает принадлежность ID, содержит этап приема сообщения AUTHORITY от второго узла, имеющего последовательность сертификатов, подтверждающую принадлежность ID.

18. Машинно-считываемый носитель, содержащий машинно-считываемые команды для выполнения этапов по п.1.

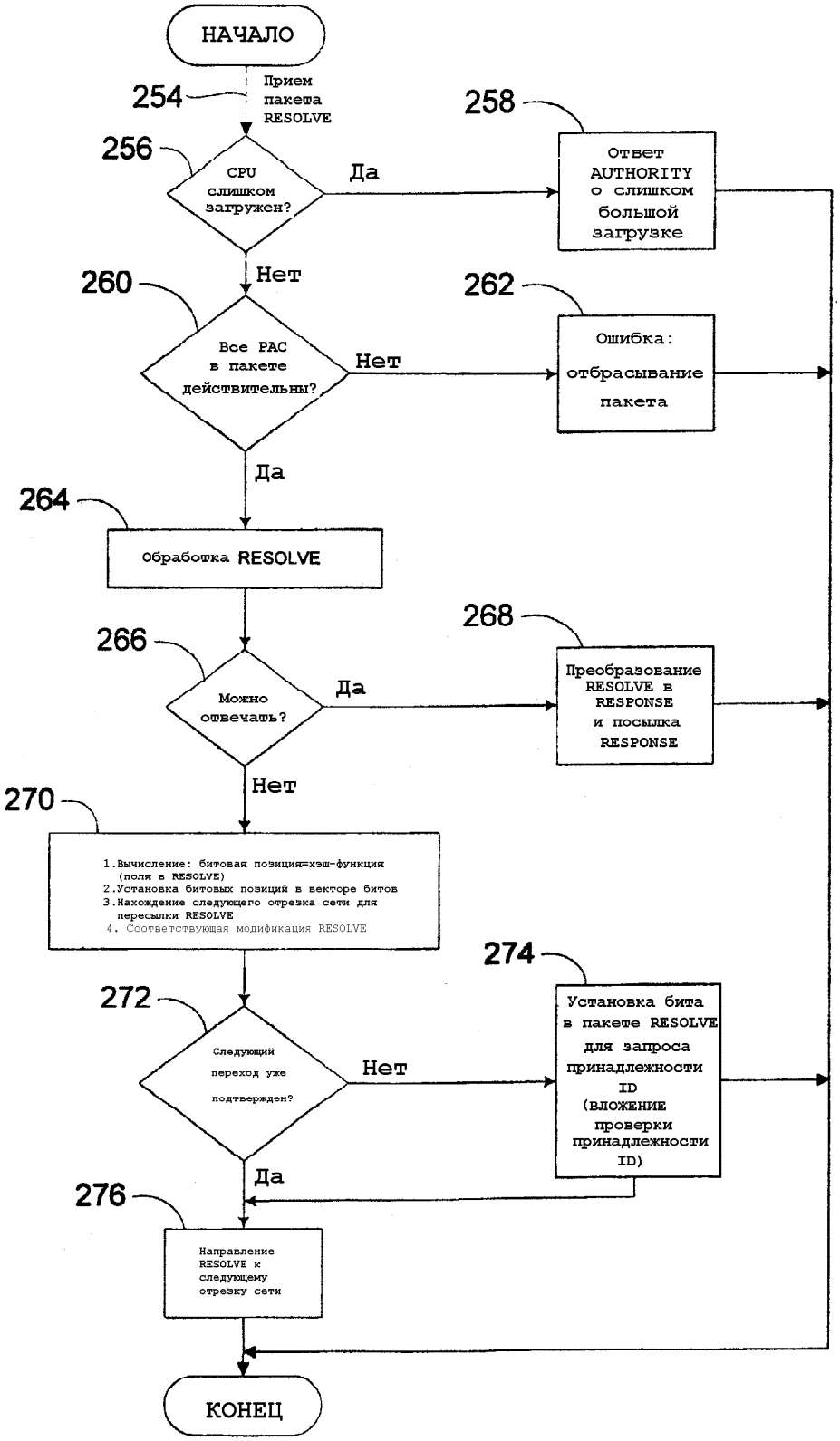
19. Машинно-считываемый носитель, содержащий машинно-считываемые команды для выполнения этапов по п.4.


НАЧАЛО
200
ПРИНЯТЫЙ ПАКЕТ AUTHORITY
202
Послано AUTHORITY в ответ на тип пакета?
РАЗРЕШЕНИЕ
ЗАПРОС
204
Удаление PAC из отдельного списка
214
AUTHORITY отрица-тельное?
Да
216
Удаление PAC из кэш-памяти и повторная обработка RESOLVE
Нет
206
Была ли запрошена последовательность сертификатов?
Да
Нет
208
Имеется ли последовательность сертификатов и является ли она действительной?
Нет
Да
218
Был ли запрос на последовательность сертификатов?
Да
Нет
220
Имеется ли последовательность сертификатов и является ли она действительной?
Нет
222
Удаление PAC из кэш-памяти
Да
210
Добавление PAC в кэш-память и пометка его как действительного
212
Удаление PAC
224
Пометка записи в кэш-памяти как действительной
КОНЕЦ

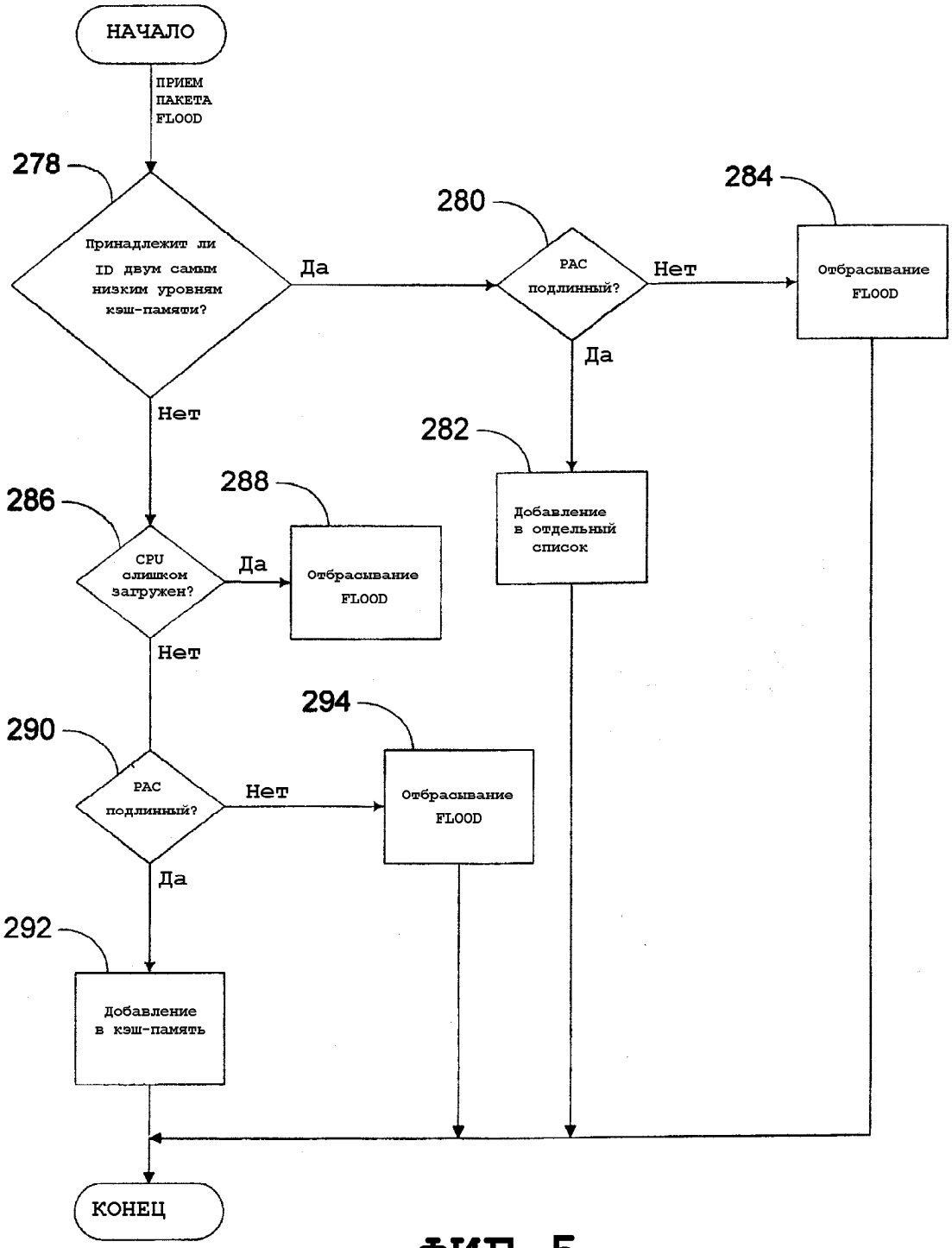
ФИГ. 2


226
Алиса
230
Боб
228
Формирование пакета SOLICIT с содержимым:
1. PAC Алисы
Посылка SOLICIT
232
PAC Алисы подлинный?
Нет
Да
Отбрасывание SOLICIT
234
236
Формирование "временных данных" (n)
Битовая позиция=хэш-функция (n, ID PNRP Алисы)
Установка вектора битов (битовая позиция)
238
Формирование подписанного ADVERTISE.
Содержание:
1. PAC Боба;
2. Зашифрованные "временные данные" (n)
(С помощью открытого ключа Алисы)
Посылка ADVERTISE
240
Подпись ADVERTISE и PAC Боба действительны?
Нет
Да
Отбрасывание ADVERTISE
241
242
Расшифровка зашифрованных "временных данных" в ADVERTISE
244
Формирование REQUEST.
Содержание:
1. "Временные данные" (n);
2. ID PNRP Алисы
Посылка REQUEST
246
Обработка REQUEST
1. Выделение ID PNRP Алисы
2. Выделение "временных данных" n.
3. Битовая позиция=хэш (n, ID PNRP Алисы)
248
Установлен ли бит на битовой позиции?
Нет
Да
Отбрасывание REQUEST
250
252
1. Очистка бита (битовой позиции)
2. Обработка REQUEST

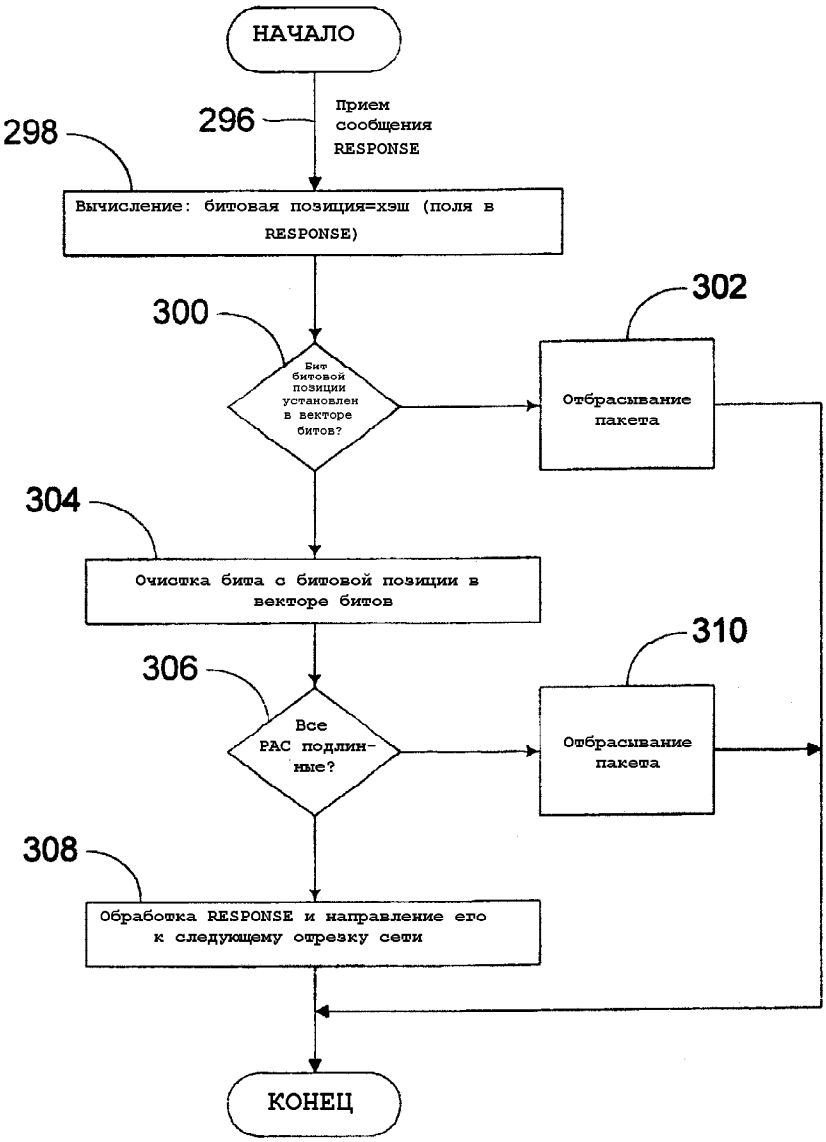
ФИГ. 3


НАЧАЛО
254
Прием пакета RESOLVE
256
CPU слишком загружен?
Да
258
Ответ AUTHORITY о слишком большой загрузке
Нет
260
Все PAC в пакете действительны?
Нет
262
Ошибка: отбрасывание пакета
Да
264
Обработка RESOLVE
266
Можно отвечать?
Да
268
Преобразование RESOLVE в RESPONSE и посылка RESPONSE
Нет
270
1.Вычисление: битовая позиция=хэш-функция (поля в RESOLVE)
2.Установка битовых позиций в векторе битов
3.Нахождение следующего отрезка сети для пересылки RESOLVE
4. Соответствующая модификация RESOLVE
272
Следующий переход уже подтвержден?
Нет
274
Установка бита в пакете RESOLVE для запроса принадлежности ID (ВЛОЖЕНИЕ проверки принадлежности ID)
Да
276
Направление RESOLVE к следующему отрезку сети
КОНЕЦ

ФИГ. 4


НАЧАЛО
ПРИЕМ ПАКЕТА FLOOD
278
Принадлежит ли ID двум самым низким уровням кэш-памяти?
Да
Нет
280
PAC подлинный?
Нет
Да
284
Отбрасывание FLOOD
282
Добавление в отдельный список
286
CPU слишком загружен?
Да
Нет
288
Отбрасывание FLOOD
290
PAC подлинный?
Нет
Да
294
Отбрасывание FLOOD
292
Добавление в кэш-память
КОНЕЦ

ФИГ. 5


НАЧАЛО
296
Прием сообщения RESPONSE
298
Вычисление: битовая позиция=хэш (поля в RESPONSE)
300
Бит битовой позиции установлен в векторе битов?
302
Отбрасывание пакета
304
Очистка бита с битовой позиции в векторе битов
306
Все РАС подлинные?
310
Отбрасывание пакета
308
Обработка RESPONSE и направление его к следующему отрезку сети
КОНЕЦ

ФИГ. 6