РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 782 153**(13) **C2**

(51) МПК
H04L 9/00 (2006.01)
G06F 21/00 (2013.01)
G06F 16/27 (2019.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
***H04L 9/00** (2022.08); **G06F 21/00** (2022.08); **G06F 16/27** (2022.08)*

(21)(22) Заявка: **2021103201, 09.02.2021**

(24) Дата начала отсчета срока действия патента:
09.02.2021

Дата регистрации:
21.10.2022

Приоритет(ы):
(22) Дата подачи заявки: **09.02.2021**

(43) Дата публикации заявки: **09.08.2022** Бюл. № **22**

(45) Опубликовано: **21.10.2022** Бюл. № **30**

Адрес для переписки:
302028, г. Орел, ул. Полесская, 55, кв. 5, Тарасенко С.С.

(72) Автор(ы):
Тарасенко Сергей Сергеевич (RU)

(73) Патентообладатель(и):
Тарасенко Сергей Сергеевич (RU)

(56) Список документов, цитированных в отчете о поиске: **WO 2019/195691 A1, 10.10.2019. US 9569771 B2, 14.02.2017. RU 2680350 C2, 19.02.2019. EA 36442 B1, 11.11.2020. WO 2020/190720 A1, 24.09.2020. US 2020/0374343 A1, 26.11.2020. WO 2020/097277 A1, 14.05.2020. US 2019/0132350 A1, 02.05.2019. WO 2020/072659 A1, 09.04.2020. US 10867057 B1, 15.12.2020.**

(54) СПОСОБ И СИСТЕМА ОРГАНИЗАЦИИ ЗАЩИЩЕННОГО ОБМЕНА ИНФОРМАЦИЕЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ БЛОКЧЕЙН И РАСПРЕДЕЛЁННЫХ СИСТЕМ ХРАНЕНИЯ ДАННЫХ

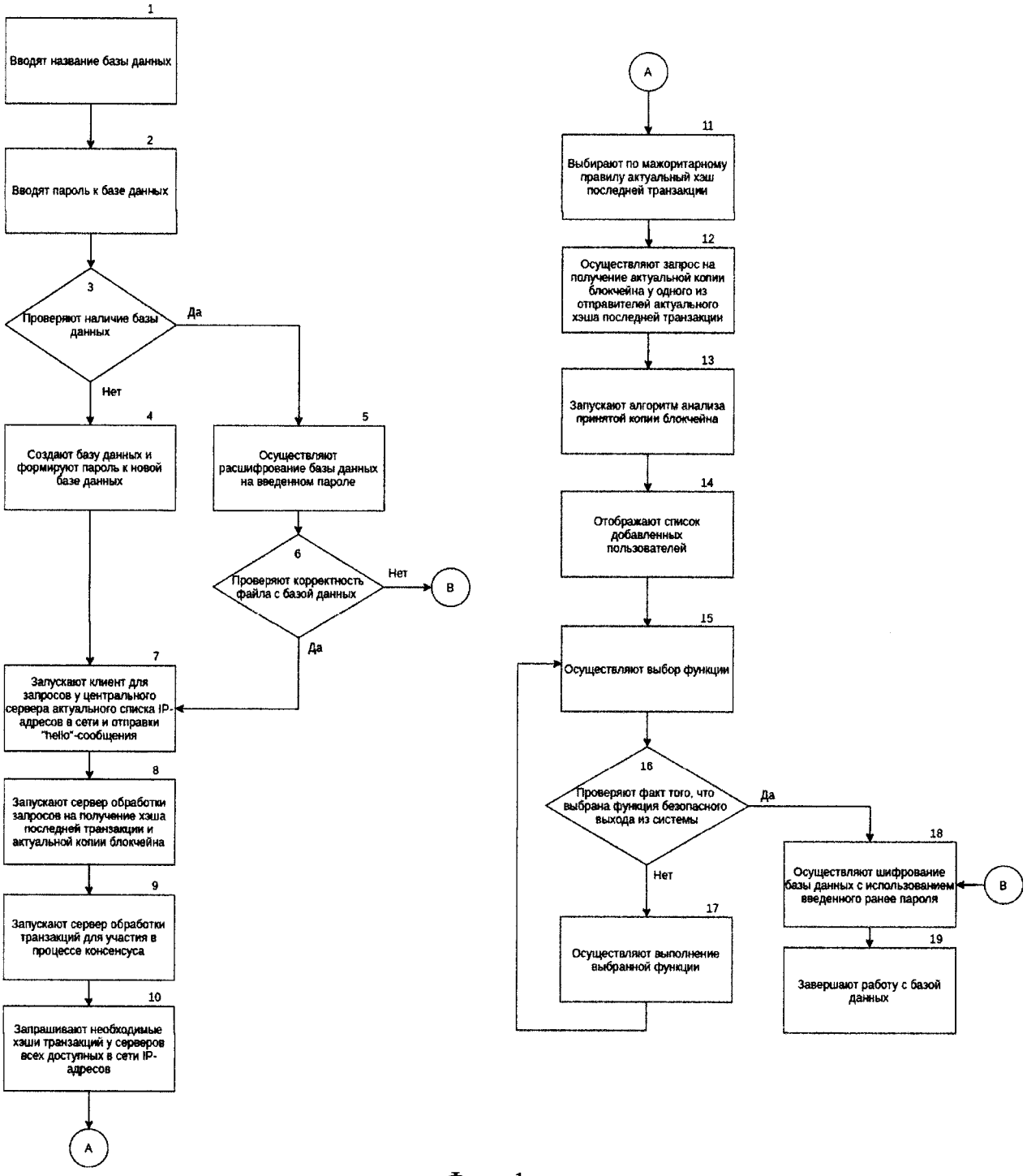
(57) Реферат:

Изобретение относится к способу и системе организации защищенного обмена информацией с использованием технологии блокчейн и распределённых систем хранения данных. Технический результат заключается в обеспечении защищенного обмена информацией. В способе вводят название базы данных, при этом дополнительно вводят пароль к этой базе данных, проверяют наличие базы данных на устройстве, при наличии базы данных осуществляют ее расшифрование с использованием симметричного шифрования AES-256, проверяют корректность файла с базой данных, при отсутствии базы данных создают новую базу данных, при этом формируют пароль к новой базе данных, необходимый для шифрования и расшифрования файла с новой базой данных по алгоритму AES-256, завершают работу с базой данных, при этом после проверки на корректность файла с базой данных либо после создания новой базы данных запускают сетевые сервисы, как минимум запускают клиент для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения», запускают сервер обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна, запускают сервер обработки транзакций для участия в процессе консенсуса, после этого запрашивают хэши транзакций у серверов всех доступных в сети IP-адресов, выбирают по мажоритарному правилу актуальный хэш последней транзакции, осуществляют запрос на получение актуальной копии блокчейна у одного из отправителей актуального хэша последней транзакции, запускают алгоритм анализа принятой актуальной копии блокчейна, отображают список добавленных пользователей, осуществляют выбор как минимум одной из функций: добавление нового пользователя, просмотр истории переписки с указанным

RU 2782153 C2

RU 2782153 C2

пользователем и отправка этому пользователю текстового сообщения, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового файла размером до 1 Кбайта, обновление ключа шифрования для указанного пользователя, отправка файла в распределенное хранилище, запрос файла из распределенного хранилища, обновление интерфейса отображения информации, безопасный выход из системы, которую необходимо выполнить системе, проверяют факт того, что выбрана функция безопасного выхода из системы, если данная функция не выбрана, то осуществляют выполнение выбранной функции, при этом после выполнения выбранной функции осуществляют выбор новой функции, если выбрана функция безопасного выхода из системы, то осуществляют шифрование базы данных с использование введенного ранее пароля по алгоритму AES-256. 2 н. и 8 з.п. ф-лы, 14 ил.



Фиг. 1

RU 2782153 C2

RU 2782153 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 782 153**(13) **C2**

(51) Int. Cl.
H04L 9/00 (2006.01)
G06F 21/00 (2013.01)
G06F 16/27 (2019.01)

(12) ABSTRACT OF INVENTION

(52) CPC
H04L 9/00 (2022.08); G06F 21/00 (2022.08); G06F 16/27 (2022.08)

(21)(22) Application: **2021103201, 09.02.2021**

(24) Effective date for property rights:
09.02.2021

Registration date:
21.10.2022

Priority:
(22) Date of filing: **09.02.2021**

(43) Application published: **09.08.2022** Bull. № **22**

(45) Date of publication: **21.10.2022** Bull. № **30**

Mail address:
302028, g. Orel, ul. Polesskaya, 55, kv. 5, Tarasenko S.S.

(72) Inventor(s):
Tarasenko Sergej Sergeevich (RU)

(73) Proprietor(s):
Tarasenko Sergej Sergeevich (RU)

RU 2782153 C2

(54) METHOD AND SYSTEM FOR ORGANIZATION OF PROTECTED INFORMATION EXCHANGE, USING BLOCKCHAIN TECHNOLOGY AND DISTRIBUTED DATA STORAGE SYSTEMS

(57) Abstract:

FIELD: information protection.

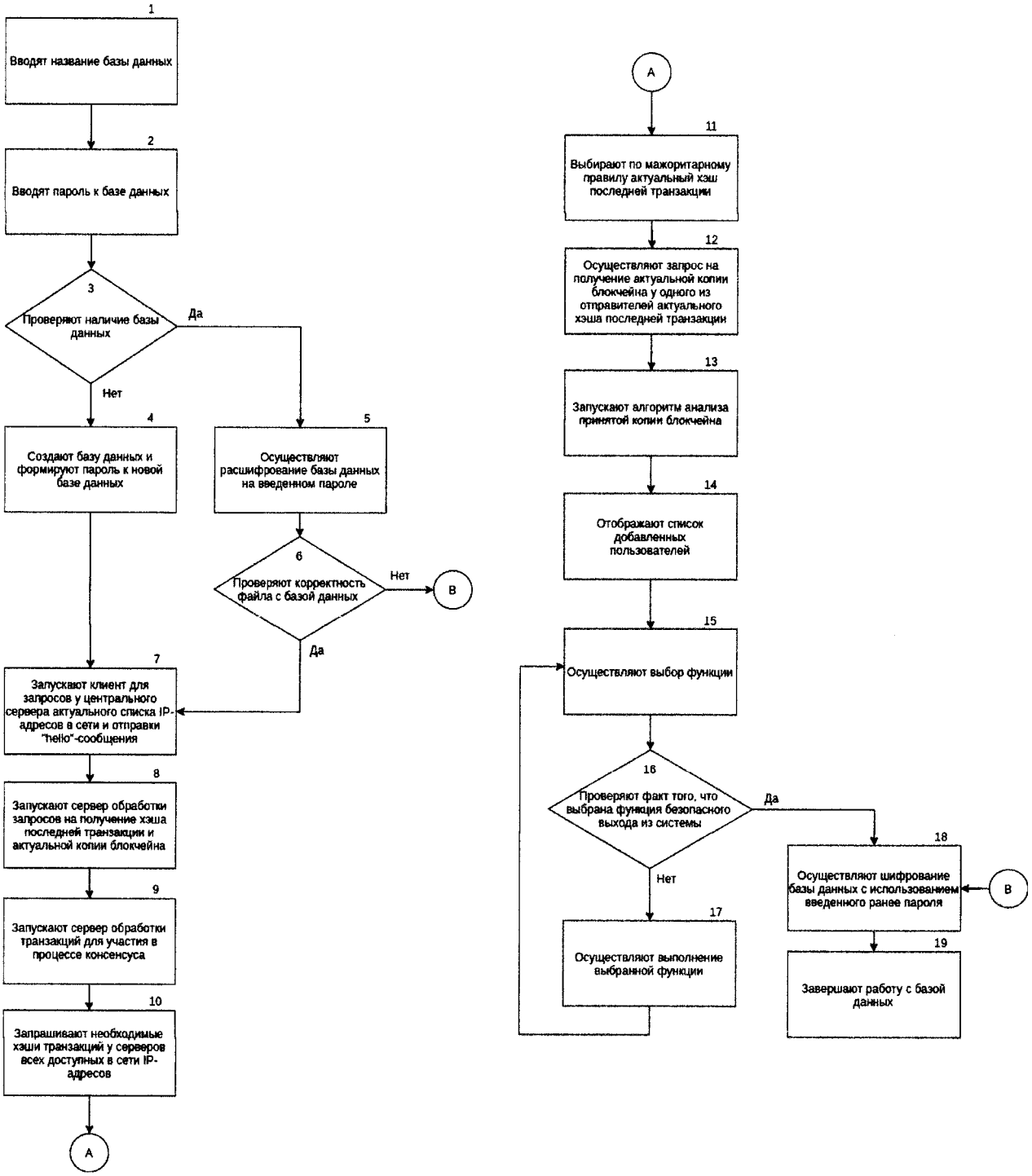
SUBSTANCE: invention relates to a method and a system for organization of protected information exchange, using a blockchain technology and distributed data storage systems. In the method, a name of a database is entered, while additionally entering a password to this database. The presence of the database on a device is checked. In case of the presence of the database, it is decrypted using symmetric AES-256 encryption, the correctness of a database file is checked. In case of the absence of the database, a new database is created, while forming a password to the new database, required to encrypt and decrypt a file with the new database, using the AES-256 algorithm. The work with the database is completed, while, after checking for the correctness of the database file or after creating a new database, network services are launched, at least a client for requests from a central server for an up-to-date list of IP addresses on a network and sending a “hello message” is launched. A request processing server is launched to receive a hash of the last transaction and an up-to-date copy of the blockchain. A transaction processing server is launched to participate in the consensus process. After that, transaction hashes are requested from servers of all IP addresses available on the network. The current hash of the last transaction is selected according to the majority rule. A request is made to receive an up-to-date copy of the blockchain from one of senders of the current hash of the last transaction. An algorithm for analyzing the received up-to-date copy of the blockchain is launched. A list of added users is displayed. At least one of functions is selected: addition of a new user, view of the history of correspondence with the specified user and sending to this user of a text message, view of the history of correspondence with the specified user and sending to this user of a text file up to 1 KB in size, update of an encryption key for the specified user, sending of a file to a distributed storage, request for a file from the distributed storage, update of an information display interface, safe logout of a system, which the system needs to perform. The fact that the

RU 2782153 C2

safe logout function is selected is checked. If this function is not selected, then, the selected function is performed, while, after performing the selected function, a new function is selected. If the safe logout function is selected, then, the database is encrypted using the previously entered password, using the AES-256 algorithm.

EFFECT: provision of protected information exchange.

10 cl, 14 dwg



Фиг. 1

RU 2782153 C2

RU 2782153 C2

Область техники

Изобретение относится к области защиты информации, в частности, к способам конфиденциальной передачи информации с использованием распределенных систем хранения данных и технологии блокчейн.

Уровень техники

Современные технологии, которыми обладают большие IT-компании, например, владельцы популярных бесплатных систем мгновенного обмена текстовыми сообщениями - мессенджеров и приложениями для обмена фотографиями и видеозаписями с элементами социальных сетей, позволяют отслеживать различную информацию о пользователях: круг общения, интересы, историю запросов, истории сообщений и т.д. В связи с этим можно утверждать, что имеется теоретическая возможность подмены переписки, а также отправки сообщений от имен пользователей. Данный тезис основан на том, что часть исходного кода вышеперечисленных приложений является закрытой, и заявления компаний об использовании «end-to-end», т.е. «сквозного», шифрования не исключает возможность сбора пользовательских данных, в том числе личных сообщений, аудио- и видеоразговоров и др. Также, данные компании имеют возможность блокировки пользователей. К тому же, нельзя исключать возможности взлома центрального сервера, и как следствие утечки данных, таких как, например, фотографий документов пользователей, номеров банковских карт, паролей т.д. При условии аутентификации пользователей через номер мобильного телефона говорить об анонимности также не приходится. Учитывая все перечисленное, можно с уверенностью сказать, что использование классических, в основе архитектуры которых лежат централизованные серверы обработки данных, мессенджеров и социальных сетей не обеспечивает анонимность пользователей и не дает гарантии конфиденциальности личных данных.

Повышение защищенности коммуникаций между пользователями сети Интернет и их анонимизация возможны путем создания мессенджера с открытым исходным кодом, использующего технологию блокчейн и распределенную систему хранения данных, а также теоретически недешифруемую (ТНДШ) систему для шифрования информации.

Для удобства описания способа и системы защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных введем ряд определений.

Блокчейн- выстроенная по определенным правилам непрерывная последовательная цепочка блоков, содержащих информацию (см. https://ru.wikipedia.org/wiki/Блокчейн).

Майнинг - деятельность по созданию новых структур, обычно речь идет о новых блоках в блокчейне, для обеспечения функционирования криптовалютных платформ (см. https://ru.wikipedia.org/wiki/Майнинг).

Майнер - объект, участвующий в майнинге (см. https://ru.wikipedia.org/wiki/Майнер).

Хеш-функция - функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определенным алгоритмом (см. https://ru.wikipedia/org/wiki/Хэш-функция).

Хэш (Хеш-сумма) - значение хеш-функции (см. https://ru.wikipedia.org/wiki/Хэш).

Транзакция - группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными (см. https://ru.wikipedia/org/wiki/Транзакция_(информатика)).

Мессенджер - службы мгновенных сообщений, программы онлайн-консультанты и программы-клиенты для обмена сообщениями в реальном времени через Интернет (см. https://ru.wikipedia/org/wiki/Система_мгновенного_обмена_сообщениям).

Proxy-сервер - промежуточный сервер или комплекс программ в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером, при этом о посредничестве могут как знать, так и не знать обе стороны, позволяющий клиентам как выполнять косвенные запросы, принимая и передавая их через прокси-сервер, к другим сетевым службам, так и получать ответы (см. https://ru.wikipedia.org/wiki/Проксисервер).

Виртуальная частная сеть (Virtual Private Network, VPN) - технология, которая расширяет частную сеть через общедоступную сеть Интернет и позволяет пользователям отправлять и получать данные через общие сети или сети общего пользования, как если бы их вычислительные устройства были подключены к частной сети (см. https://en.wikipedia.org/wiki/Virtual_private_network).

Метаинформация (метаданные) - информация о другой информации, или данные, относящиеся к дополнительной информации о содержимом или объекте (см. https://ru.wikipedia.org/wiki/Метаданные).

Консенсус - принятие решения на основе общего согласия без проведения голосования, если против него никто не выступает, либо при исключении мнения немногих несогласных участников (см. https://ru.wikipedia.org/wiki/Консенсус).

Под мажоритарным правилом выбора какого-либо решения путем голосования будем понимать правило, согласно которому избранным считается решение, получившее большинство голосов.

Под файлом-ссылкой будем понимать файл, генерируемый предлагаемой системой, содержащий в себе часть зашифрованного исходного целевого файла, ссылки на серверы, хранящие остальные части зашифрованного исходного целевого файла, а также необходимую метаинформацию для осуществления корректной сборки файла при запросе этого файла из распределенного хранилища.

Под глобальным блокчейном будем понимать версию блокчейна, которая находится у большинства участников блокчейн-коммуникации в текущий момент времени.

Известен способ и система «Method and system for storage and retrieval of blockchain blocks using galois fields» (патент US 9569771 B2 от 03.11.2016 г.), включающие в себя один или несколько блоков блокчейна, которые хранятся и извлекаются с помощью модифицированных полей Галуа в облачной или одноранговой, (например, P2P), сети связи. Модифицированное поле Галуа обеспечивает, как минимум, дополнительные уровни безопасности и конфиденциальности для блокчейнов.

Известен способ «Способ и система распределенного хранения восстанавливаемых данных с обеспечением целостности и конфиденциальности информации» (патент RU 2680350 C2 от 19.02.2019 г.), который заключается в том, что осуществляется замена отказавшего узла, хранящего данные, относящиеся к части файла данных, при этом каждым из множества доступных узлов хранения данных принимается указание от блока управления на замену отказавшего узла новым узлом хранения данных, каждый из доступных узлов хранения данных содержит множество контрольных сумм, сформированных из файла данных, которые могут быть сформированы на основании частей файла данных, используя методики кодирования со стиранием, при этом заменяющая контрольная сумма формируется на каждом из множества доступных узлов хранения данных посредством создания линейной комбинации контрольных сумм на каждом узле хранения данных, используя случайные коэффициенты, в дальнейшем эти заменяющие контрольные суммы используются для восстановления утерянного файла данных, отличающийся тем, что каждый из доступных блоков обработки данных с соответствующими узлами хранения данных содержит множество

данных, сформированных из файлов, соответствующих блокам обработки данных с узлами хранения данных, при этом множество данных предварительно подвергается процедуре блочного шифрования с нелинейными биективными преобразованиями, а сформированное множество блоков криптограмм блоков обработки данных с узлами хранения данных распределяется между доступными узлами хранения данных, в которых, используя методы многозначного помехоустойчивого кодирования, формируется соответствующее множество избыточных данных, далее поступившие от других блоков обработки данных с узлами хранения данных блоки криптограмм удаляются с целью сокращения общей избыточности, при этом сформированное множество избыточных данных с блоками криптограмм блока обработки данных с узлом хранения данных, осуществлявшего их формирование, используется для восстановления утерянных файлов данных, при этом блок восстановления данных получает информацию от блока управления в отношении того, какие блоки обработки данных с соответствующими узлами хранения данных в настоящий момент доступны и, соответственно, имеют множество информационных и избыточных данных файла, затем блок восстановления данных получает множество информационных и избыточных данных от указанных блоков обработки данных с узлами хранения данных, блок восстановления данных выполняет полное восстановление утраченных файлов данных, данные, восстановленные блоком восстановления данных, совместно с данными доступных блоков обработки данных с соответствующими узлами хранения данных передаются на вновь введенный блоком управления блок обработки данных с узлом хранения данных для формирования блоков избыточных данных.

Наиболее близкими по технической сущности к заявленным способу и системе является «Discrete blockchain and blockchain communications» (патент WO 2019/195691 A1 от 10.10.2019 г.), заключающийся в том, что шифруют и расшифровывают сигналы между устройствами, чтобы гарантировать, что сообщения, использующие технологии блокчейн, могут быть обнаружены только назначенными третьими сторонами или не обнаруживаются вообще. Для обеспечения работоспособности сети необходимы майнеры - участники сети с высокими уровнями вычислительной мощности, которые конкурируют за подтверждение транзакций путем вычисления хэшей транзакции. В данном способе могут использоваться различные виды практически недешифруемых систем симметричного шифрования и асимметричного шифрования (ПНДШ), и «Data security» (патент US 2005/0081048 A1 от 14.04.2005 г.), заключающийся в том, что шифруют по меньшей мере на одном ключе, по меньшей мере одну или несколько частей входных данных для генерации по меньшей мере одной или нескольких соответствующих частей выходных данных, которые будут храниться в одном или нескольких местах хранения, а также могут создавать по меньшей мере на одной или нескольких соответствующих частях выходных данных проверочные данные, которые должны быть сохранены в хранилище или нескольких хранилищах, находящихся в запоминающем устройстве.

Технической проблемой данных аналогов и прототипов является угроза компрометации адресной информации и пользовательских данных, хранящихся в блокчейне, а также существует угроза идентификации отправителей и получателей файлов в распределенных системах хранения данных.

Причины, по которым это может происходить следующие:

- в рассмотренных аналогах предполагается, что информация в блокчейне является либо открытой, либо зашифрованной с помощью асимметричного шифрования или симметричного шифрования с использованием ПНДШ систем, что в конечном итоге

не гарантирует абсолютную стойкость информации к дешифрованию;

- возможность сбора статистической информации о передаваемом трафике в каналах связи, что может привести к компрометации отправителей и получателей файлов в распределенной системе хранения данных, а также к возможности сбора всех частей зашифрованного файла и, следовательно, к его дешифрованию.

Техническим результатом является то, что настоящее изобретение направлено на предотвращения угрозы компрометации адресной информации и пользовательских данных, хранящихся в блокчейне, за счет использования распределенной системы хранения данных, в которой часть данных из файлов, предназначенных для отправки в удаленные хранилища данных, хранятся на локальном запоминающем устройстве. При этом генерируются ложные части этих файлов, что позволяет обмениваться транзакциями в блокчейн-коммуникации зашифрованными с использованием ТНДШ системы, исключая тем самым возможность однозначного дешифрования транзакций. Все файлы, отправляемые в распределенное хранилище, в том числе и ключи для ТНДШ систем, предварительно зашифровываются на алгоритме aes-256 в режиме гаммирования с обратной связью CFB (cipher feed back mode), в котором каждая часть зашифрованных данных зависит от другой, что исключает возможность расшифрования файла при отсутствии какой-либо части зашифрованного файла.

Другим техническим результатом является решение технической проблемы компрометации информации об отправителях и получателях файлов в распределенной системе хранения данных за счет того, что при отправке файла в распределенное хранилище данных устройство-участник блокчейн-коммуникации зашифровывает файл с помощью симметричного шифрования с использованием ПНДШ системы с применением алгоритма aes-256 в режиме CFB. При этом распределенное хранилище данных может состоять из любых сторонних хранилищ данных. Отправляемый файл разбивается на несколько частей, вводятся ложные части, перемешиваются файлы перед отправкой, назначают каждому файлу, который содержит часть зашифрованного файла или ложную часть файла, серверы, на которых будут храниться эти файлы. Вышеперечисленные меры, с одной стороны, значительно усложняют задачу сбора статистики об отправляемых файлах, а с другой стороны, гарантируют невозможность дешифрования файлов, хранящихся в распределенной системе хранения данных. С целью невозможности определения отправителя и получателя частей зашифрованного файла, все соединения осуществляют либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Раскрытие изобретения

В заявленном способе эта техническая проблема решается тем, что в способе организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных, заключающемся в том, что вводят название базы данных, вводят пароль к этой базе данных, проверяют наличие базы данных на устройстве, при наличии базы данных осуществляют ее расшифрование с использованием симметричного шифрования, проверяют корректность файла базы данных, при отсутствии базы данных создают новую базу данных, при этом формируют пароль к новой базе данных, необходимый для шифрования и расшифрования файла новой базы данных по алгоритму aes-256, завершают работу с базой данных. Дополнительно после проверки на корректность файла с базой данных, либо после создания новой базы данных, запускают сетевые сервисы, при этом как минимум, запускают клиент для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello»-сообщения, запускают сервер обработки запросов на получение

хэша последней транзакций и актуальной копии блокчейна, запускают сервер обработки транзакций, а именно, для участия в процессе консенсуса. После запуска сетевых сервисов запрашивают хэши транзакций у серверов всех доступных в сети IP-адресов, выбирают по мажоритарному правилу актуальный хэш последней транзакции, осуществляют запрос на получение актуальной копии блокчейна у одного из отправителей актуального хэша последней транзакции. Запускают алгоритм анализа принятой актуальной копии блокчейна и отображают список добавленных пользователей. Осуществляют выбор одной из функций: добавление нового пользователя, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового сообщения, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового файла размером до 1 Кбайта, обновление ключа шифрования для указанного пользователя, отправка файла в распределенное хранилище, запрос файла из распределенного хранилища, обновление интерфейса отображения информации, безопасный выход из системы, которую необходимо выполнить системе. Проверяют факт того, что выбрана функция безопасного выхода из системы, если данная функция не выбрана, то осуществляют выполнение выбранной функции, при этом после выполнения выбранной функции осуществляют выбор новой функции, если выбрана функция безопасного выхода из системы, то осуществляют шифрование базы данных с использование введенного ранее пароля по алгоритму aes-256.

Согласно одному из частных вариантов реализации при запуске сервера обработки запросов на получение хэша последней транзакций и актуальной копии блокчейна принимают сервером обработки запросов сообщения клиента с запросом. Проверяют вид запроса сервером, если поступает запрос на получение хэша последней транзакции копии блокчейна сервера, то извлекают из базы данных хэш последней транзакции, если поступает запрос на получение актуальной копии блокчейна сервера, то извлекают из базы данных актуальную копию блокчейна сервера. Формируют правильное сообщение для отправки клиенту с учетом характеристики информации, извлеченной из базы данных и отправляют сформированный ответ клиенту.

Согласно одному из частных вариантов реализации при запуске сервера обработки транзакций принимают сервером обработки транзакций транзакции от клиента. Извлекают из транзакции предлагаемый клиентом ID и хэш последней транзакции из локальной копии блокчейна клиента. Проверяют последний ID локальной копии блокчейна сервера обработки транзакций, при этом ID должен быть на единицу меньше, предлагаемого клиентом. Осуществляют проверку правильности ID, если проверка дала отрицательный результат, то отправляют клиенту сообщение об отказе данным сервером обработки транзакций подтверждать принятую от клиента транзакцию. Если проверка прошла успешно, то проверяют хэш последней транзакции локальной копии блокчейна клиента, содержащийся в присланной клиентом транзакции, и хэш последней транзакции локальной копии блокчейна сервера обработки транзакций. При этом хэш должен совпадать, если хэш не совпадает, то отправляют клиенту сообщение об отказе данным сервером обработки транзакций подтверждать принятую от клиента транзакцию. Если хэш совпадает, то отправляют клиенту ответ с положительным решением, о подтверждении данным сервером обработки транзакций принятую от клиента транзакцию. Добавляют принятую от клиента транзакцию сервером обработки транзакций в свою локальную копию блокчейна. Запускают алгоритм анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на этом сервере обработки транзакций.

Согласно одному из частных вариантов реализации при запуске алгоритма анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на сервере обработки транзакций извлекают из всей транзакции зашифрованную часть транзакции путем выделения, предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна. Расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного сервером обработки транзакций пользователя на зашифрованную часть транзакции с учетом смещения ключа, при этом зашифрованная часть транзакции расшифруется только в случае, если в поле отправителя, которое также зашифровано, будет имя пользователя на чьем ключе производится расшифровка, а в поле получателя будет имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции. Проверяют получилось ли расшифровать сообщение на одном из ключей добавленных пользователей. В случае, если получилось, то расшифрованную часть транзакции разбивают на логические элементы как минимум, время отправки, имя отправителя, имя получателя, передаваемое сообщение, которое может являться либо текстовым файлом размером до 1 Кбайта, либо текстовым сообщением, и заносят в локальную базу данных сервера обработки транзакций в открытом виде. В случае, если не получилось расшифровать сообщение на одном из ключей добавленных пользователей, то транзакция не предназначена для пользователя данного сервера обработки транзакций.

Согласно одному из частных вариантов реализации при выполнении запроса на получение актуальной копии блокчейна удаляют из базы данных предыдущую локальную копию блокчейна и записывают в базу данных, принятую от сервера обработки запросов на получение хэша последней транзакций и актуальной копии блокчейна копию локального блокчейна. Проверяют количество транзакций предыдущей локальной копии блокчейна с принятой копией. Если у предыдущей локальной копии блокчейна число транзакций больше, чем у принятой копии, то значит произошло обновление глобального блокчейна, при этом обнуляют смещения ключей для всех добавленных пользователей. В случае, когда обновление глобального блокчейна не произошло, тогда оставляют значения смещения ключей для всех добавленных пользователей без изменений. Анализируют каждую транзакцию на наличие входящих сообщений, при этом извлекают из всей транзакции зашифрованную часть транзакции путем выделения предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна. Расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного сервером обработки транзакций пользователя на зашифрованную часть транзакции с учетом смещения ключа, при этом зашифрованная часть транзакции расшифруется только в случае, если в поле отправителя, которое также зашифровано, будет имя пользователя, на чьем ключе производится расшифровка, а в поле получателя будет имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции.

Согласно одному из частных вариантов реализации, в котором выбирают одну из функций: добавление нового пользователя, при этом вводят имя нового пользователя и указывают путь к файлу с ключом для этого пользователя, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового сообщения, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового файла размером до 1 Кбайта, обновление ключа шифрования для указанного пользователя, отправка файла в распределенное хранилище, запрос файла из распределенного хранилища, обновление интерфейса отображения

пользовательской информации, безопасный выход из системы, которую необходимо выполнить системе.

Согласно одному из частных вариантов реализации при запросе отправки файла в распределенное хранилище вводят путь к исходному файлу. Зашифровывают исходный файл алгоритмом шифрования «Advanced Encryption Standard» с длинной ключа 256 бит, т.е. aes-256, в режиме гаммирования с обратной связью, при этом ключ шифрования помещают в файл-ссылку. Вычисляют хэш зашифрованного файла, при этом помещают хэш в файл-ссылку. Разбивают зашифрованный файл на N частей, являющихся отдельными файлами, случайного размера, при этом всю метаинформацию и одну из N частей случайного размера, которая не отправляется в сеть, помещают в файл-ссылку. Добавляют K ложных частей, при этом всю метаинформацию помещают в файл-ссылку. Назначают каждому из (N-1+K) файлов R серверов в распределенной системе хранения данных и вычисляют хэш каждого из этих файлов, при этом всю метаинформацию помещают в файл-ссылку. Перемешивают (N-1+K) файлов, при этом всю метаинформацию помещают в файл-ссылку. Отправляют каждый из (N-1+K) файлов на R серверов в распределенной системе хранения данных с использованием Proxy-серверов, VPN или системы подмены IP-адресов. Формируют окончательный вариант файл-ссылки.

Согласно одному из частных вариантов реализации при запросе файла из распределенного хранилища вводят путь к файлу-ссылке на исходный файл, извлекают из файла-ссылки метаинформацию об исходном файле: количество частей, на которые был разбит исходный файл, количество ложных частей, имена файлов на удаленных серверах, правила перемешивания файлов перед отправкой, IP-адреса серверов хранения файлов, ключ шифрования, на котором зашифрован весь файл, хэши частей файла, хэш всего файла; запрашивают с серверов части файлов с использованием Proxy-серверов, VPN или системы подмены IP-адресов, проверяют хэши частей файла таким образом, что если с серверов в распределенной системе хранения данных пришла как минимум одна правильная копия каждой части файла и ее хэш правильный, то переходят к следующему шагу, если хоть одно условие не выполняется, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, перемешивают принятые части файла, отделяют ложные части файла, выполняют сборку зашифрованного исходного файла, проверяют хэш всего зашифрованного файла, если проверка отрицательная, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, если проверка прошла успешно, то выполняют расшифрование зашифрованного файла.

Согласно одному из частных вариантов реализации при просмотре истории переписки с указанным пользователем и отправке ему текстового сообщения отображают предыдущую переписку с данным пользователем из базы данных. Вводят новое сообщение для пользователя и добавляют к сообщению как минимум поле времени отправки сообщения, поле имени отправителя, поле имени получателя. Выполняют шифрование этих полей на ключе данного пользователя. Извлекают из базы данных ID последней транзакции и увеличивают на единицу данное значение. Формируют новый ID для транзакции. Извлекают из базы данных хэш последней транзакции локальной копии блокчейна отправителя. При этом хэш любой транзакции, за исключением первой, формируется путем вычисления хэша от двух значений: хэша транзакции, записанной в локальной копии блокчейна отправителя до текущей транзакции и текущей транзакции в локальной копии блокчейна отправителя. Создают транзакцию для передачи, в которую входит новый ID, хэш последней транзакции локальной копии блокчейна отправителя и зашифрованные данные, зашифрованные

на шифре Вернама с использованием ключа пользователя. Следовательно, при условии, что пользовательский ключ представлен в виде одноразового блокнота, состоящего из случайного набора бит и на каждое сообщение, отводится своя последовательность бит ключа, которая в дальнейшем не может быть использована повторно в качестве ключа для другого сообщения, то можно утверждать, что выполняется условие для ТНДШ системы шифрования. Отправляют всем доступным IP-адресам из списка актуальных IP-адресов, присланным центральным сервером, находящимся в рассматриваемой нами системе, сформированную транзакцию. Принимают от всех доступных IP-адресов из списка актуальных IP-адресов, присланным центральным сервером, находящихся в рассматриваемой нами системе, ответ о готовности подтверждения серверами обработки транзакций сформированной транзакции. Проверяют по мажоритарному правилу факт подтверждения транзакции, если проверка не прошла успешно, то выполняют запрос актуальной копии блокчейна, если проверка прошла успешно, то добавляют транзакцию в локальную копию блокчейна отправителя. Добавляют в незашифрованном виде поле - время отправки транзакции, поле - имя отправителя, поле - имя получателя и передаваемое сообщение в базу данных. Выполняют смещение ключа для данного пользователя на длину зашифрованных данных.

Новая совокупность существенных признаков позволяет достичь технического результата невозможности компрометации отправителей и получателей файлов в распределенной системе хранения данных за счет того, что при запросе отправки файла в распределенное хранилище вводят путь к исходному файлу, зашифровывают исходный файл с применением алгоритма шифрования «Advanced Encryption Standard» с длинной ключа 256 бит в режиме гаммирования с обратной связью CFB, при этом ключ шифрования помещают в файл-ссылку, вычисляют хэш зашифрованного файла, помещают хэш в файл-ссылку, разбивают зашифрованный файл на N частей (файлов) случайного размера, при этом всю метаинформацию и одну из N частей случайного размера, которая не отправляется в сеть, помещают в файл-ссылку, добавляют K ложных частей, при этом всю метаинформацию помещают в файл-ссылку, назначают каждому из (N-1+K) файлов R серверов в распределенной системе хранения данных и вычисляют хэш каждого из этих файлов, при этом всю метаинформацию помещают в файл-ссылку, перемешивают (N-1+K) файлов, при этом всю метаинформацию помещают в файл-ссылку, отправляют каждый из (N-1+K) файлов на R серверов в распределенной системе хранения данных с использованием механизма подмены IP-адресов, либо через Proxy-серверы, либо через VPN, формируют окончательный вариант файл-ссылки; при запросе файла из распределенного хранилища вводят путь к файлу-ссылке на исходный файл, извлекают из файла-ссылки метаинформацию об исходном файле: количество частей, на которые был разбит исходный файл, количество ложных частей, имена файлов на удаленных серверах, правила перемешивания файлов перед отправкой, IP-адреса серверов хранения файлов, ключ шифрования, на котором зашифрован весь файл, хэши частей файла, хэш всего файла, запрашивают с серверов части файлов с использованием механизма подмены IP-адресов, либо через Proxy-серверы, либо через VPN, проверяют хэши частей файла с учетом того, что если с серверов в распределенной системе хранения данных получена, как минимум, одна копия каждой части файла и ее хэш прошел проверку на правильность, то переходят к следующему шагу, если хоть одно условие не выполняется, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, перемешивают принятые части файла, отделяют ложные части файла, выполняют сборку зашифрованного исходного файла, проверяют хэш всего

зашифрованного файла, если проверка отрицательная, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, если проверка прошла успешно, то выполняют расшифрование зашифрованного (принятого) файла.

В заявленной системе эта задача решается тем, что в системе организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных, состоящей как минимум, из одного центрального сервера, включающего модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль обработки «hello-сообщений» для приема от устройств-участников блокчейн-коммуникации «hello-сообщений» и обновление актуального списка доступных в текущий момент времени IP-адресов в сети, модуль обработки запросов на получение актуальной копии списка в текущий момент времени IP-адресов в сети для отправки актуального списка доступных в текущий момент времени IP-адресов в сети по запросу, модуль памяти для хранения актуального списка доступных на данный момент времени IP-адресов в сети, модуль питания устройства для обеспечения электроэнергией всех модулей устройства и, как минимум, трех устройств - участников блокчейн-коммуникации, включающие модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль отправки запросов на серверы для отправки запросов на серверы обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна и на серверы обработки транзакций, модуль клиента для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения» для идентификация устройства в сети как доступного посредством отправки на центральный сервер «hello-сообщения» и получение актуального списка доступных на данный момент времени IP-адресов в сети, модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна для отправки хэша последней транзакции по запросу и отправки копии локального блокчейна по запросу, модуль сервера обработки транзакций для участия в процессе консенсуса для обработки транзакций и отправки ответа на запрос о ее подтверждении или отклонении, модуль базы данных для хранения и предоставления доступа модулям ко всей необходимой информации, модуль памяти для хранения файлов и файлов-ссылок, а так же базы данных, модуль ввода/вывода информации для осуществления ввода/вывода информации в устройство, модуль питания устройства для обеспечения электроэнергией всех модулей устройства, причем каждое устройство-участник блокчейн-коммуникации взаимодействует с центральным сервером либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Новая совокупность существенных признаков позволяет достичь технического результата предотвращения угрозы компрометации адресной информации и пользовательских данных, хранящихся в блокчейне, за счет системы организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных, которая содержит как минимум, один центральный сервер, включающий модуль приема/передачи информации в сети связи, общую шину обмена информацией, модуль обработки «hello-сообщений», модуль обработки запросов на получение актуальной копии списка в текущий момент времени IP-адресов в сети, модуль памяти, модуль питания устройства, и, как минимум, три устройства - участника блокчейн-коммуникации, включающие модуль приема/передачи информации в сети связи, общую шину обмена информацией, модуль отправки запросов

на серверы, модуль клиента для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения», модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна, модуль сервера обработки транзакций для участия в процессе консенсуса, модуль базы данных, модуль памяти, модуль ввода/вывода информации, модуль питания устройства, причем каждое устройство-участник блокчейн-коммуникации взаимодействует с центральным сервером либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Проведенный анализ уровня техники позволил установить, что аналоги, характеризующиеся совокупностью признаков, тождественных всем признакам заявленного способа организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных, отсутствуют. Следовательно, заявленное изобретение соответствует условию патентоспособности «новизна».

Результаты поиска известных решений в данной и смежных областях техники с целью выявления признаков, совпадающих с отличительными от прототипа признаками заявленного объекта, показали, что они не следуют явным образом из уровня техники. Из уровня техники также не выявлена известность влияния предусматриваемых существенными признаками заявленного изобретения преобразований на достижение указанного технического результата. Следовательно, каждое из заявленных изобретений соответствует условию патентоспособности «изобретательский уровень».

Для более понятной иллюстрации технических решений согласно вариантам осуществления настоящего изобретения ниже приведено краткое описание сопроводительных чертежей.

На фиг. 1 - способ организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных;

на фиг. 2 - схема работы сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна;

на фиг. 3 - схема работы сервера обработки транзакций, принимающего участие в процессе консенсуса;

на фиг. 4 - анализ транзакции на наличие входящих сообщений;

на фиг. 5 - логика расшифрования транзакции;

на фиг. 6 - анализ принятой копии блокчейна;

на фиг. 7 - логика выбора одной из функций, реализованных в системе;

на фиг. 8 - функции отображения истории переписки с указанным пользователем и отправке ему текстового сообщения;

на фиг. 9 - схема взаимодействия с распределенной системой хранения данных;

на фиг. 10 - схема отправки файла в распределенное хранилище;

на фиг. 11 - схема запроса файла из распределенного хранилища;

на фиг. 12 - схема взаимодействия устройств системы;

на фиг. 13 - схема центрального сервера;

на фиг. 14 - схема устройства-участника процесса блокчейн-коммуникации.

Ниже будут полностью и четко описаны технические решения для вариантов осуществления настоящего изобретения со ссылками на сопроводительные чертежи. Должно быть, очевидно, что варианты осуществления, описанные ниже, являются только частью настоящего изобретения, а не всеми возможными вариантами осуществления настоящего изобретения. Все другие варианты осуществления, полученные специалистами в данной области техники на основе вариантов осуществления настоящего изобретения и не использующие творческий подход,

попадают под объем охраны настоящего изобретения.

Реализация заявленного способа заключается в следующем (фиг. 1):

1. Вводят название базы данных, в которой будет храниться вся необходимая для работы системы информация.

2. Вводят пароль к этой базе данных, для ее защиты от несанкционированного доступа.

3. Проверяют наличие базы данных - файла, содержащего базу данных.

4. Создают базу данных и формируют к ней пароль на основе введенного ранее.

5. Осуществляют расшифрование базы данных на введенном ранее пароле.

6. Проверяют корректность файла с базой данных - правильность расшифрования базы данных по сигнатуре.

7. При расшифровании файла базы данных запускают клиентскую программу, для отправки запросов центральному серверу, который отвечает за поддержание работоспособности сети путем ведения актуального списка IP-адресов доступных на данный момент в сети, на получение актуальной копии списка активных в сети на текущий момент IP-адресов, а также отправки ему «hello-сообщения», факт отправки которого означает, что клиент, отправивший это сообщение, на данный момент времени является активным участником сети.

8. Запускают сервер обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна. Хэш последней транзакции позволяет осуществить проверку всего предыдущего блокчейна на изменение, т.к. хэш каждой новой транзакции формируется из самой транзакции и хэша предыдущей транзакции. Актуальная копия блокчейна позволяет запрашивающему проанализировать все транзакции, происходившие в сети во время его отсутствия - когда запрашивающий не участвовал в процессе консенсуса.

В качестве примера поясним принцип работы сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна (фиг. 2):

8.1. Принимают сервером обработки запросов сообщения клиента с запросом.

8.2. Проверяют вид запроса: запрос на получение актуальной копии блокчейна или запрос на получение хэша последней транзакции.

8.3. Извлекают из базы данных актуальную копию блокчейна сервера, если пришедший запрос является запросом на получение актуальной копии блокчейна.

8.4. Извлекают из базы данных хэш последней транзакции сервера, если пришедший запрос является запросом на получение хэша последней транзакции.

8.5. Формируют и отправляют клиенту ответ в соответствии с принятым запросом.

9. Запускают сервер обработки транзакций для участия в процессе консенсуса. На этом сервере обрабатываются транзакции на добавление новых записей, которые инициализируют другие участники сети.

Более подробно объясним принцип работы сервера обработки транзакций, принимающего участие в процессе консенсуса (фиг. 3):

9.1. Принимают сервером обработки транзакций транзакции, поступившей от клиента.

9.2. Извлекают из транзакции предлагаемый клиентом ID и хэш последней транзакции из локальной копии блокчейна клиента.

9.3. Проверяют последний ID локальной копии блокчейна сервера обработки транзакций, при этом ID должен быть на единицу меньше, предлагаемого клиентом.

9.4. Осуществляют проверку правильности ID.

9.5. Проверяют хэш последней транзакции локальной копии блокчейна клиента, содержащийся в присланной клиентом транзакции, и хэш последней транзакции

локальной копии блокчейна сервера обработки транзакций, при этом хэш должен совпадать.

9.6. Проверяют равенство хэшей.

9.7. Отправляют клиенту ответ с положительным решением о подтверждении данным сервером обработки транзакций принятую от клиента транзакцию.

9.8. Если по результатам проверки на этапах 94 или 96 получен отрицательный результат, то отправляют клиенту сообщение об отказе данным сервером обработки транзакций подтверждать принятую от клиента транзакцию. На этом этапе обработка транзакции сервером обработки транзакций завершается.

9.9. Добавляют принятую от клиента транзакцию сервером обработки транзакций в свою локальную копию блокчейна.

9.10. Запускают алгоритм анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на этом сервере обработки транзакций.

Далее более подробно описываем принцип, по которому производится анализ транзакции на наличие входящих сообщений (фиг.4):

9.10.1. Извлекают из всей транзакции зашифрованную часть транзакции путем выделения, предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна.

9.10.2. Расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного сервером обработки транзакций пользователя на зашифрованную с использованием шифра Вернама часть транзакции с учетом смещения ключа, при этом зашифрованная часть транзакции будет считаться расшифрованной только в случае, если в поле отправителя, которое также зашифровано с использованием шифра Вернама, будет имя пользователя, на чьем ключе производится расшифровка, а в поле получателя будет имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции (фиг. 5).

9.10.3. Проверяют получилось ли расшифровать сообщение на одном из ключей добавленных пользователей, если получилось, то расшифрованная часть транзакции разбивается на логические элементы: время отправки, имя отправителя, имя получателя и передаваемое сообщение, которое может являться либо текстовым файлом размером до 1 Кбайта, либо текстовым сообщением; и заносится в локальную базу данных сервера обработки транзакций в открытом виде, если не получилось расшифровать сообщение на одном из ключей добавленных пользователей, то транзакция не предназначена для пользователя данного сервера обработки транзакций.

10. Запрашивают хэши транзакций у серверов всех доступных на данный момент времени в сети IP-адресов.

11. Выбирают по мажоритарному правилу актуальный хэш последней транзакции.

12. Осуществляют запрос на получение актуальной копии блокчейна у одного из отправителей актуального хэша последней транзакции.

13. Запускают алгоритм анализа принятой копии блокчейна. Этот алгоритм необходим для выбора из всего множества транзакций, которые произошли в сети во время отсутствия запрашивающего, тех транзакций, которые относятся к запрашивающему и дальнейшей обработки этих транзакций и запись в базу данных извлеченных из этих транзакций сведений.

Более подробно объясним принцип, по которому производится анализ принятой копии блокчейна (фиг. 6):

13.1. Удаляют из базы данных предыдущую локальную копию блокчейна и записывают в базу данных, принятую от сервера обработки запросов на получение

хэша последней транзакций и актуальной копии блокчейна, копию локального блокчейна.

13.2. Проверяют количество транзакций предыдущей локальной копии блокчейна с принятой копией, если у предыдущей локальной копии блокчейна число транзакций больше, чем у принятой копии, то значит произошло обновление глобального блокчейна.

13.3. Проверяют, произошло ли обновление глобального блокчейна. Если да, то обнуляют смещения ключей для всех добавленных пользователей, если не произошло, все смещения оставляют со старыми значениями.

13.4. Анализируют каждую транзакцию на наличие входящих сообщений.

14. Отображают список добавленных пользователей с количеством непрочитанных сообщений от каждого из них.

15. Осуществляют выбор одной из функций, реализованных в рассматриваемой системе, для выполнения.

Далее подробно описываем выбор одной из функций, реализованных в системе (фиг. 7):

15.1. Выбор номера функции для выполнения.

15.2. Проверяют выбрана ли функция №1. Если выбрана, то добавляют нового пользователя, при этом вводят имя нового пользователя и указывают путь к файлу с ключом для этого пользователя.

15.3. Проверяют выбрана ли функция №2. Если выбрана, то отображают переписку с конкретным пользователем и отправляют ему текстовое сообщение.

Далее приведено описании функции отображения истории переписки с указанным пользователем и отправке ему текстового сообщения (фиг. 8):

15.3.1. Отображают предыдущую переписку с данным пользователем из базы данных.

15.3.2. Вводят новое сообщение для пользователя.

15.3.3. Добавляют к сообщению, как минимум, поле времени отправки сообщения, поле имени отправителя, поле имени получателя и выполняют шифрование этих полей на ключе данного пользователя.

15.3.4. Извлекают из базы данных ID последней транзакции и увеличивают на единицу данное значение и формируют новый ID для транзакции.

15.3.5. Извлекают из базы данных хэш последней транзакции локальной копии блокчейна отправителя, причем, необходимо отметить, что хэш любой транзакции, за исключением первой, формируется путем вычисления хэша от двух значений: хэша транзакции, записанной в локальной копии блокчейна отправителя до текущей транзакции и текущей транзакции в локальной копии блокчейна отправителя.

15.3.6. Создают транзакцию для передачи, в которую входит новый ID, хэш последней транзакции локальной копии блокчейна отправителя и зашифрованные данные на пользовательском ключе с использованием шифра Вернама, что является ТНДШ системой при условии, что пользовательский ключ - абсолютно случайный набор бит.

15.3.7. Отправляют всем доступным IP-адресам из списка актуальных IP-адресов, присланным центральным сервером, т.е. находящимся в рассматриваемой нами системе, сформированную транзакцию.

15.3.8. Принимают от всех доступных IP-адресов из списка актуальных IP-адресов, присланным центральным сервером, т.е. находящихся в рассматриваемой нами системе, ответ о готовности подтверждения серверами обработки транзакций сформированной транзакции.

15.3.9. Проверяют по мажоритарному правилу факт подтверждения транзакции.

15.3.10. Если проверка прошла успешно, то добавляют транзакцию в локальную копию блокчейна отправителя.

15.3.11. Если проверка не прошла успешно, то выполняют запрос актуальной копии блокчейна.

15.3.12. Добавляют в незашифрованном виде поле - время отправки транзакции, поле - имя отправителя, поле - имя получателя и передаваемое сообщение в базу данных.

15.3.13. Выполняют смещение ключа для данного пользователя на длину зашифрованных данных.

15.4. Проверяют выбрана ли функция №3. Если выбрана, то отображают переписку с конкретным пользователем и отправляют этому пользователю текстовый файл размером до 1 Кбайта, причем, необходимо отметить, что размер 1 Кбайт был определен эмпирическим способом как наиболее рациональный максимальный размер транзакции для хранения в блокчейне. Максимальный размер транзакции может быть заменен на любой другой.

15.5. Проверяют выбрана ли функция №4. Если выбрана, то обновляют ключ шифрования для конкретного пользователя. Для обновления необходимо ввести имя пользователя и путь к файлу, содержащему новый ключ. Смещение для ключа будет помещено в нулевое значение.

15.6. Проверяют выбрана ли функция №5. Если выбрана, то отправляют файл в распределенное хранилище данных (фиг. 9).

Далее представлено более подробное описание отправки файла в распределенное хранилище (фиг. 10):

15.6.1. Вводят путь к исходному файлу.

15.6.2. Зашифровывают исходный файл алгоритмом шифрования «Advanced Encryption Standard» с длинной ключа 256 бит, в режиме гаммирования с обратной связью CFB, при этом ключ шифрования помещают в файл-ссылку. Режим шифрования гаммирование с обратной связью выбран для того, чтобы при отсутствии одной из частей зашифрованного файла, или неправильном порядке следования частей зашифрованного файла, невозможно было расшифровать ни файл целиком, ни его часть.

15.6.3. Вычисляют хэш зашифрованного файла, при этом помещают хэш в файл-ссылку.

15.6.4. Разбивают зашифрованный файл на N частей, каждая из которых записывается в отдельный файл, случайного размера, при этом всю метаинформацию и одну из N частей случайного размера, которая не отправляется в сеть, чтобы исключить возможность сбора всего зашифрованного файла, помещают в файл-ссылку.

15.6.5. Добавляют К ложных частей, для того, чтобы увеличить энтропию об исходном зашифрованном файле, при этом всю метаинформацию, необходимую для корректного восстановления зашифрованного файла, помещают в файл-ссылку.

15.6.6. Назначают каждому из (N-1+K) файлов R серверов в распределенной системе хранения данных - для организации дублирования и повышения отказоустойчивости системы, и вычисляют хэш каждого из этих файлов, при этом всю метаинформацию помещают в файл-ссылку.

15.6.7. Перемешивают (N-1+K) файлов - меняют очередь отправки в сеть частей зашифрованного файла, при этом всю метаинформацию помещают в файл-ссылку.

15.6.8. Отправляют каждый из (N-1+K) файлов на R серверов в распределенной системе хранения данных с использованием Proxy-серверов, VPN или системы подмены IP-адресов - для затруднения выявления отправителя каждого из файлов.

15.6.9. Формируют окончательный вариант файла-ссылки.

15.7. Проверяют выбрана ли функция №6. Если выбрана, то запрашивают файл из распределенного хранилища данных (фиг. 9).

Далее представлено более подробное описание запроса файла из распределенного хранилища (фиг. 11):

15.7.1. Вводят путь к файлу-ссылке на исходный файл.

15.7.2. Извлекают из файла-ссылки метаинформацию об исходном файле: количество частей, на которые был разбит исходный файл, количество ложных частей, имена файлов на удаленных серверах, правила перемешивания файлов перед отправкой, IP-адреса серверов хранения файлов, ключ шифрования, на котором зашифрован весь файл, хэши частей файла, хэш всего файла.

15.7.3. Запрашивают с серверов части файлов с использованием Proxy-серверов, VPN или системы подмены IP-адресов с целью затруднить определение получателя частей зашифрованного файла.

15.7.4. Проверяют хэши частей файла таким образом, что если с серверов в распределенной системе хранения данных пришла, как минимум, одна правильная копия каждой части файла и ее хэш правильный, то переходят к следующему шагу, если хоть одно из вышеперечисленных условий не выполняется, то выдают сообщение об ошибке и прекращают работу сбора исходного файла.

15.7.5. Перемешивают принятые части файла, чтобы выстроить их в правильном порядке, причем, необходимо отметить, что вся необходимая метаинформация содержится в файле-ссылке.

15.7.6. Отделяют ложные части файла.

15.7.7. Выполняют сборку зашифрованного исходного файла.

15.7.8. Проверяют хэш всего зашифрованного файла, если проверка неудачная, то выдают сообщение об ошибке и прекращают работу сбора исходного файла.

15.7.9. Выполняют расшифрование зашифрованного файла.

15.8. Проверяют выбрана ли функция №7. Если выбрана, то обновляют интерфейс отображения пользовательской информации.

15.9. Проверяют выбрана ли функция №8. Если выбрана, то осуществляют безопасный выход из системы.

16. Проверяют факт того, что выбрана функция безопасного выхода из системы.

17. Если не выбрана функция безопасного выхода из системы, то осуществляют выполнение выбранной функции. После выполнения данной функции снова осуществляется выбор функции для выполнения (этап 15).

18. Если выбрана функция безопасного выхода из системы, или результат проверки корректности расшифрования файла базы данных в пункте 6 отрицательный - файл базы данных не был корректно расшифрован, то осуществляется шифрование базы данных на введенном ранее корректном пароле.

19. Завершают работу с базой данных.

Заявленный способ организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных позволяет достичь технического результата невозможности компрометации отправителей и получателей файлов в распределенной системе хранения данных за счет того, что при отправке файла в распределенное хранилище данных устройство-участник блокчейн-коммуникации зашифровывает файл с помощью симметричного шифрования на алгоритме aes-256 в режиме CFB. При этом распределенное хранилище данных может состоять из любых сторонних хранилищ данных. Отправляемый файл разбивается на несколько частей, вводятся ложные части, перемешиваются файлы перед отправкой,

назначают каждому файлу, который содержит часть зашифрованного файла или ложную часть файла, серверы, на которых будут храниться эти файлы. Вышеперечисленные меры, с одной стороны, значительно усложняют задачу сбора статистики об отправляемых файлах, а с другой стороны, гарантируют невозможность дешифрования файлов, хранящихся в распределенной системе хранения данных. С целью невозможности определения отправителя и получателя частей зашифрованного файла, все соединения осуществляют либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Система защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных (фиг. 12) состоит из одного центрального сервера 101 (фиг. 13) и, как минимум, трех устройств-участников блокчейн-коммуникации 102 (фиг. 14).

Центральный сервер (фиг. 13) содержит модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль обработки «hello-сообщений» для приема от устройств-участников блокчейн-коммуникации (фиг. 14) «hello-сообщений» и обновление актуального списка доступных в текущий момент времени IP-адресов в сети, модуль обработки запросов на получение актуальной копии списка в текущий момент времени IP-адресов в сети для отправки актуального списка доступных в данный момент времени IP-адресов в сети по запросу, модуль памяти для хранения актуального списка доступных на текущий момент времени IP-адресов в сети, модуль питания устройства для обеспечения электроэнергией всех модулей устройства.

Устройство-участник блокчейн-коммуникации (фиг. 14) содержит модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль отправки запросов на серверы для отправки запросов на серверы обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна и на серверы обработки транзакций, модуль клиента для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения» для идентификация устройства в сети как доступного посредством отправки на центральный сервер «hello-сообщения» и получение актуального списка доступных на данный момент времени IP-адресов в сети, модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна для отправки хэша последней транзакции по запросу и отправки копии локального блокчейна по запросу, модуль сервера обработки транзакций для участия в процессе консенсуса для обработки транзакций и отправки ответа на запрос о ее подтверждении или отклонении, модуль базы данных для хранения и предоставления доступа модулям ко всей необходимой информации, модуль памяти для хранения файлов и файлов-ссылок, а так же базы данных, модуль ввода/вывода информации для осуществления ввода/вывода информации в устройство, модуль питания устройства для обеспечения электроэнергией всех модулей устройства.

Центральный сервер может быть реализован по известной схеме, например, как персональный компьютер (ПК) на базе процессора Intel Corei3-10100 с тактовой частотой 3.6 МГц, оперативной памятью DDR4 объемом 2 ГБ и установленной операционной системой (ОС) Ubuntu 16.04 LTS.

Устройство-участник блокчейн-коммуникации может быть реализовано по известной схеме, например, как персональный компьютер (ПК) на базе процессора Intel Core i3-10100 с тактовой частотой 3.6 МГц, оперативной памятью DDR4 объемом 2 ГБ и

установленной операционной системой Ubuntu 16.04 LTS.

Интерактивный процесс системы защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных описан ниже.

При входе в систему устройство-участник блокчейн-коммуникации отправляет центральному серверу «hello-сообщение», в ответ получает подтверждение приема «hello-сообщения» центральным сервером, после чего запрашивает у центрального сервера актуальный список доступных в сети IP-адресов. Вышеперечисленные действия периодически повторяются, т.к. устройства-участники блокчейн-коммуникации выходят из сети и заходят в сеть.

Далее устройству-участнику блокчейн-коммуникации необходимо получить актуальную копию глобального блокчейна сети, т.к. за время его отсутствия в сети глобальный блокчейн мог измениться. Для этого устройство-участник блокчейн-коммуникации запрашивает у всех доступных IP-адресов, а именно у серверов обработки запросов на хэш последней транзакции или на получение актуальной копии блокчейна, в сети хэш последней транзакции из их локальной копии блокчейна и выбирает по мажоритарному признаку актуальный хэш последней транзакции и запрашивает у одного из отправителей актуального хэша последней транзакции его локальную копию блокчейна, принимая ее актуальной. Далее выполняет анализ на входящие сообщения в этом блокчейне.

При отправке устройством-участником блокчейн-коммуникации одному из добавленных пользователей текстового сообщения или текстового файла размером до 1 Кбайта оно отправляет всем доступным IP-адресам в сети передаваемую транзакцию в виде: «предлагаемый ID хэш последней транзакции из своей локальной копии блокчейна зашифрованное на пользовательском ключе для ТНДШ системы сообщение». В него входят: поля времени отправки, отправителя, получателя и самой передаваемой информации.

Данная транзакция обрабатывается всеми серверами обработки транзакций и если они согласны подтвердить принятую транзакцию, то они отсылают в ответ «YES» и добавляют в свою локальную копию блокчейна присланную транзакцию, а если не согласны, то отсылают «NO». Если большинство принятых от серверов обработки транзакций ответов будут «YES», то происходит добавление транзакции в локальную копию блокчейна устройства-участника блокчейн-коммуникации, отправлявшего транзакцию в сеть, если большинство ответов будут «NO», то транзакция не добавляется и производиться запрос актуальной копии блокчейна, т.к. если большинство IP-адресов в сети не подтвердили транзакцию, значит на данный момент у отправлявшего транзакцию устройства-участника блокчейн-коммуникации содержится неправильная копия блокчейна.

Организация удаленного обмена очередными ключами для ТНДШ системы, а также любых других файлов, без угрозы их компрометации осуществляется следующим образом.

При отправке файла в распределенное хранилище данных, которое может состоять из любых сторонних хранилищ данных 103, устройство-участник блокчейн-коммуникации зашифровывает файл с помощью симметричного шифрования с использованием алгоритма aes-256 в режиме CFB на ключе для ПНДШ системы. Разбивает отправляемый файл на несколько частей, вводит ложные части, перемешивает файлы перед отправкой, назначает каждому файлу, который содержит часть настоящего зашифрованного файла или ложную часть файла, серверы, где будут храниться эти

файлы. Для того, чтобы затруднить определение получателя частей зашифрованного файла, все соединения происходят либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Для того чтобы дешифровать зашифрованный исходный файл необходимо собрать зашифрованный файл целиком и в правильном порядке. Для того, чтобы это было невозможно сделать, одну или несколько из реальных частей зашифрованного исходного файла помещаем в файл-ссылку, который находится на устройстве-участнике блокчейн-коммуникации и в будущем будет передаваться по сети только в зашифрованном на пользовательском ключе для ТНДШ системы виде, а следовательно, если не произойдет компрометация пользовательского ключа - однозначно дешифровать его будет невозможно, а следовательно невозможно получить недостающую часть исходного зашифрованного файла, а значит, даже при условии, что злоумышленник соберет воедино все части файла, отделит ложные части от настоящих, выстроит их в правильном порядке, определит какой части не хватает, то не сможет однозначно дешифровать файл, т.к. у него не будет недостающей части исходного зашифрованного файла.

Вся метаинформация о каждом отправляемом файле в распределенную систему хранения заносится в файл-ссылку: на какие серверы, или несколько серверов - для дублирования, отправлен каждый маленький файл - файл, являющийся частью исходного зашифрованного файла, хэш каждого маленького файла, хэш всего файла, часть зашифрованного исходного файла, ключ шифрования на котором был зашифрован исходный файл, правило перемешивания маленьких файлов, сколько ложных, а сколько реальных файлов и т.п.

При необходимости запроса файла из распределенного хранилища устройство-участник блокчейн-коммуникации выполняет вышеперечисленные действия в обратном порядке: обращается к файлу-ссылке, запрашивает с серверов части исходного зашифрованного файла, проверяет хэши принятых маленьких файлов, перемешивает в правильном порядке, отделяет ложные части от реальных, собирает исходный зашифрованный файл целиком путем добавления недостающих частей из файла-ссылки, проверяет хэш всего зашифрованного файла и расшифровывает.

Благодаря новой совокупности существенных признаков достигается указанный технический результат за счет использования системы организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных, которая содержит как минимум, один центральный сервер, включающий модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль обработки «hello-сообщений» для приема от устройств-участников блокчейн-коммуникации «hello-сообщений» и обновление актуального списка доступных в текущий момент времени IP-адресов в сети, модуль обработки запросов на получение актуальной копии списка в текущий момент времени IP-адресов в сети для отправки актуального списка доступных в текущий момент времени IP-адресов в сети по запросу, модуль памяти для хранения актуального списка доступных на данный момент времени IP-адресов в сети, модуль питания устройства для обеспечения электроэнергией всех модулей устройства и, как минимум, три устройства - участника блокчейн-коммуникации, включающие модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль отправки запросов на серверы для отправки запросов на серверы обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна и на серверы

обработки транзакций, модуль клиента для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения» для идентификация устройства в сети как доступного посредством отправки на центральный сервер «hello-сообщения» и получение актуального списка доступных на данный момент времени IP-адресов в сети, модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна для отправки хэша последней транзакции по запросу и отправки копии локального блокчейна по запросу, модуль сервера обработки транзакций для участия в процессе консенсуса для обработки транзакций и отправки ответа на запрос о ее подтверждении или отклонении, модуль базы данных для хранения и предоставления доступа модулям ко всей необходимой информации, модуль памяти для хранения файлов и файлов-ссылок, а так же базы данных, модуль ввода/вывода информации для осуществления ввода/вывода информации в устройство, модуль питания устройства для обеспечения электроэнергией всех модулей устройства, причем каждое устройство-участник блокчейн-коммуникации взаимодействует с центральным сервером либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

Согласно теореме Шеннона о совершенном шифре (см. Зубов А.Ю. Совершенные шифры: Вступительное слово чл.-корр. РАН Б.А. Севастьянова. - М.: Гелиос АРВ, 2003. - 160 с.) ТНДШ системы шифрования требуют, чтобы используемые ключи шифрования были равновероятны между собой - являлись случайными последовательностями бит, и, чтобы количество вариантов ключей было больше либо равно количеству вариантов передаваемой информации, предназначенной для шифрования на данном ключе.

В предлагаемом способе выполнение условия равновероятности ключей шифрования возлагается на пользователя системы (участника блокчейн-коммуникации), позволяя ему самостоятельно генерировать ключи наиболее приемлемым для него способом. Выполнение второго условия заключается в том, что для каждого отправляемого сообщения существует отдельный одноразовый блокнот, представленный в виде набора бит длиною равной длине передаваемого сообщения в бинарном виде, а, следовательно, количество вариантов ключа равно количеству вариантов передаваемой информации, который используется один лишь раз для шифрования этого сообщения. Использование этой же последовательности бит для повторного шифрования другого сообщения не допускается.

В частности, в данном способе используется «XOR-шифрование» -Шифр Вернама, который обладает абсолютной криптографической стойкостью при определенных условиях (см. Зубов А.Ю. Совершенные шифры: Вступительное слово чл.-корр. РАН Б.А. Севастьянова. - М.: Гелиос АРВ, 2003.- 160 с).

Но для того, чтобы безопасно обменяться ключами для данного алгоритма шифрования, нужен либо абсолютно надежный канал связи, а это только канал, который уже защищен ТНДШ системой, либо личная встреча. Однако, передача 1 Гбайта нового ключа для ТНДШ системы по абсолютно надежному каналу, защищенного ТНДШ системой, с израсходованием 1 Гбайта текущего ключа для ТНДШ системы, теряет здравый смысл.

Предлагаемая распределенная система хранения файлов решает проблему безопасной передачи ключевой информации для ТНДШ систем по каналам связи. При отправке файла в распределенное хранилище данных, которое может состоять из любых сторонних хранилищ данных, устройство-участник блокчейн-коммуникации зашифровывает файл с помощью симметричного шифрования с использованием

алгоритма aes-256 в режиме CFB на ключе для ПНДШ системы. Разбивает отправляемый файл на несколько частей, вводит ложные части, перемешивает файлы перед отправкой, назначает каждому файлу, который содержит часть настоящего зашифрованного файла или ложную часть файла, серверы, где будут храниться эти файлы. Для того, чтобы затруднить процесс сборки частей зашифрованного исходного файла и процесс идентификации отправителей файлов злоумышленником, все соединения происходят либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN. Так как в распределенном хранилище данных содержатся лишь зашифрованные части файла, причем, одна или несколько частей исходного зашифрованного файла содержаться только в файле-ссылке, а исходный файл зашифрован методом гаммирования с обратной связью, следовательно, без всех частей исходного зашифрованного файла дешифрование будет бессмысленно. Причем, файл-ссылка будет передаваться уже по каналу связи, защищенному ТНДШ системой шифрования. Исходя их этого, можно сделать вывод, что передавая по каналу связи, защищенному ТНДШ системой шифрования, файл-ссылку размером не более 1 Кбайта, будем получать полноценный ключ для ТНДШ системы любого размера. В качестве метода шифрования в предлагаемом способе организации защищенного обмена информацией с использованием технологии блокчейн и распределенных систем хранения данных выбран метод шифрования Вернама. Таким образом, снижается к минимуму угроза дешифрования сообщений в будущем. Однако, чтобы начать процесс коммуникации, пользователям предлагаемой системы необходимо один раз встретиться лично и обменяться ключами шифрования. Таким образом, при использовании предлагаемого способа и системы, количество личных встреч или иных способов абсолютно надежной передачи ключей шифрования для ТНДШ систем уменьшается от неопределенного количества раз, в зависимости от объема информации, которой будут обмениваться пользователи системы (участники блокчейн-коммуникации), до одного раза.

Для обеспечения анонимности в сети был изменен классический процесс консенсуса - отсутствует необходимость подтверждать правильность данных транзакций: информация - зашифрована, отправитель и получатель - зашифрованы;

Это позволяет свести консенсус к упорядочиванию всего множества транзакций всеми пользователями, доступными на данный момент времени в сети. Философия предлагаемой системы - каждый пользователь может отправлять любую информацию любому другому пользователю, но расшифровать входящее сообщение и понять, что оно адресовано ему сможет лишь пользователь, у которого есть такой же ключ шифрования, на котором зашифровано сообщение. Таким образом, имеется блокчейн, где неизвестны ни отправители, ни получатели, ни содержание сообщений, что решает вопрос анонимности пользователей и конфиденциальности передаваемых данных.

(57) Формула изобретения

1. Способ организации защищенного обмена информацией с использованием технологии блокчейн и распределённых систем хранения данных, заключающийся в том, что вводят название базы данных, при этом дополнительно вводят пароль к этой базе данных, проверяют наличие базы данных на устройстве, при наличии базы данных осуществляют ее расшифрование с использованием симметричного шифрования AES-256, проверяют корректность файла с базой данных, при отсутствии базы данных создают новую базу данных, при этом формируют пароль к новой базе данных, необходимый для шифрования и расшифрования файла с новой базой данных по алгоритму AES-256, завершают работу с базой данных, отличающийся тем, что после

проверки на корректность файла с базой данных либо после создания новой базы данных запускают сетевые сервисы, как минимум запускают клиент для запросов у центрального сервера актуального списка IP-адресов в сети и отправки «hello-сообщения», запускают сервер обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна, запускают сервер обработки транзакций для участия в процессе консенсуса, после этого запрашивают хэши транзакций у серверов всех доступных в сети IP-адресов, выбирают по мажоритарному правилу актуальный хэш последней транзакции, осуществляют запрос на получение актуальной копии блокчейна у одного из отправителей актуального хэша последней транзакции, запускают алгоритм анализа принятой актуальной копии блокчейна, отображают список добавленных пользователей, осуществляют выбор как минимум одной из функций: добавление нового пользователя, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового сообщения, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового файла размером до 1 Кбайта, обновление ключа шифрования для указанного пользователя, отправка файла в распределенное хранилище, запрос файла из распределенного хранилища, обновление интерфейса отображения информации, безопасный выход из системы, которую необходимо выполнить системе, проверяют факт того, что выбрана функция безопасного выхода из системы, если данная функция не выбрана, то осуществляют выполнение выбранной функции, при этом после выполнения выбранной функции осуществляют выбор новой функции, если выбрана функция безопасного выхода из системы, то осуществляют шифрование базы данных с использование введенного ранее пароля по алгоритму AES-256.

2. Система защищенного обмена информацией с использованием технологии блокчейн и распределённых систем хранения данных, состоящая из как минимум одного центрального сервера, включающего модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль обработки «hello-сообщений» для приема от устройств-участников блокчейн-коммуникации «hello-сообщений» и обновление актуального списка доступных в текущий момент времени IP-адресов в сети, модуль обработки запросов на получение актуальной копии списка доступных в текущий момент времени IP-адресов в сети для отправки актуального списка доступных в текущий момент времени IP-адресов в сети по запросу, модуль памяти для хранения актуального списка доступных на текущий момент времени IP-адресов в сети, модуль питания устройства для обеспечения электроэнергией всех модулей устройства и как минимум трех устройств-участников блокчейн-коммуникации, включающие модуль приема/передачи информации в сети связи для приема и передачи данных по каналам связи, общую шину обмена информацией для обмена информацией между модулями устройства, модуль отправки запросов на серверы для отправки запросов на модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна и на модуль сервера обработки транзакций для участия в процессе консенсуса, модуль клиента для запросов у центрального сервера актуального списка IP-адресов в сети связи и отправки «hello-сообщения» для идентификации устройства в сети как доступного посредством отправки на центральный сервер «hello-сообщения» и получения актуального списка доступных на текущий момент времени IP-адресов в сети, модуль сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна для отправки хэша последней транзакции по запросу и отправки копии локального блокчейна по запросу, модуль

сервера обработки транзакций для участия в процессе консенсуса для обработки транзакций и отправки ответа на запрос об ее подтверждении или отклонении, модуль базы данных для хранения данных как минимум о транзакциях, пользовательских ключах шифрования, истории сообщений, модуль памяти для хранения файлов и файлов-ссылок, а также базы данных, модуль ввода/вывода информации для осуществления ввода/вывода информации в устройство, модуль питания устройства для обеспечения электроэнергией всех модулей устройства, причем каждое устройство-участник блокчейн-коммуникации взаимодействует с центральным сервером либо через механизмы подмены IP-адресов, либо через Proxy-серверы, либо через VPN.

3. Способ по п.1, в котором при запуске сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна: принимают сервером обработки запросов сообщения клиента с запросом, проверяют вид запроса сервером, если поступает запрос на получение хэша последней транзакции копии блокчейна сервера, то извлекают из базы данных хэш последней транзакции, если поступает запрос на получение актуальной копии блокчейна сервера, то извлекают из базы данных актуальную копию блокчейна сервера, формируют сообщение для отправки клиенту с учетом характеристики информации извлеченной из базы данных, отправляют сформированный ответ клиенту.

4. Способ по п.1, в котором при запуске сервера обработки транзакций: принимают сервером обработки транзакций транзакции от клиента, извлекаютиз транзакции предлагаемый клиентом ID и хэш последней транзакции из локальной копии блокчейна клиента, проверяют последний ID локальной копии блокчейна сервера обработки транзакций, при этом ID должен быть на единицу меньше предлагаемого клиентом, осуществляют проверку правильности ID, если проверка дала отрицательный результат, то отправляют клиенту сообщение об отказе данным сервером обработки транзакций подтверждать принятую от клиента транзакцию, если проверка прошла успешно, то проверяют хэш последней транзакции локальной копии блокчейна клиента, содержащийся в присланной клиентом транзакции, и хэш последней транзакции локальной копии блокчейна сервера обработки транзакций, при этом хэш должен совпадать, если хэш не совпадает, то отправляют клиенту сообщение об отказе данным сервером обработки транзакций подтверждать принятую транзакцию от клиента, если хэш совпадает, то отправляют клиенту ответ с положительным решением о подтверждении данным сервером обработки транзакций принятую транзакцию от клиента, добавляют принятую от клиента транзакцию сервером обработки транзакций в свою локальную копию блокчейна, запускают алгоритм анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на этом сервере обработки транзакций.

5. Способ по п.4, в котором при запуске алгоритма анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на этом сервере обработки транзакций: извлекают из всей транзакции зашифрованную часть транзакции путем выделения предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна, расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного сервером обработки транзакций пользователя на зашифрованную часть транзакции с учетом смещения ключа, при этом зашифрованная часть транзакции расшифруется только в случае, если в поле отправителя, которое также зашифровано, будет имя пользователя, на чьем ключе производится расшифровка, а в поле получателя будет имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции, проверяют, получилось ли расшифровать

сообщение на одном из ключей добавленных пользователей, если получилось, то расшифрованная часть транзакции разбивается на логические элементы: время отправки, имя отправителя, имя получателя и передаваемое сообщение, которое может являться либо текстовым файлом размером до 1 Кбайта, либо текстовым сообщением, и заносится в локальную базу данных сервера обработки транзакций в открытом виде, если не получилось расшифровать сообщение на одном из ключей добавленных пользователей, то транзакция не предназначена для пользователя данного сервера обработки транзакций.

6. Способ по п.1, в котором при выполнении запроса на получение актуальной копии блокчейна: удаляют из базы данных предыдущую локальную копию блокчейна и записывают в базу данных принятую от сервера обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна копию локального блокчейна, проверяют количество транзакций предыдущей локальной копии блокчейна с принятой копией, если у предыдущей локальной копии блокчейна число транзакций больше, чем у принятой копии, то значит произошло обновление глобального блокчейна, при этом обнуляют смещения ключей для всех добавленных пользователей, в случае, когда обновление глобального блокчейна не произошло, тогда оставляют значения смещения ключей для всех добавленных пользователей прежними, анализируют каждую транзакцию на наличие входящих сообщений, при этом извлекают из всей транзакции зашифрованную часть транзакции путем выделения предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна, расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного сервером обработки транзакций пользователя на зашифрованную часть транзакции с учетом смещения ключа, при этом зашифрованная часть транзакции расшифруется только в случае, если в поле отправителя, которое также зашифровано, будет имя пользователя, на чьем ключе производится расшифрование, а в поле получателя будет имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции, если получилось расшифровать сообщение на одном из ключей добавленных пользователей, то расшифрованная часть транзакции разбивается на логические элементы, такие как время отправки, имя отправителя, имя получателя и передаваемое сообщение, которое может являться либо текстовым файлом размером до 1 Кбайта, либо текстовым сообщением, и заносится в локальную базу данных сервера обработки транзакций в открытом виде, если не получилось расшифровать сообщение на одном из ключей добавленных пользователей, то транзакция не предназначена для пользователя данного сервера обработки транзакций.

7. Способ по п.1, в котором выбирают как минимум одну из функций: добавление нового пользователя, при этом вводят имя нового пользователя и указывают путь к файлу с ключом для этого пользователя, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового сообщения, просмотр истории переписки с указанным пользователем и отправка этому пользователю текстового файла размером до 1 Кбайта, обновление ключа шифрования для указанного пользователя, отправка файла в распределенное хранилище, запрос файла из распределенного хранилища, обновление интерфейса отображения пользовательской информации, безопасный выход из системы, которую необходимо выполнить системе.

8. Способ по п.1, в котором при запросе отправки файла в распределенное хранилище вводят путь к исходному файлу, зашифровывают исходный файл с применением алгоритма шифрования «Advanced Encryption Standard» с длиной ключа 256 бит в режиме гаммирования с обратной связью CFB, при этом ключ шифрования помещают в файл-

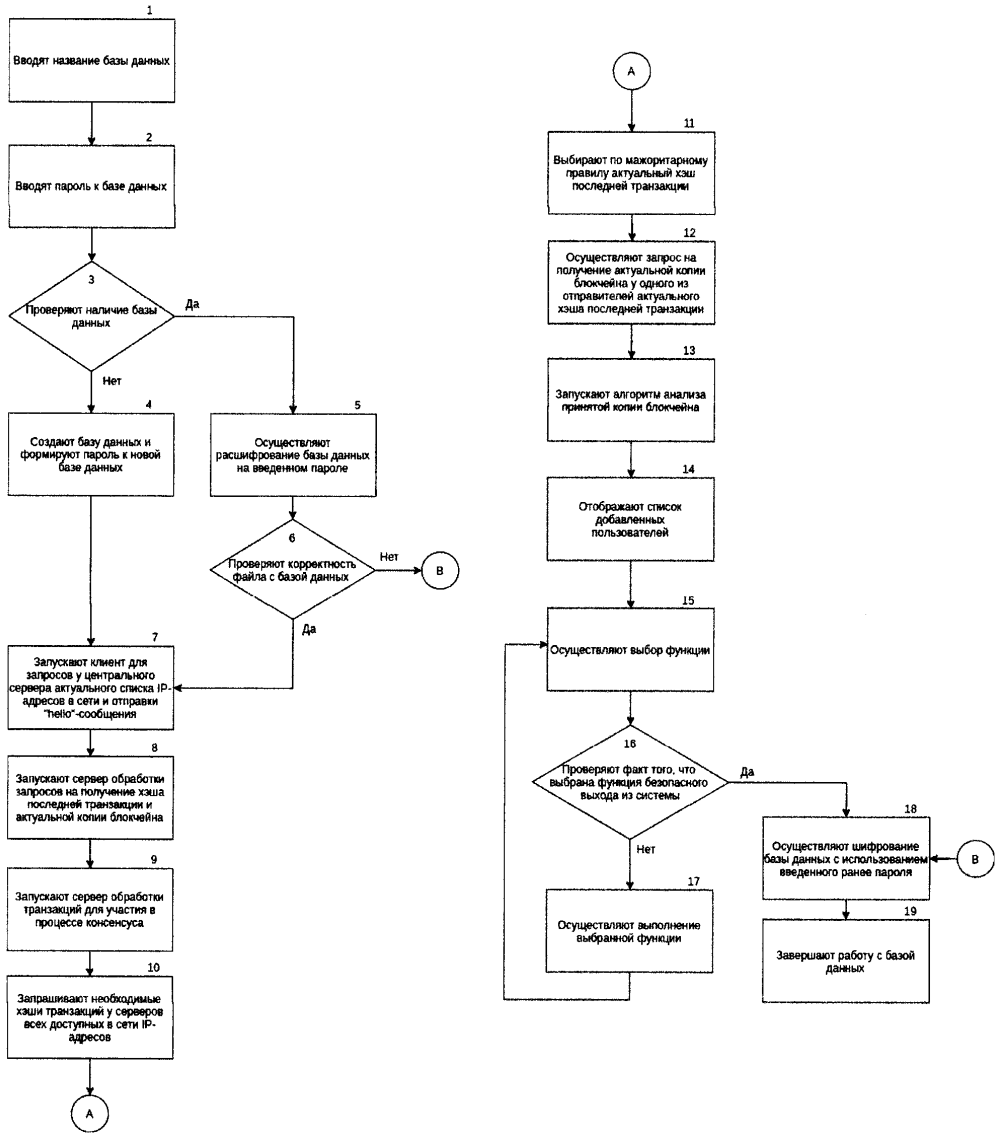
ссылку, вычисляют хэш зашифрованного файла, помещают хэш в файл-ссылку, разбивают зашифрованный файл на N частей (файлов) случайного размера, при этом всю метаинформацию и одну из N частей случайного размера, которая не отправляется в сеть, помещают в файл-ссылку, добавляют К ложных частей, при этом всю метаинформацию помещают в файл-ссылку, назначают каждому из (N-1+K) файлов R серверов в распределённой системе хранения данных и вычисляют хэш каждого из этих файлов, при этом всю метаинформацию помещают в файл-ссылку, перемешивают (N-1+K) файлов, при этом всю метаинформацию помещают в файл-ссылку, отправляют каждый из (N-1+K) файлов на R серверов в распределённой системе хранения данных с использованием механизма подмены IP-адресов либо через Proxy-серверы, либо через VPN, формируют окончательный вариант файл-ссылки.

9. Способ по п.1, в котором при запросе файла из распределенного хранилища вводят путь к файлу-ссылке на исходный файл, извлекают из файла-ссылки метаинформацию об исходном файле: количество частей, на которые был разбит исходный файл, количество ложных частей, имена файлов на удаленных серверах, правила перемешивания файлов перед отправкой, IP-адреса серверов хранения файлов, ключ шифрования, на котором зашифрован весь файл, хэши частей файла, хэш всего файла, запрашивают с серверов части файлов с использованием механизма подмены IP-адресов либо через Proxy-серверы, либо через VPN, проверяют хэши частей файла с учетом того, что если с серверов в распределённой системе хранения данных получена как минимум одна копия каждой части файла и ее хэш прошел проверку на правильность, то переходят к следующему шагу, если хоть одно условие не выполняется, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, перемешивают принятые части файла, отделяют ложные части файла, выполняют сборку зашифрованного исходного файла, проверяют хэш всего зашифрованного файла, если проверка отрицательная, то выдают сообщение об ошибке и прекращают работу сбора исходного файла, если проверка прошла успешно, то выполняют расшифрование зашифрованного (принятого) файла.

10. Способ по п.6, в котором при просмотре истории переписки с указанным пользователем и отправке ему текстового сообщения отображают предыдущую переписку с данным пользователем из базы данных, вводят новое сообщение для пользователя, добавляют к сообщению как минимум поле времени отправки сообщения, поле имени отправителя, поле имени получателя, выполняют шифрование этих полей на ключе данного пользователя, извлекают из базы данных ID последней транзакции и увеличивают на единицу текущее значение, формируют новый ID для транзакции, извлекают из базы данных хэш последней транзакции локальной копии блокчейна отправителя, создают транзакцию для передачи, в которую входит новый ID, хэш последней транзакции локальной копии блокчейна отправителя и зашифрованные данные на шифре Вернама с применением пользовательского ключа, отправляют всем доступным IP-адресам из списка актуальных IP-адресов, присланным центральным сервером, сформированную транзакцию, принимают от всех доступных IP-адресов из списка актуальных IP-адресов, присланных центральным сервером, ответ о готовности подтверждения серверами обработки транзакций сформированной транзакции, проверяют по мажоритарному правилу факт подтверждения транзакции, если проверка отрицательная, то выполняют запрос актуальной копии блокчейна согласно п.6, если проверка прошла положительно, то добавляют транзакцию в локальную копию блокчейна отправителя, добавляют в незашифрованном виде поля: время отправки транзакции, имя отправителя, имя получателя, передаваемое сообщение в базу данных,

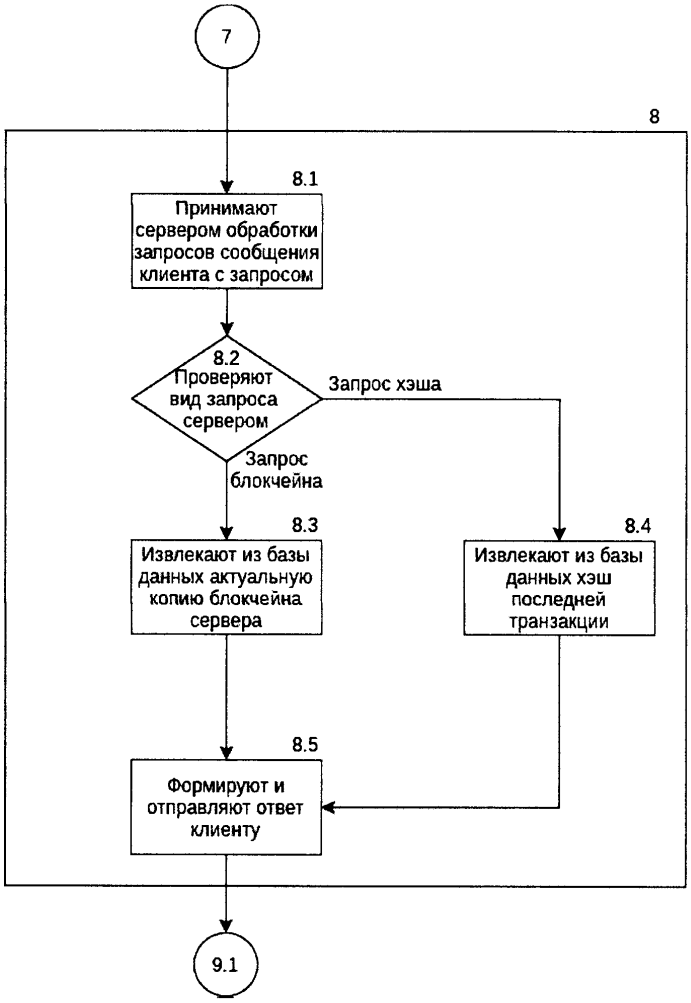
выполняют смещение ключа для данного пользователя на длину зашифрованных данных.

1


1
Вводят название базы данных
2
Вводят пароль к базе данных
3
Проверяют наличие базы данных
Да
Нет
4
Создают базу данных и формируют пароль к новой базе данных
5
Осуществляют расшифрование базы данных на введенном пароле
6
Проверяют корректность файла с базой данных
Нет
Да
B
7
Запускают клиент для запросов у центрального сервера актуального списка IP-адресов в сети и отправки "hello"-сообщения
8
Запускают сервер обработки запросов на получение хэша последней транзакции и актуальной копии блокчейна
9
Запускают сервер обработки транзакций для участия в процессе консенсуса
10
Запрашивают необходимые хэши транзакций у серверов всех доступных в сети IP-адресов
A
A
11
Выбирают по мажоритарному правилу актуальный хэш последней транзакции
12
Осуществляют запрос на получение актуальной копии блокчейна у одного из отправителей актуального хэша последней транзакции
13
Запускают алгоритм анализа принятой копии блокчейна
14
Отображают список добавленных пользователей
15
Осуществляют выбор функции
16
Проверяют факт того, что выбрана функция безопасного выхода из системы
Да
Нет
17
Осуществляют выполнение выбранной функции
18
Осуществляют шифрование базы данных с использованием введенного ранее пароля
B
19
Завершают работу с базой данных

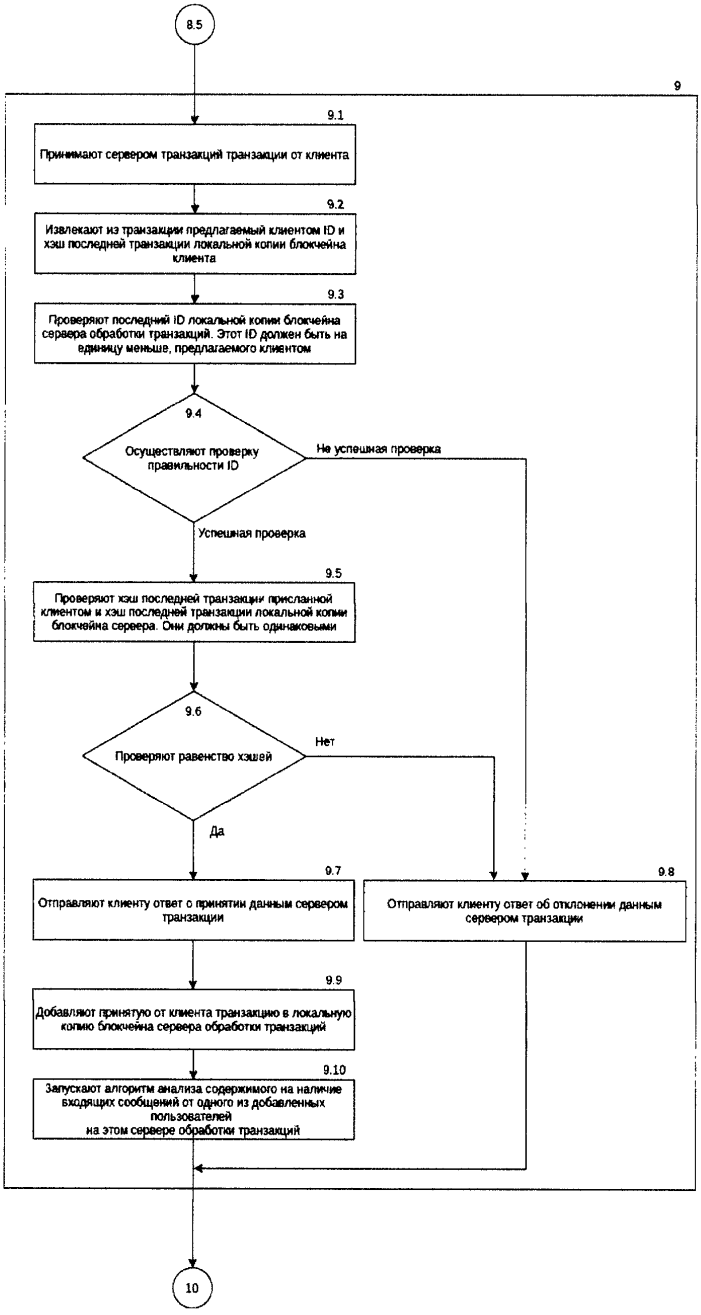
Фиг. 1

2


7
8
8.1
Принимают сервером обработки запросов сообщения клиента с запросом
8.2
Проверяют вид запроса сервером
Запрос хэша
Запрос блокчейна
8.3
Извлекают из базы данных актуальную копию блокчейна сервера
8.4
Извлекают из базы данных хэш последней транзакции
8.5
Формируют и отправляют ответ клиенту
9.1

Фиг. 2

3

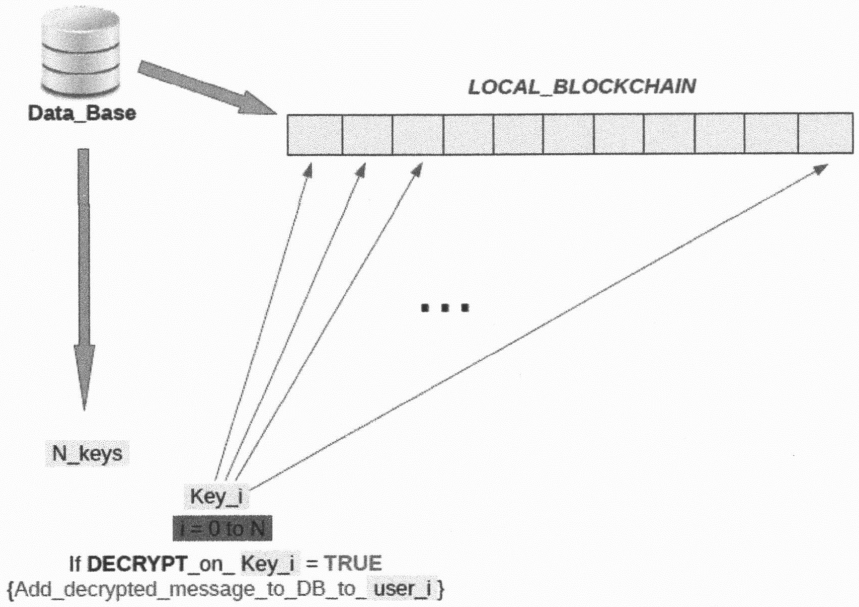

8.5
9
9.1
Принимают сервером транзакций транзакции от клиента
9.2
Извлекают из транзакции предлагаемый клиентом ID и хэш последней транзакции локальной копии блокчейна клиента
9.3
Проверяют последний ID локальной копии блокчейна сервера обработки транзакций. Этот ID должен быть на единицу меньше, предлагаемого клиентом
9.4
Осуществляют проверку правильности ID
Не успешная проверка
Успешная проверка
9.5
Проверяют хэш последней транзакции присланной клиентом и хэш последней транзакции локальной копии блокчейна сервера. Они должны быть одинаковыми
9.6
Проверяют равенство хэшей
Нет
Да
9.7
Отправляют клиенту ответ о принятии данным сервером транзакции
9.8
Отправляют клиенту ответ об отклонении данным сервером транзакции
9.9
Добавляют принятую от клиента транзакцию в локальную копию блокчейна сервера обработки транзакций
9.10
Запускают алгоритм анализа содержимого на наличие входящих сообщений от одного из добавленных пользователей на этом сервере обработки транзакций
10

Фиг. 3

4

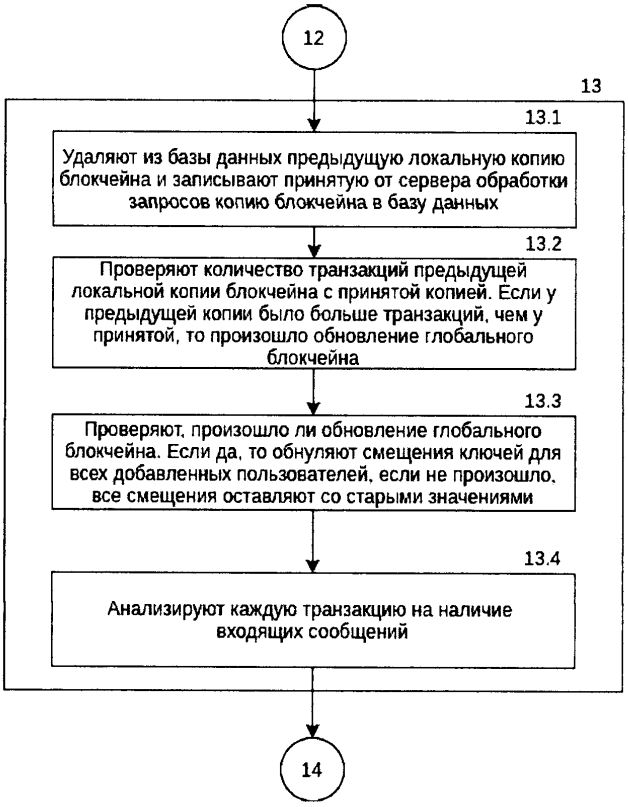

9.9
9.10
9.10.1
Извлекают из всей транзакции зашифрованную часть транзакции путем выделения предлагаемого клиентом ID и хэша последней транзакции из клиентской копии блокчейна
9.10.2
Расшифровывают закрытую часть транзакции путем наложения ключа каждого добавленного на сервере пользователя на зашифрованную часть транзакции (с учетом смещения ключа). Зашифрованная часть транзакции считается успешно расшифрованной если в поле отправителя (которое также зашифровано) - пользователь, на чьем ключе происходит попытка расшифровки, а в поле получателя - имя пользователя, чей сервер обработки транзакций выполняет расшифрование данной транзакции
9.10.3
Проверяют получилось ли расшифровать сообщение на одном из ключей добавленных пользователей, если получилось, то расшифрованная часть транзакции разбивается на логические элементы (время отправки, кто отправитель, кто получатель и само передаваемое сообщение, которое может являться либо текстовым файлом размером до 1 Кбайта, либо текстовым сообщением) и заносится в локальную базу данных сервера в открытом виде
10

Фиг. 4


Data_Base
LOCAL_BLOCKCHAIN
N_keys
Key_i
i = 0 to N
If DECRYPT_on_ Key_i = TRUE
{Add_decrypted_message_to_DB_to_ user_i }

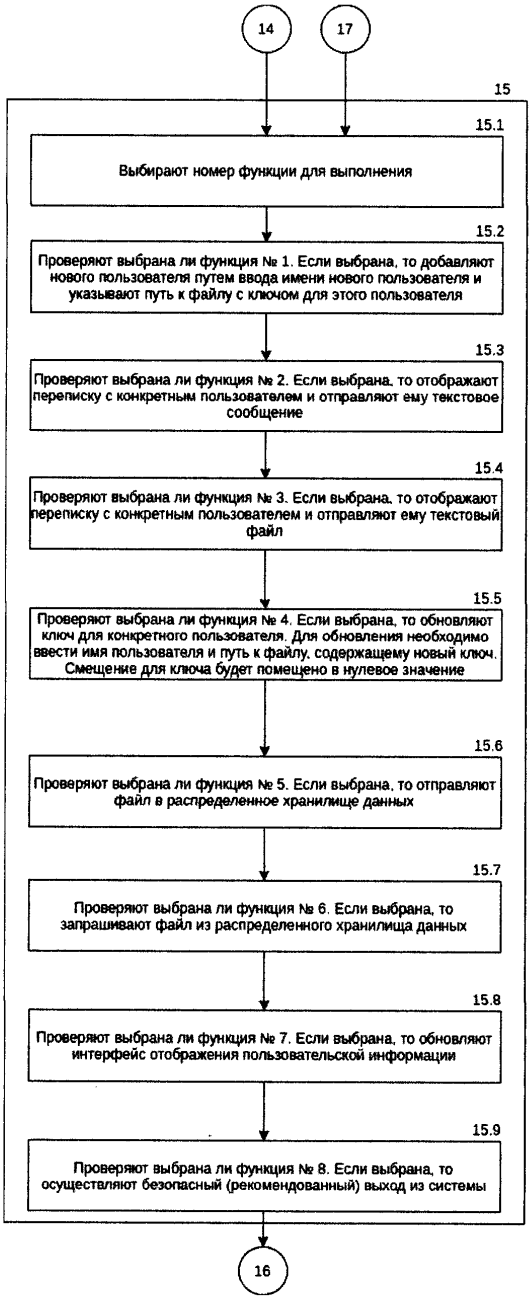
Фиг. 5

6


12
13
13.1
Удаляют из базы данных предыдущую локальную копию блокчейна и записывают принятую от сервера обработки запросов копию блокчейна в базу данных
13.2
Проверяют количество транзакций предыдущей локальной копии блокчейна с принятой копией. Если у предыдущей копии было больше транзакций, чем у принятой, то произошло обновление глобального блокчейна
13.3
Проверяют, произошло ли обновление глобального блокчейна. Если да, то обнуляют смещения ключей для всех добавленных пользователей, если не произошло, все смещения оставляют со старыми значениями
13.4
Анализируют каждую транзакцию на наличие входящих сообщений
14

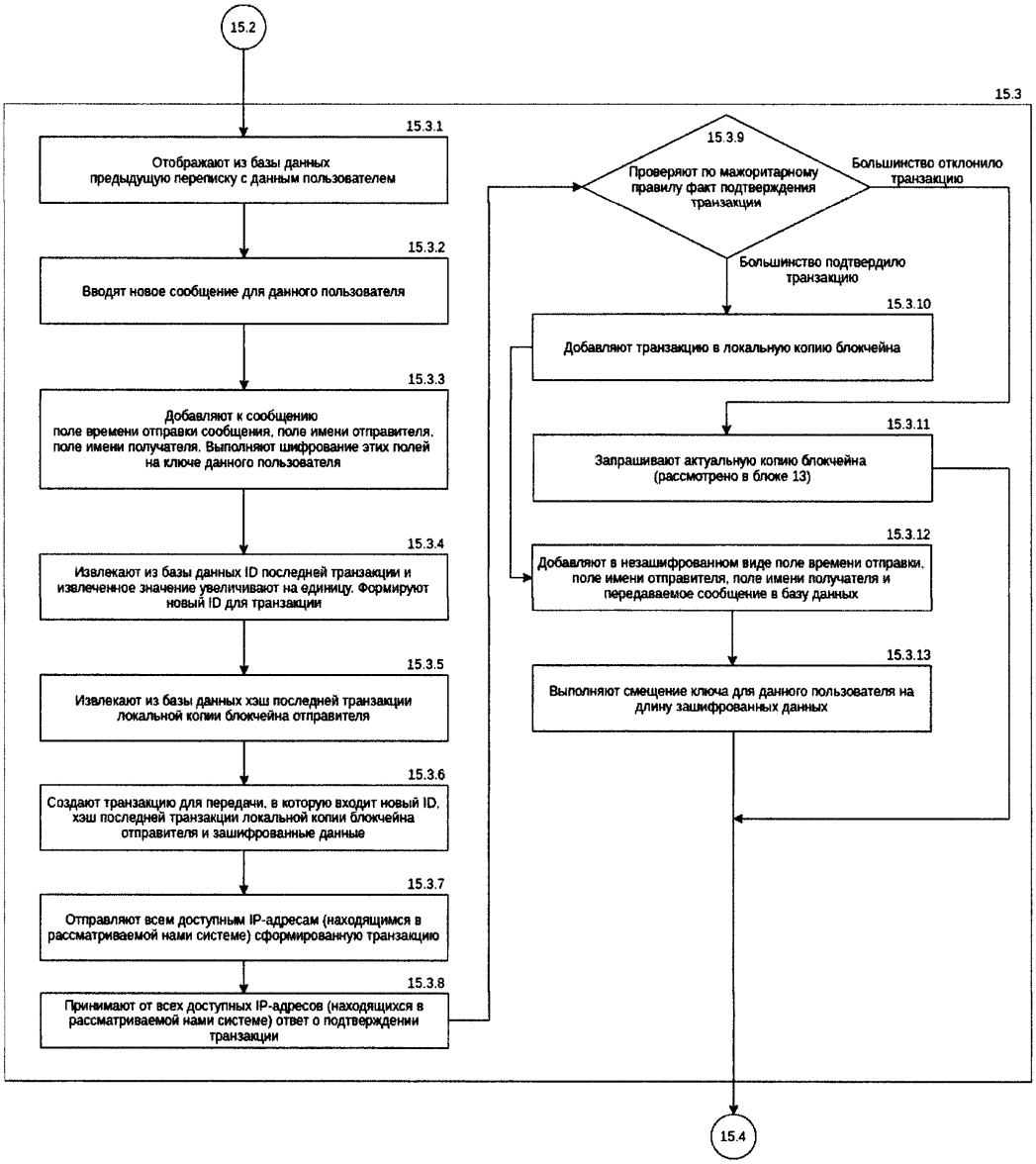
Фиг. 6

7


14
17
15
15.1
Выбирают номер функции для выполнения
15.2
Проверяют выбрана ли функция № 1. Если выбрана, то добавляют нового пользователя путем ввода имени нового пользователя и указывают путь к файлу с ключом для этого пользователя
15.3
Проверяют выбрана ли функция № 2. Если выбрана, то отображают переписку с конкретным пользователем и отправляют ему текстовое сообщение
15.4
Проверяют выбрана ли функция № 3. Если выбрана, то отображают переписку с конкретным пользователем и отправляют ему текстовый файл
15.5
Проверяют выбрана ли функция № 4. Если выбрана, то обновляют ключ для конкретного пользователя. Для обновления необходимо ввести имя пользователя и путь к файлу, содержащему новый ключ. Смещение для ключа будет помещено в нулевое значение
15.6
Проверяют выбрана ли функция № 5. Если выбрана, то отправляют файл в распределенное хранилище данных
15.7
Проверяют выбрана ли функция № 6. Если выбрана, то запрашивают файл из распределенного хранилища данных
15.8
Проверяют выбрана ли функция № 7. Если выбрана, то обновляют интерфейс отображения пользовательской информации
15.9
Проверяют выбрана ли функция № 8. Если выбрана, то осуществляют безопасный (рекомендованный) выход из системы
16

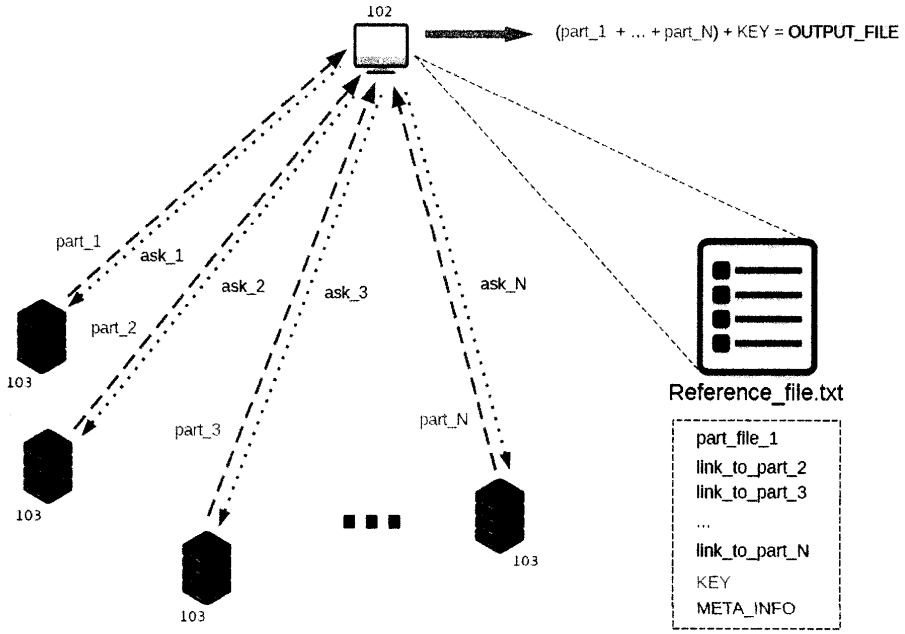
Фиг. 7

8


15.2
15.3
15.3.1
Отображают из базы данных предыдущую переписку с данным пользователем
15.3.2
Вводят новое сообщение для данного пользователя
15.3.3
Добавляют к сообщению поле времени отправки сообщения, поле имени отправителя, поле имени получателя. Выполняют шифрование этих полей на ключе данного пользователя
15.3.4
Извлекают из базы данных ID последней транзакции и извлеченное значение увеличивают на единицу. Формируют новый ID для транзакции
15.3.5
Извлекают из базы данных хэш последней транзакции локальной копии блокчейна отправителя
15.3.6
Создают транзакцию для передачи, в которую входит новый ID, хэш последней транзакции локальной копии блокчейна отправителя и зашифрованные данные
15.3.7
Отправляют всем доступным IP-адресам (находящимся в рассматриваемой нами системе) сформированную транзакцию
15.3.8
Принимают от всех доступных IP-адресов (находящихся в рассматриваемой нами системе) ответ о подтверждении транзакции
15.3.9
Проверяют по мажоритарному правилу факт подтверждения транзакции
Большинство отклонило транзакцию
Большинство подтвердило транзакцию
15.3.10
Добавляют транзакцию в локальную копию блокчейна
15.3.11
Запрашивают актуальную копию блокчейна (рассмотрено в блоке 13)
15.3.12
Добавляют в незашифрованном виде поле времени отправки, поле имени отправителя, поле имени получателя и передаваемое сообщение в базу данных
15.3.13
Выполняют смещение ключа для данного пользователя на длину зашифрованных данных
15.4

Фиг. 8

9

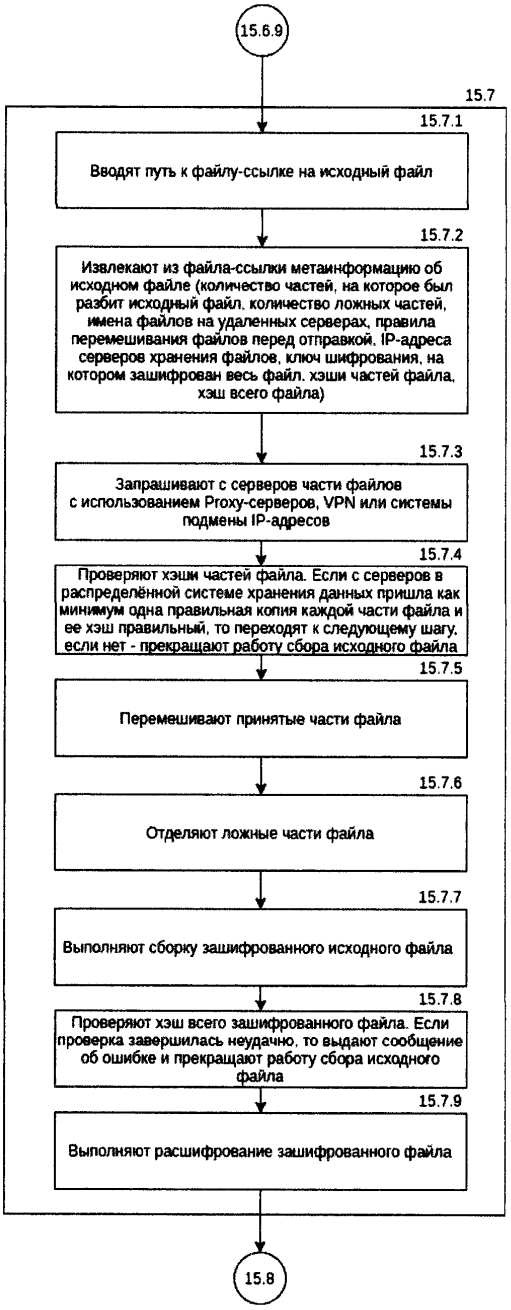

102
(part_1 + ... + part_N) + KEY = OUTPUT_FILE
part_1
ask_1
ask_2
ask_3
ask_N
part_2
part_3
part_N
103
103
103
103
Reference_file.txt
part_file_1
link_to_part_2
link_to_part_3
...
link_to_part_N
KEY
META_INFO

Фиг. 9


15.5
15.6
15.6.1
Вводят путь к исходному файлу
15.6.2
Зашифровывают исходный файл. Помещают ключ шифрования в файл-ссылку
15.6.3
Вычисляют хэш зашифрованного файла. Помещают хэш в файл-ссылку
15.6.4
Разбивают зашифрованный файл на N частей (файлов) случайного размера. Помещают всю необходимую метаинформацию и одну из N частей случайного размера (которая не отправляется в сеть) в файл-ссылку
15.6.5
Добавляют K ложных частей. Помещают всю необходимую метаинформацию в файл-ссылку
15.6.6
Назначают каждому из (N-1+K) файлов R серверов в распределенной системе хранения данных и вычисляют хэш каждого из этих файлов. Помещают всю необходимую метаинформацию в файл-ссылку
15.6.7
Перемешивают (N-1+K) файлов. Помещают всю необходимую метаинформацию в файл-ссылку
15.6.8
Отправляют каждый из (N-1+K) файлов на R серверов в распределенной системе хранения данных через Proxy-сервера, VPN или системы подмены IP-адресов
15.6.9
Формируют окончательный вариант файла-ссылки
15.7.1

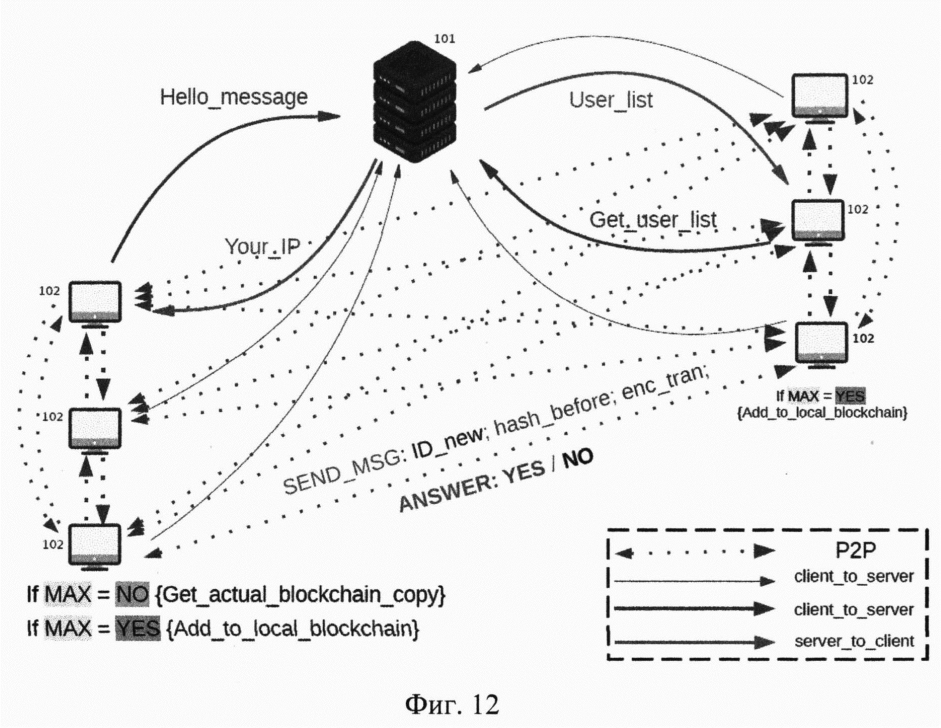
Фиг. 10

11


15.6.9
15.7
15.7.1
Вводят путь к файлу-ссылке на исходный файл
15.7.2
Извлекают из файла-ссылки метаинформацию об исходном файле (количество частей, на которое был разбит исходный файл, количество ложных частей, имена файлов на удаленных серверах, правила перемешивания файлов перед отправкой, IP-адреса серверов хранения файлов, ключ шифрования, на котором зашифрован весь файл, хэши частей файла, хэш всего файла)
15.7.3
Запрашивают с серверов части файлов с использованием Proxy-серверов, VPN или системы подмены IP-адресов
15.7.4
Проверяют хэши частей файла. Если с серверов в распределённой системе хранения данных пришла как минимум одна правильная копия каждой части файла и ее хэш правильный, то переходят к следующему шагу, если нет - прекращают работу сбора исходного файла
15.7.5
Перемешивают принятые части файла
15.7.6
Отделяют ложные части файла
15.7.7
Выполняют сборку зашифрованного исходного файла
15.7.8
Проверяют хэш всего зашифрованного файла. Если проверка завершилась неудачно, то выдают сообщение об ошибке и прекращают работу сбора исходного файла
15.7.9
Выполняют расшифрование зашифрованного файла
15.8

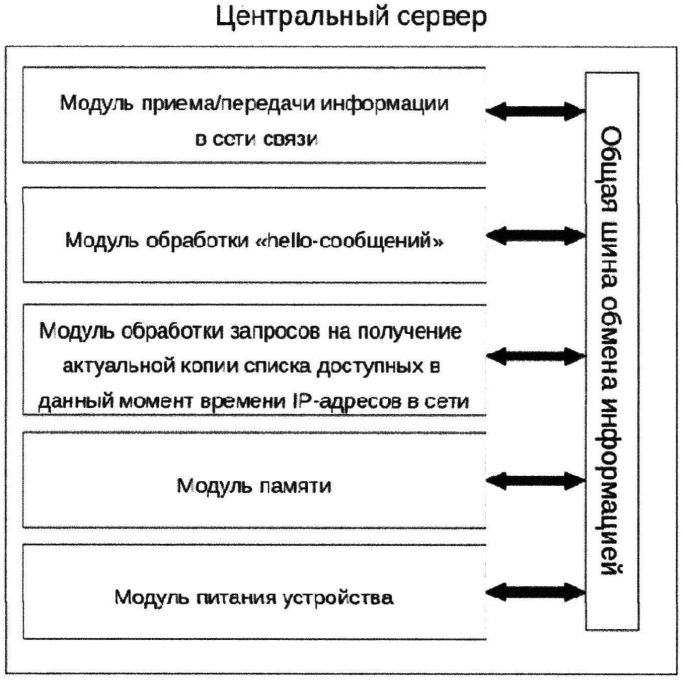
Фиг. 11

12


101
Hello_message
User_list
102
Get_user_list
Your_IP
SEND_MSG: ID_new; hash_before; enc_tran;
ANSWER: YES / NO
If MAX = YES
{Add_to_local_blockchain}
If MAX = NO {Get_actual_blockchain_copy}
If MAX = YES {Add_to_local_blockchain}
P2P
client_to_server
client_to_server
server_to_client

Фиг. 12

13


Центральный сервер
Модуль приема/передачи информации в сети связи
Модуль обработки «hello-сообщений»
Модуль обработки запросов на получение актуальной копии списка доступных в данный момент времени IP-адресов в сети
Модуль памяти
Модуль питания устройства
Общая шина обмена информацией

Фиг. 13


Устройство-участник процесса
блокчейн-коммуникации
Модуль приема/передачи информации в сети связи
Модуль отправки запросов на серверы
Модуль клиента для запросов у
центрального сервера актуального
списка IP-адресов в сети и отправки «hello-сообщения»
Модуль сервера обработки запросов на получение хэша
последней транзакции и актуальной копии блокчейна
Модуль сервера обработки транзакций
для участия в процессе консенсуса
Модуль базы данных
Модуль памяти
Модуль ввода/вывода информации
Модуль питания устройства
Общая шина обмена информацией

Фиг. 14