(19) Евразийское патентное ведомство

(11) 044169 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2023.07.27

(21) Номер заявки
201891081

(22) Дата подачи заявки
2018.05.31

(51) Int. Cl. ***G06F 21/62*** (2013.01)
G06F 12/14 (2006.01)
H04L 9/08 (2006.01)

(54) СПОСОБ И СИСТЕМА ЗАЩИЩЕННОГО ХРАНЕНИЯ ИНФОРМАЦИИ В ФАЙЛОВЫХ ХРАНИЛИЩАХ ДАННЫХ

(31) **2018120197**
(32) **2018.05.31**
(33) **RU**
(43) **2019.12.30**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Карлов Андрей Владимирович,
Фролов Михаил Леонидович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) US-B1-6947556
US-A1-20150113279
US-A1-20090144557
US-B1-7010689
US-A1-20080022134

(57) Данное изобретение в общем относится к области вычислительной техники, а в частности к способам и системам безопасной передачи критичных данных в общедоступную среду. Предложен способ защищенного хранения информации в файловых хранилищах данных, в котором получают по меньшей мере один открытый ключ по меньшей мере одного пользователя; получают по меньшей мере один файл по меньшей мере на одном клиенте; формируют по меньшей мере один ключ для шифрования по меньшей мере одного файла, который сохраняется в файловое хранилище данных; осуществляют шифрование по меньшей мере одного файла полученным ранее по меньшей мере одним ключом для шифрования файла; осуществляют шифрование по меньшей мере одного ключа, которым зашифрован по меньшей мере один файл, открытыми ключами тех пользователей, кому предоставляется доступ по меньшей мере к одному файлу; сохраняют по меньшей мере один зашифрованный на предыдущем шаге ключ в базе ключей шифрования, а по меньшей мере один зашифрованный файл в файловом хранилище данных. Технический результат - повышение безопасности защищенного хранения информации в файловых хранилищах данных.

044169 B1

044169 B1

Область техники

Данное техническое решение, в общем, относится к области вычислительной техники, а в частности к способам и системам безопасной передачи критичных данных в общедоступную среду.

Уровень техники

В настоящее время защита данных является одной из основных проблем, которую необходимо решать при разработке системы корпоративного хранения данных, для малого бизнеса или крупной корпорации. Зачастую данные могут передаваться на различные носители данных, например, таких как облачные хранилища данных или локальные. При передаче данных проблема безопасности заключается в том, что кто-то может получить доступ виртуально или физически к хранилищу и затем обратиться к данным. Злоумышленник может "взломать" систему, смонтировать носитель данных, а затем получить доступ к данным. Известные из уровня техники некоторые технические решения затрагивают эти проблемы, шифруя все или большинство данных на носителях данных, однако эти подходы страдают от ряда недостатков с точки зрения слабой стороны безопасности, проблем с реализацией и/или сложности. Например, в простых технических решениях, в которых хранят зашифрованные данные на носителе вместе с ключом данных, используемым для шифрования данных, любой, имеющий физический доступ к носителю, может извлечь ключ данных из носителя и использовать его для дешифрования данных. Кроме того, предшествующие решения обычно обеспечивают доступ к зашифрованным данным для всех, у кого есть ключ данных шифрования, но не позволяют различным сторонам отдельно получать доступ к зашифрованным данным с помощью своих собственных ключей доступа. Принимая во внимание вышесказанное, существует потребность в усовершенствованных способах защиты данных в системах корпоративного хранения данных. Критичные данные должны быть доступны только уполномоченным лицам, только тем способом, который разрешен политикой безопасности, и только с помощью средств, определенных политикой безопасности.

Сущность изобретения

Техническое решение направлено на устранение недостатков, присущих существующим решениям из известного уровня техники.

Технической задачей, поставленной в данном техническом решении, является обеспечение защищенного хранения информации в файловых хранилищах данных.

Техническим результатом, проявляющимся при решении вышеуказанной задачи, является повышение безопасности защищенного хранения информации в файловых хранилищах данных.

Указанный технический результат достигается благодаря осуществлению способа защищенного хранения информации в файловых хранилищах данных, в котором получают по меньшей мере один открытый ключ по меньшей мере одного пользователя; получают по меньшей мере один файл по меньшей мере на одном клиенте; формируют по меньшей мере один ключ для шифрования по меньшей мере одного файла, который сохраняется в файловое хранилище данных; осуществляют шифрование по меньшей мере одного файла полученным ранее по меньшей мере одним ключом для шифрования файла; осуществляют шифрование по меньшей мере одного ключа, которым зашифрован по меньшей мере один файл, открытыми ключами тех пользователей, кому предоставляется доступ по меньшей мере к одному файлу; сохраняют по меньшей мере один зашифрованный на предыдущем шаге ключ в базе ключей шифрования, а по меньшей мере один зашифрованный файл в файловом хранилище данных.

В некоторых вариантах осуществления файловое хранилище является облачным и/или локальным.

В некоторых вариантах осуществления файловое хранилище является публичным и/или частным, и/или гибридным.

В некоторых вариантах осуществления для предоставления доступа к зашифрованному файлу внутри файлового хранилища создается персонифицированная или публичная ссылка для файла, которая сохраняется в реестре ссылок на файлы.

В некоторых вариантах осуществления при использовании нескольких файловых хранилищ файлы и структура папок из всех файловых хранилищ отображаются в едином интерфейсе как едином файловом хранилище.

В некоторых вариантах осуществления при использовании нескольких файловых хранилищ в едином интерфейсе отображаются файлы из файловых хранилищ других пользователей на основе реестра ссылок на файлы.

Краткое описание чертежей

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей.

На фиг. 1 приведена блок-схема способа защищенного хранения информации в файловых хранилищах данных.

На фиг. 2 показан пример осуществления системы защищенного хранения информации в файловых хранилищах данных.

На фиг. 3 показан пример осуществления шифрования ключа, которым зашифрован файл, открытыми ключами тех пользователей, кому предоставляется доступ по меньшей мере к одному файлу.

На фиг. 4 показан пример осуществления регистрации пользователя в системе защищенного хране-

ния информации в файловых хранилищах данных.

На фиг. 5 показан пример осуществления загрузки файла пользователя в системе защищенного хранения информации в файловых хранилищах данных.

Подробное описание изобретения

Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

В данном техническом решении под системой подразумевается, в том числе компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, четко определенную последовательность операций (действий, инструкций).

Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройств хранения данных. В роли устройства хранения данных могут выступать, но не ограничиваясь, жесткие диски (HDD), флеш-память, ПЗУ (постоянное запоминающее устройство), твердотельные накопители (SSD), оптические приводы.

Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

Ассиметричные ключи - ключи, используемые в ассиметричных алгоритмах (шифрование, ЭЦП), которые являются ключевой парой.

Открытый ключ (англ. public key) - ключ, который может быть опубликован и используется для проверки подлинности подписанного документа, а также для предупреждения мошенничества со стороны заверяющего лица в виде отказа его от подписи документа. Открытый ключ подписи вычисляется как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.

Закрытый ключ (англ. public key) - ключ, известный только своему владельцу. Только сохранение пользователем в тайне своего закрытого ключа гарантирует невозможность подделки злоумышленникам документа и цифровой подписи от имени заверяющего.

Шифрование - обратимое преобразование информации в целях скрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней. Главным образом шифрование служит задачей соблюдения конфиденциальности передаваемой информации.

Токен авторизации - программный токен, который выдается пользователю после его успешной авторизации и является ключом для доступа к ресурсу (например, для доступа к облачному хранилищу).

Цифровой сертификат - выпущенный удостоверяющим центром электронный или печатный документ, подтверждающий принадлежность владельцу открытого ключа или каких-либо атрибутов.

Advanced Encryption Standard (AES) - симметричный алгоритм блочного шифрования, принятый правительством США на основе результатов проведенного конкурса в качестве стандарта шифрования и заменивший собой менее надежный алгоритм Data Encryption Standard (DES). Утвержденный алгоритм в качестве единого стандарта шифрования стал повсеместно применяться для защиты электронных данных.

Вектор инициализации - вектор, который представляет собой произвольное число, которое может быть использовано вместе с секретным ключом для шифрования данных.

Cipher block chaining (CBC) - режим сцепления блоков шифротекста -один из режимов шифрования для симметричного блочного шифра с использованием механизма обратной связи. Каждый блок открытого текста (кроме первого) побитово складывается по модулю 2 с предыдущим результатом. Одна ошибка в бите блока шифротекста влияет на расшифровку всех последующих блоков. Перестройка порядка блоков зашифрованного текста вызывает повреждения результата дешифрования.

В некоторых вариантах осуществления способ, показанный на фиг. 1, включает следующие шаги.

Предварительно осуществляют генерацию по меньшей мере одного цифрового сертификата безопасности для пользователя 200, как показано на фиг. 2. Пользователь 200 задает пароль для цифрового сертификата. Пароль может быть символьным, числовым или комбинированным. В некоторых вариантах осуществления в качестве пароля могут использоваться биометрические данные пользователя 200. Система 202, включающая удостоверяющий центр для выпуска цифрового сертификата, например, посредством сервера, генерирует цифровой сертификат и отображает на странице веб-интерфейса 201 пользователя QR-код (который содержит в себе ссылку на скачивание сгенерированного цифрового сертификата) для сканирования, например, мобильным устройством 230 связи пользователя или другим сканирующим устройством. В качестве цифрового сертификата может использоваться сертификат открытого ключа, имеющий формат X.509 v3, который описан в RFC 5280

Затем пользователь 200 сканирует посредством мобильного приложения, установленного на мобильном устройстве 203 связи QR-код, как показано на фиг. 4, который был ранее сгенерирован, получает URL-ссылку на цифровой сертификат безопасности, скачивает его, запрашивает ранее заданный пользователем пароль на сертификат и устанавливает его.

После установки цифрового сертификата пользователя, мобильное приложение, установленное на мобильном устройстве 203 связи, передает в систему 202, например следующие данные из цифрового сертификата: ФИО, e-mail. В некоторых вариантах осуществления может использоваться SSL сертификат.

В цифровом сертификате может храниться следующая информация: полное (уникальное) имя владельца сертификата; открытый ключ владельца; дата выдачи сертификата; дата окончания сертификата; полное (уникальное) имя удостоверяющего центра сертификации; цифровая подпись издателя.

Система 202 получает эти данные, создает пользователя у себя в хранилище данных, генерирует для него уникальный идентификатор (User ID), например, на основании случайных чисел, либо посредством использования хэш-функции от его данных, и добавляет запись о новом пользователе в список пользователей, который может храниться в хранилище 204 данных. Запись о пользователе может содержать следующие данные: уникальный идентификатор User ID, ФИО, e-mail, номер мобильного телефона и т.д., не ограничиваясь.

Затем пользователь 200 открывает мобильное приложение на мобильном устройстве 203 связи:

если это первый вход пользователя 200, то пользователь 200 задает пароль и/или вводит образец своего отпечатка пальца;

если пользователь 200 осуществляет вход в приложение уже не первый раз, то пользователь 200 вводит пароль и/или прикладывает ранее введенный отпечаток пальца.

В мобильном приложении на мобильном устройстве 203 связи пользователя отображается форма для ввода пароля/отпечатка пальца. Пользователь 200 вводит пароль/отпечаток пальца. Мобильное приложение, установленное на мобильном устройстве 203 связи, проверяет корректность введенного пароля/отпечатка:

если проверка прошла успешно, то осуществляют запрос на проверку цифрового сертификата в систему 202;

если проверка не прошла успешно, процесс заканчивается с ошибкой, и пользователю направляется сообщение (в мессенджере, посредством смс и/или PUSH) например, следующего содержания "Неправильно введенный пароль" или "Отпечаток пальца не подходит".

Мобильное приложение, установленное на мобильном устройстве 203 связи, отправляет запрос на проверку сертификата пользователя в систему 202, которая проверяет сертификат пользователя:

если сертификат прошел проверку, то процесс заканчивается (пользователь прошел аутентификацию);

если сертификат не прошел проверку, то процесс завершается с ошибкой. На экране мобильного приложения пользователя 200 может появиться сообщение "Сертификат не прошел проверку".

Мобильное приложение автоматически генерирует пару ключей для каждого пользователя 200 системы 202, которые сохраняются в мобильном приложении в его хранилище данных:

открытый ключ (публичный ключ);

закрытый ключ (секретный ключ).

В качестве алгоритма для генерации пары ключей могут использоваться общеизвестные из уровня техники алгоритмы генерации ключей в ассиметричном шифровании, например, такие как RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm), Elgamal (шифросистема Эль-Гамаля), Diffie-Hellman (обмен ключами Диффи-Хелмана), не ограничиваясь.

Шаг 101: получают по меньшей мере один открытый ключ (PubllserKey) по меньшей мере одного пользователя.

Мобильное приложение, установленное на мобильном устройстве 203 связи пользователя, передает в систему 202 копию открытого ключа (PubllserKey) пользователя по телекоммуникационному каналу для шэринга (от англ. "to share" - делиться), или другими словами обмена с другими пользователями, а так же передает User ID данного пользователя 200.

Дополнительно на данном шаге получают сгенерированный ранее закрытый ключ пользователя (UserKey).

В одном из вариантов осуществления телекоммуникационный канал для обмена данными мобильного устройства связи и автоматизированной системы использует телекоммуникационный протокол, выбранный из группы, состоящей из следующих протоколов передачи данных SS/7 (Signaling System #7, см., например, ITU-T Recommendation Q.700) и ISDN (Integrated Services Digital Network), WiFi (см., например, IEEE Standard 802.11).

В одном из вариантов осуществления телекоммуникационный канал, функционирует согласно стандарту, выбранному из группы, состоящей из Ethernet, WiFi (см., например, IEEE Standard 802.11) и Bluetooth (см., например, IEEE Standard 802.15), ATM (Asynchronous Transfer Mode), SS/7 (Signaling System #7; см., например, ITU-T Recommendation Q.700), X.25 (см., например, ITU-T Recommendation X.25), WiMAX (Worldwide Interoperability for Microwave Access, см., например, IEEE Standards 802.16-802.16e), SCCP (Signalling Connection Control Part), DUP (Data User Part), B-ISUP (B-ISDN User Part), ISUP (ISDN User Part), TUP (Telephone User Part), TCAP (Transaction Capabilities Application Part), SSCOP (Service-Specific Connection Oriented Protocol), H.323, SIP (Session Initial Protocol), BICC (Bearer Independent Call Control protocol), IS-41, IS-634, CAS, CS1, CS2, R2, CAMEL (Customized Applications for Mobile network Enhanced Logic), INAP (Intelligent Network Application Part), MAP (Mobile Application Part). В одном из

вариантов осуществления телекоммуникационный канал использует стек протоколов TCP/IP (Transmission Control Protocol/Internet Protocol, см. IETF RFC 1122).

Для передачи в систему 202 медиафайлов могут использовать транспортный протокол реального времени (Real-time Transport Protocol, RTP), который широко применяют для передачи синхронизированных мультимедийных данных в реальном времени, например, для передачи аудио- и видеоданных. Протокол RTP может работать на базе протокола пользовательских датаграмм (User Datagram Protocol, UDP), который, в свою очередь, может функционировать на основе протокола Интернета (Internet Protocol, IP). В транспортном протоколе RTP мультимедийные данные инкапсулируют в RTP-пакеты. Как правило, каждый тип мультимедийных данных или формат кодирования мультимедийных данных имеет отдельный формат полезной нагрузки RTP.

Сеанс протокола RTP представляет собой соединение между группой участников, осуществляющих связь при помощи протокола RTP. Он является каналом групповой связи, по которому, потенциально, может передаваться несколько RTP-потоков. RTP-поток представляет собой поток RTP-пакетов, содержащих мультимедийные данные.

Система 202 получает из мобильного приложения, установленного на мобильном устройстве 203 связи пользователя 200, копию открытого ключа (PubUserKey) и User ID пользователя. Затем система 202 ищет в списке пользователей пользователя в хранилище данных по User ID (уникальный идентификатор пользователя) и добавляет в запись о пользователе данные о его открытом ключе (PubUserKey) в хранилище данных.

Затем пользователь 200 формирует парольную фразу для доступа к бекапу ключей и запоминает ее в хранилище данных на своем мобильном устройстве связи. Мобильное приложение передает копию пары ключей в файловое хранилище 204 данных, которое было выбрано для хранения информации. В некоторых вариантах осуществления файловое хранилище 204 данных может быть облачным (на платформе Dropbox, Яндекс Диск, Google Drive и т. п.) или удаленным (локальным).

Шаг 102: получают по меньшей мере один файл по меньшей мере на одном клиенте.

Пользователь 200 выбирает один или несколько файлов, которые необходимо загрузить в хранилище данных 204, как показано на фиг. 5, и подтверждает их загрузку в файловое хранилище 204 данных посредством использования мобильного приложения, установленного на мобильном устройстве связи пользователя. В некоторых вариантах осуществления мобильное приложение осуществляет проверку названий загружаемых файлов, с названиями файлов, которые уже находятся в списке хранилища для выявления возможных коллизий.

Если в хранилище пользователя уже существует файл с идентичным названием, то на экран мобильного приложения выводится диалоговое окно, например, следующего вида "Файл с идентичным названием уже есть в списке". Пользователю предлагается либо заменить файл, либо выполнить переименование загружаемого файла для отсутствия коллизий. В некоторых вариантах осуществления изобретения новый загружаемый файл, который имеет идентичное название файлу, который находится в списке, переименовывается автоматически с добавлением, например, версии в конце файла: "документ на согласование (1)", причем каждый последующий файл с таким названием будет получать увеличенный индекс.

В контексте данного изобретения использование термина "клиент" подразумевает архитектуру "клиент - сервер". Это широко известная в уровне технике вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер - это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине. Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP, BitTorrent, потоковое мультимедиа или работа с базами данных) или в виде сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями или просмотр web-страниц во всемирной паутине). Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, ее размещают на специально выделенной вычислительной машине, настроенной особым образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики ее оборудования и программного обеспечения, ее также называют сервером, а машины, выполняющие клиентские программы, соответственно, клиентами. В данном техническом решении могут получать для загрузки в хранилище данных, например, медиафайл. Причем в качестве стандартов медиафайлов могут использоваться все существующие из уровня техники без ограничения, которые включают базовый формат медиафайла стандарта ISO (ISO/IEC 14496-12, который имеет аббревиатуру ISOBMFF), формат файлов MPEG-4 (ISO/IEC 14496-14, также называемый форматом MP4), формат файлов для видео со структурой из NAL-блоков (ISO/IEC 14496-15) и формат файлов 3 GPP (3 GPP TS 26.244, также известный под названием формата 3GP). Формат файлов ISO является базовым для получения всех упомянутых выше форматов файла (за исключением собственно формата файлов ISO). Эти форматы файлов (включая соб-

ственно формат файлов ISO) называют семейством форматов файлов ISO.

Шаг 103: формируют по меньшей мере один ключ (FileKey) для шифрования по меньшей мере одного файла, который сохраняется в файловое хранилище данных.

На данном шаге мобильное приложение, установленное на мобильном устройстве 203 связи пользователя 200, генерирует уникальный ключ (FileKey) для шифрования по меньшей мере одного файла, которым будет зашифрован файл.

Для шифрования файлов, которые пользователь планирует загрузить в файловое хранилище данных, используется например алгоритм AES/CBC/PKCS5Padding или AES/CBC/PKCS7Padding. Длина такого файлового ключа (FileKey) составляет 128 бит (цифры со значениями от 0 или 1). В некоторых вариантах реализации изобретения может использоваться ключ размером 192 или 256 бит.

В конкретном варианте осуществления используется симметричный алгоритм для шифрования файла. Для каждого файла генерируется свой уникальный файловый ключ (FileKey). За счет использования ключа в 128 бит (AES алгоритм) возможное количество комбинаций по подбору уникального ключа злоумышленниками составляет 3.4*10^38, с которым не справится даже самый быстрый суперкомпьютер на настоящий момент, что повышает надежность данного технического решения.

Для того чтобы создать ключ (FileKey) для каждого раунда, алгоритм AES использует процесс ключевого расширения. От размера ключа зависит число раундов шифрования файла: длина 128 бит - 10 раундов; длина 192 бита - 12 раундов; длина 256 бит - 14 раундов.

Файловый ключ (FileKey) состоит из 128 битов, поделенных на 16 байтов, и записывается в столбцы матрицы. Каждый столбец матрицы образует слово, т.е. фактически ключ для шифрования - это четыре слова. Из этих слов с помощью специального алгоритма образуется последовательность из 44 слов (каждое слово по 32 бита). На каждый раунд шифрования подаются по четыре слова этой последовательности.

Так называемые раундовые ключи вырабатываются из сформированного файлового ключа с помощью процедуры расширения ключа, в результате чего формируется массив раундовых ключей, из которого затем непосредственно выбирается необходимый раундовый ключ. Каждый раундовый ключ имеет длину 128 бит (или 4 четырехбайтовых слова), а длина в битах всех раундовых ключей равна 128 бит (10 раундов+1) = 1408 бит (или 44 четырехбайтовых слова). Первые четыре слова в ключевом массиве заполнены ключом шифра, из остальных выработанных 40 слов выбираются по 4 слова для ключа раунда. Выбор слов используется известный из уровня техники: первые четыре слова (они совпадают с ключом шифра) являются ключом с номером 0, следующие четыре слова -раундовым ключом для первого полного раунда и т.д.

В качестве примера осуществления файловый ключ для шифрования одного файла может иметь следующий вид: OF 15 71 C9 47 D9 E8 59 OC B7 AD DF AF 7F 67 98.

Шаг 104: осуществляют шифрование по меньшей мере одного файла сформированным ранее по меньшей мере одним ключом для шифрования файла.

Мобильное приложение на мобильном устройстве связи пользователя шифрует файл, выбранный пользователем для загрузки, ключом для шифрования файла. Таким образом, в случае использования 128-битового ключа, осуществляют 10 раундов шифрования (процедур трансформации данных, как это называется в AES). Перед первым раундом шифрования выполняется операция суммирования по модулю 2 с начальным файловым ключом.

Преобразования, выполненные в одном раунде, обозначают как Round (State, RoundKey), где переменная State - матрица, описывающая данные на входе раунда и на его выходе после шифрования; переменная RoundKey - матрица, содержащая раундовый ключ. Раунд состоит из 4 различных преобразований:

побайтовая подстановка в S-боксе с фиксированной таблицей замен;

побайтовый сдвиг строк матрицы State на различное количество байт;

перемешивание байт в столбцах;

сложение с раундовым ключом (операция XOR).

Последний раунд несколько отличается от предыдущих тем, что не задействует функцию перемешивания байт в столбцах.

Для AES длина блока входных данных и состояния постоянна и равна 128 бит, а длина ключа для шифрования составляет 128, 192, или 256 бит.

Если в файловом хранилище 204 данных пользователя отсутствует файл с идентичным названием, то выполняют шифрование файла. После завершения шифрования файла, он загружается в файловое хранилище данных в зашифрованном виде.

В некоторых вариантах осуществления изобретения в случае смены цифрового сертификата, система 202 выполняют перешифрование всех ключей файлов новым сертификатом.

После шифрования документа файловый ключ (FileKey) удаляется из мобильного приложения пользователя.

Шаг 105: осуществляют шифрование по меньшей мере одного ключа (FileKey), которым зашифрован по меньшей мере один файл, открытыми ключами (PubllserKey) тех пользователей, кому предоставляется доступ по меньшей мере к одному файлу.

Для шифрования файловых ключей (FileKey) может использоваться такой алгоритм как

RSA/ECB/PKCS1 Padding. Длина такого ключа может составлять 2048 бит. Это ассиметричный алгоритм: есть закрытый и открытый ключи. Для каждого пользователя 200 генерируется ключевая пара, закрытый ключ остается у пользователя. Открытый ключ передается на сервер и доступен любому потенциальному пользователю системы 202. С помощью него зашифровывается файловый ключ (FileKey), когда происходит шеринг файла.

Таким образом, обеспечивается безопасная передача данных в систему 202 на сервер и хранение файловых ключей (FileKey) в зашифрованном виде, т.к. в случае, если злоумышленник получит доступ к системе 202, он не сможет расшифровать файл, потому что файловый ключ (FileKey) хранится в зашифрованном виде.

Для шифрования файлового (FileKey) используется операция возведения в степень по модулю N. Для расшифрования же необходимо вычислить функцию Эйлера от числа N, для этого необходимо знать разложение числа n на простые множители.

Генерация ключевой пары осуществляется следующим образом.

1. Выбирают два простых числа p и q случайным образом (такие, что p не равно q).

2. Определяют модуль выражения N=p*q.

3. Определяют значение функции Эйлера от модуля N: $\phi(N)=(p-1)(q-1)$.

4. Выбирают число e, называемое открытой экспонентой, причем число e должно лежать в интервале $1 <e<\phi(N)$, а так же быть взаимно простым со значением функции $\phi(N)$. В качестве открытой экспоненты могут использоваться простые числа Ферма, например: 17 или 257, или 65537 и т.д.

5. Определяют число d, называемое секретной экспонентой, такое, что $d*e=1(\mathrm{mod}\phi(N))$, то есть является мультипликативно обратное к числу e по модулю $\phi(N)$. Итак, мы получаем пару ключей:

Пара (e, N) - открытый ключ. Пара (d, N) - закрытый ключ.

В некоторых вариантах осуществления технического решения для формирования пары ключей используют OpenSSL - криптографический пакет с открытым исходным кодом для работы с SSL/TLS.

Затем после формирования пары ключей, как показано на фиг 3, для шифрования по меньшей мере одного файлового ключа m, которым зашифрован по меньшей мере один файл, используют сгенерированный открытый ключ (e, N): C=E(M)=M^e mod(N).

Шаг 106: сохраняют по меньшей мере один зашифрованный на предыдущем шаге ключ в базе ключей шифрования, а по меньшей мере один зашифрованный файл в файловом хранилище данных.

В некоторых вариантах осуществления зашифрованный на предыдущем шаге ключ в базе ключей шифрования сохраняют в формате hex. HEX - формат файла, предназначенного для представления произвольных двоичных данных в текстовом виде. Например, зашифрованный ключ имеющий значение 6131, будет храниться как строка вида "0x6 0xd 0x1". Мобильное приложение передает в систему данные: зашифрованный ключ шифрования файла (FileKey) и идентификатор пользователя UserID. При этом в мобильном приложении не сохраняется копия ключа шифрования файла. Система 202 на основе полученного ключа шифрования файла генерирует уникальный идентификатор файла File ID. Мобильное приложение получает уникальный идентификатор файла File ID и сохраняет его в списке файлов. Затем мобильное приложение передает зашифрованный файл и File ID, например, в облачное хранилище данных. Облачное хранилище получает зашифрованный файл и сохраняет его.

При получении зашифрованного ключа с осуществляют его расшифровку, используя закрытый ключ (d,N), и расшифровывают ключ следующим образом M=D(C)=C^d mod(N).

При этом файловый ключ (FileKey), зашифрованный открытым ключом пользователя (PubllserKey), расшифровать можно только с помощью закрытого ключа пользователя UserKey (всегда хранится на мобильном устройстве связи пользователя).

Аспекты настоящего изобретения могут быть также реализованы с помощью устройства обработки данных, являющимся вычислительной машины из системы (или таких средств, как центральный/графический процессор или микропроцессор), которая считывает и исполняет программу, записанную на запоминающее приспособление, чтобы выполнять функции вышеописанного варианта(ов) осуществления, и способа, показанного на фиг. 1, этапы которого выполняются вычислительной машиной из системы или устройством путем, например, считывания и исполнения программы, записанной на запоминающем приспособлении, чтобы исполнять функции вышеописанного варианта(ов) осуществления. С этой целью программа предоставляется на вычислительную машину, например, через сеть или со среды для записи различных типов, служащей в качестве запоминающего приспособления (например, машиночитаемой среды).

Устройство обработки данных может иметь дополнительные особенности или функциональные возможности. Например, устройство обработки данных может также включать в себя дополнительные устройства хранения данных (съемные и несъемные), такие как, например, магнитные диски, оптические диски или лента. Компьютерные носители данных могут включать в себя энергозависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или при помощи любой технологии для хранения информации, такой как машиночитаемые инструкции, структуры данных, программные модули или другие данные. Устройство хранения данных, съемное хранилище и несъемное хранилище являются примерами компьютерных носителей данных. Компьютерные носители данных

включают в себя, но не в ограничительном смысле, оперативное запоминающее устройство (ОЗУ), постоянное запоминающее устройство (ПЗУ), электрически стираемое программируемое ПЗУ (EEPROM), флэш-память или память, выполненную по другой технологии, ПЗУ на компакт-диске (CD-ROM), универсальные цифровые диски (DVD) или другие оптические запоминающие устройства, магнитные кассеты, магнитные ленты, хранилища на магнитных дисках или другие магнитные запоминающие устройства, или любую другую среду, которая может быть использована для хранения желаемой информации и к которой может получить доступ устройство обработки данных. Любой такой компьютерный носитель данных может быть частью системы выявления и классификации причин возникновения претензий пользователей в канале банкомата. Устройство обработки данных может также включать в себя устройство(а) ввода, такие как клавиатура, мышь, перо, устройство с речевым вводом, устройство сенсорного ввода, и так далее. Устройство(а) вывода, такие как дисплей, динамики, принтер и тому подобное, также могут быть включены в состав системы. Устройство обработки данных содержит коммуникационные соединения, которые позволяют устройству связываться с другими вычислительными устройствами, например по сети. Сети включают в себя локальные сети и глобальные сети наряду с другими большими масштабируемыми сетями, включая, но не в ограничительном смысле, корпоративные сети и экстрасети. Коммуникационное соединение является примером коммуникационной среды. Как правило, коммуникационная среда может быть реализована при помощи машиночитаемых инструкций, структур данных, программных модулей или других данных в модулированном информационном сигнале, таком как несущая волна, или в другом транспортном механизме, и включает в себя любую среду доставки информации. Термин "модулированный информационный сигнал" означает сигнал, одна или более из его характеристик изменены или установлены таким образом, чтобы закодировать информацию в этом сигнале. Для примера, но без ограничения, коммуникационные среды включают в себя проводные среды, такие как проводная сеть или прямое проводное соединение, и беспроводные среды, такие как акустические, радиочастотные, инфракрасные и другие беспроводные среды. Термин "машиночитаемый носитель", как употребляется в этом документе, включает в себя как носители данных, так и коммуникационные среды. Последовательности процессов, описанных в этом документе, могут выполняться с использованием аппаратных средств, программных средств или их комбинации. Когда процессы выполняются с помощью программных средств, программа, в которой записана последовательность процессов, может быть установлена и может выполняться в памяти компьютера, встроенного в специализированное аппаратное средство, или программа может быть установлена и может выполняться на компьютер общего назначения, который может выполнять различные процессы.

Например, программа может быть заранее записана на носитель записи, такой как жесткий диск, или ПЗУ (постоянное запоминающее устройство). В качестве альтернативы, программа может быть временно или постоянно сохранена (записана) на съемном носителе записи, таком как гибкий диск, CD-ROM (компакт-диск, предназначенный только для воспроизведения), MO (магнитооптический) диск, DVD (цифровой универсальный диск), магнитный диск или полупроводниковая память. Съемный носитель записи может распространяться в виде так называемого, продаваемого через розничную сеть программного средства.

Программа может быть установлена со съемного носителя записи, описанного выше, на компьютер, или может быть передана по кабелю с сайта загрузки в компьютер или может быть передана в компьютер по сетевым каналам передачи данных, таким как ЛВС (локальная вычислительная сеть) или Интернет. Компьютер может принимать переданную, таким образом, программу и может устанавливать ее на носитель записи, такой как встроенный жесткий диск.

Процессы, описанные в этом документе, могут выполняться последовательно по времени, в соответствии с описанием, или могут выполняться параллельно или отдельно, в зависимости от характеристик обработки устройства, выполняющего процессы, или в соответствии с необходимостью. Система, описанная в этом документе, представляет собой логический набор множества устройств и не ограничивается структурой, в которой эти устройства установлены в одном корпусе.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ защищенного хранения информации в файловых хранилищах данных, включающий следующие шаги:

получают по меньшей мере один открытый ключ и по меньшей мере один закрытый ключ по меньшей мере одного пользователя, причем упомянутые открытый ключ и закрытый ключ генерируются по меньшей мере в одном устройстве обработки данных пользователя;

получают по меньшей мере один файл по меньшей мере на одном клиенте;

формируют по меньшей мере один ключ для шифрования по меньшей мере одного файла, который сохраняется в файловое хранилище данных, причем упомянутый ключ для шифрования формируется по меньшей мере в одном устройстве обработки данных пользователя;

осуществляют шифрование по меньшей мере одного файла полученным ранее по меньшей мере одним ключом для шифрования файла;

осуществляют шифрование по меньшей мере одного ключа, которым зашифрован по меньшей мере один файл, по меньшей мере одним открытым ключом тех пользователей, кому предоставляется доступ по меньшей мере к одному файлу;

сохраняют по меньшей мере один зашифрованный на предыдущем шаге ключ в базе ключей шифрования, а по меньшей мере один зашифрованный файл в файловом хранилище данных, причем для упомянутого зашифрованного файла формируется уникальный идентификатор, который передается устройством обработки данных пользователя в файловое хранилище данных.

2. Способ по п.1, характеризующийся тем, что файловое хранилище является облачным и/или локальным.

3. Способ по п.1, характеризующийся тем, что файловое хранилище является публичным, и/или частным, и/или гибридным.

4. Способ по п.1, характеризующийся тем, что для предоставления доступа к зашифрованному файлу внутри файлового хранилища создается персонифицированная или публичная ссылка для файла, которая сохраняется в реестре ссылок на файлы.

5. Способ по п.1, характеризующийся тем, что при использовании нескольких файловых хранилищ файлы и структура папок из всех файловых хранилищ отображаются в едином интерфейсе как едином файловом хранилище.

6. Способ по п.5, характеризующийся тем, что при использовании нескольких файловых хранилищ в едином интерфейсе отображаются файлы из файловых хранилищ других пользователей на основе реестра ссылок на файлы.

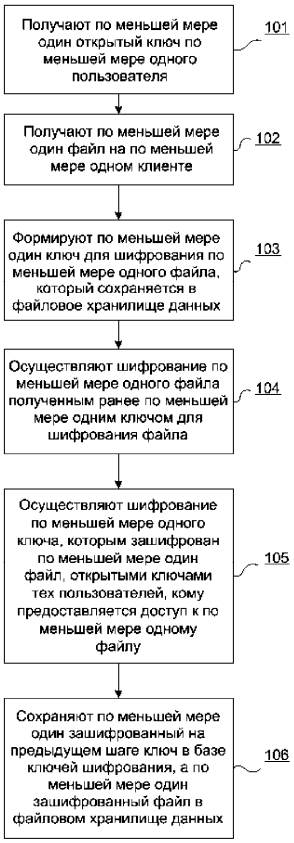
7. Система защищенного хранения информации в файловых хранилищах данных, содержащая

по меньшей мере одно устройство обработки данных;

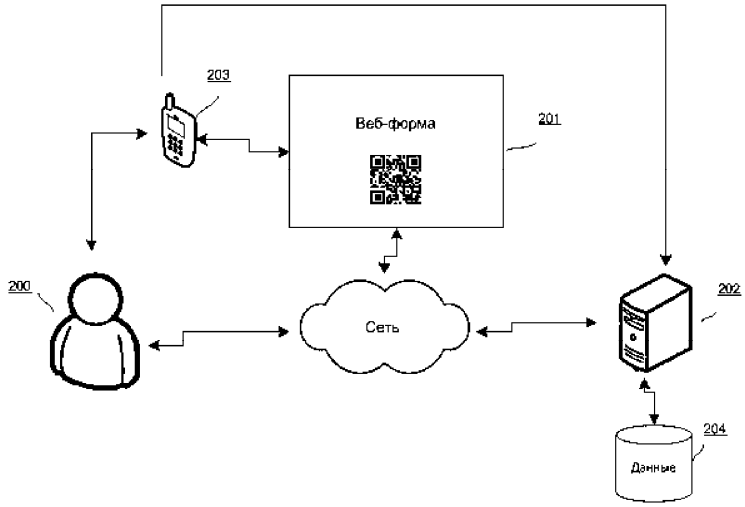
по меньшей мере одно файловое хранилище данных;

по меньшей мере одну программу,

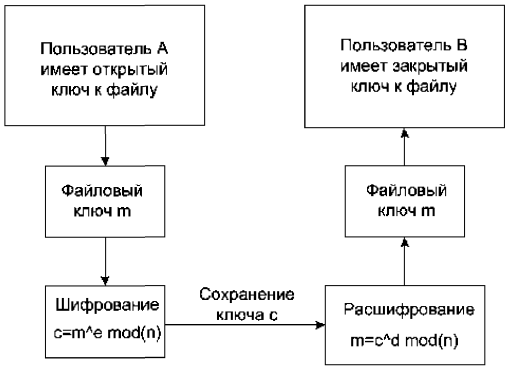
где одна или более программ хранятся на одном или более устройствах хранения данных и исполняются на одном и более устройствах обработки данных, причем одна или более программ включает инструкции для выполнения способа по п.1.



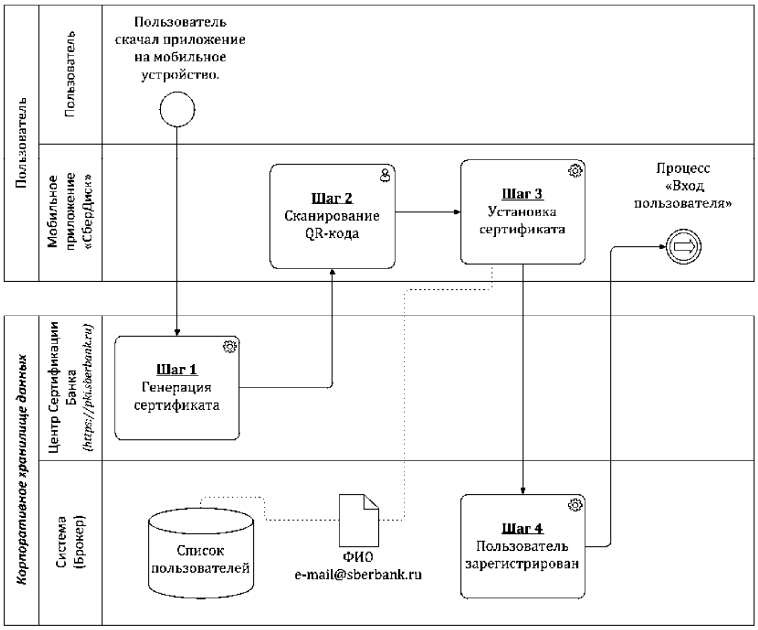
Фиг. 1



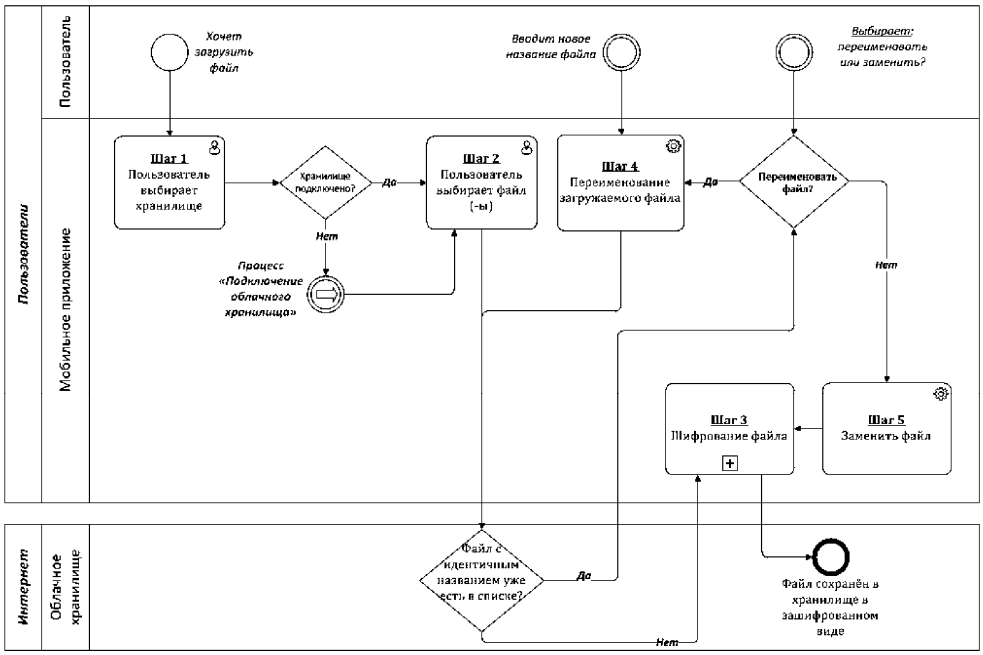
Фиг. 2



Фиг. 3



Фиг. 4


Пользователи
Пользователь
Мобильное приложение
Интернет
Облачное хранилище
Хочет загрузить файл
Вводит новое название файла
Выбирает: переименовать или заменить?
Шаг 1
Пользователь выбирает хранилище
Хранилище подключено?
Да
Нет
Шаг 2
Пользователь выбирает файл (-ы)
Шаг 4
Переименование загружаемого файла
Да
Переименовать файл?
Нет
Процесс «Подключение облачного хранилища»
Шаг 3
Шифрование файла
Шаг 5
Заменить файл
Файл с идентичным названием уже есть в списке?
Да
Нет
Файл сохранён в хранилище в зашифрованном виде

Фиг. 5