ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Բերբերյան Լևոն Սամվելի

ՏՎՅԱԼՆԵՐԻ ՀԵՌԱԿԱ ԿԱՌԱՎԱՐՄԱՆ ԲԱԶՄԱՆԴԱՄ ՀԻԵՐԱՐԽԻԿ ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄ ԵՎ ԾՐԱԳՐԱՅԻՆ ԻՐԱԿԱՆԱՑՈՒՄ

Ե.13.04 – «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսության

ՍԵՂՄԱԳԻՐ

Երևան – 2014

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Берберян Левон Самвелович

РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИЕРАРХИЧЕСКОЙ СИСТЕМЫ УДАЛЕННОГО УПРАВЛЕНИЯ ДАННЫМИ

АВТОРЕФЕРАТ
Диссертации на соискание ученой степени кандидата технических наук по специальности 05.13.04– «Математическое и программное обеспечение математических машин, комплексов, систем и сетей»

Ереван - 2014

Ատենախոսության թեման հաստատվել է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում

Գիտական ղեկավար՝ տեխ.գիտ.դոկտոր Վ.Ի.Թաիրյան
Պաշտոնական ընդդիմախոսներ՝ տեխ.գիտ.դոկտոր Հ.Հ.Հարությունյան
տեխ.գիտ.թեկնածու Ս.Ե. Աբրահամյան

Առաջատար կազմակերպություն՝ Հայաստանի պետական ճարտարագիտական համալսարան

Պաշտպանությունը կայանալու է 2014թ. հունիսի 6-ին, ժ. 16:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա և հաշվողական համակարգեր» մասնագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:
Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:
Սեղմագիրը առաքված է 2014թ. մայիսի 6-ին:

Մասնագիտական խորհրդի
գիտական քարտուղար, ֆ.մ.գ.դ. Հ. Գ. Սարուխանյան

Тема диссертации утверждена в Институте проблем информатики и автоматизации НАН РА
Научный руководитель: доктор тех.наук В.И. Таирян
Официальные оппоненты: доктор тех.наук Г.А. Арутюнян
кандидат тех.наук. С.Е. Абрамян

Ведущая организация: Государственный инженерный университет Армении

Защита состоится 6-ого июня 2014г. в 16:00 на заседании специализированного совета 037 «Информатика и вычислительные системы» Института проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака 1.
С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.

Автореферат разослан 6-ого мая 2014г.

Ученый секретарь специализированного
совета, д.ф.м.н. А. Г. Саруханян

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В начале XXI-го века вместе с информационными технологиями бурное развитие получили каналы связи, в частности, выросли их пропускные способности даже на дальних дистанциях. В связи с этим стало возможным обращение к удаленным данным на больших по сравнению с прежним скоростях с помощью таких каналов и соответственно развились инструменты, позволяющие расширить как их применение, так и географию.

Одним из применений новых возможностей стало создание программного обеспечения, дающего возможность удаленного хранения и доступа к информации. Область применения таких продуктов довольно широка. Она может включать как сферы коммерческой деятельности, так и научной.

Существуют некоторые разновидности программного обеспечения, позволяющего организовать удаленное управление данными. На сегодняшний день активно демонстрируют рост сервисы и программные приложения, называемые сетевыми хранилищами данных (cloud storage). Данный тип программного обеспечения предполагает, что можно осуществлять непосредственный доступ к данным, при этом доступ может осуществляться с различных устройств, которые могут быть расположены в различных географических локациях, существует возможность подключения механизмов синхронизации, поддерживаются механизмы защиты от отказов физических носителей, могут быть реализованы возможности по доступу к старым версиям (архиву) данных, а также присутствует возможность разрешения доступа к своим данным других пользователей.

В сфере разработки программного обеспечения такие компании, как Microsoft, Google, Apple представили свои разработки в данной области: OneDrive, Google Drive, ICloud соответственно. Вместе с развитием индустрии также появились и новые игроки рынка, специализирующиеся именно на этом сегменте, особого упоминания среди них достоин Dropbox Inc. со своим продуктом Dropbox, завоевавшим одну из лидирующих позиций на рынке, что лишний раз демонстрирует потребность в подобных решениях, а соответственно, и их актуальность.

На данный момент существуют некоторые задачи, для решения которых необходимо модифицировать реализацию сетевых хранилищ.

Одна из актуальных проблем - это проблема строгого разграничения доступа к данным. В конкретных ситуациях, исходя из требований конфиденциальности, некоторое множество данных должно быть доступно только узкому кругу пользователей, при этом отношения между ними могут быть представлены с помощью иерархии. Сервисы сетевого хранения, представленные на данный момент, не предоставляют возможностей по организации иерархии или же эти возможности покрывают лишь очень простые их разновидности. Так, например, сервис Google Drive позволяет организовать иерархию, состоящую только лишь из двух уровней.

Другой проблемой является присутствие ограничений по развертыванию и аудиту программного обеспечения сетевого хранилища. Дело в том, что подавляющее

большинство игроков рынка, кроме того, что занимаются разработкой программного обеспечения, также сами занимаются его развертыванием на серверах, выбранных ими же, не предоставляя своим пользователям таких возможностей. В то же время, программное обеспечение, находящееся сегодня в свободном доступе, не способно предоставить приемлемую реализацию даже минимального набора функций, предоставляемых популярными сервисами. Помимо того, в связи с тем, что пользователь не имеет доступа к серверному программному обеспечению, для него невозможна организация аудита данной составляющей сервиса, чем могут злоупотребить разработчики внедряя в исходный код произвольные модули, не связанные с заявленным функционалом.

Существуют также возможности по повышению уровня информационной безопасности. Такие возможности могут быть связаны с такими процедурами, как хранение данных, их передачу, а также механизмами обеспечения конфиденциальности.

В связи с вышесказанным, представляется актуальной разработка новой системы удаленного управления данными, являющейся разновидностью сетевых хранилищ, поддерживающей иерархическое разделение, позволяющей развертывание на мощностях, выбранных пользователями с общедоступными алгоритмами работы, с улучшенным уровнем информационной безопасности.

Цель и задачи работы. Целью диссертационной работы является разработка и реализация программного обеспечения для организации многопользовательского сетевого хранилища, с учетом иерархии между пользователями, с более широкими возможностями по развертыванию и аудиту, а также с повышением уровня информационной безопасности.

Для достижения указанной цели можно выделить ряд следующих задач:

- разработка модели разграничения доступа к данным многопользовательского сетевого хранилища, учитывающей иерархические связи между пользователями;
- модификация методов и механизмов управления пользовательскими учетными записями, позволяющими регулировать пользователей системы сетевого хранения данных на основе иерархий между ними;
- модификация механизмов хранения и передачи данных сетевого хранилища для повышения уровня безопасности;
- реализация программного обеспечения для организации разработанной системы, состоящей из серверного модуля, соответствующего программного интерфейса для взаимодействия с ним, а также клиентского приложения.

Объект исследования. Объектом исследования являются сервисы сетевого хранения данных, являющиеся разновидностью систем удаленного управления информацией посредством каналов связи в лице Internet сети и базовый функционал, предоставляемый ими.

Методы исследования. Исследования, проводимые в работе, основаны на анализе источников литературы по тематике систем удаленного управления данными, в частности систем сетевого хранения.

Научная новизна.

- Разработана новая модель управления доступом, адаптированная для организации многопользовательского иерархического сетевого хранилища, обладающая рядом преимуществ, которая была проверена на соответствие требованиям стандарта ролевой модели RBAC (Role Based Access Control), разработанной институтом NIST (National Institute of Standards and Technology).
- Предложены модифицированные механизмы управления учетными записями пользователей представленной модели разграничения доступа, механизмы автоматизации некоторых процессов при управлении.
- Модифицированы методы хранения и передачи данных сетевого хранилища для повышения уровня их безопасности.

Практическая значимость полученных результатов. Разработанная система может применяться для организации новых сетевых хранилищ. Кроме того, данная система может также применяться как промежуточный элемент для работы с уже с существующими сетевыми хранилищами, позволяя при этом организовать многоуровневую иерархию, более широкие возможности по аудиту, а также повысить уровень информационной безопасности.

Получены следующие результаты, представляющие практический интерес:

- разработано программное обеспечение для организации серверного модуля сетевого хранилища на основе разработанной системы;
- разработан программный интерфейс, предназначенный для взаимодействия с серверным модулем;
- реализовано клиентское приложение, взаимодействующее с серверным модулем посредством соответствующего программного интерфейса.

На защиту выносятся следующие положения:

- новая модель разграничения доступа к данным, созданная для организации многопользовательского сетевого хранилища, учитывающая иерархические связи между пользователями, а также имеющая ряд преимуществ;
- модифицированные механизмы управления пользовательскими учетными записями сетевого хранилища;
- модифицированные механизмы хранения и передачи данных сетевого хранилища;
- программное обеспечение, реализующее многопользовательское иерархическое сетевое хранилище, состоящее из серверного модуля, клиентского модуля и программного интерфейса их взаимодействия.

Внедрение. Результаты диссертационной работы внедрены на предприятии LIA-K GROUP занимающемся дистрибуцией различных товаров. В результате были модифицированы методы хранения и передачи некоторых данных данного предприятия.

Апробация работы. Основные положения и материалы диссертационной работы докладывались и обсуждались на годичных конференциях РАУ, на международной конференции по вопросам безопасности, на семинарах факультета прикладной математики и информатики и кафедры Математической кибернетики, ГОУ ВПО Российско-Армянского (Славянского) университета, семинаре Института проблем инфворматики и автоматизации НАН РА и на международной конференции CSIT-13.

Публикации. Основные результаты опубликованы в 6 научных трудах [1-6].

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка использованной литературы. Общий объем работы - 110 страниц, включая 30 рисунков, 88 наименований в списке использованной литературы.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы, сформулированы цели и задачи работы, научная новизна и практическое значение полученных результатов, а также основные положения, выносимые на защиту.

В первой главе приведен обзор литературы и анализ современного состояния рассматриваемой проблемы. Исследуется рынок сетевых хранилищ, вследствие чего приводится список их особенностей и основных функций, в ряд которых входят: возможности по предоставлению удаленного доступа к данным с различным устройств вне зависимости от их географического расположения, возможности по подключению механизмов синхронизации, поддержка механизмов защиты от отказов физических носителей, возможности по реализации доступа к старым версиям(архиву) данных, возможность разрешения доступа к своим данным других пользователей. Приводится также статистика роста популярности сервисов сетевого хранения данных и прогнозы крупных компании по их развитию.

В этой же главе анализируется архитектура программного обеспечения для организации сетевого хранилища. Существующие сервисы базируются на клиент-серверной архитектуре. Иными словами, наибольшая нагрузка и основные функции по управлению данными контролируются сервером и, соответственно, основные операции проходят с помощью его вычислительных мощностей. При таком построении важно организовать распределение обязанностей между клиентом и сервером для обеспечения безопасности, функциональности и скорости работы.

Также первая глава содержит ряд сведений и определений, необходимых при организации сетевого хранения данных. К таким сведениям относятся сведения об архитектуре клиент-сервер, о протоколе передачи данных, в частности о протоколе HTTPS, сведения об информационной безопасности, шифровании данных, авторизации и аутентификации пользователей.

Отдельный параграф первой главы посвящен основным возможностям по улучшению существующих решений на рынке сетевых хранилищ данных, связанным с организацией иерархии между пользователями, преодолением некоторых ограничений по

развертыванию серверного модуля программного обеспечения, а также повышением уровня информационной безопасности.

Во второй главе описывается предлагаемая модель управления данными сетевого хранилища с возможностью организации иерархии между пользователями.

Для реализации такой модели решается несколько задач: 1) разработка модели разграничения доступа к данным сетевого хранилища с поддержкой иерархии между пользователями; 2) модификация механизмов управления пользовательскими учетными записями; 3) выбор механизмов минимизации конфликтов при параллельном доступе к данным; 4) выбор механизмов протоколирования активности пользователей и поддержки версий данных.

В данной главе были исследованы и проанализированы основные модели разграничения доступа: мандатная, дискреционная, ролевая. Были также выявлены некоторые особенности этих моделей.

Классическая мандатная модель предоставляет возможности по организации строгой иерархии, однако она не лишена недостатков: в ней не предусмотрен обмен информаций между субъектами одного уровня; между субъектами с различным уровнем существует только односторонняя связь; существует также возможная проблема при администрировании системы доступа, ибо она предусматривает только линейную иерархию, недостаточную для решения многих задач на практике.

В дискреционной модели администратором системы выдаются привилегии, использование которых возможно по отдельности с каждым объектом системы. При большом количестве субъектов и объектов вычисления могут стать громоздкими, а модификация прав сложна в ключе администрирования. С целью облегчения администрирования дискреционной модели можно использовать методы группирования как субъектов, так и объектов, к которым осуществляется доступ, однако такой подход выходит уже за рамки модели.

В классической ролевой модели соответствующая совокупность прав для доступа к объектам системы в целом образует так называемую роль. Для простоты администрирования между ролями может быть добавлена иерархия. В создании и администрировании классическая модель проста. Данная модель является наиболее удобной в большинстве задач на практике. Ролевая модель, к сожалению, также имеет некоторые недостатки. Роли в модели носят глобальный характер, вследствие чего права субъектов действуют сразу во всей системе. Таким образом, имея совокупность прав, субъект может применить их к любым объектам системы, каковы бы ни были их типы. Получается, что субъект, который и вовсе не должен был взаимодействовать с некоторым объектом, злоупотребляет своими правами, чем может навредить другим субъектам и системе в целом. Другой проблемой является то, что в системе субъект не может единолично управлять своим собственным объектом, все права по умолчанию будут переданы другим субъектам, стоящим выше по иерархии. Также в ролевой модели совокупность операций присутствует только у ролей, объекты же не рассматривают их.

В связи с рядом недостатков, свойственных классическим моделям управления доступом, были разработаны их вариации для устранения некоторых пробелов, адаптированные под нужды конкретной задачи. В данной главе описывается модель

управления доступом, базирующаяся на ролевой, с учетом принципов мандатной и дискреционной, адаптированные под нужды сетевого хранилища данных.

Рассмотрим основные понятия, связанные с предлагаемой моделью разграничения доступа.

Множество единиц информационной системы разбивается на **объекты** и **субъекты**. Объекты – это информационные ресурсы. В качестве информационных ресурсов могут выступать файлы, записи в таблице базы данных и т.д. Субъекты – это пользователи ресурсов. Пользователи обычно реализуются с помощью пользовательских учетных записей.

Для представления объектов используется понятие класса данных. Для представления субъектов используется понятие роли.

Класс данных C_i представляет собой:

- совокупность описаний ресурсов r_i, $i=1,2,...n$, где n-количество ресурсов, заданных в данный момент в системе;
- совокупность классов данных $\{C_{j1}, C_{j2}, ..., C_{jk}\}$.

Каждый класс должен обладать набором параметров, описывающих его и соответственно ресурс или группу ресурсов, с которой он ассоциируется. Введем эти параметры по аналогии с параметрами ресурсов и групп ресурсов.

Во-первых, необходим параметр, с помощью которого можно будет выделить класс среди других, в связи с чем к каждому классу C_i приписывается **уникальный идентификатор ClassID**, позволяющий однозначно определить его среди других.

Другим типом параметров может служить **наименование класса ClassName**. Значение данного параметра не обязательно уникально. Так как пользователи системы в основном представляют собой людей, то необходим более приемлемый способ представления класса, частью которого является наименование на понятном для них языке. Тем самым, этот параметр может позволить обеспечить более дружелюбный интерфейс при работе с классами и с системой в целом.

Присутствует также параметр, представленный с помощью **переменной ClassT**, значение которой представляет собой временной промежуток. Часто требуется ограничить доступ к данным еще и по такому параметру, как время, также может потребоваться осуществление мониторинга системы по этому параметру. Данная переменная помогает решению задач, связанных с временными ограничениями, а также фильтровать данные по этому параметру. Таким образом, наличие переменной **ClassT** обеспечивает большую гибкость модели управления доступом.

Еще одним параметром является **переменная, представляющая собой набор привилегий ClassP**, доступных для класса. С помощью данного параметра представляется возможным контроль множества операций, выполняемых над ресурсом или группой ресурсов, ассоциируемых с этим классом.

Для определения ступени иерархии, на которой находится класс данных используется параметр, ссылающийся на родительский для него класс **ParentClassId**, если, конечно, такой для него присутствует. Такой параметр позволяет определить иерархию между классами данных.

Построение ступеней иерархии L_j для удобства будет осуществляться, начиная от самой высокой со значением 1 до самой низкой путем увеличения значения каждый раз на единицу. Такой подход выбран в силу того, что системы строятся, начиная с задания наивысшего приоритета.

Ну и, наконец, задается **переменная Data, хранящая данные**, представляющие собой ресурс или группу ресурсов, которые в свою очередь могут быть разбиты на категории.

Учитывая иерархию и концепцию своей структуры, классы могут быть связаны между собой отношениями наследования. В таком контексте класс на более высоком уровне включает в себя класс на более низком уровне в случае, если они входят друг с другом в отношения типа предок-потомок, где класс на более высоком уровне в соответствующей цепочке наследования выступает в роли предка для класса на более низком уровне. Иными словами, имеем следующее формальное представление:

$$C_x(L_j) \subseteq C_y(L_k),$$

где $j>k$ и $C_x(L_j)$ находится в отношениях наследования с $C_y(L_k)$, в которых $C_x(L_j)$ играет роль потомка произвольного уровня, а $C_y(L_k)$ роль предка в общей для них цепочке наследования.

Таким образом, в широком смысле, то есть в случае, когда имеют место отношения наследования, класс – это совокупность других классов, тем самым представляя собой совокупность абстрактных описаний некоторой группы или коллекции ресурсов. То есть для произвольного класса C_i справедливо:

$$C_i = \{C_{j1}, ..., C_{jk}\},$$

где $k \leq |\{Ci\}|$ и $C_{j1}, ..., C_{jk}$ – классы из множества $\{Ci\}$.

На одном уровне может присутствовать произвольное количество классов, независимых друг от друга, то есть для любых двух из них $C_x(L_j)$ и $C_y(L_j)$ выполняется условие $C_x(L_j) \neq C_y(L_j)$. Подобное расположение позволяет отразить связи между данными в реальных условиях, позволяя разграничить между собой группы, находящиеся на одном уровне иерархий, однако не взаимодействующие между собой.

В реальных условиях обычно присутствует один основной класс со всем множеством данных и параметров, который можно разбить на более простые. Этот факт связан с тем, что данные для некоторой группы можно представить единым целым. В свою очередь, эти классы также могут быть разбиты на более простые. Например, можем составить следующее множество классов: $C_1=<C_2, C_3>$; $C_2=<C_4, C_5>$; $C_5=<C_6>$.

Тем самым мы представили множество объектов системы, являющихся информационными ресурсами, задав их с помощью классов, однозначно идентифицируемых с помощью **ClassID** и именуемых с помощью **ClassName**, ввели переменную **ClassT** для разграничения доступа к ресурсам, исходя из значения времени, ввели переменную **ClassP,** содержащую набор привилегий, доступных для конкретного класса, установили иерархию, введя понятие родительского класса **ParentClassId** и определив отношения наследования.

Аналогично объектам проведем и группирование для субъектов. Множество субъектов представим с помощью **ролей,** и будем иметь дело с множеством ролей $\{R_j\}$, где $i = 1, 2, ..., n$ и n - количество ролей, определенных в нашей системе. Под ролью будем подразумевать совокупность некоторых прав и обязанностей пользователей. Так как в реальности роли

должны иметь возможность соединяться в более крупные, то зададим для них это свойство, то есть некая роль R_g из множества ролей $\{R_i\}$ может представлять собой объединение неких других ролей R_h и R_j из того же множества $\{R_i\}$.

Для большего соответствия логики работы информационной системы принципам разделения функций сотрудников коммерческих и некоммерческих организаций или других разновидностей рабочих групп была выработана идея ролевой модели контроля доступа. Основываясь на активности в системе, ролевая методика контролирует доступ пользователей к информации. Для реализации подобного подхода необходимо в первую очередь определить понятие роли в системе. В классическом понимании принято под ролью подразумевать совокупность некоторых действий и обязанностей, связанных с определённым видом деятельности. Подобный подход дает возможность не указывать явно типы доступа для каждого из пользователей к каждому объекту системы, что обычно нагромождало систему и делало ее недружелюбной для администрирования. Используя ролевую модель, достаточно определить роли, указать типы доступа к объектам для них и, соответственно, связать роли с пользователями системы.

Для определения роли в системе необходимо определить набор присущих ей характеристик. С этой целью определим набор параметров, ассоциирующихся с ролями в системе. Введем эти параметры, исходя из природы активности.

Чтобы однозначно идентифицировать роль среди других, введем для каждой роли R_i **уникальный идентификатор RoleID**, хранящий уникальное значение.

Введем параметр, позволяющий определить **наименование роли RoleName**. Значение данного параметра не обязательно уникально. Наименование роли придаст ей дополнительную описательную информацию.

Присутствует также параметр, представленный с помощью **переменной RoleT**. Значение данной переменной представляет собой временной промежуток и служит для задания временных ограничений действия роли.

Для роли определим также множество привилегий **RoleP,** ассоциируемых с ней. С помощью привилегий определяется множество элементарных операций e_z, которыми пользователю позволено пользоваться при ассоциировании с конкретной ролью, учитывая также временные ограничения. Могут быть роли с более широкими привилегиями, а также роли с более узкими, в связи с чем одна роль может, по сути, являться частью другой. Таким образом, например, возможно следующее соотношение:

$$P(R_i) \subseteq P(R_j),$$

где R_i и R_j некоторые роли из множества $\{R_i\}$.

Можно добавить к элементарным операциям опции, можно вдобавок условно разделить элементарные операции на несколько составляющих, делая модель более гибкой и прозрачной. При выполнении элементарных операций в несколько этапов можно реализовать подход разделенных обязанностей. Так, например, если множество имеющихся элементарных операций представляет собой операцию чтения и операцию записи **{read, write}**, то каждую из них можно разбить еще на несколько. Так, например, операцию чтения содержимого ресурса можно разбить на составляющие, задав опции открытия доступа на чтение, опцию получения содержимого и соответственно операцию закрытия доступа на чтение, аналогично и для других операций:

read = open read access –> read -> close read access

write = open write access -> write –> close write access

Таким образом, каждая из операций из множества **{read, write}** была разбита на операцию по открытию доступа, на осуществление самой операции и закрытие доступа. Конечно, разбиение может выглядеть и другим образом, все зависит от текущей реализации системы.

Для задания иерархии между ролями введем параметр роли, соответствующий идентификатору родительской роли, если, конечно, такая присутствует **ParentRoleID**. Для каждой из ролей определим также **параметр уровня** RL_j, **j=1,2,...n**, где **n**-количество уровней, определенных на данный момент, указывающий на ступень в иерархию, занимаемую роль и, соответственно, позволяющий определить иерархию для ролей и их групп. По аналогии с классами ресурсов построение ступеней иерархии для удобства будет осуществляться, начиная от самой высокой со значением 1 до самой низкой путем увеличения значения каждый раз на единицу в силу того, что системы строятся, начиная с задания наивысшего приоритета.

Помимо прочего, роли должны обладать параметром **RoleClassId**, являющимся ссылкой на классы данных, с которыми они ассоциируются.

Учитывая иерархию и концепцию своей структуры, роли, как и классы информационных ресурсов, могут быть связаны между собой отношениями наследования. Таким образом, роль на более высоком уровне в структуре иерархии включает в себя класс на более низком уровне в случае, если они находятся друг с другом в отношениях типа предок-потомок, где роль на более высоком уровне в соответствующей цепочке наследования выступает в качестве предка для роли на более низком уровне. Тем самым имеем следующее формальное представление данного утверждения:

$$R_x(RL_j) \subseteq R_y(RL_k),$$

где $j>k$ и $R_x(L_j)$ находится в отношениях наследования с $R_y(L_k)$, в которых $R_x(L_j)$ выступает в качестве потомка произвольного уровня, а $R_y(L_k)$ в качестве предка в общей для них цепочке наследования.

Таким образом, в широком смысле, то есть в случае когда имеют место отношения наследования, роль – это совокупность других ролей. То есть для произвольной роли R_i справедливо следующее:

$$R_i = \{R_{j1}, ..., R_{jk}\},$$

где $k \leq |\{Ri\}|$ и $R_{j1}, ..., R_{jk}$ – роли из множества $\{Ri\}$.

На одном уровне может присутствовать произвольное количество ролей, независимых друг от друга, то есть для любых двух из них $R_x(RL_j)$ и $R_y(RL_j)$ выполняется условие $R_x(RL_j) \neq R_y(RL_j)$. Подобное расположение позволяет отразить связи между ролями в реальных условиях, позволяя разграничить между собой те, что находятся на одном уровне иерархий, однако не взаимодействуют между собой.

В действительности получается, что может присутствовать одна основная роль со всем множеством привилегий, которую можно разбить на более простые. Данный факт исходит из того, что по сути роли могут быть объединены в одну роль с полным набором привилегий и обязанностей. В свою очередь, эти роль также могут быть разбиты на более

простые. Иными словами, можем составить следующее множество ролей: **R_1=<R_2,R_3>; R_2=<R_4, R_5>; R_5=<R_6>**.

Каждый пользователь может быть ассоциирован с несколькими ролями, при этом эти роли стыкуются для представляющей его учетной записи, соответственно, объединяются и привилегии, связанные с этими ролями. Одна роль может быть ассоциирована со многими пользователями, вот почему необходимо также вводить концепцию сессии с конкретной реализацией механизмов журналирования.

Бывают ситуации, когда необходимы ограничения касательно пользователей и ролей. Так, например, возможно, чтобы какая-то роль могла использоваться в какой-то момент только одним пользователем или же, учитывая в нашей реализации ролевой модели присутствие временного параметра, какая-то роль может быть доступна для определенных лиц по определенному графику. Возможно ограничение на одновременное использование ролей, то есть могут быть роли, которые не могут быть назначены одной и той же учетной записи одновременно. Могут также существовать привилегии, которые не могут быть заданы для одной и той же роли одновременно, такие роли обычно называют статически взаимоисключающими. Те же ограничения могут быть сформулированы с учетом временных параметров.

Множества классов данных и ролей строятся одновременно и рассматриваются параллельно. Это происходит потому, как, исходя из построения, эти множества тесно связаны между собой, ведь роли содержат ссылки на классы данных, с которыми ассоциируются.

Действия ролей над классами данных должны быть протоколированы с помощью специальных механизмов журналирования.

Операции субъектов над объектами через призму определенной модели доступа осуществляется путем активации для учетной записи соответствующей роли или группы ролей с «привязанным» к ним соответствующим классом данным или группой классов. При обращении пользователя к некоторому информационному ресурсу определяется принадлежность ресурса классу данных. В случае наличия для роли возможности доступа к такому ресурсу отправляются соответствующие команды на вход к буферу операций и на выходе выдается результат.

Также предлагается графическое представление модели с помощью деревьев, которое позволит сделать протекающие в ней процессы более прозрачными, повысив соответственно ее гибкость и удобство администрирования.

В работе приведена методика по регулированию ролей пользователей в модели разграничения доступа, представленной выше, для системы удаленного управления данными. Для этого вводится понятие **административных ролей**, а также определяется функционал и ограничения, связанные с ними. В этом случае под **обыкновенными** подразумеваются роли, действующие тем или иным образом на классы данных, а под **административными** роли, действующие над обыкновенными ролями.

$R_i \leftrightarrow C_i$

$AR_i \leftrightarrow R$

Для каждой административной роли в модели определим несколько следующих функций, исходя из стандартов ролевой модели и конструкции модели в данном конкретном случае:

- функция **AssignRole**, имеющая два параметра, среди которых роль и пользователь, назначаемый на нее, с помощью которой уполномоченный пользователь с административной ролью может назначить роль из множества заданных ролей R пользователю из множества зарегистрированных в системе пользователей U;
- функция **RevokeRole**, имеющая два параметра, среди которых роль и пользователь, у которого она отзывается, с помощью которой уполномоченный пользователь с административной ролью может отозвать роль из множества ролей R, ассоциируемых с пользователем из множества зарегистрированных в системе пользователей U.
- функция **AssignAdministrativeRole**, имеющая два параметра, среди которых административная роль и пользователь, назначаемый на нее, с помощью которой уполномоченный пользователь с административной ролью может назначить административную роль из множества заданных ролей AR пользователю из множества зарегистрированных в системе пользователей U;
- функция **RevokeAdministrativeRole**, имеющая два параметра, среди которых административная роль и пользователь, у которого она отзывается, с помощью которой уполномоченный пользователь с административной ролью может отозвать административную роль из множества ролей **AR**, ассоциируемых с пользователем из множества зарегистрированных в системе пользователей **U**.

Административные и обыкновенные роли могут обладать свойством наследования, описанным выше, в связи с чем для каждого из этих множеств можно организовать иерархию.

Была также проведена автоматизация некоторых процессов, проходящих в системе, связанных с управлением ее архитектурой, в список которых входят операции по созданию ролей, их удалению и т.д.

Модель успешно проверяется на соответствие требованиям стандарта ролевой модели RBAC (Role Based Access Control), разработанным институтом NIST (National Institute of Standards and Technology).

Помимо того, что модель управления доступом, предложенная в данной работе, была разработана и адаптирована к нуждам задачи по организации сетевого хранилища, а также удовлетворяет требованиям стандарта таких моделей NIST RBAC, она также обладает рядом дополнительных преимуществ. Среди таких преимуществ четкой разделение между

объектами и субъектами, позволяющее достигнуть выигрыша в производительности и уровне защиты системы при определенных обстоятельствах.

В главе выбираются механизмы минимизации конфликтов, связанные с осуществлением многопользовательского доступа и соответственно вводятся некоторые новые элементы в формальное представления модели управления данными.

Для объектов модели определим переменную B_i, отвечающую за их блокировку. Согласно требованиям данная переменная имеет три возможных значения: 1) значение, соответствующее отсутствию блокировки, 2) значение, соответствующее полной блокировке класса данных и 3) значение, соответствующее частичной блокировке, имеющее параметры, с помощью которых можно определить, на какую именно операцию по отношению к классу она применяется.

Для реализации возможности оптимистической и пессимистической блокировок введем для каждого класса данных C_i переменную V_i, хранящую числовое значение, соответствующее номеру версии этого класса. Также для каждого класса данных C_i введем переменную LU_i, хранящую идентификатор пользователя, которому удалось успешно провести последнее обновление данных, соответствующих этому классу и время самого изменения.

Исходя из архитектуры модели, блокировки также обладают свойством наследования. То есть реализации блокировок с низкой степенью детализации упрощена, так как многое уже заложено в конструкции системы. При этом введем также для этого переменную AV_i ассоциируемую с произвольным набором классов данных $\{C_{i1},\ldots,C_{ik}\}$, хранящую числовое значение соответствующе номеру версии этого набора.

Реализация неявной блокировки происходит на уровне кода программного обеспечения.

Для модели выбираются методы протоколирования активности пользователей и поддержки версионности данных. Для поддержки протоколирования и версий модифицируется под нужды разработанной модели несколько существующих методов: метод последнего изменения, метод дублирования классов данных, метод ввода модуля для хранения данных аудита.

Таким образом, во второй главе разработана модель управления доступом для иерархической многопользовательской системы управления данными, разработана методика, позволяющая управлять учетными записями пользователей в рамках предложенной модели, для минимизации конфликтов между пользователями в модель добавлено понятие блокировок нескольких разновидностей, в рамках модели выбраны механизмы протоколирования активности пользователей, а также для данных введена возможность работы с версиями.

В третьей главе описывается выбор и модификация механизмов взаимодействия с иерархической системой многопользовательского управления данными.

Для реализации подобного взаимодействия решается несколько задач: 1) выбираются и модифицируются механизмы хранения данных; 2) выбираются методы резервирования данных; 3) выбираются и модифицируются методы передачи данных; 4) выбираются механизмы аутентификации и авторизации.

Для хранения непосредственно данных системы было решено воспользоваться обыкновенной файловой системой, для хранения же информации о системе и составляющих ее архитектуры и можно воспользоваться базой данных. Реализация механизмов хранения может варьироваться, при этом реализация соответствующей модели управления будет претерпевать минимальные изменения.

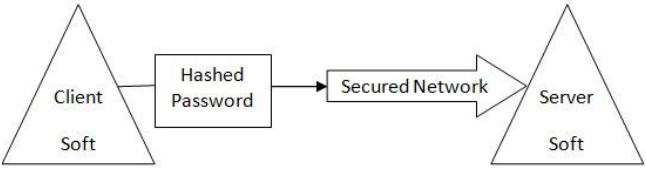
При этом сами данные опционально могут иметь возможность хранения в зашифрованном виде. Шифровка самих данных при этом будет осуществляться на основе симметричного алгоритма шифрования. Предварительное шифрование может повысить безопасность пользовательских данных.

Кроме того, для пользователей вводится понятие общего ключа шифрования иерархии, с помощью которого они смогут обмениваться зашифрованными данными, минуя возможность их хранения в открытом виде на стороне сервера.

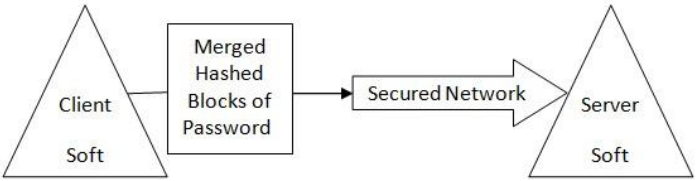
Для реализация методов резервирования данных поддерживается возможность хранения избыточных наборов данных, которым иможно воспользоваться в случае возникновения необходимости.

В настоящей работе в качестве протокола для передачи данных многопользовательской иерархической системы было решено использовать протокол HTTPS, вернее осуществить надстройки над ним. Такой выбор обусловлен тем, что в большинстве аналогов при анализе данной проблемы приходят примерно к такому решению, при этом эта задача не стоит в приоритетах настоящей диссертационной работы, в связи с чем такой выбор выглядит разумным. В случае использования для передачи данных некоего другого протокола сама реализация системы не будет существенно отличаться.

Реализация механизмов авторизации и аутентификации проводится в согласованности с рекомендациями, данными в первой главе настоящей работы. В этой части рассматриваются некоторые механизмы, подвергающиеся модификации. Так при авторизации пароли пользователей подвергаются предварительному шифрованию, в связи с чем схема авторизации имеет следующий вид:



или



В четвертой главе описывается реализация иерархического многопользовательского сетевого хранилища на основе предложенной модели управления доступом.

Описаны модули программного обеспечения и основные пользовательские интерфейсы для взаимодействия с ним.

Разработка серверного модуля осуществлялась на базе сервера Apache с применением языка программирования PHP в комбинации с SQL базами данных. Разработка клиентского приложения осуществлялась на базе платформы .NET с применением языка программирования C#.

Основные результаты диссертационной работы

1. Разработана новая модель разграничения доступа к данным, адаптированная для организации многопользовательского сетевого хранилища, учитывающая иерархические связи между пользователями, а также имеющая ряд дополнительных преимуществ [1-4].
2. Модифицированы механизмы управления пользовательскими учетными записями сетевого хранилища [6].
3. Выбраны и модифицированы механизмы хранения и передачи данных сетевого хранилища [5].
4. Реализовано программное обеспечение для организации разработанной системы, состоящее из серверного модуля, соответствующего программного интерфейса для взаимодействия с ним, а также клиентского приложения [акт о внедрении].

Список публикаций

1. Берберян Л.С. “Об одной модели управления доступом к ресурсам информационной системы”, Шестая годичная научная конференция РАУ, Ереван 2011, с.66-70.
2. Берберян Л.С. “Разработка и реализация модели информационной системы с возможностью управления доступом”, Вестник РАУ, Ереван 2012, с.42-48.
3. Берберян Л.С. “Классификация сущностей информационной системы и ее внедрение в соответствующую модель”, материалы международной научно-практической конференции “Безопасность личности: состояние и возможности обеспечения”, Пенза-Ереван-Колин, 10-11 мая 2012 года, с. 64-68.
4. Levon Berberyan "About one Extended Role-based Access Control Formal Model", Computer Sciense and Information Technologies conference, CSIT-13, Armenia, Yerevan, 23-27 September, 2013, p.99-102.
5. Berberyan L.S. "On some techniques for improving cloud storage users security", Proceedings of Engineering Academy of Armenia, volume 10, number 4, Yerevan 2013, p.764-768.
6. Берберян Л.С. "О регулировании ролей пользователей в модели разграничения доступа системы удаленного управления данными", Mathematica Montisnigri, 2014, с.98-103.

Տվյալների հեռակա կառավարման բազմանդամ հիերարխիկ համակարգի մշակում և ծրագրային իրականացում

XXI դարում արագ զարգանում են կապուղիները, մասնավորապես աճել է դրանց թողունակությունը նույնիսկ երկար հեռավորությունների վրա: Նոր հնարավորությունների կիրառումներից մեկն է այնպիսի ծրագրային ապահովման ստեղծումը, որը հնարավորություն է տալիս կազմակերպել տվյալների հեռակա պահպանումը և մուտք գործելու հնարավորությունը:

Գոյություն ունեն ծրագրային ապահովման մի քանի տեսակներ, որոնք թույլ են տալիս կազմակերպել տվյալների հեռավոր կառավարում: Այսօր ակտիվորեն աճում են ծառայությունները և ծրագրային հավելվածները, որոնք կոչվում են ցանցային կամ ամպային պահեստներ (cloud storage): Ծրագրային ապահովման այդ տեսակը ենթադրում է, որ նրա անդամները կարող են մուտք ունենալ տվյալներին տարբեր սարքավորումներից, որոնք կարող են գտնվել տարբեր աշխարհագրական վայրերում, հնարավոր է միացնել համաժամացման մեխանիզմները, աջակցվում են ֆիզիկական սարքերի անսարքությունների դեմ մեխանիզմները, հնարավոր է իրականացնել տվյալների հին տարբերակների (արխիվ) մատչելիությունը, ինչպես նաև գոյություն ունի տվյալների մուտքի հնարավորությունը այլ անդամներին փոխանցելու հնարավորություն:

Այսօր գոյություն ունեն մի շարք խնդիրներ, որոնց լուծման համար անհրաժեշտ է փոփոխել ցանցային պահեստների իրականացումը: Այդպիսի խնդիրներից մեկն է տվյալների մատչելիության խիստ սահմանափակումը՝ անդամների միջև հիերարխիայի համաձայն: Մեկ այլ խնդիր է ցանցային պահեստի ծրագրային ապահովման տեղադրման և աուդիտի հետ կապված սահմանափակումը: Գոյություն ունեն նաև տեղեկատվական անվտանգության բարելավման հնարավորություններ, որոնք կապված են այնպիսի ընթացակարգերի հետ, ինչպիսիքն են տվյալների պահպանումը, դրանց փոխանցումը, մենության ապահովման մեխանիզմները:

Ատենախոսության նպատակն է բազմանդամ ցանցային պահեստի կազմակերպման այնպիսի ծրագրային ապահովման մշակումը և իրականացումը, որը հաշվի է առնում անդամների միջև գործող հիերարխիան, տրամադրում է տեղադրման և աուդիտի հետ կապված լրացուցիչ հնարավորություններ, ինչպես նաև ապահովում է ավելի բարձր տեղեկատվական անվտանգության մակարդակ:

Գիտական նորույթը

- Մշակվել է նոր մուտքի վերահսկման մոդել, որը հարմարեցված է հիերարխիկ բազմանդամ ցանցային պահեստի կազմակերպման համար, ունի մի շարք առավելություններ, ստուգված է NIST (National Institute of Standards and Technology) ինստիտուտի կողմից մշակված RBAC (Role Based Access Control) դերային մոդելի ստանդարտի պահանջների համապատասխանությունը:
- Առաջարկված են ներկայացրած մուտքի վերահսկման մոդելի անդամների կառավարելու մոդիֆիկացրած մեխանիզմներ, որոշ կառավարման ընթացակարգերի ավտոմատացման մեխանիզմներ:

- Փոփոխման են ենթարկվել ցանցային պահեստի տվյալների պահպանման և փոխանցման մեթոդները դրանց անվտանգությունը բարելավելու նպատակով:

Արդյունքների կիրառական նշանակությունը

Մշակված համակարգը կարող է օգտագործվել նոր ցանցային պահեստների կազմակերպման համար: Այս համակարգը կարող է օգտագործվել նաև որպես միջանկյալ տարր արդեն գոյություն ունեցող ցանցային պահեստի հետ աշխատելու համար, թույլ տալով կազմակերպել բազմամակարդակ հիերարխիա, տրամադրելով աուդիտի հետ կապված լրացուցիչ հնարավորություններ, ինչպես նաև ապահովելով ավելի բարձր տեղեկատվական անվտանգության մակարդակ:

Պաշտպանությանը ներկայացվող դրույթները

- Մուտքի վերահսկման նոր մոդել ստեղծված է բազմանդամ ցանցային պահեստի կազմակերպման համար, որը հաշվի է առնում անդամների հիերարխիկ հարաբերությունները, ինչպես նաև ունի մի շարք առավելություններ:
- Ցանցային պահեստի անդամների կառավարման մոդիֆիկացրած մեխանիզմներ:
- Ցանցային պահեստի տվյալների պահպանման և փոխանցման մոդիֆիկացրած մեխանիզմներ:
- Ծրագրային ապահովում, որը իրականացնում է բազմանդամ հիերարխիկ ցանցային պահեստ, որը բաղկացած է սերվերի մոդուլից, կլիենտի մոդուլից և դրանց փոխազդեցության ծրագրային ինտերֆեյսից:

Աշխատանքի հիմնական արդյունքները

- Մշակված է մուտքի վերահսկման նոր մոդել ստեղծված բազմանդամ ցանցային պահեստի կազմակերպման համար, որը հաշվի է առնում անդամների հիերարխիկ հարաբերությունները, ինչպես նաև ունի մի շարք առավելություններ [1-4]:
- Մոդիֆիկացված են ցանցային պահեստի անդամների կառավարման մեխանիզմները[6]:
- Ընտրված և մոդիֆիկացված են ցանցային պահեստի տվյալների պահպանման և փոխանցման մեխանիզմները [5]:
- Իրականացված է ծրագրային ապահովում թույլ տվող կազմակերպել բազմանդամ հիերարխիկ ցանցային պահեստ, որը բաղկացած է սերվերի մոդուլից, կլիենտի մոդուլից և իրենց փոխազդեցության ծրագրային ինտերֆեյսից [ներդրման ակտ]:

Աշխատանքի արդյունքների ներդրումը

Աշխատանքում մշակված ցանցային պահեստի կազմակերպման ծրագրային համակարգը ներդրվել և օգտագործվում է «ԼԻԱ-Կ ԳՐՈՒՊ» ընկերությունում: Արդյունքում փոփոխվել են այդ ընկերության որոշ տվյալների պահման և փոխանցման մեթոդները:

Multi-user hierarchical remote data management system design and software implementation

In the XXI century communication channels rapidly develop, in particular, their capacities have increased even at long distances. One application of the new features was the creation of software that realize remote information storage.

There are some types of software that let you organize remote data management. Services and software applications, called cloud storage actively demonstrate growth today. This type of software can perform remote data access from different devices, which can be located in different geographical areas, it allows to enable synchronization mechanisms, hardware failsafe mechanisms are also supported, capabilities of access to data older versions(archive) may be implemented, as well as an option to allow users to share data with each other.

At this moment, there are some tasks for which you should modify the implementation of cloud storage. One of the tasks is a problem of strict data access control according to the hierarchy among users. Another problem is the presence of cloud storage restrictions on the deployment and software audition. There are also possibilities to improve the level of information security related to procedures like data storage, transmission, as well as mechanisms to ensure privacy.

The aim of the thesis is the development and implementation of software for organizing multi-user cloud storage, taking into account the hierarchy among users with more capabilities for deploying and auditing, as well as with higher level of information security.

Scientific novelty

- A new access control model, adapted to the hierarchical organization of multi-user cloud storage, was developed, which has a number of advantages and has been tested for compliance with the requirements of a role based model standard RBAC (Role Based Access Control), developed by NIST Institute (National Institute of Standards and Technology).
- The modified user accounts control mechanisms were for presented access control model mechanisms automate some of the management.
- The modified methods of data storage and transmission in cloud storage to improve security level.

Applicability of the results

The developed system can be used for the organization of new cloud storages. In addition, this system can also be used as an intermediate element to work with already existing cloud storage, while allowing for tiered hierarchy, more opportunities for audition, as well as increase the level of information security.

The following statements are presented for defense:

- a new data access control model created to organize multi-user cloud storage, which takes into account the hierarchical relationships among users , as well as has a number of additional advantages;
- modified mechanisms of cloud storage user accounts management ;

- modified cloud storage mechanisms of data savage and transmitting ;
- software that implements a hierarchical multi-user cloud storage, consisting of a server module, client module and software interface of their interaction.

The main results of the thesis are:

1. A new data access control model created to organize multi-user cloud storage, which takes into account the hierarchical relationships among users , as well as has a number of additional advantages was developed [1-4].
2. Modified cloud storage user accounts management mechanisms are presented [6].
3. Cloud storage mechanisms of data savage and transmitting were modified [5].
4. Software that implements a hierarchical multi-user cloud storage, consisting of a server module, client module and software interface of their interaction was developed [Act on the implementation].

System implementation:

The developed system can be used for the organization of new cloud storages. In addition, this system can also be used as an intermediate element to work with already existing network storage, while allowing for tiered hierarchy, more opportunities for audition, as well as increase the level of information security.

Ծավալը՝ 21 էջ: Տպաքանակը՝ 100:
ՀՀ ԳԱԱ ԻԱՊԻ կոմպյուտերային պոլիգրաֆիայի լաբորատորիա:
Երևան, Պ. Սևակի 1