УДК 004.056.5

АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ДОСТУПОМ, ОСНОВАННОЙ НА ИЕРАРХИЧЕСКИХ КЛЮЧАХ ШИФРОВАНИЯ

Сидоров Дмитрий Петрович
ФГБОУ ВПО «Мордовский государственный университет им. Н.П.Огарева»
E-mail: sidorovd@mail.ru

Аннотация. В большинстве случаев задача управления доступом к информационным ресурсам решается с помощью разграничения доступа на уровне файловой системы. Однако, такой способ имеет множество недостатков как в плане обеспечения безопасности информации, так и в плане удобства его практического использования. В статье рассматривается архитектура системы управления доступом к информации, основанной на применении иерархических ключей шифрования.
Ключевые слова: архитектура; система управления доступом; асимметричный алгоритм; шифрование; ключ; иерархическая информация.

Сегодня информация – это важнейшая часть активов любой организации. Эффективный контроль над этими активами, их защита от несанкционированного доступа, хищения и любого иного не предусмотренного регламентом использования приобретает для организаций одно из первостепенных значений. С массовым внедрением компьютеров во все сферы деятельности человека объем информации, хранимой в электронном виде, вырос в тысячи раз. И теперь скопировать за полминуты и унести дискету с файлом, содержащим необходимую информацию, намного проще, чем копировать или переписывать кипу бумаг. А с появлением компьютерных сетей даже отсутствие физического доступа к компьютеру перестало быть гарантией сохранности информации.

В большинстве компьютерных систем вся обрабатываемая информация разделяется в зависимости от своей важности на так называемые «классы безопасности», которые образуют иерархию. Причем эта иерархия, как правило, является древовидной. Кроме того, характерной является ситуация, когда информация внутри каждого класса безопасности дополнительно упорядочена по временным интервалам.

Чаще всего задача управления доступом к информационным ресурсам решается с помощью разграничения доступа на уровне файловой системы. Однако, такой способ имеет множество недостатков как в плане обеспечения безопасности информации, так и в плане удобства его практического использования. Более надежным способом является шифрование информации. На рынке программного обеспечения существует большое количество продуктов, обеспечивающих шифрование данных. Однако, они мало пригодны для управления доступом к информационным ресурсам, имеющим иерархическую структуру. Применительно к иерархической информации данный подход означает, что необходимо использовать иерархические ключи шифрования. Метод управления доступом на основе ключей шифрования и алгоритм генерации иерархических ключей шифрования, зависящих от времени, подробно рассматривался в [1].

Одной из самых распространенных на сегодняшний день моделей построения распределенных вычислительных систем является технология клиент/сервер. При построении системы управления доступом к информационным ресурсам желательно придерживаться следующих правил:

1. поддержка открытых стандартов, подразумевающая соответствие общепринятым стандартам;
2. масштабируемость – программное обеспечение должно работать с приемлемой произво-

дительностью без внесения в него существенных изменений при увеличении мощности и количества используемого оборудования;

3. многозвенность – каждый уровень системы (клиент, Web-сервер, сервер приложений) отвечает и реализует функции, наиболее присущие ему;
4. аппаратно-платформенная независимость программного обеспечения, используемого при разработке системы.

Исходя из этих требований, наиболее удачным представляется подход, основанный на архитектуре клиент/сервер и Web-технологии. При этом предлагается использовать следующую структуру системы управления доступом, представляющую собой многозвенную архитектуру и состоящую из следующих уровней (рис. 1):

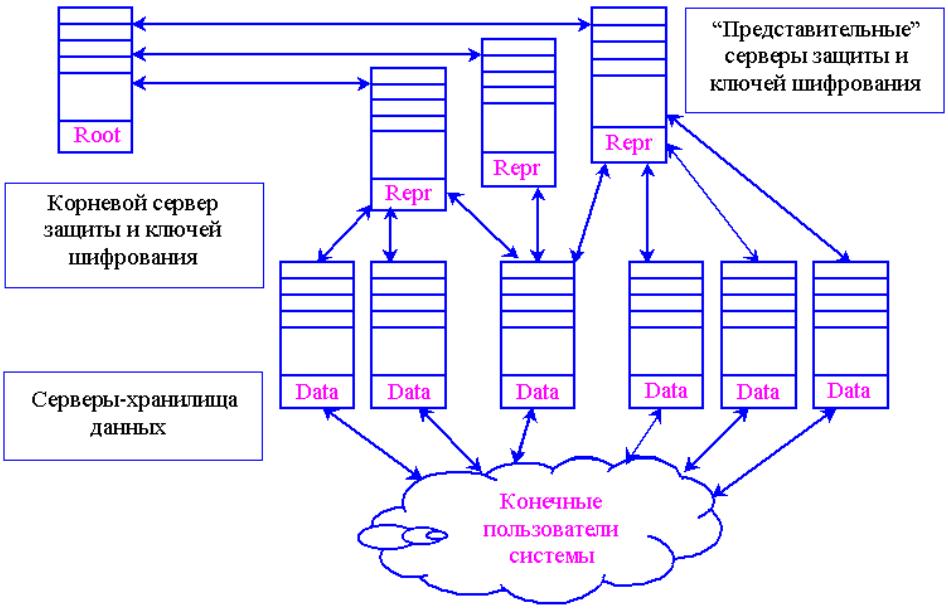

Рис. 1. Структура серверов системы управления доступом

1. Клиентский уровень. В него входит терминальный компьютер пользователя под управлением операционной системы Windows 9x/ME/NT/2000/XP с установленным на нем специально разработанным клиентским приложением для доступа к данным.
2. Уровень серверов данных. Специальные серверы, основная задача которых, как следует из названия, непосредственное хранение информации и предоставление к ней доступа.
3. Уровень представительных серверов. Выполняют функции серверов авторизации для хранилищ данных и при необходимости выдают ключи конечным пользователям.
4. Уровень корневого сервера. «Ядро» системы управления доступом. Хранит базу данных пользователей, все параметры системы безопасности, а также производит аутентификацию пользователей.

Обмен данными в приведенной системе состоит из следующих операций: аутентификации, сохранения, получения и удаления данных.

Операция аутентификации является самой ресурсоемкой. Она производится в несколько этапов, при этом задействуются все уровни взаимодействия, включая корневой сервер. Сначала клиент посылает серверу данных запрос на аутентификацию. Сервер данных принимает запрос аутентификации и отвечает подтверждением принятия данных, не разрывая соединения, пересылает данные представительному серверу. Представительный сервер проверяет и пересылает запрос на корневой сервер. Серверу данных при этом выдается подтверждение о пересылке, клиент также получает подтверждение от сервера данных. Образуется канал «клиент – сервер данных – представительный сервер – корневой сервер». Корневой сервер проверяет учетные данные пользователя, затем посылает ответ. В случае отказа аутентификации по каналу посылается сообщение об отказе и канал разрывается. В случае успешной аутентификации по каналу посылается сообщение из трех частей. Одна часть дан-

ных передается непосредственно пользователю. Она содержит временный ключ, который будет использоваться для шифрования передаваемых данных в данном сеансе обмена (сессии). Кроме того, пользователь получает от сервера данных идентификатор сессии. В одну сессию может производиться любое количество операций обмена.

Получение данных начинается с необязательной аутентификации. После этого клиент посылает запрос серверу данных. В ответ возвращаются требуемые данные, минимальный класс доступа и дополнительная информация о сохранившем данные пользователе. Кроме того, сервер данных может отправить запрос на добавление записи о доступе к данным в журнал представительному серверу.

Сохранение данных начинается с их зашифрования. Пользователь указывает, для какого минимального класса он хочет их зашифровать. Затем производится аутентификация и подается запрос на сохранение данных. Серверу передаются сведения о минимально необходимом для расшифрования данных классе, производится авторизация пользователя на доступ к сохраняемым данным (пользователь не имеет возможности изменять данные других пользователей). В случае успешной авторизации сохраненные данные помечаются информацией о пользователе, а представительному серверу посылается запрос на добавление сведений об изменении данных в журнал.

Удаление данных производится аналогично сохранению, за исключением того, что передается только запрос на удаление. Удаляемые (или заменяемые при сохранении) данные могут перемещаться сервером данных в архив, если таковой ведется. Следует отметить, что архив не является доступным извне и используется в исключительных случаях для восстановления важных данных.

Смысл такой схемы в том, чтобы, во-первых, защитить корневой сервер от прямых атак (предполагается, что сетевой сегмент между представительным и корневым сервером изолирован от более крупных сетей и безопасен), а во-вторых – снять с корневого сервера нагрузку по дополнительной проверке запросов (за исключением проверки корректности). Таким образом, если представительный сервер подвергнется атаке со стороны внешней сети, безопасность корневого сервера не будет нарушена. Это позволяет быть уверенным в том, что атака на представительный сервер не приведет к какой либо утере/утечке ключей и прочих параметров системы безопасности. Даже если атака вызовет отказ представительного сервера, это приведет лишь к разрыву связи между пользователями данного представительного сервера и корневым сервером, то есть к отказу в обслуживании. Но в общем случае пользователи могут воспользоваться любым представительным сервером. Также такая схема позволяет уменьшить требования к оборудованию. Каналы связи между корневым сервером и представительными серверами могут быть весьма быстродействующими за счет их малой протяженности. Наличие нескольких представительных серверов позволяет распределить нагрузку по обработке пользователей между ними. Кроме того, в приведенной схеме возможен вариант с организацией кластера безопасных корневых серверов, что позволяет еще больше распределить нагрузку по обслуживанию пользователей.

Данная архитектура позволяет скрыть от конечного пользователя внутреннюю структуру системы и максимально оптимизировать загрузку вычислительных средств. Более подробное описание предлагаемой архитектуры рассматривалось в [2].

Вероятные атаки на сервер данных приведут, максимум, к потере данных пользователей (самый маловероятный вариант) или к отказу сервера (также маловероятно). Отказ сервера не является фатальным, поскольку работоспособность сервера в любом случае будет восстановлена за минимально необходимое для этого время. Основные последствия удавшихся атак – временные неудобства для пользователей.

Представительный сервер является наименее необходимым в плане функциональности, но наиболее необходимым в плане обеспечения безопасности звеном системы. Он способен принять «на себя» большинство атак, причем эти атаки не будут представлять опасности для системы в целом. Даже в случае отказа любого представительного сервера его функции возьмут на себя другие представительные сервера, а его функциональность может быть

восстановлена в кратчайшие сроки. Наибольшие потери от атак – последние файлы журнала, которые очень большой ценности в обычных рабочих условиях не представляют. Кроме того, при надлежащей архивации и этих потерь можно избежать.

Атаки на корневой сервер являются самыми опасными атаками, могущими привести к его отказу и компрометации всей системы безопасности. Избежать их можно, принимая надлежащие меры по изоляции и обеспечению безопасности данного сервера и осуществляя над ним постоянный контроль. Также не следует открыто хранить файлы параметров иерархии, не рекомендуется сохранять их и в файловой системе корневого сервера. Коммерчески может быть оправдано хранение этих параметров на карте Flash-памяти, физически носимой с собой администратором безопасности системы.

Для реализации предложенной структуры серверов системы был выбран следующий подход. Все серверы являются серверами Web, поэтому весь обмен данными построен на протоколе HTTP. Протокол HTTP имеет много расширений, таких как HTTPS, что дает возможность реализовать на базе специализированного Web- или proxy-сервера защиту канала передачи данных. Кроме того, использование Web-сервера подразумевает абсолютную переносимость системы на любые программные/аппаратные платформы, поддерживающие HTTP. Разработанный программный комплекс состоит из трех серверных и двух клиентских приложений.

К клиентским приложениям относятся, собственно, клиент доступа к данным и административное приложение для первоначального задания иерархии пользователей и дальнейшего управления ключевой подсистемой (продления ключей, добавления и удаления пользовательских учетных данных и т. п.). Эти компоненты были реализованы в среде Borland Delphi 7 для платформа Win32.

К серверным приложениям относятся компоненты для корневых серверов, представительных серверов и серверов-хранилищ данных. Для реализации данных компонент был выбран Web-язык PHP версии 4. Данный Web-язык без особых проблем устанавливается на любой совместимый Web-сервер (Apache, fhttpd, IIS/PWS, Netscape, OmniHTTPd), он достаточно производителен, безопасен и не создает сложностей в настройке при корпоративном использовании.

Применение системы управления доступом к информации на базе иерархии пользователей обширно. Такая система вполне может применяться в любых организациях/проектах, где есть иерархическое деление пользователей. Но разработанная система имеет еще одно преимущество: возможность разделения доступа к данным по времени. Это дает еще одну область применения: в системах, где доступ пользователей к данным должен быть ограничен определенным сроком. Это электронные подписки, медиа-службы, периодические электронные издания, поддержка обновления программного продукта на определенный срок.

Литература

1. Федосин С.А., Сидоров Д.П. Криптографическое управление доступом к иерархической информации // Информационные технологии в электротехнике и электроэнергетике: Материалы IV Всерос. науч.-техн. конф. Чебоксары: изд-во Чуваш. ун-та, 2002. − С. 333-338.
2. Сидоров, Д.П. Защита информационных ресурсов системы дистанционного обучения / Д.П. Сидоров, А.Ю. Асемов, С.А. Федосин // Инженерное образование. − 2003. − вып. 1. − С. 66 − 69.

ARCHITECTURE OF ACCESS CONTROL SYSTEM BASED ON HIERARCHICAL ENCRYPTION KEYS

Sidorov Dmitry Petrovich

Abstract. It is common to control access of the users to informational resources using built-in services provided by the file systems. However, this approach has major disadvantages both in maintaining data security and ease of practical use. This article presents architecture of an access control system based on hierarchical encryption keys.

Keywords. Architecture; access control system; asymmetric encryption algorithm; encryption, encryption key; hierarchical data.