Защита информационных ресурсов системы дистанционного обучения

Мордовский государственный университет им. Н.П. Огарева

ABSTRACT: Information resources of remote training system are considered a structure with hierarchical access scheme. There was developed the open and secret keys generation algorithm, based on hierarchical classes and subclasses, and also on a time interval. The infocommunication structure is suggested along with two protocols of information interchange.

Д.П. Сидоров,
А.Ю. Асемов,
С.А. Федосин

С массовым внедрением компьютеров во все сферы деятельности человека объем информации, хранимой в электронном виде, вырос в тысячи раз. В настоящее время практически любая организация имеет в своем распоряжении несколько компьютеров, которые, как правило, объединены в локальную сеть. В таких сетях существуют разделяемые информационные ресурсы, к которым имеют доступ все пользователи. Однако, на практике, информация обычно разделяется на классы, в зависимости от своей важности. При этом, необходимо контролировать доступ пользователей к информации в соответствии с их полномочиями.

В последние годы значительное развитие получили системы дистанционного образования. Эти системы позволяют оперативно получать теоретическую информацию, выполнять практические задания с поддержкой преподавателя, возможно, находящегося за несколько тысяч километров, присутствовать на Internet-лекциях и семинарах и т.п.

Курсы дистанционного образования чаще всего являются платными. Это порождает необходимость защиты информации от несанкционированного доступа. Такая защита может выполняться и на уровне аутентификации авторизации, несмотря на сомнительность этого метода. Конечно же, более надежной защитой является шифрование информации, но и оно не предотвратит возможной утечки.

Кроме того, в системах дистанционного образования существует четкая иерархия пользователей. Администратор системы следит за всей системой дистанционного образования и обычно имеет доступ ко всем данным. У преподавателей ведущих курс в системе должна быть возможность создавать и модифицировать учебные курсы, у преподавателей получать и представлять учебные курсы (возможно, не все), частично их модифицировать и создавать учебные программы и курсы лекций. У тех, кто ведет практическую часть должен быть доступ к курсу лекций (на чтение, а возможно и частично на изменение) и возможность создавать практические задания, которые могут быть скорректированы преподавателями и ведущими преподавателями. Обучаемые же должны иметь доступ только к получению информации и к своим личным данным отчетам, контрольным и т.п. Такая схема порождает достаточно сложную структуру, сильно упрощенная схема которой приводится ниже (рис.1):

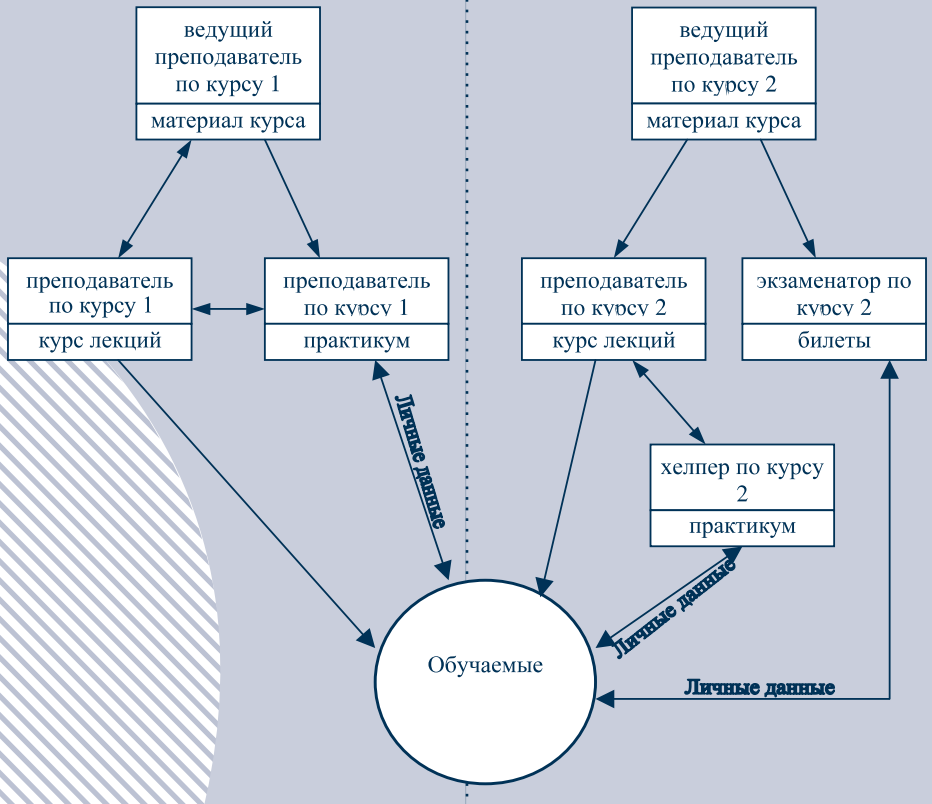

Рис.1. Упрощенная схема пользователей системы дистанционного образования.

В данном случае обычным шифрованием данных не обойтись многим пользователям придется знать ключи многих других пользователей. Конечно, при этом неизбежна «утечка» ключей, путаница с большим их количеством и, как следствие, нагрузка на администратора и несанкционированный доступ к информации. Вот тут на помощь и приходит иерархическая система защиты, которая базируется, соответственно, на иерархической схеме шифрования.

В общем случае иерархия классов имеет вид дерева, например, как на рис.2. Рассмотрим задачу управления доступом к иерархической информации.

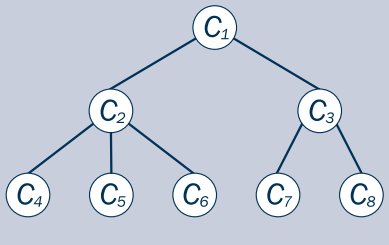

Рис. 2. Пример иерархии классов

Пусть c_i, $i=\overline{1,m}$ - классы информации. Предположим, что они упорядочены при помощи некоторого бинарного отношения " ". При этом очевидным является следующее требование: если пользователь имеет доступ в класс c_i, то он должен иметь доступ и в любой класс такой, что c_j c_i.

Для защиты информации от несанкционированного доступа, обычно применяется шифрование, то есть для каждого класса c_i назначается ключ κ_i и вся информация в этом классе шифруется при помощи данного ключа. Таким образом, пользователь, получая доступ к классу c_i вместе с ключом κ_i должен хранить и ключи от всех нижестоящих классов. Это весьма неудобно.

Возникает проблема назначения криптографического ключа κ_i для класса c_i, таким образом, чтобы мы могли использовать κ_i для вычисления κ_j, но только для классов, удовлетворяющих условию c_j c_i. В такой системе каждый пользователь приписывается к некоторому классу c_i в соответствии со своим статусом. Затем, получив ключ , он может расшифровать данные в классе . Более того, он

может расшифровать данные в классе c_j, но только в том случае если c_j c_i.

Рассмотрим ситуацию, когда информация внутри каждого класса дополнительно разделяется на части по времени ее занесения в этот класс. Для того, чтобы пользователю не хранить у себя ключи для всех t $[t_1,t_2]$, удобным является следующий подход. Пользователь получает не сами ключи $\kappa_{i,t}$, t $[t_1,t_2]$, а некоторую вспомогательную ключевую информацию $I(i,t_1,t_2)$. С ее помощью он может вычислить любой ключ $\kappa_{i,t}$, но только для c_j c_i и t_1 t t_2.

Подробное описание алгоритма рассматривалось в [1, 2]. Здесь лишь укажем, что он использует криптосистему RSA и его безопасность основывается на криптостойкости последней.

Конечно, схема шифрования это важнейшая часть системы защиты. Но при реализации системы защиты в компьютерной сети появляется ряд факторов, которые заставляют разрабатывать достаточно сложные схемы серверов защиты. Упрощенная структура иерархической системы безопасности в компьютерной сети такова (рис.3).

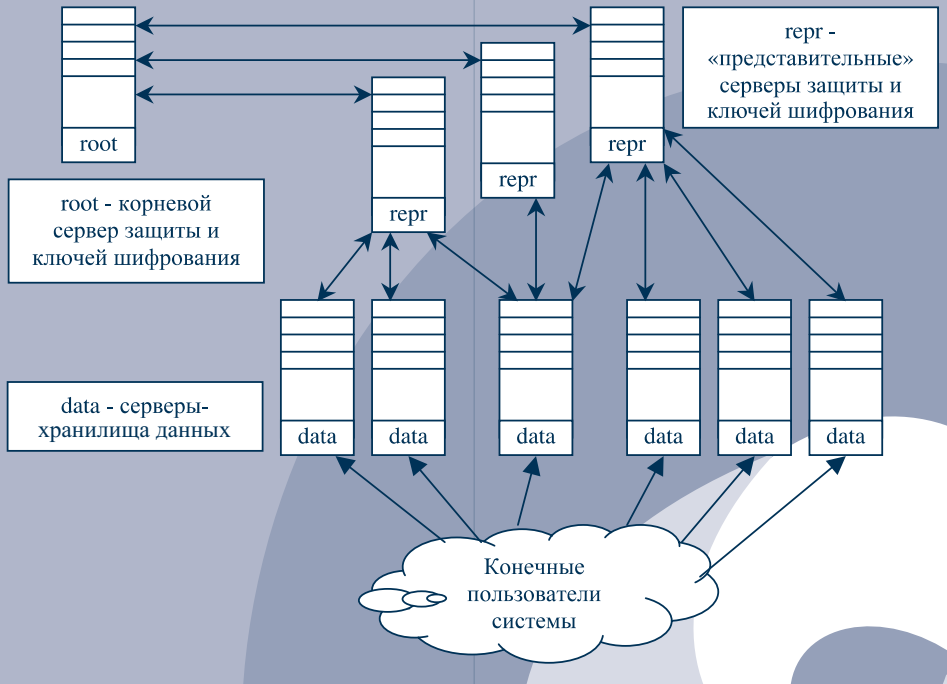

Рис.3. Структура серверов иерархической системы безопасности.

Основные преимущества иерархической системы безопасности заключаются в следующем.

Во-первых, корневой сервер безопасности является полностью изолированным, доступ к нему имеют лишь т.н. «представительные» серверы.

Представительные серверы также выполняют функции серверов авторизации для хранилищ данных и при необходимости выдают ключи конечным пользователям. Это позволяет быть уверенным в том, что атака на представительный сервер не приведет к какой либо утере/утечке ключей и прочих параметров системы безопасности, а также к останову корневого сервера (который, как уже было сказано, является физически изолированным). Также, в случае отказа любого из представительных серверов, другие представительные серверы могут взять на себя его функции. Такое положение вещей позволяет выполнять балансировку нагрузки на представительные серверы, поскольку они являются равноправными.

Во-вторых, представительные серверы выполняют лишь транспортную функцию. Расшифровка ими данных, идущих к хранилищу или конкретному пользователю невозможна (ключ каждого представительного сервера находится в младшем по отношению ко всем остальным или не связанном с остальной иерархией классе безопасности). Однако, представительный сервер также всегда получает некоторые контрольные данные, дабы предотвратить подделку (спуфинг) сетевого трафика.

В-третьих, хранилища данных являются хранилищами в прямом смысле слова. Несмотря на их активное взаимодействие с представительными серверами, данные, хранимые на них, не зависят от системы иерархии классов безопасности. Сервер-хранилище данных попросту «не знает» о существовании какой-либо иерархии, он выполняет аутентификацию/авторизацию пользователя и выдает ему данные, зашифрованные полученным от представительного сервера временным ключом.

В четвертых, данные могут распространяться по электронной почте, в виде файлов на дискетах и т.п. В этих случаях не нужен ни один специализированный сетевой сервер. Для рассылки множеству пользователей может использоваться один и тот же пакет данных - каждый пользователь сможет расшифровать и

прочитать только предназначенную ему и более низшим классам часть пакета.

Если сравнивать иерархическую систему безопасности с широко известными системами, такими, как шифрованная файловая система EFS, можно отметить следующее.

В файловой системе EFS восстановление данных в случае утери ключа пользователя невозможна. Это объясняется тем, что ключи пользователей не вычисляются из ключей «старших» пользователей, а просто хранятся в базе данных безопасности. EFS также является не иерархической, а просто разделенной по доступу системой т.е. классов пользователей в ней не существует. В EFS есть агенты восстановления, которые могут читать любые данные, но сущность такой схемы сводится к тому, что вместе с каждым файлом хранится ключ зашифровавшего его пользователя, зашифрованный общем ключом агентов восстановления. В случае утери базы данных безопасности (и, как следствие, ключа агентов восстановления) восстановление данных становится невозможным.

В иерархической системе безопасности восстановление данных при утере секретных ключей возможно с помощью личного ключа и открытых данных администратора или старших пользователей. Из этих ключей легко вычисляются все ключи для расшифровки данных более младших пользователей, поэтому всегда есть возможность расшифровать все имеющиеся данные и перешифровать их новыми ключами после восстановления иерархии системы безопасности. В системах дистанционного образования для расшифрования данных потребуются ключи всех преподавателей, ведущих курс, т.к. администратор не является «старшим» пользователем системы.

Рассмотрим программное обеспечение всех узлов иерархической системы защиты и взаимодействие между этими узлами. Система получила название HSS, или Hierarchical Security System (иерархическая система безопасности). Термин HSS здесь и далее объединяет все, что относится к самой системе.

Корневой сервер является, по сути, ядром системы. На этапе развертывания на нем с помощью редактора иерархии классов пользователей создается сама структура иерархической системы. Затем запускается служба корневого сервера безопасности, которая отвечает на запросы представительных (и только представительных) серверов. Изоляция корневого сервера от внешней сети позволяет быть уверенным в отсутствии «поддельных» запросов, но все же для передачи данных между корневым и представительным сервером используется шифрование ключом класса данного представительного сервера.

Для передачи данных между корневым сервером и серверами-хранилищами, а также конечными пользователями используется шифрование ключами этих серверов/пользователей, что не дает возможности оператору представительного сервера прочесть данные. В HSS есть возможность использования стандартных протоколов наряду со специально разработанными, что дает системе еще один плюс возможность развертывания практически в любой совместимой с HTTP и/или XML среде.

Представительные серверы берут на себя основную нагрузку. Они выполняют функции «посредника» при обмене серверов «внешнего мира» с корневым сервером безопасности. Получив, к примеру, запрос на аутентификацию пользователя от сервера-хранилища данных, представительный сервер направляет данный запрос на корневой сервер безопасности, а затем отправляет полученный зашифрованный результат запросившему аутентификацию серверу.

Серверы-хранилища данных являются просто серверами, предоставляющими функции доступа к данным конечных пользователей. Для серверов-хранилищ используется аутентификация пользователя с помощью запроса к представительному серверу. Передаваемые пользователю данные шифруются выданным ему временным ключом. Серверы-хранилища используют HTTP для общения с пользователем.

Пользователь может как запрашивать данные, так и размещать их на сервере. Права пользователя на запрос/размещение данных определяются средствами самого сервера. Система HSS подразумевает, что пользователь может получать любые данные, но при сохранении данных могут действовать ограничения класса. Другие пользователи должны быть проинформированы о классе оставившего данные пользователя, чтобы не выполнять ненужных запросов на данные, им недоступные.

В системах дистанционного образования хранилища данных содержат материалы курсов, практические задания, их решения (преподавательские и пользовательские), личные данные пользователей, сведения о сдаче экзаменов и т.п. Разделение доступа и четкий учет классов и прав пользователей, а также общий доступ к данным ключевые моменты работоспособности данной системы. HSS является достаточно удобным средством для обеспечения защиты данных в подобных системах с большим количеством пользователей и общими данными.

Литература

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.

Федосин С. А., Сидоров Д. П. Назначение криптографических ключей, зависящих от времени, для иерархии классов. Информационные технологии и системы в образовании, науке, бизнесе: Сборник материалов III Международной научно-технической конференции. Пенза, 2002. С. 103-105.