Алферов И. Л.
Московский инженерно-физический институт (государственный университет)

АУТЕНТИФИКАЦИЯ КЛИЕНТОВ ПРИ ДОСТУПЕ К ИНФОРМАЦИИ В ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ СЕТЯХ ХРАНЕНИЯ ДАННЫХ

Рассматривается задача аутентификации в объектно-ориентированных сетях хранения данных. Предложена модель решения этой задачи, новый протокол аутентификации и доказана его корректность.

Введение

Потребность современных информационных центров в хранении больших объемов данных в режиме непрерывной доступности привела к появлению двух наиболее распространенных архитектур — сетей хранения данных (СХД, англ. Storage Area Network) и сетевых хранилищ данных (Network Attached Storage). Обе архитектуры предназначены для обеспечения эффективного доступа к данным со стороны клиентских узлов сети, но каждая обладает своими собственными недостатками, сужающими область ее возможного практического применения. В частности, использование блочного интерфейса работы с данными не позволяет реализовать эффективную модель безопасности, так как управление доступом на уровне блоков требует чрезмерных затрат ресурсов, а на уровне логических разделов или устройств хранения данных (УХД) — не обладает достаточной гибкостью [1].

Для преодоления недостатков совместными усилиями представителей различных производителей (HP, IBM, Intel, Seagate и др.) в OSD Technical Workgroup в составе SNIA (Storage Networking Industry Association) относительно недавно была разработана архитектура объектного хранения данных (Object Storage Architecture). Ключевая концепция, лежащая в основе этой архитектуры, заключается в хранении совокупности данных и метаданных в виде объектов — абстрактных сущностей, которые могут представлять собой файл, запись базы данных или целую файловую систему. При этом функции управления дисковым пространством, синхронизации доступа и интеллектуального кэширования выполняются УХД в зависимости от типов и атрибутов хранимых объектов. Одним из важнейших преимуществ данного подхода является возможность организации масштабируемого разделяемого доступа к информации с контролем на уровне объектов.

Важным этапом развития технологии объектного хранения данных стало принятие стандарта ANSI INCITS 400-2004 [2], определяющего расширения интерфейса SCSI для взаимодействия с объектными УХД (Object Storage Device, OSD). Работа [3] дополняет стандарт детальным описанием протокола аутентификации клиентов и обоснованиями его разработки.

В данной статье отмечены существенные недостатки предложенного в стандарте подхода к аутентификации и предложен альтернативный протокол аутентификации, позволяющий их преодолеть.

1. Модель безопасности объектных СХД

Предложенная в стандарте [2] модель безопасности для объектных СХД основана на классической мандатной схеме. В реализации механизмов безопасности задействованы следующие компоненты сети:

· объектные УХД;

· менеджер политики безопасности (policy manager);

· менеджер безопасности (security manager);

· клиентские приложения.

Менеджер политики безопасности выполняет две основные функции:

· координирует ограничения доступа между УХД и клиентами, подготавливая соответствующие политике безопасности мандаты (capability), которые впоследствии используются клиентами как основание для получения доступа к объектам и командным функциям на устройствах хранения;

· согласованно с УХД предотвращает небезопасное или временно нежелательное использование пространства на физических носителях.

Главной функцией менеджера безопасности является подготовка удостоверений (credential) в ответ на запросы клиентов. Удостоверение представляет собой структуру данных, содержащую подготовленный менеджером политики безопасности мандат и криптографические данные для проверки его аутентичности. Вместе с удостоверениями менеджер безопасности отправляет клиентам мандатные ключи (capability key), выработанные на основе общих секретных ключей УХД и менеджера безопасности, устанавливаемых OSD-командами SET KEY и SET MASTER KEY. Удостоверение дает клиенту возможность доступа к определенным ресурсам СХД, а мандатный ключ позволяет проверять аутентичность команд и данных, которыми обмениваются УХД и клиенты.

Протокол обмена между клиентом и менеджером безопасности в стандарте не определяется — описан только формат возвращаемого удостоверения. Использование безопасного взаимодействия клиентов с менеджером безопасности является критичным для обеспечения безопасности сети в целом. Отсутствие описания протокола получения удостоверений в стандарте свидетельствует о неполноте используемой схемы аутентификации.

УХД для каждого полученного от клиента запроса проверяет, что:

· удостоверение не модифицировано (т. е. оно сгенерировано менеджером безопасности и содержит код проверки целостности, выработанный с помощью секретного ключа, известного только менеджеру безопасности и данному УХД);

· удостоверение было законно получено клиентом от менеджера безопасности или от другого клиента (т. е. клиент знает соответствующий удостоверению мандатный ключ и использует его для получения корректного кода проверки целостности или значений команд);

· запрошенная команда разрешена мандатом, содержащимся в удостоверении.

Для выполнения команд клиент запрашивает удостоверения и мандатные ключи у менеджера безопасности, после чего посылает удостоверения УХД как составную часть запросов команд, содержащих код проверки целостности, полученный с помощью мандатного ключа. Формальное доказательство криптографической стойкости описанного протокола до настоящего времени не производилось. Взаимодействие компонентов объектной СХД в рамках стандартной модели безопасности продемонстрировано на рис. 1.

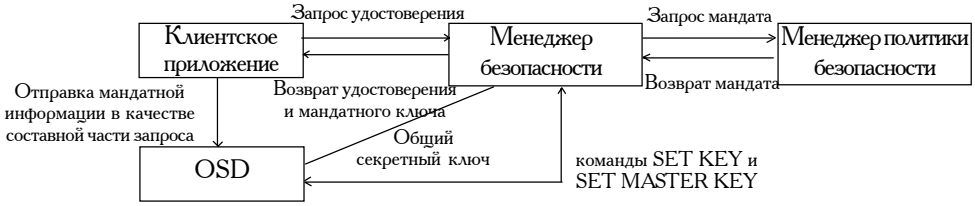

Рис. 1. Взаимодействие устройств сети в стандартной модели безопасности

В модели не предусматриваются механизмы обеспечения конфиденциальности данных, передаваемых между клиентом и УХД. Несмотря на то, что в некоторых случаях имеется возможность использования транспортного протокола SCSI для защиты конфиденциальности трафика (например, с помощью протоколов безопасности Fibre Channel [5]), отсутствие данного механизма в базовом командном протоколе OSD означает невозможность фактической реализации политики разграничения доступа клиентов к объектам в большинстве случаев. Описанная схема предусматривает только мандатные ключи, используемые для проверки аутентичности передаваемых данных.

Одним из достоинств стандартной модели безопасности является централизованное хранение политики безопасности, упрощающее процедуры ее сопровождения и снижающее сложность соответствующих устройств. Однако централизация достигается за счет применения удостоверений, с помощью которых клиенты получают доступ к ресурсам сети. Так как все удостоверения выдает менеджер безопасности, при достаточно большом количестве хранимых в сети объектов и работающих с ними клиентов, производительность менеджера может стать недостаточной. При этом менеджер безопасности также становится единой точкой отказа и приоритетной мишенью для сетевых атак.

Необходимость получения у менеджера безопасности реквизитов доступа при каждом обращении клиента к новому объекту, хранимому в СХД, предполагает дискреционную организацию политики безопасности. Однако последнее время все большую популярность получает ролевая модель управления доступом [6]. Исследования [7, 8] показали, что она имеет высокую экономическую эффективность и отвечает современным требованиям коммерческого и государственного секторов. Данная модель упрощает процесс администрирования системы управления доступом и является более гибкой, чем традиционные дискреционные и мандатные модели. Ниже будет показано, что в объектных СХД ролевая модель управления доступом позволяет не только упростить сопровождение политики безопасности, но и существенно снизить нагрузку на менеджера безопасности.

Таким образом, выявляются следующие основные недостатки стандартной схемы аутентификации клиентов в объектной СХД: отсутствие фиксированного механизма получения удостоверений от менеджера безопасности; отсутствие формального доказательства криптографической стойкости протокола аутентификации; отсутствие поддержки защищенных соединений между клиентами и УХД для передачи данных с обеспечением конфиденциальности и целостности; потребность в частых взаимодействиях между клиентами и менеджером безопасности для получения удостоверений, приводящая к наличию в СХД единой точки отказа; ориентация на дискреционную модель управления доступом, требующую повышенных затрат ресурсов.

2. Протокол аутентификации

Определим список требований к протоколу аутентификации, который позволил бы устранить представленные недостатки: осуществление взаимной аутентификации между клиентами и УХД; поддержка механизмов установления защищенных соединений, обеспечивающих защиту информации, передаваемой между клиентами и УХД; поддержка ролевой модели управления доступом; устойчивость к различным сетевым атакам, в том числе к атакам методами повтора сеанса, включения в канал, деперсонификации, отражения, атакам по выбранному тексту и атакам методом форсированной задержки; отсутствие единой точки отказа в системе и соответствующее распределение функций безопасности между менеджером безопасности и УХД; минимизация количества обменов между клиентом и менеджером безопасности, позволяющая снизить количество передаваемого по сети служебного трафика и уменьшить нагрузку на менеджера безопасности; относительно небольшое количество ключей доступа, используемых в системе; отсутствие избыточных операций и элементов в сообщениях протокола; формально доказанная стойкость протокола к атакам противника в условиях заданных предположений относительно среды применения.

Общая последовательность взаимодействия клиента с менеджером безопасности и УХД в предлагаемом протоколе по сравнению со стандартным протоколом остается неизменной. Рассмотрим различные виды ключей доступа.

· Мандатный ключ. Мандатный ключ [2, 9] указывает права доступа клиента к некоторому объекту. Например, мандатный ключ может генерироваться по формуле $capKey = MAC_K(accessrights,O,expiry)$, где K — общий ключ менеджера безопасности и устройства, хранящего объект O, MAC — код аутентификации сообщений (например, HMAC [10]), $expiry$ — срок действия ключа. Мандатные ключи используются в стандартном протоколе аутентификации клиентов в объектных СХД.

· Личный ключ. Личный ключ [11, 12] позволяет УХД проверить личность конкретного клиента. Его вычисление может проводиться посредством формулы $idKey = MAC_K(identity,expiry)$, где K — общий ключ менеджера безопасности и устройства, $identity$ — идентификатор клиента. При использовании личных ключей устройствам необходимо хранить таблицы прав доступа совместно с каждым объектом.

· Ролевой ключ. Ролевой ключ [12] используется для аутентификации в системах с поддержкой ролевой модели управления доступом. Он позволяет УХД проверить, назначена ли клиенту запрошенная роль. Вычисление такого ключа может осуществляться как $roleKey = MAC_K(role,expiry)$, где K — общий ключ менеджера безопасности и устройства, либо личный ключ клиента, вычисленный согласно соотношению $idKey = MAC_K(id,H(roles),expiry)$, где $H(roles)$ — хэш-функция от конкатенации всех ролей, назначенных клиенту.

Для реализации ролевой модели управления доступом наилучшим образом подходит сочетание личных и ролевых ключей в одном протоколе аутентификации. При этом соответствие пользователей и назначенных им ролей сопровождается в базе данных менеджера безопасности, а соответствие ролей и назначенных им привилегий сопровождается на УХД в виде ролевых списков контроля доступа. Согласно исследованиям изменения в соответствии привилегий и ролей происходят намного реже, чем в соответствии ролей и пользователей, поэтому такое распределение функций управления политикой безопасности позволяет сократить как количество ключей доступа, используемых в системе, так и количество обменов, совершаемых в единицу времени между клиентами и менеджером безопасности.

Для обеспечения своевременного вступления в силу изменений в политике безопасности могут быть предусмотрены два метода аннулирования личных ключей, выданных менеджером безопасности: включение в личный ключ срока его действия; включение в личный ключ его версии и сопровождение списка аннулированных версий ключей, который менеджер безопасности распределяет по УХД.

Предлагается два варианта протокола аутентификации. Ниже приведены их спецификации. Они различаются количеством раундов обмена и требованием синхронизации системных часов участников. В зависимости от среды функционирования устройств объектной СХД может быть полезен тот или иной вариант.

1. Начальные условия и предположения. Менеджер безопасности является доверенным участником. Он отвечает за безопасную настройку системы, необходимые параметры безопасности, такие как ключи, а также их безопасное хранение. Менеджер имеет информацию о легитимных пользователях системы и располагает методами безопасной аутентификации и обмена с клиентами. В случае компрометации менеджера безопасности компрометируется вся СХД.

Клиенты или конечные пользователи являются недоверенными участниками. Они могут выполнять все виды активных и пассивных атак, например, деперсонификацию легитимных клиентов и попытки осуществления несанкционированного доступа. Коммуникационные соединения являются полностью открытыми. В связи с этим противник может выполнять активные и пассивные атаки, такие как прослушивание, маскарад, вставка и модификация данных и т. д. УХД являются доверенными участниками с точки зрения выполнения своих функций, то есть предоставляют доступ только легитимным пользователям системы.

2. Обозначения.

A — клиент СХД, B — УХД, T — менеджер безопасности, I_A — идентификатор клиента A, I_B — идентификатор УХД B, $idKey_{AB}$ — личный ключ клиента A для взаимодействия с устройством

B, $roleKey_{AB}$ — ролевой ключ клиента A для взаимодействия с устройством B при определенном наборе активных ролей, *roleList* – конкатенация всех назначенных клиенту A ролей, *activeRoleList* — конкатенация активируемых в сеансе ролей клиента A, L — срок действия и/или номер версии личного ключа клиента A, K_{AT} — общий секретный ключ менеджера безопасности и клиента A, K_{BT} — общий секретный ключ менеджера безопасности и устройства B, E — симметричный алгоритм шифрования, MAC — код аутентификации сообщений, H — хэш-функция с трудно обнаружимыми коллизиями, N_A, N_{A1}, N_{A2} — выбранные клиентом A случайные числа, N_B — выбранное УХД B случайное число, T_A — метка времени по системным часам клиента A, k_A — сгенерированный клиентом A сеансовый подключ, k_B — сгенерированный устройством B сеансовый подключ, k — сеансовый ключ защищенного соединения между A и B, f — функция комбинирования ключей от двух аргументов.

3. Предварительный этап протокола (исходная настройка). A и T разделяют ключ K_{AT}, B и T разделяют ключ K_{BT}. T хранит базу данных, определяющую соответствие клиентов и назначенных им ролей. A назначен список ролей *roleList*.

Определения:

$idKey_{AB} =_{def} MAC_{K_{BT}}(I_A, H(roleList), L)$, $roleKey_{AB} =_{def} MAC_{idKey_{AB}}(activeRole\ List)$, $k =_{def} f(k_A, k_B)$.

4. Рабочий этап протокола (сообщения).

Вариант I:

$$(1) A \rightarrow T : I_A, I_B, N_A$$
$$(2) A \leftarrow T : E_{K_{AT}}(N_A, idKey_{AB}, roleList, L, I_B)$$
$$(3) A \rightarrow B : I_A, roleList, L, activeRoleList, E_{roleKey_{AB}}(I_A, T_A, k_A)$$
$$(4) A \leftarrow B : E_{roleKey_{AB}}(T_A, k_B)$$

Вариант II:

$$(1)\ A \rightarrow T : I_A, I_B, N_{A_1}$$
$$(2)\ A \leftarrow T : E_{K_{AT}}(N_{A_1}, idKey_{AB}, roleList, L, I_B)$$
$$(3)\ A \leftarrow B : N_B$$
$$(4)\ A \rightarrow B : I_A, roleList, L, activeRole\ List, E_{roleKey_{AB}}(I_A, T_A, k_A)$$
$$(5)\ A \leftarrow B : E_{roleKey_{AB}}(N_{A_2}, k_B)$$

5. Действия, выполняемые участниками протокола.

Вариант I:

· A генерирует N_A и отправляет T сообщение (1).

· T осуществляет запрос к базе данных и определяет список доступных ролей для A. После этого T генерирует для A личный ключ $idKey_{AB}$ с заданным сроком действия/версией L, шифрует полученное значение N_A и другие данные на ключе K_{AT} и отправляет сообщение (2).

· A расшифровывает сообщение (2) и проверяет, что полученный идентификатор УХД и N_A совпадают с отправленными в сообщении (1). В случае если проверка неудачна, A пытается повторить протокол. Далее A выбирает необходимый начальный набор активных ролей *activeRoleList* для установки защищенного сеанса связи с B, вычисляет ролевой ключ $roleKey_{AB}$, генерирует сеансовый подключ k_A, формирует и посылает B сообщение (3).

· Получив сообщение (3), B проверяет, что полученное множество активизируемых ролей *activeRoleList* является подмножеством множества доступных ролей *roleList*, вычисляет хэш-функцию H от полученного списка *roleList* и, используя полученные в сообщении значения, вычисляет личный ключ $idKey_{AB}$ и ролевой ключ $roleKey_{AB}$. Используя ролевой ключ, B расшифровывает вторую часть сообщения (3), извлекает идентификатор I_A, метку времени T_A и сеансовый подключ k_A, проверяя,

что: идентификаторы I_A в обеих частях сообщения совпадают; метка времени T_A действительна, т. е. находится в пределах допустимого отклонения от системного времени B; системное время B находится в пределах срока действия личного ключа L, либо в списке аннулированных версий ключей отсутствует версия личного ключа L. Если проверка проходит, то B объявляет аутентификацию A успешной, иначе отказывает A в обслуживании.

· B вырабатывает сеансовый подключ k_B, конструирует и отправляет A сообщение (4), содержащее зашифрованные T_A и k_B. Используя подключи k_A и k_B, B вычисляет сеансовый ключ k, который используется для организации защищенного соединения с A.

· A расшифровывает сообщение (4). Если метка времени T_A совпадает с отправленной в сообщении (3), A объявляет аутентификацию B успешной и вычисляет сеансовый ключ k.

Вариант II:

Последовательность действий во втором варианте протокола отличается от первого тем, что обмен клиента A с УХД B начинается с запроса и получения от B сгенерированного им случайного числа N_B. Затем свежесть сообщений (4) и (5) гарантируется передаваемыми в них вместо метки времени T_A случайными числами N_B и N_{A2}.

6. Результаты: A и B взаимно аутентифицированы. A и B разделяют общий совместно вычисленный сеансовый ключ k. B доверяет списку допустимых ролей клиента *roleList* (т. е. аутентичным источником данного списка является менеджер T) в течение срока действия L, либо до момента помещения версии L личного ключа A в список аннулированных версий ключей. B доверяет списку активируемых ролей клиента *activeRoleList*, т. е. аутентичным источником списка является A, и все роли списка являются допустимыми, в течение срока действия L, либо до момента помещения версии L личного ключа A в список аннулированных версий ключей.

7. Примечания. Так как в варианте I используются метки времени, системные часы всех участников должны быть синхронизированы и защищены. Если в варианте II не используются поля сроков действия, то протокол может выполняться устройствами, у которых системные часы работают асинхронно или отсутствуют.

Клиент может использовать свой личный ключ для аутентификации без дополнительного взаимодействия с T, то есть без обмена сообщениями (1) и (2), до тех пор, пока не истечет срок действия ключа или не будет аннулирована версия ключа.

3. Анализ корректности протокола

Для того чтобы выявить возможные слабые места предложенного протокола и доказать его стойкость в условиях заданных предположений о доверии, проведем его формальный анализ с помощью BAN-логики, представляющей собой эффективную методологию анализа протоколов аутентификации. В процессе анализа будем использовать обозначения и постулаты, изложенные в работе [13] и в спецификации протокола. Так как в протоколе используются конструкции, не предусмотренные классической BAN-логикой, введем некоторые дополнительные формальные обозначения и постулаты:

1) $P \prec X$ — утверждение о знании участником P формулы X. Данное утверждение является более слабым, чем $P \lhd X$, подразумевающее передачу X в одном из сообщений, отправленных другими участниками протокола;

2) $[X]_{\overline{K}}$ — код аутентификации формулы X на ключе K, позволяющий установить знание X участником протокола без непосредственной передачи X по каналу связи;

3) $X \mapsto P$ — утверждение о принадлежности характеристики X участнику P;

4) $P \xleftrightarrow{K\rightarrow} Q$ — P и Q могут использовать подключ K, выработанный P, для генерации общего сеансового ключа. Подключ не может быть раскрыт никем, кроме P, Q и участников, которым они доверяют;

5) $P \xleftarrow{\leftarrow K} Q$ — P и Q могут использовать подключ K, выработанный Q.

Некоторые постулаты, касающиеся знания формул:

$$\frac{P \lhd X}{P \prec X}, \quad \frac{P \mid\equiv Q \xleftarrow{K} P}{P \prec K}, \quad \frac{P \mid\equiv \xrightarrow{K} P}{P \prec K, P \prec K^{-1}}, \quad \frac{P \mid\equiv \xrightarrow{K} Q}{P \prec K}, \quad \frac{P \prec X, P \prec K}{P \prec \{X\}_K}, \quad \frac{P \prec \{X\}_K, P \prec K}{P \prec X},$$

$$\frac{P \prec X, P \prec K}{P \prec [X]_K}, \quad \frac{P \lhd \{X\}_K, P \prec K, P \mid\equiv \#(X)}{P \lhd K}.$$

Последний постулат позволяет участнику P установить факт использования ключа K одним из других участников протокола в текущей транзакции.

Некоторые дополнительные постулаты:

$$\frac{P \mid\equiv \xleftarrow{K} P, P \lhd [X]_K, P \prec X}{P \mid\equiv Q \prec X}, \quad \frac{P \mid\equiv \xleftarrow{K} P, P \lhd (X, [X]_K)}{P \mid\equiv Q \mid\sim (X, [X]_K)}, \quad \frac{P \lhd (X, [X]_K), P \prec K}{P \lhd K}.$$

Формализация варианта I протокола может быть записана следующим образом:

$$(2) A \leftarrow T: \{N_A, A \xleftarrow{idKey_{AB}=[roleList \mapsto A,L]_{K_{BT}}} B, roleList \mapsto A, L\}_{K_{AT}},$$

$$(3) A \rightarrow B: roleList \mapsto A, L, activeRoleList \mapsto A, \{T_A, A \xleftarrow{k_A \rightarrow} B\}_{roleKey_{AB}=[activeRoleList \mapsto A]_{idKey_{AB}}},$$

$$(4) A \leftarrow B: \{T_A, A \xleftarrow{\leftarrow k_B} B\}_{roleKey_{AB}}.$$

Первое сообщение опущено, так как оно не меняет логических свойств протокола. Случайное число N_A, срок действия/номер версии L и метка времени T_A рассматриваются как значения, обеспечивающие свежесть сообщений. Свойства принадлежности списков ролей клиенту A используются в протоколе для доказательства корректности реализации политики безопасности на этапе аутентификации. Запишем формализацию предположений протокола.

$$A \mid\equiv A \xleftarrow{K_{AT}} T,\ T \mid\equiv A \xleftarrow{K_{AT}} T,\ A \mid\equiv A \xleftarrow{k_A \rightarrow} B,\ T \mid\equiv (roleList \mapsto A, L),$$

$$B \mid\equiv B \xleftarrow{K_{BT}} T,\ T \mid\equiv B \xleftarrow{K_{BT}} T,\ B \mid\equiv A \xleftarrow{\leftarrow k_B} B,\ T \mid\equiv A \xleftarrow{idKey_{AB}=[roleList \mapsto A,L]_{K_{BT}}} B,$$

$$A \mid\equiv activeRole\ List \mapsto A, \quad A \mid\equiv B \Rightarrow A \xleftarrow{\leftarrow k} B, \quad B \mid\equiv A \Rightarrow A \xleftarrow{k \rightarrow} B, \quad B \mid\equiv \#(T_A),$$

$$A \mid\equiv T \Rightarrow (roleList \mapsto A, L), \quad B \mid\equiv T \Rightarrow (roleList \mapsto A, L), \quad A \mid\equiv \#(N_A), \quad B \mid\equiv \#(L),$$

$$A \mid\equiv T \Rightarrow A \xleftarrow{idKey} B, \quad B \mid\equiv A \Rightarrow activeRole\ List \mapsto A, \quad A \mid\equiv \#(T_A).$$

Рассмотрим пошагово изменение областей доверия участников. При получении сообщения (2) клиентом A, используя известные постулаты BAN-логики, получаем:

$$A \lhd \{N_A, A \xleftarrow{idKey_{AB}=[roleList \mapsto A,L]_{K_{BT}}} B, roleList \mapsto A, L\}_{K_{AT}},$$

$$A \mid\equiv T \mid\sim (N_A, A \xleftarrow{idKey_{AB}=[roleList \mapsto A,L]_{K_{BT}}} B, roleList \mapsto A, L),$$

$$A \mid\equiv T \mid\equiv (roleList \mapsto A, L),$$

$$A \mid\equiv A \xleftarrow{idKey_{AB}=[roleList \mapsto A,L]_{K_{BT}}} B,$$

$$A \mid\equiv A \xleftarrow{roleKey_{AB}=[activeRoleList \mapsto A]_{idKey_{AB}}} B.$$

Далее A передает B сообщение (3), B выполняет вычисление ролевого ключа:

$$B \lhd (roleList \mapsto A, L, activeRoleList \mapsto A, \{T_A, A \xleftarrow{k_A \rightarrow} B\}_{roleKey_{AB}=[activeRoleList \mapsto A]_{idKey_{AB}}}),$$

$$B \prec (roleList \mapsto A, L, activeRoleList \mapsto A),$$

$$B \prec roleKey_{AB} = [activeRoleList \mapsto A]_{idKey_{AB}}.$$

Зная ролевой ключ, B может расшифровать защищенную часть сообщения и проверить свежесть метки времени T_A, откуда следует, что один из других участников протокола действительно использовал ролевой и личный ключ в текущей транзакции.

$$B \triangleleft \{T_A, A \xleftrightarrow{k_A \rightarrow} B\}_{roleKey_{AB}}, B \prec roleKey_{AB}, B \mid\equiv \#(T_A, A \xleftrightarrow{k_A \rightarrow} B),$$
$$B \triangleleft roleKey_{AB} = [activeRole\ List \mapsto A]_{idKey_{AB}},$$
$$B \triangleleft idKey_{AB} = [roleList \mapsto A, L]_{K_{BT}}.$$

Это позволяет B удостовериться в корректности личного ключа клиента $idKey_{AB}$ и полученного списка допустимых ролей.

$$B \mid\equiv B \xleftrightarrow{K_{BT}} T, B \triangleleft (roleList \mapsto A, L, [roleList \mapsto A, L]_{K_{BT}}),$$
$$B \mid\equiv T \mid\equiv (roleList \mapsto A, L),$$
$$B \mid\equiv A \xleftrightarrow{idKey_{AB} = [roleList \mapsto A, L]_{K_{BT}}} B.$$

Для осуществления логических переходов здесь используется доверие устройства B к свежести срока действия/номера версии L. Протокол остается защищенным от replay-атак за счет использования метки времени T_A.

Следующим шагом является обеспечение доверия B к ролевому ключу и списку активируемых ролей, выбор которых осуществляет клиент A. Проверка вложенности множества *activeRoleList* в множество *roleList* описано в спецификации и не проверяется формальной логикой:

$$B \mid\equiv A \xleftrightarrow{idKey_{AB}} B, B \triangleleft (activeRoleList \mapsto A, [activeRoleList \mapsto A]_{idKey_{AB}}),$$
$$B \mid\equiv A \mid\sim (activeRoleList \mapsto A, [activeRoleList \mapsto A]_{idKey_{AB}}),$$
$$B \mid\equiv A \xleftrightarrow{roleKey_{AB} = [activeRoleList \mapsto A]_{idKey_{AB}}} B.$$

Формально свежесть здесь обеспечивается сроком действия/номером версии L личного ключа $idKey_{AB}$, что сохраняет безопасность протокола при компрометации просроченных/аннулированных ключей. В итоге устройство B убеждается в достоверности сеансового подключа A и генерирует сеансовый ключ защищенного соединения, а также отправляет сообщение (4), на основе которого A убеждается в достоверности сеансового подключа B и в свою очередь генерирует сеансовый ключ:

$$B \mid\equiv A \mid\sim (T_A, A \xleftrightarrow{k_A \rightarrow} B), \qquad A \triangleleft \{T_A, A \xleftrightarrow{\leftarrow k_B} B\}_{roleKey_{AB}},$$
$$B \mid\equiv A \mid\equiv (T_A, A \xleftrightarrow{k_A \rightarrow} B), \qquad A \mid\equiv B \mid\equiv (T_A, A \xleftrightarrow{\leftarrow k_B} B),$$
$$B \mid\equiv A \xleftrightarrow{k_A \rightarrow} B, \qquad A \mid\equiv A \xleftrightarrow{\leftarrow k_B} B,$$
$$B \mid\equiv A \xleftrightarrow{k = f(k_A, k_B)} B, \qquad A \mid\equiv A \xleftrightarrow{k = f(k_A, k_B)} B.$$

Таким образом, предложенный протокол достигает следующих результатов:

$$A \mid\equiv A \xleftrightarrow{k} B, \qquad B \mid\equiv A \xleftrightarrow{k} B,$$
$$A \mid\equiv (roleList \mapsto A, L), \qquad B \mid\equiv (roleList \mapsto A, L),$$
$$A \mid\equiv activeRoleList \mapsto A, \qquad B \mid\equiv activeRoleList \mapsto A,$$
$$B \mid\equiv A \mid\equiv activeRoleList \mapsto A.$$

УХД и клиент разделяют общий совместно выработанный сеансовый ключ, а также доверяют спискам допустимых и активированных ролей. Формальный анализ доказал стойкость протокола для заданных начальных предположений о доверии участников и выявил отсутствие в протоколе избыточных элементов и операций.

4. Заключение

Итак, в статье показано соответствие предложенного протокола представленному выше набору требований к протоколу аутентификации клиентов при доступе к информации в объектных СХД, устраняющему недостатки схемы аутентификации клиентов, предложенной в стандарте ANSI. В настоящее время технология объектного хранения данных развивается стремительными темпами, и рассмотренный альтернативный протокол может использоваться в дальнейших исследованиях безопасности объектных СХД.

СПИСОК ЛИТЕРАТУРЫ:

1. *Vacca J.* The Basics of SAN Security — Part I, http://www.enterprisestorageforum.com/sans/features/article.php/1431341. July 2002.
2. ANSI INCITS 400-2004 — Information Technology — SCSI Object-Based Storage Device Commands (OSD). Dec. 2004.
3. *Factor M., Nagle D., Naor D., Riedel E., Satran J.* The OSD Security Protocol // Proc. of the 3rd IEEE International Security In Storage Workshop. Dec. 2005.
4. *Kent S., Seo K.* Security Architecture for the Internet Protocol // RFC 4301, Internet Engineering Task Force. Dec. 2005.
5. Fibre Channel Security Protocols (FC-SP), INCITS xxx-200x, T11/Project 1570-D/Rev 1.8, http://www.t11.org/ftp/t11/pub/fc/sp/06-157v3.pdf. June 2006. Work in progress.
6. ANSI INCITS 359-2004 — Information Technology — Role Based Access Control.
7. *Ferraiolo D., Gilbert D., Lynch N.* An Examination of Federal and Commercial Access Control Policy Needs // NIST-NCSC National Computer Security Conference. Sep. 1993.
8. *Gallaher M., O'Connor A., Kropp B.* The Economic Impact of Role-Based Access Control // NIST Planning Report. Mar. 2002.
9. *Reed B., Chron E., Burns R., Long D.* Authenticating Network-Attached Storage // IEEE Micro. Jan. 2000. Vol. 20.
10. *Bellare M., Canetti R., Krawczyk H.* Keying Hash Function for Message Authentication // Lecture Notes in Computer Science. 1996.
11. *Azagury A., Canetti R., Factor M., Halevi S., Henis E., Naor D., Rinetzky N., Rodeh O., Satran J.* A Two Layered Approach for Securing an Object Store Network // Proc. of the 1st IEEE International Security In Storage Workshop. Dec. 2002.
12. *Kher V., Kim Y. Decentralized Authentication Mechanisms for Object-based Storage Devices // Proc. of the 2nd* IEEE International Security in Storage Workshop. Nov. 2003.
13. *Burrows M., Abadi M., Needham R.* A Logic of Authentication // Proc. of the Royal Society of London. 1989.