УДК 681.3:519.872

А.В. Анисимов, А.В. Зубенко

ДИНАМИЧЕСКИЕ КОАЛИЦИИ — НОВАЯ ПАРАДИГМА В ОБЛАСТИ РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНО-КОММУНИКАЦИОННЫХ СИСТЕМ. Ч.2. ОБЗОР И СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПРАКТИЧЕСКИХ МЕТОДОВ ПОСТРОЕНИЯ ДИНАМИЧЕСКИХ КОАЛИЦИОННЫХ СРЕД

Статья является продолжением обзора, посвященного защите информации в групповых коалиционных объединениях. Рассматриваются существующие практические подходы к построению различных коалиционных сред, приводится их сравнительная характеристика.

Введение

Существует ряд различных технологических подходов к построению коалиционных сред, отражающих разносторонние проблемы коалиционной тематики. Главная задача при создании прототипа коалиционной среды состоит в устранении несоответствий между условиями, выдвигаемыми перед коалиционной средой, приводимыми в первой части обзора, и технологическими возможностями, присущими современным стандартным средствам разработки распределенных защищенных приложений.

Так, общепринятые в качестве стандарта протоколы установления защищенных двунаправленных сессий SSH и SSL(TLS) не реализуют динамическую модель авторизации и аудита доверительных отношений, изменчивых по своей природе, что особенно актуально в коалиционной среде.

Изначально авторизованный в контексте SSH-сессии пользователь получает неограниченный по временному параметру доступ к удаленной системе. Аналогично SSL-соединение автоматически не прерывается по истечении срока действия сертификата, принятого удаленной системой на этапе выполнения протокола рукопожатия. Данные ограничения не позволяют использовать вышеупомянутые два стандартных протокола в массивно распределенных системах, охватывающих множественные административные домены, характеризующиеся изменяемыми доверительными отношениями. Проекты DisCo и Yalta рассматривают задачи адаптации и изменения стандартных технологий построения защищенных распределенных приложений, устраняющих описанные ограничения.

Не соответствует классу коалиций распространенная технология защиты информации, основанная на использовании виртуальных частных сетей, применяемая в корпоративных средах. Наиболее широко используемый протокол IPSec, на основе которого создаются системы типа виртуальных частных сетей (VPN), характеризуется сложностью процесса конфигурирования, которое выполняется вручную администраторами узлов. Данное свойство также исключает возможность его использования при создании динамически изменяемой защищенной коммуникационной среды. Проект PREMYSES предлагает вариант адаптации IPSec к применению в динамических средах.

В целом, уровень средств защиты стандартных промышленных технологий распределенного программирования является недостаточным для разработки защищенных

информационных сред типа динамических коалиций.

Примером может служить технология Jini, позволяющая перейти от уровня сетевых протоколов к интерфейсам объектов в сети, предлагая, таким образом, универсальный метод предоставления различных сетевых сервисов. Данная уникальная и широко используемая технология не имеет средств аутентификации, контроля доступа и отказоустойчивости. Проект Yalta предлагает решение данной проблемы благодаря внедрению доверительного прокси-сервера между поставщиками и потребителями Jini-услуг, находящимися в различных доверительных административных доменах.

Иная, не решаемая традиционными методами проблема, возникающая в коалиционных средах, состоит в необходимости предоставления методов объединенного управления ресурсами, совместно используемыми партнерами по коалиции. Так, изъятие отделяющимся партнером собственных ресурсов может лишить оставшихся участников необходимой информации и нарушить целостность информационной среды. Подобные методы требуют специальных протоколов доступа к общим ресурсам, исключающих возможность несанкционированного доступа к критическим ресурсам одним из участников коалиции. Исследования, проводимые группой в МэрилендСком университете, направлены на создание средств эффективного объединенного управления общими ресурсами в коалиционных партнерских средах.

Распределенные среды также диктуют специфические требования к механизмам управления криптографическими параметрами. Централизованный подход к физическому хранению скрытых составляющих общих коалиционных криптографических ключей порождает ряд уязвимых мест в системе безопасности коалиции. Требуются новые методы распределенного хранения ключей и выполнения криптографических операций над ними.

Зачастую для группового сотрудничества необходим метод надежного широковещания. Интеграция методов групповой криптографии и технологий надежного широковещания позволяет строить защищенные групповые коалиционные системы, которые могут служить составными компонентами более сложных коалиционных схем.

Процессы, происходящие в коалиционных средах, могут включать определенные действия, основанные на выполнении мобильного кода на удаленных системах. Выполнение мобильного кода в средах, характеризующихся изменчивостью доверительных отношений, требует защищенности процесса его передачи, а также предоставления эффективных средств защиты удаленных систем от действий злоумышленного загруженного кода. Динамически конфигурируемые среды выполнения на сегодняшний момент также являются малоизученной тематикой.

Таким образом, очевидно, что коалиционная тематика происходит из практической необходимости создания нового поколения систем защиты информации, адекватно отвечающих условиям динамики, присущим изменчивым партнерским средам. Сегодняшние стандарты и подходы не способны предоставить решения для ряда практических задач, возникающих в процессе моделирования динамической партнерской среды.

1. Инфраструктуры открытых ключей

Среди множества методов и протоколов аутентификации методы, основанные на использовании инфрастуктур открытых ключей,

занимают особое место, поскольку они не только позволяют аутентифицировать оппонента, но и предоставляют эффективный защищенный метод распространения открытых составляющих пар криптографических ключей. Формат X.509 представляет собой цифровой сертификат, ассоциирующий субъект в сети с его открытым ключом, который оппоненты могут использовать для проведения различных криптографических операций. Для удостоверения подлинности сертификата и информации, скрытой в нем, сертификат подписывается доверительной по отношению к обоим оппонентам третьей стороной, открытый ключ которой известен. Данная процедура позволяет реализовать распределенную инфрастуктуру открытых ключей. Сертификат X.509 версии 3 содержит следующие основные поля:

- версия;
- идентификатор алгоритма подписи (алгоритм, параметры);
- имя объекта, выдавшего сертификат;
- срок действия (не ранее, не позже);
- имя субъекта;
- информация об открытом ключе субъекта (алгоритмы, параметры, ключ);
- уникальный идентификатор объекта, выдавшего сертификат;
- уникальный идентификатор субъекта;
- расширения;
- цифровая подпись (алгоритмы, параметры, непосредственная подпись);

Сертификаты X.509 — наиболее популярная форма построения инфраструктур открытых ключей как в Интернете, так и в закрытых корпоративных средах. Данная инфрастуктура была принята в качестве стандарта во многих широко используемых протоколах защищенного обмена информацией: SSL, IPSec, S/MIME. Наиболее известной альтернативой X.509 является инфраструктура публичных ключей PGP, главным образом применяемая для защиты электронной почты индивидуальными пользователями.

Основной проблемой, связанной с использованием инфраструктур открытых ключей является отсутствие методов своевременного оповещения об отзыве сертификатов, утративших срок действия. Известны два способа верификации действительности сертификата. Первый метод, изначально предложенный в стандарте X.509, основан на использовании списков отозванных сертификатов (CRL). Второй метод, предложенный в 1999 году, это протокол онлайновой проверки статуса протокола (OCSP). Однако оба данных средства не отвечают требованиям к надежному, в реальном масштабе времени, оповещению взаимодействующих сторон о недействительности сертификата оппонента, поскольку применимы лишь на этапе установления защищенного соединения с использованием сертификата. Коалиционные приложения требуют нового подхода к использованию инфраструктуры открытых ключей.

2. Разработки по тематике динамических коалиций, проводимые университетами Нью-Йорка и Аризоны

Университеты Нью-Йорка и Аризоны проводят совместные разработки в рамках единого проекта под названием “Распределенные сокровищницы. Эффективное защищенное разделение ресурсов в динамических средах” (“Distributed Sanctuaries. Efficient, Secure Resource Sharing in Dynamic Environments”). Новизна их подхода заключается в рассмотрении коалиционной архитектуры как взаимодействующих распределенных процессов и сервисов, которые

наделены определенным уровнем доверия по отношению друг к другу и способны самостоятельно порождать необходимые новые процессы без участия инициировавших из пользователей [1]. Таким образом, исследователи предлагают способ приведения сложных автоматизированных распределенных вычислительных систем к классу динамических коалиций. Далее приводится краткое описание системы DisCo [2], являющейся одним из главных компонентов, разработанных в рамках проекта и доступных для ознакомления и использования.

Цель DisCo – предоставление готовых программных средств для построения коалиционной среды, которая интерпретируется как совокупность комплексов распределенных приложений, инсталлируемых и запускаемых в пределах множества административных доменов, доверительные отношения между которыми нестабильны. Приложения служат для предоставления различных сервисов потребителям в сети. Архитектура подобных приложений состоит из динамически загружаемых компонентов, параллельно взаимодействующих в сети посредством защищенных и отказоустойчивых методов. Приведем типичный сценарий коалиционного приложения, поддерживаемого DisCo. Необходимо загрузить код, выпущенный организацией O_1, и выполнить на компьютере, находящемся под администрированием организации O_2, от имени пользователя, представляющего организацию O_3. DisCo предлагает программный интерфейс для написания подобных приложений на языке Java, имеющий следующие свойства:

1. Модулярная и, следовательно, легко заменимая абстракция управления доступом, основанная на условии частичного доверия между партнерами.
2. Коммуникационная инфраструктура и динамически конфигурируемая среда выполнения, использующая вышеупомянутую абстракцию механизма авторизации.
3. Непосредственное представление продолжительных и изменчивых с течением времени отношений авторизации, включающее механизмы реакции на происходящие изменения их характеристик.

Не привязываясь к определенному механизму авторизации, DisCo использует подход, основанный на адаптации единой настраиваемой схемы авторизации для всех решений, связанных с необходимостью принятия решений по предоставлению доступа. Предлагая в качестве готового решения специально разработанную систему управления доверительными отношениями dRBAC, DisCo также позволяет разработчикам использовать альтернативные схемы авторизации.

Архитектура DisCo состоит из четырех основных взаимосвязанных программных компонентов.

Единая абстракция управления доступом. Все стороны, вовлеченные в процесс авторизации, идентифицируются при помощи открытых ключей. Единая абстракция Authorizer используется для авторизации любых доверительных отношений. Компоненты DisCo определяют объекты Authorizer для любых интерфейсов, реализующих процессы, требующие контроля доступа. Результатом каждого успешного запроса на авторизацию является объект AuthMonitor, ролью которого затем является постоянный мониторинг состояния доверительных отношений между элементами коалиции, вовлеченными в данный процесс взаимодействия. Примером реализации данной абстракции является система dRBAC [3].

Система **dRBAC.** dRBAC представляет собой систему управления доверительными отношениями, основанную на ролевых критериях, разработанную для использования в средах типа динамических коалиций. Главными понятиями, лежащими в основе dRBAC, являются *роли* (roles), *делегирования* (delegations) и *хранилища делегирований* (wallets). Права любого субъекта на выполнение определенных действий подтверждаются фактом принадлежности субъекта к одной из необходимых для выполнения действия dRBAC-ролей. Делегирования являются мандатами, выражающими эквивалентность прав доступа пользователей, обладающих различными ролями, причем пользователи и роли могут быть определены в различных административных доменах. Каждое делегирование криптографически подписывается выпустившим его пользователем или организацией.

Общий вид делегирования следующий:

$$[Subject \rightarrow Object]\ Issuer,$$

где *Subject* — это субъект (пользователь, организация или роль); *Object* — это роль; *Issuer* — субъект. Данная запись обозначает, что субъект *Subject* имеет права, определенные для роли *Object*, что криптографически подтверждено выпускающим данное делегирование субъектом *Issuer*.

Роль *Object* имеет общий вид *OEntity.Oname,* что позволяет переписать делегирование как

$$[Subject \rightarrow OEntity.Oname]\ Issuer,$$

где *Oentity* — это субъект, определивший роль, а *Oname* — ее название.

В базовом случае, когда субъект *OEntity* совпадает с *Issuer*, делегирование является *самоподписным*.

Делегирование типа

$$[Subject \rightarrow Object']\ Issuer$$

означает, что субъект *Subject* не только получает права роли *Object*, но также имеет дополнительное право последующего делегирования данной роли. Подобный механизм позволяет строить транзитивные цепи делегирований, в результате применения которых субъекты получают права на роли, определенные издателем начального делегирования от промежуточных уполномоченных посредников. Делегирования подобного рода, когда *OEntity* и *Issuer* представляют собой различные субъекты, называются делегированием *третьей стороной*.

Расширение базовой модели делегирования позволяет увеличить возможности dRBAC и способствовать ее адаптации к потребностям реальных практических сред. Два дополнительных свойства называются *оценочными атрибутами* и *информацией об управлении мандатами*.

Оценочные атрибуты определяют дополнительные условия, связанные с выпускаемым делегированием, и указываются совместно с базовой информацией в следующем виде:

$$[Subject \rightarrow Object\ with\ A.Attribute1\ \langle Operator\rangle = \langle Value\rangle\ \langle and\ B.Attribute2\ \langle Operator\rangle = \langle Value\rangle\rangle^*]\ C,$$

где *A*, *B* и *C* могут быть одним и тем же субъектом, различными субъектами или их комбинацией, оператор *with* предшествует первому атрибуту, последующие атрибуты указываются через оператор *and*. Доступные значения *Operator* следующие:

- (–) вычитание положительного значения (0 по умолчанию);
- (*) умножение на коэффициент от 0 до 1 (1 по умолчанию).
- (<) нахождение минимального значения атрибута в цепочке

(+∞ по умолчанию).

Приведем пример делегирования с оценочными атрибутами, заимствованный из [3]:

$$[BigISP.member \rightarrow AirNet.member \text{ with } AirNet.BW <= 100 \text{ and } AirNet.storage \text{ -= } 20]\ Sheila$$

Данное делегирование означает, что элемент организации *BigISP* получает права доступа, определенные для члена организации *AirNet*, однако с ограничением по двум параметрам. *BW* (bandwidth – пропускная способность) не превышает 100 единиц, а объем хранилища данных *Storage* на 20 единиц меньше, чем у элемента *AirNet*. Делегирование подписано менеджером *AirNet*, пользователем *Sheila* (соответствующие делегирования опускаются).

Успешная авторизация действий пользователя в рамках dRBAC заключается в нахождении цепочки делегирований, необходимых для авторизации запрашиваемых действий. Возможность построения подобной цепочки носит название *доказательства* (proof). Доказательство записывается в виде $P \Rightarrow R$, что означает: субъект P имеет права, определенные для роли R. Информация об управлении мандатами, являющаяся вторым фундаментальным расширением dRBAC, необходима при построении доказательств. На ее основе строится алгоритм построения доказательств, описанный в [3].

Инфраструктура dRBAC предоставляет механизмы для создания и опубликования делегирований, обнаружения и проверки действительности необходимых делегирований, а также продолжительного мониторинга данной действительности на протяжении необходимого для сотрудничества периода времени. Для реализации данных операций над делегированиями строится сеть распределенных объектов-хранилищ для делегирований. Все типы операции выполняются по отношению к локальному хранилищу. Выпускаемые субъектами делегирования помещаются в локальные хранилища, которые содержат целостное представление о всех используемых мандатах при помощи механизма подписи на интересующие его делегирования.

Если запрос типа “имеет ли субъект P права, ассоциируемые с ролью R”, записываемый в виде “$?: P \Rightarrow R$”, возвращает доказательство, объектом, возвращаемым в действительности, является *контролер действительности доказательства* (proof monitor). При изменении статуса делегирований, влияющих на действительность доказательства, субъект, запросивший его, оповещается через интерфейс обратной связи. Аналогично, если хранилище на текущий момент не может предоставить доказательства для некоего запроса, заинтересованный субъект может зарегистрировать интерфейс обратной связи, активизируемый в случае обнаружения необходимого доказательства.

Более подробное описание механизмов, лежащих в основе dRBAC, приведено в [3].

Средства взаимодействия между компонентами посредством технологии SwitchBoard. Необходимость наличия стойкого к отказам защищенного канала связи между компонентами DisCo породила необходимость разработки специализированной коммуникационной инфрастуктуры, получившей название SwitchBoard [4]. Коммуникационные каналы SwitchBoard создаются динамически на основе загружаемых модулей, предоставляющих свойства шифрования, отказоустойчивости, авторизации, а также интерфейсы для передачи по-

токовых данных, объектов и вызовов удаленных процедур, известных как механизм RPC.

SwitchBoard представляет собой единую коммуникационную основу для всех DisCo-приложений, единой коалиционной среды. Двумя основными компонентами SwitchBoard являются:

Транспортный модуль. Ответственный за предоставление защищенных отказоустойчивых каналов между парой хостов. Все коммуникационные сессии между клиентом и сервером обслуживаются единым защищенным соединением по аналогии с SSH. Криптографические функции и контроль состояния соединения относятся к функциям данного модуля.

Сервисы и продолжительная авторизация. Транспортный уровень SwitchBoard доступен для множества сервисов, выполняемых на едином физическом хосте. По аналогии со средствами обнаружения доступных Jini-услуг, SwitchBoard предлагает реестр доступных сервисов. Соединения между клиентами и сервисами авторизуются, и действительность мандата партнера контролируется обеими сторонами через объект, реализующий Authorizer интерфейс. По умолчанию предлагается использование dRBAC.

SwitchBoard является аналогом шлюза защиты Yalta, рассматриваемого ниже в данной статье, и так же, как и последний, предоставляет средства защиты транспорта данных и продолжительного мониторинга доверительных отношений между партнерами коалиции. Дополнительное преимущество — наличие реестра сервисов для партнеров.

Система динамического контроля за ходом выполнения мобильного кода ***Runtime.*** Проблема защиты систем от действий недоброжелательного или некачественного мобильного кода, загружаемого из удаленных источников, остро стоит в процессе создания коалиционных моделей. DisCo подчеркивает актуальность данной проблемы и предлагает специальное решение в качестве использования системы Runtime. Данный компонент DisCo предлагает средства безопасной загрузки кода из внешних источников с учетом свойств конфиденциальности и целостности, а также его инсталляции и выполнения в безопасном виртуальном контейнере, ограниченном в соответствии с доверительными отношениями между субъектом, выпустившим код, и субъектом, от имени которого происходит его выполнение. Для транспорта кода используется SwitchBoard, а для установления и мониторинга доверительных отношений предлагается использование dRBAC-делегирований. Динамическое изменение параметров виртуального контейнера в соответствии с изменениями, произошедшими с делегированиями, реализуется при помощи надстроек над Java-механизмом SecureClassLoader.

Детальное описание механизмов Runtime приводится в [5].

3. Возможность применения групповых коммуникационных систем в процессе создания коммуникационной инфраструктуры коалиции

Групповые коммуникационные системы (далее — ГКС) определяются как приложения, служащие для обеспечения сотрудничества в пределах группы равноправных пользователей, предоставляющих надежную и упорядоченную доставку сообщений участникам группы, а также возможные сервисы мониторинга и управления составом группы. Существующие примеры приложений группового сотрудничества, такие, как системы проведения конференций, основаны на

применении широковещания — технологии, позволяющей получать информацию множеству участников сессии от одного отправителя без необходимости установления множества однонаправленных соединений между отправителем и каждым из получателей. Стандартным методом реализации подобной технической задачи является использование сетевой инфрастуктуры, поддерживающей IP-широковещание, что накладывает необходимость установки, конфигурирования и поддержки специального сетевого оборудования и программного обеспечения, однако это ресурсоемкий подход. ГКС предлагают альтернативный подход к реализации широковещания, эмулируя его на прикладном уровне и не требуя для этого специфической инфраструктуры.

ГКС являются хорошей основой для создания распределенных систем партнерского взаимодействия, сочетающихся с динамическими коалициями. В первую очередь, реализация широковещания, нередко требуемого приложениями в коалиционной среде для разделения определенного рода ресурсов, уже заложена в основу ГКС. Предоставление сервисов надежной, даже в случае отказа оборудования или каналов передачи данных, и упорядоченной доставки сообщений в рамках коммуникационной группы, являющаяся характерной чертой и преимуществом ГКС, также отвечает требованиям коалиционной модели. Следовательно, использование результатов, достигнутых в рамках исследований ГКС при выборе технологий, необходимых для построения коммуникационной инфраструктуры коалиции, является возможным подходом. В случае успешной интеграции протокола управления составом группы, если он реализован в используемой ГКС, с групповой стратегией коалиции получаем еще один дополнительный плюс. Так что, проектируя технологический каркас коалиции, следует в первую очередь обратить внимание на существующие ГКС. В данной статье рассматривается ГКС Spread как наиболее функциональная из известных на сегодня систем подобного типа, популярность использования которой быстро растет.

3.1. Spread — наиболее функциональная из известных ГКС. Рассмотрим пример ГКС, уже используемую на практике и разработанную исследовательским центром при университете Джона Хопкинса (The Center for Networking and Distributed Systems – CNDS) [6], деятельность которого уже не один год тесно связана с изучением различных аспектов групповых коммуникаций. Spread [7] отвечает всем типичным требованиям, выдвигаемым к ГКС, и, кроме этого, обладает рядом дополнительных преимуществ. В то время как существующие протоколы, предоставляющие надежное IP-широковещание, как правило, фокусируют внимание на поддержке глобальных сетей, обслуживающих теоретически любое, как угодно значительное количество пользователей, Spread не поддерживает групп такого размера, но реализует более существенную модель надежности и дополнительные сервисы, к примеру, упорядочивание сообщений. Разработчики системы Spread пошли по пути создания ГКС, предоставляющей поддержку широкого набора существующих на практике приложений партнерского взаимодействия. В частности, Spread была с успехом применена при построении приложений аудита и мониторинга распределенных систем, конференционного программного обеспечения, систем использования разделяемой памяти, высоконадежных служб (управление финансовыми операциями,

авиаперелетами, контроль за ходом военных операций), репликации баз данных и т.д. Методы надежного IP-широковещания на сегодня еще являются предметом исследований. Реализация модели надежности в данной технологии переносится на уровень приложения, так как на транспортном уровне уведомления о доставке быть не может, поскольку отсутствует единый конкретный получатель. Поэтому для транспорта пакетов может использоваться лишь UDP-протокол, а надежность доставки реализуется за счет контроля на уровне приложения. Модель обеспечения надежности доставки сообщений, реализованная в Spread, универсальна для любых типов приложений, использующих его в качестве коммуникационного канала. В этом состоит важное преимущество Spread перед IP-широковещанием, позволяющее использовать данный инструментарий широким прикладным образом.

Среди наиболее важных свойств Spread следует выделить следующие:

- эффективный и надежный канал передачи сообщений;
- надежное широковещание от любого количества отправителей любому количеству получателей;
- масштабируемые групповые сервисы значительного количества активных групп;
- службы поддержки состава группы, способные проинформировать все компоненты приложений о существующих в данной среде аналогичных компонентах и предоставляющие механизмы восстановления работы при отказе программного или аппаратного обеспечения на определенном участке сети;
- упорядоченная посылка сообщений в группы — все получатели овладевают сообщениями в одном и том же порядке;
- FIFO-упорядочивание потоков данных позволяет приложениям легко реализовать поддержку потоковых данных, таких, как аудио и видео;
- кроссплатформенность: Spread позволяет поддерживать взаимодействие приложений, выполняемых под различными операционными системами и их версиями: Unix (BSD, Linux, Solaris, Irix, AIX, Mac OS X, и др.), Windows (2000/NT/9x), а также MacOS X;
- предоставление API для широкого набора современных популярных языков программирования: C/C++, Java, Perl, Python и Ruby;

Spread состоит из двух компонентов, а именно выполняемого демона, который запускается на каждой из машин, составляющих соответственно группу машин-демонов, и библиотеки, предоставляющей программный интерфейс для написания групповых широковещательных приложений. Демоны поддерживают поток широковещательных приложений между собой, предоставляя гарантии упорядочения и доставки, устанавливая факты узлов и каналов в сети и управляя всеми подключенными к ним клиентскими приложениями.

Когда приложение отсылает сообщение через Spread, оно выбирает уровень сервиса для данного сообщения. Уровень сервиса определяет, какой тип упорядочения и гарантии доставки будет применен по отношению к данному сообщению в Spread окружении. Приложение может выбирать различные уровни для разных сообщений. Возможные уровни, определяющие комбинации параметров приведены ниже в табл. 1. Пояснения значения последних приведено ниже.

Порядок

- *Отсутствует* — любые сообщения с таким же значением могут пребывать как до, так и после данного.

Таблица 1

Тип Spread-сервиса	**Порядок**	**Надежность**
UNRELIABLE_MESS	Отсутствует	Ненадежный
RELIABLE_MESS	Отсутствует	Надежный
FIFO_MESS	FIFO по отправителю	Надежный
CAUSAL_MESS	Причинный	Надежный
AGREED_MESS	Полный	Надежный
SAFE_MESS	Полный	Безопасный

• *FIFO по отправителю* — все сообщения с данным значением как минимум FIFO-упорядочены по каждому из отправителей.

• *Причинный* (по Лэмпорту) — сообщения от всех отправителей упорядочиваются согласно причинному порядку [8].

• *Полный* — все сообщения от всех отправителей пребывают в точно таком порядке ко всем получателям.

Надежность

• *Ненадежный* — сообщение может быть утрачено и не восстанавливается Spread-окружением.

• *Надежный* — сообщение будет гарантированно доставлено всем членам группы, даже в случае возможных потерь в сети.

• *Безопасный* — данный сервис максимально безопасен с точки зрения сохранения согласованности в системе. Сообщение будет доставлено клиенту только тогда, когда демон, к которому он присоединен, уведомлен, что все остальные демоны также владеют сообщением.

Начиная с версии 3.0.16 Spread включает в себя методы аутентификации и авторизации пользователей, участвующих в коммуникационном процессе, что также может быть использовано на этапе построения УЗ коалиции [9].

Spread свободно распространяется и может быть использована согласно Spread Open-Source License. CNDS обозначает следующие вопросы, открытые для исследования:

• интеграция Spread с другими прикладными приложениями, отвечающими проблематике коалиций;

• изучение модели группового доверия, т.е. разработка алгоритмов вычисления степени доверия участникам группового сотрудничества;

• реализация групповой стратегии и выхода за рамки одноранговых групп пользователей;

• изучение объединения групп с различной стратегией.

Все перечисленные вопросы актуальны для тематики динамических коалиций.

Сотрудники CNDS обращают внимание на необходимость развития компонентов коалиционной модели, указанных в Ч.1. данной статьи, что позволило бы им построить собственную коалиционную модель на основе Spread. В первую очередь речь идет о расширении функциональности данной ГКС путем добавления методов управления групповой стратегией.

3.2. Защищенная ГКС Secure Spread и библиотека Cliques. Как одно из направлений своей деятельности, CNDS развивает тематику интеграции ГКС с технологиями защиты информации. Помимо стандартной версии Spread, предоставляющей коммуникационную основу для широковещательных приложений, CNDS также предложила

версию Secure Spread [10], задачей которой является расширение Spread до уровня защищенной ГКС посредством реализации криптографических сервисов. По сути, факт разработки данного продукта и является причиной повышенного интереса к исследованиям CNDS в рамках рассмотрения тематики динамических коалиций, так как защищенные ГКС — важная вспомогательная тема для проектирования коалиционных моделей. Данная система реализует конфиденциальность групповых коммуникаций посредством установления группового симметричного ключа и дальнейшего шифрования разделяемых данных с помощью алгоритма Blowfish, а также предоставляет возможность взаимной аутентификации взаимодействующих сторон. Для разделения группового ключа Secure Spread включает в себя программный интерфейс к библиотеке Cliques [11], реализующей известные на сегодня методы решения задачи установления групповых ключей. На данный момент библиотека поддерживает пять алгоритмов:

- GDH — протокол, основанный на групповом расширении известного двустороннего алгоритма разделения ключа Диффи – Хеллмана.
- CKD — централизованное распределение ключа с сервером ключей, динамически выбираемым среди участников группы; используется попарный алгоритм Диффи – Хеллмана для установления защищенной передачи группового ключа каждому из участников.
- TGDH — комбинирует алгоритм бинарных деревьев и GDH.
- STR — вариант TGDH с применением несбалансированных деревьев.
- BD — протокол, предложенный Бурместером – Десмедтом, еще один вариант расширения алгоритма Диффи – Хеллмана.

Протоколом, предложенным по умолчанию, является GDH. На данный момент это единственный протокол, поддерживающий каскадные групповые события. Для четырех других алгоритмов возможность изменения состава группы во время установления группового ключа отсутствует. В последующих версиях Secure Spread планируется надежная реализациях всех пяти методов.

Для детального изучения, анализа и сравнения производительности в различных прикладных условиях упомянутых выше криптографических методов следует обратиться к работе [12].

Как видим, Secure Spread адресует криптографическую проблему распределения групповых ключей, поэтому разработчики коалиций выбирают данную ГКС для реализации широковещательных защищенных сервисов при построении коалиционной архитектуры. Конкретные примеры применения Secure Spread можно найти в разд. 5 и 6.

В заключение раздела, посвященного групповым коммуникационным системам, добавим, что для более детального ознакомления с архитектурой Spread/Secure Spread следует обратиться к источникам [12–14].

4. Рабочие группы GSEC и MSEC, действующие в рамках IETF. Протокол GSKAMP

Говоря о проблеме недостаточной изученности методов построения коалиционных сред, нельзя обойти стороной вопрос вклада IETF в решение данной проблемы. Рано или поздно любой публичный Интернет-проект, претендующий на стандартизацию, должен оказаться на рассмотрении в этой авторитетной организации или одного из ее партнеров. Изу-

чение групповых, в том числе защищенных, взаимодействий не явилось исключением из этого правила. В общем случае законченность и успех любого технического начинания обычно идеально отражены его состоянием в рамках документов IETF [15]. Отсутствие хотя бы одного документа, касающегося тематики, перешедшего из состояния проекта в стандарты, лишний раз подчеркивает актуальность и необходимость проводимых в данной области исследований. Различные области, имеющие отношение к коалиционным схемам, главным образом разные аспекты групповых взаимодействий, время от времени находят свое отражение в потоке работ, рассматриваемых IETF. Выделим наиболее интересные из них. Еще в 1997 году был опубликован и получил экспериментальный статус протокол управления групповыми ключами GKMP — Group Key Management Protocol [16] — одна из первых попыток стандартизации процесса защиты групповых коммуникаций в рамках IETF. Однако протокол стандартом так и не стал. В 2000-м году развитие тематики защиты групп получает новый импульс — организуются две рабочие IRTF/IET-группы GSEC и MSEC [17], целью которых становится рассмотрение вопросов обеспечения безопасности в рамках взаимодействия больших и малых групп в сетевых окружениях, предоставляющих различные широковещательные методы передачи большого объема данных. GSEC вновь поднимает вопросы реализации аспектов безопасности, отсутствие которых лишает современные технологии группового взаимодействия целостности и требует стандартизации. Перечислим некоторые из областей интересов групп:

- управление групповой политикой;
- децентрализованное распределение групповых ключей;
- технологии защиты для открытых и закрытых групп. Открытыми

называем группы, позволяющие обмен данными с внешними абонентами.

Подобные группы могут накладывать условие существования '"публичного"

группового ключа;

- управление групповыми ключами. Рассмотрение новых протоколов распространения общего группового ключа для всех участников группового взаимодействия, имеющих различный вычислительный потенциал. Данная тематика особенно актуальна для беспроводных сетей;
- надежное широковещание. Взаимосвязь между защищенными и надежными методами широковещания до конца не изучена на данный момент.

Главным результатом работы групп является опубликование проекта (draft-документа [18]) нового протокола GSAKMP — Group Secure Association Key Management Protocol — протокола управлением ключами защищенных групповых соединений, предоставляющего защищенную модель для создания криптографических групп в сети [19, 20]. Помимо введения необходимых формальных определений, необходимых для рассмотрения данной тематики, авторы предлагают четкие механизмы определения групповой политики, принятия решений в области контроля доступа на стадии создания группы, генерации групповых ключей, делегирования групповой стратегии и распада группы. По нашему мнению, GSAKMP — наиболее полноценный из существующих на данный момент проектов протоколов группового защищенного взаи-

модействия. Но, к сожалению, он по-прежнему пребывает в стадии проектировки.

Также была предложена иная версия протокола — GSAKMP Light [21] — упрощенного варианта GSAKMP, целью применения которого является сокращение количества обмениваемых сообщений на этапе создания защищенной группы, что делает данную операцию менее ресурсо- и временноемкой.

5. Проект Yalta — новое поколение технологии Jini

Проект Yalta [22] на сегодня является одним из передовых исследований в рамках коалиционной тематики. Задачей Yalta стало расширение результатов, достигнутых в области распределенного программирования с конечной целью получения систем, способных обеспечить реализацию распределенного защищенного группового сотрудничества в разных сферах человеческой деятельности, включая поддержку сотрудничества географически удаленных военных союзников. Yalta предлагает расширения существующих в Java мощных методов распределенного программирования — технологии Jini [23] и одной из ее главных компонентов — технологии JavaSpaces [23]. Данные технологии являются стандартом для построения промышленных распределенных приложений, но они не обладают удовлетворяющими коалиционной схеме средствами защиты. Для построения данного уровня защиты реализуется масштабируемая и стойкая к попыткам внешнего несанкционированного вторжения система сертификации открытых ключей с использованием пороговой криптографии и избыточных серверов сертификации, предложенная Малкином [24], а также создается эффективная служба отзыва сертификатов на основании подписи и голосования. Реализация данных механизмов основана на специально разработанных расширениях стандартной Java-библиотеки JCE (Java Cryptography Extension), позволяющей использовать известные криптографические протоколы и стандарты на Java-платформе.

Высокоуровневая архитектура коалиции, предлагаемая Yalta, изображена на рис. 1.

Доменом в рамках коалиции будем считать независимую административную область, или же информационную систему отдельного члена коалиции.

Приложения всех доменов разделяют единое пространство приложений с целью разделения общих данных. Единая разделяемая всеми доменами инфраструктура публичных ключей (пространство PKI) служит для выдачи сертификатов, идентифицирующих действующие в рамках коалиции субъекты, и изменения их статуса. Пространство управляется набором серверов сертификации, функционирующих в каждом из доменов. В каждом из административных доменов коалиции действует одно или несколько приложений-агентов, обеспечивающих сервис предоставления приложениям информации о потерявших действительность сертификатах (CRN-агенты). Таким образом, динамическая природа коалиции в архитектуре Yalta выражается в реализации механизма своевременной реакции на отзыв криптографических удостоверений.

Архитектура Yalta объединяет передовые результаты в сфере защиты информации и систем инфраструктуры публичных ключей, искусственного интеллекта и агентных моделей, а также распределенных вычислений и методов разделения совместно используемых ресурсов.

Технологический каркас, на котором строится Yalta-базированная коалиция, использует Java-платформу, эксплуатируя

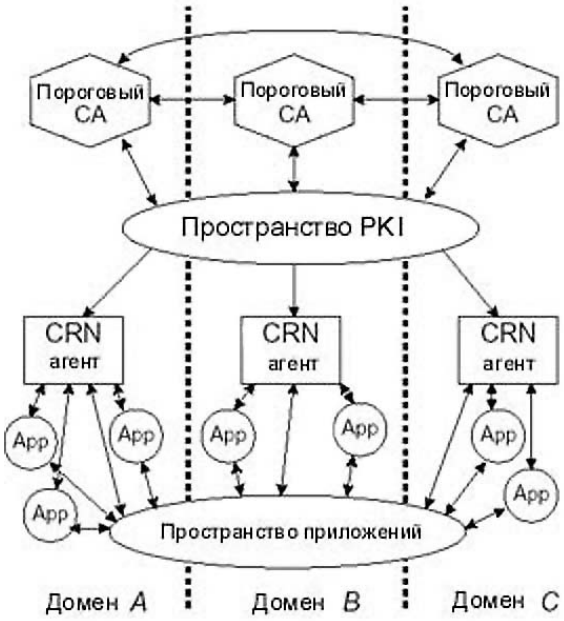

Рис. 1. Прикладная архитектура Yalta-коалиции

ее широкие возможности, а также обеспечивая межплатформенную переносимость и, как следствие, легкую масштабируемость конечной системы. Важно также, что Java — популярный, полностью объектно-ориентированный язык программирования, что позволяет конечным разработчикам использовать все преимущества ООП при написании прикладных приложений, являющихся составными частями Yalta-коалиции. Приведем список промышленных J2EE-технологий, задействованных при проектировании архитектуры Yalta: Jini, JavaSpaces, RMI поверх SSL, JSSE, JAAS, JCE, JTCA, Overture, JERI. Основой для реализации разделения ресурсов является метод разделения кортежей (tuplespaces), впервые предложенный в координационном языке Linda. Данная технология предлагает использование контекстно-адресуемой разделяемой памяти. Для репликации разделяемых кортежей применен Secure Spread, кратко описанный в разд. 3.2 и имеющий Java-прикладной интерфейс, что позволило использовать Spread в Yalta-среде. Именно Spread был выбран из ряда тестируемых ГКС, что служит подтверждением того факта, что ГКС с успехом могут и будут применяться при построении коалиционных моделей. Также следует отметить распределенную концепцию хранения разделяемых данных как ключевую особенность Yalta, что позволяет избежать негативных последствий использования централизованных подходов, хотя данный подход порождает дополнительные затраты на широковещательную передачу данных.

Уровень защиты коалиции, базированный на инфраструктуре публичных ключей X.509 и протоколе OCSP, полностью прозрачен для конечных приложений и реализует взаимную аутентификацию клиента и сервера, конфиденциальность данных, передаваемых из и в разделяемое пространство данных, своевременную реакцию на изменения в политике предоставления доступа к ресурсам, возможность разграничения доступа к данным на основании рода запрашиваемой операции и сертификата запрашивающей стороны.

Разработчики учли необходимость предоставления сервисов защиты любым приложениям, взаимодействующим в рамках коалиции, независимо от используемых на транспортном и прикладном уровне протоколов, и не остановились на реализации данных сервисов лишь для специфических приложений, использующих технологию разделяемого пространства кортежей. Предложенный шлюз защиты состоит из динамически загружаемых модулей, реализованных по типу прокси-сервисов для конкретных протоколов уровня приложения (HTTP, POP3, IMAP, NNTP). Шлюз реализует поддержку SSL для данных протоколов и кооперируется с CRN-сервисом коалиции. Посредством использования данного шлюза элементы коалиции могут устанавливать защищенные соединения с требуемыми сервисами, как внутренним в рамках коалиции, так и внешним по отношению к ней. Схема изображена на рис. 2.

Данная идея является сочетанием SSL-базированной VPN-системы и брандмауэра и адаптирована только к коалиционной динамической среде. Благодаря этому очевидно, что успешная коалиция строится на основе известных классических методов защиты информации в распределенных окружениях.

Другим ключевым звеном архитектуры модуля безопасности Yalta-коалиции является предложенный прокси-шлюз для Jini-сервисов, цель которого — предоставить конфиденциальность передачи, аутентификацию, контроль доступа, балансировку нагрузки и отказоустойчивость для произвольных Jini-сервисов, функционирующих в коалиционной среде. В силу того, что текущая реализация Jini не имеет необходимых механизмов защиты, эта часть проекта является расширением следующего поколения Jini, разрабатываемого в данный момент.

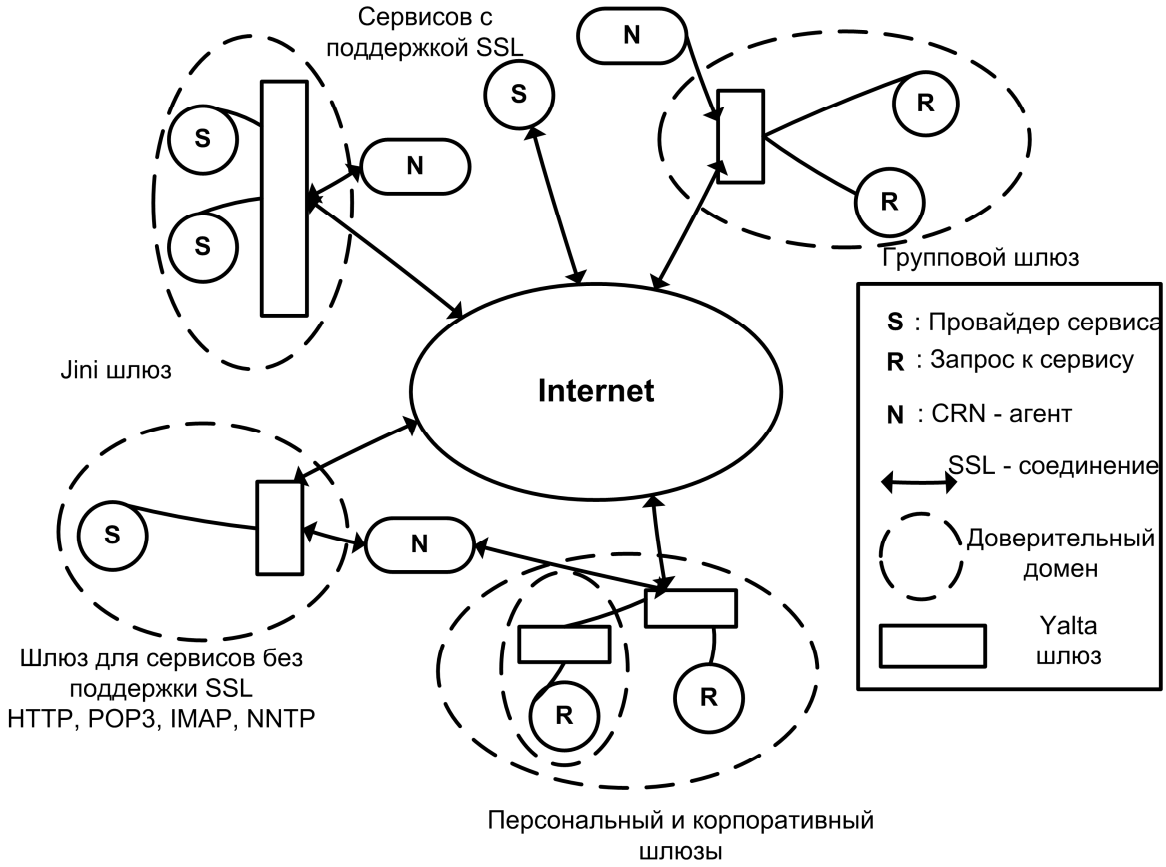

Рис. 2. Шлюз безопасности Yalta

Для детального ознакомления с архитектурой и непосредственно Java-технологиями, на которых базируется Yalta, следует обратиться к [25-27]. Там же можно найти и схематические примеры создания коалиции, динамического разграничения доступа к ее ресурсам, исключения участников, а также другие примеры действий, определяемых групповой политикой.

Yalta предлагает множество оригинальных идей, реализация которых может стать стандартом для будущего поколения распределенных вычислительных систем.

6. Исследования по теме динамических коалиций, проводимые в Мэрилендском университете

Мэрилендский университет ведет работу над проектом под названием “Интегрированные защищенные службы для управления динамическими коалициями (Intergrated Security Services for Dynamic Coalition Management — ISSDCM [28]). В рамках проекта предложен еще один вариант практической реализации коалиционного окружения [29], краткое описание которого приводится в данном разделе. Главной задачей проекта является создание механизма множественного владения общими для коалиционных партнеров ресурсами. На практике в процессе группового сотрудничества часто возникают ситуации, в которых определенный ресурс принадлежит нескольким равноправным владельцам, однако не существует единого механизма управления подобного рода ресурсами.

Характерная частная проблема, описывающая данную сложную зависимость ресурса от нескольких владельцев, возникает при выходе одного из них из состава коалиции. Должны существовать механизмы ограничения его действий, не позволяющие изъять из коалиционной среды ресурсы, необходимые другим участникам для продолжения работы. Данная тематика отражает сложность построения групповой стратегии, удовлетворяющей полноценной схеме группового сотрудничества. В [30-31] изложены подходы к решению данной задачи и изложена логика, описывающая протокол процесса авторизации запроса на получение доступа к совместному ресурсу посредством применения пороговых сертификатов, система подписи которых совпадает с предложенной в архитектуре Yalta (она заимствована в рамках обмена технологиями), а также предложен язык согласования, используемый для установления режима общего доступа, определяющего согласованные членами коалиции условия доступа к общим и индивидуальным для каждого домена ресурсам. В качестве основы построения данного языка взят RCL2000, в общем случае применимый для формализации ограничений в широко распространенных системах RBAC (Role-based Access Control — контроль доступа на основе ролей). Исследователи расширяют данный язык при помощи дополнительных элементов и функций, описывающих сущности коалиционных объектов и процессов, таких, как административные домены, совместно разделяемые ресурсы, права доступа, роли, присвоение ролей пользователям и пр. Для установления режима общего доступа к разделяемым ресурсам необходимо наличие единого для всех доменов органа управления атрибутами, выдающего и отзывающего сертификаты атрибуции. Последние определяют принадлежность пользователя домена к определенной роли. Также доказано, что необходима принудительная процедура отзыва сертификатов атрибуции при отзыве серти-

фиката идентификации пользователя во избежание получения несанкционированного доступа к информации после исключения пользователя из домена.

Результатом данной работы является практическая реализация предложенной теоретической модели. Архитектура коалиции представлена на рис. 3.

Коалиция из двух доменов является лишь примером, отображающим схему, масштабируемую на случай N доменов. Ее краткое описание приводится ниже (с целью получения полного описания см. [29]).

Каждый из доменов обладает собственным сервером сертификации CA (Certification Authority), выдающим сертификаты идентификации своим пользователям, а также базу данных собственных ресурсов и политики доступа к ним. На каждом из доменов также установлен набор инструментария для управления ресурсами коалиции CRM (Coalition Resource Management), состоящий из сервера сертификации доступа PKA CA (Public Key Access CA), RBAC модуля RCC (Role Control Center) и сервиса защищенной групповой коммуникации. Основывая коалицию, партнеры устанавливают сервер ресурсов коалиции, состоящий из веб-сервера, предлагающего интерфейс для доступа к разделяемым ресурсам и сервера защищенной групповой коммуникации, а также общий орган управления атрибутами AA (Attribute Authority). Последний также состоит из RBAC-модуля и сервера сертификации доступа SACA (Shared Access CA), скрытый ключ которого распределяется между всеми доменами. Далее домены устанавливают режим общего доступа, используя разрабатываемый группой исследователей специальный инструментарий [31]. Посредством RCC установленные режимом директивы передаются во все до-

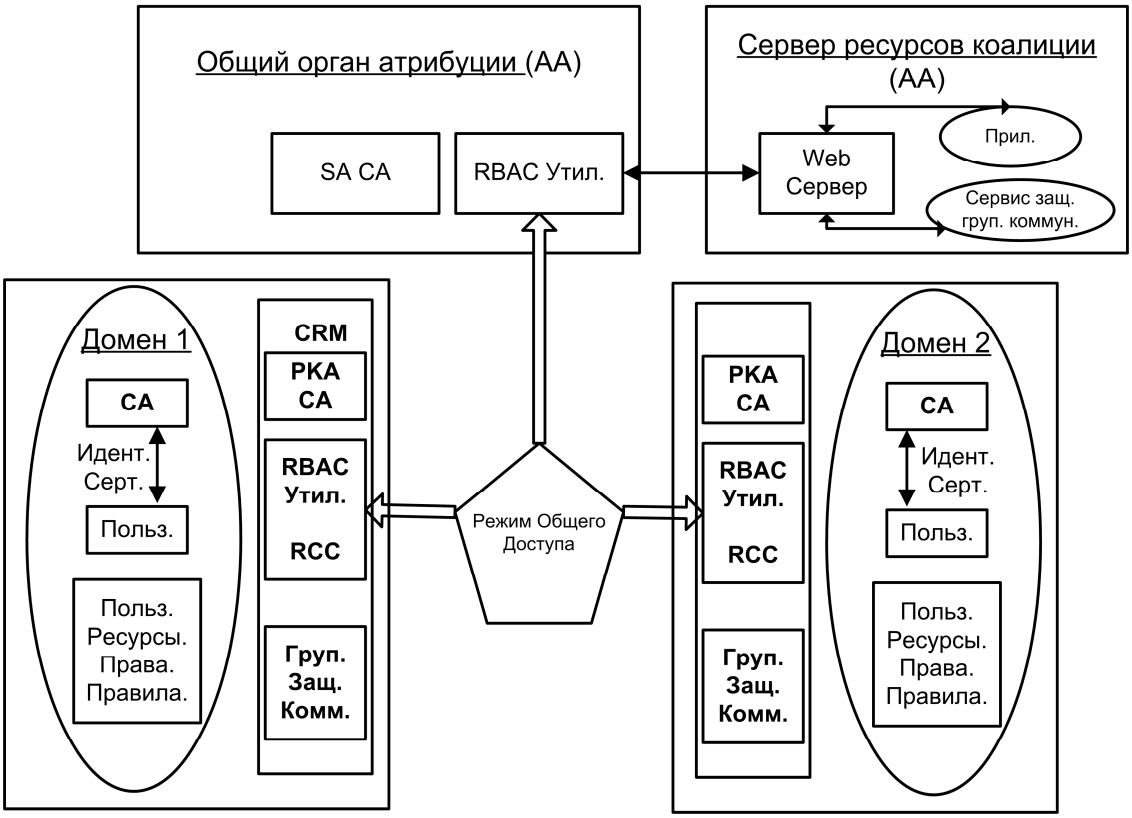

Рис. 3. Архитектура ISSDCM-коалиции

мены, а также на AA. Согласно согласованному режиму общего доступа AA устанавливает общие ресурсы на веб-сервере, а сервер сертификатов доступа выпускает сертификаты на имена тех пользователей, которые получили права на использование ресурсов. Данные сертификаты совместно подписываются частями скрытого ключа, распределенными между доменами. Сертификаты для доступа к приватным ресурсам домена внешними пользователями выпускаются севером сертификации данного домена. После распространения всех сертификатов установление общего режима доступа заканчивается и пользователи могут получить доступ к необходимым ресурсам.

Заметим, что в рамках реализации данной архитектуры имел место перенос на новую программную платформу уже известных технологий. Снова задействован Secure Spread, используемый для осуществления переноса общего режима доступа на все RCC и для реализации групповых приложений, а также разработчиками был легально позаимствован компонент Yalta, отвечающий за распределенное управление сертификатами. Вся система является Windows-ориентированной. Так, RCC, реализованный на Java, интегрирован с Active Directory на Windows 2000 и использует имеющиеся в нем роли и методы определения прав доступа к ресурсам, а также возможность хранения сертификатов, а используемый веб-сервер — Microsoft Internet Information Server. Spread и OpenSSL, используемые для генерации сертификатов доступа, также имеют Windows-версии.

Приведем основные отличия данной практической реализации от DisCo и Yalta:

- Использование общего сервера для хранения ресурсов.
- Строгая привязанность к единой операционной системе и встроенным в нее методам контроля доступа к ресурсам.
- Отсутствие методов поддержки универсальных приложений (см. шлюз защиты Yalta). Поддерживаются лишь веб-приложения.
- Наличие четкого механизма согласования правил доступа к совместно разделяемым ресурсам, основанного на строгой математической модели.

7. Сравнительная характеристика рассмотренных коалиционных моделей

Предлагаемые взгляды на рассмотрение коалиционной тематики в рамках данных исследований различаются по многим критериям. Краткая сравнительная характеристика рассматриваемых в данном обзоре подходов к построению коалиционных сред предлагается в виде табл. 2.

Единственный не рассматриваемый детально проект PREMYSES [32] посвящен вопросу адаптации VPN-систем [33] к потребностям динамических коалиционных окружений. В качестве решения предлагается использовать протокол CAP (Configuration Agreement Protocol — протокол согласования конфигураций), позволяющий динамически изменять конфигурацию активного оборудования, поддерживающего функциональность глобальной сети, и, таким образом, применить стандартный сетевой криптографический протокол IPSec [34] для защиты данных, передаваемых в рамках коалиции, участники которой находятся в разных сегментах глобальной сети. Процесс согласования новой конфигурации описан формальными средствами [35]. Предложенная практическая аппаратно-зависимая реализация также обеспечивает поддержку широковещательных приложений, таких, как видеоконференции. Обладая CAP-совместимыми маршрутизаторами, партнеры по коалиции могут быстро и эффективно реагировать на изменения в составе элементов и преодолевать сетевые распады без ущерба для защищенных внутренних коалиционных связей.

Разнообразие предлагаемых подходов оставляет широкую возможность выбора модели, наиболее близко отвечающей требованиям, возникающим в реально существующих распределенных средах.

Выводы

Данная статья завершала обзор по тематике динамических коалиций. В ней рассматривались практические аспекты построения динамических коалиций, приводились известные на сегодня методы расширения распределенных систем до систем класса динамических коалиций. Конечным результатом работы является приведенная сравнительная характеристика наиболее значительных исследований по данной тематике.

1. *Dasgupta P., Karamcheti V., Kedem Z.* Efficient and Secure Information Sharing in Distributed, Collaborative Environments // Proc. of the 3-rd Intern. Workshop on Communication-based Systems. — Berlin. — 2000, March. — P. 16.
2. *Freudenthal E., Karamcheti V.* DisCo: Middleware for Securely Deploying Decomposable Services in Partly Trusted Environments // Proc. of

Таблица 2

Характеристика	**ISSDCM**	**Yalta**	**PREMYSES**	**DisCo**
Поддержка совместного обладания ресурсами	Да	Нет	Нет	Нет
Поддержка взаимодействия агентных систем	Нет	Да	Да	Да
Наличие математической модели	Да	Нет	Да	Да
Централизованный подход к разделению ресурсов	Да	Нет	Нет	Да
Гибкий модуль аутентификации и авторизации	Да	Да	Нет	Да
Использование Spread как составного компонента	Да	Да	Нет	Нет
Механизм реализации криптографических сервисов	X.509 SSL/TL	X.509 SSL/TL	IPSec	X.509 SwitchBoa

Intern. Conf. on Distributed Computing Systems, ICDCS'2003. — 2003. — P. 10.
3. *dRBAC*: Distributed Role-based Access Control for Dynamic Coalition Environments / E. Freudenthal, T. Pesin, L. Port, E. Keenan, V. Karamcheti // Jbid — Viena, 2002. — P. 10.
4. *Switchboard*: Secure, Monitored Connections for Client-Server Communication / E. Freudenthal, T. Pesin, L. Port, E. Keenan, V. Karamcheti // Proc. of Intern. Conference on Distributed Computing Systems, ICDCS'2002. — Viena, 2002. — P. 6.
5. *Wahba H., Freudenthal E.* DisCo Runtime: Secure Code Distribution and Dynamic Runtime Permissions in a P2P Environment // Technical note, Parallel and Distributed Systems Group, NY University, July 2003. — http://www.cs.nyu.edu/pdsg/projects /disco/publications/runtime-tn2003.pdf.
6. *The* Center for Networking and Distributed Systems — CNDS. — http://www.cnds.jhu.edu/
7. *Stanton J.* Spread User's Guide. — http://www.spread.org/docs/guide/
8. *Lamport.* Time, Clocks and the Ordering of Events in a Distributed System // Comm. ACM. — 1978. — **21**, 7. — P. 558–565.
9. *Amir Y., Nita-Rotaru C., Stanton J.* Framework for Authentication and Access Control of Client-Server Group Communication Systems // Proc. of the 3-rd Intern. Workshop on Networked Group Communications. — London. — 2001. — Nov. — P. 12.
10. *Secure* Spread. — http://www.cnds.jhu.edu/ /research/group/secure_spread/
11. *Cliques* Research Project Homepage. — http://sconce.ics.uci.edu/cliques/
12. *On* the Performance of Group Key Agreement Protocols / Y. Amir, Y. Kim, C. Nita-Rotaru, G. Tsudik // Proc. of Intern. Conf. on Distributed Computing Systems, ICDCS'2002. — Viena, 2002. — P. 23.
13. *Exploring* Robusteness in Group Key Agreement / Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, J. Stanton, G. Tsudik // Proc. of the 21th IEEE Intern. Conf. on Distributed Computing Systems. — Phoenix, Arizona. — 2001, April. — P. 399–408.
14. *Scaling* Secure Group Communication: Beyond Peer-to-Peer / Y. Amir, C. Nita-Rotaru, J. Stanton, G. Tsudik // Proc. of DISCEX III Conf. — Washington DC. — 2003, April. — P. 12.
15. *The* Internet Engineering Task Force. — http://www.ietf.org/
16. *RFC* 2094. Group Key Management Protocol (GKMP) Architecture. — http://www.ietf.org/ /rfc/rfc2026.txt?number = 2094
17. *IETF* Secure Multicast Group (SmuG). — http://www.securemulticast.org/
18. *RFC* 2026. The Internet Standards Process. — http://www.ietf.org/rfc/rfc2026.txt?number=2026
19. *Group* Secure Association Key Management Protocol (GSAKMP) / H. Harney, A. Colegrove, E. Harder, U. Meth, R. Fleischer — 2000, Nov.— http://www.ietf.org/internet-drafts/draft-harney-sparta-gsakmp-sec-02.txt
20. *SPARTA* Inc. GSAKMP research accomplishments. — http://www.isso.sparta.com/gsakmp/
21. *GSAKMP* Light Protocol. — http://www.isso.sparta.com/gsakmp/docs/ /GSAKMP%20Light1_v1.2.doc
22. *Yalta* Project Homepage. — http://projects.anr.mcnc.org/Yalta/
23. *Таненбаум Э., Ван Стеен М.* Распределенные системы: принципы и парадигмы. — Изд.дом. "Питер", 2003. — 877 с.
24. *Malkin M., Wu T., Boneh D.* Experimenting with Shared Generation of RSA keys // Proc. Internet Society's Symp. on Network and Distributed System Security. — 1999, Feb. — P. 43–56.
25. *Dynamic* PKI and Secure Tuplespaces for Distributed Coalitions / T.J. Smith, G.T. Byrd, X. Wu, H. Xin, K. Thangavelu, R. Wang, A. Shah // Proc. of DISCEX III Conf. — Washington DC. — 2003, April. — P. 12.
26. *Yalta*: A Collaborative Space for Secure Dynamic Coalitions / G.T. Byrd, F. Gong, C. Sargor, T.J. Smith // Proc. of IEEE 2nd SMC Inform. Assurance Workshop. — New York. — 2001. — P. 10.
27. *Dynamic* Coalitions Principle Investigators Conference Slides. — Tech. Rep. July 8, 2002 Newport, RI. — http://www.anr.mcnc.org/ /projects/Yalta/DC-PI-08Jul2002.pdf

28. *Integrated* Security Services for Dynamic Coalition Management. — http://www.ece.umd.edu/ /~gligor/ISSDCM2003/ISSDCM2003.htm
29. *Integrated* Security Services for Dynamic Coalitions / H. Khurana, S. Gavrila, R. Bobba, R. Koleva, A. Sonalker, E. Dinu, V. Gligor, J. Baras // Proc. of DISCEX III Conf. — Washington DC. — 2003, April. — P. 3.
30. *Khurana H.* Negotiation and Management of Coalition Resources. PhD Dissertation. — Electrical and Computer Engineering Department, University of Maryland, College Park, MD. — 2002, August. — P. 160.
31. *Bharadwaj V, Baras J.* A Framework for Automated Negotiation of Access Control Policies. // Proc. of DISCEX III Conf. — Washington DC. — 2003, April. — P. 16.
32. *Dynamic* Coalitions-Policy Representation, Management and Infrastructure Systems (DC-PREMISYS). — http://govt.argreenhouse.com/ /DC-PREMISYS/
33. *Virtual* Private Network Consortium. — http://www.vpnc.org
34. *IP* Security Protocol (IPSec). — http://www.ietf.org/html.charters/ipsec-charter.html
35. *Dynamic* Coalitions-Policy Representation, Management and Infrastructure Systems (DC-PREMISYS) / W. Stephens, B. Coan, S. Narain, V. Kaul, K. Parmeswaran, T. Cheng. — http://govt.argreenhouse.com/DC-PREMISYS/ /dcpremisysOverview.pdf

Получено ***6.05.04***

Об авторах

Анисимов Анатолий Васильевич,
д-р физ.-мат. наук, профессор, декан

Зубенко Антон Витальевич,
аспирант

Место работы авторов:
Киевский национальный университет им. Т. Шевченко, ф-т кибернетики, Украина, Киев, просп. Академика Глушкова, 6, корп. 2

Тел. (044) 259 0427
E-mail: ava@mi.unicyb.kiev.ua
zubenko@softhome.net