МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА
Механико-математический факультет

На правах рукописи

Андреев Олег Олегович

Логико-языковые средства описания моделей логического разграничения доступа

Специальность 05.13.19 — методы и системы защиты информации, информационная безопасность

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Москва 2010

Работа выполнена на кафедре вычислительной математики Механико-математического факультета и в Институте проблем информационной безопасности Московского государственного университета имени М. В. Ломоносова.

Научный руководитель:	доктор физико-математических наук, профессор *Васенин Валерий Александрович.*
Официальные оппоненты:	доктор физико-математических наук, профессор *Грушо Александр Александрович*
	доктор технических наук, профессор *Заборовский Владимир Сергеевич*
Ведущая организация:	Научно-исследовательский институт системных исследований РАН

Защита состоится «28» апреля 2010 г. в 16 часов 45 минут на заседании диссертационного совета Д 501.002.16 при Московском государственном университете имени М. В. Ломоносова по адресу: РФ, 119991, Москва, ГСП-1, Ленинские горы, дом 1, Московский государственный университет имени М. В. Ломоносова, Механико-математический факультет, аудитория 14-08.

С диссертацией можно ознакомиться в библиотеке Механико-математического факультета МГУ (Главное здание, 14 этаж).
Автореферат разослан «27» марта 2010 г.

Ученый секретарь
диссертационного совета,
Д 501.001.84 при МГУ
доктор физико-математических наук А. А. Корнев

Общая характеристика работы

Актуальность темы. В связи с активным использованием информационно-вычислительных комплексов для решения практически значимых задач все больший интерес в последнее время проявляется к средствам описания политик их информационной безопасности и, в частности, к моделям и механизмам логического разграничения доступа (ЛРД) к ресурсам таких комплексов[1].

Положения законов и подзаконных актов, нормативно-регламентирующих документов, стандартов и рекомендаций в области обеспечения безопасности информационных технологий определяют ряд требований к механизмам защиты в автоматизированных системах, в том числе — к программным механизмам логического разграничения доступа. К числу требований, предъявляемых к таким механизмам, используемым в автоматизированных системах с повышенными требованиями к их защищенности, относится требование существования формальных моделей, на основе которых они функционируют[2]. Формальное описание модели ЛРД сложно организованной компьютерной системы является крайне трудоемкой задачей в связи с гетерогенностью средств описания составляющих ее моделей в отдельных компонентах программной системы.

Механизмы ЛРД, встроенные в различные компоненты, зачастую обладают собственным языком описания набора правил, в соответствии с которыми доступ разрешается или запрещается. Это обстоятельство затрудняет анализ со стороны пользователя или администратора соответствия механизмов ЛРД, которые реализуются в отдельных компонентах сложной системы, положениям безопасного использования ресурсов системы в целом. Отмеченные выше и ряд других недостатков эксплуатирующихся в настоящее время систем стимулируют работы по созданию новых логико-языковых средств описания моделей ЛРД, таких как eXtended Access Control Markup Language. Данные средства предоставляют администратору системы возможность самому определить модель ЛРД, в большей степени соответствующую потребностям защищаемого информационно-вычислительного комплекса, или выбрать необходимую по требованиям информационной безопасности модель среди предлагаемых другими разработчиками. Однако, и это следует отметить, постоянно выполняемые на протяжении всего времени функционирования информационной системы проверки на разрешение или запрещение доступа при использовании сложных моделей ЛРД, основанных на такого рода языковых средствах, становятся существенно более затратными в смысле потребляемых при их использовании вычислительных ресурсов, чем при применении более простых моделей. Последнее обстоятельство приводит к замедлению работы информационной системы. Таким

[1]Васенин В. А. Проблемы математического, алгоритмического и программного обеспечения компьютерной безопасности в Интернет. Материалы конференции МаБИТ-03. Москва, 2004. — С. 81–99.

[2]Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. Руководящий документ ФСТЭК от 30 марта 1992 года / ФСТЭК. — 1992.

Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Руководящий документ ФСТЭК от 30 марта 1992 года / ФСТЭК. — 1992.

образом, одной из задач, важных для успешного внедрения и использования подобных средств в реальных информационно-вычислительных комплексах, является разработка, внедрение и оптимизация механизмов ЛРД, основанных на выразительных логико-языковых средствах.

В ходе жизненного цикла информационных систем происходят постоянные изменения, обновления, объединения и разделения, связанные с соответствующими процессами, происходящими в организациях, которые контролируют соблюдение требований безопасности ресурсов таких систем. Такие изменения должны отражаться в политиках информационной безопасности организаций и, в частности, в моделях ЛРД. Процесс создания и изменения моделей ЛРД и, в частности, их интеграции или разделения на несколько моделей является крайне трудоемким и подверженным ошибкам.

Отмеченные выше обстоятельства определяют актуальность задачи автоматизированного анализа корректности моделей и механизмов ЛРД, которая, в свою очередь, во многом зависит от свойств языка их описания. Настоящая работа направлена на разработку нового, более удобного для использования на практике языка описания моделей ЛРД к ресурсам информационно-вычислительных (автоматизированных) систем и решение представленных выше задач применительно к такому языку.

Цель диссертационной работы состоит в исследовании методов описания моделей логического разграничения доступа, в разработке и практической реализации на этой основе программных средств разграничения доступа пользователей к ресурсам компьютерных систем. Для достижения этой цели сформулированы и решаются следующие задачи:

- создание формальной модели языка описания моделей логического разграничения доступа;
- разработка на ее основе языка описания моделей логического разграничения доступа, эффективного с позиций предъявляемых к нему требований;
- разработка алгоритмов анализа (проверки) свойств моделей, представленных на основе разработанного языка;
- реализация программных средств разграничения доступа, основанных на разработанном языке и повышение их производительности;
- проведение тестовых испытаний программных средств разграничения доступа.

Цель работы и перечисленные задачи соответствуют положениям паспорта специальности 05.13.19 — методы и системы защиты информации, информационная безопасность.

На защиту выносятся следующие основные результаты.

- Формальное описание класса моделей логического разграничения доступа, которое включает в себя достаточное широкое с практической точки зрения множество моделей.
- Логико-языковые средства (язык), позволяющие эффективно с позиции предъявляемых к ним требований описывать рассматриваемый класс моделей.
- Алгоритм, позволяющий производить сравнения моделей, описанных с помощью разработанного языка, на предмет того, является ли одна из них «включенной» в другую.
- Программные средств разграничения доступа, построенные на основе разработанного языка и предназначенные для их внедрения в ядро ОС Linux и в программы, написанные на языках C и Python.
- Методы повышения эффективности программных механизмов логического разграничения доступа, позволяющие существенно ускорить принятие решения о предоставлении доступа к информационным активам, вычислительным и коммуникационным ресурсам защищаемых систем, а также результаты тестовых испытаний предложенных методов.

Методы исследования. Для формализации ряда используемых в настоящей работе понятий и проведения строгих доказательств используются следующие методы:

- методы математической логики, включая исчисление предикатов;
- методы дискретной математики, в том числе, теории графов, теории булевых функций;
- методы программной инженерии.

Научная новизна результатов диссертации состоит:

- в разработке новых подходов к описанию формальных моделей логического разграничения доступа на основе логико-языковых средств;
- в создании способов сравнения описанных таким образом моделей;
- в исследовании вопросов программной реализации механизмов разграничения доступа на основе разработанных логико-языковых средств.

Практическая значимость диссертационной работы заключается в разработанном автором языке описания моделей логического разграничения доступа, методах анализа моделей, описанных с помощью этого языка, а также в реализованных программных механизмах разграничения доступа, основанных на разработанном языке и предназначенных для внедрения в ядро Linux и прикладные программы.

Внедрение результатов работы. Результаты работы нашли свое применение в процессе выполнения проектов: «Методы и средства противодействия

компьютерному терроризму: механизмы, сценарии, инструментальные средства и административно-правовые решения» (НИР 2005-БТ-22.2/001 в рамках ФЦП «Исследования и разработки по приоритетным направлениям науки и техники»). Получено свидетельство об официальной регистрации программы для ЭВМ «Набор специализированных дистрибутивов ОС Linux с повышенными требованиями к защищенности» (свидетельство № 2006613706).

Апробация работы. Результаты работы докладывались на научных конференциях «Математика и безопасность информационных технологий» (2005–2006 гг.), «Актуальные проблемы вычислительной математики» (2006 г.), на семинаре «Проблемы современных информационно-вычислительных систем» под руководством д. ф.-м. н., проф. В. А. Васенина (механико-математический факультет МГУ имени М. В. Ломоносова, 2005, 2009 гг.).

Публикации. По результатам работы опубликовано 7 научных статей, из них — одна статья в журнале из перечня ведущих рецензируемых изданий, рекомендованных ВАК [1].

Материалы работы вошли в главу 3 опубликованной в 2008 году коллективной монографии «Критически важные объекты и кибертерроризм. Часть 2. Аспекты программной реализации средств противодействия» под ред. В. А. Васенина [6].

Личный вклад автора заключается в проведенном им анализе существующих логико-языковых средств описания моделей логического разграничения доступа, в разработке нового языка описания таких моделей, алгоритма анализа моделей, представленных на разработанном языке, а также в создании программных средств, реализующих разграничение доступа на основе нового языка.

Структура работы. Работа состоит из введения, пяти глав, заключения, списка литературы и двух приложений. Общий объем диссертации — 109 страниц. Список литературы включает 61 наименование.

Содержание работы

Во введении сформулирована цель диссертационной работы, обоснована ее актуальность, аргументирована научная новизна и практическая значимость полученных результатов, представлены выносимые на защиту результаты исследований.

Первая глава диссертации посвящена результатам выполненных автором исследований, направленных на систематизацию и обобщение сведений об используемых в настоящее время на практике моделях логического разграничения доступа (ЛРД) и способах их описания. Рассмотрены преимущества и недостатки каждой из представленных моделей ЛРД и средств их описания. Выделяются ключевые задачи,

решение которых представлено в работе.

Принятый в России международный стандарт ГОСТ Р ИСО/МЭК 15408 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий»[3] является одним из ключевых действующих нормативных документов в области обеспечения безопасности информационных технологий. В данном стандарте отмечается как необходимость использования механизмов ЛРД (функциональное требование), так и важность управления и анализа конфигурации средств защиты в целом (одно из требований доверия). В Руководящих документах ФСТЭК России отмечаются требования по необходимости проведения формального доказательства корректности работы механизмов защиты для подконтрольных объектов с высокими уровнями защищенности.

Первый раздел главы посвящен описанию основных понятий в области разграничения доступа. На настоящее время наиболее распространенным определением логического разграничения доступа в контексте защиты информации является следующее: разграничением доступа называется процесс, позволяющий на основе анализа некоторой информации определить, какие действия данный субъект может производить по отношению к данному объекту. В ходе работы информационной системы, подлежащей защите, при каждом обращении к ее ресурсам со стороны пользователей происходит запрос на доступ к подсистеме ЛРД, соответствующей запрашиваемому ресурсу. В том случае, если запрашиваемое действие входит в список действий, которые данный пользователь может выполнять с данным ресурсом, доступ разрешается, в противном случае — в доступе отказывается.

На практике каждая программная подсистема, реализующая механизмы ЛРД основывается на некоторой формальной модели, отражающей общие свойства процесса принятия решения о доступе субъекта к объекту. К таким моделям, которые в контексте настоящей работы именуются базовыми, относятся дискреционная, многоуровневая (мандатная), ролевая и другие модели. Каждая из этих моделей представляет собой описание набора правил, на основании анализа которых принимается решение о доступе. Вместе с тем, реализация механизмов ЛРД к ресурсам каждой конкретной компьютерной системы, как правило, основывается на более сложных чем базовые моделях, которые учитывают особенности такой системы, среды ее окружения и положения политики ее информационной безопасности. В каждом из этих случаев базовые модели или их комбинации детализируются целым рядом дополнительных ограничений и правил. В отличие от базовых такие модели в контексте работы именуются детализированными. В настоящее время разработаны и используются различные базовые модели ЛРД. Описание наиболее распространенных из них, а именно — многоуровневой модели Белла-ЛаПадулы[4], дискреционной[5] и ролевой[6] представлены в работе.

---

[3]ГОСТ Р ИСО/МЭК 15408–1–2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. — М.: ИПК Издательство стандартов, 2002.

[4]Bell D., LaPadula L. Secure Computer Systems: Mathematical Foundations, Technical Report. Mitre Corporation, 1973.

[5]A Guide to Understanding Discretionary Access Control in Trusted Systems, NCSC-TG-003. National Computer Security Center, 1987.

[6]Ferrariolo D., Kuhn R. Role-Based Access Control, // Proceedings of 15th National Computer Security Conference, 1992.

Основы модели Белла-ЛаПадулы формулируются следующим образом:

- субъект не имеет права читать данные из объектов с уровнем секретности выше, чем у него;
- субъект не имеет права писать в объекты с уровнем секретности ниже, чем у него.

Как можно видеть, система ЛРД, основанная на этих правилах, гарантирует конфиденциальность данных, то есть, сохранение их секретности. Модель Белла-ЛаПадулы имеет и ряд существенных недостатков, включая главный из них — ориентированность исключительно на защиту конфиденциальности информации.

С неформальных позиций дискреционное разграничение доступа определяется следующим образом: у каждого объекта существует субъект, называемый владельцем, и именно он определяет права на доступ к этому объекту для других субъектов. Дискреционная модель является одной из самых гибких моделей ЛРД. Подобная гибкость является преимуществом этой модели ЛРД и одной из главных причин ее широкого распространения. Вместе с тем, из последнего свойства следуют и недостатки дискреционного разграничения доступа, а именно:

- матрица разграничения доступа может быть слишком большой в крупной информационной системе с большим числом субъектов и объектов, что затрудняет ее проверку и анализ на соответствие требованиям информационной безопасности, принятым в этой системе;
- невозможно ограничить субъектов-владельцев объектов в предоставлении ими доступа другим субъектам;
- зачастую, определение владельца объекта не представляется возможным.

Ролевая модель базируется на понятиях субъектов, называемых в ней пользователями, ролей и привилегий. В этом модели понятия объекта и доступа к нему заменяются понятием привилегии, означающим, возможностью выполнить какое-либо действие. Каждый пользователь может принадлежать к одной или нескольким ролям, которые характеризуют его положение и должность в организации, которой, в свою очередь, принадлежит система. В модели отсутствует понятие владельца информации. Предполагается, что владельцем информации является организация, которой принадлежит информационная система. Зачастую ролевое разграничение не позволяет задать необходимые требования к разграничению доступа. Так, с помощью ролевой модели невозможно описать такое разграничение доступа, при котором разрешение или запрет доступа будет зависеть от количества предыдущих доступов или от текущего времени.

С развитием, активным использованием во всех сферах человеческой деятельности, и усложнением информационных систем изменяются и требования к информационной безопасности, в том числе, к правилам ЛРД к их ресурсам. Традиционно используемые базовые модели ЛРД, такие как дискреционная или ролевая, а также механизмы на их основе не всегда оказываются достаточно выразительными для

описания всех требований по безопасности, которые предъявляются к подобным системам. По этой причине возникает необходимость в разработке моделей ЛРД, ориентированных на конкретные информационные системы или требования по их безопасности, которые предъявляют использующие эти системы организации. Сама по себе разработка модели не является в большинстве случаев трудоемкой процедурой. Однако задача реализации механизмов ЛРД на ее основе и их корректное использование в составе программных средств — достаточно сложный и подверженный ошибкам процесс. С целью его упрощения и автоматизации в 1990-х годах начали развиваться языковые средства описания моделей ЛРД для решения отмеченных выше задач. В случае использования механизмов ЛРД, основанных на подобном языке, достаточно описать модель в терминах этого языка.

Во втором разделе первой главы описывается один из представительных языков описания моделей ЛРД общего назначения, которым в настоящее время является eXtended Access Control Markup Language (XACML)[7], принятый в качестве стандарта комитетом Oasis Open. Одним из ключевых понятий, на основе которого формулируются модели в стандарте XACML, является понятие атрибута. Атрибутом является свойство субъекта, объекта, типа доступа или окружения. Каждый атрибут имеет имя и фиксированный тип данных, которые могут в нем содержаться. Разграничение доступа производится на основе значений атрибутов субъекта, объекта, типа доступа и среды окружения, которые учитываются при принятии решения о доступе.

Язык XACML предоставляет широкий спектр средств для описания моделей ЛРД. Вместе с тем, он имеет существенные недостатки, которые затрудняют как теоретические исследования этого языка, так и практическую реализацию механизмов ЛРД, созданных на его основе. Одним из важнейших недостатков языка является его сложность, как чисто синтаксическая, так и сложность описания семантики. Синтаксическая сложность проявляется в том, что запись даже простых моделей ЛРД с помощью XACML излишне длинна и плохо воспринимается пользователем. Семантическая сложность заключается в большом количестве понятий, которые определяются в стандарте языка. Это обстоятельство затрудняет разработку формальной модели языка в целом. Ни один из результатов исследований, проводившихся на этом направлении до настоящего времени, не привел к ее созданию. Отсутствует математическая модель процесса функционирования механизмов ЛРД, основывающихся на описанных с помощью его частных моделей. В заключении главы констатируется, что отмеченные обстоятельства приводят к необходимости разработки нового языка описания моделей ЛРД, позволяющего адекватно описывать широкий класс таких моделей, важных с практической точки зрения.

Во второй главе рассматривается разработанный автором язык описания моделей ЛРД. Представлены базовые понятия, синтаксис и формальное математическое описание достаточно представительного класса моделей ЛРД, а также примеры использования языка для их описания.

---

[7] eXtensible Access Control Markup Language (XACML) Commitee Specification. OASIS Open, 2003. Электронная версия печатной публикации. http://www.oasis-open.org/committees/xacml/

В первом разделе второй главы описываются понятия, на которых основывается предлагаемый автором язык. Основным понятием, на котором базируются модели ЛРД, описывающиеся с использованием этого языка, кроме понятий субъекта, объекта и типа доступа, является понятие атрибута безопасности, в дальнейшем для краткости именуемого просто атрибутом. Каждый объект, субъект, тип доступа, а также окружение может иметь некоторые задаваемые пользователем или механизмом ЛРД атрибуты. Под окружением имеется в виду часть информационной системы, общая для всех субъектов и объектов, не зависящая от них.

Модель ЛРД формулируется в виде некоторых условий на атрибуты субъекта, объекта, типа доступа и окружения. В дальнейшем, в целях сокращения текста системное окружение будем называть окружением. Механизм ЛРД, программно реализующий модель, описанную на разработанном автором языке, разрешает доступ в том случае, если условие выполняется, и запрещает его в противном случае. Таким образом, модель можно представлять как функцию от атрибутов, возвращающую истинное или ложное значение. Процесс принятия решения о разрешении или запрещении доступа будет в дальнейшем называться применением модели к запросу на доступ, а само решение — результатом применения.

Разработанный автором язык включает в себя средства задания моделей ЛРД, которые могут иметь так называемые пост-действия. В качестве таких рассматриваются действия, которые должны быть выполнены при каждой попытке доступа. Также, как и в случае с атрибутами, реализация механизмов ЛРД на основе нового языка должна поддерживать определенный набор пост-действий.

Представление модели ЛРД в рассматриваемом языке структурно подразделяется на модели и правила. Модель состоит из области применения, набора правил и других моделей, алгоритма комбинирования результатов, а также набора пост-действий. Область применения модели является условием, определяющим, к каким запросам на доступ она может быть применена. От алгоритма комбинирования зависит процесс применения модели к запросу на доступа. После каждого применения модели к доступу выполняются пост-действия, содержащиеся в данной модели.

Правило представляет собой базовую единицу модели, состоящую из области применения, условия и результата. Условие определяет результат применения правила к запрашиваемому доступу. Оно зависит от атрибутов субъекта, объекта, типа доступа и окружения. В случае отсутствия такого условия считается, что оно всегда выполняется. Результатом правила может быть либо «разрешено», либо «запрещено».

Список синтаксических конструкций, на основе которых строится описание модели ЛРД на представленном языке, приводится во втором разделе второй главы. Одной из конструкций языка, является ссылка, которая может содержаться в любой модели ЛРД и указывать на другую модель. В представленной работе описан алгоритм, преобразующий модель ЛРД со ссылками к эквивалентной ей форме, в которой ссылки на другие модели отсутствуют. Такое преобразование возможно, если в исходной модели отсутствуют взаимные ссылки с другими моделями. Для проверки отсутствия таких ссылок автором приводится алгоритм, а также доказывается его корректность. Алгоритм состоит в выполнении следующей последовательности

действий.

- Строится направленный граф, называемый графом зависимостей, вершинами которого являются отдельные модели.
- Между двумя вершинами графа зависимостей ставится направленное ребро в том случае, если модель, соответствующая началу ребра, включает модель, соответствующую концу ребра.
- Проверяется существование в графе направленных циклов.
- В том случае, если циклы существует, проверка завершается с ошибкой.
- В противном случае, описания моделей не являются взаимно исключающими.

В работе доказано следующее утверждение.

Утверждение 1.1. *Алгоритм проверки наличия взаимных включений корректен, а именно — завершает свою работу за конечное время.*

Для формализации основных понятий разработанного автором языка в четвертом разделе второй главы вводятся следующие определения. Пусть заданы:

- S, множество имен атрибутов субъектов;
- O, множество имен атрибутов объектов;
- E, множество имен атрибутов окружения;
- A, множество имен атрибутов типов доступа;
- $V = \mathsf{Boolean} \cup \mathsf{Integer} \cup \mathsf{Float} \cup \mathsf{String} \cup \mathsf{Sets} \cup \mathsf{Maps}$, множество значений атрибутов, где $\mathsf{Boolean}$ обозначает предопределенный булевый тип, $\mathsf{Integer}$ — целочисленный, Float — вещественный, String — строковый, Sets — объединение типов «множество фиксированного типа» и Maps — объединение множеств типа «ассоциативный массив фиксированного типа»;
- $\mathsf{Type} : S \cup O \cup E \cup A \to 2^V$, функция, устанавливающие каждому имени атрибута множество значений, которые этот атрибут может принимать;
- $F \subset V^V \cup V^{V \times V} \cup \ldots \cup V^{V \times \ldots \times V} \cup \ldots$, множество частичных функций с различным количеством аргументов, определенных на подмножествах V и возвращающих значения из V.

Определение 1.1. *Множеством выражений в заданных выше обозначениях называется множество, которое по индукции строится следующим образом.*

1. *Выражениями являются элементы множества* $S \cup O \cup E \cup A$ *— все возможные имена атрибутов, а также элементы множества* V *— константы.*
2. *В том случае, если* $v_1, \ldots, v_n$ *— выражения, а* f *— функция из множества* F *с количеством аргументов* n*, то* $f(v_1, \ldots, v_n)$ *является выражением.*

Типом выражения v *называется множество возможных значений выражения. Оно определяется следующим образом:*

1. $\mathrm{Type}(v)$ *в том случае, если* v *принадлежит* $S \cup O \cup E \cup A$*;*
2. Boolean *в том случае, если* v *является константой из множества* Boolean*;*
3. *аналогично определяется тип выражения для остальных констант;*
4. *множество возвращаемых значений функции* f *в том случае, если выражение имеет вид* $f(v_1, \ldots, v_n)$*.*

Правильно типизированными выражениями называется множество выражений, которое строится следующим образом:

1. *элементы множеств* $S \cup O \cup E \cup A$ *и* V *являются правильно типизированными выражениями;*
2. *в том случае, если* $v_1, \ldots, v_n$ *— правильно типизированные выражения, имеющие типы* $T_1, \ldots, T_n$*, и* f *— функция из множества* F *с количеством аргументов* n*, определенная в том числе на множестве* $T_1 \times \ldots \times T_n$*, то* $f(v_1, \ldots, v_n)$ *является правильно типизированным выражением.*

В дальнейшем в работе рассматриваются лишь правильно типизированные выражения.

Определение 1.2. *Условием на атрибуты в указанных выше обозначениях называется такое правильно типизированное выражение* v*, типом которого является* Boolean.

На основе данных выше определений правильно типизированного выражения и условия на атрибуты строится определение правила и модели ЛРД.

Определение 1.3. *Правилом называется пара* $(\mathrm{Target}, \mathrm{Condition})$*, где* Target *и* Condition *— условия на атрибуты. Компонента*Target *соответствует цели правила,* Condition *— условию правила.*

Моделью называется четверка $(\mathrm{Target}, \mathrm{Models}, \mathrm{Alg}, \mathrm{Obligations})$*. При этом:*

- Target *— условие на атрибуты, соответствующее цели модели;*
- Models *— последовательность правил или моделей, используемых в основной модели;*
- Alg *— алгоритм комбинирования, один из множества* $\{\mathrm{deny-overrides}, \mathrm{permit-overrides}, \mathrm{first-applicable}\}$*;*
- Obligations *— последовательность пост-действий, представляемых кортежами вида* $(\mathrm{name}, \mathrm{attr}_1, \ldots, \mathrm{attr}_n)$*, где* $\mathrm{name} \in \mathrm{String}$ *— вид обязательного действия,* $\mathrm{attr}_1, \ldots, \mathrm{attr}_n \in S \cup O \cup E \cup A$ *— имена атрибутов, значения которых будут переданы.*

Автором показывается, если не рассматривать пост-действия, то приведенная в предыдущем определении тройка (Target; Models; Alg) может быть представлена в виде одного условия на атрибуты, которая для краткости будет именоваться формулой доступа модели. Такое представление используется в дальнейшем в алгоритме анализа моделей ЛРД, описанных с помощью разработанного языка. Для доказательства этого утверждения в работе сформулирована и доказывается следующая теорема.

Теорема 1.1. *Для любой модели, задаваемой с помощью языка описания моделей ЛРД, существует условие на используемые в ней атрибуты, которое истинно тогда и только тогда, когда модель разрешает доступ.*

В третьей главе рассматриваются современные подходы к анализу моделей ЛРД, обосновывается необходимость разработки новых подходов и алгоритмов, которые могут быть применены в задаче изучения таких моделей, описанных с помощью представленного во второй главе языка.

В первом разделе главы 3 приводятся существующие методы анализа свойств моделей ЛРД. Одним из широко распространенных методов подобного анализа является анализ, основанный на поиске пути в графе допустимых доступов и на алгоритмах переписывания графов[8]. В случае использования конструкций из теории графов для описания моделей ЛРД представляется целесообразным применить к исследованию таких моделей методы поиска и преобразования графов. Задаваемые свойства, как правило, связаны с наличием пути между некоторыми вершинами графа.

Другим распространенным способом проверки свойств механизма ЛРД является «верификации на модели». Такой класс методов носит название «model checking», в русскоязычных источниках используются наименования «методы верификации моделей» или «методы верификации на модели»[9]. Подобные методы основаны на задании модели как системы переходов между состояниями («машины состояний»), для каждого из которых задан набор истинных в нем атомарных высказываний пропозициональной логики. Проверяемое свойство специфицируется в виде формулы временной логики, накладывающей ограничения на состояния в форме путей в заданной системе переходов. В зависимости от используемого языка временной логики некоторым образом ограничивается последовательность состояний, содержащихся в допустимых путях системы переходов

Существующие методы анализа моделей ЛРД по сути не подходят для моделей, описанных на разработанном автором языке. Причиной такого несоответствия

[8]Koch M., Mancini L.V., Parisi-Presicce F. Decidability of safety in graph-based models for access control. // In European Symposium on Research in Computer Security, pp. 299–243, 2002.

Rozenberg G., editor. Handbook of Graph Grammar and Computing by Graph Transformation. Vol. 1: Foundations. — World Scientific, 1997.

[9]Guelev D., Ryan M.D. и Zhang N. Evaluating access control policies through model checking. Proceedings of Eighth Information Security Conference (ISC05), 2005.

Morisset C., Santana de Oliveira A. Automated Detection of Information Leakage in Access Control. Universite Paris, Vandoeuvre les Nancy, 2007.

Шапченко К. А.. Современные методы проверки свойств безопасности в моделях логического разграничения доступа. // Проблемы информатики. — No3, 2009. — 124 Новосибирск: НГТУ, 2009. — С. 22–32.

является тот факт, что эти методы требуют либо представления каждой отдельной модели в виде отдельной, характерной именно для данного метода формы, которой может быть модель, или один из видов логики, либо в рамках этих методов формулируется сам язык описания доступа в виде модели или одного из видов логики. В том случае, когда каждую модель ЛРД для анализа ее свойств необходимо записывать в специальной форме, теряется основное преимущество представленного языка, а именно — единообразие описания всех используемых моделей. В другом случае, когда язык описания формулируется в виде модели, результаты анализа получаются слишком общими и не имеющими перспектив их практического применения. Отмеченные обстоятельства приводят к необходимости разработки методов анализа моделей, формализованных с помощью представленного языка, которые позволяли бы проверять практически значимые свойства таких моделей по их описанию.

В процессе обновления политики информационной безопасности автоматизированной системы или объединения политик безопасности отдельных ее компонентов, ресурсы которых подлежат защите, возникает необходимость изменить или объединить в единую несколько моделей ЛРД. При проведении такого объединения или изменения зачастую возникает необходимость проверить, является ли вновь полученная модель более «слабой», то есть, менее ограничивающей, чем предыдущие модели. Проведение такой проверки позволяет гарантировать, что у потенциальных злоумышленников не появилось новых возможностей доступа к защищаемым ресурсам информационной системы. Во втором разделе главы 3 приводится разработанный автором алгоритм, позволяющий определить, является ли одна модель ЛРД, описанная с помощью разработанного языка, более слабой чем другая.

В работе определяется, что одна модель слабее другой, если первая модель разрешает все доступы, которые разрешает вторая модель. С формальной точки зрения это свойство записывается следующим образом.

Определение 1.4. *Одна модель ЛРД с формулой разрешения доступа* F_1 *называется более слабой, чем другая, с формулой разрешения доступа* F_2 *в том случае, когда истинно следствие* $\mathsf{F}_2 \Rightarrow \mathsf{F}_1$. *Одна модель ЛРД называется более сильной, чем вторая, если вторая является более слабой, чем первая.*

Разработанный автором и представленный в данной работе алгоритм позволяет проверять, является ли одна модель ЛРД более сильной, чем вторая, при определенных ограничениях на подаваемые ему на вход модели. Ограничение налагается на формулы разрешения доступа моделей и, как следствие, на формулы Target и Condition их правил. Оно состоит в том, что множество F функций, на основе которого строятся эти формулы, содержит только булевы связки $\wedge$, $\vee$, $\neg$, а также операции сравнения $=$, $\neq$, $<$, $>$.

Определение 1.5. *Назовем литералом одно из следующих выражений:*

- *булева константа;*
- *булев атрибут или отрицание булевого атрибута;*

- *выражение вида* (a op b), *где* a *и* b *— атрибуты или константы, а* op *— операция из множества* $\{=, \neq, <, >\}$ *(то есть, множество разрешенных функций за исключением логических связок).*

Множество формул разрешения доступа моделей, которые могут быть поданы на вход представляемому алгоритму, называются допустимым множеством. Автором доказывается следующее утверждение.

Теорема 1.2. *Любую формулу разрешения доступа из допустимого множества можно представить в виде булевой формулы, в которой переменными являются литералы.*

Перед описанием основного алгоритма анализа свойства моделей ЛРД, описанных на разработанном автором языке, в работе представляется дополнительный алгоритм, который используется в одном из шагов основного.

Дополнительный алгоритм получает на вход конечный набор отношений вида $\{a_i\ op_i\ b_i\}$, где a_i и b_i — атрибуты или константы, а op_i — одна из функций =, ≠, <, >.

Определим понятие противоречивости такого набора отношений.

Определение 1.6. *Набор отношений называется противоречивым, если не существует такого сопоставления атрибутам значений соответствующих типов, при котором каждое из отношений было истинно.*

Сопоставление атрибутам значений называется означиванием атрибутов, а атрибуты, которым сопоставлено значение — означенными. Дополнительный алгоритм проверяет поданный на вход набор отношений на противоречивость.

Как легко заметить, каждому из отношений набора можно поставить в соответствие определенный тип. Этот тип является типом атрибутов и констант, участвующих в рассматриваемом отношении. Использование в одном отношении атрибутов или констант двух разных типов запрещено определениями функций =, ≠, < и >. Таким образом, весь набор разбивается на три группы отношений, имеющих целочисленный, вещественный и строковый типы, соответственно.

Типы атрибутов, использующихся в каждой из групп, отличаются. По этой причине множества атрибутов, использующихся в каждой из групп, не пересекаются, а, следовательно, для доказательства непротиворечивости изначального набора отношений необходимо проверить каждую из них. Проверка группы на противоречивость для случая, когда атрибуты имеют целочисленный, вещественный или строковый тип, производится одинаковым способом. Причина в том, что в таких группах допустимы одни и те же операции, а именно — операции сравнения =, ≠, <, >. В работе описываются действия на последовательных шагах алгоритма для этого случая.

1. По группе отношений строится граф с вершинами, соответствующими атрибутам и константам, которые используются в ее отношениях.

2. Вершины, которые в соответствии с условиями группы находятся в отношении равенства, «склеиваются». В том случае, если «склеиваются» вершины, находящиеся в отношении «не равны», либо вершины, соответствующие различным константам, группа является противоречивой.

3. После «склейки» вершин в графе расставляются направленные ребра между вершинами, находящимися в отношении «меньше» или «больше», ребра направлены от большей вершины к меньшей.

4. В том случае, если в графе существуют направленные циклы, группа является противоречивой.

5. В том случае, если в графе существуют направленные пути, ведущие от меньше константы к большей или равной, группа противоречива.

6. В противном случае группа является непротиворечивой.

Автором доказываются следующие утверждения, обосновывающие корректность представленного дополнительного алгоритма.

Утверждение 1.2. *Если указанный алгоритм выдает результат о противоречивости группы отношений, то используемым в нем атрибутам не могут быть сопоставлены значения таким образом, чтобы отношения из группы стали истинны.*

Утверждение 1.3. *Если при применении указанного выше алгоритма получается результат о непротиворечивости группы отношений, то существует означивание используемых в ней атрибутов.*

Как отмечалось ранее в работе, при представлении дополнительного алгоритма, в случае рассмотрения группы отношений, соответствующей целочисленному типу значений атрибутов, указанные выше шаги алгоритма не являются достаточными для определения непротиворечивости такой группы. В связи с изложенными выше соображениями, необходимо расширить описываемый алгоритм дополнительным шагом. На этом шаге должны рассматриваться все отношения частичного порядка, соответствующие построенному графу. Для каждого из таких отношений должна проверяться возможность означивания атрибутов методом, указанным выше в доказательстве корректности алгоритма.

Утверждение 1.4. *Представленный ранее в работе алгоритм проверки непротиворечивости группы отношений, дополненный указанным шагом, является корректным.*

После рассмотрения дополнительного алгоритма в работе описываются шаги основного алгоритма. Его суть заключается в проверке того факта, является ли первая модель более сильной, чем вторая, в предположении что на вход поданы две модели ЛРД, удовлетворяющие указанному ограничению. Формула разрешения доступа первой модели будет обозначаться как F1, формула разрешения доступа второй —

как F2. Алгоритм в соответствии с определением должен выполнять проверку истинности формулы $F1 \Rightarrow F2$, на всех возможных значениях аргументов, которая эквивалентна проверке тождественной ложности формулы $F1 \wedge \neg F2$, обозначаемой в дальнейшем F. Необходимо отметить, что формула F удовлетворяет тому же ограничению, что и формулы F1 и F2.

Представленный в работе алгоритм состоит из следующих шагов.

1. Формула F рассматривается как булева формула относительно введенного определения литерала и приводится к виду дизъюнктивной нормальной формы с помощью эффективного алгоритма.

2. Каждый из дизъюнктов, полученный на первом шаге, должен быть ложен при любых значениях атрибутов, ложность которого необходимо проверить, выглядит в общем виде как $a_1 \wedge \ldots \wedge a_n \wedge (a_{n+1}\ op_1\ b_{n+1}) \wedge \ldots \wedge (a_{n+k}\ op_k\ b_{n+k})$, где $a_1, \ldots, a_{n+k}$, b_1, ..., b_{n+k} — атрибуты, отрицания атрибутов или константы, а op_1, ..., op_k — операции из разрешенного множества.

   Набор отношений $(a_{n+1}\ op_1\ b_{n+1})$, ... $(a_{n+k}\ op_k\ b_{n+k})$ проверяется на противоречивость представленным выше алгоритмом проверки набора отношений на противоречивость. Если набор противоречив, дизъюнкт всегда ложен, иначе существуют значения атрибутов, при которых дизъюнкт истинен.

Автором доказывается утверждение о корректности разработанного им алгоритма проверки тождественной ложности формулы.

Теорема 1.3. *Алгоритм проверки тождественной ложности формулы разрешения доступа корректен, то есть, дает результат о тождественной ложности поданной на вход формулы в том и только том случае, когда для любых значений атрибутов, использующихся в формуле, она является ложной.*

Четвертая глава посвящена вопросам программной реализации механизмов ЛРД на основе разработанного языка. Рассматриваются основные принципы построения архитектуры средств ЛРД и их место в программных комплексах защиты информационной безопасности. Приводятся технические аспекты внедрения механизмов ЛРД в ядро Linux и в другие программные системы. Исследуются вопросы повышения производительности разработанных механизмов.

Разработанные в рамках диссертации механизмы безопасности в первую очередь предназначаются для внедрения в специализированные программные дистрибутивы с повышенным уровнем защищенности на базе ядра ОС Linux. В контексте требований существующих нормативно-правовых документов следует отметить, что задача построения таких дистрибутивов непосредственно связана с управлением функциями безопасности, отдельные требования к которому представлены в ГОСТ Р ИСО/МЭК 15408 «Общие критерии», а именно — в описании класса требований FMT («Управление безопасностью»). Создание таких дистрибутивов позволит не только более эффективно проводить изменения настроек механизмов безопасности (включая «атрибуты безопасности» в терминологии стандарта), но и предоста-

вит возможность организовать проверку выполнения ряда требований по безопасности состояния защищаемой системы до и после таких изменений.

Одним из направлений практического использования языка описания механизмов ЛРД для достижения целей, поставленных перед разработчиками дистрибутива программного обеспечения, является внедрение этого языка в средства описания моделей, на основе которых формулируются все механизмы ЛРД, использующиеся в составе отдельных компонентов компьютерной системы. Формальное описание на одном языке всех моделей ЛРД, используемых в такой системе, позволяет существенно упростить для администратора задание единой политики ее информационной безопасности.

Программные комплексы, входящие в состав дистрибутива ОС, делятся на ядро Linux, программное обеспечение промежуточного уровня (middleware) и прикладные программные комплексы. В рамках работы решены задачи, направленные как на внедрение программных реализаций механизмов ЛРД на основе разработанного автором языка в ядро ОС, так и на упрощение использования таких механизмов в прикладных программах.

В связи с повышенным интересом к информационной безопасности и, соответственно, к средствам разграничения доступа, в последнее время стали широко выполняться проекты по модификации ядра ОС Linux для внесения в него дополнительных возможностей по разграничению доступа. Самыми распространенными проектами в этой области являются SELinux и RSBAC.

Во втором разделе четвертой главы описывается разработанная автором реализация механизма ЛРД, который внедряется в ядро ОС Linux. Эта реализация представляет собой модуль разграничения доступа для подсистемы RSBAC и набор дополнительных утилит, которые позволяют загружать модели ЛРД и настраивать атрибуты и дополнительные параметры. Данная реализация позволяет разграничивать доступ пользователей к локальным файлам и директориям. Атрибуты подразделяются на определяемые реализацией, определяемые администратором безопасности (выделенный пользователь в RSBAC) и определяемые пользователем.

Внедрение механизмов ЛРД в различные приложения и программные средства промежуточного уровня представляет собой достаточно трудоемкий процесс. Его реализация может быть облегчена, если общая часть таких механизмов выносится в отдельный программный модуль. Такой модуль может затем использоваться при создании любых программных комплексов, доступ в которых должен разграничиваться с помощью механизмов, описанных на представленном выше языке. В рамках выполнения данной диссертации разработана библиотека, которая может использоваться в любых программах, написанных на языке C. Реализована также дополнительная библиотека, взаимодействующая с первой и позволяющая использовать механизмы ЛРД на основе предложенного языка в программах, написанных на языке Python. Детали реализации этих библиотек описываются в третьем разделе четвертой главы.

Выразительная способность языков описания моделей ЛРД тесно связана с вопросами производительности механизмов обеспечения информационной безопасности. Использование языковых средств для формального описания сложных моде-

лей, состоящих из десятков и сотен правил, может существенно замедлить работу информационной системы, особенно, если проверки на разрешение доступа должны производиться часто. В четвертом разделе данной главы приводятся разработанные автором методы повышения эффективности вычисления ответа на запрос о доступе. Эти методы могут применяться в механизмах, основанных на представленном в работе языке.

Одним из практически значимых способов, позволяющих оптимизировать механизмы ЛРД, является способ, который называется эквивалентным преобразованием моделей.

Определение 1.7. *Эквивалентным преобразованием называется преобразование одной модели ЛРД в другую — такую, что все доступы, разрешаемые первой, разрешаются и второй, при этом выполняются необходимые пост-действия, и такое же условие верно для запрещаемых доступов.*

Первым классом эквивалентных преобразований, которые используются для повышения производительности поиска необходимых правил, являются преобразования, направленные на построение общей области применимости модели. Эти преобразования состоят в сужении области применимости модели таким образом, чтобы полученная в результате преобразования модель была эквивалентной исходной. Для этого область применимости модели сужается так, чтобы ей удовлетворяли запросы на доступ, которые удовлетворяют областям применимости содержащихся в этой модели правил. В работе формализуется представленный класс преобразований и доказывается его эквивалентность.

Утверждение 1.5. *Пусть* Target *— условие на атрибуты, ограничивающее область применения модели,* Target_1, ..., Target_n *— условия на атрибуты, описывающие области применимости входящих в нее правил и моделей. Пусть* Additional *— условие на атрибуты, при этом при любых значениях атрибутов верны следствия* $\mathsf{Target}_1 \Rightarrow \mathsf{Additional}$, ..., $\mathsf{Target}_n \Rightarrow \mathsf{Additional}$.

В таком случае, преобразованная модель, содержащая те же правила и модели, с тем же алгоритмом комбинирования и пост-условиями, что и исходная, но имеющая функцию ограничения области $\mathsf{Target} \wedge \mathsf{Additional}$, *является эквивалентной изначальной.*

Вторым классом эквивалентных преобразований, рассматриваемых в работе, является разделение моделей. Эти преобразования производятся таким образом, что одна модель ЛРД делится на две, области применимости которых в их объединении порождают область применимости изначальной модели. Общая модель содержит эти две модели, причем алгоритмы комбинирования правил у общей модели, двух содержащихся в ней моделей и исходной модели совпадают. После этого в каждой из получившихся моделей оставляются лишь те правила, область применимости которых пересекается с областью применимости содержащей их модели. Полученные в результате таких операций две модели будут «в сумме» эквивалентны исходной. В работе в виде утверждения представляется формальное определение данного преобразования.

Утверждение 1.6. *Пусть* Target *— условие на атрибуты, ограничивающее область применения модели,* Target_1, ..., Target_n, Target_{n+1}, ..., Target_{n+k} *— условия на атрибуты, ограничивающие область определения входящих в нее правил и моделей. Пусть* Additional_1 *и* Additional_2 *— условия на атрибуты такие, что при любых значениях атрибутов истинно следствие* $\text{Additional}_1 \vee \text{Additional}_2 = \text{Target}$. *Кроме этого, пусть условие* $\text{Additional}_1 \wedge \text{Additional}_2 = \text{False}$ *справедливо при любых значениях атрибутов.*

Рассмотрим две модели, алгоритм комбинирования и пост-действия каждой из которых совпадают с исходной моделью. При этом область применимости первой модели — Additional_1, *а второй —* Additional_2. *В первой модели содержатся правила, области применимости которых не противоречат* Additional_1, *то есть, если область применимости правила* Target_i, *то условие* $\text{Target}_i \Rightarrow \neg\text{Additional}_1$ *должно быть ложно хотя бы при каких-то значениях входящих в него атрибутов. Во второй модели содержатся правила, области применимости которых не противоречат* Additional_2

Рассмотрим модель, которую назовем преобразованной. Такая модель содержит указанные выше две модели и алгоритм комбинирования совпадает с аналогичным алгоритмом исходной модели. В таком случае, преобразованная модель эквивалентна исходной.

Автором приведен сравнительно простой алгоритм эквивалентного преобразования моделей ЛРД, основанный на двух представленных выше классах преобразований. Этот алгоритм состоит из двух основных шагов:

- получение области допустимых значений для каждого атрибута субъекта, объекта или доступа;
- разделение на подмодели по принципу принадлежности атрибута диапазону значений.

Другим представленным методом повышения эффективности вычисления разрешенных доступов является использование кэширования. Кэш-массив — это ассоциативный массив, который ставит в соответствие запросу на доступ результат его вычисления. Такой способ повышения производительности в механизмах ЛРД крайне эффективен в тех случаях, когда доступ субъекта к объекту происходит существенно чаще изменения политики разграничения доступа. Под изменением политики понимается изменение модели ЛРД или атрибутов субъекта или объекта. В таких условиях кэширование позволяет заменить повторные проверки сложных условий при запросах на доступ простым поиском по кэш-массиву, который выполняется очень быстро. В работе приводится анализ наиболее эффективных способов выбора ключа кэширования и политики кэширования, а также структуры данных, используемой для кэш-массива.

Последним рассматриваемым в работе способом оптимизации механизмов ЛРД является генерация исполняемого кода. При использовании этого метода модель ЛРД при ее подключении в механизм разграничения доступа преобразуется в исполняемый код под архитектуру процессора целевой системы. После этого во вре-

мя работы системы, предоставляющей доступ, вызывается сгенерированный код. В рамках данной работы рассмотрен способ генерации исполняемого кода с помощью виртуальной машины Low Level Virtual Machine[10].

Пятая глава содержит описание методик проверки разработанных механизмов, а также результаты проведенных тестовых испытаний. Тесты делятся на три части: проверка выполнения функциональных требований, предъявляемых к механизмам ЛРД, основанным на моделях, описанных с помощью разработанного автором языка; оценка эффективности методов повышения производительности обработки запросов на предмет разрешения доступа, предложенных в предыдущей главе; оценка степени замедления работы программных средств, в которые внедрены разработанные механизмы.

Проверка выполнения функциональных требований, которые предъявляются к механизмам ЛРД, реализующим модели на предлагаемом языке, производилась с помощью разработанной автором библиотеки разграничения доступа.

Для тестирования было создано тестовое приложение, которое имитировало информационную систему с несколькими тестовыми субъектами и объектами с различными значениями атрибутов. В ходе проверки для каждой пары тестовых субъектов и объектов, а также для каждого из двух типов доступа («чтение» и «запись»), в тестовом приложении создавался запрос на доступ, который затем проверялся на допустимость с помощью представленной в предыдущей главе библиотеки разграничения доступа. Каждое из решений о доступе совпало с решением, которое должно было быть дано в соответствии с определением языка, представленным во второй главе.

Тесты всех трех методов оптимизации механизмов ЛРД, предназначенные для определения эффективности каждого из методов, проводились с помощью разработанной библиотеки разграничения доступа.

К сожалению, в режиме открытого использования отсутствуют большие модели ЛРД, использующиеся в сложно организованных, практически значимых информационных системах. С учетом этого обстоятельства, для тестирования представленных способов повышения производительности было разработано средство для автоматизированной генерации моделей ЛРД на разработанном автором языке.

В разных моделях используется различное количество правил, от 100 до 10000. Всего было сгенерировано три модели с количеством правил — 100, 1000 и 10000, соответственно. В качестве алгоритма комбинирования правил был выбран «приоритет запрещения», пост-действия отсутствовали.

Для тестирования использовалась реализация библиотеки, которая дополнялась одним из разработанных методов повышения производительности. Тестовое приложение подключало эту библиотеку и подавало поток запросов на доступ, после чего сравнивалось время вынесения решения о доступе в обоих случаях. Эффективность метода повышения производительности можно оценивать отношением времени вынесения решения о доступе в базовом случае к тому же времени, в случае оптимизированного механизма.

[10]Low Level Virtual Machine, 2009. Электронная версия печатного документа. http://www.llvm.org

В следующей таблице представлены результаты применения базовой и оптимизированной версии механизма ЛРД в случае применения метода повышения производительности «эквивалентные преобразования моделей».

Количество правил в модели ЛРД	Время на разрешение запроса на доступ, базовая реализация, мс	Время на разрешение запроса на доступ, оптимизированная реализация, мс
100	534	61
1000	5249	149
10000	52983	260

Легко заметить, что производительность оптимизированной реализации на порядки выше базовой, причем причем это различие возрастает при увеличении размера модели ЛРД.

В следующей таблице представлены результаты сравнения производительности базовой и оптимизированной версии механизма ЛРД в случае использования метода «преобразование в исполняемый код».

Количество правил в модели ЛРД	Время на разрешение запроса на доступ, базовая реализация, мс	Время на разрешение запроса на доступ, оптимизированная реализация, мс
100	550	180
1000	5293	2075
10000	51004	26421

Следует отметить, что производительность оптимизированной реализации в несколько раз выше базовой, хотя и не достигает таких показателей ускорения, как метод повышения производительности, указанная выше.

Для тестирования на предмет оценки степени замедления работы подлежащих защите средств вычислительной техники и/или автоматизированных систем выбрана ОС на базе ядра Linux. В ходе тестирования сравнивалась производительность прикладного обеспечения, запущенного в ОС с подключенным модулем разграничения доступа ядра, описанным ранее, и без него. Необходимо отметить, что само по себе использование надстройки RSBAC даже без подключенных модулей ЛРД вносит некоторое замедление в работу ядра, однако это влияние не является существенным.

В качестве модели ЛРД для тестирования выбрана модель Белла-ЛаПадулы, описание которой на представленном языке приведено в работе. В тестовую систему были предварительно добавлены 99 пользователей. Каждому из этих пользователей был задан атрибут level с произвольным значением. Кроме этого, в тестовую систему был добавлен один пользователь, который был владельцем процессов, запускавшихся во время тестирования. Ему были установлен самый низкий уровень

секретности в модели Белла-ЛаПадулы.

В качестве тестовых примеров были выбраны:

- компиляция ядра linux-2.6.28;
- копирование каталога с большим количеством вложенных подкаталогов и файлов (100 подкаталогов, в каждом из них еще 10 подкаталогов, в каждом из которых 100 файлов);
- копирование большого файла (100 Мб) блоками различных размеров.

Тестовый пример	Время выполнения с отключенным модулем разграничения доступа, с	Время выполнения с включенным модулем разграничения доступа, с
Компиляция ядра	215	217
Копирование каталога с большим числом вложенных файлов	11.5	12.1
Копирование большого файла блоками по 4 кб	0.83	0.87
Копирование большого файла блоками по 1024 кб	0.68	0.70

Как следует из представленных результатов, время выполнения задач с включенным в состав ядра ОС модулем разграниченяи доступа незначительно отличается от аналогичного времени в отсутствии такого модуля. В некоторых случаях эти замедления находятся в пределах погрешности измерения. Исходя из приведенных выше экспериментальных данных можно сделать вывод, что предложенные в диссертации методы оптимизации механизмов ЛРД, основанных на разработанном автором языке, позволяют достаточно эффективно использовать такие механизмы в современных программных системах.

В **заключении** перечисляются основные результаты диссертационной работы.

- Формально описан класса моделей логического разграничения доступа, который включает в себя достаточное широкий с практической точки зрения набор таких моделей, используемых в системах с повышенными требованиями к защищенности их ресурсов.
- Разработано и представлено логико-языковое средство (язык), позволяющее описывать модели, содержащиеся в рассматриваемом классе моделей логического разграничения доступа.
- Предложен алгоритм, позволяющий определить факт, включения одной модели логического разграничения доступа, описанной с помощью представленного языка, в другую. Доказана корректность этого алгоритма.

- Реализованы программные механизмы логического разграничения доступа, построенные на основе разработанного языка и предназначенные для их внедрения в ядро ОС Linux и в программы, написанные на языках C и Python.
- Рассмотрены и критически проанализированы несколько методов повышения эффективности обработки запросов на доступ в предлагаемых программных механизмах. На их основе разработаны способы практической реализации механизмов логического разграничения доступа, корректность которых доказательно обоснована.
- Проведены тестовые испытания разработанных механизмов логического разграничения доступа на предмет: проверки выполнения функциональных требований; оценки эффективности методов повышения производительности обработки запросов на предмет разрешения доступа; оценки степени замедления работы программных средств, в которых используются разработанные механизмы.

Автор выражает глубокую благодарность своему научному руководителю доктору физико-математических наук, профессору Валерию Александровичу Васенину за постановку задач и постоянное внимание к работе.

Публикации по теме диссертации

1. **О. О. Андреев. Интеграция моделей логического разграничения доступа, описанных на специализированном языке. // Информационные технологии. — М.: Новые технологии. — 2009. — № 12. — С. 29–33.**

2. О. О. Андреев. О методах оптимизации механизмов разграничения доступа, основанных на логико-языковых средствах. // Проблемы информатики. — Новосибирск: НГТУ. — 2009. — № 2.

3. В. А. Васенин, К. А. Шапченко, О. О. Андреев. Математические модели и механизмы логического разграничения доступа в операционной системе Linux: текущее состояние и перспективы развития. // Математика и безопасность информационных технологий. Материалы конференции в МГУ 25–28 октября 2006 г.. — М.: МЦНМО, 2007. С. 159–171. (О. О. Андрееву принадлежат результаты по исследованию способов описания моделей логического разграничения доступа в защищенных операционных системах).

4. О. О. Андреев. Язык описания моделей разграничения доступа и его реализация в ядре операционной системы Linux. // Математика и безопасность информационных технологий. Материалы конференции в МГУ 2–3 ноября 2005 г. — М.: МЦНМО, 2006, С. 305–-322.

5. Набор специализированных дистрибутивов ОС Linux с повышенными требованиями к защищенности. / В. А. Васенин, К. А. Шапченко, А. Н. Водомеров,

А. В. Инюхин, О. О. Андреев, В. Б. Савкин. // Свидетельство о государственной регистрации программы для ЭВМ № 2006613706. — 2006. (О. О. Андрееву принадлежат результаты по разработке механизмов логического разграничения доступа на основе логико-языковых способов описания моделей).

6. К. А. Шапченко, О. О. Андреев, В. Б. Савкин, А. А. Иткес. Специализированные дистрибутивы операционной системы Linux с повышенным уровнем защищенности. // Критически важные объекты и кибертерроризм. Часть 2. Аспекты программной реализации средств противодействия. / О. О. Андреев и др. Под ред. В. А. Васенина. — М.: МЦНМО, 2008. — 607 с. — С. 168–216. (О. О. Андрееву принадлежат результаты по разработке механизма разграничения доступа, включенного в ядро ОС Linux).

7. К. А. Шапченко, О. О. Андреев. Подход к управлению настройками механизмов безопасности в дистрибутивах ОС Linux. // Материалы Четвертой международной научной конференции по проблемам безопасности и противодействия терроризму. Московский государственный университет им. М. В. Ломоносова. 30–31 октября 2008 г. Том 2. Материалы Седьмой общероссийской научной конференции «Математика и безопасность информационных технологий» (МаБИТ-2008). — М.: МЦНМО, 2009. — С. 153–160. (О. О. Андрееву принадлежат результаты по разработке методов согласования моделей разграничения доступа на основе логико-языкового подхода).