

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение высшего профессионального образования «Уральский государственный университет им. А.М. Горького»

ИОНЦ «Информационная безопасность»

математико-механический факультет

кафедра алгебры и дискретной математики

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

Теоретические основы компьютерной безопасности

Учебное пособие

Автор: профессор кафедры алгебры
и дискретной математики
Н.А. Гайдамакин

Екатеринбург
2008

Гайдамакин Н.А.

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ
КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ**

2008 г.

ОГЛАВЛЕНИЕ

I. ИСХОДНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ	6
1.1. СОДЕРЖАНИЕ И ОСНОВНЫЕ ПОНЯТИЯ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ.....	6
1.1.1. История развития теории и практики обеспечения компьютерной безопасности	6
1.1.2. Содержание и структура понятия компьютерной безопасности	9
1.1.3. Общая характеристика принципов, методов и механизмов обеспечения компьютерной безопасности	12
1.2. УГРОЗЫ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ.	15
1.2.1. Понятие угроз безопасности, их классификация и идентификация.....	15
1.2.2. Методы оценивания угроз.....	19
1.3. ПОЛИТИКА И МОДЕЛИ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ	22
1.3.1. Понятие политики и моделей безопасности информации в компьютерных системах.....	22
1.3.2. Субъектно-объектная модель компьютерной системы в механизмах и процессах коллективного доступа к информационным ресурсам	23
1.3.3. Монитор безопасности и основные типы политик безопасности	28
1.3.4. Гарантирование выполнения политики безопасности	35
II. МОДЕЛИ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ	43
2.1. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ ДИСКРЕЦИОННОЙ ПОЛИТИКИ	43
2.1.1. Общая характеристика моделей дискреционного доступа. Пятимерное пространство Хартсона.....	43
2.1.2. Модели на основе матрицы доступа	46
2.1.3. Модели распространения прав доступа	50
2.1.3.1. Модель Харисона-Руззо-Ульмана (HRU-модель)	50
2.1.3.2. Модель типизованной матрицы доступа.....	53
2.1.3.3. Модель TAKE-GRANT.....	56
2.1.3.4. Расширенная модель TAKE-GRANT.....	67
2.2. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ МАНДАТНОЙ ПОЛИТИКИ	72
2.2.1. Общая характеристика политики мандатного доступа.....	72
2.2.2. Модель Белла-ЛаПадулы и ее расширения	77

2.2.3. Основные расширения модели Белла-ЛаПадулы	80
2.3. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ ТЕМАТИЧЕСКОЙ ПОЛИТИКИ	86
2.3.1. Общая характеристика тематического разграничения доступа	86
2.3.2. Тематические решетки.....	90
2.3.3. Модель тематико-иерархического разграничения доступа	106
2.4. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ РОЛЕВОЙ ПОЛИТИКИ	112
2.4.1. Общая характеристика моделей разграничения доступа на основе функционально-ролевых отношений.....	112
2.4.2. Формальная спецификация и разновидности ролевых моделей	115
2.4.3. Индивидуально-групповое разграничение доступа	125
2.5. АВТОМАТНЫЕ И ТЕОРЕТИКО-ВЕРОЯТНОСТНЫЕ МОДЕЛИ НЕВЛИЯНИЯ И НЕВЫВОДИМОСТИ.....	133
2.5.1. Понятие и общая характеристика скрытых каналов утечки информации.....	133
2.5.2. Модели информационного невмешательства и информационной невыводимости	135
2.5.3. Нейтрализация скрытых каналов утечки информации на основе технологий "представлений" и "разрешенных процедур"	140
2.6. МОДЕЛИ И ТЕХНОЛОГИИ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ	143
2.6.1. Общая характеристика моделей и технологий обеспечения целостности данных	143
2.6.2. Дискреционная модель Кларка-Вильсона	144
2.6.3. Мандатная модель Кена Биба	145
2.6.4. Технологии параллельного выполнения транзакций в клиент- серверных системах (СУБД)	148
2.7. МЕТОДЫ И ТЕХНОЛОГИИ ОБЕСПЕЧЕНИЯ ДОСТУПНОСТИ (СОХРАННОСТИ) ДАННЫХ.....	152
2.7.1. Резервирование, архивирование и журнализация данных.....	152
2.7.2. Технологии репликации данных.....	155
2.8. ПОЛИТИКА И МОДЕЛИ БЕЗОПАСНОСТИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ	159
2.8.1. Общая характеристика проблем безопасности в распределенных компьютерных системах.....	159
2.8.2. Модели распределенных систем в процессах разграничения доступа.....	162
2.8.3. Зональная модель разграничения доступа к информации в распределенных компьютерных системах.....	167

III. МЕТОДЫ АНАЛИЗА И ОЦЕНКИ ЗАЩИЩЕННОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ	178
3.1. ТЕОРЕТИКО-ГРАФОВЫЕ МОДЕЛИ КОМПЛЕКСНОЙ ОЦЕНКИ ЗАЩИЩЕННОСТИ.....	178
3.2. МЕТОДЫ АНАЛИЗА И ОПТИМИЗАЦИИ ИНДИВИДУАЛЬНО-ГРУППОВЫХ СИСТЕМ РАЗГРАНИЧЕНИЯ ДОСТУПА	181
3.2.1. Теоретико-графовая модель системы индивидуально-групповых назначений доступа к иерархически организованным объектам	181
3.2.2. Пространственно-векторная модель и характеристики системы рабочих групп пользователей	186
УПРАЖНЕНИЯ	190
I. По модели HRU	190
II. По модели TAM	192
III. По модели TAKE-GRANT	196
IV. По расширенной модели TAKE-GRANT	199
V. По модели Белла-ЛаПадуллы	201
VI. По модели тематического разграничения доступа на основе иерархических рубрикаторов.....	203
VII. По модели ролевого доступа при иерархически организованной системе ролей	206
VIII. По модели анализа индивидуально-групповых систем назначения доступа к иерархически организованным объектам доступа.....	209
ЛИТЕРАТУРА.....	212

I. ИСХОДНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

1.1. СОДЕРЖАНИЕ И ОСНОВНЫЕ ПОНЯТИЯ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

1.1.1. История развития теории и практики обеспечения компьютерной безопасности

Проблемы и задачи обеспечения безопасности информации, сохранности информационных ресурсов, охраны разного рода тайн возникли и решались задолго до компьютерной эры. Однако массовая компьютеризация всех сфер жизни, постепенный перевод основных информационных потоков в производстве и управлении в компьютерную форму и в компьютерные технологии обусловили качественные изменения той роли, которую играет безопасность и защита информации.

Современные компьютерные информационные технологии, предоставив новые, немыслимые ранее инфраструктуру и инструментарий жизнедеятельности и коммуникаций, качественно изменили и обострили проблему безопасности информации. Возможности несанкционированного доступа к информации, возможности несанкционированного получения и, как правило, без существенных организационных и материальных затрат огромных массивов данных, составляющих в ряде случаев ценнейшие корпоративные ресурсы, возможности мгновенного разрушения информационных ресурсов, хранящихся или использующихся в компьютерной форме, предопределили перевод задач обеспечения безопасности информации из разряда вспомогательных, обеспечивающих, в число основных приоритетов и условий.

В практическом плане задачи обеспечения безопасности компьютерной информации возникли в 70-х годах в связи с созданием и внедрением автоматизированных информационных систем в процессы информационного обеспечения деятельности крупных и средних предприятий и организаций. Потребовалась теоретическая база, программно-технические решения и механизмы обеспечения безопасности при коллективной обработке общих информационных ресурсов. Именно в то время появились первые работы по политике (методологии) и моделям защиты компьютерной информации. Такие исследователи, как Л.Дж. Хоффман, Р. Хартсон, М. Харрисон, У. Руззо, Дж. Ульман, Д. Белл, Л. ЛаПадула и др., внесли значительный вклад в создание теории безопасности компьютерной информации. В этом же ряду нельзя не упомянуть отечественных исследователей того периода, в частности В. Герасимова, В. Владиславского и В. Герасименко, внесших свой вклад в исследование теоретических основ компьютерной безопасности.

Сформировались три составляющих и, соответственно, три, хотя и взаимосвязанных, но различных направления защиты компьютерной информации – обеспечение конфиденциальности информации, обеспечение целостности данных, обеспечение сохранности и работоспособности данных.

Вероятно ввиду того, что наиболее сильные потребности в защите компьютерной информации в тот период исходили из военной сферы, основное внимание исследователей было сосредоточено на проблемах обеспечения конфиденциальности данных, основным ключом к разрешению которых были выбраны позаимствованные из "бумажной" сферы методы ограничения и разграничения доступа. В результате проблема разграничения доступа к данным с той поры и по сей день стала центральным элементом систем безопасности компьютерной информации.

К концу 70-х годов были разработаны исходные модели безопасности компьютерных систем, обеспечивающие те или иные из трех составляющих безопасности информации, и программно-технические решения построения и функционирования защищенных компьютерных систем, в частности, технологии и протоколы парольной аутентификации, криптографические методы и средства защиты информации и т. д. Одной из наиболее известных работ, представившей обобщенный анализ теоретических и практических аспектов защиты компьютерной информации того периода стала вышедшая в 1977 году книга Л.Дж. Хоффмана "Современные методы защиты информации".

Созданные в тот период модели дискреционного и мандатного разграничения доступа послужили методологической основой для разработки первых стандартов безопасности компьютерных систем, в частности, известной "оранжевой книги", впервые опубликованной в 1983 г. Кроме того, исходные модели дискреционного разграничения доступа, в частности модель Хариссона-Руззо-Ульмана, модель мандатного (полномочного) доступа Белла-ЛаПадулы явились основой для последующих исследований, повлекших разработку новых подходов к разграничению доступа в 80-е и 90-е годы. Свой вклад в развитие моделей разграничения доступа этого периода внесли Дж. МакЛин, К.Лендвер, Дж. Гоген, Дж. Мезигер, В. Варахаджан и др.

В 90-е годы к исследованиям процессов защиты компьютерной информации более активно подключились отечественные исследователи. В этом ряду следует отметить, прежде всего, труды В.А. Герасименко, разработавшего системно-концептуальный подход к обеспечению информационной безопасности автоматизированных систем обработки данных. А.А. Грушо и Е.Е. Тимонина представили доказательный подход к проблеме гарантированности защиты информации в компьютерной системе, а также провели математический анализ ряда задач и решений в теории защиты информации применительно к различным разновидностям компью-

терных систем. А.А. Грушо в сферу исследований были введены новые виды так называемых скрытых каналов утечки информации, основывающихся на использовании статистических характеристик работы компьютерной системы. В работах С.П. Расторгуева и А.Ю. Щербакова была представлена теория разрушающих программных воздействий, составившая теоретическую базу методов и механизмов борьбы с вредоносными программными средствами. Кроме того, А.Ю. Щербаковым на основе положений исходных моделей разграничения доступа была разработана стройная субъектно-объектная модель компьютерной системы, на базе которой сформированы фундаментальные для сферы защиты информации и, в особенности, для процессов разграничения доступа понятия информационных потоков и доступов в компьютерной системе.

Заметный вклад в исследование теоретических основ компьютерной безопасности внесли представители Санкт-Петербургской научной школы во главе с П.Д. Зегждой и научной школы Института криптографии, связи и информатики (ИКСИ) Академии ФСБ России во главе с Б.А. Погореловым. В частности, под руководством П.Д. Зегжды разработана таксономия брешей и изъянов в системах защиты компьютерных систем. В практических разработках специалистов Санкт-Петербургской школы представлен также целый ряд интересных технических решений по созданию защищенных компьютерных систем, в частности, организационно-иерархическая система разграничения доступа.

Представителями школы ИКСИ (П.Н. Девянин, Д.И. Правиков, А.Ю. Щербаков, С.Н. Смирнов, Г.В. Фоменков и др.) помимо исследований в сфере криптографической защиты информации был проведен анализ решений и механизмов защиты информации в основных разновидностях компьютерных систем, подготовлена целая серия учебных изданий, что позволило сформировать методическую базу подготовки специалистов в сфере компьютерной безопасности.

1.1.2. Содержание и структура понятия компьютерной безопасности

Понятие компьютерной безопасности является видовым по отношению к более широкому (родовому) понятию **"информационная безопасность"**, под которой понимается *состояние защищенности информационной сферы (предприятия, организации, общества, государства) от внутренних и внешних угроз*.

Методологический анализ родового понятия "информационная безопасность" показывает, что ключевыми являются следующие аспекты – информационная сфера (объект), угрозы (внутренние и внешние) и состояние защищенности (предмет объекта).

В этой логике сфера понятия "компьютерная безопасность" сужается до объекта, именуемого **"компьютерной системой"**, под которой будем понимать *человеко-машинную систему, представляющую совокупность электронно-программируемых технических средств обработки, хранения и представления данных, программного обеспечения (ПО), реализующего информационные технологии осуществления каких-либо функций, и информации (данных)*. В развитии этой логики, под **компьютерной безопасностью** понимается *состояние защищенности (безопасность) информации (данных) в компьютерных системах и безотказность (надежность) функционирования компьютерных систем*. В результате составляющими компьютерной безопасности выступают *безопасность информации (данных), накапливаемых, обрабатываемых в компьютерной системе, и безотказность (безотказность, надежность) функций КС*.

Содержательный анализ самого понятия **"информация"** (*сведения (сообщения, данные) независимо от формы их представления¹*), особенностей процессов и технологий ее сбора, обработки, хранения, представления и выдачи показывает, что безотносительно к функционально-содержательной стороне работы с информацией (данными) понятие **"безопасность информации"** включает три составляющих:

- обеспечение конфиденциальности;
- обеспечение целостности;
- обеспечение доступности.

При этом под **конфиденциальностью** информации понимается *специфическое свойство отдельных категорий (видов) информации, которое субъективно устанавливается ее обладателем, когда ему может быть причинен ущерб от ознакомления с информацией неуполномоченных на то лиц, при условии того, что обладатель принимает меры по организации доступа к информации только уполномоченных лиц*. Таким образом, обеспечение безопасности информации в КС означает, в первую очередь, обес-

¹ ФЗ "Об информации, информационных технологиях и о защите информации".

печение ее конфиденциальности (если характер информации является таковым, т.е. конфиденциальным), заключающееся в обеспечении такого порядка работы с информацией, когда она известна только определенному установленному кругу лиц (пользователей КС).

Под **целостностью** информации (данных) понимается *неискаженность, достоверность, полнота, адекватность и т.д. информации, т.е. такое ее свойство, при котором содержание и структура данных определены и изменяются только уполномоченными лицами и процессами*. Таким образом, обеспечение безопасности информации в КС означает в т.ч. такой порядок и технологию работы с ней, когда информация изменяется, модифицируется только уполномоченными лицами и в процессах ее передачи, хранения не возникают (устраняются) искажения, ошибки.

И, наконец, под [правомерной] **доступностью** информации (данных) понимается *такое свойство информации, при котором отсутствуют препятствия доступа к информации и закономерному ее использованию владельцем или уполномоченными лицами*. В результате безопасность информации в КС, в т.ч., обеспечивается ее сохранностью, способностью к восстановлению при сбоях и разрушениях, а также в отсутствии препятствий работы с ней уполномоченных лиц.

Важно подчеркнуть, что только одновременное обеспечение всех трех составляющих (конфиденциальности, целостности и доступности) дает состояние безопасности информации. Термин "защищенность" в большинстве случаев является тождественным термину "безопасность".

Второй стороной компьютерной безопасности в рамках приведенного выше определения является безопасность (безотказность, надежность) функций компьютерных систем.

Суть и особенности такого специфического инструментария человеческой деятельности, как "компьютерные системы", в свою очередь, определяют две составляющие **безопасности функций** КС:

- обеспечение безотказности реализации функций;
- обеспечение аутентичности реализации функций.

Первая составляющая определяется *безотказностью оборудования* (технических средств обработки, хранения, передачи и представления информации) и *безотказностью программного обеспечения* (отсутствие сбоев в работе программного обеспечения).

Вторая составляющая (аутентичность функций) определяется *целостностью ПО и целостностью программно-аппаратной конфигурации КС* (параметров, настройки, состава ПО и оборудования).

При этом следует отметить, что две составляющие компьютерной безопасности являются взаимозависимыми. В частности, при нарушении безопасности информации в КС (нарушении конфиденциальности, целостности и/или доступности данных) в большинстве случаев нарушается (не обеспечивается) безопасность функций КС. Однако обратное в общем слу-

чае неверно. Иначе говоря, информация КС может находиться в безопасном состоянии с т.зр. ее конфиденциальности, целостности и сохранности, но в результате сбоев оборудования или ПО, нарушения целостности ПО или целостности программно-аппаратной конфигурации, функции КС не будут реализовываться или будут реализовывать неадекватно.

Следует также отметить, что в силу исторических особенностей развития электронно-вычислительной техники, две составляющие компьютерной безопасности рассматривались и развивались в определенной степени независимо и параллельно, и более того, вторая составляющая (безопасность функций) рассматривалась в контексте обеспечения надежности вычислительной техники (оборудования и программного обеспечения). Исходя из этого, и в литературе, и в стандартах, в т.ч. в настоящее время под компьютерной безопасностью понимается в первую очередь (и в основном) первая ее составляющая, т.е. безопасность информации в КС.

Этого же подхода будем придерживаться и в рамках данного курса, поскольку методы и механизмы обеспечения функций КС в контексте безотказности оборудования и программного обеспечения являются предметом рассмотрения других дисциплин.

Таким образом, предметом изложения в дальнейшем будут методы, механизмы, их математическая формализация (модели), которые обеспечивают при воздействии угроз конфиденциальность, целостность и [правомерную] доступность информации в компьютерных системах.

1.1.3. Общая характеристика принципов, методов и механизмов обеспечения компьютерной безопасности

На основе анализа теоретических и практических аспектов обеспечения компьютерной безопасности можно выделить ряд *общих принципов* создания и эксплуатации защищенных компьютерных систем (в которых обеспечивается безопасность информации).

Принцип *разумной достаточности*. Внедрение в архитектуру, в алгоритмы и технологии функционирования КС защитных механизмов, функций и процедур объективно вызывает дополнительные затраты, издержки при создании и эксплуатации, ограничивает, снижает функциональные возможности КС и параметры ее эффективности (быстродействие, задействуемые ресурсы), вызывает неудобства в работе пользователям КС, налагает на них дополнительные нагрузки и требования — поэтому защита должна быть разумно достаточной (на минимально необходимом уровне).

Принцип *целенаправленности*. Заключается в том, что применяемые меры по устранению, нейтрализации (либо обеспечению снижения потенциального ущерба) должны быть направлены против перечня угроз (опасностей), характерных для конкретной КС в конкретных условиях ее создания и эксплуатации.

Принцип *системности*. Выбор и реализация защитных механизмов с должны производиться с учетом системной сути КС, как организационно-технологической человеко-машинной системы, состоящей из взаимосвязанных, составляющих единое целое функциональных, программных, технических, организационно-технологических подсистем.

Принцип *комплексности*. При разработке системы безопасности КС необходимо использовать защитные механизмы различной и наиболее целесообразной в конкретных условиях природы — программно-алгоритмических, процедурно-технологических, нормативно-организационных, и на всех стадиях жизненного цикла — на этапах создания, эксплуатации и вывода из строя.

Принцип *непрерывности*. Защитные механизмы КС должны функционировать в любых ситуациях в т.ч. и внештатных, обеспечивая как конфиденциальность, целостность, так и сохранность (правомерную доступность).

Принцип *управляемости*. Подсистема безопасности КС должна строиться как система управления — объект управления (угрозы безопасности и процедуры функционирования КС), субъект управления (средства и механизмы защиты), среда функционирования, обратная связь в цикле управления, целевая функция управления (снижение риска от угроз безопасности до требуемого (приемлемого) уровня), контроль эффективности (результативности) функционирования.

Принцип *сочетания унификации и оригинальности*. С одной стороны с учетом опыта создания и применения АИС, опыта обеспечения безопасности КС должны применяться максимально проверенные, стандартизированные и унифицированные архитектурные, программно-алгоритмические, организационно-технологические решения. С другой стороны, с учетом динамики развития ИТ, диалектики средств нападения и развития должны разрабатываться и внедряться новые оригинальные архитектурные, программно-алгоритмические, организационно-технологические решения, обеспечивающие безопасность КС в новых условиях угроз, с минимизацией затрат и издержек, повышением эффективности и параметров функционирования КС, снижением требований к пользователям.

Организационно-технологический и человеко-машинный характер природы КС определяют обширный набор методов и механизмов обеспечения информационной безопасности (см. рис. 1.1).



Рис.1.1. Систематика методов и механизмов обеспечения компьютерной безопасности

В первую очередь можно выделить ряд методов и механизмов, непосредственно обеспечивающих конфиденциальность, целостность и доступность данных, такие как разграничение доступа к данным, контроль и управление информационной структурой данных, контроль и обеспечение

ограничений целостности данных, механизмы криптографического скрывания данных (шифрования), механизмы ЭЦП, обеспечивающие целостность данных в процессах их передачи и хранения, а также механизмы контроля и удаления остаточной информации на носителях данных после завершения их обработки и в освобождаемых областях оперативной памяти.

Также важнейшее значение для обеспечения компьютерной безопасности имеют методы и механизмы общесистемного характера, которые можно разделить на общеархитектурные и инфраструктурные с т.зр. программно-технической структуры современных КС.

Основополагающими среди общеархитектурных являются механизмы идентификации и аутентификации, обеспечивающие исходный и обязательный рубеж безопасности в КС, методы и механизмы управления памятью, изоляции процессов и управления транзакциями в клиент-серверных системах.

Не менее важное значение имеют методы и механизмы инфраструктурного характера, в особенности для обеспечения информационной безопасности в распределенных КС – контроль и управление программно-технической конфигурацией КС, управление сеансами работы пользователей, управление доступом пользователей с рабочих станций КС, управление (контроль) сетевыми соединениями в КС, управление инфраструктурой сертификатов криптоключей, обеспечивающих механизмы шифрования данных и электронно-цифровой подписи.

Обязательными для обеспечения информационной безопасности КС, находящими отражение в стандартах защищенности, имеют методы и механизмы обеспечивающего (профилактирующего) характера, среди которых, в первую очередь следует отметить методы протоколирования и аудита событий, методы и механизмы резервирования и архивирования, журнализации процессов изменения данных. Следует также отметить важность механизмов профилактики носителей данных их учета и контроля в организационно-технологическом контуре КС. Кроме того, человеко-машинный характер природы КС как особого инструментария деятельности предопределяет существенное значение для обеспечения информационной безопасности нормативно-организационной регламентации использования (эксплуатации) КС, процедур обучения, нормативно-административного побуждения и принуждения пользователей по вопросам обеспечения безопасности.

1.2. УГРОЗЫ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

1.2.1. Понятие угроз безопасности, их классификация и идентификация

Понятие угрозы безопасности является наряду с понятием безопасности информации краеугольным камнем в сфере компьютерной безопасности, поскольку выбор защитных механизмов определяется исходя из целей устранения, нейтрализации угроз, снижения последствий (ущерба) от их возможного проявления и воздействия.

Под *угрозой безопасности* КС будем понимать совокупность условий и факторов, определяющих потенциальную или реально существующую опасность нарушения конфиденциальности, целостности, [правоммерной] доступности информации и/или снижения надежности [безотказности и аутентичности] реализации функций КС.

Сложная и многогранная природа рассматриваемых объектов безопасности (компьютерные системы и информация в них) определяют огромное число факторов, условий, которые составляют или могут составить угрозы безопасности. Как и в других подобных случаях, когда рассматривается большое или неопределенное количество объектов, важную познавательную и прикладную роль играет *систематизация* (приведение в систему, т.е. в нечто целое, представляющее собой единство закономерно расположенных и находящихся во взаимной связи частей; выстраивание в определенный порядок).

Частным случаем систематизации выступает *классификация* – последовательное деление понятий (предметов исследования), проводимое по характеристикам и параметрам, существенным с точки зрения исследовательской задачи.

На рис.1.2 приведена классификация угроз компьютерной безопасности по различным критериям (основаниям) – по природе происхождения, по направлениям осуществления, по объекту воздействию, по способу осуществления и т.д.

Как видно из приведенной схемы, действительно угрозы компьютерной безопасности характеризуются различной природой, признаками и т.д., и поэтому классификационные схемы играют важную познавательско-исследовательскую роль.

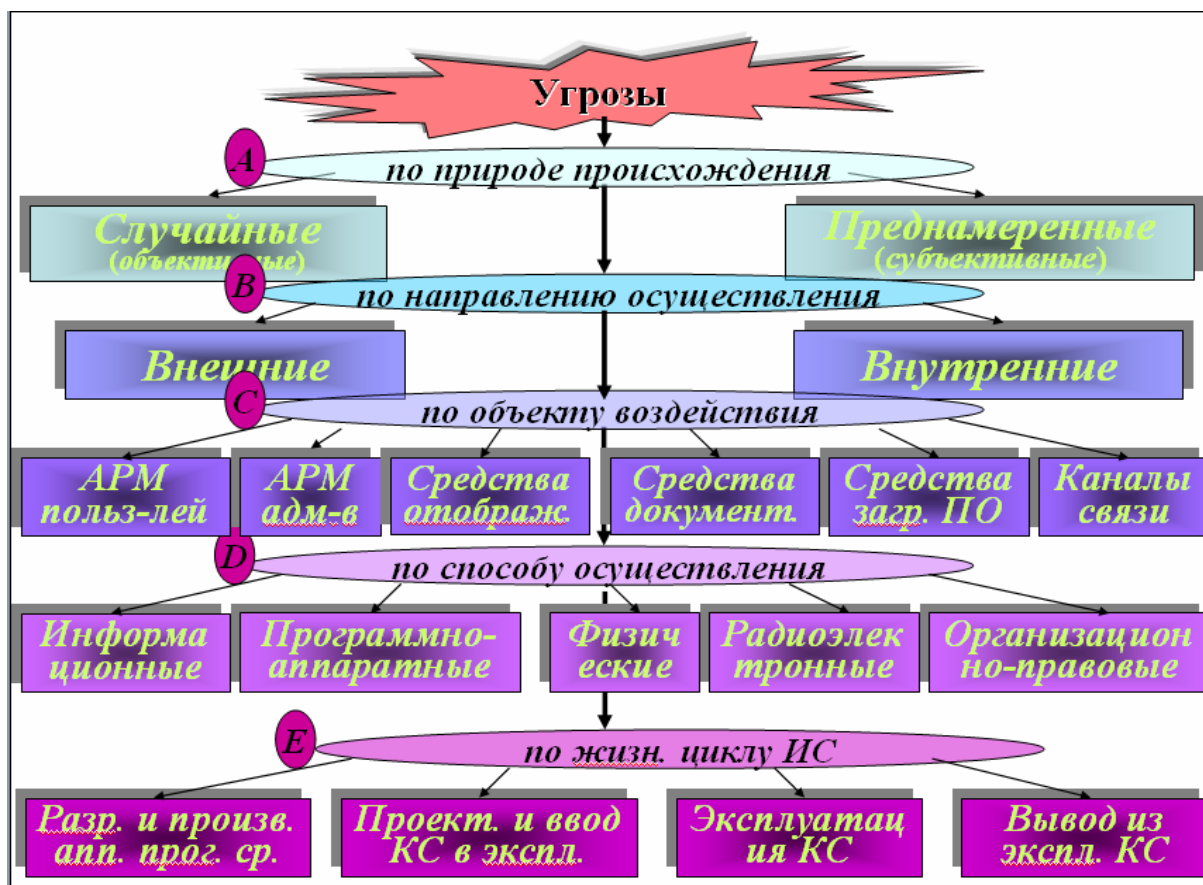


Рис.1.2. Схема классификации угроз компьютерной безопасности по различным критериям (основаниям)

Различают два вида классификационного (систематизированного) деления – таксономическое (родоводовое), представленное на рис.1.2, и мереологическое (по принципу "часть-целое").

При таксономическом делении предмет исследования (все многообразие, вся возможная совокупность элементов/экземпляров предмета исследования) разделяется на классы-таксоны, так, чтобы любой экземпляр/элемент обязательно попал какой-либо класс (т.е. для него нашелся бы класс), и так, чтобы один экземпляр/элемент попадал бы только и только в один класс (т.е. так, чтобы одновременно не попадал в два или несколько классов).

В теоретико-множественной трактовке таксономическое классификационное деление означает разбиение множества на два или более непересекающихся подмножеств, объединение которых дает полное исходное множество, а пересечение – пусто:

$$O = O_1 \cup O_2, \quad O_1 \cap O_2 = \emptyset.$$

Принцип разбиения множества на классы называется **критерием** (основанием) **классификации**. Его выбор/обоснование определяется при-

родой объектов исследования, а также самой исследовательской целью (задачей), и является наиболее сложной, как правило, неформализуемой стороной построения классификационных схем. Иначе говоря, критерий (основание) классификации определяется на эвристической основе.

Во многих случаях систематизация объектов исследования по таксономически-классификационным схемам включает многоуровневое деление, когда выделенные классы, в свою очередь, разбиваются на подклассы и т.д. При этом должны обеспечиваться два правила таксономического деления – в рамках данного класса подклассы должны выделяться на основании одного и того же критерия (ошибка классификации – "сбивчивое деление", когда один подкласс выделен по одному основанию, другой подкласс этого же класса по другому основанию); и второе правило – основания для разбиения на подклассы в разных классах должны быть одноуровневого характера (ошибка классификации – "скачок в делении").

Кроме теоретико-познавательных функций классификационные схемы (в данном случае классификационные схемы угроз) обеспечивают важные прикладные функции, а именно – полноту анализа при идентификации угроз для конкретной компьютерной системы. Поясним сказанное. Поскольку правильно построенная таксономически-классификационная схема обладает свойством полноты, то, анализируя на ее основе наличие/отсутствие угроз соответствующих классов, подклассов и т.д., можно обеспечить полноту анализа при формировании подмножества угроз для данной КС. Кроме того, классификационные схемы помогают также систематизировать выбор защитных мер, которые могут устранять сразу целый класс (с соответствующими подклассами) угроз.

Подчеркнем еще раз, что вышесказанное справедливо при обеспечении полноты и правильности классификационных схем, что подлежит обоснованию/доказательству.

Отмеченный прикладной аспект классификации угроз обуславливает нормативно-методическое закрепление составленных и апробированных в теоретическом и практическом отношении классификационных схем угроз в специальных стандартах. Сложившаяся терминология в этой области использует термин "каталогизация" угроз.

Таким образом, **каталогизация** угроз представляет собой составление и закрепление в стандартах таксономически-классификационных схем угроз, которые используются для идентификации угроз в процессах выбора защитных мер, методов и механизмов обеспечения безопасности при создании и эксплуатации защищенных компьютерных систем.

В качестве примера можно привести российский ГОСТ 51275-99 "Защита информации. Объект информатизации. Факторы, воздействующие на информацию", основанный на таксономической классификации всех возможных факторов, способных [негативно] воздействовать на информацию в компьютерных системах. По ГОСТ угрозы делятся на классы,

подклассы, группы, подгруппы, виды, подвиды. На рис.1.3. приведена схема классификации угроз/факторов до 3-го уровня деления (т.е. до групп факторов).

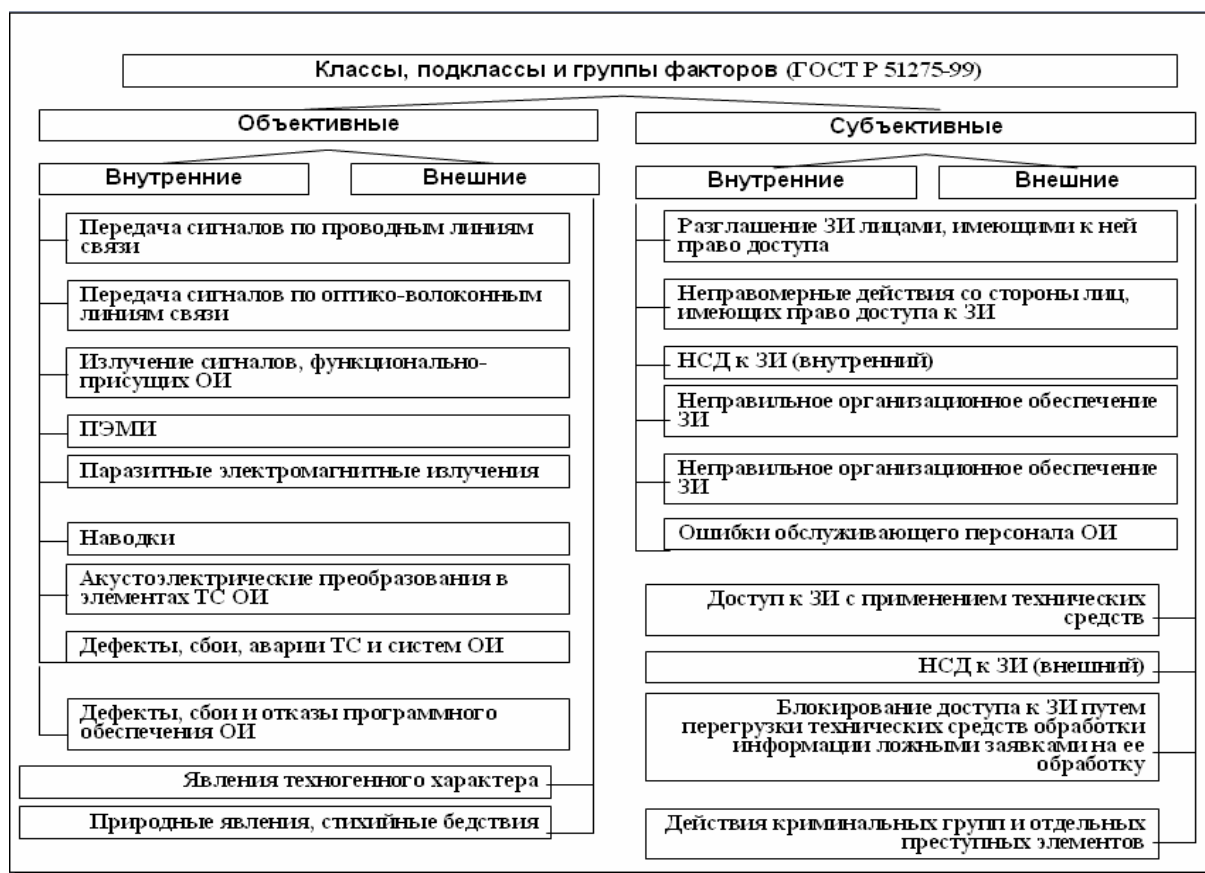


Рис.1.3. Схема классификации угроз по ГОСТ Р 51275-99

На основании каталогов, представляющие все поле угроз (все множество угроз) осуществляется определение тех из них, которые характерны, актуальны для конкретной компьютерной системы и конкретных условий ее функционирования. Данный процесс называется **идентификацией** угроз.

Идентификация угроз включает выявление угроз для конкретной компьютерной системы (из всех возможных), присвоение выявленным угрозам уникальных идентификаторов и **спецификацию** (описание) угроз.

Как правило, стандартами безопасности устанавливаются требования к спецификации выявленных угроз по определенному набору параметров, среди которых кроме идентификатора, требуется указать источник (природу происхождения) угрозы, активы (объекты КС), которые могут быть подвергнуты воздействию угрозы (на которые направлена угроза), способы осуществления угрозы, возможные уязвимости, которые м.б. использованы для реализации угрозы.

1.2.2. Методы оценивания угроз

Помимо идентификации и спецификации угроз важное значение для выбора и обоснования защитных мер играет **оценивание** угроз, под которым понимается формирование оценок идентифицированных и специфицированных угроз с точки зрения потерь, ущерба, возможных от реализации (воздействия) соответствующих угроз.

Основными факторами оценки являются возможность реализации угрозы и возможный ущерб от реализации угрозы. Общая схема оценки приведена на рис.1.4.

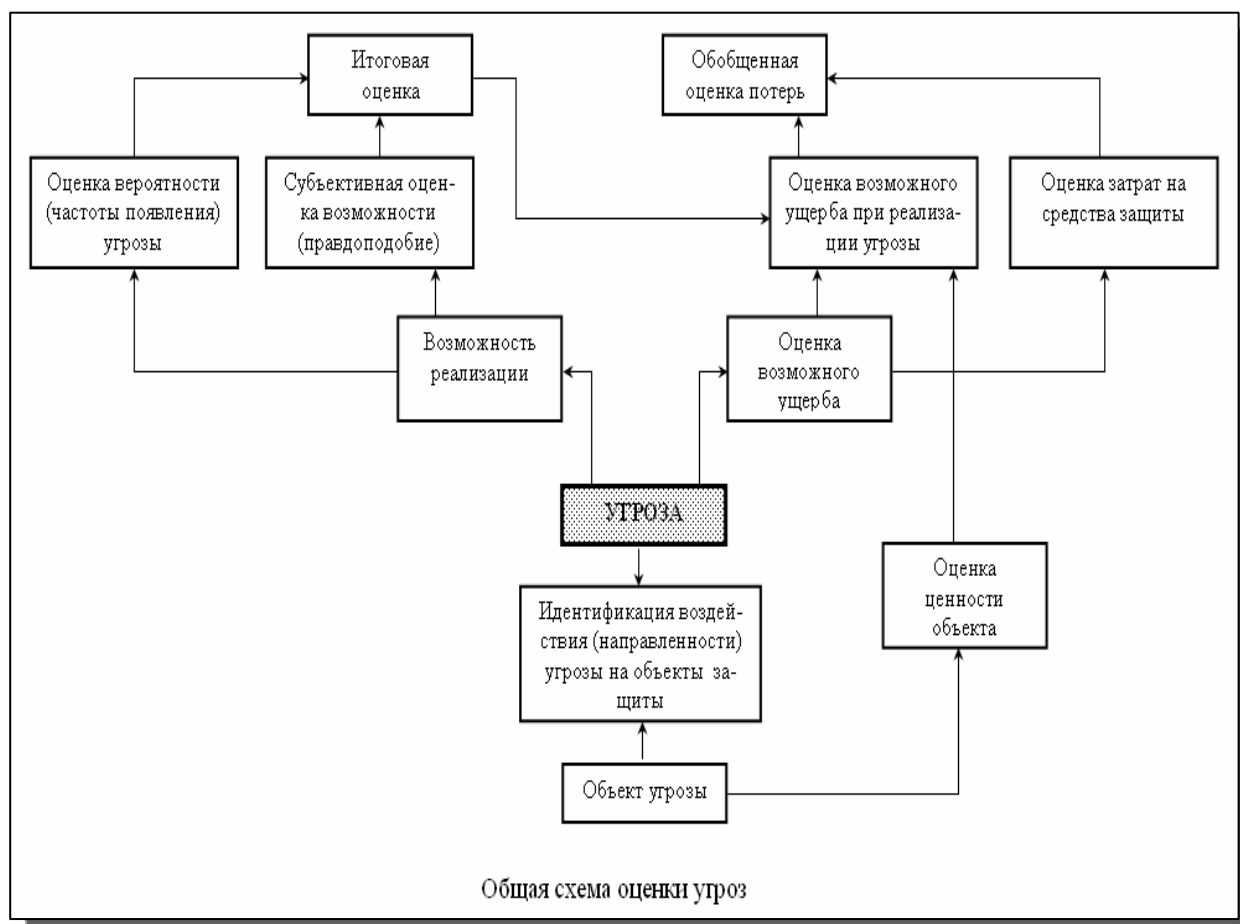


Рис.1.4. Общая схема оценивания угроз

Основными трудностями при оценивании угроз являются проблемы выбора шкал и способов оценки по отмеченным факторам.

Естественным параметром и шкалой оценки возможности реализации угроз является оценка вероятности их реализации. Природа некоторых видов угроз позволяет вычислять эти вероятности на основе известных соответствующих физических закономерностей (**априорный подход**), но все же в большинстве случаев построить и обосновать аналитические соотношения для вычисления вероятностей реализации угроз не представляется возможным. К примеру, на основе Пуассоновского распределения вероят-

ности моторных ошибок человека-оператора при вводе информации с клавиатуры вероятность угрозы, обусловленной данным фактором составляет $2 \cdot 10^{-2} \dots 4 \cdot 10^{-3}$. Данная оценка дает возможность определить важный параметр защитных мер, в частности, количество символов пароля и количество попыток его набора, при котором в рамках задания определенного уровня значимости ошибки 2-го рода (ошибка правильной аутентификации) легальный пользователь войдет в систему.

В некоторых случаях возможен *апостериорный подход*, основанный на накопленной статистике проявления соответствующей угрозы в данной или подобной компьютерной системе (в подобных условиях). Оценки вероятности реализации угрозы при этом вычисляются на основе методов статистических оценок.

Альтернативой аналитическому и статистическому подходу является *метод экспертных оценок*, широко используемый для оценок сложных, неформализуемых объектов.

Суть метода экспертных оценок заключается в том, что в качестве инструментария оценок (в качестве измерительного прибора) выступают специалисты-эксперты, которые на основе профессионального опыта, глубокого представления многокомпонентной природы оцениваемых объектов, дают эвристические оценки по одному или группе параметров.

В кратком изложении методика экспертных оценок включает следующие этапы.

1. Отбор экспертов (формальные и неформальные требования к специалистам-экспертам, метод «снежного кома», когда известного специалиста просят назвать других ему известных специалистов, в свою очередь, опрашивают их, и т.д. когда множество экспертов прекращает расширяться, на практике количество экспертов 10-12).
2. Выбор параметров, по которым оцениваются объекты (при этом определяются сущностные параметры оценивания, которые должны выражать природу оцениваемых объектов и быть независимыми друг от друга, определяются веса параметров).
3. Выбор шкал оценивания и методов экспертного шкалирования). Применяются порядковые, ранговые шкалы, интервальные, абсолютные и др. шкалы. В качестве методов шкалирования выступают ранжирование объектов по предпочтительности выраженности оцениваемого параметра (порядковая шкала оценки), попарные оценки сравнительной предпочтительности во всех возможных парах оцениваемых объектов, и непосредственная оценка выраженности оцениваемого параметра (например, эксперты непосредственно дают оценку вероятности реализации угроз, в других случаях на основе специальных балльных шкал оценки).
4. Выбор и осуществление процедуры опроса экспертов (с непосредственным взаимодействием экспертов или без взаимодействия, т.н. итерационный метод опроса "Дельфи", когда эксперты непосредственно не взаимодействуют, но после каждого тура опроса им сооб-

щают усредненные оценки прежнего тура и просят на этой основе скорректировать свои прежние оценки, исключая тем самым влияние на результаты опроса мнений конкретных "авторитетов", и т.д.).

5. Агрегирование оценок, анализ их устойчивости и согласованности, осуществляемые на основе подходов, подобных методам обработки статистических данных.

Следует отметить, что экспертные оценки, несмотря на их "субъективность" на основе хорошо подобранных экспертных комиссии, правильно установленных методов шкалирования и опроса, при соответствующей обработке дают результаты, действенность которых многократно апробированы в крупных проектах и процедурах, не допускающих другие, в особенности, аналитические и статистические подходы.

1.3. ПОЛИТИКА И МОДЕЛИ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

1.3.1. Понятие политики и моделей безопасности информации в компьютерных системах

Фундаментальным понятием в сфере защиты информации компьютерных систем является политика безопасности. Под ней понимают интегральную совокупность норм и правил, регламентирующих процесс обработки информации, выполнение которых обеспечивает состояние защищенности информации в заданном пространстве угроз. Формальное выражение политики безопасности (математическое, схемотехническое, алгоритмическое и т. д.) называют моделью безопасности.

Модели безопасности играют важную роль в процессах разработки и исследования защищенных компьютерных систем, так как обеспечивают системотехнический подход, включающий решение следующих важнейших задач:

- выбор и обоснование базовых принципов архитектуры защищенных компьютерных систем (КС), определяющих механизмы реализации средств и методов защиты информации;
- подтверждение свойств (защищенности) разрабатываемых систем путем формального доказательства соблюдения политики безопасности (требований, условий, критериев);
- составление формальной спецификации политики безопасности как важнейшей составной части организационного и документационного обеспечения разрабатываемых защищенных компьютерных систем.

По сути модели безопасности являются исходным связующим элементом в триаде "Заказчик (Потребитель)-Разработчик (Производитель)-Эксперт (Аудитор)". На основе моделей безопасности заказчики могут формулировать те требования к защищенным КС, которые соответствуют политике безопасности, технологическим процессам обработки информации, принятым в своих организациях и предприятиях. Разработчики на основе моделей безопасности формируют технико-технологические требования и программно-технические решения по разрабатываемым системам. Эксперты, основываясь на моделях безопасности, строят методики и спецификации оценки защищенности конкретных систем, осуществляют сертификацию разработанных систем по требованиям защиты информации.

1.3.2. Субъектно-объектная модель компьютерной системы в механизмах и процессах коллективного доступа к информационным ресурсам

Большинство моделей разграничения доступа основывается на представлении КС как совокупности субъектов и объектов доступа.

Приведем основные положения субъектно-объектной формализации компьютерных систем в аспекте безопасности информации.

1. В КС действует дискретное время.
2. В каждый фиксированный момент времени t_k КС представляет собой конечное множество элементов, разделяемых на два подмножества:
 - подмножество субъектов доступа S ;
 - подмножество объектов доступа O .

Определение 1.3.1. *Под субъектом доступа понимается активная сущность КС, которая может изменять состояние системы через порождение процессов над объектами, в том числе, порождать новые объекты и инициализировать порождение новых субъектов.*

Определение 1.3.2. *Под объектом доступа понимается пассивная сущность КС, процессы над которой могут в определенных случаях быть источником порождения новых субъектов.*

В модели предполагается наличие априорно безошибочного механизма различения активных и пассивных сущностей (т. е. субъектов и объектов) по свойству активности, что можно проиллюстрировать интуитивно понятными различиями между файлом с кодом программы и исполняемой (запущенной) программой, порождающей процессы над объектами системы.

Кроме того, предполагается также, что в любой момент времени t_k , в том числе и в начальный, множество субъектов доступа не пусто.

3. Пользователи КС представлены одним или некоторой совокупностью субъектов доступа, действующих от имени конкретного пользователя.

Определение 1.3.4. *Под пользователем КС понимается лицо, внешний фактор, аутентифицируемый некоторой информацией, и управляющий одним или несколькими субъектами, воспринимающий объекты и получающий информацию о состоянии КС через субъекты, которыми он управляет.*

Таким образом, в субъектно-объектной модели понятия субъектов доступа и пользователей не тождественны. Предполагается также, что пользовательские управляющие воздействия не могут изменить свойств самих субъектов доступа, что в общем случае не соответствует реальным КС, в которых пользователи могут изменять свойства субъектов, через из-

менение программ (исполняемых файлов). Однако подобная идеализация позволяет построить четкую схему процессов и механизмов доступа, а угрозы безопасности, возникающие вследствие подобных реалий, рассматривать в контексте гарантий выполнения политики безопасности (политики разграничения доступа) через механизмы неизменности свойств КС (т. н. изолированная программная среда).

4. Субъекты КС могут быть порождены из объектов только активной сущностью (другим субъектом).

Определение 1.3.5. Объект o_i называется источником для субъекта s_m если существует субъект s_j , в результате воздействия которого на объект o_i возникает субъект s_m .

Соответственно, субъект s_j , называется активизирующим для субъекта s_m .

Для описания процессов порождения субъектов доступа вводится следующее обозначение:

$Create(s_j, o_i) \rightarrow s_m$ – "из объекта o_i порожден субъект s_m при активизирующем воздействии субъекта s_j ".

Create называют операцией порождения субъектов. Отметим также, что ввиду того, что в КС действует дискретное время, то под воздействием активизирующего субъекта в момент времени t_k , новый субъект порождается в момент времени t_{k+1} .

Результат операции **Create** зависит как от свойств активизирующего субъекта, так и от свойств объекта-источника. К примеру, субъект пользователя в виде работающего текстового редактора при открытии файла в формате другого текстового редактора может быть не способным активизировать находящиеся там процедуры обработки данных, а в лучшем случае быть способным только их прочесть. Другой пример – командный интерпретатор ОС по команде пользователя не может запустить на исполнение текстовый файл и создать таким образом субъект пользователя. В таких случаях $Create(s_j, o_i) \rightarrow \emptyset$.

Анализ архитектуры вычислительной системы фон Неймана, на базе которой функционируют КС, показывает, что введенное понятие субъекта доступа и процесса его порождения требует связывания субъекта с определенным объектом (объектами), отражающим состояние действующего субъекта в различные моменты времени.

Определение 1.3.6. Объект o_i в момент времени t_k ассоциирован с субъектом s_m , если состояние объекта повлияло на состояние субъекта в следующий момент времени t_{k+1} (т. е. субъект s_m использует информацию, содержащуюся в объекте o_i).

Из определения 1.3.6 следует, что объект-источник в момент порождения субъекта является ассоциированным с ним, а в последующие моменты времени может перестать быть или остаться ассоциированным с ним. К примеру, исполняемые файлы программ являются ассоциированными с

субъектом только в момент его порождения, так как в процессе инициализации (запуска) код программы из исполняемого файла копируется в специальную область памяти (сегмент кода), откуда впоследствии собственно и извлекаются команды-инструкции выполнения программы. Следовательно, файл на диске с исполняемым кодом программы после ее запуска перестает быть ассоциированным с субъектом, порожденным запуском программы. Напротив, в некоторых СУБД со встроенными системами программирования интерпретаторского типа команды-инструкции по обработке данных в каждый момент времени могут извлекаться непосредственно из файлов базы данных, располагаемых на диске. В этом случае, соответственно, файл базы данных продолжает оставаться ассоциированным с субъектом, порожденным открытием (запуском) соответствующего файла базы данных.

Активная сущность субъектов доступа заключается в их возможности осуществлять определенные действия над объектами, что объективно приводит к возникновению *потоков информации*. Исходя из этого, центральным положением субъектно-объектной модели является следующее.

5. Все процессы безопасности в КС описываются доступами субъектов к объектам, вызывающими потоки информации.

Определение 1.3.7. *Потоком информации между объектом o_i и объектом o_j называется произвольная операция над объектом o_j , реализуемая в субъекте s_m и зависящая от объекта o_i .*

Для описания потоков вводят следующее обозначение:

$Stream(s_m, o_i) \rightarrow o_j$ – "поток информации от объекта $o_i(o_j)$ к объекту $o_j(o_i)$ в субъекте s_m (через субъект s_m)".

Поток может осуществляться в виде различных операций над объектами – чтение, изменение, удаление, создание и т. д. Объекты o_i и o_j , участвующие в потоке, могут быть как источниками, так и приемниками информации, могут быть как ассоциированными с субъектом, так и неассоциированными, а также могут быть пустыми ($\emptyset$) объектами (например, при создании или удалении файлов). Следует особо подчеркнуть, что согласно *определению 1.3.7* потоки информации могут быть только между объектами, а не между субъектом и объектом, в виду того, что субъект это активная сущность, т. е. действия, процессы и т. д., а информация – пассивная сущность, которая может размещаться, извлекаться, порождаться, изменяться и т. д. только в объектах. Активная роль субъекта заключается в самой реализации потока, в его локализации в субъекте (через субъект), в том числе, через задействование в потоке ассоциированных с субъектом объектов (например, буферов оперативной памяти). В этом отношении более детальный анализ понятия субъектов доступа, *определений 1.35 и 1.3.6* показывает, что ассоциированные объекты могут быть разделены на два вида:

- функционально-ассоциированные объекты;
- ассоциированные объекты-данные.

Функционально-ассоциированные объекты влияют (определяют) на сами процессы субъекта (например, состояние сегмента кода определяет свойства субъекта в следующий момент времени). *Ассоциированные объекты-данные* выступают в роли аргументов в операциях, порождающих потоки информации (например, буферы оперативной памяти, в которых помещается для отображения на экране информация при чтении файла). Таким образом, если на первый взгляд в потоке участвует только один (одни) субъект(ы), то, как правило, при более пристальном взгляде можно увидеть, что в данной операции участвуют еще и ассоциированные с субъектом доступа объекты.

Заметим также в развитие *положения 5*, что, исходя из *определения 1.3.7*, поток всегда инициируется (порождается) субъектом доступа. На этом основании вводится следующее центральное в политике и моделях разграничения доступа понятие.

Определение 1.3.8. *Доступом субъекта s_m к объекту o_j называется порождение субъектом s_m потока информации между объектом o_j и некоторым(и) объектом o_i (в т. ч., но не обязательно, объект o_i ассоциирован с субъектом s_m).*

Формальное *определение 1.3.8* понятия доступа дает возможность средствами субъектно-объектной модели перейти непосредственно к описанию процессов безопасности информации в защищенных КС. С этой целью вводится множество потоков P для всей совокупности фиксированных декомпозиций КС на субъекты и объекты во все моменты времени (множество P является объединением потоков по всем моментам времени функционирования КС).

С точки зрения процессов безопасности, трактуемой как состояние защищенности информации в КС, множество потоков P разбивается на два непересекающихся подмножества P_N и P_L :

$$P = P_L \cup P_N,$$

$$P_L \cap P_N = \emptyset,$$

где P_L – множество потоков, вызываемых легальными (безопасными) доступами;

P_N – множество опасных, нарушающих состояние защищенности информации (конфиденциальность, целостность и доступность информации) потоков в КС.

На основе множества потоков дается следующее понятие, составляющее основу формализации политики разграничения доступа в моделях безопасности.

Определение 1.3.9. *Правила разграничения доступа субъектов к объектам есть формально описанные потоки, принадлежащие множеству P_L .*

Определение 1.3.9 завершает основные положения субъектно-объектной модели КС, на методологическом фундаменте которой строится большинство моделей разграничения доступа, выражающих, собственно, подходы, принципы и механизмы правил разграничения доступа (политику разграничения доступа), а также формальные их спецификации (сами модели разграничения доступа). Ввиду того, что *определение 1.3.9* не конкретизирует и не детализирует конкретных механизмов фильтрации потоков на опасные и безопасные, то можно говорить, что субъектно-объектная модель КС инвариантна относительно любой принимаемой в КС политики безопасности.

Добавим, кроме того, что во многих источниках и, в особенности, в нормативных документах по защите информации в КС, основываясь на понятии правил разграничения доступа, вводят производные термины в виде санкционированных и несанкционированных доступов.

1.3.3. Монитор безопасности и основные типы политик безопасности

Анализ практического опыта по защите компьютерной информации, а также основных положений субъектно-объектной модели КС позволяет сформулировать несколько аксиоматических условий, касающихся структуры и функционирования защищенных КС.

Аксиома 1.3.1. *В защищенной КС в любой момент времени любой субъект и объект должны быть персонифицированы (идентифицированы¹) и аутентифицированы².*

Данная аксиома определяется самой природой и содержанием процессов коллективного доступа пользователей к ресурсам КС. Если какие-либо субъекты (пользователи) имеют возможность выдать себя в КС за других субъектов (пользователей) или если имеют возможность подменять (выдавать) одни объекты доступа за другие, то ни о какой безопасности, защищенности речи быть не может. Таким образом, аксиома 1.3.1 выражает необходимое условие безопасности (защищенности) информации в КС, а процедуры, механизмы и системы, осуществляющие идентификацию и аутентификацию пользователей, их субъектов и объектов доступа, являются исходным и важнейшим программно-техническим рубежом защиты информации в КС.

Аксиома 1.3.2. *В защищенной КС должна присутствовать активная компонента (субъект, процесс и т. д.) с соответствующим объектом(ами)-источником, которая осуществляет управление доступом и контроль доступа субъектов к объектам.*

В литературе для данной активной компоненты утвердился термин "монитор безопасности". Понятие монитора безопасности позволяет выразить схемотехнический аспект защиты информации в КС в виде схемы, представленной на рис.1.5. В структуре большинства типов программных средств, на основе которых строятся информационные системы (ОС, СУБД), можно выделить ядро (ядро ОС, машина данных СУБД), в свою очередь, разделяемое на компоненту представления информации (файловая система ОС, модель данных СУБД) и на компоненту доступа к данным (система ввода-вывода ОС, процессор запросов СУБД), а также надстройку (утилиты, сервис, интерфейсные компоненты). Инициализированные субъекты при осуществлении процессов доступа обращаются за сервисом, функциями к ядру системы – см. рис. 1.5.а.

¹ Идентификация – различение и представление экземпляров сущностей по именам-идентификаторам.

² Аутентификация – проверка и подтверждение подлинности идентифицированных экземпляров сущностей.

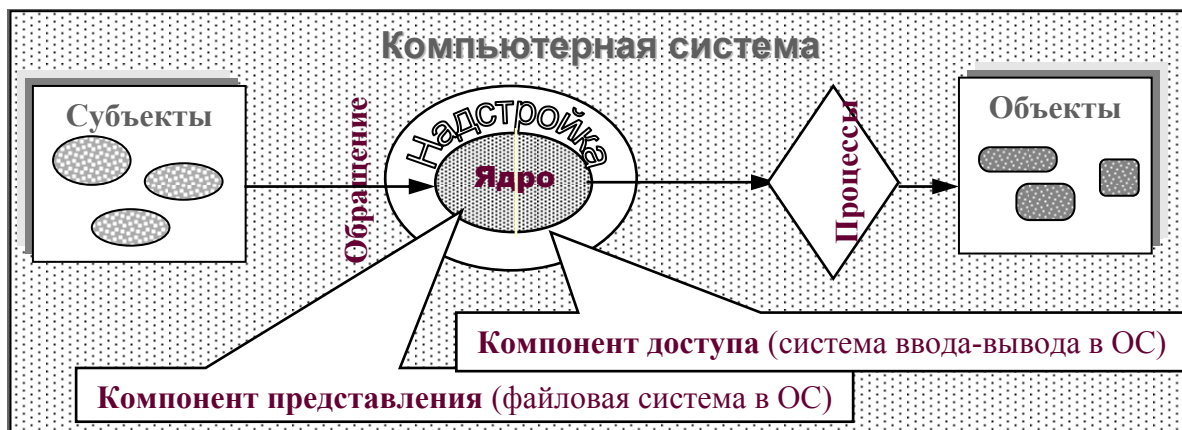


Рис.1.5.а. Системотехнический аспект незащищенной КС

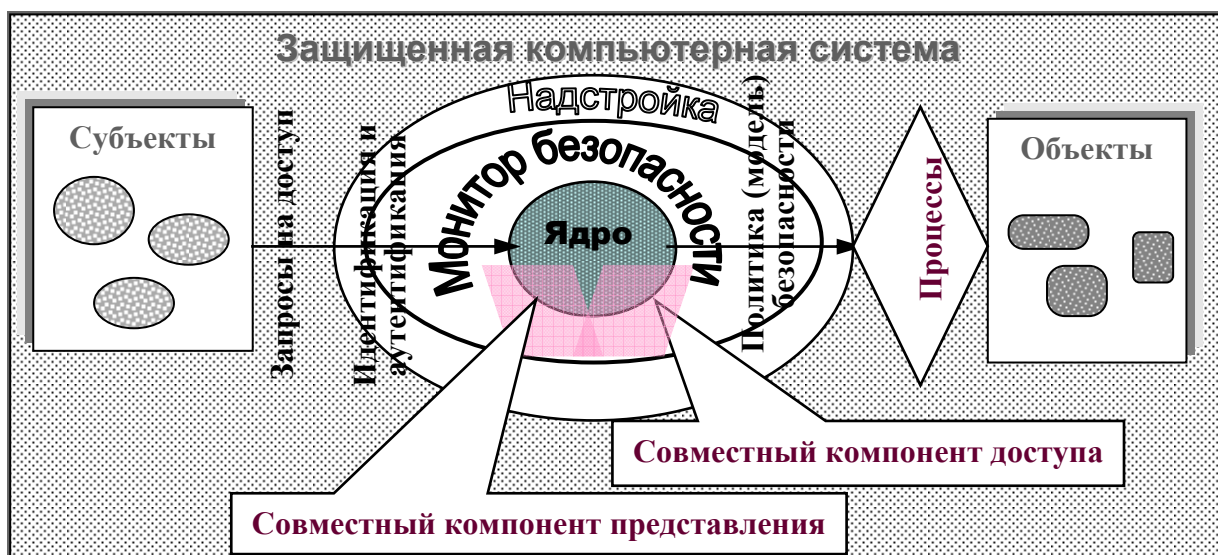


Рис.1.5.б. Системотехнический аспект защищенной КС

В защищенной системе появляется дополнительная компонента, обеспечивающая процессы защиты информации, прежде всего, процедуры идентификации/аутентификации, а также управление доступом на основе той или иной политики безопасности (разграничения доступа) – см. рис. 1.5.б. Ввиду того, что как само ядро КС (компонент представления и компонент доступа), так и процессы разграничения доступа неразрывно связаны с представлением информации и манипулированием с ней, то монитор безопасности должен быть интегрирован непосредственно в ядро системы. Иногда говорят, что монитор безопасности должен быть реализован на нулевом уровне (на уровне ядра) системы. В этом отношении заметим, что более правильный подход заключается в такой разработке компо-

нентов ядра КС, которые бы изначально строились на основе определенной модели безопасности (модели разграничения) доступа.

В практическом плане, в том числе и с учетом отечественных и международных нормативных требований по сертификации защищенных систем, к реализации монитора безопасности предъявляются следующие обязательные требования:

1. *Полнота*. Монитор безопасности должен вызываться (активизироваться) при каждом обращении за доступом любого субъекта к любому объекту, и не должно быть никаких способов его обхода.
2. *Изолированность*. Монитор безопасности должен быть защищен от отслеживания и перехвата своей работы.
3. *Верифицируемость*¹. Монитор безопасности должен быть проверяемым (само- или внешне тестируемым) на предмет выполнения своих функций.
4. *Непрерывность*. Монитор безопасности должен функционировать при любых штатных и нештатных, в том числе и аварийных ситуациях².

Таким образом, именно монитор безопасности в защищенной системе является субъектом осуществления принятой политики безопасности, реализуя через алгоритмы своей работы соответствующие модели безопасности. В этом отношении большое значение имеет следующее аксиоматическое положение.

Аксиома 1.3.3. *Для реализации принятой политики безопасности, управления и контроля доступа субъектов к объектам необходима (должна существовать) информация и объект(ы), ее содержащий(ие) (помимо информации для идентификации и аутентификации пользователей).*

Из аксиомы 1.3.3 следует, что монитор безопасности, в свою очередь, как и любая активная сущность в КС, является субъектом с соответствующим объектом-источником и ассоциированными объектами. Отсюда вытекают следующие важные следствия.

Следствие 1.3.1 (из аксиомы 1.3.3). *В защищенной КС существуют особая категория субъектов (активных сущностей), которые не инициализируют и которыми не управляют пользователи системы – т. н. системные процессы (субъекты), присутствующие (функционирующие) в системе изначально.*

¹ Требования изолированности и верифицируемости являются следствием более общих требований, связанных с гарантиями выполнения политики безопасности.

² Заметим, что большинство изъянов в системах защиты КС, способы успешных атак на защищенные КС основываются на нарушениях или неполной реализации данных требований.

К числу подобных системных субъектов относится исходный системный процесс, который инициализирует первичные субъекты пользователей, а также монитор безопасности, который управляет доступами субъектов пользователей к объектам системы¹. Соответственно, для обеспечения защищенности в КС свойства системных субъектов должны быть неизменными, от чего напрямую зависят гарантии безопасности.

Следствие 1.3.2 (из аксиомы 1.3.3). Ассоциированный с монитором безопасности объект, содержащий информацию по системе разграничения доступа, является наиболее критическим с точки зрения безопасности информационным ресурсом в защищенной КС.

Действительно возможность несанкционированно изменять, удалять данный объект может полностью разрушить или дискредитировать всю систему безопасности КС. Поэтому способы и особенности реализации данного объекта имеют определяющее значение для защищенности информации в КС.

Информация в ассоциированном с монитором безопасности объекте должна касаться конкретных зарегистрированных в системе пользователей и конкретных объектов системы. Следовательно, для планирования и управления системой разграничения доступа конкретного коллектива пользователей КС должна быть предусмотрена процедура доступа к данному объекту со стороны внешнего фактора, т. е. через субъект(ы) пользователя. Отсюда вытекает еще одно следствие.

Следствие 1.3.3 (из аксиомы 1.3.3). В защищенной системе может существовать доверенный пользователь (администратор системы), субъекты которого имеют доступ к ассоциированному с монитором безопасности объекту-данным для управления политикой разграничения доступа.

Заметим также, что субъекты, иницируемые администратором системы, не являются элементами или процессами монитора безопасности, а лишь обеспечивают монитор безопасности конкретной информацией для управления и контроля доступом субъектов к объектам системы.

Принципы, способы представления и реализация ассоциированных с монитором безопасности объектов определяются типом политики безопасности и особенностями конкретной КС.

Несмотря на то, что к настоящему времени разработано и апробировано в практической реализации большое количество различных моделей безопасности КС, все они выражают несколько исходных политик безо-

¹ Во многих системах основные функции монитора безопасности, за исключением функции по реализации процедур идентификации/аутентификации, как раз и сосредоточиваются в исходном системном процессе.

пасности. В упрощенной трактовке политику безопасности понимают как общий принцип (методологию, правило, схему) безопасной работы (доступа) коллектива пользователей с общими информационными ресурсами. При этом согласно *определению 1.3.9* важнейшее значение имеет *критерий безопасности* доступов субъектов к объектам, т. е. правило разделения информационных потоков, порождаемых доступами субъектов к объектам, на опасные и неопасные.

Методологической основой для формирования политик безопасности в защищенных КС послужили реальные организационно-технологические схемы обеспечения безопасности информации во вне (до) компьютерных сферах. Многие подходы к защите компьютерной информации были "подсмотрены", в частности, в сфере работы с "бумажными" конфиденциальными документами, проще говоря, в сфере делопроизводства.

Выделяется две основных (базовых) политики безопасности – дискреционная и мандатная. В еще не до конца устоявшейся терминологии сферы защиты компьютерной информации, первую называют политикой избирательного доступа, а вторую – политикой полномочного¹ доступа. Следует отметить, что известные модели ролевого доступа выделяют в группу особой "ролевой политики безопасности". Кроме того в документальных информационно-поисковых системах применяется политика тематического разграничения доступа, также как и другие политики "подсмотренная" во внекомпьютерной (библиотечно-архивной) сфере.

Модели, выражающие ту или иную политику безопасности², подробно рассматриваются в соответствующих главах. Здесь же мы ограничимся общей их характеристикой, отталкиваясь от основных понятий и, в частности, *определений 1.3.8, 1.3.9* субъектно-объектной модели КС.

Политика дискреционного (избирательного) доступа. *Множество безопасных (разрешенных) доступов P_L задается для именованных пользователей (субъектов) и объектов явным образом в виде дискретного набора троек "Пользователь(субъект)-поток(операция)-объект".*

Принцип дискреционной политики разграничения доступа можно охарактеризовать схемой "каждый-с каждым", т. е. иными словами для любой из всевозможных комбинаций "пользователь (субъект)-ресурс (объект)" должно быть явно задано ("прописано") разрешение/запрещение доступа и вид соответствующей разрешен-

¹ Отметим, что на наш взгляд термин "полномочный" относится не только и не столько к собственно мандатной политике, сколько характеризует уже упомянутую суть любой политики безопасности – принцип (способ, механизм и т. п.) наделения пользователей полномочиями (правом) работы с необходимой им информацией, при котором в смысле определенного критерия обеспечивается состояние защищенности информации в КС.

² Напомним еще раз, что в данной работе политику безопасности мы рассматриваем, прежде всего, и, в основном, в контексте политики разграничения доступа.

ной/запрещенной операции (*Read, Write* и т. д.). Таким образом, при дискреционной политике разграничение доступа осуществляется самым детальным образом – до уровня отдельно взятого субъекта, отдельно взятого объекта доступа и отдельно взятой операции.

Политика мандатного (полномочного) доступа. Множество безопасных (разрешенных) доступов P_L задается неявным образом через введение для пользователей-субъектов некоторой дискретной характеристики доверия (уровня допуска), а для объектов некоторой дискретной характеристики конфиденциальности (грифа секретности), и наделение на этой основе пользователей-субъектов некими полномочиями порождать определенные потоки в зависимости от соотношения "уровень допуска-поток(операция)-уровень конфиденциальности".

Таким образом, в отличие от дискреционной политики, при мандатной политике разграничение доступа производится менее детально – до уровня группы пользователей с определенным уровнем допуска и группы объектов с определенным уровнем конфиденциальности. Заметим также, что уменьшение гранулированности доступа создает условия для упрощения и улучшения управления доступом ввиду существенного уменьшения количества субъектов управления и контроля.

Политика тематического доступа. Множество безопасных (разрешенных) доступов P_L задается неявным образом через введение для пользователей-субъектов некоторой тематической характеристики – разрешенных тематических информационных рубрик, а для объектов аналогичной характеристики в виде набора тематических рубрик, информация по которым содержится в объекте, и наделение на этой основе субъектов-пользователей полномочиями порождать определенные потоки в зависимости от соотношения "набор тематических рубрик субъекта–набор тематических рубрик объекта".

Как и при мандатном доступе, тематический принцип определяет доступ субъекта к объекту неявно, через соотношение предъявляемых специальных характеристик субъекта и объекта и, соответственно, по сравнению с дискреционным принципом существенно упрощает управление доступом.

Политика ролевого доступа. Множество безопасных (разрешенных) доступов P_L задается через введение в системе¹ дополнительных абстрактных сущностей – ролей, выступающих некими "типовыми" (ролевыми) субъектами доступа, с которыми ассоциируются конкретные пользователи (в роли которых осуществляют доступ), и наделение ро-

¹ и, соответственно, в субъектно-объектной модели КС.

левых субъектов доступа на основе дискреционного или мандатного принципа правами доступа к объектам системы.

Ролевая политика разграничивает доступ не на уровне пользователей-субъектов, а на уровне ролей, являющихся группами однотипного доступа к объектам системы, и на этой основе развивает ту или иную базовую политику безопасности (дискреционную или мандатную). Поэтому в большинстве источников ролевой принцип разграничения доступом не выделяется в отдельную политику, а рассматривается в качестве неких дополнений к моделям дискреционного или мандатного доступа.

Следует также отметить, что в практике функционирования защищенных компьютерных систем широко используется временные и маршрутные (в распределенных КС) ограничения доступа, что позволяет, в принципе, говорить о временной и маршрутной политике безопасности, которые, дополняют отмеченные базовые политики безопасности.

Каждая политика безопасности требует определенной информации для разграничения доступа в конкретной системе, локализуемой в объекте, ассоциированном с монитором безопасности. Для моделей дискреционного доступа эта информация представляет список разрешенных троек "субъект(пользователь)-операция-объект". Для управления доступом в системах с мандатным доступом необходима информация по уровням допуска субъектов и грифам конфиденциальности объектов. В системах ролевого доступа помимо информации, регламентирующей доступ ролей к объектам (на основе дискреционного или мандатного принципа), необходима информация по ассоциации пользователей-субъектов с ролями. При тематическом доступе необходима информация по тематическим рубрикам пользователей-субъектов и объектов.

Конкретная модель безопасности детализирует и формализует (в виде аналитических соотношений, алгоритмов, и т. д.) общий принцип разграничения доступа на основе одной из рассмотренных политик, а иногда некоторой их совокупности. В конкретной КС разработчики строят и реализуют оригинальные программно-технические решения, воплощающие модели безопасности, в том числе структуру, функции, программно-техническое воплощение монитора безопасности.

1.3.4. Гарантирование выполнения политики безопасности

Положения субъектно-объектной модели КС, в частности, понятия доступа субъектов к объектам и политики безопасности позволяют сформулировать следующий общий критерий безопасности КС.

Определение 1.3.10. Компьютерная система безопасна тогда и только тогда, когда субъекты не имеют никаких возможностей нарушать (обходить) установленную в системе политику безопасности.

Субъектом обеспечения политики безопасности выступает монитор безопасности. Его наличие в структуре КС, соответственно, является *необходимым* условием безопасности. Что касается условий *достаточности*, то, очевидно, они заключены, несмотря на тавтологичность выражения, прежде всего, в безопасности самого монитора безопасности.

Подтверждением данного тезиса является обязательное включение в состав спецификаций по созданию (разработке) и оценке (сертификации) защищенных КС требований корректности, верификации, адекватности и т. д. средств защиты информации (т. е. монитора безопасности) во всех, в том числе, и отечественных стандартах и руководящих документах по компьютерной безопасности.

Необходимость доказательного подхода к гарантиям обеспечения защищенности информации в КС в отечественной литературе была впервые поставлена в работах А.А.Грушо. В этих работах приведен пример гарантированно (т. е. математически доказанной) защищенной системы обработки информации на основе определенных предположений и условий.

В развитие методологии данного подхода А.Ю.Щербаковым предложена модель гарантированности выполнения политики безопасности в более широких рамках и условиях субъектно-объектной модели КС.

Приведем основные положения данной модели.

Автором модели, прежде всего, было отмечено влияние на безопасность системы не только доступов (потоков) к объектам, осуществляемых субъектами пользователей, но и того, какого типа субъектами пользователи осуществляют доступ к объектам. К примеру, доступ пользователя к файлу базы данных через СУБД порождает информационный поток одного типа с определенными регламентациями-ограничениями, а доступ к тому же файлу базы данных с помощью дискового редактора – информационный поток другого типа, через который пользователь может получить не предназначенную, вообще говоря, ему информацию. При этом с формальной точки зрения политика безопасности, определяющая правомерность самого факта доступа данного пользователя к файлу базы данных соблюдается и в том и другом случае.

Отсюда следует, что правила разграничения доступа, составляющие основу политики безопасности, должны включать и правила порождения (инициализации) пользователями субъектов доступа¹.

В технологическом плане выполнение данного требования приводит к необходимости расщепления монитора безопасности на два отдельных субъекта:

- монитор безопасности объектов;
- монитор безопасности субъектов.

Вводятся соответствующие определения.

Определение 1.3.11. *Монитором безопасности объектов (МБО) называется субъект, активизирующийся при возникновении потока между любыми объектами, порождаемого любым субъектом, и разрешающий только те потоки, которые принадлежат множеству P_L .*

Определение 1.3.12. *Монитором безопасности субъектов (МБС) называется субъект, активизирующийся при любом порождении субъектов, и разрешающий порождение субъектов из фиксированного подмножества пар активизирующих субъектов и объектов-источников.*

Определение 1.3.12, по сути, вводит в состав политики безопасности КС в качестве дополнительной составной части специальную политику порождения пользователями субъектов доступа. Соответственно, как и у любого субъекта, у МБС должен быть объект-источник, функционально-ассоциированный объект (исполняемый код в оперативной памяти) и ассоциированный объект-данные, содержащий необходимую информацию по политике порождения пользователями субъектов доступа в системе – см. рис. 1.6.

Вторым аспектом, подмеченным в плане гарантий выполнения политики безопасности, является *неизменность свойств субъектов доступа* в процессе функционирования КС. Многие известные атаки на защищенные КС как раз и осуществляются по сценарию подмены кода программ, запускаемых на выполнение регламентированных функций (т. е. фактически подмены свойств субъектов). Данное требование имеет отношение к любым субъектам доступа любых пользователей, но особо для системных субъектов, и, в частности, для монитора безопасности.

¹ Данное требование на первый взгляд может показаться избыточным, т. к. порождение субъектов (операция *Create*), в свою очередь, основывается также на возможности (праве) доступа пользователя к соответствующему объекту-источнику (файлу соответствующей программы). Однако из примера с файлом базы данных легко представить ситуацию, когда пользователь может иметь законный доступ как к самой СУБД (чтобы инициализировать субъект для регламентированного доступа к файлу БД), так к дисковому редактору, чтобы инициализировать субъект для регламентированного доступа к объектам другого типа.

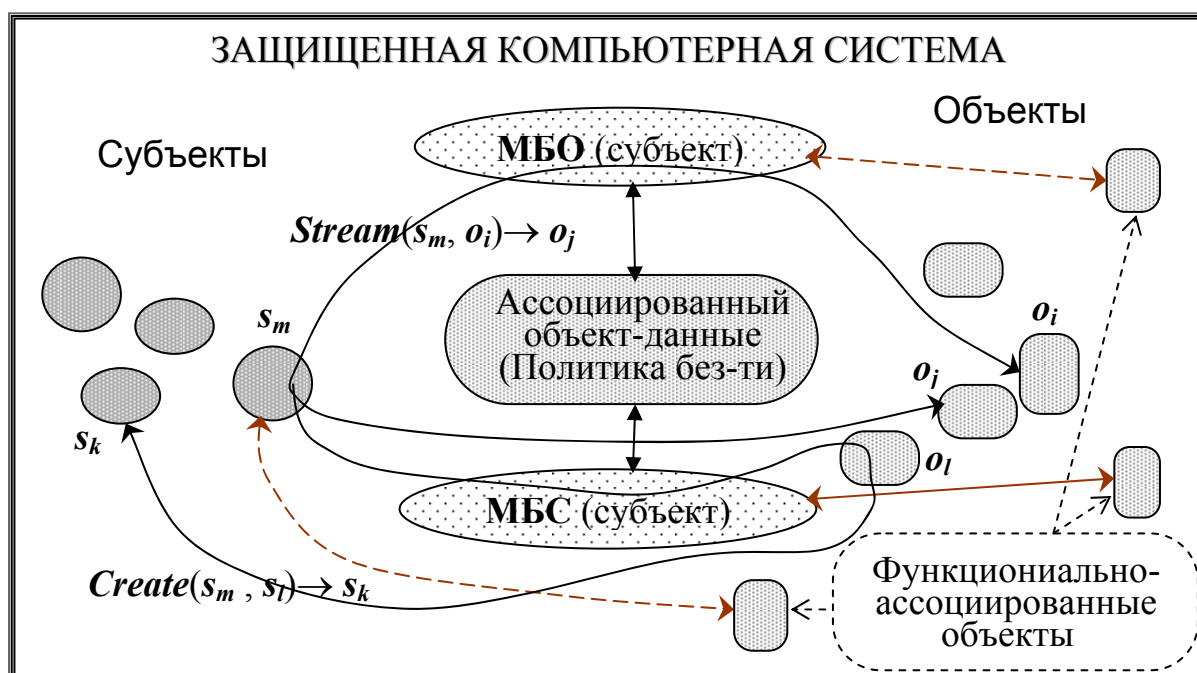


Рис.1.6. Порождение потоков (Stream) и субъектов (Create) с учетом МБО и МБС

Для рассмотрения условий неизменности субъектов, вводятся следующие определения.

Определение 1.3.13. Объекты o_i и o_j тождественны в момент времени t ($o_i[t] \equiv o_j[t]$), если они совпадают как слова, записанные в одном языке.

Тождественность объектов по определению 1.3.13 основывается не на физической тождественности, а на тождественности до уровня последовательности символов из алфавита языка представления. Для иллюстрации понятия тождественности приведем пример эквивалентности (тождественности) двух файлов на основе побайтного сравнения, один из которых размещен на диске, другой в оперативной памяти, и находящихся, соответственно, в разной реализации по физическим процессам функционирования носителей (т. е. являющихся физически не тождественными).

Введенное понятие тождественности объектов позволяет перейти к рассмотрению понятия тождественности и неизменности субъектов доступа.

Определение 1.3.14. Субъекты s_i и s_j тождественны в момент времени t , если попарно тождественны все ассоциированные с ними объекты.

Определения 1.3.13 и 1.3.14 неявно требуют наличия в КС специального механизма сортировки однотипных объектов и их попарного сравнения, и, кроме того, обуславливают следующее важное следствие.

Следствие 1.3.4 (из определений 1.3.13 и 1.3.14). *Порожденные субъекты тождественны, если тождественны все порождающие субъекты и объекты-источники.*

Обоснованность данного следствия вытекает из тождества функционально-ассоциированных объектов в порождающих субъектах, которые отвечают за порождение нового субъекта, а также из тождественности аргументов операции порождения, т. е. ассоциированных объектов-данных и объектов-источников, которые определяют свойства порождаемых субъектов.

Очевидно, что субъекты, осуществляющие доступ к объектам системы, в том числе и к объектам, ассоциированным с другими субъектами, могут тем самым влиять на них и изменять их свойства. Поэтому вводится следующее определение.

Определение 1.3.15. *Субъекты s_i и s_j называются невлияющими друг на друга (или корректными относительно друг друга), если в любой момент времени отсутствует поток (изменяющий состояние объекта) между любыми объектами o_{ik} и o_{jl} , ассоциированными, соответственно с субъектами s_i и s_j . При этом объекты o_{ik} не ассоциированы с субъектом s_j , а объекты o_{jl} не ассоциированы с субъектом s_i .*

Отметим, что термин "изменение состояния объекта" в определении 1.3.15 трактуется как нетождественность (в смысле определения 1.3.13) объекта с самим собой в различные моменты времени.

Анализ понятия ассоциированных с субъектом объектов позволяет ввести еще более жесткое определение по влиянию одних субъектов на других.

Определение 1.3.16. *Субъекты s_i и s_j называются абсолютно невлияющими друг на друга (или абсолютно корректными относительно друг друга), если в условиях определения 1.3.15 множества ассоциированных объектов указанных субъектов не имеют пересечения.*

На основании данного определения можно сформулировать достаточное условие гарантированного выполнения политики безопасности.

Утверждение 1.3.1 (достаточное условие гарантий безопасности 1). *Монитор безопасности объектов разрешает порождение потоков только из множества P_L , если все существующие в системе субъекты абсолютно корректны относительно него и друг друга.*

Доказательство. Из условия абсолютной корректности любых субъектов с **МБО** вытекает отсутствие потоков, которые могут изменить функционально-ассоциированные и ассоциированные объекты-данные с **МБО** и тем самым изменить его свойства для осуществления обхода (нарушения)

политики безопасности. С другой стороны, отсутствуют также потоки между ассоциированными объектами и всех любых других субъектов, и, следовательно, отсутствует возможность изменения одними субъектами свойств других субъектов для возможного нарушения (обхода) политики безопасности. Тем самым утверждение доказано. ■

Утверждение 1.3.1 для обеспечения гарантий безопасности накладывает чрезвычайно жесткие условия, практически не выполнимые на практике, или существенно снижающие функциональные возможности КС (отсутствие общих объектов-источников для запуска программ разными пользователями, отсутствие общих участков памяти, буферов для обмена данными и т. п.).

Для исследования подходов, в большей степени возможных при практической реализации, вводятся понятия замкнутости и изолированности подмножества субъектов системы.

Определение 1.3.17. *КС называется замкнутой по порождению субъектов, если в ней действует МБС, разрешающий порождение только фиксированного конечного подмножества субъектов для любых объектов-источников, рассматриваемых для фиксированной декомпозиции КС на субъекты и объекты.*

Эквивалентным понятием для замкнутой по порождению субъектов системы является понятие "изолированной программной среды" (ИПС). Механизм замкнутой программной среды сокращает множество возможных субъектов до некоторого множества фиксированной мощности, но при этом не гарантирует отсутствие некорректных субъектов внутри замкнутой среды.

Определение 1.3.18. *Множество субъектов КС называется изолированным (абсолютно изолированным), если в ней действует МБС и субъекты из порождаемого множества корректны (абсолютно корректны) относительно друг друга и МБС.*

Из данного определения вытекают следующие следствия.

Следствие 1.3.5. (из определения 1.3.18). *Любое подмножество субъектов изолированной (абсолютно изолированной) КС, включающее МБС, также составляет изолированную (абсолютно изолированную) среду.*

Следствие 1.3.6. (из определения 1.3.18). *Дополнение изолированной (абсолютно изолированной) среды субъектом, корректным (абсолютно корректным) относительно любого из числа входящих в изолированную (абсолютно изолированную) среду, оставляет ее изолированной (абсолютно изолированной).*

На этой основе можно сформулировать другое условие достаточности гарантий выполнения политики безопасности.

Утверждение 1.3.2. (достаточное условие гарантий безопасности 2). Если в абсолютно изолированной КС существует **МБО** и порождаемые субъекты абсолютно корректны относительно **МБО**, а также существует **МБС**, который абсолютно корректен относительно **МБО**, то в КС реализуется только доступ, описанный политикой разграничения доступа.

Доказательство. По определению 1.3.18 и следствиям из него в системе могут существовать только абсолютно корректные относительно **МБС** и друг друга субъекты из некоторого их конечного множества. Следовательно, отсутствует возможность изменения свойств **МБС**. Абсолютная корректность **МБС** и других субъектов по отношению к **МБО** обеспечивает отсутствие возможностей изменения свойств **МБО**, что в итоге автоматически обеспечивает разрешение только тех потоков, которые входят в множество P_L . Утверждение доказано. ■

В отличие от первого условия достаточности гарантий выполнения политики безопасности, второе условие менее жестко, так как накладывает условия абсолютной корректности не на все множество возможных субъектов, а лишь на фиксированное их подмножество, образующее замкнутую (изолированную) программную среду (ИПС).

И все же, требование абсолютной корректности, хотя и для фиксированного подмножества субъектов, является также чрезвычайно жестким и трудно выполнимым на практике без существенного снижения функциональных возможностей КС.

Дальнейший анализ подходов к гарантиям безопасности, точнее, к возможностям реализации ИПС, показал необходимость включения требований по неизменности свойств субъектов, основанных на неких дополнительных процедурах, связанных с порождением субъектов.

Определение 1.3.19. Операция порождения субъектов $Create(s_k, o_m) \rightarrow s_l$ называется порождением с контролем неизменности объекта, если для любого момента времени $t > t_0$, в который активизирована операция порождения объекта **Create**, порождение объекта s_l возможно только при тождественности объекта-источника относительно момента t_0 , т. е. при $o_m[t] \equiv o_m[t_0]$.

Из определения 1.3.19 вытекает следующее важное следствие, имеющее непосредственное отношение к неизменности свойств субъектов доступа, как важнейшего условия обеспечения политики безопасности в системе.

Следствие 1.3.7. (из определения 1.3.19). В условиях определения 1.3.19 порожденные субъекты $s_l[t_1]$ и $s_l[t_2]$ тождественны, если $t_1 > t_0$ и $t_2 > t_0$. При $t_1 = t_2$ порождается один и тот же объект.

Введение понятия порождения субъектов с контролем неизменности объектов позволяет сформулировать и доказать такое достаточное условие

для обеспечения ИПС, которое может быть практически реализовано в реальных КС.

Утверждение 1.3.3. (базовая теорема ИПС). *Если в изолированной КС, в которой действует порождение субъектов с контролем неизменности объекта, в момент времени t_0 через любой субъект к любому объекту существуют только потоки, не противоречащие условию корректности (абсолютной корректности), то в любой момент времени $t_k > t_0$ КС также остается изолированной (абсолютно изолированной).*

Доказательство. Условие корректности для потоков в начальный момент t_0 времени функционирования системы обеспечивает исходную "правильность" всех объектов-источников для последующего порождения субъектов системы. Исходя из этого, и основываясь на *определении 1.3.19* и *следствии 1.3.7* из него, все последующие порождения субъектов будут происходить с неизменностью их свойств, в том числе с сохранением состояния взаимной корректности. Следовательно, в любой момент времени будут соблюдаться требования изолированности КС. Утверждение доказано. ■

Утверждение 1.3.3 имеет важнейшее значение с точки зрения достаточных условий для обеспечения гарантий выполнения политики безопасности в защищенных КС. Вместе с тем, при практической реализации условий *утверждения 1.3.3* также возникает несколько серьезных проблем. Одна из них, если так можно выразиться, созвучна с известной проблемой "ахиллесовой пяты". А именно, с исходным состоянием КС, в котором должны быть только потоки, гарантирующие корректность исходных субъектов.

Проблемы и пути обеспечения исходной корректности (изолированности) КС основываются на определенных принципах и процедурах загрузки (ступенчатой инициализации) КС в начальный момент времени.

Вторая проблема связана с понижением производительности (быстродействия) КС в связи с осуществлением процедур контроля неизменности объектов-источников при порождении субъектов.

От себя также заметим, не умаляя достоинств и теоретического значения рассмотренной модели гарантий выполнения политики безопасности, что ее авторами упущен еще один важный в практическом плане аспект. В частности, как уже отмечалось в *следствии 1.3.1* из *аксиомы 1.3.3*, для управления конкретными параметрами системы разграничения доступа могут (должны) существовать субъекты доверенных пользователей (администраторов), имеющих доступ к ассоциированному с МБО объекту-данным. Однако рассмотренные выше условия гарантий безопасности основываются на понятии корректности (невлияния) относительно МБО и МБС всех субъектов доступа. Согласно *определению 1.3.15* корректность

субъектов обеспечивается их невозможностью изменять состояние всех ассоциированных с другими субъектами объектов, в том числе и объектов-данных, ассоциированных с **МБО**, содержащих конкретную информацию по политике разграничения доступа. Следовательно, полная и строгая реализация условий ИПС из-за невозможности впоследствии изменять ассоциированные с **МБО** объекты потребует изначального встраивания в систему конкретных параметров разграничения доступа (по конкретным и, соответственно, изначально зарегистрированным в системе пользователям, и конкретным, изначально созданным в системе объектам доступа), что лишает такую систему какой-либо универсальности и гибкости.

Тем не менее, теоретическая значимость рассмотренной модели неоспорима, так как создает инвариантную основу по отношению к любым политикам и моделям разграничения доступа для гарантий выполнения политики безопасности в защищенных КС.

II. МОДЕЛИ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

2.1. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ ДИСКРЕЦИОННОЙ ПОЛИТИКИ

Модели безопасности, строящиеся на субъектно-объектной модели КС, еще называют *моделями конечных состояний*. В данных моделях инициализация информационных потоков трактуется как запросы субъектов на доступ к объектам, которые в зависимости от политики безопасности разрешаются или запрещаются. Осуществление субъектом разрешенного доступа к объекту переводит систему в следующий момент времени в другое состояние, рассматриваемое как совокупность состояний субъектов и объектов системы.

Проблема безопасности в КС рассматривается с точки зрения анализа и исследования условий, правил, порядка и т. п. разрешений запросов на доступ, при которых система, изначально находясь в безопасном состоянии, за конечное число переходов перейдет также в безопасное состояние.

2.1.1. Общая характеристика моделей дискреционного доступа. Пятимерное пространство Хартсона

Политика дискреционного доступа охватывает самую многочисленную совокупность моделей разграничения доступа, реализованных в большинстве защищенных КС, и исторически является первой, проработанной в теоретическом и практическом плане.

Первые работы по моделям дискреционного доступа к информации в КС появились еще в 60-х годах и подробно представлены в литературе. Наиболее известные из них – модель АДЕПТ-50 (конец 60-х годов), пятимерное пространство Хартсона (начало 70-х годов), модель Хариссона-Руццо-Ульмана (середина 70-х годов), модель Take-Grant (1976 г.). Авторами и исследователями этих моделей был внесен значительный вклад в теорию безопасности компьютерных систем, а их работы заложили основу для последующего создания и развития защищенных КС.

Модели дискреционного доступа непосредственно основываются и развивают субъектно-объектную модель КС как совокупность некоторых множеств взаимодействующих элементов (субъектов, объектов и т. д.). Множество (область) безопасных доступов в моделях дискреционного доступа определяется дискретным набором троек "Пользователь (субъект)-поток (операция)-объект".

Конкретные модели специфицируют способ представления области безопасного доступа и механизм проверки соответствия запроса субъекта

на доступ области безопасного доступа. Если запрос не выходит за пределы данной области, то он разрешается и выполняется. При этом постулируется, что осуществление такого доступа переводит систему в безопасное состояние.

Специфика и значение моделей заключается в том, что исходя из способа представления (описания) области безопасного доступа и механизма разрешений на доступ анализируется и доказывается, что за конечное число переходов система останется в безопасном состоянии.

Модель дискреционного доступа, предложенная Хартсоном, вероятно наиболее наглядно в формальном плане иллюстрирует дискреционный принцип разграничения доступа, выраженный языком реляционной алгебры. Приведем ее основные положения в кратком изложении.

1. Система представляется совокупностью пяти наборов (множеств):

- множества пользователей U ;
- множества ресурсов R ;
- множества состояний S ;
- множества установленных полномочий A ;
- множества операций E .

2. Область безопасности представляется декартовым произведением:

$$A \times U \times E \times R \times S . \quad (2.1.1)$$

3. Пользователи подают запросы на доступ к ресурсам, осуществление которых переводит систему в новое состояние. Запросы на доступ представляются четырехмерными кортежами

$$q = (u, e, R', s) , \quad (2.1.2)$$

где $u \in U$, $e \in E$, $s \in S$, $R' \subseteq R$ (R' - требуемый набор ресурсов).

Таким образом, запрос на доступ представляет собой подпространство четырехмерной проекции пространства безопасности. Запрос удовлетворяется, если он полностью заключен в области безопасности (2.1).

4. Процесс организации доступа алгоритмически описывается следующим образом.

4.1. Определить из U те группы пользователей, к которым принадлежит u . Затем выбрать из A те спецификации, которым соответствуют выделенные группы пользователей. Этот набор полномочий $F(u)$ определяет привилегию пользователя u .

4.2. Определить из множества A набор полномочий $P = F(e)$, которые устанавливают e как основную операцию. Набор полномочий $P = F(e)$ определяет *привилегию операции e* .

4.3. Определить из множества A набор полномочий $P = F(R')$, разрешающих доступ к набору ресурсов R' . Набор полномочий $P = F(R')$ определяет *привилегию ресурсов R'* .

Полномочия, которые являются общими для всех трех привилегий, образуют так называемый *домен полномочий запроса $D(q)$*

$$D(q) = F(u) \cap F(e) \cap F(R') .$$

4.4. Убедиться, что запрашиваемый набор ресурсов R' полностью содержится в домене запроса $D(q)$, т. е. любой r из набора R' хотя бы один раз присутствует среди элементов $D(q)$.

4.5. Осуществить разбиение $D(q)$ на эквивалентные классы так, чтобы в один класс попадали полномочия (элементы $D(q)$), когда они специфицируют один и тот же ресурс r из набора R' .

В каждом классе произвести операцию логического **ИЛИ** элементов $D(q)$ с учетом типа операции e .

В результате формируется новый набор полномочий на каждую единицу ресурса, указанного в $D(q)$ - $F(u, q)$. Набор $F(u, q)$ называется *фактической привилегией* пользователя u по отношению к запросу q .

4.6. Вычислить условие фактического доступа (**ЕАС**), соответствующее запросу q , через операции логического **ИЛИ** по элементам полномочий $F(u, q)$ и запрашиваемым ресурсам r из набора R' , и получить тем самым набор R'' – набор фактически доступных по запросу ресурсов.

4.7. Оценить **ЕАС** и принять решение о доступе:

- разрешить доступ, если R'' и R' полностью перекрываются;
- отказать в доступе в противном случае.

Заметим, что при всей своей наглядности модель Хартсона обладает одним, но существенным недостатком – безопасность системы на основе данной модели строго не доказана. Пользователи, осуществляя законный доступ к ресурсам, могут изменять состояния системы, в том числе, изменять множество ресурсов R . Тем самым может изменяться и сама область безопасности. Сохранится ли в таком случае безопасность системы? Модель ответа на данный вопрос не дает.

2.1.2. Модели на основе матрицы доступа

В теоретическом и практическом плане наибольшее развитие и применение получили дискреционные модели, основанные на *матрице доступа*. В данных моделях область безопасного доступа строится как прямоугольная матрица (таблица), строки которой соответствуют субъектам доступа, столбцы объектам доступа, а в ячейках записываются разрешенные операции соответствующего субъекта над соответствующим объектом – см. рис. 2.1.

		Объекты доступа					
		o_1	o_2	...	o_j	...	o_N
Субъекты доступа	s_1		w				
	s_2	r					
	...						
	s_i				r,w		
	...						
	s_M						e

Обозначения:

- w – "изменение объекта";
- r – "чтение объекта";
- e – "запуск объекта на выполнение".

Рис. 2.1. Матрица доступа

"Прописанные" в ячейках матрицы права доступа в виде разрешенных операций над объектами определяют виды безопасных доступов соответствующего субъекта к соответствующему объекту. Для выражения типов разрешенных операций используются специальные обозначения, составляющие основу (алфавит) некоторого языка описания политики разграничения доступа – см. рис. 2.1. Таким образом, в рамках дискреционной политики каждая ячейка агрегирует некоторое подмножество троек "субъект-операция(поток)-объект".

В соответствии с *аксиомой 1.3.3* матрица доступа представляет ассоциированный с монитором безопасности объект, содержащий информацию по политике разграничения доступа в конкретной системе. Соответственно, принцип (структура) организации, размещение, а также процессы создания, изменения матрицы доступа определяются конкретными моделями и конкретными программно-техническими решениями КС, в которых они реализуются.

По *принципу организации матрицы доступа* в реальных системах используются два подхода – централизованный и распределенный.

При *централизованном* подходе матрица доступа создается как отдельный самостоятельный объект с особым порядком размещения и доступа к нему. Количество объектов доступа и порождаемых пользователями субъектов доступа в реальных КС может достигать очень больших величин, и, кроме того, подвержено динамическому изменению. Поэтому при централизованном подходе в большинстве систем строки матрицы доступа характеризуют не субъектов, а непосредственно самих пользователей и их группы, зарегистрированные для работы в системе. Для уменьшения количества столбцов матрицы, объекты доступа КС могут агрегироваться в две группы – группу объектов, доступ к которым не ограничен (т. е. разрешен любым пользователям по любым операциям), и группу объектов собственно дискреционного (избирательно разграничительного) доступа. Соответственно, в матрице доступа представляются права пользователей только к объектам второй группы, что позволяет существенно уменьшить ее размерность. Наличие или создание в матрице доступа столбца (строки) для какого-либо объекта фактически означает его регистрацию в системе в качестве объекта дискреционного доступа с соответствующими правами соответствующих пользователей. Наиболее характерным и известным примером такого подхода являются т. н. "биты доступа" в UNIX-системах.

При *распределенном* подходе матрица доступа как отдельный объект не создается, а представляется или так называемыми "*списками доступа*", распределенными по объектам системы, или так называемыми "*списками возможностей*", распределенными по субъектам доступа. В первом случае каждый объект системы, помимо идентифицирующих характеристик, наделяется еще своеобразной биркой, непосредственно связанной с самим объектом, и представляющей, по сути, соответствующий столбец матрицы доступа¹. Во втором случае своеобразную бирку с перечнем разрешенных для доступа объектов (по сути, строку матрицы доступа) получает каждый субъект при своей инициализации². Добавим также, что через "списки возможностей" (маркеры доступа) субъектов реализуется широко используемый на практике *механизм привилегий*, наделяющий субъектов правами выполнять определенные операции над всеми объектами или их определенными группами (типами объектов), что формирует дополнительные аспекты дискреционной политики разграничения доступа.

И централизованный, и распределенный принцип организации матрицы доступа имеет свои преимущества и недостатки, присущие в целом

¹ В "сжатом виде" – без хранения пустых ячеек.

² Отметим, что в данном случае фактически наблюдается разновидность централизованного способа, т. к. для создания "списков возможностей", называемых еще маркерами доступа, необходима информация о разрешенных доступах пользователей, сосредоточенная в объекте, ассоциированном с монитором безопасности (т. н. массив учетных записей пользователей), который в этом случае представляет ту же матрицу доступа.

централизованному и децентрализованному принципам организации и управления.

По *механизму создания и изменения матрицы доступа*, т. е. фактически по *принципу управления доступом*, выделяются также два подхода:

- принудительное управление доступом;
- добровольное управление доступом.

Принцип *принудительного управления доступом* основывается на парадигме доверенных субъектов и непосредственно вытекает из *следствия 1.3.3 аксиомы 1.3.3* по созданию и функционированию защищенных систем. Согласно принудительному способу право создания и изменения матрицы доступа имеют только субъекты администратора системы, который при регистрации для работы в системе нового пользователя создает с соответствующим заполнением новую строку матрицы доступа, а при возникновении нового объекта, подлежащего избирательному доступу, образует новый столбец матрицы доступа.

Нетрудно видеть, что такой способ приемлем только при фиксированном или ограниченном количестве объектов доступа или требует агрегирования объектов доступа в определенные группы и перехода к управлению правами доступа по группам (типам) объектов доступа. Подобный подход наиболее широко представлен в базах данных, где управление доступом в большинстве случаев осуществляется на уровне логических информационных объектов (в реляционных СУБД – таблицы), представляющих агрегирование однотипных элементарных объектов доступа – записей (в реляционных СУБД – кортежи, т. е. табличные строки).

Принцип *добровольного управления доступом* основывается на парадигме "владения" объектами.

Определение 2.1.1. *Владельцем объекта доступа называется пользователь, инициализировавший поток, в результате которого объект возник в системе¹, или определенный владельцем иным образом (получивший право владения объектом).*

Тем самым, в дополнение к основным положениям субъектно-объектной модели вводится специальное отображение множества объектов на множество субъектов доступа, называемое владением, ставящее в каждый фиксированный момент времени каждому объекту системы подмножество субъектов доступа, инициализированных пользователем-владельцем объекта.

Добровольное управление доступом выражается следующим правилом:

Правило 2.1.1. *Права доступа к объекту определяют (устанавливают) их владельцы.*

¹ Создание, копирование, переименование или физическое внесение (вместе с носителем).

Из данного правила следует, что заполнение и изменение ячеек матрицы доступа осуществляют субъекты пользователей-владельцев соответствующих объектов.

Нетрудно видеть, что подобный подход обеспечивает управление доступом в тех системах, в которых количество объектов доступа является значительным или неопределенным, так как переносит процесс управления на владельцев объектов, мощность подмножества объектов управления для которых в большинстве случаев существенно меньше общей мощности множества объектов в системе. Такая ситуация наиболее характерна для операционных систем.

Во многих КС право владения объектом его прежним владельцем может быть передано другому пользователю¹. Кроме того, как уже отмечалось, в ОС, составляющих основу любых КС, декомпозиция системы на субъекты и объекты может меняться в различные моменты времени (в результате операций создания, копирования, переименования и удаления объектов, с одной стороны, а с другой, в результате инициализации и прекращения функционирования пользователями субъектов доступа). В результате матрица доступа имеет динамический характер (появляются или уничтожаются строки или столбцы, изменяется содержимое ячеек). Поэтому права доступа в таких системах могут "гулять", распространяться по субъектам системы. В этом случае возникает проблема самого понятия безопасности в смысле главного метода ее обеспечения – разграничения доступа, и требуется исследование условий и *процессов распространения прав доступа*.

В теоретическом плане впервые данная проблема была исследована Харрисоном, Руззо и Ульманом, которые для этого разработали специальную формальную модель дискреционного доступа, называемую по их имени, или сокращенно модель **HRU**. Основные положения модели рассматриваются в следующем параграфе.

¹ При этом, разумеется, прежний владелец право владения автоматически теряет.

2.1.3. Модели распространения прав доступа

В моделях распространения прав доступа проблема безопасности КС рассматривается с точки зрения анализа возможности или невозможности получения каким-либо субъектом определенных прав доступа к определенному объекту. Иначе говоря, анализируются, прежде всего, изменения прав доступа субъектов к объектам в результате некоторых обусловленных операций (переходов), а не сами процессы осуществления доступов субъектов к объектам.

2.1.3.1. Модель Харисона-Руззо-Ульмана (HRU-модель)

Основные положения модели сводятся к следующему.

1. КС представляется тройкой сущностей:

- множеством исходных объектов $O (o_1, o_2, \dots, o_M)$;
- множеством исходных субъектов $S (s_1, s_2, \dots, s_N)$, при этом $S \subseteq O$;
- матрицей доступа A , каждая ячейка которой специфицирует права доступа к объектам из конечного набора прав доступа $R (r_1, r_2, \dots, r_K)$, т. е. $A[s, o] \subseteq R$.

Права доступа r_i , размещаемые в ячейках матрицы доступа $A[s, o]$, определяют совокупность допустимых (разрешенных) операций над объектом из полного набора возможных операций над объектами.

Заметим также, что модель HRU несколько отличается от рассмотренной в п. 1.3.2 субъектно-объектной модели КС, представляя субъектов доступа "активизированными" состояниями некоторого подмножества объектов системы (т. е. $S \subseteq O$), что, с одной стороны, огрубляет саму суть субъектов доступа, но, с другой стороны позволяет ввести понятие доступа субъекта к субъекту.

2. Функционирование системы рассматривается исключительно с точки зрения изменений в матрице доступа. Возможные изменения определяются шестью примитивными операторами Op :

- *Enter r into $A[s, o]$* – ввести право r в ячейку $A[s, o]$;
- *Delete r from $A[s, o]$* – удалить право r из ячейки $A[s, o]$;
- *Create subject s* – создать субъект s (т. е. новую строку матрицы A);
- *Create object o* – создать объект o (т. е. новый столбец матрицы A);
- *Destroy subject s* – уничтожить субъект s ;
- *Destroy object o* – уничтожить объект o .

В результате выполнения примитивного оператора осуществляется переход КС из состояния $Q = (S, O, A)$ в новое состояние $Q' = (S', O', A')$.

В табл. 2.1 приведены условия и особенности изменения состояния системы под воздействием примитивных операторов.

3. Состояния системы изменяются под воздействием запросов на модификацию матрицы доступа в виде команд следующего формата:

$$\begin{aligned}
 &\textbf{Command } \alpha(x_1, x_2, \dots, x_k) \\
 &\text{if } r_1 \text{ in } A[x_{s_1}, x_{o_1}] \text{ and} \quad (\text{условия выполнения команды}) \\
 &\quad r_2 \text{ in } A[x_{s_2}, x_{o_2}] \text{ and} \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad r_m \text{ in } A[x_{s_m}, x_{o_m}] \\
 &\text{then} \\
 &\quad op_1, op_2, \dots, op_n \quad (\text{операторы, составляющие команду})
 \end{aligned}
 \tag{2.1.3}$$

Каждое состояние системы Q_i является результатом выполнения некоторой команды α_i , применимой по ее условиям к предыдущему состоянию Q_{i-1}

$$Q_i = \alpha_i(Q_{i-1}), \tag{2.1.4}$$

и определяет отношения доступа, которые существуют между сущностями системы в виде множества субъектов, объектов и матрицы прав доступа.

Таблица 2.1

Примитивный оператор модели HRU	Условия выполнения	Новое состояние системы
<i>Enter r into $A[s, o]$</i>	$s \in S,$ $o \in O$	$S' = S, O' = O, A'[s, o] = A[s, o] \cup \{r\},$ если $(s', o') \neq (s, o) \Rightarrow A'[s', o'] = A[s, o]$
<i>Delete r from $A[s, o]$</i>	$s \in S,$ $o \in O$	$S' = S, O' = O, A'[s, o] = A[s, o] \setminus \{r\},$ если $(s', o') \neq (s, o) \Rightarrow A'[s', o'] = A[s, o]$
<i>Create subject s'</i>	$s' \notin S$	$S' = S \cup \{s'\}, O' = O \cup \{s'\},$ если $(s, o) \in S \times O \Rightarrow A'[s, o] = A[s, o],$ если $o \in O' \Rightarrow A'[s', o] = \emptyset,$ если $s \in S' \Rightarrow A'[s, s'] = \emptyset$
<i>Create object o'</i>	$o' \notin O$	$S' = S, O' = O \cup \{o'\},$ если $(s, o) \in S \times O \Rightarrow A'[s, o] = A[s, o],$ если $s \in S' \Rightarrow A'[s, o] = \emptyset$
<i>Destroy subject s'</i>	$s' \in S$	$S' = S \setminus \{s'\}, O' = O \setminus \{s'\},$ если $(s, o) \in S' \times O' \Rightarrow A'[s, o] = A[s, o]$
<i>Destroy object o'</i>	$o' \in O \setminus S$	$S' = S, O' = O \setminus \{o'\},$

		если $(s,o) \in S' \times O' \Rightarrow A'[s,o] = A[s,o]$
--	--	--

4. Безопасность системы определяется некоторыми условиями на начальное состояние системы Q_0 , а также особенностями системы команд α . Формулируется следующий критерий безопасности:

Определение 2.1.2. (Критерий безопасности в модели HRU). Система является безопасной относительно права r , если для заданного начального состояния $Q_0 = (S_0, O_0, A_0)$ не существует применимой к Q_0 последовательности команд, в результате которой право r будет занесено в ячейку матрицы $A[s,o]$, в которой оно отсутствовало в начальном состоянии Q_0 .

Действительно из самого понятия разграничения доступа вытекает необходимость наложения ограничений на определенные отношения доступа (определенных субъектов к определенным объектам по определенным операциям). Очевидно, что безопасность (состояние защищенности информации) в системе, заключается в том, чтобы эти ограничения соблюдались все время функционирования системы, и, в том числе, в начальном состоянии Q_0 . Иначе говоря, в неформальном выражении безопасность системы будет определяться тем, возможно или нет в процессе функционирования системы получение некоторым субъектом определенного права доступа к некоторому объекту, которым он изначально (т. е. в начальном состоянии Q_0) не обладал.

Для рассмотрения условий, при которых выполняется данный критерий, вводится понятие монооперационных систем.

Определение 2.1.3. Система называется монооперационной, если каждая команда α выполняет один примитивный оператор op_i .

Авторы модели HRU представили следующие теоремы, доказательство которых не приводится ввиду их громоздкости и доступности в литературе.

Теорема 2.1.1. Существует алгоритм, который проверяет, является ли исходное состояние монооперационной системы безопасным для данного права r .

Из теоремы 2.1.1 следует, что проблема безопасности может быть решена для монооперационных систем. К сожалению, теорема доказывает только само существование проверяющего алгоритма, но не дает каких-либо рекомендаций или других оснований для его разработки и построения.

Харрисон, Руззо и Ульман показали также, что безопасными могут быть монотонные системы (не содержащие операций *Delete* и *Destroy*), системы, не содержащие операций *Create*, и моно-условные системы (в которых запросы содержат только одно условие). Вместе с тем, все подобные системы существенно ограничены в функциональности.

Теорема 2.1.2. *Задача определения безопасности для данного права r в системах с запросами произвольного вида (2.1.3) является алгоритмически неразрешимой.*

Говоря иными словами, *теорема 2.1.2* утверждает, что поведение систем на основе модели HRU с точки зрения безопасности является непредсказуемым!!!

Помимо проблем с неопределенностью распространения прав доступа в системах на основе модели HRU была подмечена еще одна серьезная проблема – отсутствие контроля за порождением потоков информации, и, в частности, контроля за порождением субъектов, следствием чего могут быть так называемые "*тройанские*" программы. В модели HRU "правильность", легитимность иницируемых из объектов-источников субъектов доступа никак не контролируется. В результате, злоумышленник в системе может осуществить неправомерный доступ к информации на основе подмены свойств субъектов доступа.

Данные результаты, опубликованные в середине 70-х годов, вызвали обескураживающий эффект среди исследователей теории компьютерной безопасности, но, вместе с тем, стимулировали поиски других подходов к обеспечению проблемы безопасности.

В частности, для смягчения условий, в которых можно производить формальное доказательство безопасности, а также для введения контроля за порождением объектов была предложена модель "типизированной матрицы доступа" (модель Type Access Matrix – TAM).

2.1.3.2. Модель типизированной матрицы доступа

1. КС представляется четверкой сущностей:
 - множеством исходных объектов $O(o_1, o_2, \dots, o_M)$;
 - множеством исходных субъектов $S(s_1, s_2, \dots, s_N)$, при этом $S \subseteq O$;
 - матрицей доступа A , каждая ячейка которой специфицирует права доступа к объектам из конечного набора прав доступа $R(r_1, r_2, \dots, r_K)$, т. е. $A[s, o] \subseteq R$;
 - множеством (конечным набором) типов безопасности $T(t_1, t_2, \dots, t_L)$, с одним из которых создается любой объект (включая субъекты). Тип объекта впоследствии не меняется (т. н. *сильная организация типов*). Таким образом, в системе задана функция $f_t: O \rightarrow T$, ставящая в соответствие каждому объекту некоторый тип.
2. Вводятся примитивные операторы, изменяющие состояние матрицы доступа с учетом типизации объектов (объектов и субъектов доступа):

- *Enter r into $A[s,o]$* – ввести право r в ячейку $A[s,o]$;
- *Delete r from $A[s,o]$* – удалить право r из ячейки $A[s,o]$;
- *Create subject s of type t* – создать субъект s типа t ;
- *Create object o of type t* – создать объект o типа t ;
- *Destroy subject s* – уничтожить субъект s ;
- *Destroy object o* – уничтожить объект o .

В результате выполнения примитивного оператора осуществляется переход КС из состояния $Q = (S, O, A)$ в новое состояние $Q' = (S', O', A')$.

В табл. 2.2 приведены условия и особенности изменения состояния системы под воздействием примитивных операторов с учетом типизации объектов.

3. Состояния системы так же, как и в модели **HRU**, изменяются под воздействием запросов на модификацию матрицы доступа в виде команд, формат которых с учетом типизованного характера объектов системы имеет следующий вид:

$$\begin{aligned}
 &\textbf{Command } \alpha(x_1:t_1, x_2:t_2, \dots, x_k:t_k) \\
 &\text{if } r_1 \text{ in } A[x_{s_1}, x_{o_1}] \text{ and } \quad (\text{условия выполнения команды}) \\
 &\quad r_2 \text{ in } A[x_{s_2}, x_{o_2}] \text{ and} \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad r_m \text{ in } A[x_{s_m}, x_{o_m}] \\
 &\text{then} \\
 &\quad op_1, op_2, \dots, op_n \quad (\text{операторы, составляющие команду}).
 \end{aligned} \tag{2.1.5}$$

Таким образом, при выполнении команд на модификацию матрицы доступа вводится контроль типов фактических параметров команды, т. е. контроль типов объектов и субъектов, задействованных в условиях выполнения команды. На этой основе можно сформулировать ограничения, накладываемые на команды, при которых могут быть смягчены условия безопасности системы.

4. Для формулирования ограничений на запросы команд перехода в новое состояние системы определяются отношения между типами, в частности, понятие дочернего и родительского типа.

Таблица 2.2

Примитивный оператор модели ТАМ	Условия выполнения	Новое состояние системы
<i>Enter r into $A[s,o]$</i>	$s \in S,$ $o \in O$	$S'=S, O'=O, \forall o \in O \quad f'_i(o)=f_i(o),$ $A'[s,o]=A[s,o] \cup \{r\},$

		если $(s', o') \neq (s, o) \Rightarrow A'[s', o'] = A[s, o]$
Delete r from $A[s, o]$	$s \in S,$ $o \in O$	$S' = S, O' = O, \forall o \in O f'_i(o) = f_i(o),$ $A'[s, o] = A[s, o] \setminus \{r\},$ если $(s', o') \neq (s, o) \Rightarrow A'[s', o'] = A[s, o]$
Create subject s' of type t_s	$s' \notin S$	$S' = S \cup \{s'\}, O' = O \cup \{s'\},$ $f'_i(s') = t_s, \forall o \in O f'_i(o) = f_i(o),$ если $(s, o) \in S \times O \Rightarrow A'[s, o] = A[s, o],$ если $o \in O' \Rightarrow A'[s', o] = \emptyset,$ если $s \in S' \Rightarrow A'[s, s] = \emptyset$
Create object o' of type t_o	$o' \notin O$	$S' = S, O' = O \cup \{o'\},$ $f'_i(o') = t_o, \forall o \in O f'_i(o) = f_i(o)$ если $(s, o) \in S \times O \Rightarrow A'[s, o] = A[s, o],$ если $s \in S' \Rightarrow A'[s, o] = \emptyset$
Destroy subject s'	$s' \in S$	$S' = S \setminus \{s'\}, O' = O \setminus \{s'\},$ $\forall o \in O f'_i(o) = f_i(o), f'_i(s') = \text{не определено}$ если $(s, o) \in S' \times O' \Rightarrow A'[s, o] = A[s, o]$
Destroy object o'	$o' \in O \setminus S$	$S' = S, O' = O \setminus \{o'\},$ $\forall o \in O' f'_i(o) = f_i(o), f'_i(o') = \text{не определено}$ если $(s, o) \in S' \times O' \Rightarrow A'[s, o] = A[s, o]$

Определение 2.1.4. Тип t_i является дочерним типом в команде α , если в теле α имеет место один из следующих элементарных операторов: "Create subject s' of type t_i " или "Create object o' of type t_i ". В противном случае тип t_i является родительским типом.

Основываясь на понятии родительских и дочерних типов, можно описать взаимосвязи между различными типами с помощью графа отношений "наследственности" при операциях порождения сущностей (субъектов и объектов). Такой граф является ориентированным (орграфом) и называется "графом создания". Множество вершин графа образуется множеством типов безопасности T . Ребро от вершины t_i к вершине t_j в графе имеется тогда и только тогда, когда существует команда α , в которой тип t_i является родительским, а тип t_j дочерним.

Также, как и в модели HRU, используется понятие монотонной (MTAM) системы, которая не содержит примитивных операторов **Delete** и **Destroy**.

Определение 2.1.5. Реализация MTAM является ациклической тогда и только тогда, когда ее граф создания не содержит циклов.

В работе [29] показано, что критерий безопасности Хариссона-Руззо-Ульмана (определение 2.1.2) разрешим для ациклических реализаций системы с монотонной ТАМ, а требование одноусловности можно заменить требованием ациклическости графа создания системы. Требование ацикличес-

ности с формальной точки зрения позволяет существенно сократить и ограничить количество путей на графе создания при рассмотрении и доказательстве безопасности в системе. С неформальной точки зрения ацикличность означает, что последовательность состояний системы должна следовать определенному маршруту на графе создания, так как возникновению новых сущностей в системе должно предшествовать наличие объектов определенных родительских типов. В результате поведение системы становится более предсказуемым.

Кроме того, на основе специальной системы типов (например, типы "файл", "программа" для объектов; "user", "administrator", "auditor" для субъектов) в КС на основе **TAM** возможна организация эффективного контроля за порождением субъектов с нейтрализацией проблемы "тройных" программ.

2.1.3.3. Модель TAKE-GRANT

Еще одной моделью, имеющей важное теоретическое значение в исследовании процессов распространения прав доступа в системах, основанных на политике дискреционного доступа, является модель **TAKE-GRANT**, представленная Джонсом, Липтоном и Шнайдером в 1976 г.

Модель **TAKE-GRANT**, отталкиваясь от основных положений субъектно-объектной модели КС, использует аппарат теории графов для моделирования системы разграничения доступа и процессов ее изменения.

Основные положения модели сводятся к следующему.

1. КС рассматривается как граф $\Gamma(O, S, E)$, в котором множество вершин представлено (см. рис. 2.2):
 - множеством объектов O доступа;
 - множеством субъектов S доступа, причем $S \subseteq O$,а множество ребер:
 - множеством E установленных прав доступа (x, y, α) субъекта x к объекту y с правом α из конечного набора прав $\alpha \subseteq R(r_1, r_2, \dots, r_k) \cup \{t, g\}$, в том числе с двумя специфическими правами – правом **take** (t – право брать права доступа у какого-либо объекта по отношению к другому объекту) и правом **grant** (g – право предоставлять права доступа к определенному объекту другому субъекту).

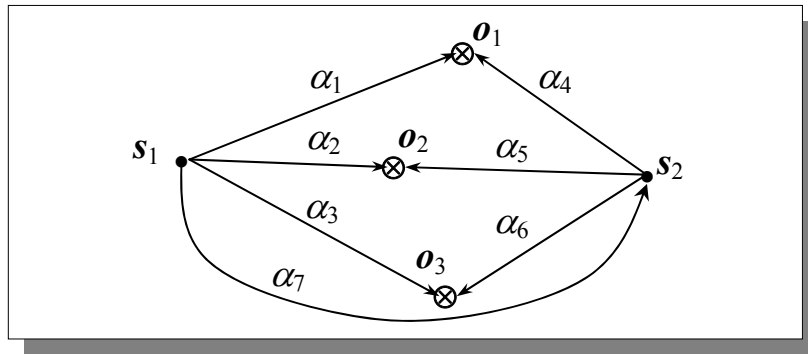


Рис.2.2. Граф доступов Γ в модели **TAKE-GRANT** (обозначения: $\bullet$ – вершины, соответствующие субъектам, $\otimes$ – вершины, соответствующие объектам доступа, $\alpha_i \subseteq R$ – права доступа)

Как и в модели **HRU**, субъекты рассматриваются в качестве активизированного состояния некоторых объектов, и поэтому граф Γ не является двудольным, так как могут существовать дуги (права доступа) между субъектами.

2. Состояния КС (т. е. состояние системы разграничения доступа) изменяются под воздействием команд 4-х видов.

2.1. Команда "**Брать**" – $take(\alpha, x, y, z)$ – см. рис. 2.3.

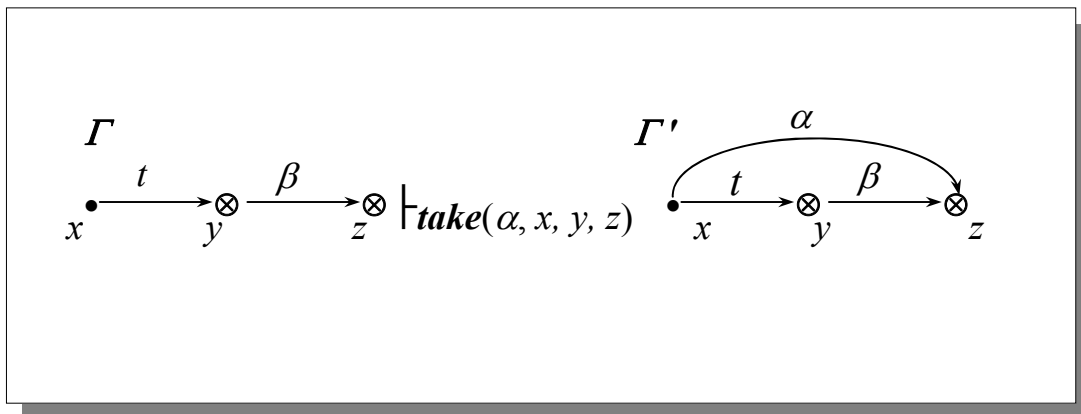


Рис. 2.3. Изменение состояния фрагмента графа доступов Γ по команде "**Брать**" – субъект x берет права доступа $\alpha \subseteq \beta$ на объект z у объекта y (обозначения: $\vdash_c$ – переход графа Γ в новое состояние Γ' по команде c ; $x \in S$; $y, z \in O$)

2.2. Команда "**Давать**" – $grant(\alpha, x, y, z)$ – см. рис. 2.4.

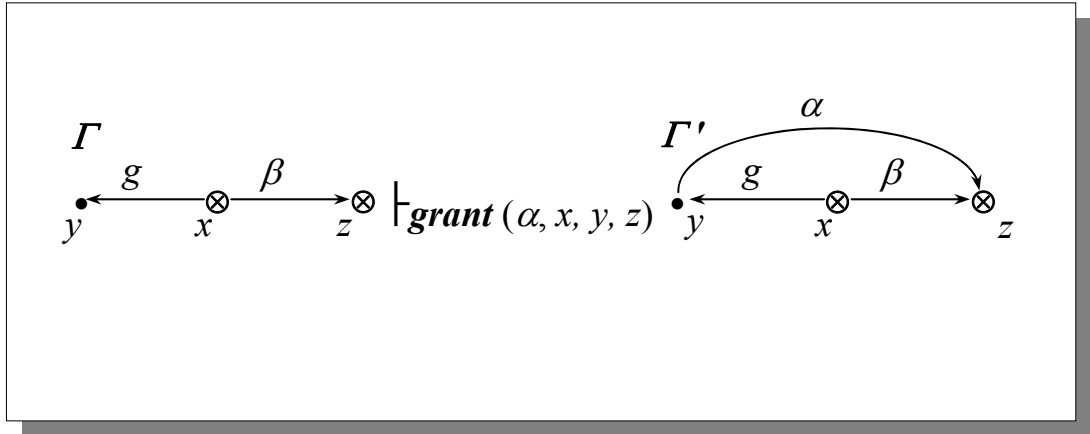


Рис. 2.4. Субъект x дает объекту y право $\alpha \subseteq \beta$ на доступ к объекту z

2.3. Команда "Создать" – $create(\beta, x, y)$ – см. рис. 2.5.

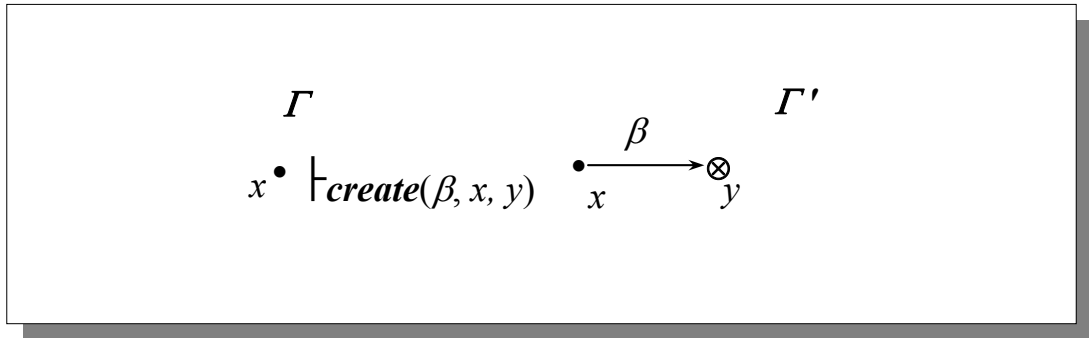


Рис. 2.5. Субъект x создает объект y с правами доступа на него $\beta^1 \subseteq R$ (y – новый объект, $O' = O \cup \{y\}$)

2.4. Команда "Удалить" – $remove(\alpha, x, y)$ – см. рис. 2.6.

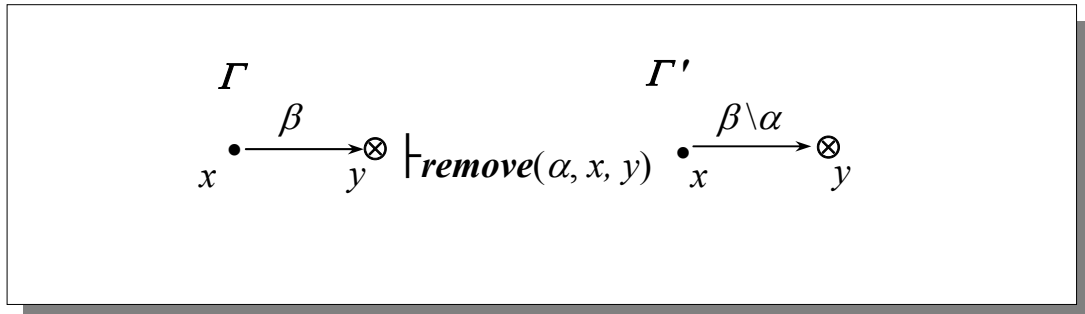


Рис. 2.6. Субъект x удаляет права доступа $\alpha \subseteq \beta$ на объект y

¹ в т. ч. с правами t , или g , или $\{t, g\}$.

В табл. 2.3 приведены условия и особенности изменения состояния системы под воздействием представленных выше команд.

3. Безопасность системы рассматривается с точки зрения возможности получения каким-либо субъектом прав доступа к определенному объекту (в начальном состоянии $\Gamma_0(O_0, S_0, E_0)$ такие права отсутствуют) при определенной кооперации субъектов путем последовательного изменения состояния системы на основе выполнения команд, представленных в табл. 2.3. Предметом анализа при этом являются установленные в начальный момент времени отношения между субъектами по получению и передаче прав доступа на объекты системы, а также возможные ограничения на дальнейшую кооперацию субъектов в процессе функционирования системы. Рассматриваются два возможных варианта:

- санкционированное получение прав доступа (безопасное функционирование системы);
- похищение прав доступа (условия, в которых безопасность не обеспечивается).

Таблица 2.3

Команда модели TAKE-GRANT	Условия выполнения	Новое состояние системы
$take(\alpha, x, y, z)$	$x \in S, (x, y, t) \in E,$ $(y, z, \beta)^1 \in E$ $x \neq z, \alpha \subseteq \beta$	$S'=S, O'=O,$ $E= E' \cup \{(x, z, \alpha)\}$
$grant(\alpha, x, y, z)$	$x \in S, (x, y, g) \in E,$ $(y, z, \beta) \in E$ $x \neq z, \alpha \subseteq \beta$	$S'=S, O'=O,$ $E= E' \cup \{(y, z, \alpha)\}$
$create(\beta, x, y)$	$x \in S, y \notin O$	$O'=O \cup \{y\},$ $S'=S \cup \{y\}$, если y – субъект $E= E' \cup \{(y, z, \beta)\}$
$remove(\alpha, x, y)$	$x \in S, y \in O,$ $(x, y, \beta) \in E, \alpha \subseteq \beta$	$S'=S, O'=O,$ $E= E' \setminus \{(x, y, \alpha)\} \cup \{(x, y, \beta)\}$

3.1. Санкционированное получение прав доступа

Вводятся следующие определения².

Определение 2.1.6. Для исходного состояния системы $\Gamma_0(O_0, S_0, E_0)$ и прав доступа $\alpha \subseteq R$ предикат "возможен доступ(α, x, y, Γ_0)" является истинным тогда и только тогда, когда существуют графы досту-

¹ (x, y, α) – обозначение дуги от субъекта x к объекту y с правами доступа α .

² п.п. 3.1, 3.2 излагаются по работе [18] с небольшими добавлениями и комментариями.

пов системы $\Gamma_1(O_1, S_1, E_1)$, $\Gamma_2(O_2, S_2, E_2)$, ..., $\Gamma_N(O_N, S_N, E_N)$, такие, что:

$$\Gamma_0(O_0, S_0, E_0) \vdash_{c_1} \Gamma_1(O_1, S_1, E_1) \vdash_{c_2} \dots \vdash_{c_N} \Gamma_N(O_N, S_N, E_N) \text{ и } (x, y, \alpha) \in E_N$$

где $c_1, c_2, \dots, c_N$ – команды вида 2.1, 2.2, 2.3 и 2.4.

Говоря иначе, доступ субъекта x к объекту y с правом $\alpha \subseteq R$, отсутствующий в начальном состоянии системы $(x, y, \alpha) \notin E_0$, возможен тогда и только тогда, когда существует последовательность перехода системы из состояния в состояние $(\Gamma_0, \Gamma_1, \dots, \Gamma_N)$ под воздействием команд 2.1, 2.2, 2.3 и 2.4.

Определение 2.1.7. Вершины графа доступов являются *tg-связными* (соединены *tg-путем*), если в графе между ними существует такой путь, что каждая дуга этого пути выражает право t или g (без учета направления дуг).

Вершины непосредственно *tg-связаны*, если *tg-путь* между ними состоит из единственной дуги.

Справедлива следующая теорема.

Теорема 2.1.3. В графе доступов $\Gamma_0(S_0, O_0, E_0)$, содержащем только вершины-субъекты, предикат "возможен доступ(α, x, y, Γ_0)" истинен тогда и только тогда, когда выполняются следующие условия:

Условие 2.1.3.1. Существуют субъекты $s_1, \dots, s_m$ такие, что $(s_i, y, \gamma_i) \in E_0$ для $i=1, \dots, m$ и $\alpha = \gamma_1 \cup \dots \cup \gamma_m$.

Условие 2.1.3.2. Субъект x соединен в графе Γ_0 *tg-путем* с каждым субъектом s_i для $i=1, \dots, m$.

Доказательство. Доказательство теоремы проводится для $m=1$, так как схема доказательства для этого случая легко обобщается на случай $m>1$.

При $m=1$ условия 2.1.3.1 и 2.1.3.2 формулируются следующим образом:

Условие 2.1.3.1. Существует субъект s такой, что справедливо $(s, y, \alpha) \in E_0$.

Условие 2.1.3.2. Субъекты x и s соединены *tg-путем* в графе Γ_0 .

Необходимость. Пусть истинен предикат "возможен доступ(α, x, y, Γ_0)". Тогда по определению 2.6 существует последовательность графов доступов $\Gamma_1(O_1, S_1, E_1)$, $\Gamma_2(O_2, S_2, E_2), \dots, \Gamma_N(O_N, S_N, E_N)$ такая, что: $\Gamma_0 \vdash_{c_1} \Gamma_1 \vdash_{c_2} \dots \vdash_{c_N} \Gamma_N$ и $(x, y, \alpha) \in E_N$, при этом N является минимальным, т. е. $(x, y, \alpha) \notin E_{N-1}$. Необходимость условий 2.1.3.1 и 2.1.3.2 можно доказать индукцией по N .

При $N=0$ очевидно $(x, y, \alpha) \in E_0$. Следовательно, условия 2.1.3.1, 2.1.3.2 выполнены.

Пусть $N > 0$, и утверждение теоремы истинно для $\forall k < N$. Тогда $(x, y, \alpha) \notin E_0$ и дуга (x, y, α) появляется в графе доступов Γ_N в результате применения к графу Γ_{N-1} некоторой команды c_N . Очевидно, это не команда "Создать" или "Удалить". Если c_N является командой "Брать" (или "Давать"), то по его определению $\exists s' \in S_{N-1}: (x, s', t) \in E_{N-1}$ (или $(s', x, g) \in E_{N-1}$), $(s', y, \alpha) \in E_{N-1}$ и $c_N = \text{take}(\alpha, x, s', y)$ (или $c_N = \text{grant}(\alpha, s', x, y)$).

Возможны два случая: $s' \in S_0$ и $s' \notin S_0$.

$s' \in S_0$. Тогда истинен предикат "возможен доступ $(\alpha, s', y, \Gamma_0)$ ", при этом число преобразований графов меньше N . Следовательно, по предположению индукции $\exists s \in S_0: (s, y, \alpha) \in E_0$ и s' соединен с s tg -путем в графе Γ_0 . Кроме этого, истинен предикат "возможен доступ (t, x, s', Γ_0) " (или "возможен доступ (g, s', x, Γ_0) "), при этом число преобразований графов меньше N . Следовательно, по предположению индукции $\exists s'' \in S_0: (s'', s', t) \in E_0$ и s'' соединен с x tg -путем в графе Γ_0 (или $(s'', x, g) \in E_0$ и s'' соединен с s' tg -путем в графе Γ_0). Таким образом, $\exists s \in S_0: (s, y, \alpha) \in E_0$ и субъекты x, s соединены tg -путем в графе Γ_0 . Выполнение условий 2.1.3.1 и 2.1.3.2 для случая $s' \in S_0$ доказано.

$s' \notin S_0$. Число N преобразований графов минимально, поэтому новые субъекты создаются только в тех случаях, когда без этого невозможна передача прав доступа. Следовательно, преобразования графов отвечают следующим требованиям:

- 1) субъект-создатель берет на созданный субъект максимально необходимый набор прав $\{t, g\}$;
- 2) каждый имеющийся в графе Γ_0 субъект не создает более одного субъекта;
- 3) созданный субъект не создает новых субъектов;
- 4) созданный субъект не использует правило "Брать" для получения прав доступа на другие субъекты.

Из перечисленных требований следует, что $\exists M < N - 1$, $\exists s'' \in S_0: c_M = \text{create}(\{g, t\}, s'', s')$, $c_N = \text{take}(\alpha, x, s', y)$ и истинен предикат "возможен доступ $(\alpha, s'', y, \Gamma_0)$ ". Отсюда следует, что истинен предикат "возможен доступ (t, x, s', Γ_M) ", а так как s'' – единственный субъект в графе Γ_M , имеющий права на субъект s' , то по предположению индукции s'' соединен с x tg -путем в графе Γ_0 . Из истинности предиката "возможен доступ $(\alpha, s'', y, \Gamma_0)$ " и по предположению индукции следует, что s и s'' соединены tg -путем в графе Γ_0 . Следовательно, $\exists s \in S_0: (s, y, \alpha) \in E_0$ и x, s соединены tg -путем в графе Γ_0 . Выполнение условий 2.1.3.1 и 2.1.3.2 для случая $s' \notin S_0$ доказано. Индуктивный шаг доказан.

Достаточность. Пусть выполнены условия 2.1.3.1 и 2.1.3.2 Доказательство проведем индукцией по длине tg -пути, соединяющего субъекты x и s .

Пусть $N = 0$. Следовательно, $x = s$, $(x, y, \alpha) \in E_0$ и предикат "возможен доступ (α, x, y, Γ_0) " истинен.

Пусть $N = 1$, т. е. $\exists s \in S_0: (s, y, \alpha) \in E_0$ и субъекты x , s непосредственно tg -связаны. Тогда возможны четыре случая такого соединения x и s , для каждого из которых нетрудно указать последовательность преобразований графа, требуемую для передачи прав доступа (см. рис. 2.7).

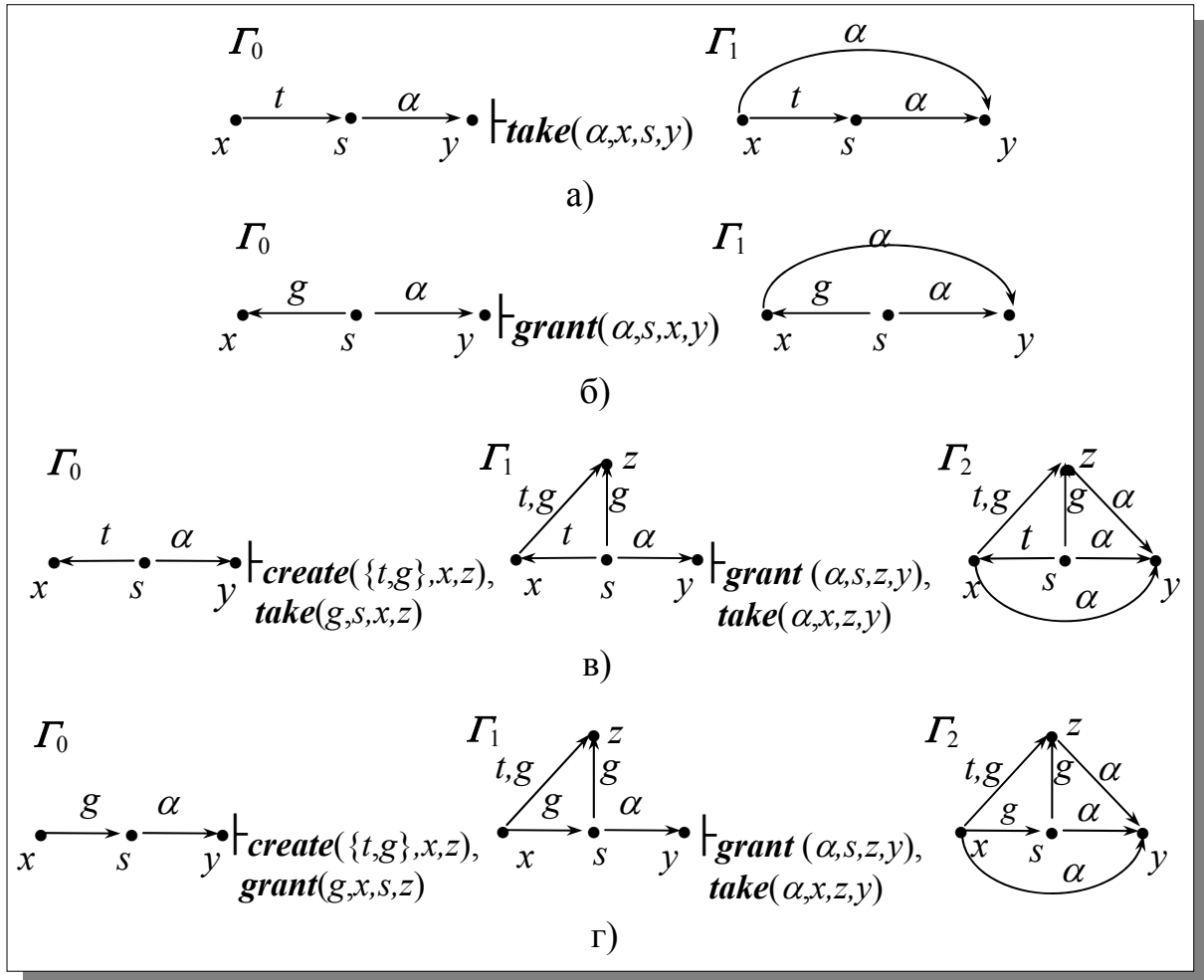


Рис. 2.7. Возможные случаи непосредственной tg -связности x и s , а также последовательность преобразований графа Γ_0 до получения доступа $(x, y, \alpha) \in E_N$

Таким образом, при $N = 1$ предикат "возможен доступ (α, x, y, Γ_0) " истинен.

Пусть $N > 1$. Рассмотрим вершину z , находящуюся на tg -пути между x и s , и являющуюся смежной с s в графе Γ_0 . Тогда по доказанному для случая $N = 1$ существует последовательность преобразований графов дос-

тупов $\Gamma_0 \vdash_{c_1} \Gamma_1 \vdash_{c_2} \dots \vdash_{c_K} \Gamma_K : (z, y, \alpha) \in E_K$ и длина tg -пути между z и x равна $N-1$, что позволяет применить предположение индукции.

Теорема доказана. ■

Несмотря на кажущуюся сложность *теоремы 2.1.3*, она иллюстрирует вообще говоря довольно простой и интуитивно понятный порядок вещей – если кто-либо желает получить какие-либо права на кого-то, то при отсутствии ограничений на взаимоотношения субъектов, он может скооперироваться с третьим(ми) субъектом, кто такие права уже имеет, и позаимствовать их у него.

Авторы модели провели анализ санкционированного получения прав доступа также и в графе произвольного вида, состоящего как из субъектов, так и объектов. Для рассмотрения предиката "возможен доступ" в произвольном графе вводится ряд дополнительных понятий.

Определение 2.1.8. *Островом в произвольном графе доступов Γ называется его максимальный tg -связный подграф, состоящий только из вершин субъектов.*

Заметим, что остров, по сути, характеризует некую группу субъектов, кооперация которых в правах доступа не ограничена и процессы получения доступа внутри группы описываются *теоремой 2.3*.

Определение 2.1.9. *Мостом в графе доступов Γ называется tg -путь, концами которого являются вершины-субъекты; при этом словарная запись tg -пути должна иметь вид $\vec{t}^*, \vec{t}^*, \vec{t}^* \vec{g} \vec{t}^*, \vec{t}^* \vec{g} \vec{t}^*$, где символ $*$ означает многократное (в том числе нулевое) повторение.*

Определение 2.1.10. *Начальным пролетом моста в графе доступов Γ называется tg -путь, началом которого является вершина-субъект; при этом словарная запись tg -пути должна иметь вид $\vec{t}^* \vec{g}$.*

Определение 2.1.11. *Конечным пролетом моста в графе доступов Γ называется tg -путь, началом которого является вершина-субъект; при этом словарная запись tg -пути должна иметь вид $\vec{t}^*$.*

Справедлива следующая теорема.

Теорема 2.1.4. *В произвольном графе доступов $\Gamma_0 (S_0, O_0, E_0)$ предикат "возможен доступ(α, x, y, Γ_0)" истинен тогда и только тогда, когда выполняются условия 2.1.4.1, 2.1.4.2 и 2.1.4.3:*

Условие 2.1.4.1. *Существуют объекты $s_1, \dots, s_m$ такие, что $(s_i, y, \gamma_i) \in E_0$ для $i=1, \dots, m$ и $\alpha = \gamma_1 \cup \dots \cup \gamma_m$.*

Условие 2.1.4.2. *Существуют вершины-субъекты $x_1', \dots, x_m'$ и $s_1', \dots, s_m'$ такие, что:*

- $x = x_i'$ или x_i' соединен с x начальным пролетом моста для $i=1, \dots, m$;
- $s_i = s_i'$ или s_i' соединен с s_i конечным пролетом моста для $i=1, \dots, m$.

Условие 2.1.4.3. Для каждой пары (x'_i, s'_i) , $i=1, \dots, m$, существуют острова $I_{i1}, \dots, I_{iu_i}$, $u_i \geq 1$, такие, что $x'_i \in I_{i1}$, $s'_i \in I_{iu_i}$, и мосты между островами I_{ij} и I_{ij+1} .

Доказательство. Доказательство теоремы проводится для $m=1$, так как схему доказательства для этого случая легко продолжается на случай $m>1$. Для большей наглядности доказательства можно привести пример рассматриваемого по условиям теоремы графа доступа – см. рис. 2.8.

При $m=1$ условия 2.1.4.1, 2.1.4.2 и 2.1.4.3 формулируются следующим образом:

Условие 2.1.4.1. $\exists s \in O_0 : (s, y, \alpha) \in E_0$.

Условие 2.1.4.2. $\exists x', s' \in S_0 :$

- $x = x'$ или x' соединен с x начальным пролетом моста;
- $s = s'$ или s' соединен с s конечным пролетом моста.

Условие 2.1.4.3. Существуют острова $I_1, \dots, I_u$ такие, что $x' \in I_1$, $s' \in I_u$ и мосты между островами I_j и I_{j+1} для $j=1, \dots, u-1$.

Необходимость. Пусть истинен предикат "возможен доступ (α, x, y, Γ_0) ". По определению истинности предиката существует последовательность графов доступов $\Gamma_1(O_1, S_1, E_1)$, $\Gamma_2(O_2, S_2, E_2), \dots, \Gamma_N(O_N, S_N, E_N)$, такая, что: $\Gamma_0 \vdash_{c_1} \Gamma_1 \vdash_{c_2} \dots \vdash_{c_N} \Gamma_N$ и $(x, y, \alpha) \in E_N$, при этом N является минимальным, т. е. $(x, y, \alpha) \notin E_{N-1}$. Докажем необходимость условий 2.1.4.1, 2.1.4.2, 2.1.4.3 индукцией по N .

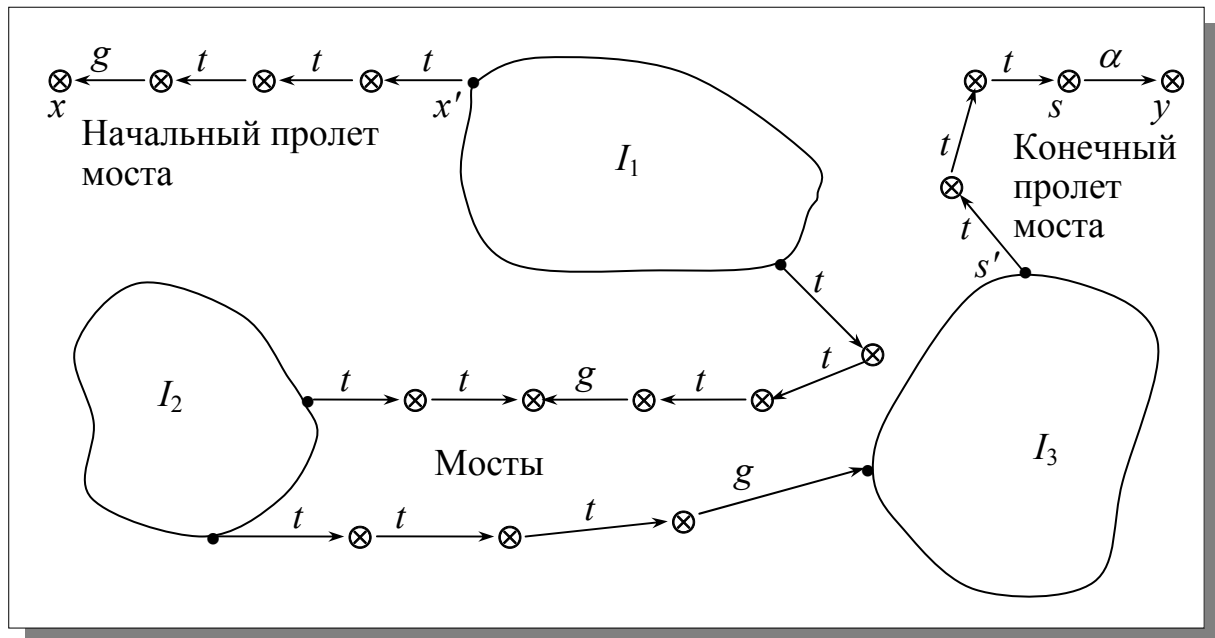


Рис. 2.8. Пример графа доступов с возможностью передачи объекту x прав доступа α на объект y

При $N=0$ очевидно $(x, y, \alpha) \in E_0$. Следовательно, условия 2.1.4.1, 2.1.4.2, 2.1.4.3 выполнены.

Пусть $N>0$ и утверждение теоремы истинно для $\forall k < N$. Тогда $(x, y, \alpha) \notin E_0$ и дуга (x, y, α) появляется в графе доступов Γ_N в результате применения к графу Γ_{N-1} некоторой команды c_N . При этом возможны два случая $x \in S_0$ и $x \notin S_0$.

Если $x \notin S_0$, то $\exists x_1 \in S_{N-1} : c_N = \text{grant}(\alpha, x_1, x, y)$. С учетом минимальности N и замечаний, сделанных при доказательстве теоремы 2.1.3, можно считать, что $x_1 \in S_0$. Следовательно:

1. Истинен предикат "возможен доступ(g, x_1, x, Γ_0)" с числом преобразований графов, меньшим N . Тогда по предположению индукции выполнены условия 2.1.4.1, 2.1.4.2, 2.1.4.3:

- $\exists x_2 \in O_0 : (x_2, x, g) \in E_0$;
- $\exists x' \in S_0$, соединенный с x_2 конечным пролетом моста;
- существуют острова $I_1, \dots, I_v, v \geq 1$ такие, что $x_1 \in I_v, x' \in I_1$, и мосты между островами I_j и I_{j+1} ;

2. Истинен предикат "возможен доступ(α, x_1, y, Γ_0)" с числом преобразований графов, меньшим N . Тогда по предположению индукции выполнены условия 2.1.4.1, 2.1.4.2, 2.1.4.3:

- $s \in O_0 : (s, y, \alpha) \in E_0$;
- $\exists s' \in S_0 : s = s'$ или s' соединен с s конечным пролетом моста;
- существуют острова $I_1, \dots, I_u, (v-u) \geq 1$, такие, что $x_1 \in I_v, s' \in I_1$ и мосты между островами I_j и I_{j+1} , для $j = v, \dots, u-1$.

Путь, соединяющий вершины x', x_2, x есть начальный пролет моста. Таким образом, для случая $x \notin S_0$ условия выполняются, и индуктивный шаг доказан.

Если $x \in S_0$, то п.1 условия 2.1.4.2 теоремы 2.1.4 очевидно выполняется. Многократно применяя технику доказательства, использованную выше, можно доказать индуктивный шаг и в данном случае.

Достаточность. Условия 2.1.4.1, 2.1.4.2, 2.1.4.3 конструктивны.

По условию 2.1.4.1 существует объект s , который обладает правами α на объект y . По п. 2 условия 2.1.4.2 существует субъект s' , который либо совпадает с s , либо по конечному пролету моста может забрать у субъекта s права α на объект y .

По теореме 2.1.3 права доступа, полученные одним субъектом, принадлежащим острову, могут быть переданы любому другому субъекту острова. По условию 2.1.4.3 между островами существуют мосты, по которым возможна передача прав доступа. В качестве примера на рис. 2.9 представляется последовательность преобразований графа доступов при передаче прав по мосту вида $\bar{t} \xrightarrow{g} \bar{t}$. По п.1 условия 2.1.4.2 теоремы 2.1.4 существует субъект x' , который или совпадает с x , или, получив права дос-

тупа, может передать их x по начальному пролету моста. Теорема доказана. ■

Смысл *теоремы 2.1.4* состоит в том, что если в системе разграничения доступа в начальном состоянии между двумя какими-либо объектами имеется tg -путь, включающий, в том числе, мосты между островами, то найдется такая последовательность команд вида 2.1, 2.2 и 2.3, в результате которой первый объект получит необходимые права доступа над другим объектом. Существенным при этом является отсутствие каких-либо ограничений на кооперацию субъектов и объектов доступа, в частности, отсутствие запретов на передачу прав доступа к объекту субъектами, изначально обладающими необходимыми правами на объект, представляющий интерес для других субъектов. Говоря иначе, подобный порядок вещей характеризует идеальное сотрудничество и доверие между субъектами доступа.

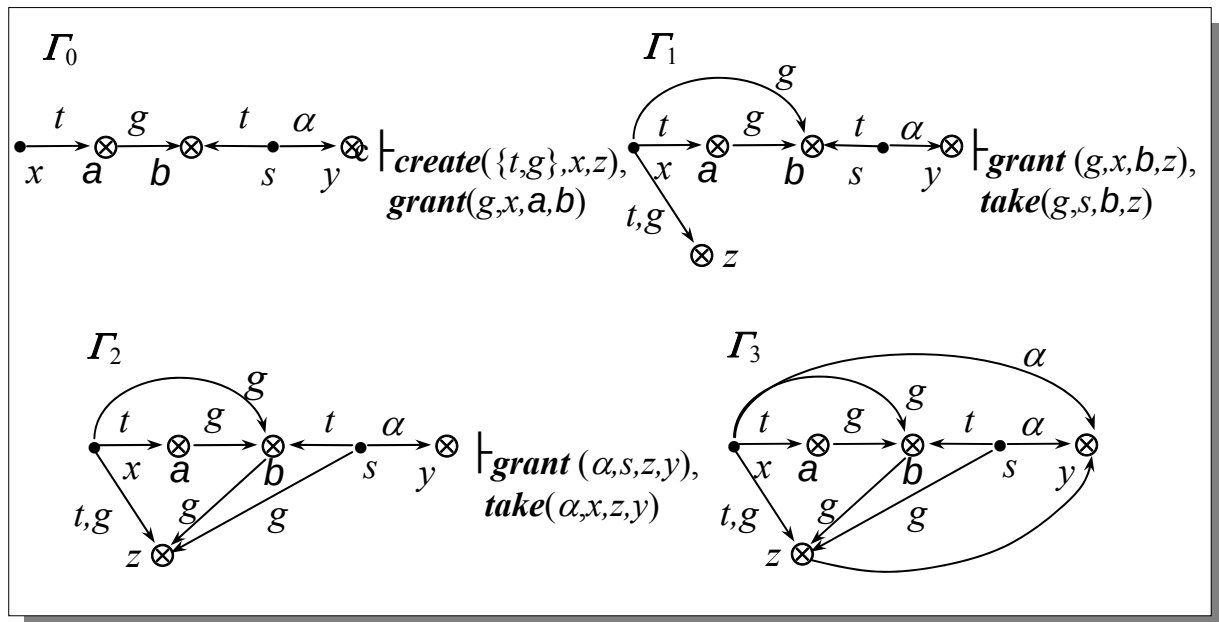


Рис. 2.9. Пример передачи прав доступа по мосту вида $\bar{t} \bar{g} \bar{t}$

В модели также анализируются условия, в которых получение прав доступа возможно и без участия в передаче прав субъектов (операция **grant**), изначально обладающими правами доступа на объект.

3.2. Похищение прав доступа

Пусть $x, y \in \mathbf{O}_0$ – различные объекты графа доступа $\Gamma_0 (\mathbf{S}_0, \mathbf{O}_0, \mathbf{E}_0)$.

Определение 2.1.12. Для исходного состояния системы $\Gamma_0 (\mathbf{O}_0, \mathbf{S}_0, \mathbf{E}_0)$ и прав доступа $\alpha \subseteq \mathbf{R}$ предикат "возможно похищение (α, x, y, Γ_0) " является истинным тогда и только тогда, когда существуют графы доступов системы $\Gamma_1 (\mathbf{O}_1, \mathbf{S}_1, \mathbf{E}_1), \Gamma_2 (\mathbf{O}_2, \mathbf{S}_2, \mathbf{E}_2), \dots, \Gamma_N (\mathbf{O}_N, \mathbf{S}_N, \mathbf{E}_N)$ такие, что:

$\Gamma_0(O_0, S_0, E_0) \vdash_{c_1} \Gamma_1(O_1, S_1, E_1) \vdash_{c_2} \dots \vdash_{c_N} \Gamma_N(O_N, S_N, E_N)$ и $(x, y, \alpha) \in E_N$

где $c_1, c_2, \dots, c_N$ – команды вида 2.1, 2.2, 2.3 и 2.4;

при этом, если $\exists (s, y, \alpha) \in E_0$, то $\forall z \in S_j, j=0, 1, \dots, N$ выполняется:

$c_1 \neq \text{grant}(\alpha, s, z, y)$.

Говоря иначе, согласно определению 2.1.11 похищением прав является процесс получения прав доступа на какой-либо объект без предоставления прав третьим субъектам со стороны субъекта, обладающего в начальном состоянии требуемыми правами на объект "интереса".

Справедлива следующая теорема.

Теорема 2.1.5. В произвольном графе доступов $\Gamma_0(S_0, O_0, E_0)$ предикат "возможно похищение (α, x, y, Γ_0) " истинен тогда и только тогда, когда выполняются условия 2.1.5.1, 2.1.5.2 и 2.1.5.3:

Условие 2.1.5.1. $(x, y, \alpha) \notin E_0$.

Условие 2.1.5.2. Существуют субъекты $s_1, \dots, s_m$ такие, что $(s_i, y, \gamma_i) \in E_0$ для $i=1, \dots, m$ и $\alpha = \gamma_1 \cup \dots \cup \gamma_m$.

Условие 2.1.5.3. Являются истинными предикаты "возможен доступ (t, x, s_i, Γ_0) " для $i=1, \dots, m$.

Доказательство. Аналогично доказательству теоремы 2.1.4.

Отметим, что теорема 2.1.5 также выражает интуитивно понятную ситуацию – если политика разграничения доступа в КС запрещает субъектам, имеющим в исходном состоянии права доступа к определенным объектам, непосредственно предоставлять эти права другим субъектам¹, которые изначально такими правами не обладают, то, тем не менее, такие первоначально "обделенные" субъекты могут получить данные права при наличии в графе доступов возможностей получения доступа с правом t к первым субъектам.

Таким образом, модель TAKE-GRANT играет важную методологическую роль, предоставляя теоретико-графовый инструмент анализа систем разграничения доступа с точки зрения санкционированного и несанкционированного со стороны определенных субъектов распространения прав доступа в рамках дискреционной политики.

2.1.3.4. Расширенная модель TAKE-GRANT

Выразительные методологические возможности модели TAKE-GRANT позволили на ее основе разработать *расширенную модель TAKE-GRANT*, играющую важную роль в исследовании возможностей *неявных информационных потоков*² в дискреционных системах разграничения доступа.

¹ т. е. такие потоки, для которых в графе доступов Γ_0 нет дуг "g" от этих субъектов.

² Применительно к системам с мандатным принципом разграничения доступа используется термин "скрытый канал утечки информации".

Введем следующее определение.

Определение 2.1.13. *Неявным информационным потоком между объектами системы называется процесс переноса информации между ними без их непосредственного взаимодействия.*

Наиболее простым и наглядным примером неявного информационного потока является наличие общего буфера (объекта с правом доступа к нему **Read, Write**) у двух субъектов. Тогда один из субъектов, просматривая (читая) информацию в буфере, может искать и находить информацию из объектов, которые доступны другому пользователю, и информация из которых в процессе работы с ними может оказаться в общем буфере или, скажем, в общей мусорной информационной корзине. В результате может существовать поток без непосредственного взаимодействия субъекта с объектом доступа.

Основные положения *расширенной модели TAKE-GRANT* сводятся к следующему.

1. КС рассматривается как граф $\Gamma(O, S, E)$, в котором множество вершин представлено:
 - множеством объектов O доступа;
 - множеством субъектов S доступа, причем $S \subseteq O$,
 а множество ребер:
 - множеством установленных прав доступа (x, y, α) субъекта x к объекту y с правом α из набора прав доступа R , включающего всего два вида (методов) доступа – **Read** и **Write**.
2. Для исследования процессов возникновения неявных информационных потоков вводятся шесть команд (операций) преобразования графа доступов¹, каждая из которых сопровождается порождением мнимой дуги, собственно и отображающей неявный информационный поток между объектами системы:

2.1. Команда (без названия) – см. рис. 2.10.

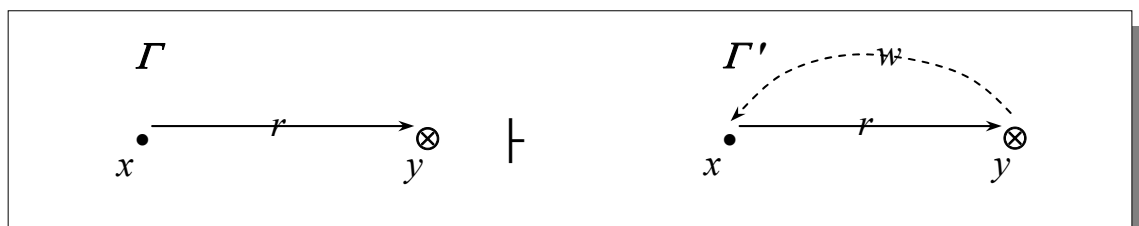


Рис. 2.10. Субъект x получает возможность записи (в себя²) информации, осуществляя доступ r к объекту y

¹ Для акцентирования характера неявных информационных потоков как потенциально-возможных команды преобразований графа доступов в обычной модели **TAKE-GRANT** называют командами по правилам "де-юре", в расширенной модели командами по правилам "де-факто".

² Здесь и далее в расширенной модели **TAKE-GRANT** потоки из $[v]$ субъекта понимаются как потоки из $[v]$ ассоциированных с ним (с субъектом) объектов-данных.

2.2. Команда (без названия) – см. рис. 2.11.

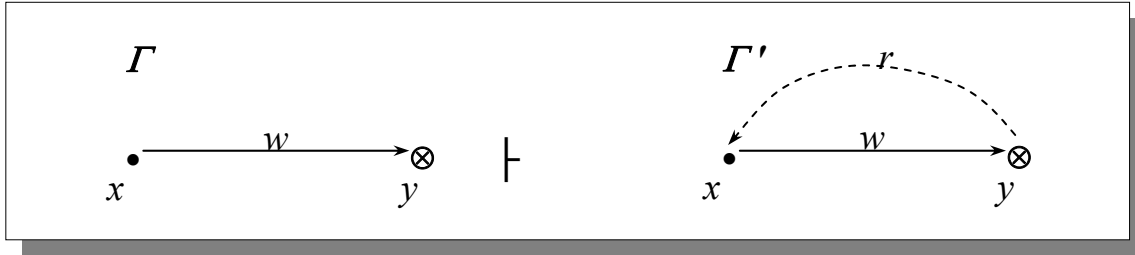


Рис. 2.11. Субъект x получает возможность чтения информации, осуществляя доступ w к объекту y

2.3. Команда **post** (x, y, z) – см. рис. 2.12.

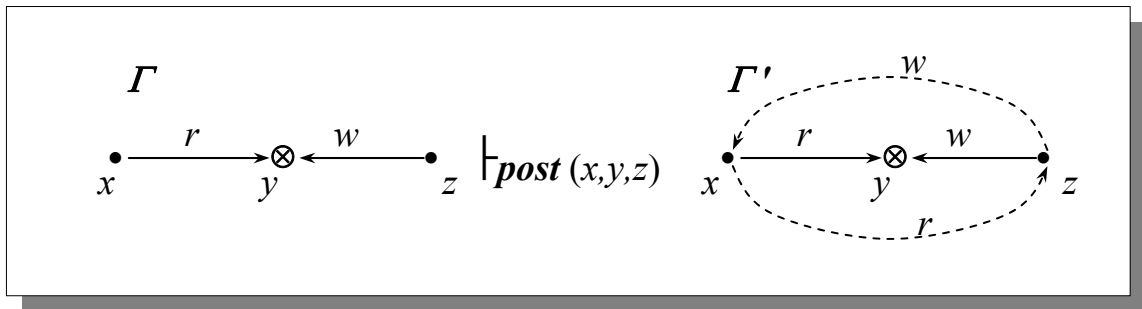


Рис. 2.12. Субъект x получает возможность чтения информации от (из) другого субъекта z , осуществляя доступ r к объекту y , к которому субъект z осуществляет доступ w , а субъект z , в свою очередь, получает возможность записи своей информации в субъект x

2.4. Команда **spy** (x, y, z) – см. рис. 2.13.

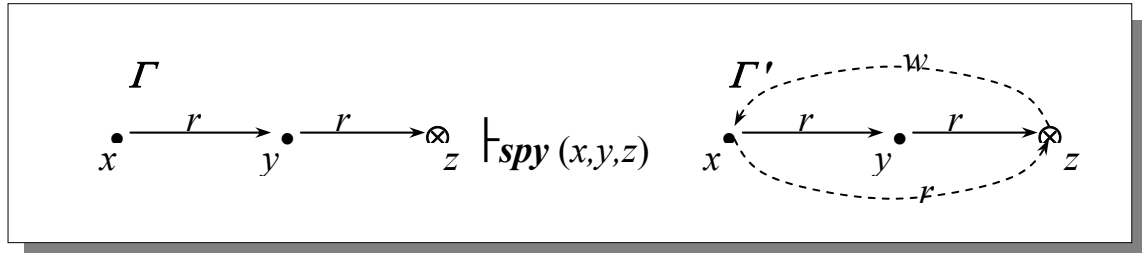


Рис. 2.13. Субъект x получает возможность чтения информации из объекта z , осуществляя доступ r к субъекту y , который, в свою очередь, осуществляет доступ r к объекту z , при этом также у субъекта x возникает возможность записи к себе информации из объекта z

2.5. Команда **find** (x, y, z) – см. рис. 2.14.

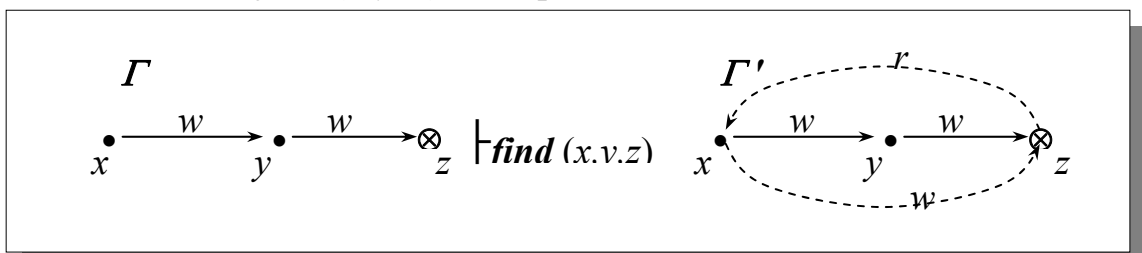


Рис. 2.14. Субъект x получает возможность чтения информации из объекта z , осуществляя доступ w к субъекту y , который, в свою очередь, осуществляет

доступ w к объекту z , при этом также у субъекта x возникает возможность записи к себе информации из объекта z

2.6. Команда **pass** (x, y, z) – см. рис. 2.15.

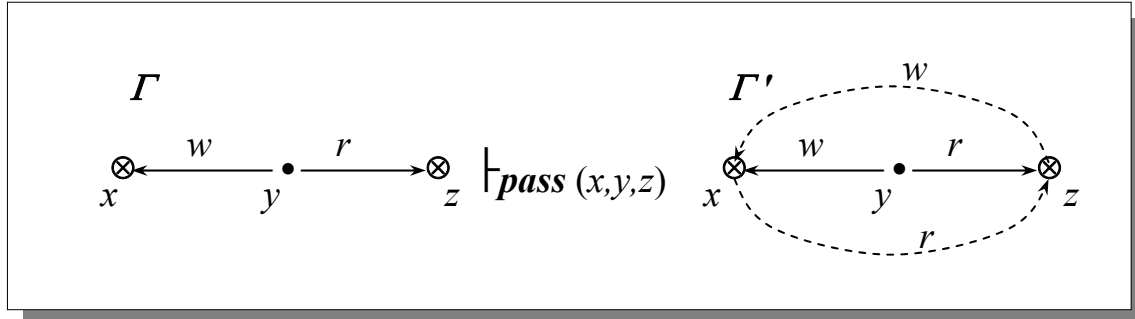


Рис. 2.15. При осуществлении субъектом y доступа r к объекту z возникает возможность внесения из него информации в другой объект x , к которому субъект y осуществляет доступ w , и, кроме того, возникает возможность получения информации (чтения) в объекте x из объекта z

3. Анализ возможности возникновения неявного информационного канала (потока) между двумя произвольными объектами (субъектами) x и y системы осуществляется на основе поиска и построения в графе доступов пути между x и y , образованного мнимыми дугами, порождаемыми применением команд 2.1,..., 2.6 к различным фрагментам исходного графа доступов.

Процедуры построения такого пути могут основываться на известных поисковых алгоритмах в графах, в частности, на методах поиска в глубину (ПВГ) и поиска в ширину (ПВШ), широко применяемых в сфере задач дискретной оптимизации. Алгоритмический характер подобных процедур позволяет их легко автоматизировать. Заметим также, что такие процедуры могут приводить к построению не одного, а нескольких путей возможного возникновения неявных информационных потоков. При этом становится возможным анализ и решение, если так можно выразиться, "тонких" задач относительно систем разграничения доступа:

- при допущении справедливости гипотезы, или при наличии достоверных фактов о состоявшемся неявном информационном потоке от одного объекта(субъекта) к другому объекту(субъекту), анализировать и выявлять круг возможных субъектов-"заговорщиков" подобного несанкционированного информационного потока;
- для какой-либо пары объектов (субъектов) осуществлять анализ определенных количественных характеристик возможности неявного информационного потока между ними по тому или иному маршруту.

Решение задач первого вида возможно на основе уже упоминавшихся методов ПВГ и ПВШ на графах. В качестве количественных характеристик в задачах второго вида могут быть выбраны характеристики, выражающие

в интервальной или ранговой шкале вероятность возникновения канала в зависимости от количественных параметров (длина пути канала на графе доступов) или качественных параметров (вид канала – сочетанием каких команд "де-факто" и по каким субъектам образован соответствующий путь на графе доступов).

Возможные каналы неявных информационных потоков определяются конкретным графом доступов КС, который, в свою очередь, отражает установленные для пользователей-субъектов права доступа к объектам системы в рамках принятой политики разграничения доступа. Система разграничения доступа должна обеспечивать доступ пользователей только к тем объектам, которые им необходимы для выполнения функциональных обязанностей или необходимы на основе других соображений. Иначе говоря, структура системы объектов доступа и система разграничения доступа к ним должны быть построены таким образом, чтобы не было избыточных прав доступа, но и не было препятствий в доступе к действительно необходимым объектам. При этом предоставление прав доступа пользователю к требуемому объекту в рамках дискреционного доступа может быть организовано по различным схемам – через различные цепочки последовательно иницилируемых субъектов, через различное сочетание прав доступа по прямым назначениям и прав доступа к объекту, иерархически наследуемых от объектов-контейнеров, в которые входит объект.

Отсюда следует, что задача проектирования системы разграничения доступа (формирование исходного графа доступов Γ_0) имеет не единственное решение, а множество решений. С точки зрения *расширенной модели TAKE-GRANT* каждое такое решение характеризуется своими каналами возможных неявных информационных потоков, выражающиеся, в том числе, и определенными количественными показателями, что может быть использовано в качестве основы критерия для оптимизации системы разграничения доступа.

2.2. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ МАНДАТНОЙ ПОЛИТИКИ

2.2.1. Общая характеристика политики мандатного доступа

Исходным толчком к разработке мандатной политики, вероятно, послужили проблемы с контролем распространения прав доступа, и, в особенности, проблема "троянских" программ в системах с дискреционным доступом. Исследователи и критики дискреционной политики, понимая, что основная проблема дискреционных моделей заключается в отсутствии контроля за информационными потоками, проанализировали, каким образом подобные проблемы решаются в секретном делопроизводстве.

Было отмечено следующее.

1. Разграничение доступа и порядок работы с конфиденциальными "бумажными" документами организуются на основе парадигмы градации доверия определенным группам работников в отношении государственных секретов определенной степени важности. С этой целью вводится система *уровней безопасности*, или иначе уровней секретности. Работники с самым высоким уровнем безопасности (уровнем доверия), могут работать с документами самой высокой степени секретности. На более низком уровне доверия, т. е. на более низком уровне безопасности, вводятся ограничения в отношении работы с документами более высокого уровня секретности и т. д. Соответственно, все работники получают т. н. *допуск* к работе с секретными документами определенного уровня, а документы снабжаются специальной меткой, отражающей требования к уровню безопасности при работе с ними, – т. н. *гриф секретности*.
2. Критерием безопасности является невозможность получения информации из документов определенного уровня безопасности работником, чей уровень безопасности, т. е. уровень доверия, ниже, чем уровень безопасности соответствующих документов.

Данный критерий безопасности фактически означает запрет определенных информационных потоков, которые трактуются как опасные и недопустимые.

Кроме того, были проанализированы правила и система назначений, изменений, лишений и т. д. допусков сотрудников к работе с секретными документами, правила создания, уничтожения документов, присвоения или изменения грифов их секретности, в том числе и рассекречивания, а также другие особенности работы с секретными документами. В частности было отмечено, что правила получения доступа к документам различаются в зависимости от характера работы с ними – изучение (*чтение*) или изменение (создание, уничтожение, внесение дополнений, редактирование, т. е. *за-*

пись в них). На этой основе было "выявлено" два основных правила, гарантирующих безопасность:

Правило 2.2.1.(no read up (NRU) – нет чтения вверх). *Работник не имеет права знакомиться с документом (читать), гриф секретности (уровень безопасности) которого выше его степени допуска (уровня безопасности).*

Правило 2.2.2.(no write down (NWD) – нет записи вниз). *Работник не имеет права вносить информацию (писать) своего уровня секретности в документ с более низким уровнем секретности (с более низким грифом секретности).*

Первое правило является естественным и очевидным способом обеспечения безопасности при осуществлении *информационных потоков из документов к работникам* и иначе может быть сформулировано так: – работнику нельзя получать информацию, уровень секретности которой выше его уровня доверия. Второе правило обеспечивает безопасность при осуществлении *информационных потоков от работника к документу* и иначе может быть сформулировано так: – работнику нельзя передавать информацию своего уровня секретности в тех случаях, когда в результате передачи с ней могут ознакомиться работники с более низким уровнем безопасности.

Формализация данных механизмов разграничения доступа в секретном делопроизводстве применительно к субъектно-объектной модели КС показала необходимость решения ряда следующих задач:

- разработка процедур формализации правила **NRU**, и в особенности правила **NWD**;
- построение формального математического объекта и процедур, адекватно отражающих систему уровней безопасности (систему допусков и грифов секретности).

Нетрудно увидеть, что при представлении работников, работающих с секретными документами, субъектами доступа, а секретных документов в качестве объектов доступа КС, буквальное следование правилу **NWD** приводит к автоматическому включению в механизмы обеспечения безопасности субъективного фактора в лице субъекта-пользователя, который при внесении информации в объекты-документы с более низким грифом секретности должен субъективно оценить соответствие вносимой информации уровню безопасности документа. Задача исключения данного субъективного фактора может решаться различными способами, самым простым из которых является полный запрет изменения субъектами (доступ *write*) объектов с уровнем безопасности, более низким, чем уровень безопасности соответствующих субъектов.

При этом, однако, помимо существенного снижения функциональности КС, логика такого запрета, автоматически приводит к "не

запрету" (т. е. к разрешению) возможностей изменения объектов-документов с более высоким уровнем безопасности, чем уровень безопасности соответствующих субъектов-пользователей. Действительно, внесение информации более низкого уровня секретности в объекты с более высоким уровнем секретности не может привести к нарушению безопасности системы в смысле рассмотренного выше критерия безопасности¹.

В качестве основы для решения второй задачи при создании моделей мандатного доступа был использован аппарат математических решеток.

Введем следующие определения.

Определение 2.2.1. Решеткой уровней безопасности Λ_L называется формальная алгебра $\Lambda_L(L, \leq, \bullet, \otimes)$,

где L – базовое множество уровней безопасности;

$\leq$ – оператор, который определяет частичное нестрогое отношение порядка для элементов множества L , обладающее свойствами антисимметричности, транзитивности и рефлексивности²:

$$\forall l \in L: l \leq l \quad (\text{рефлексивность}),$$

$$\forall l_1, l_2 \in L: (l_1 \leq l_2 \wedge l_2 \leq l_1) \Rightarrow l_1 = l_2 \quad (\text{антисимметричность});$$

$$\forall l_1, l_2, l_3 \in L: (l_1 \leq l_2 \wedge l_2 \leq l_3) \Rightarrow l_1 \leq l_3 \quad (\text{транзитивность});$$

$\bullet$ – оператор, задающий для любой пары l_1, l_2 элементов множества L единственный элемент наименьшей верхней границы:

$$l_1 \bullet l_2 = l \Leftrightarrow l_1, l_2 \leq l \wedge \forall l' \in L: (l' \leq l) \Rightarrow (l' \leq l_1 \vee l' \leq l_2);$$

$\otimes$ – оператор, задающий для любой пары l_1, l_2 элементов множества L единственный элемент наибольшей нижней границы:

$$l_1 \otimes l_2 = l \Leftrightarrow l \leq l_1, l_2 \wedge \forall l' \in L: (l' \leq l_1 \wedge l' \leq l_2) \Rightarrow (l' \leq l).$$

Определение 2.2.2. Функцией уровня безопасности $\mathcal{F}_L: X \rightarrow L$ называется однозначное отображение множества сущностей КС $X = S \cup O$ во множество уровней безопасности L решетки Λ_L .

Обратное отображение $\mathcal{F}_L^{-1}: L \rightarrow X$ задает разделение всех сущностей КС на классы безопасности X_i такие, что:

$$X_1 \cup X_2 \cup \dots \cup X_N = X,$$

где N – мощность базового множества уровней безопасности L ;

$$X_i \cap X_j \equiv \emptyset, \quad \text{где } i \neq j;$$

$$\forall x' \in X_i \Rightarrow \mathcal{F}_L(x') = l_i, \quad \text{где } l_i \in L.$$

¹ Вместе с тем, заметим, несколько забегая вперед, что данное утверждение не совсем корректно. Доступ **write** в реальных КС фактически равнозначен доступу **read-write**, так как писать в документ, не "видя" его, невозможно, если только не применяется более специфичный метод доступа **append** – добавление в конец объекта.

² На практике множество уровней безопасности L является не просто частично упорядоченным, а линейно-упорядоченным, соответственно решетка Λ_L является линейной решеткой.

Покажем, что введенные решетка и функция уровней безопасности адекватно отражают парадигму градуированного доверия и критерий безопасности систем на ее основе.

1. Предположим, что информация может передаваться от сущностей класса X_i к сущностям класса X_j и наоборот, т. е. от сущностей класса X_j к сущностям класса X_i . Тогда для выполнения критерия безопасности сущности классов X_i и X_j должны образовывать один общий класс X_{ij} . Для того чтобы не создавать избыточных классов, необходимо, чтобы отношение, задаваемое оператором доминирования $\leq$, было антисимметричным.
2. Предположим, что информация может передаваться от сущностей класса X_i к сущностям класса X_j , и, кроме того, от сущностей класса X_j к сущностям класса X_k . Если каждая такая передача безопасна в отдельности, то, очевидно, безопасна и передача информации от сущностей класса X_i к сущностям класса X_k . Таким образом, отношение, задаваемое оператором $\leq$, должно быть транзитивным.
3. Внутри класса сущности имеют одинаковый уровень безопасности. Следовательно, передача информации между сущностями одного класса безопасна. Отсюда следует сравнимость по оператору $\leq$ сущностей одного класса между собой и самих с собой, т. е. рефлексивность отношения $\leq$.
4. Предположим, что имеется два различных класса сущностей X_i и X_j . Тогда, из соображений безопасности очевидно, что существует только один класс X' , потоки от сущностей которого безопасны по отношению к сущностям класса X_i или класса X_j , совпадающий с классом X_i или с классом X_j , и не имеется никакого другого класса X'' , менее безопасного чем X' , и с такими же возможностями по потокам к сущностям классов X_i и X_j . Это означает, что X' должен быть наименьшей верхней границей по уровням безопасности классов X_i и X_j .
5. Аналогично по условиям предыдущего пункта должен существовать ближайший снизу к классам X_i и X_j класс X' , такой, что потоки от сущностей классов X_i и X_j к сущностям класса X' безопасны, и совпадающий с классом X_i или с классом X_j , при этом не имеется никакого другого класса X'' , более безопасного, чем X' , и с такими же возможностями по потокам от сущностей классов X_i и X_j . Это означает, что X' должен быть наибольшей нижней границей по уровням безопасности классов X_i и X_j .

Таким образом, аппарат решетки и функция уровней безопасности действительно адекватно отражают сущность принципов и отношений политики мандатного разграничения доступом, на базе которых строятся конкретные модели, специфицирующие, в том числе, формализацию правил **NRU** и **NWD**, а также другие особенности мандатного доступа.

В заключение общей характеристики политики мандатного доступа отметим, что в ней разграничение доступа осуществляется до уровня классов безопасности сущностей системы. Иначе говоря, любой объект определенного уровня безопасности доступен любому субъекту соответствующего уровня безопасности (с учетом правил **NRU** и **NWD**). Нетрудно видеть, что мандатный подход к разграничению доступа, основываясь только лишь на идеологии градуированного доверия, без учета специфики других характеристик субъектов и объектов, приводит в большинстве случаев к *избыточности* прав доступа для конкретных субъектов в пределах соответствующих классов безопасности, что противоречит самому понятию разграничения доступа. Для устранения данного недостатка мандатный принцип разграничения доступа дополняется дискреционным внутри соответствующих классов безопасности. В теоретических моделях для этого вводят *матрицу доступа*, разграничивающую разрешенный по мандатному принципу доступ к объектам одного уровня безопасности.

2.2.2. Модель Белла-ЛаПадулы и ее расширения

Первой формальной моделью мандатного доступа является модель, разработанная еще в 1972–1975 г.г. американскими специалистами – сотрудниками MITRE Corporation Дэвидом Беллом и Леонардом ЛаПадулой (D.Elliott Bell, Leonard J.LaPadula), названная по их имени и сыгравшая огромную методологическую роль в развитии теории компьютерной безопасности.

Основные положения модели Белла-ЛаПадулы сводятся к следующему.

1. Модель системы $\Sigma(v_0, Q, Ft)$ представляется совокупностью:
 - множества объектов O доступа;
 - множества субъектов S доступа;
 - множества прав доступа R (в т. н. "классической" модели Белла-ЛаПадулы) всего два элемента – *read* и *write*);
 - матрицы доступа $A[s, o]$;
 - решетки Λ_L уровней безопасности L субъектов и объектов системы;
 - функции $F_L: S \cup O \rightarrow L$, отображающей элементы множеств S и O на множество L ;
 - множества состояний системы V , которое определяется множеством упорядоченных пар (F_L, A) ;
 - начального состояния $v_0 \in V$;
 - набора запросов Q субъектов на доступ (осуществление операций) к объектам, выполнение которых переводит систему в новое состояние;
 - функции переходов $Ft: (V \times Q) \rightarrow V^*$, которая переводит систему из одного состояния V в другое V^* при выполнении запросов из Q .
2. Состояния системы разделяются на опасные и безопасные. Для анализа и формулировки условий, обеспечивающих безопасность состояний системы, вводятся следующие определения:

Определение 2.2.3. Состояние называется безопасным по чтению (или просто безопасным) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ чтения к объекту, уровень безопасности этого субъекта доминирует над уровнем безопасности этого объекта:

$$\forall s \in S, \forall o \in O, read \in A[s, o] \rightarrow F_L(s) \geq F_L(o).$$

Определение 2.2.4. Состояние называется безопасным по записи (или *-безопасным¹) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ записи к объекту, уровень безопасности этого объекта доминирует над уровнем безопасности этого субъекта:

¹ В литературе можно встретить следующие обозначения: ss-безопасность – безопасность по чтению, *-безопасность – безопасность по записи, ds-безопасность – безопасность (разрешенность) по матрице доступа.

$$\forall s \in S, \forall o \in O, \text{write} \in A[s, o] \rightarrow F_L(o) \geq F_L(s).$$

Определение 2.2.5. Состояние системы безопасно тогда и только тогда, когда оно безопасно и по чтению, и по записи.

На основе введенных понятий, которые, как нетрудно видеть, выражают правила **NRU** и **NWD** политики мандатного доступа, авторы модели сформулировали следующий критерий безопасности.

Определение 2.2.6. (Критерий безопасности в модели Белла-ЛаПадулы). Система $\Sigma(v_0, Q, F_T)$ безопасна тогда и только тогда, когда ее начальное состояние v_0 безопасно и все состояния, достижимые из v_0 путем применения конечной последовательности запросов из Q , безопасны.

На основе данного критерия Белл и ЛаПадула доказали теорему, получившую название "основной теоремы безопасности" (**ОТБ**).

Теорема 2.2.1. (Basic Security Theorem). Система $\Sigma(v_0, Q, F_T)$ безопасна тогда и только тогда, когда:

1. Состояние v_0 безопасно, и
2. Функция переходов F_T такова, что для любого состояния v , достижимого из v_0 при выполнении конечной последовательности запросов из множества Q таких, что при $F_T(v, q) = v^*$, где $v = (F_L, A)$ и $v^* = (F_L^*, A^*)$, переходы системы из состояния в состояние подчиняются следующим ограничениям для $s \in S$ и для $o \in O$
 - если $\text{read} \in A^*[s, o]$ и $\text{read} \in A[s, o]$, то $F_L^*(s) \geq F_L^*(o)$;
 - если $\text{read} \in A[s, o]$ и $F_L^*(s) < F_L^*(o)$, то $\text{read} \notin A^*[s, o]$;
 - если $\text{write} \in A^*[s, o]$ и $\text{write} \notin A[s, o]$, то $F_L^*(s) \geq F_L^*(o)$;
 - если $\text{write} \in A[s, o]$ и $F_L^*(s) < F_L^*(o)$, то $\text{write} \notin A^*[s, o]$.

Доказательство.

Необходимость. Предположим, что система безопасна. Тогда состояние v_0 безопасно по определению. Предположим также, что существует некоторое состояние v , достижимое из состояния v_0 путем исполнения конечной последовательности запросов из Q , при которых $F_T(v, q) = v^*$, и при этом одно из первых двух ограничений (по чтению) не выполняется. Тогда, хотя состояние v^* и является достижимым, но небезопасно по определению 2.2.3. Если в состоянии v^* не выполняется одно из двух последних (по записи) ограничений, то в этом случае состояние v^* также будет небезопасным, но уже по определению 2.2.4. Таким образом, условия теоремы действительно необходимы для обеспечения безопасности системы.

Достаточность. Предположим, что система небезопасна. Тогда должно быть небезопасным либо состояние v_0 , либо состояние v , достижимое из v_0 путем исполнения конечной последовательности запросов из Q . Исходное состояние v_0 безопасно по условию теоремы. Тогда, так как v_0 безопасно, небезопасно какое-либо состояние v^* , переход в которое осуществляется из безопасного состояния v : $F_T(v, q) = v^*$. Однако это противо-

речит четырем ограничениям на переходы $\mathcal{F}_T$ по условиям теоремы. Следовательно, такой переход невозможен, и данное утверждение (о том, что система небезопасна) неверно. Таким образом, условия теоремы достаточны для безопасности системы $\Sigma(v_0, Q, \mathcal{F}_T)$. Теорема доказана. ■

Ввиду формального характера изложения и доказательства **ОТБ** несколько подробнее поясним ее суть. Четыре ограничения на функцию переходов фактически выражают требования соблюдения в каждом состоянии системы правил **NRU** и **NWD** *независимо от предыстории* доступов конкретных субъектов к конкретным объектам:

- если какой-либо субъект имеет разрешение на доступ к какому-либо объекту и в предыдущем состоянии и в новом (следующем) состоянии по матрице доступа, то и в новом состоянии это возможно только лишь при доминировании его уровня безопасности над уровнем объекта;
- если какой-либо субъект в предыдущем состоянии имел права чтения какого-либо объекта, а при переходе в новое состояние понизил свой уровень безопасности ниже уровня безопасности объекта, то право чтения по матрице доступа соответствующего субъекта к соответствующему объекту должно быть аннулировано;
- если как в предыдущем, так и в следующем состоянии какой-либо субъект по матрице доступа имеет право записи в какой-либо объект, то это возможно только лишь при условии, что и в новом состоянии уровень безопасности объекта будет доминировать над уровнем безопасности субъекта;
- если в предыдущем состоянии по матрице доступа какой-либо субъект имел право записи в какой-либо объект, а при переходе системы в новое состояние уровень объекта понизился до уровня ниже уровня субъекта, то право записи субъекта в данный объект по матрице доступа в новом состоянии должно быть аннулировано.

Иначе говоря, при переходах системы в новое состояние в матрице доступа не должно возникать никаких новых, и не должно сохраняться никаких старых отношений доступа, которые были бы небезопасны в смысле правил **NRU** и **NWD**.

Таким образом, ограничения, накладываемые на функцию перехода $\mathcal{F}_T$, по сути, совпадают с критериями безопасности и в этом отношении **ОТБ** избыточна.

Модель Белла-ЛаПадулы сыграла огромную роль в развитии теории компьютерной безопасности, и ее положения были введены в качестве обязательных требований к системам, обрабатывающим информацию, содержащую государственную тайну, в стандартах защищенных КС, в частности, в известной "Оранжевой книге" (1983 г.). Одной из сильных сторон модели Белла-ЛаПадулы является автоматическое решение проблемы "троянских" программ. Процесс переноса информации с помощью "троянской" программы из объекта, доступ к которому субъекту не разрешен по матрице доступа, регламентируется как и любые другие потоки прин-

ципами **NRU** и **NWD**. Таким образом, если такой перенос информации и произойдет, то он будет укладываться в ограничения *определений 2.2.3 и 2.2.4*, и, следовательно, с точки зрения критерия безопасности по *определению 2.2.6* система останется в безопасном состоянии.

2.2.3. Основные расширения модели Белла-ЛаПадулы

При практической реализации положений модели Белла-ЛаПадулы в реальных КС возник ряд трудностей, послуживших основанием для многочисленных работ по ее критическому анализу. Основные из этих трудностей связаны с проблемами переходных процессов, изменяющих доверительные характеристики (уровни безопасности) субъектов и объектов доступа, а также с невозможностью ограничиться в реальных КС процессами, сопровождающимися только лишь однонаправленными информационными потоками. В частности, один из наиболее известных исследователей модели Белла-ЛаПадулы, МакЛин привел концептуальное описание **Z**-системы, удовлетворяющей условиям **ОТБ**, но, вместе с тем, обеспечивающей возможность получения доступа любым субъектом к любому объекту по любому методу (*read* и *write*). В **Z**-системе субъекты и объекты в промежутках переходов системы из одного состояния в другое имеют возможности изменять свой уровень безопасности, при этом ни одно из ограничений **ОТБ** не нарушается. Тем не менее, последовательно снижая классификацию (уровень) объекта до своего уровня, или вовсе его деклассифицируя, субъект может получить доступ по чтению к нужному объекту. Также, последовательно понижая свой уровень безопасности, субъект может получить доступ к нужному объекту по записи.

Причина проблемы **Z**-системы, как указал МакЛин, в том, что модель Белла-ЛаПадулы никаким образом не специфицирует и не регламентирует процессы назначения и изменения уровня безопасности сущностей системы, т. е. является чрезмерно абстрактной по отношению к реальным процессам в КС.

Проведя обстоятельные и глубокие исследования классической модели Белла-ЛаПадулы, МакЛин построил несколько *расширений* модели Белла-ЛаПадулы, преодолевающих некоторые недостатки или приближающих к "реальности" исходную каноническую модель. Первое из этих расширений связано с исследованием условий и ограничений, накладываемых на функцию перехода (модель с *безопасной функцией перехода*).

Для рассмотрения условий безопасности функции перехода МакЛин разделил общую функцию уровней безопасности $\mathcal{F}_L: S \cup O \rightarrow L$ на функцию уровней безопасности субъектов $\mathcal{F}_{L_s}: S \rightarrow L$ и функцию уровней безопасности объектов $\mathcal{F}_{L_o}: O \rightarrow L$.

В модели с безопасной функцией переходов на этой основе вводятся следующие определения.

Определение 2.2.7. Функция перехода $F_T: (V \times Q) \rightarrow V$ является безопасной по чтению, если для любого перехода $F_T(v, q) = v^*$ выполняются следующие три условия:

- если $read \in A^*[s, o]$ и $read \notin A[s, o]$, то $F_{L_s}(s) \geq F_{L_o}(o)$ и $F_L = F_L^*$;
- если $F_{L_s} \neq F_{L_s}^*$, то $A = A^*$, $F_{L_o} = F_{L_o}^*$, для $\forall s$ и o , у которых $F_{L_s}^*(s) < F_{L_o}^*(o)$, $read \notin A[s, o]$;
- если $F_{L_o} \neq F_{L_o}^*$, то $A = A^*$, $F_{L_s} = F_{L_s}^*$, для $\forall s$ и o , у которых $F_{L_s}^*(s) < F_{L_o}^*(o)$, $read \notin A[s, o]$.

Определение 2.2.8. Функция перехода $F_T: (V \times Q) \rightarrow V$ является безопасной по записи, если для любого перехода $F_T(v, q) = v^*$ выполняются следующие три условия:

- если $write \in A^*[s, o]$ и $write \notin A[s, o]$, то $F_{L_s}(s) \geq F_{L_o}(o)$ и $F_L = F_L^*$;
- если $F_{L_s} \neq F_{L_s}^*$, то $A = A^*$, $F_{L_o} = F_{L_o}^*$, для $\forall s$ и o , у которых $F_{L_s}^*(s) > F_{L_o}^*(o)$, $write \notin A[s, o]$;
- если $F_{L_o} \neq F_{L_o}^*$, то $A = A^*$, $F_{L_s} = F_{L_s}^*$, для $\forall s$ и o , у которых $F_{L_s}^*(s) > F_{L_o}^*(o)$, $write \notin A[s, o]$.

Определение 2.2.9 (Критерий безопасности МакЛина). Функция перехода $F_T: (V \times Q) \rightarrow V$ является безопасной тогда и только тогда, когда она безопасна и по чтению, и по записи.

Смысл ограничений на безопасность функции перехода в определениях 2.2.8 и 2.2.9 в неформальном выражении заключается в том, что в процессе перехода может меняться, во-первых, только лишь один из компонентов состояния системы безопасности (или отношение доступа определенного субъекта к определенному объекту, т. е. ячейка матрицы доступа, или решетка уровней безопасности субъектов, или решетка уровней безопасности объектов), а, во-вторых, при условии того, что правила **NRU** и **NWD** соблюдаются как в предыдущем, так и в новом состоянии.

Основываясь на введенных определениях, МакЛин сформулировал и доказал следующую теорему.

Теорема 2.2.2. (Теорема безопасности МакЛина). Система $\Sigma(v_0, Q, F_T)$ безопасна в любом состоянии и в процессе переходов между ними, если ее начальное состояние v_0 безопасно, а ее функция перехода удовлетворяет критерию безопасности МакЛина.

Условия безопасности переходов МакЛина снимают проблему **Z**-системы, но, тем не менее, не решают всех проблем практической реализации КС на основе модели Белла-ЛаПадулы. В частности, введенные условия по определениям 2.2.7 и 2.2.8 накладывают ограничения, но не специфицируют сам процесс изменения уровней безопасности сущностей системы. Иначе говоря, вопросы о том, кто, при каких условиях имеет право изменять уровни безопасности сущностей системы (т. е. инициировать соответствующие переходы), по-прежнему остаются неопределенными.

МакЛин для решения данной проблемы расширил модель Белла-ЛаПадулы введением *уполномоченных субъектов*.

Определение 2.2.10. *Подмножество субъектов $\mathcal{P}(\mathcal{S})$, которым разрешается инициировать переходы системы, изменяющие уровни безопасности субъектов или объектов, называется уполномоченными (доверенными) субъектами.*

На этой основе в модели вводится дополнительная функция управления уровнями безопасности.

Определение 2.2.11. *Функцией управления уровнями безопасности $\mathcal{F}_\mathcal{C}: \mathcal{S} \cup \mathcal{O} \rightarrow \mathcal{P}(\mathcal{S})$ называется отображение множества сущностей КС (субъектов и объектов) на подмножество уполномоченных субъектов.*

Заметим, что в общем виде функция $\mathcal{F}_\mathcal{C}$ задается не однозначным отображением¹ и определяет тем самым в системе с уполномоченными субъектами для каждого объекта или субъекта одного или нескольких субъектов, которым разрешено изменять уровни безопасности соответствующих сущностей.

Реализация системы с уполномоченными субъектами требует изменения также и функции перехода системы, а именно, ее *авторизации*.

Определение 2.2.12. *Функция перехода $\mathcal{F}_\mathcal{T}^\mathcal{A}: (\mathcal{S} \times \mathcal{V} \times \mathcal{Q}) \rightarrow \mathcal{V}$ в системе с уполномоченными субъектами $\Sigma(\mathbf{v}_0, \mathcal{Q}, \mathcal{F}_\mathcal{T}^\mathcal{A})$ называется авторизованной тогда и только тогда, когда для каждого перехода $\mathbf{t}^\mathcal{A}(\mathbf{s}, \mathbf{v}, \mathbf{q}) = \mathbf{v}^*$, при котором $\mathbf{v} = (\mathcal{F}_\mathcal{T}, \mathbf{A})$ и $\mathbf{v}^* = (\mathcal{F}_\mathcal{T}^*, \mathbf{A}^*)$, выполняется следующее условие:*

$$\forall \mathbf{x} \in \mathcal{S} \cup \mathcal{O} : \text{если } \mathcal{F}_\mathcal{L}^*(\mathbf{x}) \neq \mathcal{F}_\mathcal{L}(\mathbf{x}), \text{ то } \mathbf{x} \in \mathcal{P}(\mathcal{S}) .$$

Аппарат уполномоченных субъектов на основе определений 2.2.10, 2.2.11 и 2.2.12 разрешает неопределенность процессов изменения уровней безопасности сущностей системы в абстрактной модели Белла-ЛаПадулы и позволяет разработчикам специфицировать и строить реальные КС с мандатным принципом разграничения доступа.

Вместе с тем, введение уполномоченных субъектов не обеспечивает автоматически всех условий и ограничений безопасности, а только лишь специфицирует процесс изменения уровней безопасности сущностей системы. Следовательно, вопросы безопасности функционирования системы с уполномоченными субъектами должны в достаточном отношении рассматриваться с точки зрения дополнительных условий, определяемых ОТБ и теоремой МакЛина (по безопасной функции перехода).

Более того, внимательный анализ идеологии уполномоченных субъектов показывает наличие внесения в процессы безопасности субъективного фактора. Доверенные субъекты, изменяя уровни безопасности субъектов и объектов системы, должны действовать корректно, к примеру, снижать

¹ т. е., строго говоря, является не функцией, а именно отображением.

гриф секретности объекта, когда из него удалена вся информация более высокого уровня и т. п.

Очевидно, как некая альтернатива уполномоченным субъектам, избавляющая процессы безопасности от субъективного фактора, была предложена мандатная *модель Low-Watermark (LWM)*.

Неформальное выражение существа модели **LWM** сводится к следующему. Если по определенным соображениям субъектам нельзя отказывать в доступе *write* к любым объектам системы, то для исключения опасных информационных потоков "сверху вниз", т. е. от сущностей с высоким уровнем безопасности к сущностям с более низким уровнем безопасности, необходимо дополнить определенными правилами операцию *write* и ввести дополнительную операцию *reset*.

Если субъекту требуется внести информацию в объект с более низким, чем у субъекта, уровнем безопасности (что запрещено правилом **NWD**), то субъект может подать команду *reset*, в результате которой уровень безопасности объекта автоматически повышается до максимального уровня безопасности в системе. В результате операции *reset* объект согласно правилу **NWD** становится доступным по записи (в том числе и для всех других субъектов системы). При этом, однако, опасности перетекания информации "сверху вниз" не создается, так как по чтению система как прежде, так и в дальнейшем руководствуется безопасным правилом **NRU**.

Нетрудно увидеть, что данный порядок порождает тенденцию деградации (огрубления) системы разграничения доступа, при которой все объекты через какое-то время могут получить высшие грифы секретности и, соответственно, стать недоступными по чтению для всех субъектов, не обладающих высшим уровнем доверия. Поэтому в модели **LWM** операция *write* дополняется следующим условием. Если в результате операции *write* в объект реально вносится информация и уровень безопасности объекта строго выше уровня безопасности субъекта, то перед внесением в объект информации вся прежняя информация из него удаляется (стирается), а по окончании операции записи уровень безопасности объекта автоматически понижается до уровня безопасности субъекта.

В модели **LWM** доказываемая, что введение операции *reset* и видоизмененной операции *write* оставляет систему безопасной в смысле критерия безопасности Белла-ЛаПадулы.

От себя заметим, что более логичным, но в определенном смысле зеркальным по отношению к модели **LWM**, мог бы быть подход, при котором субъектам разрешается производить запись в объекты с более низким уровнем безопасности при условии того, что при завершении операции уровень безопасности объекта автоматически повышается до уровня безопасности субъекта. При этом возможность опасных потоков "сверху вниз" также как и в модели **LWM** нейтрализуется.

Еще одним расширением модели Белла-ЛаПадулы, имеющим важное прикладное значение, является введение методологии и техники **совместного (группового) доступа**.

Как уже отмечалось, политика и, соответственно, модели мандатного доступа основываются на принципах, заимствованных из правил и системы секретного делопроизводства, принятых во многих странах, в частности в США. Одним из таких правил, является то, что некоторые наиболее критичные с точки зрения безопасности процессы, например, изменение уровня конфиденциальности документов, осуществляются путем совместных действий (одновременных или последовательных) нескольких работников – исполнителя (владельца) документа и руководителя (администратора).

Подобные приемы усиления безопасности применяются и в других сферах, например в бухгалтерии, где критичные операции приема/выдачи финансовых средств осуществляются совместными действиями кассира и контроллера, каждый из которых подписывает или заверяет соответствующий платежный документ. Еще одним из подобных правил является порядок доступа к индивидуальным ячейкам-хранилищам клиентов банков, когда для их открытия требуется одновременно два ключа – один от клиента, другой от служащего (администратора) банка.

Для реализации технологии совместного доступа в мандатной модели добавляется функция группового доступа.

Определение 2.2.13. Функцией группового доступа $F_G: S \rightarrow \mathcal{P}(S) \setminus \emptyset$ называется отображение множества субъектов KS S на множество непустых подмножеств субъектов $S_G = \mathcal{P}(S) \setminus \emptyset$, каждое из которых образует группой субъект доступа $s_G \in S_G$.

На этой основе в матрице доступа $A[s, o]$ системы добавляются строки, соответствующие субъектам группового доступа.

Помимо дополнения матрицы доступа системы групповыми субъектами, требуется также разработка механизмов реализации для них правил мандатного доступа, т. е. правил **NRU** и **NWD**, имея ввиду то, что в состав подмножества субъектов, образующих субъект группового доступа, могут входить субъекты с различным уровнем безопасности. С этой целью вводятся дополнительные функции уровней безопасности субъектов группового доступа.

Определение 2.2.14. Функцией уровня безопасности $F_L^L: S_G \rightarrow L$ называется однозначное отображение множества субъектов группового множества S_G во множество уровней безопасности L решетки Λ такое, что $F_L^L(s_G)$ является наибольшей нижней границей уровней безопасности для множества субъектов s , входящих в s_G .

Определение 2.2.15. Функцией уровня безопасности $F_L^H: S_G \rightarrow L$ называется однозначное отображение множества субъектов группового множества S_G во множество уровней безопасности L решетки Λ

такое, что $\mathcal{F}_L^H(s_G)$ является наименьшей верхней границей уровней безопасности для множества субъектов s , входящих в s_G .

Введение функций $\mathcal{F}_L^L$ и $\mathcal{F}_L^H$ дает возможность переопределить критерий безопасности (определения 2.2.4, 2.2.5 и 2.2.6) исходной модели мандатного доступа Белла-ЛаПадулы.

Определение 2.2.16. *Состояние системы называется безопасным по чтению тогда и только тогда, когда для каждого индивидуального или группового субъекта, осуществляющего в этом состоянии доступ чтения к объекту, уровень безопасности, задаваемый функцией $\mathcal{F}_L$ для индивидуального субъекта или функцией $\mathcal{F}_L^L$ для группового субъекта, доминирует над уровнем безопасности объекта.*

Определение 2.2.17. *Состояние системы называется безопасным по записи тогда и только тогда, когда для каждого индивидуального или группового субъекта, осуществляющего в этом состоянии доступ записи к объекту, уровень безопасности объекта доминирует над уровнем безопасности субъекта, задаваемого функцией $\mathcal{F}_L$ для индивидуального субъекта или функцией $\mathcal{F}_L^H$ для группового субъекта.*

Иными словами, доступ по чтению для групповых субъектов безопасен тогда, когда самый низший уровень безопасности из множества индивидуальных субъектов, образующих групповой субъект, выше или равен уровню безопасности объекта чтения. Доступ по записи для групповых субъектов безопасен тогда, когда самый высокий уровень безопасности из всех индивидуальных субъектов, входящих в групповой субъект, ниже или равен уровню безопасности объекта записи.

Остальные условия безопасного функционирования системы с субъектами группового доступа задаются условиями соответствующих теорем (ОТБ или теоремы МакЛина).

Таким образом, расширения модели Белла-ЛаПадулы обеспечивают устранение многих недостатков исходной модели, и, кроме того, могут комбинироваться для обеспечения более реализуемых на практике спецификаций политики мандатного доступа в реальных КС.

Тем не менее, расширения модели Белла-ЛаПадулы не снимают всех недостатков мандатного доступа. В частности, как уже указывалось, мандатный доступ снимает проблему "троянских программ", но только с точки зрения опасных потоков "сверху вниз". Однако в пределах одного класса безопасности, вопросы доступа решаются, как и в дискреционных моделях, на основе матрицы доступа, и, следовательно, для полного устранения проблемы "троянских программ" и в системах мандатного доступа требуется более тщательный и детализированный контроль информационных потоков, в том числе механизмов ИПС.

2.3. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ ТЕМАТИЧЕСКОЙ ПОЛИТИКИ

2.3.1. Общая характеристика тематического разграничения доступа

Как уже не раз отмечалось многие механизмы и принципы разграничения доступа к информации "подсмотрены" во внекомпьютерной сфере, что обеспечило формирование двух базовых политик безопасности – дискреционного и мандатного доступа. Вместе с тем, данные политики безопасности отражают не все организационно-технологические схемы защиты конфиденциальной информации, исторически наработанные и апробированные в "бумажных" сферах.

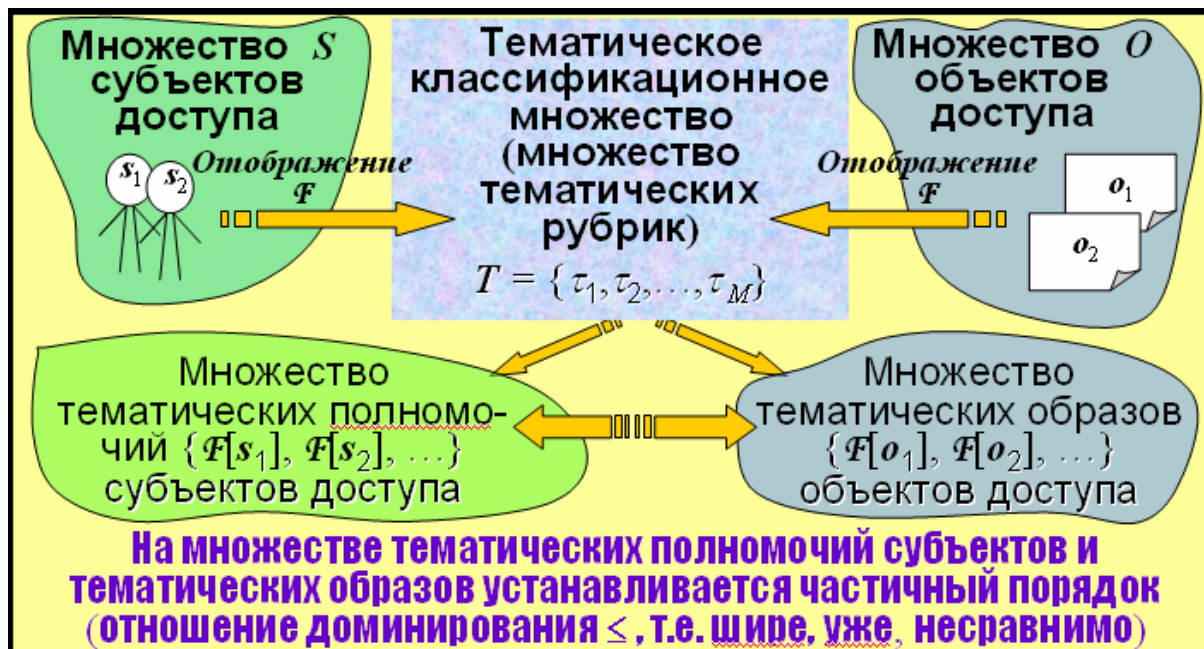
Важным аспектом, присутствующим в практике разграничения доступа к "бумажным" ресурсам является тематическая "окрашенность" информационных ресурсов предприятий, учреждений по организационно-технологическим процессам и профилям деятельности¹. Организация доступа сотрудников к информационным ресурсам организации (в библиотеках, архивах, документальных хранилищах) осуществляется на основе *тематических классификаторов*. Все документы информационного хранилища тематически индексируются, т. е. соотносятся с теми или иными тематическими рубриками классификатора. Сотрудники предприятия согласно своим функциональным обязанностям или по другим основаниям получают права работы с документами определенной тематики. Данный подход, в сочетании с избирательным и мандатным доступом, обеспечивает более адекватную и гибкую настройку системы разграничения доступа на конкретные функционально-технологические процессы, предоставляет дополнительные средства контроля и управления доступом.

Помимо уже указанной MLS-решетки, попытки отразить тематический аспект в организации разграничения доступа, имелись еще в ранних работах по моделям дискреционного доступа, в частности, в одной из первых дискреционных моделей АДЕПТ-50, в модели на основе типизованной матрицы доступа, в расширениях базовой модели безопасности ОС UNIX – модели DTE. Вместе с тем, использование тематических категорий субъектов и объектов доступа в данных моделях являлось второстепенным фактором, служащим для уточнения или корректировки некоторых аспектов базовой политики безопасности. Кроме того, назначение субъектам и объектам доступа единичных категорий из простого неупорядоченного списка тематик обеспечивает лишь наиболее простой и грубый способ тематического разграничения доступа к информации.

¹ Документы "кадровые", "бухгалтерские", "конструкторско-технологические" и т. д. Пример из другой сферы – информационные ресурсы по артиллерии, по авиации, по флоту и т. д.

В общем плане принцип тематической политики безопасности можно пояснить схемой, приведенной на рис. 2.16.

Рис. 2.16. Общий принцип тематической политики безопасности



Анализ библиотечных и других автоматизированных систем документального поиска, основанных на тематическом индексировании содержания документов (текстов), показывает, что определяющее значение в таких системах имеет тематико-классификационная схема, в большинстве случаев именуемая *тематическим рубрикатом*. Применяются три основных способа тематической классификации:

- перечислительная классификация (дескрипторный подход);
- систематизированная классификация (иерархический подход);
- аналитико-синтетическая классификация (фасетный подход).

При дескрипторном подходе классификационная схема представляет неупорядоченный¹ перечень тематик (рубрик, предметов, ключевых слов и т. д.), произвольной совокупностью которых можно отразить содержание (тематику) конкретного документа – см. рис. 2/17.

Сотрудники предприятия получают права работы с документами определенного набора тематических рубрик, т.е. фактически так же, как и документальные ресурсы тематически классифицируются.

¹ Неупорядоченный в смысле отсутствия каких-либо отношений (старшинства, ассоциативности и т. п.) между рубриками.

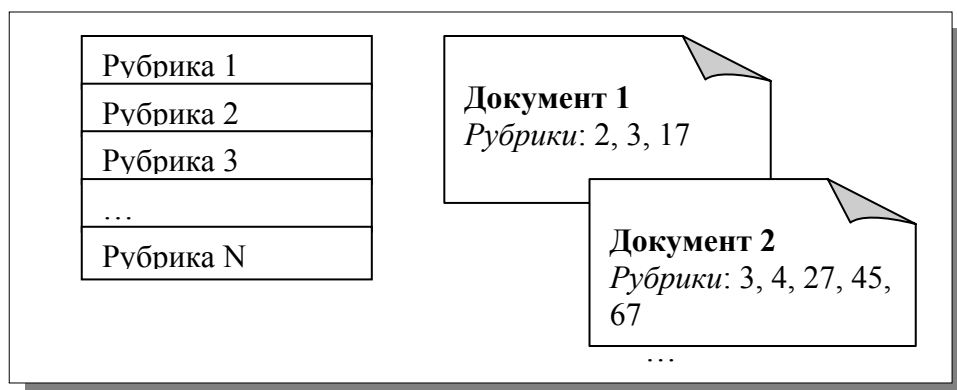


Рис. 2.17. Дескрипторная классификация

При иерархическом подходе тематико-классификационная схема представляет корневую древовидную структуру, основанную на таксономическом принципе ("род-вид"). Тематика документа определяется тематическими узлами классификатора с автоматическим распространением на объект классификации всех соответствующих подчиненных тематических узлов – см. рис. 2.18. Сотрудникам предприятия предоставляется право работы с документами, тематика которых соответствует одному или нескольким узлам классификатора с автоматическим распространением прав работы на документы с тематикой соответствующих подчиненных узлов.

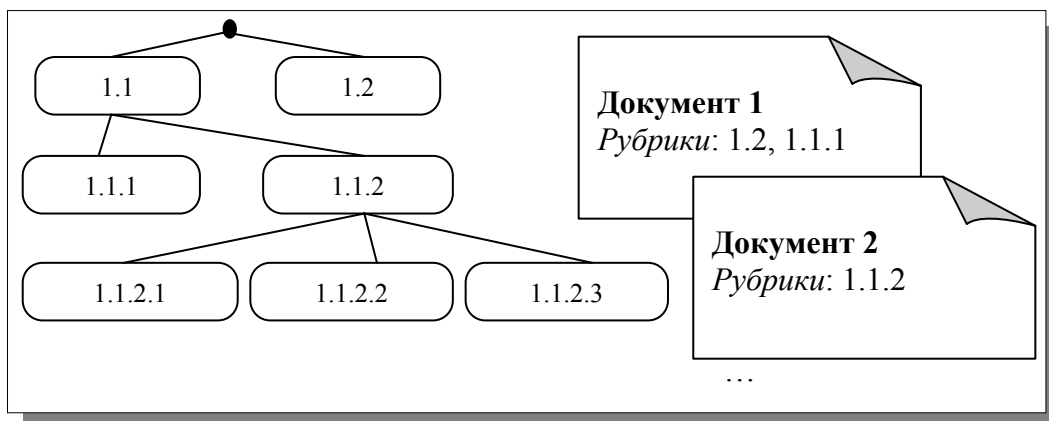


Рис. 2.18. Иерархическая классификация

Основу фасетной классификации составляет определенное количество тематических блоков-фасет¹, отражающих логику соответствующей предметной сферы (например, "материал"- "конструкция"- "нагрузки" или "время"- "место"- "состав"- "действие" и т. д.). Каждый фасет, в свою очередь, представляет классификационную подсистему тематик (рубрик) иерархического типа. Тематика конкретного информационного объекта строится на основе сочетания тематических узлов фасетных рубрик – см.

¹ Фасет – в буквальном смысле – аспект, грань.

рис. 2.19. Права доступа сотрудников к документам определяются также сочетанием определенных рубрик по фасетам классификатора.

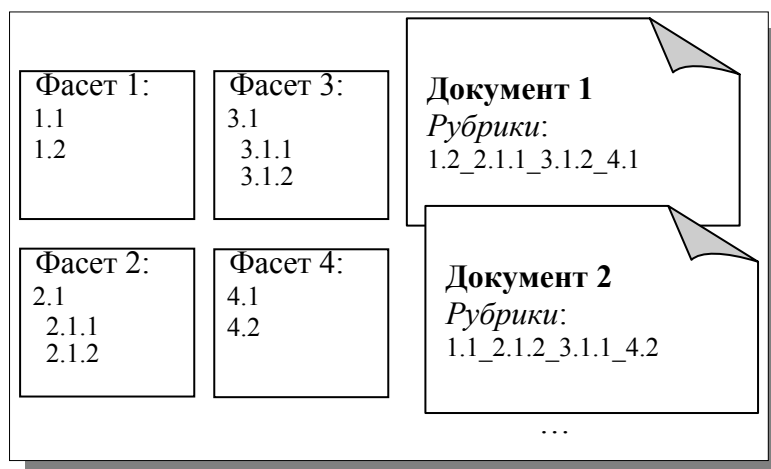


Рис. 2.19. Фасетная классификация

Общим правилом (критерием) предоставления доступа пользователям к ресурсам (объектам) по всем способам классификации является следующее:

Определение 2.3.1 (Общее правило тематического доступа). *Документ доступен сотруднику, если он не содержит рубрик, запрещенных (не разрешенных) для данного сотрудника.*

В документальных хранилищах, библиотеках и архивах ввиду их очевидного предназначения под доступом понимается, прежде всего, чтение документов. Иначе говоря, в данных системах присутствует перенос информации (информационные потоки) только из документов к сотрудникам¹.

В системах, где документы не только изучаются, используются, но и создаются, корректируются (например, системы делопроизводства и документооборота), возможны как потоки из документов к сотрудникам, так от сотрудников в документы. Однако, совершенно ясно, что в "бумажном контуре" при создании и, особенно, при изменении документов вносить или изменять информацию в документ, не "видя" все остальное содержимое документа невозможно. Поэтому, в отличие от правила **NWD** мандатного доступа, кстати сказать, огрубленного и идеалистически абстрактного для реализации в "бумажном контуре", в системах делопроизводства в большинстве случаев используются следующие правила:

¹ Отметим, что данная картина полностью вписывается в понятие потока, введенного в п. 1.1.3, имея в виду то, что сотрудник, читая (изучая) документ, вносит (конспектирует) его информацию в свою рабочую тетрадь, которая в данном контексте является информационным объектом, ассоциированным с сотрудником.

- тематика создаваемого (редактируемого) документа назначается (уточняется) исполнителем документа (автором, ответственным и т. д.) в пределах тематики данного исполнителя¹;
- создание или изменение документов может происходить на основе объединения (слияния) в один общий документ совокупности других документов, при этом тематика общего документа формируется как объединенная тематика совокупности вошедших в него документов, а процедуру объединения (слияния) осуществляет сотрудник, допущенной ко всей объединенной тематике.

Отмеченные правила преследуют цель обеспечения конфиденциальности (безопасности) информации в системе, что можно сформулировать следующим общим критерием безопасности тематического разграничения доступа.

Определение 2.3.2 (Критерий безопасности тематического доступа). Система безопасна, если пользователи ни при каких обстоятельствах не могут получить доступ к информации "не своей" тематики.

Система и порядок тематического разграничения доступа регламентируются нормативными документами предприятий, организаций, ведомств, и совершенно очевидно, что, автоматизируя документальные информационные системы, подобные регламентации необходимо воспроизводить в виде формализованных процедур, схем представления и манипулирования с информацией в КС, т. е. в виде моделей тематического разграничения доступа.

2.3.2. Тематические решетки

Как и в моделях мандатного доступа, контроль безопасности информационных потоков при тематическом доступе базируется на использовании аппарата решеточно упорядоченных множеств.

Результаты дескрипторной тематической классификации сущностей системы (тематики $\mathcal{F}_d[x_i]$) образуют тематическую решетку, являющуюся решеткой подмножеств $\Lambda_d(\mathcal{P}_d, \subseteq, \cup, \cap)$ множества T_d ($\mathcal{P}_d$ – множество подмножеств множества T_d).

Частичный порядок на множестве $\mathcal{P}_d$ задается отношением теоретико-множественного включения $\subseteq$, определяющего доминирование тематической классификации сущности x_i над тематической классификацией сущности x_j в том случае, когда $\mathcal{F}_d[x_j] \subseteq \mathcal{F}_d[x_i]$. При этом отношение, задаваемое операцией включения, обладает свойствами рефлексивности, антисимметричности и транзитивности. Наименьшая верхняя и наибольшая нижняя граница для любой пары тематических классификаций сущностей

¹ Автор-исполнитель документа, например, устанавливает набор ключевых слов, выражающих тематику содержимого созданного (измененного) документа.

системы $\mathcal{F}_d[x_i]$ и $\mathcal{F}_d[x_j]$ задаются операциями теоретико-множественного объединения $\mathcal{F}_d[x_i] \cup \mathcal{F}_d[x_j]$ и пересечения $\mathcal{F}_d[x_i] \cap \mathcal{F}_d[x_j]$, соответственно.

При иерархической тематической классификации применяются *монорубрицированный* и *мультирубрицированный* подходы, соответствующие двум различным направлениям в архитектуре и механизмах функционирования информационно-поисковых документальных систем.

Определение 2.3.8. *Монорубрицированной иерархической тематической классификацией $\mathcal{F}_{и1}[x]$ называется отображение множества сущностей системы $X = S \cup O$ на множество тематических рубрик иерархического классификатора $T_{и}$ такое, что любой сущности системы $x \in X$ соответствует определенная и единственная тематическая рубрика $\tau \in T_{и}$ совместно со всеми подчиненными узлу τ рубриками.*

Таким образом, в системах с монорубрицированным отображением каждому объекту o соответствует определенная и единственная тематическая рубрика τ (узел корневого дерева) с автоматическим включением в характеристику информационного содержания объекта o тематики всех узлов рубрикатора, подчиненных узлу τ . Данный подход характеризует инфологическую сущность информационно-поисковых каталогов, в которых объекты доступа (документы) технологически размещаются в узлах каталога. При этом структура каталога собственно и выражает тематический рубрикатор.

Определение 2.3.9. *Мультирубрицированной иерархической тематической классификацией $\mathcal{F}_{и2}[x]$ называется отображение множества сущностей системы $X = S \cup O$ на множество тематических рубрик иерархического классификатора $T_{и}$ такое, что любой сущности системы $x \in X$ соответствует набор тематических рубрик $\{\tau_{x1}, \tau_{x2}, \dots\}$, причем рубрики $\tau_{x1}, \tau_{x2}, \dots$ не находятся между собой в подчинении и никакая их совокупность не образует полный набор сыновей какого-либо узла классификатора.*

Первое условие в определении мультирубрицированной классификации обусловлено тем, что если сущность КС отображается на определенный узел корневого дерева, то по смыслу иерархически организованного множества $T_{и}$ данное отображение включает и все подчиненные по дереву соответствующие узлы. Отсюда явно включать в классификацию узлы, находящиеся между собой в иерархическом подчинении, не имеет никакого смысла.

Второе условие обусловлено тем, что если определенная совокупность узлов составляет полный набор узлов, являющихся непосредственными потомками какого-либо узла дерева, то по таксономическому свойству иерархического рубрикатора это означает, что в данной ситуации вместо подобной совокупности узлов достаточно использовать узел-родитель.

Таким образом в системах с мультирубрицированным отображением каждому объекту o может соответствовать не одна, а некоторая совокупность тематических рубрик $\{\tau_{o1}, \tau_{o2}, \dots\}$, т. е. узлов корневого дерева, при условии того, что данные узлы не находятся в отношении друг друга в иерархическом подчинении и никакая их совокупность не образует полный набор сыновей какого-либо иерархического узла. При этом в характеристику информационного содержания объекта o включается также тематика всех узлов рубрикатора, подчиненных узлам $\tau_{o1}, \tau_{o2}, \dots$. Подобный подход характерен для индексных информационно-поисковых систем, строящихся на основе иерархических рубрикаторов, в частности, для информационно-поисковых систем, основанных на тезаурусах.

Тематические разрешения пользователям в системах и первого и второго вида, иначе говоря, отображения субъектов доступа на множество вершин иерархического рубрикатора, в большинстве случаев производится мультирубрицированным образом.

Тематическая решетка при монорубрицированном отображении на множестве $T_{\mathbf{n}} = \{\tau_1, \tau_2, \dots, \tau_M\}$ корневое дерево иерархического рубрикатора задает отношение частичного порядка $\leq$, обладающее свойствами рефлексивности, антисимметричности и транзитивности.

Для построения решетки на множестве $T_{\mathbf{n}}$ помимо отношения частичного порядка необходимо задать операции (механизмы), определяющие для любой пары рубрик (τ_i, τ_j) наименьшую верхнюю и наибольшую нижнюю границы.

Дополняя множество $T_{\mathbf{n}}$ пустой рубрикой (элементом τ_0) и замыкая на него все концевые (листовые) вершины (см. рис. 2.20), получаем решеточно упорядоченное множество, в котором для любой пары элементов (τ_i, τ_j) имеется наименьшая верхняя $\sup\{\tau_i, \tau_j\}$ и наибольшая нижняя граница $\inf\{\tau_i, \tau_j\}$.

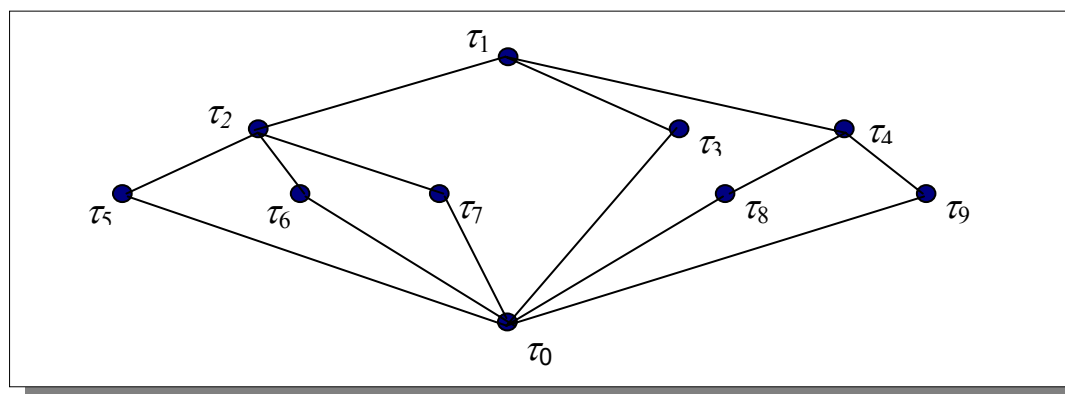


Рис. 2.20. Переход к решеточно упорядоченному множеству путем добавления пустого элемента

Определение 2.3.10. Наименьшей общей верхней границей $\sup_{\mathbf{I}}\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}\} = \tau_i$ для набора рубрик $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ будем называть вершину τ_i , являющуюся по корневому дереву иерархического рубрикатора наименьшим общим предком вершин $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$.

Определение 2.3.11. Наибольшей общей нижней границей $\inf_{\mathbf{I}}\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}\} = \tau_i$ для набора рубрик $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ будем называть вершину τ_i , являющуюся в графе, образованном корневым деревом иерархического рубрикатора с замыканием всех его листовых вершин на пустую вершину, наибольшей общей подчиненной¹ рубрик $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$.

При этом из определения 2.3.11 следует, что если вершины $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ попарно сравнимы, т. е. находятся на разных уровнях одной ветви дерева, то наибольшей нижней границей $\inf_{\mathbf{I}}$ является вершина, находящаяся на самом нижнем уровне соответствующего участка данной ветви дерева, т. к. является ближайшей, подчиненной снизу всем, включая саму себя из набора $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}\}$. При несравнимости (неподчиненности друг другу) хотя бы в одной паре рубрик произвольного набора $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ наибольшим общим потомком $\inf_{\mathbf{I}}$ является пустая вершина τ_0 , которую можно трактовать как отсутствие в объекте классификации информации, описываемой иерархическим рубрикатором.

В результате на множестве $T_{\mathbf{I}\emptyset} = T_{\mathbf{I}} \cup \tau_0$ имеем решетку $\Lambda_{\mathbf{I}}(T_{\mathbf{I}\emptyset}, \leq, \sup_{\mathbf{I}}, \inf_{\mathbf{I}})$, основывая на которой монорубрицированную тематическую классификацию сущностей системы можно построить модель тематического разграничения доступа с полным контролем информационных потоков.

Использование отношения $\leq$ и операций $\sup_{\mathbf{I}}, \inf_{\mathbf{I}}$ требует в практических системах переборно-циклических алгоритмов прохождения иерархического рубрикатора, что может вызывать повышенные вычислительные затраты. Альтернативой в подобных случаях может стать использование листовой тематической решетки, изоморфной решетке $\Lambda_{\mathbf{I}}(T_{\mathbf{I}\emptyset}, \leq, \sup_{\mathbf{I}}, \inf_{\mathbf{I}})$.

Введем следующее понятие.

Определение 2.3.12. Листовым тематическим множеством $T^{\mathbf{L}}(\tau_i)$ вершины τ_i называется совокупность всех листовых вершин $\tau^{\mathbf{L}}_j$, подчиненных данной вершине тематического классификатора $T_{\mathbf{I}}$.

¹ Включая подчиненность вершины самой себе.

Отметим, что для листовой вершины τ_j ее листовое тематическое множество $\mathcal{T}^{\text{Л}}(\tau_j)$ представляется одноэлементным множеством, единственный элемент которого совпадает с самой листовой вершиной.

На множестве листовых подмножеств $\mathcal{T}^{\text{Л}}$ как и на множестве $\mathcal{P}$ подмножеств любого множества отношение включения $\subseteq$ задает частичный порядок.

Справедливо следующее утверждение.

Лемма 2.3.1. *Частично упорядоченное относительно отношения включения $\subseteq$ множество листовых тематических множеств $\{\mathcal{T}^{\text{Л}}(\tau_1), \mathcal{T}^{\text{Л}}(\tau_2), \dots, \mathcal{T}^{\text{Л}}(\tau_M)\}$ вершин иерархического рубрикатора изоморфно частично упорядоченному множеству тематических рубрик $\{\tau_1, \tau_2, \dots, \tau_M\}$, задаваемому иерархическим рубрикатором.*

Доказательство. Поскольку каждая внутренняя вершина классификационного дерева имеет не менее двух сыновей, при условии $\tau_i \neq \tau_j$ множества $\mathcal{T}^{\text{Л}}(\tau_i)$ и $\mathcal{T}^{\text{Л}}(\tau_j)$ отличаются друг от друга хотя бы одной листовой вершиной, т. е. также не совпадают. Отсюда следует, что отображение $\tau_i \rightarrow \mathcal{T}^{\text{Л}}(\tau_i)$ биективно.

Покажем далее, что отношение $\leq$ и отношение несравнимости любых двух произвольных тематических рубрик τ_i и τ_j сохраняются (адекватно воспроизводятся) на их образах, т. е. на множествах $\mathcal{T}^{\text{Л}}(\tau_i)$ и $\mathcal{T}^{\text{Л}}(\tau_j)$.

Пусть для вершин τ_i и τ_j выполняется $\tau_i \leq \tau_j$ (τ_i подчинена τ_j). С учетом подчиненности вершины τ_i вершине τ_j , они обе находятся на одной общей ветви тематического дерева, и листовое тематическое множество $\mathcal{T}^{\text{Л}}(\tau_j)$ охватывает (включает) листовое тематическое множество $\mathcal{T}^{\text{Л}}(\tau_i)$.

Таким образом, имеем:

$$\mathcal{T}^{\text{Л}}(\tau_i) \subseteq \mathcal{T}^{\text{Л}}(\tau_j) ,$$

что означает доминирование образа τ_j над образом τ_i .

Если вершины τ_i и τ_j несравнимы, т. е. находятся в разных ветвях тематического дерева, то совершенно очевидно, что они не имеют общих подчиненных листовых вершин. Иначе говоря, пересечение их листовых тематических множеств пусто:

$$\mathcal{T}^{\text{Л}}(\tau_i) \cap \mathcal{T}^{\text{Л}}(\tau_j) = \emptyset .$$

Отсюда следует, что в этом случае множества $\mathcal{T}^{\text{Л}}(\tau_i)$ и $\mathcal{T}^{\text{Л}}(\tau_j)$ несравнимы по отношению включения $\subseteq$.

Из доказанного следует, что отображение $\tau_i \rightarrow \mathcal{T}^{\text{Л}}(\tau_i)$ сохраняет рассматриваемые частичные порядки в обе стороны.

Таким образом, данное отображение является изоморфизмом частично упорядоченного множества тематических рубрик на частично упорядоченное множество листовых тематических множеств. ■

Исходя из изоморфизма множеств $\{\tau_1, \tau_2, \dots, \tau_M\}$ и $\{\mathcal{T}^{\text{Л}}(\tau_1), \mathcal{T}^{\text{Л}}(\tau_2), \dots, \mathcal{T}^{\text{Л}}(\tau_M)\}$, использование в процедурах принятия решения о доступе или непосредственно тематических рубрик классификатора или их образов в виде соответствующих листовых тематических множеств эквивалентно.

Однако, как уже отмечалось, для полного контроля информационных потоков требуется не просто частичный порядок, а решетка. С этой целью необходимо задать границы для любой пары листовых множеств.

При этом использование простых операций теоретико-множественного объединения и пересечения листовых множеств может приводить к неадекватным результатам, в частности выводить свои результаты за пределы множества листовых множеств вершин рубрикатора. Операция пересечения фактически приводит к необходимости добавления к листовым множествам пустого множества, что по доказанному изоморфизму соответствует добавлению к корневому дереву рубрикатора пустой вершины с замыканием на нее всех листовых вершин. Операция объединения может давать своим результатом множество листовых вершин, которому не найдется по иерархическому отношению родительской внутренней вершины рубрикатора.

Проиллюстрируем данные ситуации на следующем примере. Рассмотрим рубрикатор, приведенный на рис. 2.20.

Для рубрик τ_2 и τ_3 имеем:

$$\mathcal{T}^{\text{Л}}(\tau_2) = \{\tau_5, \tau_6, \tau_7\},$$

$$\mathcal{T}^{\text{Л}}(\tau_3) = \{\tau_3\},$$

$$\mathcal{T}^{\text{Л}}(\tau_2) \cap \mathcal{T}^{\text{Л}}(\tau_3) = \emptyset,$$

$$\mathcal{T}^{\text{Л}}(\tau_2) \cup \mathcal{T}^{\text{Л}}(\tau_3) = \{\tau_3, \tau_5, \tau_6, \tau_7\} \neq \{\tau_3, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\} = \mathcal{T}^{\text{Л}}(\tau_1).$$

По иерархическому рубрикатору наименьшей верхней границей для рубрик τ_2 и τ_3 является корень рубрикатора τ_1 , листовое множество которого совпадает с полным множеством всех листовых вершин $\{\tau_3, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$.

Справедливо следующее утверждение.

Лемма 2.3.2. *Теоретико-множественное пересечение $\cap$ листовых множеств вершин иерархического рубрикатора $\mathcal{T}^{\text{Л}}(\tau_i) \cap \mathcal{T}^{\text{Л}}(\tau_j)$ эквивалентно операции взятия наибольшей нижней границы рубрик $\inf_{\text{и}}\{\tau_i, \tau_j\}$.*

Доказательство. Возможны две ситуации.

Рубрики τ_i и τ_j несравнимы.

Рубрики τ_i и τ_j сравнимы, и без ограничения общности положим $\tau_i \leq \tau_j$.

В первом случае по определению операции $\inf_{\mathbf{i}}$ результатом взятия наибольшей нижней границей будет пустая вершина в соответствующем графе $\inf_{\mathbf{i}}\{\tau_i, \tau_j\} = \emptyset$ (см. рис. 2.20).

С другой стороны, очевидно, что при несравнимости вершин τ_i и τ_j теоретико-множественное пересечение их листовых множеств пусто $\mathcal{T}^{\mathbf{L}}(\tau_i) \cap \mathcal{T}^{\mathbf{L}}(\tau_j) = \emptyset$.

Во втором случае по определению операции $\inf_{\mathbf{i}}$ результатом взятия наибольшей нижней границей будет вершина τ_i – $\inf_{\mathbf{i}}\{\tau_i, \tau_j\} = \tau_i$.

С другой стороны также очевидно, что $\mathcal{T}^{\mathbf{L}}(\tau_i) \cap \mathcal{T}^{\mathbf{L}}(\tau_j) = \mathcal{T}^{\mathbf{L}}(\tau_i)$, так как из определения листовых множеств и условия $\tau_i \leq \tau_j$ следует, что $\mathcal{T}^{\mathbf{L}}(\tau_i) \subseteq \mathcal{T}^{\mathbf{L}}(\tau_j)$.

Таким образом, в любом случае операции $\mathcal{T}^{\mathbf{L}}(\tau_i) \cap \mathcal{T}^{\mathbf{L}}(\tau_j)$ и $\inf_{\mathbf{i}}\{\tau_i, \tau_j\}$ дают эквивалентный результат в обе стороны изоморфного по отношению частичного порядка отображения $\tau_i \rightarrow \mathcal{T}^{\mathbf{L}}(\tau_i)$. ■

Для построения операции взятия наименьшей верхней границы листовых множеств, изоморфной операции $\sup_{\mathbf{i}}$, введем следующее понятие.

Определение 2.3.13. *Иерархическим объединением $\cup_{\mathbf{ил}}$ листовых множеств $\mathcal{T}^{\mathbf{L}}(\tau_i)$, $\mathcal{T}^{\mathbf{L}}(\tau_j)$ будем называть листовое множество $\mathcal{T}^{\mathbf{L}}(\tau_k)$ вершины иерархического рубрикатора τ_k , являющейся ближайшим общим предком вершин τ_i и τ_j , порождающих соответствующие листовые множества.*

Совершенно очевидно, что по самому определению 2.3.13 операция $\cup_{\mathbf{ил}}$ эквивалентна операции $\sup_{\mathbf{i}}$. Заметим также, что операции $\cup_{\mathbf{ил}}$ и $\cap$ не выводят свои результаты за пределы множества $\mathcal{T}^{\mathbf{L}}$ листовых множеств вершин иерархического рубрикатора. Таким образом, на множестве $\mathcal{T}^{\mathbf{L}}$ имеем решетку $\Lambda_{\mathbf{i}}(\mathcal{T}^{\mathbf{L}}, \subseteq, \cup_{\mathbf{ил}}, \cap)$, изоморфную решетке $\Lambda_{\mathbf{i}}(\mathbf{T}_{\mathbf{i}\emptyset}, \leq, \sup_{\mathbf{i}}, \inf_{\mathbf{i}})$.

Использование решетки листовых множеств $\Lambda_{\mathbf{i}}(\mathcal{T}^{\mathbf{L}}, \subseteq, \cup_{\mathbf{ил}}, \cap)$ позволяет применять единые механизмы тематического разграничения доступа в системах основанных на дескрипторной и иерархически монорубрицированной тематической классификации.

Для удобства рассмотрения решеток при мультирубрицированном подходе введем следующие понятия.

Определение 2.3.14. *Иерархическим сжатием $\vee_{\mathbf{i}}$ множества элементов $\{\tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_L}\}$ ($L < M$), являющихся вершинами корневого дерева, задающего частичный порядок на множестве рубрик иерархического классификатора $\mathbf{T}_{\mathbf{i}} = \{\tau_1, \tau_2, \dots, \tau_M\}$, будем называть операцию взятия*

ближайшей большей по иерархии дерева вершины τ_k в том и только в том случае, если множество $\{\tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_L}\}$ составляет полный набор вершин, непосредственно подчиненных вершине (родителю) τ_k .

Определение 2.3.15. Множество $I_P \subseteq T_H$ будем называть рубрикаторным идеалом при выполнении двух условий:

- 1) если для любых $\tau \in T_H$ и $\tau' \in I_P$ из $\tau \leq \tau'$ вытекает $\tau \in I_P$, т. е. I_P является порядковым идеалом в корневом дереве рубрик;
- 2) если $\{\tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_L}\} \subseteq I_P$ и $\forall \tau_k \{ \tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_L} \} = \tau_k$, то в состав множества I_P входит и $\tau_k \in I_P$,

где $L \leq M$.

Таким образом, рубрикаторный идеал является множеством вершин иерархического классификатора, которое замкнуто относительно операции взятия меньшего, т. е. является, по сути, объектом, называемым порядковым идеалом. Вместе с тем в отличие от классического порядкового идеала, он обязательно включает вершины-родители, при вхождении в идеал полного набора их сыновей. На рис. 2.21 представлены примеры рубрикаторных идеалов.

С содержательной точки зрения рубрикаторные идеалы представляют объекты, на которые согласно определению 2.3.9 при мультирубрицированной тематической классификации могут отображаться сущности КС (субъекты и объекты доступа), т. е. множество рубрик иерархического классификатора с обязательным наличием подчиненных вершин для каждой нелистой вершины.

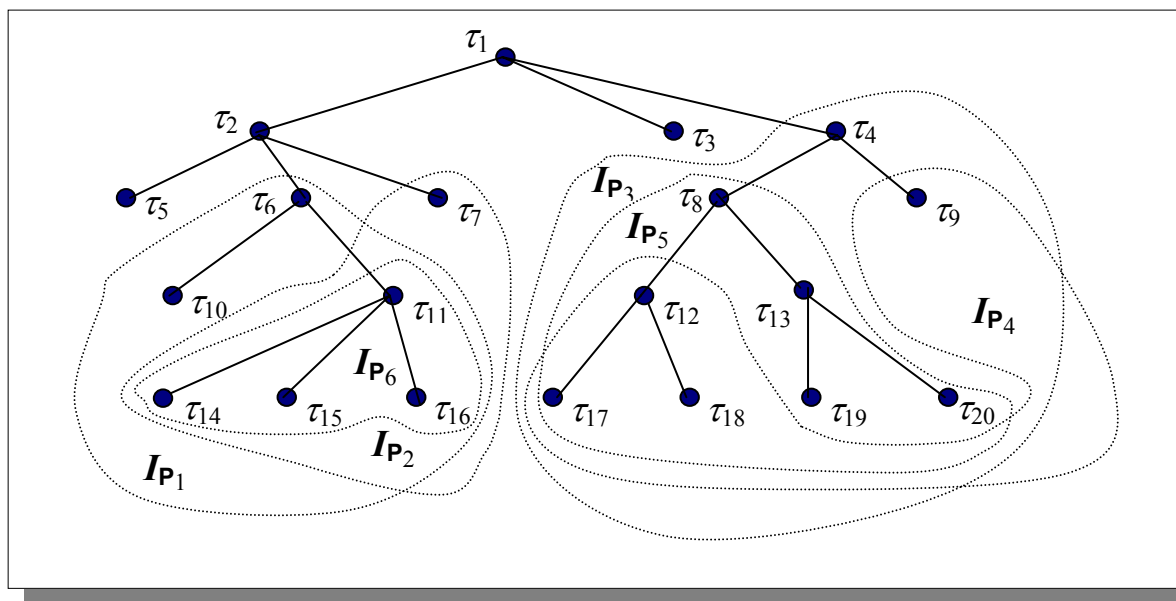


Рис. 2.21. Примеры рубрикаторных идеалов

Множество всех рубрикаторных идеалов, которые могут быть определены на корневом дереве рубрик (на иерархическом классификаторе) будем обозначать I_P .

Отношение включения $\subseteq$ одних рубрикаторных идеалов, являющихся подмножеством вершин корневого дерева рубрик, в другие рубрикаторные идеалы задает на множестве I_P частичный порядок. Отметим также, что пустое множество по *определению 2.3.15* является также рубрикаторным идеалом.

Очевидна справедливость следующего утверждения.

Лемма 2.3.3. *Пересечение $\cap$ любых рубрикаторных идеалов корневого дерева рубрик является также рубрикаторным идеалом.*

Вместе с тем, если рассматривать простое теоретико-множественное объединение $\cup$ рубрикаторных идеалов, то нетрудно заметить, что его результатом могут быть множества, в которых не выполняется второе условие *определения 2.3.15*. В частности, по примерам, представленным на рис. 4.7, не является рубрикаторным идеалом теоретико-множественное объединение рубрикаторных идеалов I_{P_4} и I_{P_5} , так как не содержит вершины τ_4 , хотя включает полный набор ее сыновей $\{\tau_8, \tau_9\}$.

Основываясь на использовании понятия иерархического сжатия по *определению 2.3.14*, можно ввести специальную модификацию операции теоретико-множественного объединения для рубрикаторных идеалов – $\cup_{\text{ир}}$.

Определение 2.3.16. *Объединением $\cup_{\text{ир}}$ рубрикаторных идеалов $I_{P_i} \cup I_{P_j}$ будем называть их теоретико-множественное объединение, дополненное вершинами, полученными иерархическим сжатием подмножеств вершин, начиная с множества $I_{P_i} \cup I_{P_j}$.*

Тогда справедливо следующее утверждение.

Лемма 2.3.4. *Объединение $\cup_{\text{ир}}$ любых рубрикаторных идеалов корневого дерева рубрик является также рубрикаторным идеалом¹.*

В качестве примера отметим, что по представленным на рис. 2.21 рубрикаторным идеалам имеем: $I_{P_6} = I_{P_1} \cap I_{P_2}$, а $I_{P_3} = I_{P_4} \cup_{\text{ир}} I_{P_5}$.

Из *определения 2.3.15*, лемм 2.3.3 и 2.3.4 вытекает важное следствие.

Следствие 2.3.4.1. *Множество всех рубрикаторных идеалов дерева рубрик является решеткой $\Lambda_{\text{и}}(I_P, \subseteq, \cap, \cup_{\text{ир}})$ относительно теоретико-множественного включения $\subseteq$, операций теоретико-множественного пересечения $\cap$ и специальной операции (по определению 2.3.16) объединения $\cup_{\text{ир}}$.*

В результате имеем тематическую решетку, на базе которой можно строить модели тематического разграничения доступа при мультирубрицированной тематической классификации сущностей КС.

¹ С учетом второго условия по *определению 2.3.15*.

Вместе с тем на практике при мультирубрицированной классификации указываются только узловые тематические рубрики без явного включения в тематику сущностей КС соответствующих подчиненных рубрик. Поэтому введем следующее понятие.

Определение 2.3.17. *Мультирубрикой T^M_i называется любое подмножество $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_L}\}$ ($L \leq M$) множества вершин корневого дерева, задающего частичный порядок на множестве $T_{\mathbf{i}} = \{\tau_1, \tau_2, \dots, \tau_M\}$, при выполнении следующих двух условий:*

1) *любая вершина τ_{i_k} несравнима с любой другой вершиной τ_{i_m} того же подмножества¹ –*

$$\tau_{i_k} \diamond \tau_{i_m}, \quad k \neq m;$$

2) *в $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_L}\}$ не содержится полного набора сыновей ни одной из вершин.*

Множество всех мультирубрик на корневом дереве рубрик будем обозначать T^M ($T^M \in T^M$).

Покажем, что между рубрикаторными идеалами и мультирубриками имеется взаимно однозначное соответствие.

Для удобства введем несколько вспомогательных определений.

Определение 2.3.18. *Порожденным (частичным порядком) множеством будем считать наименьшее множество, содержащее порождающее (исходное) множество и замкнутое относительно операции взятия меньшего.*

Пусть $M \subseteq T_{\mathbf{i}}$. Через $\langle M \rangle$ обозначим рубрикаторный идеал, порожденный множеством M , т. е. пересечение всех рубрикаторных идеалов, содержащих M . Ясно, что $\langle M \rangle$ является наименьшим рубрикаторным идеалом, содержащим M .

Определение 2.3.19. *Элемент τ рубрикаторного идеала $I_{\mathbf{p}}$ будем называть максимальным, если для любого $\tau' \in I_{\mathbf{p}}$ либо $\tau' \leq \tau$, либо τ' и τ несравнимы.*

Обозначим через A множество всех максимальных элементов из рубрикаторного идеала $I_{\mathbf{p}}$. Тогда по определению 4.18 очевидно, что A является порождающим множеством рубрикаторного идеала $I_{\mathbf{p}}$. Заметим, далее, что по определению 4.18 также очевидно, что A состоит из попарно несравнимых элементов. Подобные подмножества будем называть *антицепями* (в отличие от цепей, образуемых множеством попарно сравнимых элементов).

Очевидна справедливость следующего утверждения.

¹ $\diamond$ - знак несравнимости.

Лемма 2.3.5. *Антицепь, образуемая множеством всех максимальных элементов некоторого рубрикаторного идеала, является мультирубрикой, порождающей соответствующий рубрикаторный идеал.*

Таким образом, имеется естественное взаимно однозначное соответствие между мультирубриками и рубрикаторными идеалами. Отметим, что пустой мультирубрике соответствует пустой рубрикаторный идеал. Кроме того, каждой рубрике отвечает одноэлементная мультирубрика, ее содержащая. Для удобства в дальнейшем одноэлементную мультирубрику будем отождествлять с ее единственной рубрикой.

Покажем далее, что на множестве мультирубрик можно построить решетку, эквивалентную решетке рубрикаторных идеалов $\Lambda_{\mathbf{u}}(\mathbf{I}_{\mathbf{P}}, \subseteq, \cap, \cup_{\text{ир}})$.

С этой целью, прежде всего, введем понятие доминирования мультирубрик.

Определение 2.3.20. *Мультирубрика $\mathcal{T}^{\mathbf{M}}_i$ доминирует¹ над мультирубрикой $\mathcal{T}^{\mathbf{M}}_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\} \leq \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_I}\}$ в том и только в том случае, когда для любого $m=1, \dots, J$ существует $k=1, \dots, I$ такое, что $\tau_{j_m} \leq \tau_{i_k}$ (вершина τ_{j_m} подчинена по корневому дереву вершине τ_{i_k}):*

$$\forall \tau_{j_m} \in \mathcal{T}^{\mathbf{M}}_j, \exists \tau_{i_k} \in \mathcal{T}^{\mathbf{M}}_i, \tau_{j_m} \leq \tau_{i_k}.$$

Легко проверить, что введенное отношение доминирования мультирубрик $\leq$ рефлексивно, транзитивно и антисимметрично, т. е. является отношением частичного порядка.

В частности, вернувшись еще раз к корневому дереву на рис. 4.7, можно привести следующие примеры доминирования мультирубрик:

$\{\tau_5, \tau_6\} \leq \{\tau_2, \tau_3\}$, $\{\tau_3, \tau_5, \tau_6\} \leq \{\tau_2, \tau_3\}$, $\{\tau_7, \tau_{11}\} \leq \{\tau_2, \tau_3\}$, $\{\tau_{13}, \tau_{17}\} \leq \{\tau_7, \tau_8\}$, $\{\tau_{12}, \tau_{19}\} \leq \{\tau_7, \tau_8\}$, $\{\tau_9, \tau_{10}, \tau_{13}, \tau_{14}, \tau_{16}, \tau_{18}\} \leq \{\tau_3, \tau_4, \tau_6\}$, $\{\tau_{11}, \tau_{12}, \tau_{19}\} \leq \{\tau_3, \tau_4, \tau_6\}$ и т. д.

Справедливо следующее утверждение.

Теорема 2.3.3. *Частично упорядоченное относительно отношения включения $\subseteq$ множество рубрикаторных идеалов изоморфно частично упорядоченному по отношению доминирования $\leq$ множеству мультирубрик иерархического классификатора.*

Доказательство.

Лемма 2.3.5 определяет биективность отображения $\mathbf{I}_{\mathbf{P}_i} \rightarrow \mathcal{T}^{\mathbf{M}}_i$. Действительно, разные рубрикаторные идеалы имеют различные наборы максимальных элементов и, следовательно, им соответствуют различные мультирубрики. В свою очередь, различные мультирубрики порождают различные рубрикаторные идеалы.

¹ Для обозначения доминирования мультирубрик будем использовать тот же знак $\leq$.

Пусть для рубрикаторных идеалов I_{P_j} и I_{P_i} выполняется $I_{P_j} \subseteq I_{P_i}$. Тогда очевидно для мультирубрик T^M_j и T^M_i выполняется условие *определения 2.3.20*, т. е. $T^M_j \leq T^M_i$.

Верно и обратное. Пусть мультирубрики сравнимы и $T^M_j \leq T^M_i$. Тогда для рубрики из набора $\{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ найдется рубрика-доминанта в наборе $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$. Отсюда следует, что подмножества, порождаемые рубриками из набора $\{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ включаются в подмножества рубрик, порождаемых соответствующими доминантами из набора $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$. Следовательно, и все множество рубрик рубрикаторного идеала I_{P_j} входит (является подмножеством) рубрик, составляющих рубрикаторный идеал I_{P_i} .

Таким образом, отображение $I_{P_i} \rightarrow T^M_i$ сохраняет частичные порядки в обе стороны. ■

Кроме отношения доминирования $\leq$, задающего на множестве мультирубрик частичный порядок, для построения решетки мультирубрик необходимо задать операции, формирующие наименьшую верхнюю и наибольшую нижнюю границы для любой пары мультирубрик.

Исходя из самого смысла мультирубрики (*определение 4.17*), понятия доминирования мультирубрик (*определение 4.20*), можно установить следующий порядок объединения мультирубрик.

Определение 2.3.21. Объединением $\cup_M$ мультирубрик $T^M_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$ и $T^M_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ называется операция формирования множества вершин иерархического рубрикатора $T^{M\cup} = T^M_i \cup_M T^M_j$ на основе следующего алгоритма:

1) Формируется теоретико-множественное объединение множеств вершин, составляющих мультирубрики –

$$T^\cup = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\} \cup \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\};$$

2) Формируется набор вершин $T^{M\cup'}$ путем исключения из набора $T^\cup$ тех вершин, которые доминируются¹ хотя бы одной другой вершиной из того же набора $T^\cup$ –

$$(\tau_k \in T^\cup \wedge \tau_k \notin T^{M\cup'}) \equiv (\exists \tau_m \in T^\cup \wedge \tau_m \leq \tau_k \wedge \tau_m \neq \tau_k);$$

3) Формируется итоговый набор вершин $T^{M\cup}$ путем добавления в набор $T^{M\cup'}$ результатов иерархического сжатия по всем подмножествам набора вершин $T^{M\cup'}$ и одновременным исключением соответствующих наборов сыновей при непустом результате сжатия.

¹ Без учета доминирования вершины самой себя.

Иными словами – для построения мультирубрики, являющейся объединением двух мультирубрик, необходимо в теоретико-множественном объединении исходных мультирубрик оставить максимальные (*определение 2.3.19*) элементы и произвести при необходимости иерархические сжатия.

Справедливо следующее утверждение.

Лемма 2.3.6. Множество рубрик $\mathcal{T}^{\mathbf{M}\cup} = \mathcal{T}^{\mathbf{M}}_i \cup_{\mathbf{M}} \mathcal{T}^{\mathbf{M}}_j$, формируемое на основе объединения мультирубрик по определению 4.21,

- a) является мультирубрикой;
- b) доминирует над мультирубриками $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$, т. е. $\mathcal{T}^{\mathbf{M}}_i \leq \mathcal{T}^{\mathbf{M}\cup} \wedge \mathcal{T}^{\mathbf{M}}_j \leq \mathcal{T}^{\mathbf{M}\cup}$;
- c) является наименьшей верхней границей мультирубрик $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$.

Доказательство.

По второму и третьему шагу алгоритма формирования множества $\mathcal{T}^{\mathbf{M}\cup}$ в нем остаются только максимальные элементы и отсутствуют полные наборы сыновей каких-либо вершин классификатора. Следовательно, по определению 2.3.17 набор $\mathcal{T}^{\mathbf{M}\cup}$ является мультирубрикой.

По второму шагу алгоритма формирования $\mathcal{T}^{\mathbf{M}\cup}$ для любой вершины из наборов, составляющих мультирубрики $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$, в $\mathcal{T}^{\mathbf{M}\cup}$ обязательно найдется вершина-доминанта. Следовательно, по определению 4.20 $\mathcal{T}^{\mathbf{M}\cup}$ доминирует над $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$.

Наибольшая близость сверху к мультирубрикам $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$ мультирубрики $\mathcal{T}^{\mathbf{M}\cup}$ обеспечивается первым шагом алгоритма, в результате которого в первичном наборе $\mathcal{T}^{\cup}$ оказываются самые ближайшие по иерархии рубрики (т. е. равные сами себе). Второй шаг оставляет только максимальные элементы, что минимально необходимо для обеспечения доминирования результата над объединяемыми рубриками. Третий шаг также не может внести "избыточности" доминирования, так как при иерархическом сжатии берутся *ближайшие* родители. Таким образом, более близкой сверху к мультирубрикам $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$, чем мультирубрика $\mathcal{T}^{\mathbf{M}\cup}$, существовать не может. ■

Аналогичным образом установим следующий порядок пересечения мультирубрик.

Определение 2.3.22. Пересечением $\cap_{\mathbf{M}}$ мультирубрик $\mathcal{T}^{\mathbf{M}}_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$ и $\mathcal{T}^{\mathbf{M}}_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ называется операция формирования множества вершин иерархического рубрикатора $\mathcal{T}^{\mathbf{M}\cap} = \mathcal{T}^{\mathbf{M}}_i \cap_{\mathbf{M}} \mathcal{T}^{\mathbf{M}}_j$ на основе следующего алгоритма:

- 1) Из множества вершин мультирубрики $\mathcal{T}^{\mathbf{M}}_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$ формируется множество вершин $\mathcal{T}^{\mathbf{M}'}_i$, которые доминируются¹ хотя бы одной вершиной из множества вершин другой мультирубрики $\mathcal{T}^{\mathbf{M}}_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$;
- 2) Из множества вершин мультирубрики $\mathcal{T}^{\mathbf{M}}_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ формируется множество вершин $\mathcal{T}^{\mathbf{M}'}_j$, которые доминируются хотя бы одной вершиной из множества вершин первой мультирубрики $\mathcal{T}^{\mathbf{M}}_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$;
- 3) Формируется теоретико-множественное объединение $\mathcal{T}^{\mathbf{M} \cap} = \mathcal{T}^{\mathbf{M}'}_i \cup \mathcal{T}^{\mathbf{M}'}_j$.

Покажем справедливость следующего утверждения.

Лемма 2.3.7. Множество рубрик $\mathcal{T}^{\mathbf{M} \cap} = \mathcal{T}^{\mathbf{M}}_i \cap_{\mathbf{M}} \mathcal{T}^{\mathbf{M}}_j$, формируемое на основе пересечения мультирубрик по определению 4.22,

- a) является мультирубрикой;
- b) доминируется мультирубриками $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$, т. е. $\mathcal{T}^{\mathbf{M} \cap} \leq \mathcal{T}^{\mathbf{M}}_i \wedge \mathcal{T}^{\mathbf{M} \cap} \leq \mathcal{T}^{\mathbf{M}}_j$;
- c) является наибольшей нижней границей мультирубрик $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$.

Доказательство.

По определению 2.3.17 в множествах $\mathcal{T}^{\mathbf{M}}_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_l}\}$ и $\mathcal{T}^{\mathbf{M}}_j = \{\tau_{j_1}, \tau_{j_2}, \dots, \tau_{j_j}\}$ могут быть только несравнимые вершины. По первому и второму шагу алгоритма в множествах $\mathcal{T}^{\mathbf{M}'}_i$ и $\mathcal{T}^{\mathbf{M}'}_j$ остаются также несравнимые вершины.

Пусть также от противного, $\mathcal{T}^{\mathbf{M} \cap} = \mathcal{T}^{\mathbf{M}'}_i \cup \mathcal{T}^{\mathbf{M}'}_j$ содержит полный набор V сыновей некоторой рубрики τ , скажем $V = \{\tau_1, \tau_2, \dots\}$. Тогда либо рубрики $\{\tau_1, \tau_2, \dots\}$ "пришли" в $\mathcal{T}^{\mathbf{M} \cap}$ из одной мультирубрики, что невозможно по определению 2.3.17, либо из разных мультирубрик, часть из $\mathcal{T}^{\mathbf{M}'}_i$, часть из $\mathcal{T}^{\mathbf{M}'}_j$. Для определенности без ограничения общности положим, что $V = \{\tau_1, \tau_2\}$, $\tau_1 \in \mathcal{T}^{\mathbf{M}}_i$, $\tau_2 \in \mathcal{T}^{\mathbf{M}}_j$ – см. рис. 2.22.

¹ В данном случае вершина доминируется, в том числе, и сама собой, т. е. той же вершиной, находящейся в другой мультирубрике.

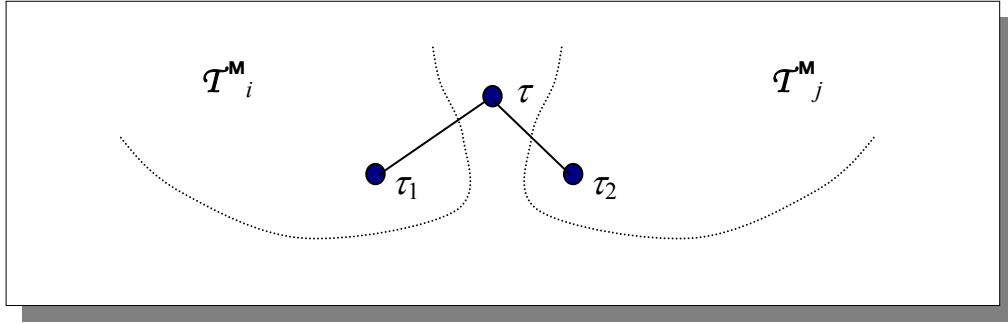


Рис. 2.22. Пример фрагмента корневого дерева для иллюстрации пересечения мульти-рубрик

В этом случае, чтобы рубрика τ_1 оказалась в $\mathcal{T}^{\mathbf{M}\cap}$, по *определению* 2.3.22 она сама или τ должны быть в мультирубике $\mathcal{T}^{\mathbf{M}}_j$. Однако тогда в $\mathcal{T}^{\mathbf{M}}_j$ также присутствует или полный набор сыновей $V = \{\tau_1, \tau_2\}$ рубрики τ , или сравнимые рубрики $\tau_1 \leq \tau_2$, что также невозможно по *определению* 2.3.17.

Если некоторая рубрика τ оказалась в $\mathcal{T}^{\mathbf{M}\cap}$, то, во-первых, она доминируется сама собой в той рубрике, в которой находилась до процедуры пересечения, скажем в $\mathcal{T}^{\mathbf{M}}_i$. Во-вторых, для попадания $\tau \in \mathcal{T}^{\mathbf{M}}_i$ в $\mathcal{T}^{\mathbf{M}\cap}$ необходимо, чтобы в $\mathcal{T}^{\mathbf{M}}_j$ существовала рубрика-доминанта $\tau' \in \mathcal{T}^{\mathbf{M}}_j$, $\tau \leq \tau'$. Отсюда следует, что мультирубика $\mathcal{T}^{\mathbf{M}\cap}$ доминируется и той и другой из исходных мультирубрик $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$.

Предположим также от противного, что существует некоторая мультирубика $\mathcal{T}^{\mathbf{M}'}$, более близкая по доминированию к мультирубикам $\mathcal{T}^{\mathbf{M}}_i$ и $\mathcal{T}^{\mathbf{M}}_j$ — $(\mathcal{T}^{\mathbf{M}\cap} \leq \mathcal{T}^{\mathbf{M}'} \leq \mathcal{T}^{\mathbf{M}}_i) \wedge (\mathcal{T}^{\mathbf{M}\cap} \leq \mathcal{T}^{\mathbf{M}'} \leq \mathcal{T}^{\mathbf{M}}_j)$. Тогда имеем $(\tau \leq \tau' \leq \tau^{(i)}) \wedge (\tau \leq \tau' \leq \tau^{(j)})$, где $\tau \in \mathcal{T}^{\mathbf{M}\cap}$, $\tau' \in \mathcal{T}^{\mathbf{M}'}$, $\tau^{(i)} \in \mathcal{T}^{\mathbf{M}}_i$ и $\tau^{(j)} \in \mathcal{T}^{\mathbf{M}}_j$. С другой стороны, по алгоритму, задаваемому *определением* 4.17, τ должна входить или в $\mathcal{T}^{\mathbf{M}}_i$, или в $\mathcal{T}^{\mathbf{M}}_j$. Если $\tau \neq \tau' \neq \tau^{(i)} \neq \tau^{(j)}$, тогда, с учетом того, что $(\tau \leq \tau') \wedge (\tau \leq \tau^{(i)}) \wedge (\tau \leq \tau^{(j)})$, или в мультирубике $\mathcal{T}^{\mathbf{M}}_i$, или в мультирубике $\mathcal{T}^{\mathbf{M}}_j$ окажутся сравнимые рубрики $(\tau \text{ и } \tau^{(i)} \text{ или } \tau \text{ и } \tau^{(j)})$, что противоречит условиям *определения* 2.3.17. Таким образом, предполагаемая ситуация возможна только при условии $\tau = \tau'$, что означает, что $\mathcal{T}^{\mathbf{M}\cap}$ и есть ближайшая снизу мультирубика. ■

В результате, на основе введенных отношения доминирования $\leq$ (*определение* 2.3.20), операций объединения $\cup_{\mathbf{M}}$ (*определение* 2.3.21) и пересечения $\cap_{\mathbf{M}}$ (*определение* 2.3.22) мультирубрик, а также с учетом *лемм* 2.3.5, 2.3.6 и 2.3.7, получаем на множестве мультирубрик $\mathbf{T}^{\mathbf{M}}$ решетку $\Lambda_{\mathbf{I}}(\mathbf{T}^{\mathbf{M}}, \leq, \cap_{\mathbf{M}}, \cup_{\mathbf{M}})$, эквивалентную (изоморфную) решетке рубрикаторных идеалов $\Lambda_{\mathbf{I}}(\mathbf{I}_{\mathbf{P}}, \subseteq, \cap, \cup_{\mathbf{IP}})$. Решетка $\Lambda_{\mathbf{I}}(\mathbf{I}_{\mathbf{P}}, \subseteq, \cap, \cup_{\mathbf{IP}})$ и, в особенности, ре-

сетка мультибрук $\Lambda_{\mathbf{m}}(\mathbf{T}^{\mathbf{m}}, \leq, \cap_{\mathbf{m}}, \cup_{\mathbf{m}})$, как более адекватная механизмам мультибрукцированной тематической классификации, могут быть положены в основу моделей разграничения доступа с полным контролем информационных потоков.

2.3.3. Модель тематико-иерархического разграничения доступа

Основные положения модели сводятся к следующему.

1. Информационно-логическая схема предметной области системы представляется тематическим иерархическим классификатором (рубрикатором).

Рубрикатор включает конечное множество тематических рубрик $T_{\text{и}} = \{\tau_1, \tau_2, \dots, \tau_M\}$, на котором установлен частичный порядок, задаваемый корневым деревом.

2. Множество сущностей системы $X = S \cup O$ тематически классифицируется на основе отображения на множество мультирубрик T^M , определенных на корневом дереве иерархического рубрикатора.

В рамках мультирубрицированного отображения $F_{\text{и2}}[x]$ сущностей КС на иерархический рубрикатор будем считать, что существует функция тематического окрашивания f_m , которая в каждый момент времени для любой сущности системы $x \in X$ определяет соответствующую ей мультирубрику:

$$f_m[x] = T^M_i, \quad (2.3.1)$$

где $x \in S \cup O$, $T^M_i \in T^M$.

3. Функционирование системы сопровождается переходами из одного состояния (момент времени t_k) в другое состояние (момент времени t_{k+1}). В результате перехода в системе возникают новые отношения доступа (новые потоки) между существующими сущностями ($X = S \cup O$), либо возникают/удаляются новые сущности – субъекты и объекты доступа.

В дальнейшем ограничимся двумя видами потоков – на запись (w) в объект и на чтение (r) из объекта, представленных на рис. 2.23.

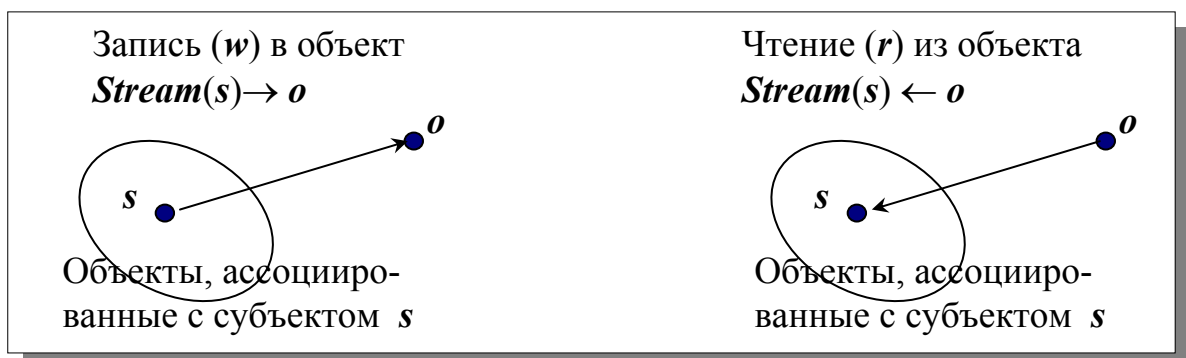


Рис. 2.23. Потоки на запись и чтение

Множество потоков обладает свойством транзитивности. В частности, с точки зрения безопасности чтение субъектом s_m из одного объекта o_i

и запись в другой объект o_j и, далее, чтение другим субъектом s_n из объекта o_j и запись в объект o_k , тождественно потоку из информации объекта o_i в объект o_k через субъекты s_m и s_n – $\text{Stream}(s_m, s_n, o_i, o_j) \rightarrow o_k$.

С учетом понятия субъекта и объекта доступа (определения 1.3.2, 1.3.3), источника для субъекта доступа (определение 1.3.4), объектов, ассоциированных с субъектом доступа (определение 1.3.5), процессы порождения субъектов и объектов доступа будем определять потоками, представленными на рис. 2.24.

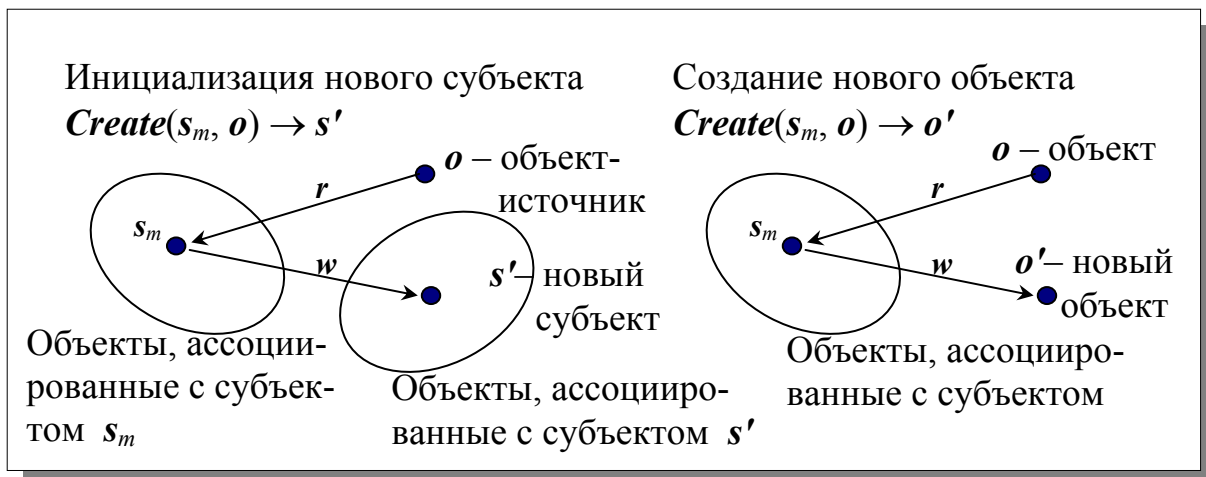


Рис. 2.24. Потоки при создании субъектов и объектов доступа

Таким образом, переходы системы, сопровождающиеся созданием новых объектов или субъектов доступа, описываются, как и переходы, вызываемые доступами существующих субъектов к существующим объектам, также двумя видами потоков – на чтение и на запись.

Проблема разграничения (безопасности) доступа заключается в синтезе таких механизмов принятия решения о правомочности доступа субъектов к объектам, при которых с учетом транзитивности потоков реализуется некоторое подмножество безопасных по определенному критерию потоков.

4. На основе анализа сущности мультирубрицированной тематической классификации и с целью конкретизации общего критерия безопасности тематического доступа по *определению 2.3.2*, сформулируем следующий очевидный критерий безопасности для системы $\Sigma_{\text{Тии}}$.

Определение 2.3.23. Система безопасна тогда и только тогда, когда в ней отсутствуют потоки следующих видов:

- от сущностей с более широкой тематикой к сущностям с более узкой тематикой;
- между несравнимыми по тематике сущностями.

5. Переходы системы, обусловленные запросами и осуществлением доступов существующих субъектов к существующим объектам, санк-

ционируются монитором безопасности объектов на основе следующих правил, вытекающих из критерия, задаваемого *определением 2.3.23*.

Правило 2.3.1. *Доступ субъекта s к объекту o , вызывающий поток по чтению $Stream(s) \leftarrow o$, неопасен и может быть МБО разрешен тогда и только тогда, когда мультирубрика субъекта доминирует над мультирубрикой объекта:*

$$f_m[s] \geq f_m[o] . \quad (2.3.2)$$

Правило 2.3.2. *Доступ субъекта s к объекту o , вызывающий поток по записи $Stream(s) \rightarrow o$, неопасен и может быть МБО разрешен тогда и только тогда, когда мультирубрика объекта доминирует над мультирубрикой субъекта:*

$$f_m[o] \geq f_m[s] . \quad (2.3.3)$$

Из правила 2.3.2 следует, что при создании нового объекта o' МБО должен присвоить ему мультирубрику, тождественную или более широкую, чем мультирубрика субъекта, осуществляющего операцию создания объекта - $f_m[o'] \geq f_m[s]$. Данная процедура может быть полностью автоматизирована (строгий принцип) или осуществляться на основе специальных запросов пользователю (нестрогий принцип). При строгом принципе МБО присваивает новому объекту мультирубрику субъекта. При нестрогом принципе МБО на основе специального запроса к пользователю может присвоить новому объекту любую мультирубрику, доминирующую (более широкую) над мультирубрикой субъекта.

6. Переходы системы, связанные с порождением новых объектов и субъектов доступа санкционируются МБО и МБС на основе следующих правил.

Правило 2.3.3. *Порождение субъектом s нового объекта o' , в том числе и за счет чтения из другого объекта o , неопасно и может быть МБО разрешено тогда и только тогда, когда мультирубрика субъекта доминирует над мультирубрикой объекта o , при этом МБО присваивает новому объекту o' мультирубрику, доминирующую над мультирубрикой субъекта:*

$$f_m[o] \leq f_m[s] \leq f_m[o'] . \quad (2.3.4)$$

При инициализации нового субъекта доступа ситуация иная. Исходя из аксиоматического *предположения 4.4*¹, монитор безопасности субъектов

¹ Процессы изменения тематического классификатора или тематик пользователей рассматриваются как процессы создания новой системы тематического разграничения

реализует следующее правило (аналогичное соответствующим правилам в моделях $\Sigma_{\text{Тдс}}$ и $\Sigma_{\text{Тднс}}$).

Правило 2.3.4. *Инициализация субъектом s нового субъекта s' посредством воздействия на объект источник o неопасна и может быть МБС разрешена тогда и только тогда, когда мультирубрика субъекта доминирует над мультирубрикой объекта-источника, при этом МБС присваивает новому субъекту мультирубрику, тождественную мультирубрике инициализирующего субъекта:*

$$f_m[o] \leq f_m[s] \equiv f_m[s'] . \quad (2.3.5)$$

7. В системе коллективного доступа помимо потоков, вызываемых одиночными доступами (один субъект к одному объекту), могут существовать потоки, вызываемые множественными доступами (один субъект одновременно к нескольким объектам, несколько субъектов одновременно к одному объекту).

Санкционирование подобных потоков осуществляется на основе следующего правила.

Правило 2.3.5. *Одновременный множественный доступ субъекта s к объектам $o_1, o_2, \dots$ или субъектов $s_1, s_2, \dots$ к объекту o может быть разрешен (неопасен) тогда и только тогда, когда каждый одиночный доступ из запрашиваемой совокупности доступов, образующих множественный доступ, удовлетворяет правилам 2.3.1, 2.3.2, 2.3.3 и 2.3.4.*

Правило 2.3.5 вытекает из транзитивности множества потоков, а также участия в потоках информации объектов, ассоциированных с субъектами доступа, что с точки зрения безопасности обуславливает эквивалентность последовательности одиночных потоков и соответствующего множественного доступа. Кроме того, отметим, что это правило распространяется, в том числе, и на процессы порождения новых объектов и инициализации новых субъектов доступа.

Справедливо следующее утверждение.

Теорема 2.3.4. *В системе с отображением множества субъектов и объектов доступа на множество тематических мультирубрик, в которой доступы санкционируются по правилам 2.3.1, 2.3.2, 2.3.3, 2.3.4 и 2.3.5, реализуется множество только таких потоков, которые удовлетворяют критерию безопасности по определению 2.3.23.*

Доказательство.

Во-первых, отметим, что по правилам 2.3.1...2.3.5 потоки между несравнимыми по мультирубрикам сущностями системы невозможны.

Во-вторых, утверждение очевидно в случае одиночных доступов субъектов к объектам, в том числе и для потоков, связанных с созданием новых объектов и инициализацией новых субъектов доступа.

В случае множественных доступов рассмотрим те из них, которые по свойству транзитивности вызывают потоки между объектами или субъектами – см. рис. 2.25

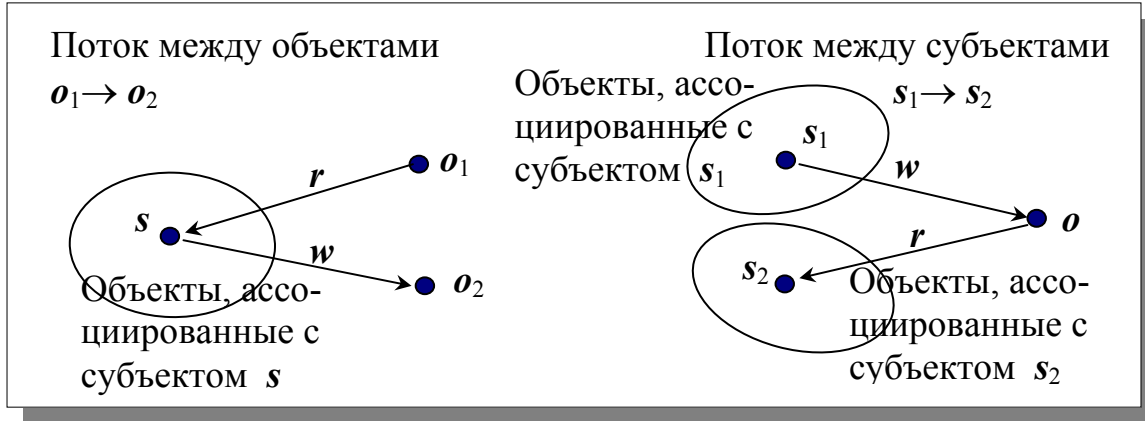


Рис. 2.25. Множественные доступы, вызывающие по свойству транзитивности потоки между объектами и между субъектами

По условиям теоремы при санкционировании потока $o_1 \rightarrow o_2$ имеем:

$$f_m[s] \geq f_m[o_1] \quad \wedge \quad f_m[o_2] \geq f_m[s] .$$

Отсюда следует, что:

$$f_m[o_2] \geq f_m[o_1] .$$

Таким образом, осуществляется неопасный поток от сущности с менее широкой тематикой к сущности с более широкой тематикой.

Аналогично по условиям теоремы при санкционировании потока $s_1 \rightarrow s_2$ имеем

$$f_m[s_1] \leq f_m[o] \quad \wedge \quad f_m[s_2] \geq f_m[o] .$$

Отсюда следует, что

$$f_m[s_2] \geq f_m[s_1] .$$

Таким образом, и в этом случае осуществляется неопасный поток от сущности с менее широкой тематикой к сущности с более широкой тематикой.

Нетрудно видеть, что любые другие множественные потоки являются комбинацией двух рассмотренных потоков $o_1 \rightarrow o_2$ и $s_1 \rightarrow s_2$, из чего по

свойству транзитивности вытекает наличие в системе только неопасных потоков, т. е. потоков, удовлетворяющих критерию безопасности по *определению 2.3.23*. ■

Таким образом, компьютерная система, построенная на основе спецификаций модели тематико-иерархического разграничения доступа, функционирует в рамках требований безопасности тематической политики.

2.4. МОДЕЛИ БЕЗОПАСНОСТИ НА ОСНОВЕ РОЛЕВОЙ ПОЛИТИКИ

2.4.1. Общая характеристика моделей разграничения доступа на основе функционально-ролевых отношений

В основе рассмотренных ранее политик безопасности лежат отношения между отдельным пользователем (субъектом) и объектом доступа, определяемые либо внешним фактором (дискреционный доступ), либо уровнем безопасности (мандатный доступ), либо тематикой информации (тематический доступ).

Вместе с тем, анализ различных организационно-управленческих и организационно-технологических схем, показывает, что в реальной жизни сотрудники предприятий, учреждений выполняют определенные функциональные обязанности не от своего личного имени, а в рамках некоторой должности. Должность, которую можно трактовать как определенную роль, представляет некоторую абстрактную, точнее обобщенную сущность, выражающую определенный тип функций и тип положения работника (подчиненность, права и полномочия). Таким образом, в реальной жизни в большинстве организационно-технологических схем права и полномочия предоставляются конкретному сотруднику не лично (непосредственно), а через назначение его на определенную должность (роль), с которой он и получает некоторый типовой набор прав и полномочий.

Еще одним аспектом реальных организационно-технологических и управленческих схем является использование понятий прав и полномочий, как неких процедур над ресурсами системы, отражающих организационно-технологические процессы предметной области КС. Иначе говоря, права и полномочия сотрудникам по их должностям предоставляются не на уровне элементарных операций над ресурсами (читать, изменять, добавлять, удалять, создавать), а на уровне совокупностей элементарных операций, сгруппированных в отдельные логически обобщенные процедуры обработки информации (например, кредитные или дебетные операции над определенными бюджетами).

Таким образом, политика разграничения доступа в компьютерных системах, автоматизирующих те или иные организационно-технологические или организационно-управленческие процессы, должна строиться на основе функционально-ролевых отношений, складывающихся в предметной области КС.

Впервые подобный подход был рассмотрен в конце 70-х – начале 80-х годов в исследованиях по процессам разграничения доступа корпорации IBM и получил название *ролевого управления доступом*. В начале 80-х годов была представлена модель Лендвера-МакЛина, встречающаяся в ли-

тературе также под названием **MMS-модели**¹, сочетающая дискреционный и мандатный принципы разграничения доступа с использованием понятия и механизма ролей. Несколько позже появились и формальные выражения ролевых основ управления доступом (**Role-Based Access Control – RBAC**).

Основой ролевых моделей, как отмечалось, является введение в субъектно-объектную модель КС дополнительной категории активных сущностей – ролей. Можно дать следующее формальное определение роли.

Определение 2.4.1. *Ролью называется активно действующая в КС абстрактная сущность, обладающая логически взаимосвязанным набором полномочий, необходимых для выполнения определенных функциональных обязанностей пользователями системы.*

Полномочия, как уже отмечалось, трактуются, как право осуществлять некоторые функционально-логические процедуры над всей совокупностью объектов системы или над определенной их группой. При этом, однако, в известных формальных ролевых моделях не вводятся отдельные механизмы спецификации полномочий, а используется традиционный набор элементарных методов доступа (чтение, запись, и т. д.). В то же время в таких широко распространенных разновидностях систем, как СУБД, подобные спецификации функционально-логических процедур над данными используются повсеместно. Основу обработки данных в реляционных СУБД составляют запросы, обособляющие в отдельные именованные сущности операции над данными (инструкции *SELECT*, *INSERT*, *UPDATE*, *DELETE*), объекты данных (таблицы) и результаты обработки. Сконструированные и выраженные на языке SQL запросы хранятся в БД вместе с данными и составляют отдельную группу объектов (сущностей) базы данных. Пользователям системы предоставляются права запускать определенные запросы, что можно интерпретировать как дискреционный способ предоставления полномочий по обработке данных.

В операционных системах, ввиду их бóльшей универсальности и ориентированности на самый широкий круг предметных областей, полномочия ролей (например, для ролей администраторов, аудиторов, или полномочия для рабочих групп пользователей) определяются чаще всего на основе дискреционного принципа через права по определенным методам доступа к определенным объектам системы или к объектам отдельных категорий (к спискам доступа, к журналу аудита и т. д.). Подобный подход называют механизмом *привилегий*.

Введение ролей приводит к двухэтапной организации системы разграничения доступа:

- I. Создание ролей и определение их полномочий (прав доступа к объектам);

¹ Модель системы военных сообщений – *Military Message System*.

II. Назначение ролей пользователям системы.

Соответственно формальные спецификации ролевых моделей должны регламентировать тем или иным способом, точнее в рамках той или иной политики, и определение полномочий ролям и назначение ролей пользователям.

Управление доступом в ролевых системах требует разбиения процесса функционирования системы и работы пользователя на *сеансы*, в каждом из которых, в свою очередь, выделяется две фазы:

1. Авторизация в данном сеансе пользователя с одной или несколькими разрешенными (назначенными на втором этапе организации доступа) для него ролями;
2. Разрешение или запрещение субъектам пользователя доступа к объектам системы в рамках полномочий соответствующих ролей, с которыми авторизован в данном сеансе пользователь.

Нетрудно увидеть, что ролевые модели сочетают мандатный подход к организации доступа через определенную агрегацию субъектов и объектов доступа, и тем самым обеспечивают жесткость правил разграничения доступа, и дискреционный подход, обеспечивающий гибкость в настройке системы разграничения доступа на конкретные функционально-организационные процессы предметной области КС. Данные особенности ролевой политики позволяют строить системы разграничения доступа с хорошей управляемостью в сложных системах с большим количеством пользователей и объектов, и поэтому находят широкое применение в практических системах.

2.4.2. Формальная спецификация и разновидности ролевых моделей

Приведем формальную спецификацию ролевой модели разграничения доступа.

1. КС представляется совокупностью следующих множеств:

- множества пользователей U ;
- множества ролей $\mathcal{R}$;
- множества полномочий P ;
- множества сеансов C работы пользователей с системой.

Множество полномочий P в общем виде задается специальными механизмами, объединяющими операции доступа и объекты доступа, например, запросами на обработку данных в СУБД, или иными именованными процедурами обработки данных, в том числе возможно высокого логического уровня.

2. Ролевые отношения устанавливаются следующими отображениями множеств сущностей системы:

$\mathcal{F}_{P\mathcal{R}}: P \times \mathcal{R}$ – отображение множества полномочий на множество ролей;

$\mathcal{F}_{U\mathcal{R}}: U \times \mathcal{R}$ – отображение множества пользователей на множество ролей.

Нетрудно видеть, что отображения $\mathcal{F}_{P\mathcal{R}}$ и $\mathcal{F}_{U\mathcal{R}}$ обеспечивают первый и второй этапы процессов организации системы ролевого доступа. При этом отображение $\mathcal{F}_{U\mathcal{R}}$ может реализовываться механизмами одной из базовых политик разграничения доступа – матрицей "Пользователи-Роли", или на основе соотношения степеней допуска пользователей и грифов конфиденциальности ролей, или на основе соотношения разрешенных тематик пользователей и тематики ролей.

3. Управление доступом в системе осуществляется на основе введения следующих функций:

$f_{\text{user}}: C \rightarrow U$ – значением функции $u = f_{\text{user}}(c)$ является пользователь $u \in U$, осуществляющий данный сеанс c работы с системой;

$f_{\text{roles}}: C \rightarrow \mathcal{R}$ – значением функции $\mathcal{R} = f_{\text{roles}}(c)$ является набор ролей $\mathcal{R} \subseteq \mathcal{R}$ из доступных пользователю, по которым пользователь работает (осуществляет доступ) в данном сеансе $c \in C$;

$f_{\text{permissions}}: C \rightarrow P$ – значением функции $P = f_{\text{permissions}}(c)$ является набор полномочий $P \subseteq P$, доступных по всем ролям, задействованным пользователем в данном сеансе $c \in C$;

4. Основное правило (критерий безопасности) ролевого доступа определяется следующим образом.

Правило 2.4.1. Система функционирует безопасно, если и только если любой пользователь $u \in U$, работающий в сеансе $c \in C$, может осуществлять действия (операции, процедуры) в рамках полномочия $p \in P$, при условии:

$$p \in \mathcal{P},$$

где $\mathcal{P} = f_{\text{permissions}}(c)$.

Нетрудно видеть, что основной акцент в процессах организации и управления доступом при ролевой политике заключается в особенностях отображения множества пользователей на множество ролей $\mathcal{F}_{U\mathcal{R}}$ и ограничений, накладываемых на функцию авторизации $f_{\text{roles}}(c)$ пользователя в данном сеансе с разрешенными ему отношением $\mathcal{F}_{U\mathcal{R}}$ ролями. Выражаясь предметным языком, можно так сформулировать основные вопросы организации ролевого доступа:

Сколько и каких ролей может быть назначено для работы с системой одному пользователю?

Сколько и какие роли может одновременно задействовать один пользователь в одном сеансе работы с системой?

Еще одним существенным обстоятельством являются возможные отношения между ролями, в том, числе возможная передача (делегирование) полномочий и прав от одних ролей другим ролям.

В зависимости от особенностей разрешения данных вопросов выделяют несколько разновидностей ролевых моделей:

- с иерархической организацией системы ролей;
- с взаимоисключающими на любые (все) сеансы ролями (модель статического распределения обязанностей);
- с взаимоисключающими на один сеанс ролями (модель динамического распределения обязанностей);
- с количественными ограничениями по ролям;
- с группированием ролей и полномочий.

Приведем краткую характеристику указанных разновидностей ролей.

Иерархическая система ролей

Данная разновидность ролевых моделей является наиболее близкой к реальным организационно-технологическим и организационно-управленческим схемам на предприятиях и в организациях. Должности сотрудников предприятий, организаций в большинстве случаев образуют иерархически подчиненные структуры. На рис. 2.26 представлены примеры иерархической системы ролей-должностей.

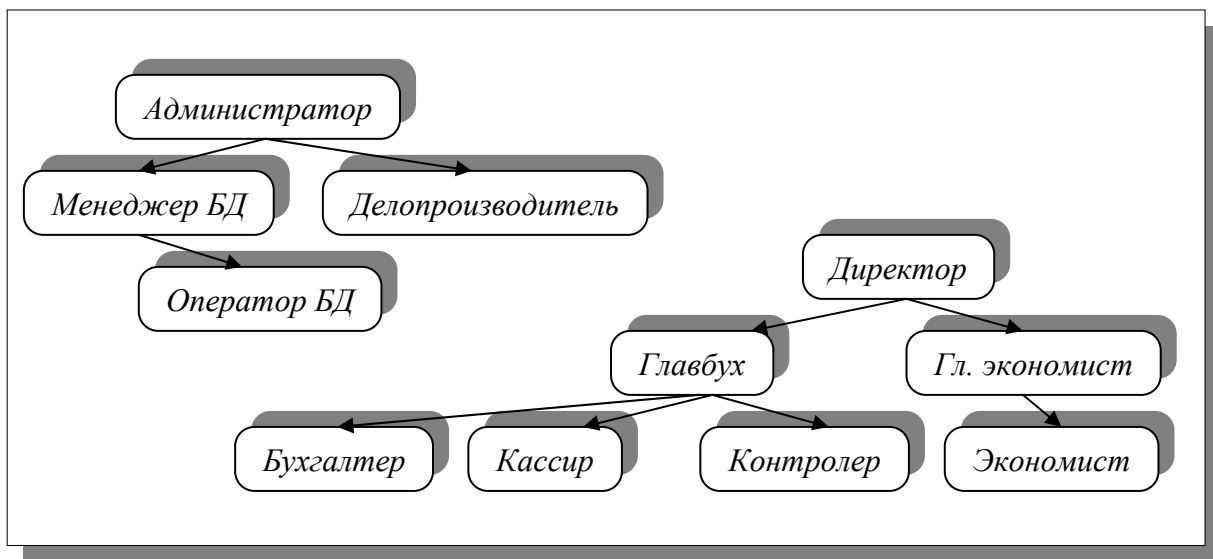


Рис. 2.26. Примеры иерархической организации системы должностей (ролей)

При этом помимо управленческого аспекта подчиненность ролей в большинстве случаев включает *наследование* прав и полномочий. В иерархически организованных структурах возможны два направления наследования полномочий и прав – "снизу" и "сверху".

При наследовании "сверху" подчиненный субъект помимо своих индивидуальных (так называемых явных) прав и полномочий получает (наследует) права и полномочия родителей. Подобный подход широко применяется в организации доступа к объектам, образующим иерархически организованную структуру, а также в системах индивидуально-группового доступа.

Модели с иерархически организованными ролями основаны на механизме наследования "снизу", при котором старшая в иерархии роль получает (владеет) права и полномочия непосредственно подчиненных ролей и т. д.

С формальной точки зрения модель с иерархической системой ролей включает введение дополнительных и уточнение исходных отношений сущностей и функций системы –

$F_{\mathcal{R}\mathcal{R}}: \mathcal{R} \times \mathcal{R}$ – отношение частичного порядка на множестве ролей, определяющее иерархию (подчиненность) ролей и задающее оператор доминирования ролей $\geq$, такое что:

если для $\rho_1, \rho_2 \in \mathcal{R}$, $\rho_1 \geq \rho_2$, то ρ_1 находится в иерархии ролей выше, чем ρ_2 ;

$F^h_{U\mathcal{R}}: U \times \mathcal{R}$ – отображение множества пользователей на множество ролей, причем вместе с каждой ролью ρ из набора назначенных для пользователя ролей, в него включаются и все роли ρ' , подчиненные ρ :

для $\forall \rho, \rho' \in \mathcal{R}, u \in U$: $\rho \geq \rho' \wedge (u, \rho) \in F^h_{U\mathcal{R}}(u) \Rightarrow (u, \rho') \in F^h_{U\mathcal{R}}(u)$;

$f_{\text{roles}}^h : C \rightarrow \mathcal{R}$ – значением функции является набор ролей $\mathcal{R} \subseteq \mathcal{R}$, из доступных пользователю, по которым пользователь работает (осуществляет доступ) в данном сеансе $c \in C$, с учетом иерархически подчиненных ролей:

$$f_{\text{roles}}^h(c) \subseteq \{\rho_i \mid (\exists \rho' \geq \rho_i \ (f_{\text{user}}(c), \rho') \in F_{U\mathcal{R}}^h(u))\} ;$$

$f_{\text{permissions}}^h : C \rightarrow \mathcal{P}$ – значением функции $\mathcal{P} = f_{\text{permissions}}^h(c)$ является совокупность полномочий $\mathcal{P} \subseteq \mathcal{P}$, доступных по всем ролям $\mathcal{R} = f_{\text{roles}}^h(c)$, задействованным пользователем в данном сеансе $c \in C$ (с учетом полномочий подчиненных ролей).

В плане критического анализа модели с иерархической организацией системы ролей отметим, что в более строгом смысле необходимо регламентировать возможные типы отношений $F_{P\mathcal{R}}$, т. е. особенности отображения множества полномочий на множество иерархически организованных ролей. В частности, принципиальное значение имеет вопрос о теоретико-множественном соотношении полномочий старшей роли и ролей, непосредственно ей подчиненных. Другим принципиальным вопросом в этом же плане является возможность или невозможность назначения одного и того же полномочия одновременно двум ролям, находящимся в иерархическом подчинении.

Анализируя особенности прав и полномочий соподчиненных должностей в реальных организационно-управленческих схемах, можно предложить три подхода к структуре отношения $F_{P\mathcal{R}}$ в системе с иерархически организованной системой ролей:

- строго таксономический листовый подход;
- не таксономический листовый подход;
- иерархически охватный подход.

При строго таксономическом листовом подходе все множество полномочий разбивается на непересекающиеся подмножества, назначаемые отношением $F_{P\mathcal{R}}$ листовым ролям корневого графа иерархии ролей (см. рис. 2.27):

$$F_{P\mathcal{R}}^h(\rho^{\Pi_j}) = \{p_{j_1}, p_{j_2}, \dots\} ,$$

$$F_{P\mathcal{R}}^h(\rho^{\Pi_j}) \cap F_{P\mathcal{R}}^h(\rho^{\Pi_i}) \cap \dots = \emptyset ,$$

$$F_{P\mathcal{R}}^h(\rho^{\Pi_j}) \cup F_{P\mathcal{R}}^h(\rho^{\Pi_i}) \cup \dots = \mathcal{P} .$$

Полномочия старших ролей, т. е. ролей, которым соответствуют внутренние вершины корневого графа иерархии ролей, определяются как объединение прав и полномочий непосредственно подчиненных ролей, которым в графе иерархии ролей соответствуют вершины-сыновья:

$$F_{P\mathcal{R}}^h(\rho^{\Pi_k}) = F_{P\mathcal{R}}^h(\rho_{k_i}) \cup F_{P\mathcal{R}}^h(\rho_{k_j}) \cup \dots ,$$

где $\{\rho_{k_i}, \rho_{k_j}, \dots\}$ – полный набор ролей-сыновей для роли ρ_k^I .

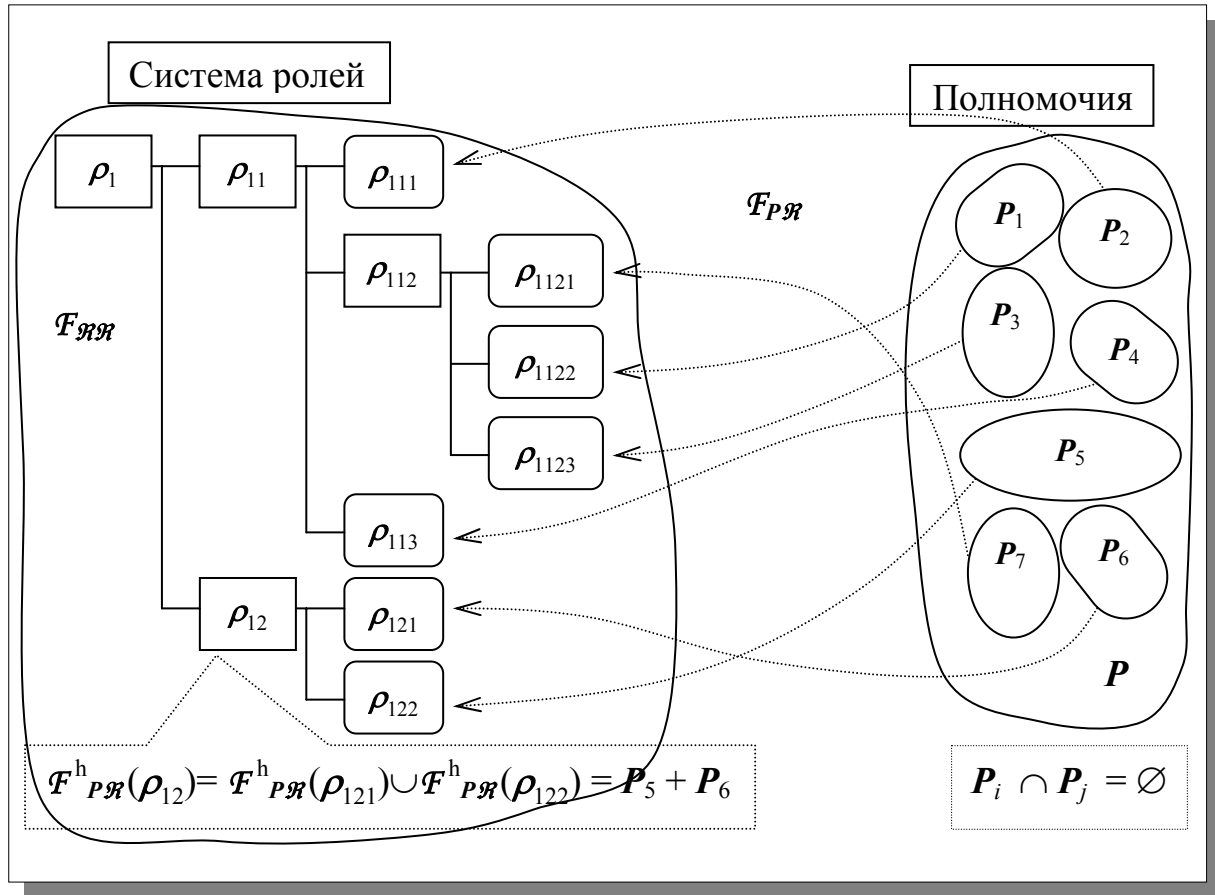


Рис. 2.27. Пример таксономического листового подхода к назначению полномочий в иерархической системе ролей

При нетаксономическом листовом подходе права и полномочия непосредственно получают также только листовые роли. Полномочия старших ролей также образуются объединением полномочий непосредственно подчиненных ролей. При этом, однако, пересечение наборов полномочий листовых ролей-братьев, т. е. подчиненных общей старшей роли, может быть не пустым:

$$F^h_{PЖ}(\rho^I_j) \cap F^h_{PЖ}(\rho^I_i) \cap \dots \neq \emptyset.$$

Нетрудно видеть, что подобный подход в отличие от строго таксономического не обеспечивает автоматического наделения корневой в иерархии роли полным набором полномочий системы.

При иерархически охватном подходе в набор прав и полномочий старших ролей непосредственно включаются только те права и полномочия, которые не вошли в наборы прав и полномочий подчиненных ролей:

$$F^h_{P\mathcal{R}}(\rho_k) = \{p_{k_1}, p_{k_2}, \dots\},$$

$$F^h_{P\mathcal{R}}(\rho^I_k) \cap F^h_{P\mathcal{R}}(\rho_i) = \emptyset,$$

где $\{\rho^I_k \geq \rho_i\}$.

При этом в соответствии с функциями f^h_{roles} и $f^h_{permissions}$ при иерархически охватном подходе итоговый набор полномочий пользователя, действовавшего в сеансе s старшую роль ρ^I_k , включает набор прав и полномочий всех ролей, подчиненных роли ρ^I_k .

Таксономический листовой подход отражает те предметные области, для которых набор полномочий (прав) рассматривается как общий ресурс, распределяемый между низовыми звеньями системы, к примеру, сельскохозяйственные земельные угодья, наделяемые крепостным крестьянам, из совокупности наделов которых складываются владения феодалов. Другой пример – структуры со строгой организационно-управленческой вертикалью – право руководить рядовыми сотрудниками имеют только их непосредственные руководители (нижнее управленческое звено); руководители более высокого управленческого уровня право непосредственного руководства рядовыми сотрудниками получают на основе объединения соответствующих прав подчиненных руководителей более низкого управленческого звена.

Второй подход идеологически сходен с первым, за исключением того, что конкретное право, полномочие может предоставляться не одной листовой роли, а сразу нескольким.

Иерархически охватный подход отражает более сложные, но и более распространенные предметные области, в которых на множестве прав и полномочий, в свою очередь, может быть задано отношение частичного порядка, что приводит к тому, что часть полномочий может быть присвоена ролям только определенного уровня иерархии. В подобных системах старшей роли (должности) непосредственно назначаются только права и полномочия соответствующего уровня иерархии, а права и полномочия подчиненных ролей наследуются и применяются автоматически.

Взаимоисключающие роли (статическое распределение обязанностей)

Особенностью некоторых предметных областей является запрет на предоставление определенного набора прав или полномочий в совокупности одному работнику. Подобные ограничения призваны усилить безопасность путем уменьшения вероятности злоумышленных действий по наборам некоторых связанных критических процессов и процедур. Если данные процедуры могут выполняться только разными работниками, то для осуществления злоумышленных действий потребуется сговор соответ-

вующих работников. Вероятность подобного (злоумышленного) развития ситуации в случае, когда все соответствующие права и полномочия имеет один человек существенно выше¹.

В формальном плане данный подход требует отображения множества ролей на множество подмножеств "несовместимых" ролей. При реализации подобного подхода положения и спецификации ролевой модели дополняются специальной функцией $f_{\text{exclusive}}(\rho)$, которая для каждой роли задает набор несовместимых с ней ролей.

С учетом возможной несовместимости ролей на отображение $\mathcal{F}_{U\mathcal{R}}$ накладывается следующее ограничение:

$$\text{если } (u, \rho) \in \mathcal{F}_{U\mathcal{R}}(u) \wedge \rho' \in f_{\text{exclusive}}(\rho) \Rightarrow (u, \rho') \notin \mathcal{F}_{U\mathcal{R}}(u) .$$

Если трактовать роль пользователя в КС как аналог служебной должности, то вырожденным, но повсеместно распространенным вариантом данного подхода является запрет назначения одному пользователю более двух ролей (нельзя исполнять сразу две должности по основному месту работы на постоянной основе).

Заметим также, что с формальной точки зрения запрет назначения пользователям взаимоисключающих ролей касается не собственно пользователей, а их учетных записей в КС. В результате часть элементов контрольно-ограничительных процедур в данной модели лежит вне компьютерной сферы, в частности, в запрете одному пользователю иметь несколько различных учетных записей в КС.

Взаимоисключающие роли (динамическое распределение обязанностей)

Во многих организационно-управленческих и организационно-технологических структурах работникам приходится совмещать выполнение определенных групп обязанностей или подменять других работников на определенное время. В этом случае распределение функциональных или служебных обязанностей имеет динамический характер и организуется, скажем, на каждый конкретный рабочий день, смену или иной временной управленческо-технологический период. При этом часть полномочий, групп процедур также может иметь критический с точки зрения безопасности характер в плане одновременного их выполнения одним работником.

В ролевой модели для разрешения подобных критических ситуаций может применяться механизм взаимоисключающих на один сеанс ролей. Как и в предыдущем случае, множество ролей отображается на множество подмножеств "несовместимых" ролей на основе использования функции $f_{\text{exclusive}}(\rho)$. При этом, однако, каких-либо ограничений на отношение $\mathcal{F}_{U\mathcal{R}}$,

¹ В этом плане можно привести пример запрета занятия одним лицом одновременно трех высших должностей в советском государстве – высшей партийной должности, высшей должности в исполнительной власти и высшей должности в законодательной ветви власти.

т. е. запретов на назначение ролей одному пользователю, в том числе, и взаимоисключающих ролей, не накладывается.

Решение проблемы безопасности осуществляется через запрет на взаимодействие в одном сеансе одним пользователем сразу нескольких взаимоисключающих ролей, скажем одновременно роли кассира-оператора и контролера-аудитора в финансовых процедурах. В формальном плане подобные ограничения обеспечиваются функцией f_{roles} , которая в каждом сеансе для конкретного пользователя задает активизированный им набор ролей с учетом ограничений на взаимоисключающие роли:

$$\forall \rho_1, \rho_2 \in \mathcal{R}, \rho_1 \in f_{\text{roles}}(c) \wedge \rho_2 \in f_{\text{exclusive}}(\rho_1) \Rightarrow \rho_2 \notin f_{\text{roles}}(c).$$

В качестве примера подобного подхода отметим повсеместно практикуемый запрет на одновременное использование в одном сеансе работы с КС ролей "администратора" и "аудитора безопасности" в КС.

Ролевой метод динамического распределения обязанностей при правильной организации ролей относительно включения в них процедур, вызывающих те или иные информационные потоки, может в значительной степени усилить мандатные модели разграничения доступа. В частности, таким образом можно организовать контроль и исключение некоторых потенциально опасных ситуаций, например, осуществление доступа к ценной (категорированной) информации и одновременный запуск недоверенных программ и процессов.

Количественные ограничения по ролям

Отдельным направлением идеологии исключения потенциально опасных ситуаций, связанных с набором критических по отношению к безопасности системы прав и полномочий, является возможность наложения количественных ограничений на права и полномочия, агрегируемые одной отдельно взятой ролью. В этом случае ограничения вводятся на отношение $\mathcal{F}_{\mathcal{P}\mathcal{R}}$ через введение функции $f^{\text{cardinality}}: \mathcal{P} \rightarrow \mathcal{N}$, определяющей для каждой роли количество включенных в нее прав и полномочий¹:

$$\text{для } \forall p_k \mid \{ \rho_m \mid (p_k, \rho_m) \in \mathcal{F}_{\mathcal{P}\mathcal{R}}(p_k) \} \mid \leq f^{\text{cardinality}}(p_k).$$

В других ситуациях требуется предоставление определенных полномочий как можно меньшему или строго ограниченному количеству сотрудников, скажем допуск к определенным опасным работам, право ознакомления с определенной информацией и т. п. Для реализации подобных ограничений в ролевой модели вводится функция $f^{\text{cardinality}}: \mathcal{R} \rightarrow \mathcal{N}$, определяющая число пользователей, получивших назначения на данную роль, и на этой основе устанавливаются ограничения на отношение $\mathcal{F}_{\mathcal{U}\mathcal{R}}$:

¹ где $\mathcal{N}$ – множество натуральных чисел.

для $\forall \rho_k \mid \{u_m \mid (\rho_k, u_m) \in F_{U\mathcal{R}}(u_m)\} \mid \leq f^{\text{cardinality}}(\rho_k)$.

Количественные ограничения могут в значительной степени облегчить администрирование систем, где требуется особо тщательный контроль за распределением прав и ответственности в коллективе пользователей.

Группирование ролей и полномочий

В некотором смысле противоположным по отношению к количественным ограничениям является группирование ролей и полномочий.

По смыслу и определению роль представляет группирование прав и полномочий в отдельную логически обособленную их совокупность, имеющую самостоятельное значение в предметной области КС. Поэтому, прежде всего, устанавливается механизм, контролирующий агрегирование в одну роль некоторых логически связанных прав и полномочий. Для этого на множестве полномочий P вводится функция $f^{\text{prerequisite}}: P \rightarrow \mathcal{P}$, устанавливающая для каждого права (полномочия) p набор $\mathcal{P} \subseteq P$ логически связанных прав и полномочий, и введение следующего ограничения на отношение $F_{P\mathcal{R}}$:

для $\forall (p, \rho) \in F_{P\mathcal{R}}(p) \wedge p' \in f^{\text{prerequisite}}(p) \Rightarrow (p', \rho) \in F_{P\mathcal{R}}(p)$.

Вместе с тем, могут наблюдаться такие ситуации, когда самостоятельное значение имеют и отдельные подгруппы среди более общей логически обособленной группы прав и полномочий. В этом случае создается группа логически связанных ролей, каждая из которых может выполняться только в совокупности с другими ролями данной группы. В этих ситуациях назначение подобной роли для пользователя может иметь смысл только тогда, когда он одновременно получает и другие роли из соответствующего логически связанного набора ролей. Для реализации подобного подхода на множестве ролей $\mathcal{R}$ вводится функция $f^{\text{prerequisite}}: \mathcal{R} \rightarrow \mathcal{R}$, определяющая для любой роли ρ набор $\mathcal{R} \in \mathcal{R}$ логически связанных с ней ролей, и устанавливается соответствующее ограничение на отношение $F_{U\mathcal{R}}$:

для $\forall (u, \rho) \in F_{U\mathcal{R}}(u) \wedge \rho' \in f^{\text{prerequisite}}(\rho) \Rightarrow (u, \rho') \in F_{U\mathcal{R}}(u)$.

В заключение рассмотрения разновидностей ролевых моделей в плане их критического анализа заметим, что помимо необходимости отмеченной ранее дополнительной регламентации отображения $F_{P\mathcal{R}}$ при иерархической структуре системы ролей, можно было бы по аналогии с тематико-иерархическим подходом ввести понятие *мультиролей*, и на этой основе более строго регламентировать назначение пользователям иерархически

организованных ролей, а также механизмы взаимоисключающих ролей, количественных ограничений и группирования ролей.

В практическом применении могут использоваться комплексные подходы, сочетающие все рассмотренные разновидности ролевых моделей, что позволяет существенно упростить проектирование и администрирование систем разграничения доступа для КС, автоматизирующих сложные, нетривиальные организационно-технологические и организационно-управленческие схемы и процессы. Вместе с тем, в ролевых моделях нет строгих доказательств безопасности системы в соответствии с определенными формализованными критериями. Поэтому их безопасность основывается на контрольных механизмах дискреционных или мандатных моделей, средствами которых регулируется доступ ролевых субъектов к объектам системы.

2.4.3. Индивидуально-групповое разграничение доступа

Технологии рабочих групп пользователей, лежащие в основе систем индивидуально-группового разграничения доступа, представляют упрощенную, но наиболее широко применяемую разновидность ролевой политики разграничения доступа.

Основные положения модели индивидуально-группового разграничения доступа сводятся к следующему.

1. КС представляется совокупностью следующих наборов сущностей:

- множества объектов доступа $O(o_1, o_2, \dots, o_M)$;
- множества пользователей $U(u_1, u_2, \dots, u_N)$;
- множества рабочих групп пользователей $G(g_1, g_2, \dots, g_K)$;
- множества прав доступа и привилегий¹ $R(r_1, r_2, \dots, r_J)$;
- матрицей доступа A размерностью $((N+K) \times M)$, каждая ячейка которой специфицирует права доступа и привилегии пользователей или их рабочих групп к объектам из конечного набора прав доступа и привилегий $R(r_1, r_2, \dots, r_J)$, т. е. $A[u, o] \subseteq R$, $A[g, o] \subseteq R$.

Определение 2.4.2. Рабочей группой называется совокупность пользователей КС, объединенных едиными правами доступа к объектам и (или) едиными привилегиями (полномочиями) выполнения определенных процедур обработки данных.

Нетрудно видеть, что рабочая группа представляет упрощенный аналог некоторой роли, которая формируется "снизу", т. е. на основе агрегирования в одну общую сущность пользователей, потребности в доступе которых к объектам системы подобны или близки. С учетом того, что полное совпадение потребностей пользователей в правах доступа к объектам системы в большинстве случаев маловероятно, у пользователей сохраняются и индивидуальные права доступа, которые не охватываются правами рабочей группы.

2. Групповые отношения в системе устанавливаются отображением множества пользователей на множество рабочих групп:

$F_{UG}: U \times G$ – такое, что одна рабочая группа объединяет нескольких пользователей, а один пользователь может входить в несколько рабочих групп.

Выражаясь языком моделей организации данных, можно отметить, что между сущностями "Рабочие группы" и сущностями "Пользователи" устанавливаются связи "Многие-ко-Многим". Данное обстоятельство обусловлено тем, что если рабочие группы рассматривать как агрегирование полномочий для выполнения пользователями определенных функциональ-

¹ Напомним, что как уже отмечалось в п. 2.3, под привилегией понимается наделение пользователей правом выполнения определенных процедур над всеми объектами системы или их определенной частью, например, привилегия владельцев объектов заключается в праве изменения списков доступа соответствующих объектов.

ных задач (обязанностей), то в реальных ситуациях один пользователь последовательно или на постоянной основе может выполнять сразу несколько задач (обязанностей), и ему тем самым необходимы права сразу нескольких рабочих групп. С другой стороны, сходные задачи могут решаться сразу несколькими пользователями, и, отсюда, одна рабочая группа объединяет нескольких пользователей¹.

Заметим также, что отношение $\mathcal{F}_{UG}$ обеспечивает второй этап организации ролевого разграничения доступа. При этом, однако, в отличие от классической ролевой политики, в технологии рабочих групп разделение процесса функционирования системы на *сеансы* не является принципиальным, ввиду того, что пользователь, входя в систему, всегда приобретает помимо своих индивидуальных прав одновременно и права доступа всех рабочих групп, в которые он включен по отношению $\mathcal{F}_{UG}$.

3. Функционирование системы индивидуально-группового разграничения доступа основывается на введении и использовании следующих функций:

$f_{\text{groups}}: U \rightarrow G$ – значением функции $f_{\text{groups}}(u) = \mathcal{G}$ является набор рабочих групп $\mathcal{G} = \{g_{u_1}, g_{u_2}, \dots\} \subseteq G$, в которые пользователь u включен по отображению $\mathcal{F}_{UG}$;

$f_{\text{users}}: G \rightarrow U$ – значением функции $\mathcal{U} = f_{\text{users}}(g)$ является набор пользователей $\mathcal{U} = \{u_{g_1}, u_{g_2}, \dots\} \subseteq U$, которые рабочая группа g включает по отношению $\mathcal{F}_{UG}$.

В практическом плане реализация функций f_{groups} и f_{users} производится посредством построения бинарной матрицы "Пользователи-Рабочие Группы", ячейки которой заполняются при установлении отношения $\mathcal{F}_{UG}$.

4. Управление индивидуально-групповым доступом в системе осуществляется на основе следующего правила (критерия безопасности) индивидуально-группового доступа.

Правило 2.4.2. Система функционирует безопасно, если и только если любой пользователь $u \in U$ по отношению к любому объекту $o \in O$ может осуществлять доступ с правами $\mathcal{R}$, не выходящими за пределы совокупности индивидуальных прав $A[u, o]$ и прав рабочих групп $A[g^{(u)}_{i}, o]$, в которые пользователь входит по отношению $\mathcal{F}_{UG}$:

$$\mathcal{R} \subseteq \{A[u, o] \cup A[g_{u_1}, o] \cup A[g_{u_2}, o] \cup \dots\},$$

где $\{g_{u_1}, g_{u_2}, \dots\} = f_{\text{groups}}(u)$.

Таким образом, права доступа пользователя к объекту складываются из его индивидуальных прав и прав всех рабочих групп, в которые он включен по отношению $\mathcal{F}_{UG}$.

¹ В большинстве практических систем явного запрета на вырожденные рабочие группы, т. е. в которые не входит ни один пользователь, не производится.

При этом следует отметить, что рабочие группы в полном смысле слова не являются ролями, так как, хотя и представлены в матрице доступа отдельными строками, тем не менее, не реализуются в виде отдельных и самостоятельно действующих субъектов доступа КС. В практическом плане это означает, что в КС не инициализируются отдельные субъекты доступа (процессы, запущенные программы, приложения и т. п.), действующие от имени рабочих групп. Иначе говоря, в системе индивидуально-группового доступа действуют только субъекты пользователей, права которых в процессах доступа к объектам определяются на основе совокупности индивидуальных и групповых прав.

Заметим также, что в приведенной спецификации модели индивидуально-группового доступа, как и в других разновидностях ролевых моделей, нет доказательства безопасного функционирования системы при выполнении правил разграничения доступа (в данном случае *правила 5.2*). Поэтому нетрудно видеть, что в смысле механизмов безопасности модель индивидуально-группового доступа полностью идентична дискреционным моделям, основанным на матрице доступа.

Содержание и, если так можно выразиться, сила моделей индивидуально-группового доступа, определившие их чрезвычайно широкое распространение, в другом. Через агрегирование субъектов доступа (пользователей) в рабочие группы удается существенно упростить проектирование (синтез) и управление (анализ) системы дискреционного разграничения доступа в ситуациях с большим количеством пользователей и большим количеством объектов доступа, т. к. в дискреционных системах, напомним, требуется отдельное и явное "прописывание" в матрице доступа (в списках доступа объектов) разрешений на доступ для любой пары "субъект(пользователь)-объект" и по любому методу доступа (чтение, запись и т. д.). Поэтому предоставление пользователям прав доступа к объектам через рабочие группы позволяет существенно уменьшить количество индивидуальных назначений, упростить, упорядочить проектирование и анализ системы коллективного доступа к ресурсам КС.

Как и в классическом ролевом подходе, дополнительными аспектами организации системы разграничения доступа являются возможные *отношения на множестве прав и отношения на множестве рабочих групп*.

В рамках теоретико-множественной модели прав и привилегий (множество **R**) все конкретные права доступа и привилегии (элементы множества **R**) равнозначны и независимы. Однако в реальности это не всегда так. К примеру, право *Modify* "сильнее", т. е. охватывает права *Write* и *Read*. В большинстве случаев можно считать, что отношения "силы" (охватности) прав (методов) доступа рефлексивны, антисимметричны и транзитивны. В формальном плане это означает, что на множестве **R** устанавливается отношение частичного порядка:

$\mathcal{F}_{RR}: R \times R$ - отношение, определяющее относительную силу (охватность) одних прав доступа по отношению к другим правам и задающее оператор доминирования $\geq$ такое, что

если для $r_1, r_2 \in R$, $r_1 \geq r_2$, то право (метод доступа) r_1 сильнее, чем r_2 , т. е. охватывает, включает потоки информации, вызываемые методом доступа r_2 .

В результате простое объединение наборов прав при использовании *правила 5.2* дополняется следующим ограничением:

$$\forall r_1, r_2 \in R, r_1 \geq r_2 \wedge r_2 \in \mathcal{R} \Rightarrow r_1 \in \mathcal{R},$$

где $\mathcal{R} = \{A[u, o] \cup A[g_{u_1}, o] \cup A[g_{u_2}, o] \cup \dots\};$

u и o – пользователь и объект доступа, соответственно;

$\{g_{u_1}, g_{u_2}, \dots\} = f_{\text{groups}}(u)$ – набор рабочих групп, в которые включен пользователь u .

Теоретико-множественное объединение индивидуальных и групповых прав доступа с учетом данного ограничения будем обозначать $\cup^<$. Таким образом, итоговый набор прав субъекта-пользователя в системе индивидуально-группового разграничения доступа по *правилу 2.4.2* определяется следующим соотношением:

$$\mathcal{R} = \{A[u, o] \cup^< A[g_{u_1}, o] \cup^< A[g_{u_2}, o] \cup^< \dots\}.$$

Существенным фактором при определении прав доступа, складывающихся по нескольким возможностям доступа (например, по индивидуальным и групповым назначениям; по прямым назначениям к объекту и по наследуемым правам доступа к объекту от контейнера), является обеспечение в определенных случаях гарантированного непредоставления (гарантированного запрета) определенному пользователю определенного права доступа к определенному объекту. Если права доступа определяются на основе одного варианта, скажем только на основе индивидуальных назначений без использования рабочих групп и иных агрегаций, то непредоставление прав доступа эквивалентно отсутствию разрешения на доступ.

В случае же индивидуально-группового доступа при отсутствии индивидуального права на доступ к объекту или права на доступ к объекту по какой-либо одной рабочей группе, пользователь может получить нежелательный доступ к данному объекту по другой рабочей группе. При этом такие ситуации могут быть вполне обоснованными, так как для всех остальных членов группы, по которой пользователь получает нежелательный доступ, данные права являются обоснованными и необходимыми, а исключить пользователя из данной группы также нельзя, так как он при этом может потерять набор других необходимых ему прав.

Разрешение данной проблемы осуществляется двумя способами:

- введением механизма ограниченного членства пользователя в группе;

- введением специфического права доступа, именуемого "*запрет доступа*".

Первый способ основывается на введении для каждого члена рабочей группы индивидуального "фильтра" наследования групповых прав. При этом, однако, данный способ, предоставляя возможность гибкой настройки индивидуально-групповых прав доступа на различные организационно-технологические и управленческие схемы, тем не менее, для решения проблемы гарантированного непредоставления прав требует просмотра и анализа всех групповых назначений, в которые включен данный пользователь.

Для вводимого же права "*запрет доступа*" устанавливается самый высокий приоритет в отношении частичного порядка на множестве прав доступа R . В результате проблема гарантированного непредоставления права доступа решается автоматически. Если среди индивидуальных либо групповых назначений пользователя к объекту имеется явный запрет на доступ, то все остальные назначенные права отвергаются.

Во многих практических системах допускается включение одних рабочих групп в другие, иначе говоря, членами рабочих групп кроме пользователей могут быть и коллективные пользователи, т. е. другие рабочие группы.

Ясно, что с учетом сущности рабочих групп, как объединений пользователей, подобные отношения должны быть исключительно транзитивными. Иначе говоря, ситуации, когда группа A входит в группу B , группа B входит в группу C , а группа C входит в группу A , недопустимы и бессмысленны. Ясно также, что отношения рабочих групп должны быть антисимметричны – если группа A входит в группу B , то группа B , в свою очередь, не может входить в группу A . Кроме того, смыслу группы как объединению пользователей не противоречит утверждение, что группа входит сама в себя, иначе говоря, отношения групп рефлексивны.

Отсюда следует, что на множестве рабочих групп G можно устанавливать отношение

$\mathcal{F}_{GG}: G \times G$ - отношение частичного порядка, определяющее иерархию (вложенность) рабочих групп и задающее оператор доминирования $\geq$ такое, что

если для $g_1, g_2 \in G$, $g_1 \geq g_2$, то g_1 включает g_2 .

Соответственно для управления доступом в системах с вхождением одних рабочих групп в другие требуется введение и использование дополнительной функции

$f_{\text{groups}}^h: G \rightarrow \mathcal{G}$ – значением функции $f_{\text{groups}}(g)$ является набор рабочих групп $\{g_{g_1}, g_{g_2}, \dots\} \subseteq G$, в которые рабочая группа g включена по отношению $\mathcal{F}_{GG}$.

В практическом плане реализация функции f_{groups}^h совместно с функциями f_{groups} и f_{users} может производиться на основе расширения бинарной матрицы "Пользователи-Группы" строками, соответствующими рабочим группам системы, т. е. построением матрицы "(Пользователи+Группы)-Группы".

Частичный порядок на множестве групп, если так можно выразиться, "слабее" частичного порядка на множестве иерархической системы ролей-должностей, так как одна рабочая группа может входить сразу в несколько рабочих групп, часть рабочих групп может не входить ни в какие другие группы и не включать в себя другие группы. Это означает, что системы с иерархически организованными ролями описываются графом в виде корневого дерева, а система рабочих групп графом произвольного вида, в том числе и с иерархически организованными фрагментами – см. рис. 2.28. Главной особенностью графа рабочих групп должно быть отсутствие циклов, порождаемых цепочками транзитивности.

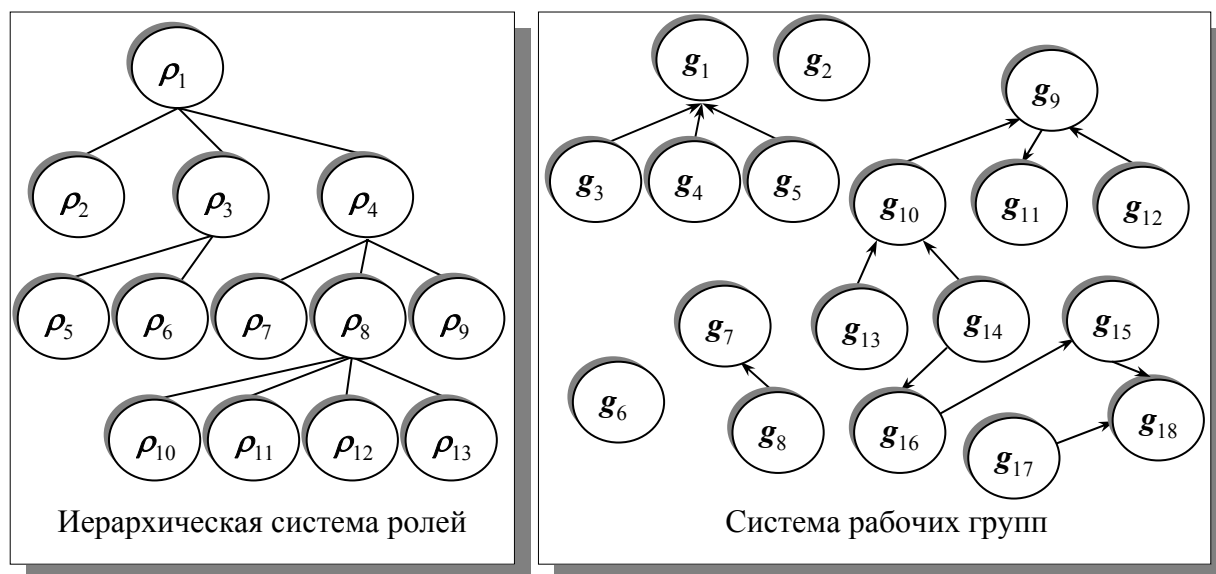


Рис. 2.28. Пример организации системы ролей и рабочих групп

В практических системах проблема транзитивности отношений в системе взаимосвязанных рабочих групп решается двумя способами:

- построением графа включения групп, поиском и исключением (разрывом) из него циклов (см. п. 9.2);
- разделением множества групп на два подмножества – группы, которые могут входить в другие группы, но состоять только из пользователей, т. е. не могут содержать другие группы; и группы которые могут состоять как из пользователей, так и из групп, но сами входить в другие группы не могут.

Как правило, при втором способе группы первого вида не имеют своих собственных групповых прав и привилегий, а используются лишь для того, чтобы комплектовать рабочие группы второго вида "групповым"

способом. Тем самым, обеспечивается структура, описываемая двудольным графом, все пути в котором имеют длину, равную единице, принципиально исключая на этой основе возможность транзитивных связей, порождающих циклы¹ (см. рис. 2.29).

Из самого понятия рабочих групп как объединений сущностей однотипного доступа следует механизм наследования прав и полномочий "сверху", что также принципиально отличает технологию рабочих групп от ролевой модели. Данный механизм наследования можно выразить следующим правилом определения прав доступа и привилегий рабочих групп пользователей.

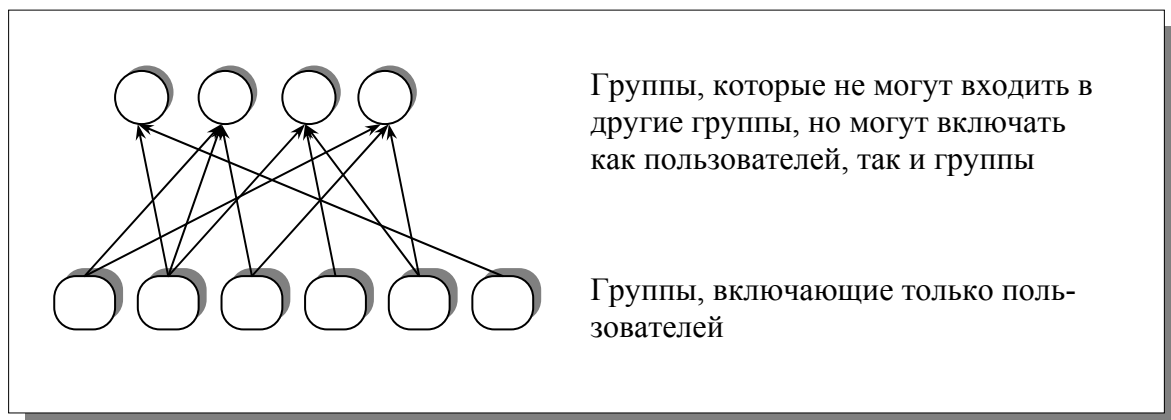


Рис. 2.29. Двудольная система рабочих групп

Правило 2.4.3. Полный набор прав и полномочия $\mathcal{R}_g$ рабочей группы пользователей g к объекту o определяются совокупностью набора прав и привилегий $A[g, o]$, непосредственно предоставленных данной группе, и набором прав и привилегий рабочих групп $A[g_{g_i}, o]$, в которые по отношению $\mathcal{F}_{GG}$ может входить данная рабочая группа:

$$\mathcal{R}_g(o) \subseteq \{A[g, o] \cup^< A[g_{g_1}, o] \cup^< A[g_{g_2}, o] \cup^< \dots\},$$

где $\{g_{g_1}, g_{g_2}, \dots\} = f_{\text{groups}}^h(g)$.

Соответственно, критерий безопасности индивидуально-группового доступа необходимо скорректировать, включая в итоговый набор прав пользователя права его рабочих групп с учетом правила 4.3:

$$\mathcal{R} \subseteq \{A[u, o] \cup^< \mathcal{R}_{g_{u_1}}(o) \cup^< \mathcal{R}_{g_{u_2}}(o) \cup^< \dots\},$$

¹ Минимальная длина пути для образования цикла в графе с наличием между двумя вершинами более одной дуги равна двум, в графе с наличием между двумя вершинами только одной дуги – трем.

где $\mathcal{R}_{g_i}(o)$ – полный прав по *правилу 5.3* рабочей группы g_i , в которую входит пользователь u .

В заключение отметим, что модель индивидуально-группового разграничения доступа обладает существенными выразительными возможностями при организации коллективного доступа к ресурсам в сложных системах с большим количеством пользователей и большим количеством сложно организованных и многочисленных ресурсов. Вместе с тем, ее использование помимо слабости защитных свойств дискреционного принципа разграничения доступа, на котором она основывается, таит и некоторые другие "подводные" камни, такие как избыточность и дублирование в предоставлении прав доступа пользователей к объектам системы. Кроме того, в ряде случаев проектирование системы рабочих групп представляет нетривиальную задачу, и требует применения дополнительных теоретико-графовых алгоритмов и пространственно-векторных моделей.

2.5. АВТОМАТНЫЕ И ТЕОРЕТИКО-ВЕРОЯТНОСТНЫЕ МОДЕЛИ НЕВЛИЯНИЯ И НЕВЫВОДИМОСТИ

2.5.1. Понятие и общая характеристика скрытых каналов утечки информации

Одной из, пожалуй, самых трудно решаемых проблем безопасности в КС, в том числе и основанных на моделях мандатного доступа, является проблема *скрытых каналов утечки информации*.

Определение 2.5.1. *Скрытым¹ каналом утечки информации называется механизм, посредством которого в КС может осуществляться информационный поток (передача информации) между сущностями в обход политики (правил) разграничения доступа².*

На первый взгляд данное определение может показаться странным с точки зрения того, что любая модель разграничения доступа должна предусматривать, в том числе, формальное доказательство невозможности "опасных" доступов (недопустимых информационных потоков). Вместе с тем, как рассматривалось при анализе моделей дискреционного и мандатного доступа, подобное доказательство производится с точки зрения определенного критерия безопасности. В частности, в моделях дискреционного доступа таким критерием является невозможность осуществления доступа субъектов к объектам вне явных разрешений, "прописанных" в матрице доступа. В моделях мандатного доступа недопустимыми являются потоки "сверху вниз" – от сущностей с высоким уровнем безопасности к сущностям более низких уровней безопасности путем нарушения правил **NRU** и **NWD**.

Таким образом, к числу скрытых каналов утечки информации относятся рассмотренные ранее потоки за счет "тройных программ" и неявные информационные потоки в системах на основе дискреционных моделей.

Применительно к моделям мандатного доступа понятие скрытого канала утечки информации можно конкретизировать следующим образом.

Определение 2.5.2. *Скрытым каналом утечки информации в системах мандатного доступа называется механизм, посредством которого может осуществляться передача информации от сущностей с высоким уровнем безопасности к сущностям с низким уровнем безопасности без непосредственного нарушения правил **NRU** и **NWD**.*

Суть скрытых каналов заключается в том, что в определенных случаях информацию можно получить или передать и без непосредственного

¹ В литературе используется также термин "тайный канал утечки информации".

² А.А.Грушо представлено определение информационного потока, включающее понятие скрытых каналов утечки информации на основе теоретико-множественного представления информации (множество слов в некотором языке).

осуществления операций *read/write* к объектам, в частности на основе рассмотрения и анализа определенных процессов и параметров системы. Например, если по правилу **NRU** нельзя читать секретный файл, но можно "видеть" его объем, то высокоуровневый субъект, меняя по определенному правилу объем секретного файла, может таким кодированным образом передавать секретную информацию низкоуровневому объекту. Аналогичным образом информация от высокоуровневых субъектов может передаваться по количеству создаваемых или удаляемых секретных файлов, получить доступ по чтению к которым низкоуровневые субъекты не могут, но "видеть" их наличие, и, соответственно, определять их количество могут.

Другие возможности "тайной" передачи информации могут основываться на анализе временных¹ параметров протекания процессов в КС. Без гарантий полноты классификации скрытые каналы утечки информации можно разделить на три вида:

- скрытые каналы по памяти (на основе анализа объема и других статистических параметров объектов системы);
- скрытые каналы по времени (на основе анализа временных параметров протекания процессов системы);
- скрытые статистические каналы (на основе анализа статистических параметров процессов системы²).

Требования по перекрытию и исключению скрытых каналов впервые были включены в спецификацию уровней защиты КС, предназначенных для обработки сведений, составляющих гостайну в США в "Оранжевой книге".

¹ Ударение на букве "ы".

² Данный вид скрытых каналов введен А.А. Грушо.

2.5.2. Модели информационного невмешательства и информационной невыводимости

Теоретические основы подходов к решению проблемы скрытых каналов базируются на работах Д. Денинга, исследовавшего принципы анализа потоков данных в программном обеспечении и принципы контроля совместно используемых ресурсов. Отталкиваясь от идей Денинга Дж. Гогеном, Дж. Мезигером был предложен автоматный подход к исследованию вопросов скрытых каналов, относящийся к интерфейсу защищенных КС.

Другой подход к анализу скрытых каналов утечки информации базируется на теоретико-информационной интерпретации процессов функционирования КС в идеологии *информационной невыводимости* и *информационного невмешательства*.

Существо данного подхода заключается в отказе от рассмотрения процесса функционирования КС как детерминированного процесса. При рассмотрении моделей конечных состояний (HRU, TAKE-GRANT, Белла-ЛаПадулы) предполагалось, что функция перехода в зависимости от запроса субъекта и текущего состояния системы однозначно определяет следующее состояние системы. Нетрудно заметить, что в системах коллективного доступа (много пользователей, много объектов) переходы, следовательно, и состояния системы, обуславливаются большим количеством самых разнообразных, в том числе и случайных факторов, что делает *процесс функционирования системы вероятностным*.

При таком допущении политика мандатного доступа требует определенной модификации и, в частности, теоретико-вероятностной трактовки процессов функционирования КС и опасных информационных потоков. Приведем основные положения теоретико-информационных моделей.

1. КС рассматривается как совокупность двух непересекающихся множеств сущностей:

- множества высокоуровневых объектов H ;
- множества низкоуровневых объектов L .

Говоря иными словами, КС представляется мандатной системой с решеткой, состоящей всего из двух уровней безопасности – высокого и низкого, и, соответственно, с невозможностью обычных (*read/write*) информационных потоков "сверху вниз".

2. Состояние любого объекта $h \in H$ или $l \in L$ является случайным. Понятие информационной невыводимости основывается на следующем определении "опасных" потоков:

Определение 2.5.3. В системе присутствует информационный поток от высокоуровневых объектов к низкоуровневым, если некое возможное

значение переменной в некотором состоянии низкоуровневого объекта невозможно одновременно с определенными возможными значениями переменных состояний высокоуровневых объектов.

Нетрудно видеть, что данное определение на основе вероятностной трактовки расширяет понятие обычных опасных потоков "сверху вниз" и характеризует без конкретизации механизма каналы скрытой утечки информации. Низкоуровневые объекты, анализируя определенные параметры своих состояний, могут делать выводы об определенных параметрах состояний высокоуровневых объектов, получая от них тем самым высокоуровневую информацию.

Состояние, задаваемое *определением 3.20*, можно выразить следующим соотношением:

$$\text{если } p(h) > 0, p(l) > 0, \text{ то и } p(h|l) > 0, \quad (2.5.1)$$

где $p(h)$, $p(l)$ – безусловные вероятности состояний, соответственно, высокоуровневого и низкоуровневого объектов;
 $p(h|l)$ – условная вероятность состояния высокоуровневого объекта при определенных значениях состояния низкоуровневого объекта.

3. Формулируется следующий критерий информационной невыводимости:

Определение 2.5.4. Система безопасна в смысле информационной невыводимости, если в ней отсутствуют информационные потоки вида, задаваемого определением 2.5.3.

Анализ условий обеспечения критерия информационной невыводимости в виде соотношения (2.5.1) показывает, что его требования являются чрезвычайно жесткими и достижимы, в частности, при полной изоляции высокоуровневых объектов от низкоуровневых. Действительно, при $p(h) > 0$, $p(l) > 0$ справедливо следующее соотношение:

$$p(l|h) = p(h, l)/p(h) = p(h|l)p(l)/p(h), \quad (2.5.2)$$

откуда с учетом того, что $p(h|l) > 0$, следует, что и $p(l|h) > 0$.

Неравенство нулю условной вероятности $p(l|h)$ означает, что требование отсутствия выводимости высокоуровневой информации на основе анализа состояний низкоуровневых объектов, одновременно приводит и к обратному, т. е. отсутствию возможностей выводимости низкоуровневой информации из анализа состояний высокоуровневых объектов. Данное свойство является избыточным и противоречит основным положениям мандатной политики, а именно – неопасности и допустимости потоков "снизу вверх" от низкоуровневых сущностей к сущностям с более высокими уровнями безопасности.

Нетрудно видеть, что полная изоляция разноуровневых сущностей системы является тривиальным и далеко не всегда приемлемым решением

проблемы безопасности, так как существенно ограничивает функциональные возможности КС.

Другой подход, основывается на идеологии информационного невмешательства.

4. Понятие опасных потоков в идеологии информационного невмешательства трактуется в следующем смысле:

Определение 2.5.5. В системе присутствует информационный поток от высокоуровневых объектов к низкоуровневым, если информация (состояние) низкоуровневых объектов зависит от информации высокоуровневых объектов.

Условия определения 2.5.5 можно отразить следующим соотношением:

$$p(l|h) = p(l) . \quad (2.5.3)$$

Отсюда, однако, нетрудно показать, что при $p(h)>0$, $p(l)>0$ данное соотношение равносильно соотношению

$$p(h|l) = p(h) . \quad (2.5.4)$$

Включение в рассмотрение временного аспекта позволяет соотношения (2.5.3) и (2.5.4) представить в следующем виде:

$$p(l_t|h_{t-1}) = p(l_t) , \quad (2.5.5)$$

$$p(h_t|l_{t-1}) = p(h_t) , \quad (2.5.6)$$

где h_t , l_t – переменные, характеризующие состояния высокоуровневых и низкоуровневых объектов в момент времени t .

Неформальная интерпретация соотношений (2.5.5) и (2.5.6) заключается в том, что на состояние высокоуровневых объектов $p(h_t)$ в текущий момент времени не влияет состояние (и, следовательно, информация) низкоуровневых объектов в предшествующий момент времени $p(l_t)$, и наоборот. Иначе говоря, разноуровневые объекты не имеют возможности влиять на последующие состояния объектов другого (не своего) уровня.

Вместе с тем, анализ процессов функционирования КС показывает, что соотношения (2.5.5) и (2.5.6) являются также чрезвычайно жесткими, фактически совпадающими с требованиями полной изоляции разноуровневых сущностей. Выполнение соотношений (2.5.5) и (2.5.6) может вступать в определенных ситуациях в противоречие с требованиями обеспечения безопасности информации не только в смысле конфиденциальности, но и в смысле ее законной доступности, сохранности. В частности, для обеспечения сохранности информации в КС может использоваться журнал фиксации текущих изменений всех объектов системы с целью обеспечения возможностей восстановления информации при сбоях и разрушениях. Ясно,

что данный объект должен быть высокоуровневым, а его состояние должно зависеть от информации в предшествующий момент времени как высокоуровневых, так и низкоуровневых объектов. Более того, из данного примера видно, что с точки зрения безопасности информации могут быть обоснованными и обратные ситуации, когда значение низкоуровневого объекта в текущий момент времени определено значением высокоуровневого объекта в предыдущий момент времени, в частности, при восстановлении из высокоуровневого журнала изменений КС значения низкоуровневого файла после сбоя или разрушения последнего.

Поэтому *определение 2.5.5* опасных потоков в смысле информационного невмешательства смягчается более приемлемым, как с точки зрения безопасности, так и с точки зрения реальностей по обеспечению функциональности КС, следующим определением.

Определение 2.5.6. В системе присутствует информационный поток от высокоуровневых объектов к низкоуровневым, если имеется возможность накопления информации в низкоуровневых объектах о значениях высокоуровневых объектов.

С формальной стороны *определение 2.5.6* означает то, что совместный анализ значений состояний низкоуровневого объекта в текущий момент времени l_t и в предшествующий l_{t-1} не давал бы никакой новой информации о состоянии высокоуровневого объекта в предыдущий момент времени h_{t-1} :

$$p(l_t|h_{t-1}, l_{t-1}) = p(l_t|l_{t-1}) . \quad (2.5.7)$$

Нетрудно показать, что из соотношения (2.5.7) вытекает следующее равносильное соотношение:

$$p(h_{t-1}|l_t, l_{t-1}) = p(h_{t-1}|l_{t-1}) , \quad (2.5.8)$$

которое выражает запрет обратных информационных потоков из l_t в h_{t-1} , но и незапрещенность потоков из l_t в h_{t+1} .

Таким образом, высокоуровневые объекты могут принимать информацию о состоянии низкоуровневых объектов в предыдущие моменты времени, что обеспечивает существенно более широкие возможности для построения функциональных КС.

На основе понятия информационных потоков в смысле информационного невмешательства по определению 2.5.6 вводится следующий критерий безопасности.

Определение 2.5.7. Система безопасна в смысле информационного невмешательства, если в ней отсутствуют информационные потоки вида, задаваемого определением 2.5.5, т. е. выполняется следующее соотношение:

$$p(l_t|h_t, l_t) = p(l_t|l_t) , \quad (2.5.9)$$

где $t, t' = 0, 1, 2, \dots$, и $t' < t$.

Таким образом, модели информационного невмешательства и информационной невыводимости обеспечивают теоретико-методологическую базу для поиска и разработки программно-технических решений, в частности, в системах представления информации и интерфейса КС для нейтрализации скрытых каналов утечки информации.

2.5.3. Нейтрализация скрытых каналов утечки информации на основе технологий "представлений" и "разрешенных процедур"

Идеи информационной невыводимости и информационного невмешательства послужили теоретической основой широко применяемых для разграничения доступа в современных КС технологий "представлений" (views) и "разрешенных процедур". Эти технологии исторически возникли как программно-технологические решения разграничения доступа в СУБД и впоследствии получили распространение на другие разновидности программного обеспечения компьютерных систем.

Введем следующее определение.

Определение 2.5.8. *"Представлением" информации КС называется процедура формирования и предоставления именованному пользователю после его входа в систему и аутентификации необходимого ему подмножества информационных объектов КС, в том числе, с возможным их количественным и структурным видоизменением исходя из задач разграничения доступа к информации.*

Таким образом, в технологиях "представления" пользователи, входя и работая в системе, "видят" и оперируют не с реальной, "настоящей", а с виртуальной системой, формируемой индивидуально для каждого пользователя. В результате задача разграничения доступа решается автоматически и, можно сказать, в буквальном смысле. Проблемы безопасности при этом сводятся исключительно к скрытым каналам утечки информации, рассмотрение и нейтрализация которых осуществляется на основе анализа условий и процедур, обеспечивающих выполнение критериев безопасности по *определениям 2.5.4 и 2.5.7.*

Нетрудно видеть, что техника "представлений" автоматически решает проблему скрытых каналов утечки информации первого вида. В частности, если пользователю предоставлено право **read** к каталогу, в котором физически расположены файлы, к части которых пользователь не имеет никаких прав доступа, то, открывая свое "представление" каталога, пользователь "увидит" в нем только те объекты (файлы), к которым у него есть право доступа. Следовательно, получить какую-либо не предназначенную данному пользователю информацию посредством "наблюдения" за наличием, объемом и количеством недоступных объектов невозможно.

Техника "представлений" особенно развита и широко применяется в реляционных СУБД, где "представления" реализуются естественным образом на основе авторизованных с пользователями запросов на выборку данных, автоматически запускаемых при входе пользователя в систему и формирующих для него "свое" представление базы данных. Основные проблемы разграничения доступа при этом заключаются в том, чтобы пользо-

ватели не "почувствовали" и не получили каким-либо образом не предназначенную им и "невидимую" ими информацию, в том числе и *не вывели* информацию о наличии в системе недоступной им информации! Решение данных непростых вопросов и доказательство нейтрализации скрытых каналов в этих случаях основывается на использовании положений моделей информационной невыводимости и информационного невмешательства.

Вместе с тем, техника "представлений", хотя и является эффективным средством решения проблемы скрытых каналов утечки информации, тем не менее, не может перекрыть весь их спектр, в особенности, скрытые каналы второго и третьего вида, основывающиеся на анализе временных и статистических параметров процессов КС.

Часть из подобных проблем разграничения доступа связана с интерфейсом КС, а их разрешение основывается на технике "разрешенных процедур".

Определение 2.5.9. *"Системой разрешенных процедур" назовем разновидность интерфейса КС, когда при входе в систему аутентифицированным пользователям предоставляется только лишь возможность запуска и исполнения конечного набора логико-технологических процедур обработки информации без возможности применения элементарных методов доступа (**read, write, append, create, delete** и т. п.) к информационным объектам системы.*

Таким образом, в системах с интерфейсом "разрешенных процедур" идеология разграничения доступа доведена, если так можно выразиться, до крайности, т. е. пользователи не "видят" информационные объекты системы вовсе, а имеют возможность только лишь "нажимать кнопки" и выполнять операции на уровне логико-технологических процедур предметной области КС.

Нетрудно заметить, что компьютерная система при этом для пользователей превращается в дискретный автомат, получающий команды на входе и выдающий обработанную информацию на выходе.

Впервые подобный подход к представлению КС был рассмотрен Гогеном (J. Goguen) и Мезигером (J. Meseguer), предложившими **автоматную модель информационного невлиния** (невмешательства), подробное изложение которой приведено в работе – **GM-модель**.

В **GM-модели** компьютерная система рассматривается как детерминированный автомат, на вход которого поступает последовательность команд пользователей, разделяемых по мандатному принципу на два непересекающихся множества - группу высокоуровневых пользователей и группу низкоуровневых пользователей. В зависимости от команды пользователя на входе КС-автомат формирует реакцию – вывод системы, который "видит" пользователь на выходе. Критерием безопасности в системе является требование "несмешивания" ввода высокоуровневого пользователя с выводом низкоуровневого пользователя.

Для реализации критерия безопасности в модели **GM** вводится функция вывода (*out*), задающая вывод системы, в зависимости от команды пользователя, его уровня безопасности и предыстории (*traces*) команд ввода в системе. Если при формировании вывода для низкоуровневого пользователя в предыстории команд наблюдается смешивание, т. е. присутствуют вводы высокоуровневых пользователей, то посредством применения специальной процедуры очищения (*purge*) строится подправленная трассировка (предыстория) поведения системы, по которой функция *out* формирует вывод низкоуровневому пользователю.

В результате при проектировании систем с подобной идеологией интерфейса, в том числе систем "разрешенных процедур", необходимо рассмотреть все множество условий и спецификаций, имеющих отношение к возможному смешиванию вводов/выводов пользователей, и верифицировать на этой основе безопасность системы.

2.6. МОДЕЛИ И ТЕХНОЛОГИИ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ

2.6.1. Общая характеристика моделей и технологий обеспечения целостности данных

Как уже отмечалось, понятие целостности данных является фундаментальным в области компьютерной безопасности и включает ряд специфических свойств, связанных с таким объектом, как данные.

В информатике "**данные**" понимаются как *информация, представленная в формализованном виде, что обеспечивает возможность ее хранения, обработки и передачи*. Таким образом, данные характеризуются конкретной формой (способом) представления и неразрывно связаны с обработкой, хранением и передачей информации. К примеру, информация о сотруднике какой-либо организации с точки зрения ее хранения, обработки и передачи представлена *в виде данных* на учетной карточке кадрового подразделения с формализованной структурой (поля с определенными правилами заполнения).

Информация на стадии данных характеризуется определенной *формой представления* и дополнительной характеристикой, выражаемой термином "*структура*".

В результате, как уже отмечалось, под **целостностью** информации понимается *неискаженность, достоверность, полнота, адекватность и т.д. информации, т.е. такое ее свойство, при котором ее содержание и структура [данных] определены и изменяются только уполномоченными лицами и процессами*.

С учетом специфики используемых механизмов обеспечения целостности данных можно выделить три стороны этого понятия в отношении соответствующих видов данных в компьютерных системах.

Правильность. Заключается в отсутствии логических ошибок в структуре и ошибок в содержании (в значениях) данных при их обработке.

Неискаженность. Заключается в отсутствии подделки данных или возникновения ошибок в данных при их передаче в линиях связи, а также при хранении в компьютерных системах.

Неизменность. Заключается в тождественности данных определенному эталону.

В зависимости от превалирования того или иного аспекта целостности и области использования данных выделяют:

- специальные технологии разграничения доступа данных, обеспечивающие их целостность в смысле правильности, неискаженности и неизменности, основывающиеся на т.н. *моделях целостности данных*;

- криптографические технологии *электронной цифровой подписи*, обеспечивающие целостность данных в смысле их нескаженности при передаче в линиях связи и хранении в компьютерных системах;
- технологии параллельного *выполнения транзакций* в клиент-серверных системах (СУБД).

2.6.2. Дискреционная модель Кларка-Вильсона

Как и в дискреционных моделях разграничения доступа, обеспечивающих конфиденциальность данных, модель Кларка-Вильсона основывается на "тройках", только на "тройках целостности" — «**субъект-операция(транзакция), не нарушающая целостность-объект**».

Приведем основные положения модели.

1. Все множество объектов D разделяется на объекты CDI , требующие контроля целостности (*constrained data items*), и объекты UDI , не требующие контроля целостности (*unconstrained data items*).

$$D = CDI \cup UDI, \quad CDI \cap UDI = \emptyset$$

2. На множестве элементарных операций над объектами выделяются совокупности (последовательности), обособляющиеся в логически самостоятельные сущности, называемые процедурами преобразования TP (*transformation procedures*).
3. Дополнительно вводится особый класс процедур IVP над данными, которые обеспечивают проверку целостности контролируемых данных (*integrity verification procedures*).
4. Те процедуры преобразования данных TP , применение к результатам которых процедур проверки целостности IVP дает положительный результат, называются «*корректно (правильно, хорошо) сформированными транзакциями*».

Устанавливаются следующие **правила функционирования** системы.

- C1.** Множество всех процедур контроля целостности IVP должно содержать процедуры контроля целостности любого элемента данных из множества всех CDI .
- C2.** Все процедуры преобразования TP должны быть хорошо сформированными транзакциями, т.е. не нарушать целостности данных, и применяться только по отношению к списку элементов (объектов) CDI , устанавливаемых администратором системы.
- E1.** Система должна контролировать допустимость применения TP к элементам CDI в соответствии со списками, указанными в правиле **C2**.
- E2.** Система должна поддерживать список разрешенных конкретным пользователям процедур преобразования TP с указанием допустимого для каждой TP и данного пользователя набора обрабатываемых элементов CDI (т.е. тройки «**субъект- TP -объект CDI** »)

- С3.** Список, определенный правилом **С2**, должен отвечать требованию разграничения функциональных обязанностей (в т.ч. совместного выполнения).
- Е3.** Система должна аутентифицировать всех пользователей, пытающихся выполнить какую-либо процедуру преобразования **ТР**.
- С4.** Каждая **ТР** должна записывать в журнал регистрации информацию, достаточную для восстановления полной картины каждого применения этой **ТР**. Журнал регистрации – это специальный элемент **СДИ**, предназначенный только для добавления в него информации.
- С5.** Специальные **ТР** могут корректно обрабатывать **УДИ**, превращая их в **СДИ**.
- Е4.** Только специально уполномоченный субъект (пользователь) может изменять списки, определенные в правилах **С2** и **Е2**. Этот субъект не имеет права выполнять какие-либо действия, если он уполномочен изменять регламентирующие эти действия списки.

2.6.3. Мандатная модель Кена Биба

В содержательном и формальном (математическом) плане модель К.Биба является инверсией мандатной модели Белла-ЛаПадулы.

Ее основные положения сводятся к следующему.

- 1. Система защиты** представляет совокупность
 - множества субъектов **S**
 - множества объектов **O**
 - множества операций над объектами доступа **R** (два элемента - *read* и *write*)
 - решетки уровней безопасности **Λ** субъектов и объектов (решетка уровней целостности данных)
 - функции **F**, отображающей элементы множеств **S** и **O** в **Λ**
 - множества состояний системы **V**, которое определяется множеством упорядоченных пар **(F, A)**
 - начального состояния **v₀**
 - набора запросов **Q**
 - функции переходов **T: (V × Q) → V**, которая переводит систему из одного состояния в другое при выполнении запросов субъектов на доступ к объектам
- 2. Критерий безопасности (обеспечения целостности)** заключается в недопустимости опасных потоков «снизу вверх», т.к. такие потоки могут нарушить целостность объектов более высокого уровня безопасности ("загрязнить" объекты более высокого уровня целостности/чистоты).
- 3. Правила безопасности доступов** являются инверсными (обратными по отношению к соответствующим правилам в модели Белла-ЛаПадулы:

- запись данных субъектом $s \in S$ в объект $o \in O$ с более высоким уровнем безопасности возможна тогда и только тогда, когда $F(s) < F(o)$. Таким образом (**no write up - NWU**) – **нельзя писать вверх**, т.к. в результате может произойти нарушение целостности («загрязнение») объекта;

- чтение данных субъектом $s \in S$ из объекта $o \in O$ с более низким уровнем безопасности возможно тогда и только тогда, когда $F(o) < F(s)$. Таким образом (**no read down - NRD**) – **нельзя читать вниз**, т.к. в результате может произойти нарушение целостности («загрязнение») субъекта.

Соответственно, как и в модели Белла-ЛаПадулы, при данных правилах и ограничениях, накладываемых на функцию перехода, математически доказывается безопасность функционирования системы с т.зр. обеспечения целостности данных в рамках отмеченного выше критерия безопасности.

Основным критическим моментом в отношении модели К.Биба являются некоторые сомнения относительно адекватности и правомочности отображения такого свойства данных как "целостность" дискретно-упорядоченным множеством. Во многих случаях интуитивное представление о целостности данных является бинарным – есть ли целостность или нет. Хотя, справедливости ради следует признать, что если целостность данных рассматривать в смысле их "чистоты", то модель К.Биба является совершенно адекватной алгебраической структурой.

Как и в случае модели Белла-ЛаПадулы, рассматривают разновидности модели К.Биба. Когда по некоторым соображениям нельзя отказывать субъектам в возможности чтения "вниз", предусматривается автоматическое понижение уровня целостности ("чистоты") субъекта до уровня объекта. Аналогично, когда по некоторым соображениям субъектам нельзя отказывать в возможности записи "вверх", то предусматривается автоматическое снижение уровня целостности объекта (его "чистоты") до уровня целостности субъекта. И в том и другом случае безопасность функционирования системы в смысле отмеченного ранее критерия сохраняется.

При этом нетрудно увидеть, что системы с подобными расширениями модели К.Биба обладают, если так можно выразиться, тенденцией к деградации, или лучше сказать к "загрязнению", когда через некоторое время уровень целостности всех субъектов снизится до минимального, или во втором случае, до минимального может снизиться уровень целостности всех объектов доступа.

В практическом плане для создания защищенных компьютерных систем (и в смысле обеспечения конфиденциальности и в смысле обеспечения целостности данных) важным является объединение инверсных моделей Белла-ЛаПадулы и К.Биба.

Такое объединение может осуществляться в одном из трех вариантов.

-
1. На основе двух различных решеток. В таких системах у субъектов и объектов доступа две метки доступа – уровень конфиденциальности, и уровень целостности. Решение о безопасности доступа принимается одновременно по правилам модели Белла-ЛаПадулы и модели К.Биба. Нетрудно увидеть что, при таком подходе в организации доступа возможны тупики, когда по правилам модели Белла-ЛаПадулы доступ может быть разрешен, а по правилам модели К.Биба – нет. И наоборот.
 2. На основе одной общей решетки уровней безопасности (конфиденциальности/целостности). Опять-таки нетрудно видеть, что в таких системах разрешенными являются только доступы субъектов к объектам в пределах одного уровня безопасности (т.н. "равное чтение" и "равная запись").
 3. На основе одной общей решетки, но с двумя метками безопасности – по конфиденциальности и по целостности с противоположным характером их назначения/присвоения. Субъекты и объекты с высокими требованиями *конфиденциальности* располагаются на *высоких* уровнях иерархии решетки (секретные данные и доверенные по секретам пользователи). Субъекты и объекты с высокими требованиями *целостности* располагаются на *нижних* уровнях иерархии решетки (системное ПО и программисты).

Отметим, что, несмотря на сложность классификации субъектов и объектов доступа, именно третий вариант находит применение в современных компьютерных системах, в частности, в СУБД, где реализуется мандатная политика безопасности.

2.6.4. Технологии параллельного выполнения транзакций в клиент-серверных системах (СУБД)

Сердцевиной и основой *эффективности функционирования многопользовательских систем "Клиент-сервер"* является **эффективное управление транзакциями**. Собственно само понятие транзакции возникло именно в процессе исследования принципов построения и функционирования централизованных многопользовательских реляционных СУБД. Транзакции играют важную роль в *механизме обеспечения целостности* базы данных. Ограничения целостности непосредственно проверяются по завершению очередной транзакции. Если условия ограничений целостности данных не выполняются, то происходит "откат" транзакции (выполняется SQL-инструкция *ROLLBACK*), в противном случае транзакция фиксируется (выполняется SQL-инструкция *COMMIT*).

В парадигме клиент-серверных систем единичные действия пользователей с базой данных ассоциированы с транзакциями. В том случае, когда *от разных пользователей поступают транзакции*, время выполнения которых перекрывается, монитор транзакций обеспечивает специальную *технология их параллельного (параллельно-последовательного) выполнения и изоляции (устранения взаимного влияния и затруднений друг другу)* с тем, чтобы избежать нарушений согласованного состояния данных и других **издержек параллельной обработки данных**. К числу подобных издержек относятся:

- потерянные изменения;
- "грязные" данные;
- неповторяющиеся чтения.

Потерянные изменения возникают тогда, когда две транзакции одновременно *изменяют один и тот же объект базы данных*. В том случае, если в силу каких-либо причин, например, из-за нарушений целостности данных, происходит откат, скажем, второй транзакции, то вместе с этим отменяются и все изменения, внесенные в соответствующий период времени первой транзакцией. В результате первая еще не завершившаяся транзакция при повторном чтении объекта не "видит" своих ранее сделанных изменений данных. Очевидным *способом преодоления* подобных ситуаций является запрет изменения данных любой другой транзакцией до момента завершения первой транзакции — т.н. **блокировка** объекта.

"Грязные" данные возникают тогда, когда *одна транзакция изменяет* какой-либо объект данных, а *другая транзакция в этот момент читает данные из того же объекта*. Так как первая транзакция еще не завершена, и, следовательно, не проверена согласованность данных после проведенных, или вовсе еще только частично проведенных изменений, то вторая транзакция может "видеть" соответственно несогласованные, т.е. "гряз-

ные" данные. Опять-таки *способом недопущения* таких ситуаций, может быть *запрет чтения объекта* любой другой транзакцией, пока не завершена первая транзакция, его изменяющая.

Неповторяющиеся чтения возникают тогда, когда *одна транзакция читает* какой-либо объект базы данных, а *другая* до завершения первой его *изменяет* и успешно *фиксируется*. Если при этом первой, еще не завершенной, транзакции требуется повторно прочесть данный объект, то она "видит" его в другом состоянии, т.е. чтение не повторяется. Способом недопущения подобных ситуаций в противоположность предыдущему случаю является *запрет изменения* объекта любой другой транзакцией, когда первая транзакция на чтение еще не завершена.

Механизм изоляции транзакций и преодоления ситуаций несогласованной обработки данных в общем виде основывается на технике **сериализации транзакций**. План (способ) выполнения совокупности транзакций называется *сериальным*, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного их выполнения. Сериализацией транзакций в этом смысле является организация их выполнения по некоторому сериальному плану. Существуют два различных подхода сериализации транзакций:

- синхронизационные захваты (блокировки) объектов базы данных;
- временные метки объектов базы данных.

В первом подходе определяется два основных *режима захватов* – **совместный режим** (Shared) и **монопольный режим** (eXclusive). При совместном режиме осуществляется разделяемый захват, требующий только операций чтения. Поэтому такой захват еще называют *захватом по чтению*. При монопольном режиме осуществляется неразделяемый захват, требующий операций обновления данных. Такой захват, соответственно, еще называют *захватом по записи*.

Наиболее распространенным вариантом первого подхода является реализация **двухфазного протокола синхронизационных захватов (блокировок) объектов** базы данных – 2PL (Two-Phase Locks). В соответствии с данным протоколом выполнение транзакции происходит *в два этапа*. На *первом этапе (первая фаза)* перед выполнением любой операции транзакция запрашивает и накапливает захваты необходимых объектов в соответствующем режиме. После получения и накопления необходимых захватов (блокировок) осуществляется *вторая фаза* – фиксация изменений (или откат по соображениям целостности данных) и освобождение захватов.

При построении сериальных планов *допускается совмещение только захватов по чтению*. В остальных случаях, транзакции должны "ждать" когда необходимые объекты разблокируются (освободятся). Более изощренные стратегии сериализации транзакций основываются на **"гранулировании"** объектов захвата (файл базы данных, отдельная таблица, стра-

ница файла данных, отдельная запись–кортеж). Соответственно при этом *расширяется номенклатура синхронизационных режимов захватов*, например, в совместном режиме (по чтению) может быть захвачен и в целом файл и отдельные его страницы, или в другом случае обеспечивается совместный режим захвата файла с возможностью монопольного захвата отдельных его объектов – таблиц, страниц или записей–кортежей). Грануляция синхронизационных блокировок позволяет строить более эффективные планы сериализации транзакций.

Существенным *недостатком* синхронизационных захватов является возможность возникновения *тупиковых ситуаций* (Deadlock). Предположим, одна транзакция на первой фазе установила монопольный захват одного объекта, а другая транзакция монопольный захват второго объекта. Для осуществления полного набора своих операций первой транзакции еще требуется совместный захват второго объекта, а второй транзакции, соответственно, совместный захват первого объекта. Ни одна из транзакций не может закончить первую фазу, т.е. полностью накопить все необходимые захваты, т.к. требуемые объекты уже монопольно захвачены, хотя были свободны к моменту начала осуществления транзакций.

Непростой проблемой является *автоматическое обнаружение* (распознавание) таких *тупиковых ситуаций* и их разрешение (*разрушение*). Распознавание тупиков основывается на *построении и анализе графа ожидания транзакций*, состоящего из вершин–транзакций и вершин–объектов захвата. Исходящие из вершин–транзакций дуги к вершинам–объектам соответствуют требуемым захватам, а дуги, исходящие из вершин–объектов к вершинам–транзакциям соответствуют полученным захватам. При тупиковых ситуациях в графе ожидания транзакций наблюдаются петли (циклы), обнаружение которых обеспечивается специальным алгоритмизируемым механизмом редукции графа, подробное изложение которого можно найти в специальной литературе по теории графов. Отметим только, что применение подобного механизма распознавания тупиков требует построения и постоянного поддержания (обновления) графа ожидания транзакций, что *увеличивает накладные расходы* при выполнении транзакции и снижает производительность обработки данных.

Еще одной проблемой является технология, или лучше сказать, *алгоритм разрушения тупиков*. В общем плане такие алгоритмы основываются на выборе *транзакции–жертвы*, которая временно откатывается для предоставления возможности завершения выполнения операций другим транзакциям, что, конечно же, в определенной степени нарушает принцип изолированности пользователей. В качестве "жертвы" выбирается или самая "дешевая" транзакция в смысле затрат на выполнение, или транзакция с наименьшим *приоритетом*.

Более простой *альтернативой* технике синхронизационных захватов является *техника временных меток*. Суть этого метода заключается в

том, что каждой транзакции приписывается *временная метка, соответствующая моменту начала выполнения транзакции*. При выполнении операции над объектом транзакция "помечает" его своей меткой и типом операции (чтение или изменение). Если при этом другой транзакции требуется операция над уже "помеченным" объектом, то выполняются действия по следующему алгоритму:

- проверяется, не закончилась ли транзакция, первой "пометившая" объект;
- если первая транзакция закончилась, то вторая транзакция помечает его своей меткой и выполняет необходимые операции;
- если первая транзакция не закончилась, то проверяется конфликтность операций (напомним, что конфликтно любое сочетание, кроме "чтение—чтение");
- если операции неконфликтны, то они выполняются для обеих транзакций, а объект до завершения операций помечается меткой более поздней, т.е. более молодой транзакции;
- если операции конфликтны, то далее происходит откат более поздней транзакции и выполняется операция более ранней (старшей) транзакции, а после ее завершения, объект помечается меткой более молодой транзакции и цикл действий повторяется.

В результате того, что при таком алгоритме конфликтность транзакций определяется более грубо, чем при синхронизационных блокировках, реализация *метода временных меток* вызывает *более частые откаты транзакций*. Несомненным же *достоинством* метода временных меток является *отсутствие тупиков*, и, следовательно, отсутствие накладных расходов на их распознавание и разрушение.

2.7. МЕТОДЫ И ТЕХНОЛОГИИ ОБЕСПЕЧЕНИЯ ДОСТУПНОСТИ (СОХРАННОСТИ) ДАННЫХ

2.7.1. Резервирование, архивирование и журнализация данных

Как отмечалось важнейшей составной частью безопасности информации в компьютерных системах является *обеспечение правомерной доступности данных*, что достигается:

- отсутствием препятствий в правомерном доступе к данным (обеспечивается политикой и механизмами разграничения доступа);
- сохранностью файлов данных БД (обеспечивается профилактикой носителей данных, специальными организационными, программно-техническими и технико-технологическими мерами, именуемыми терминами "резервирование", "архивирование" и "журнализация" данных)
- восстановлением данных в случае программно-аппаратных сбоев, ошибочных действий пользователей либо действий злоумышленников, приводящих к уничтожению (потере, разрушению) файлов данных (обеспечивается резервированием/архивированием, журнализацией данных, специальными технологиями репликация данных).

Резервирование данных заключается в:

- организации и поддержании организационно-технологической системы создания и обновления (поддержания актуальности) копий файлов данных;
- установлении и поддержании режима размещения, хранения и использования (доступа) копий файлов данных для их восстановления в случае сбоев и разрушений.

Резервирование осуществляется либо средствами копирования файлов ОС, либо СУБД (специальными режимами работы СУБД или специальными утилитами СУБД). Особой разновидностью является т.н. "горячее" резервирование в виде постоянного и непрерывного функционирования 2-х или более равнозначных («зеркальных») копий БД в КС, относящихся к т.н. «системам реального времени», в которых критичным является обеспечение непрерывности функционирования. При этом все изменения данных одновременно и параллельно фиксируются в зеркальных копиях БД, при сбое одной копии функционирование КС обеспечивается другой «зеркальной» копией.

Архивирование представляет, по сути, другое название резервирования данных, поскольку из-за большого размера файлов данных (баз данных), создание резервных копий осуществляется с одновременным «сжатием» файлов данных. При этом сохраненная и «сжатая» копия БД называется «архивом» БД.

Журнализация данных, осуществляемая совместно с архивированием, т.е. с системой создания и периодического обновления архивов БД, за-

ключается в ведении специальных журналов текущих изменений данных для – а) аудита действий пользователей, и б) для восстановления актуального состояния БД из существующего архива и произведенных с момента его создания (последнего обновления) изменений данных.

Схематически суть процедур архивирования/журнализации для обеспечения сохранности данных приведена на рис. 2.30.

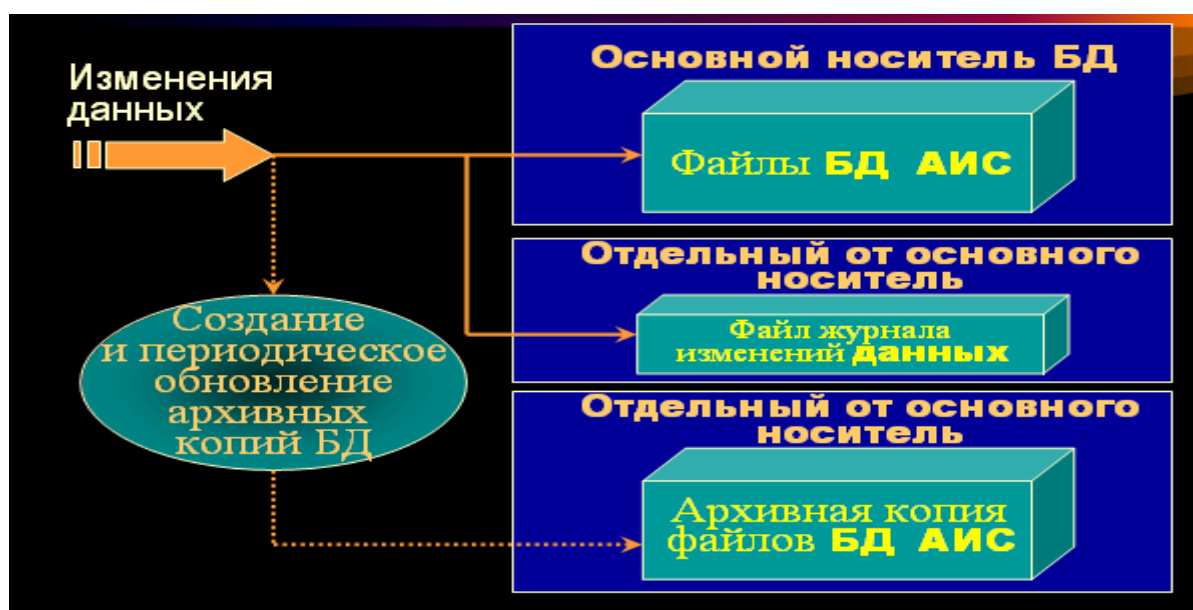


Рис. 2.30. Принцип организации процедур архивирования/журнализации данных

Восстановление актуального состояния данных на основе систем архивирования/журнализации представлено на рис. 2.31.

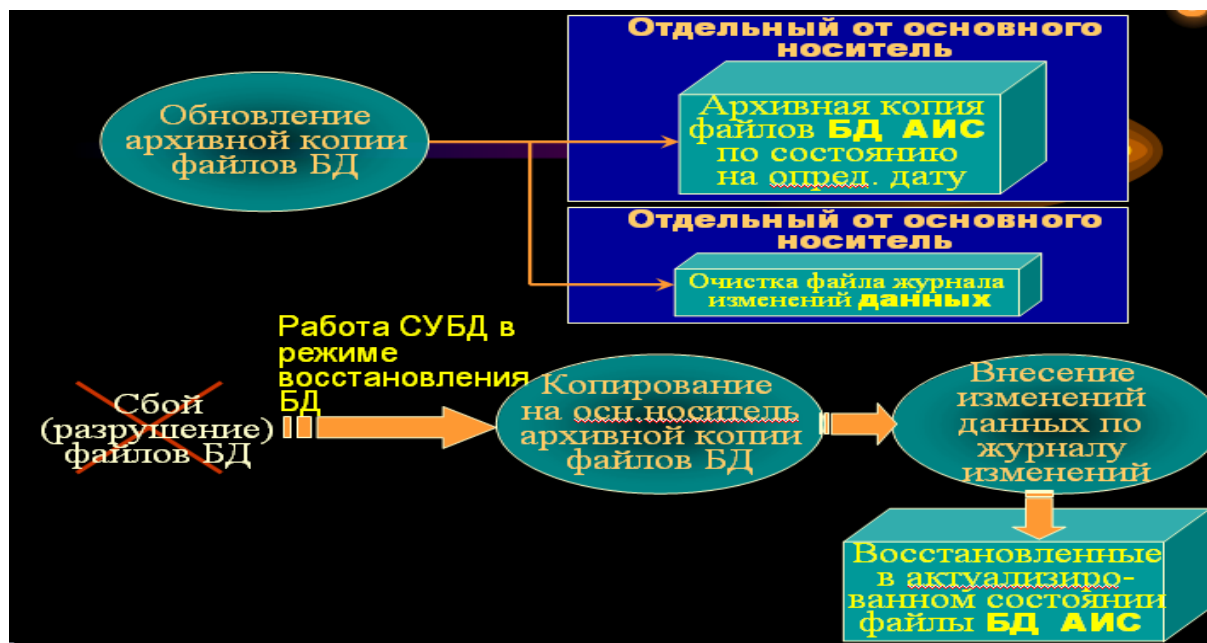


Рис. 2.31. Схема восстановления данных на основе архивирования/журнализации

При этом различают системы с синхронной журнализацией, когда транзакции (изменения данных) фиксируются в базе данных одновременно с фиксацией в журнале изменений данных, и системы асинхронной журнализации, когда изменения данных в журнал вносятся периодически (по мере накопления изменений).

2.7.2. Технологии репликации данных

Технологии репликации составляют в определенном смысле альтернативный клиент-серверным системам способ создания распределенных компьютерных систем, и, кроме того, обеспечивают дополнительные к процедурам архивирования/журнализации возможности обеспечения сохранности и восстановления данных.

Репликой называют особую копию базы данных для размещения на другом компьютере сети с целью автономной работы пользователей с одинаковыми (согласованными) данными общего пользования.

Основная идея репликации заключается в том, что пользователи работают автономно с одинаковыми (общими) данными, растиражированными по локальным базам данных, обеспечивая с учетом отсутствия необходимости передачи и обмена данными по сети **максимальную для своих вычислительных установок производительность**. Кроме того, при сбое данных на одной вычислительной установке имеется возможность восстановления данных из идентичных копий/реплик, располагаемых на других вычислительных установках. Программное обеспечение СУБД для реализации такого подхода соответственно дополняется *функциями тиражирования (репликации)* баз данных, включая тиражирование как самих данных и их структуры, так и системного каталога с информацией о размещении реплик, иначе говоря, с информацией о конфигурировании построенной таким образом распределенной системы.

При этом, однако, возникают *две проблемы* обеспечения одного из основополагающих принципов построения и функционирования распределенных систем, а именно – *непрерывности согласованного состояния данных*:

- обеспечение согласованного состояния во всех репликах количества и значений общих данных;
- обеспечение согласованного состояния во всех репликах структуры данных.

Обеспечение согласованного состояния общих данных, в свою очередь, основывается на реализации одного из *двух принципов*:

- *принципа непрерывного размножения обновлений* (любое обновление данных в любой реплике должно быть немедленно размножено);
- *принципа отложенных обновлений* (обновления реплик могут быть отложены до специальной команды или ситуации).

Принцип непрерывного размножения обновлений является основополагающим при построении так называемых "*систем реального времени*", таких, например, как системы управления воздушным движением, системы бронирования билетов пассажирского транспорта и т.п., где требуется не-

прерывное и точное соответствие реплик или других растиражированных данных во всех узлах и компонентах подобных распределенных систем.

Реализация принципа *непрерывного размножения обновлений* заключается в том, что *любая транзакция считается успешно завершенной, если она успешно завершена на всех репликах системы*. На практике реализация этого принципа встречает существенные затруднения, связанные с **тупиками**, как и в технологиях синхронизационных захватов систем "Клиент–сервер". Предположим, что на одной вычислительной установке пользователь обновляет данные в своей реплике. На время осуществления транзакции (транзакций) соответствующие записи в базе данных этой реплики ядром локальной СУБД заблокированы от изменения другими пользователями. Вместе с тем, транзакция может быть зафиксирована и, следовательно, разблокированы соответствующие данные только тогда, когда данная транзакция послана и также завершена на других репликах системы. Предположим также, что в другой реплике системы, находящейся на другом компьютере сети, в это же время другой пользователь проводит свои обновления (транзакции) с теми же записями, которые, естественно, в этот момент также заблокированы от изменений для других пользователей. Так образуется тупик. Одна транзакция не может быть зафиксирована в своей реплике, потому что заблокированы соответствующие записи в другой реплике. А разблокировка этих записей в другой реплике также невозможна до тех пор, пока не разблокируются соответствующие записи в первой реплике, т.е. когда завершится транзакция в первой реплике. Создается тупиковая ситуация.

Для *обнаружения* (распознавания) *тупиков* в реплицированных системах применяются *такие же алгоритмы*, которые были разработаны в *мониторах транзакций* централизованных систем "Клиент–сервер", – строится и поддерживается аналогичный граф ожидания транзакций, и применяются аналогичные алгоритмы распознавания и разрушения тупиков, основанные в основном на технике приоритетов.

В целом ряде предметных областей распределенных информационных систем режим реального времени с точки зрения непрерывности согласования данных не требуется. Такие системы автоматизируют те организационно–технологические структуры, в которых информационные процессы не столь динамичны. Если взять, к примеру, автоматизированную информационную систему документооборота, то традиционная "скорость" перемещения и движения служебных документов соответствует рабочему дню или в лучшем случае рабочим часам. В этом случае обновление реплик распределенной информационной системы, если она будет построена на технологии реплицирования, требуется, скажем, только лишь один раз за каждый рабочий час, или за каждый рабочий день.

Такого рода информационные системы можно строить на основе **принципа отложенных обновлений**. Накопленные в какой-либо реплике

изменения данных *специальной командой* пользователя направляются для обновления всех остальных реплик систем. Такая *операция* называется **синхронизацией реплик**. Возможность *конфликтов и тупиков* в этом случае при синхронизации реплик *существенно снижается*, а немногочисленные подобные конфликтные ситуации легко разрешить *организационными мерами*.

Решение *второй проблемы согласованности данных*, а именно – **согласованности структуры данных**, осуществляется через *частичное отступление*, как и в системах "Клиент–сервер", *от принципа отсутствия центральной установки* и основывается на **технике "главной" реплики**.

Суть этой техники заключается в том, что *одна из реплик* базы данных системы объявляется *главной*. При этом *изменять структуру базы данных можно только в главной реплике*. Эти изменения структуры данных *тиражируются на основе принципа отложенных обновлений*, т.е. через специальную *синхронизацию реплик*. Частичность отступления от принципа отсутствия центральной установки заключается в том, что в отличие от чисто централизованных систем, *выход из строя главной реплики не влечет сразу гибель всей распределенной системы*, так как остальные реплики продолжают функционировать автономно. Более того, на практике СУБД, поддерживающие технологию реплицирования, позволяют пользователю с определенными полномочиями (администратору системы) преобразовать любую реплику в главную и тем самым полностью восстановить работоспособность всей системы.

Процесс синхронизации реплик в современных СУБД включает обмен только теми данными, которые были изменены или добавлены в разных репликах. С этой целью в системном каталоге базы данных создаются специальные таблицы текущих изменений и организуется система *глобальной идентификации (именования) всех объектов распределенной системы*, включая раздельное поименование одинаковых объектов (вплоть до записей таблиц) в разных репликах¹. Такой подход несколько увеличивает объем базы данных, но позволяет существенно ограничить *транспортные расходы* на синхронизацию реплик.

Важным, с точки зрения гибкости и эффективности функционирования распределенных информационных систем, построенных на технологиях реплицирования, является возможность создания так называемых **частичных реплик** и включения в реплики как **реплицируемых**, так и **нереплицируемых объектов**. Частичной репликой называется *база данных, содержащая ограниченное подмножество записей полной реплики*. Распространенным способом создания частичных реплик является использование фильтров, устанавливаемых для конкретных таблиц полной (главной) реплики. Частичные реплики позволяют решить некоторые проблемы, связан-

¹ Т.н. техника глобальных уникальных идентификаторов (GUID).

ные с разграничением доступа к данным и повышают производительность обработки данных. Так, например, в реплику базы данных для определенного подразделения во многих случаях целесообразно реплицировать только те записи таблицы "Сотрудники", которые относятся к данному подразделению, исключив тем самым доступ к другим записям. Техника частичных реплик также снижает затраты на синхронизацию реплик, так как ограничивает количество передаваемых по сети изменений данных.

Возможность включения в реплики объектов базы данных, которые не подлежат репликации, позволяет *более гибко и адекватно настроить схему и прочие объекты БД (запросы, формы и отчеты) на специфику предметной области, особенности ввода данных и решаемые информационные задачи по конкретному элементу распределенной системы.*

2.8. ПОЛИТИКА И МОДЕЛИ БЕЗОПАСНОСТИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ

Рассмотренные ранее политики и модели безопасности неявно позиционировались в отношении компьютерной системы, представляющей единое целое¹, имеющей единое ядро и монитор безопасности. Вместе с тем архитектура и процессы функционирования реальных КС всегда характеризуются распределенностью.

2.8.1. Общая характеристика проблем безопасности в распределенных компьютерных системах

С точки зрения безопасности можно выделить три аспекта распределенности в компьютерных системах:

- для программного обеспечения современных КС характерен модульный принцип программно-технической реализации ядра системы и, в том числе, монитора безопасности, в результате чего отдельные функции безопасности *распределены между программно-техническими компонентами* (модулями) системы;
- в процессах управления доступом монитор безопасности использует ассоциированный с ним объект-данные, содержащий информацию по политике разграничения доступа в конкретной КС, который может реализовываться как отдельный (централизованный) информационный объект, или как *распределенная информационная структура*;
- за исключением такого вида, как отдельные изолированные автоматизированные рабочие места (АРМ), современные КС представляют собой системы, объединяющие *физически распределенные* вычислительные установки (вычислительные сети и системы).

Модульный характер ядра КС предопределяет проблему сопряжения защитных механизмов (идентификация/аутентификация, инициализация процессов-субъектов, управление памятью, управление доступом, аудит и т. д.), распределенных по отдельным программным (аппаратно-программным) модулям ядра системы. Проблема сопряжения распределенных защитных механизмов является чрезвычайно важной для обеспечения безопасности КС, но выходит за рамки тематики пособия

Если рассматривать ассоциированный с монитором безопасности объект, содержащий информацию по разграничению доступа субъектов к объектам системы, как некую базу данных, то можно выделить структурно-логическую и физическую распределенность. Структурно-логическая рас-

¹ В литературе иногда используются термины "монолитная система", система с заданным периметром и спецификациями.

пределенность определяется информационно-логической структурой (моделью организации данных) и конкретной моделью разграничения доступа. К примеру, в дискреционных моделях таким объектом может быть матрица доступа, которая реализуется как одна из системных таблиц в СУБД или как совокупность распределенных по объектам списков доступа в ОС. При этом, однако, логически распределенные элементы структуры ассоциированного с монитором безопасности объекта могут размещаться в пределах одной вычислительной установки или быть, к тому же, и физически распределенными по различным вычислительным установкам.

Физическая распределенность по различным вычислительным установкам ассоциированного с монитором безопасности объекта в большинстве случаев смыкается с физической распределенностью самой КС, т. е. с третьим аспектом распределенности, который требует уточнения и конкретизации самого понятия "Распределенная КС".

С учетом рассмотрения, прежде всего, политики разграничения доступа в дальнейшем изложении будем придерживаться следующей трактовки понятия "Распределенная КС".

Определение 2.8.1. Распределенной КС называется система, состоящая более чем из одного локального сегмента, представляющего обособленную совокупность субъектов и объектов доступа.

Нетрудно видеть, что центральным в понятии распределенной КС является способ (принцип, механизм и т. д.) обособления совокупности субъектов и объектов доступа в отдельную самостоятельную сущность, т. е. в локальный сегмент.

Выделим три способа (критерия) обособления подмножества субъектов и объектов в локальный сегмент:

- группирование некоторого подмножества субъектов доступа на основе их порождения и управления одним общим субъектом (системным процессом);
- локализация некоторого подмножества субъектов и объектов доступа в рамках некоторой технической компоненты КС;
- присвоение всем субъектам и объектам некоторого уникального идентификатора (адреса) в едином информационном (адресном) пространстве и разделение этого пространства на области, обособляющие локальные сегменты.

Первый способ характерен для выделения в сети компонентов (субъектов и объектов) АС, построенной в идеологии "Клиент-сервер", особенно для ее разновидности с "тонкими" клиентами. В подобных системах большинство процессов клиентов (субъектов доступа) порождаются (управляются, выполняются) главным системным процессом в виде сервера БД или сервера приложений. Соответственно вопросы безопасности в подобных распределенных системах обеспечиваются единым субъектом, функционирующим также на сервере системы. В частности, процессы кол-

лективного доступа к общим данным управляются и регламентируются специальной программной компонентой, именуемой *монитором транзакций*. Монитор транзакций совместно с **МБО** обеспечивает все три аспекта безопасности данных – конфиденциальность (через определенную политику разграничения доступа), целостность и доступность (через специальные протоколы совместного выполнения и фиксации транзакций).

Второй способ, наиболее характерный для ЛВС, основывается на привязке субъектов и объектов доступа или их общей с точки зрения безопасности и разграничения доступа информации (например, общей базы учетных записей) к одной или нескольким выделенным вычислительным установкам. Можно также отметить, что на практике подобным образом в большинстве случаев обособляется одна вычислительная установка или сегмент ЛВС.

Третий способ является в некотором смысле виртуальной альтернативой второму способу и применяется в большинстве случаев совместно с ним. В частности, подобным образом создаются виртуальные защищенные сети, физическое расположение отдельных компонент которых не имеет самостоятельного значения.

Принципиальным и наиболее существенным во всех трех подходах является то, что локализуемый сегмент с точки зрения безопасности и разграничения доступа рассматривается как монолитная (нераспределенная) КС. Таким образом, в дальнейшем будем опираться на следующее аксиоматическое предположение.

Предположение 2.8.1. В локальном сегменте КС функционирует единый для всех субъектов и объектов монитор безопасности, реализующий локальную политику безопасности (политику разграничения доступа).

Предметом политики разграничения доступа в распределенных компьютерных системах является рассмотрение принципов организации и механизмов доступа, в том числе и межсегментного, при котором реализовывалась бы некая общая политика безопасности, обеспечивающая как внутрисегментную, так и межсегментную безопасность.

2.8.2. Модели распределенных систем в процессах разграничения доступа

Анализ практических аспектов реализации распределенных КС позволяет выделить с точки зрения процессов безопасности две основные разновидности их архитектуры – см. рис. 2.32:

- система взаимодействующих сегментов;
- система "локальный сегмент - внешняя среда".

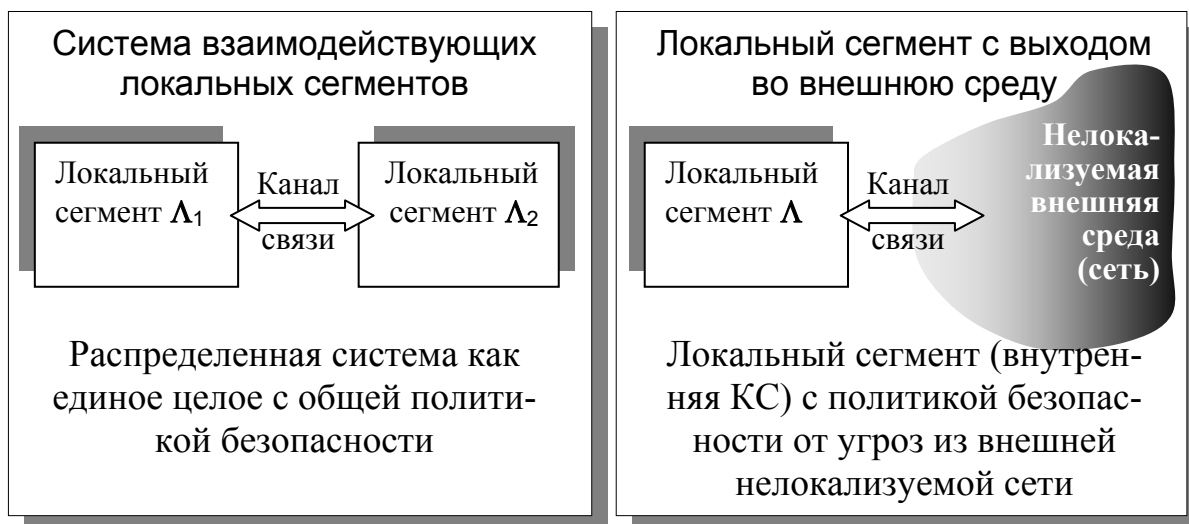


Рис. 2.32. Типы распределенных КС с точки зрения безопасности

Главным отличием первого типа распределенных систем является принципиальная возможность в любой момент времени декомпозиции их состава и структуры на конечное число компонентов, что позволяет фиксировать их границы, и на этой основе рассматривать совокупность взаимодействующих сегментов как обособленную (выделенную из внешнего окружения) распределенную систему. Заметим при этом, что в большинстве случаев архитектура подобных систем строится на основе принципа *открытости*, т. е. с возможностью добавления или исключения дополнительных компонентов (локальных сегментов). Однако возможность в любой момент времени фиксации границ распределенных систем первого типа позволяет определять границы их безопасности и на этой основе строить общую (согласованную) политику безопасности в распределенной системе, в т. ч. возможно, но не обязательно, имеющую свою специфику в локальных сегментах.

В распределенных системах второго типа фиксировать состав и структуру внешней по отношению к локальному сегменту стороны системы принципиально невозможно по техническим или организационным основаниям. В результате отсутствует возможность определения границ распределенной системы, и в плане безопасности речь можно вести только о

безопасности локального сегмента от внешней среды (внешней сети), называемого в этом случае внутренней КС (внутренней сетью).

Канал связи на рис. 2.32 подразумевает наличие специального телекоммуникационного субъекта (системного процесса) st , который обеспечивает взаимодействие локального сегмента с "внешним миром", т. е. с другими локальными сегментами или внешней сетью. Само же взаимодействие заключается в наличии информационных потоков из локального сегмента в другие (внешние) сегменты и наоборот.

Основываясь на понятиях субъекта и объекта КС, а также на понятии доступа субъекта к объекту, сформулируем важнейшее для безопасности распределенных систем понятие удаленного доступа.

Определение 2.8.2. Удаленным доступом $p^{\text{out}} = \text{Stream}(s_m, o_i) \rightarrow o_j$ субъекта s_m пользователя в локальном сегменте Λ_1 к объекту o_j в локальном сегменте Λ_2 называется порождение субъектом s_m через телекоммуникационные субъекты $s^{(1)}_t$ и $s^{(2)}_t$ локальных сегментов Λ_1 и Λ_2 потока информации между объектом o_j и некоторым(и) объектом o_i сегмента Λ_1 (в т. ч. объект(ы) o_i возможно, но не обязательно ассоциирован(ы) с субъектом s_m).

Опуская некоторые детали, понятие удаленного доступа в субъектно-объектной модели КС можно проиллюстрировать схемой, приведенной на рис. 2.33.

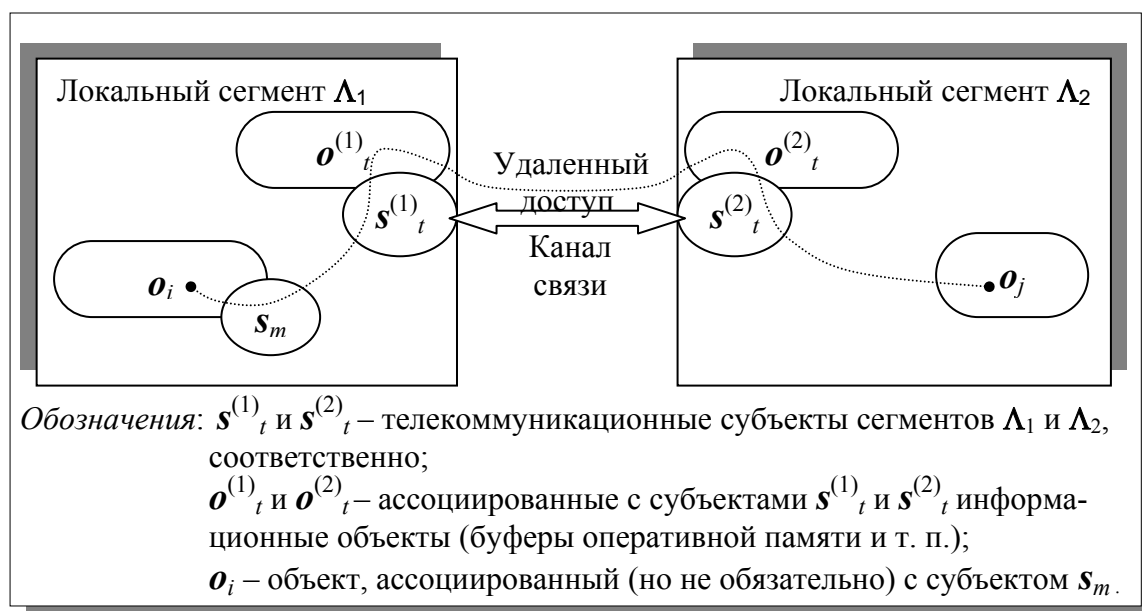


Рис. 2.33. Структура информационного потока при удаленном доступе

Отметим, что поток информации, вызываемый удаленным доступом, может быть как однонаправленным, причем и в ту (перенос информации из локального сегмента в другой сегмент), и в другую сторону (перенос

информации из другого сегмента в данный сегмент), так и двунаправленным (с одновременным переносом информации в обе стороны).

Еще одним принципиальным аспектом распределенных систем с точки зрения удаленных доступов является неявно заложенное в *определении 2.8.2* предположение о том, что пользователь локального сегмента Λ_1 по каким-либо причинам не может осуществлять доступ к объекту o_j непосредственно в самом локальном сегменте Λ_2 . Иначе бы необходимость в удаленных доступах отпала. Таким образом, является обоснованным еще одно аксиоматическое предположение.

Предположение 2.8.2. *Множество пользователей U распределенной системы взаимодействующих локальных сегментов $\{\Lambda_1, \Lambda_2, \dots, \Lambda_K\}$ является объединением непересекающихся подмножеств $\{U_1, U_2, \dots, U_K\}$ пользователей соответствующих локальных сегментов.*

Следует отметить, что, строго говоря, в *предположении 2.8.2* речь идет, не о пользователях, а об их учетных записях, которые являются объектами-источниками для порождения мониторами безопасности первичных субъектов пользователей. Поэтому один пользователь принципиально может иметь несколько учетных записей, распределенных по различным локальным сегментам. При этом, однако, для распределенной системы субъекты одного пользователя, инициированные по различным учетным записям, соответствуют различным пользователям.

Исходя из *предположения 2.8.2*, отметим также, что часто пользователей, осуществляющих удаленный доступ из одного локального сегмента к объектам другого локального сегмента, называют *удаленными пользователями*.

При рассмотрении взаимодействия локальных сегментов под углом зрения на удаленные доступы выделяется самый верхний слой политики безопасности в распределенной КС – политика отношений между локальными сегментами, выражаемая парадигмой *доверия* или иначе *доверительными отношениями*.

Определение 2.8.3. *Доверительными отношениями между локальными сегментами Λ_1 и Λ_2 называется составная часть общесистемной политики безопасности, определяющая возможность осуществления удаленных доступов пользователей одного локального сегмента к объектам другого локального сегмента.*

Можно выделить следующие виды доверительных отношений:

- одноуровневые отношения доверия, подразделяемые, в свою очередь, на
 - отношения одностороннего доверия;
 - отношения двустороннего доверия;
- иерархические отношения доверия.

При отношениях одностороннего доверия, скажем, локального сегмента Λ_1 к локальному сегменту Λ_2 принципиально возможны удаленные доступы субъектов пользователей локального сегмента Λ_2 к объектам локального сегмента Λ_1 , но удаленные доступы субъектов пользователей локального сегмента Λ_1 к объектам локального сегмента Λ_2 принципиально невозможны. Выражение "принципиально возможны" означает необходимость более детальной и конкретной регламентации (разрешения) конкретных доступов в соответствующем локальном сегменте (в данном случае в локальном сегменте Λ_1 и, добавим, по правилам политики безопасности локального сегмента Λ_1).

При отношениях двустороннего доверия принципиально возможны удаленные доступы субъектов одного локального сегмента к объектам другого локального сегмента и, наоборот, с той лишь оговоркой, что конкретное подмножество разрешенных удаленных доступов должно быть регламентировано в (и по правилам) соответствующих локальных сегментах.

Иерархические отношения доверия относятся к определенной разновидности архитектуры распределенных систем первого типа, когда некоторые локальные сегменты вложены в другие локальные сегменты. С точки зрения *определения 2.8.1* это означает, что обособленная совокупность субъектов и объектов внутреннего (вложенного) локального сегмента является одновременно некоторым подмножеством совокупности субъектов и объектов внешнего локального сегмента. Отношения иерархического доверия можно назвать неравноправными отношениями двустороннего доверия. Это означает, что так как субъекты пользователей внутреннего локального сегмента являются подмножеством субъектов доступа внешнего локального сегмента, каких-либо принципиальных ограничений для удаленных доступов "снизу вверх", т. е. субъектов внутреннего локального сегмента к объектам внешнего локального сегмента, не существует. Удаленные доступы субъектов внешнего локального сегмента к объектам внутреннего локального сегмента принципиально возможны, но они должны быть регламентированы во внутреннем локальном сегменте именно как разрешенные удаленные доступы пользователей внешнего локального сегмента.

Можно сформулировать два направления обеспечения безопасности в распределенных КС:

- выделение специального системного субъекта, обеспечивающего "внешнюю" безопасность;
- реализация общего, в том числе возможно с распределенной архитектурой, монитора безопасности, обеспечивающего единую (согласованную) политику безопасности распределенной компьютерной системы.

Первое направление, проистекающее в первую очередь из методов решения проблем безопасности в системах "Локальный сегмент КС (внутренняя сеть) – Внешняя среда (внешняя сеть)", впоследствии получило распространение и в распределенных системах первого типа (в системах взаимодействующих локальных сегментов) в виде широко известных в настоящее время *межсетевых экранов*. Метод межсетевого экранирования основан на анализе потоков информации низкого уровня и фильтрации их на множество опасных и неопасных по априорно определенным критериям.

Задачи второго направления были поставлены еще в сетевой интерпретации "оранжевой книги" ("Trusted Network Interpretation", 1987 г.)¹ в виде требований по реализации сетевого монитора безопасности (монитора обращений) – **NTCB** (Network Trusted Computer Base).

В более широкой трактовке в соответствии с *определением 2.8.2* удаленный доступ не обязательно сопровождается созданием нового объекта на той или иной стороне доступа. Кроме того, и это наиболее существенно, удаленный доступ к объекту осуществляется субъектом, управляемым пользователем, находящимся (работающим) в другом локальном сегменте. При этом если вернуться к схеме на рис. 2.33, то цепочку " $s_m(\Lambda_1) - s^{(1)}_t - s^{(2)}_t$ " можно интерпретировать как внутренний субъект доступа в локальном сегменте Λ_2 удаленного пользователя, работающего в локальном сегменте Λ_1 .

Поэтому в более широком аспекте второго направления обеспечения безопасности распределенных КС требуется рассмотрение условий, допускающих инициализацию в локальных сегментах субъектов доступа удаленных пользователей.

В заключение отметим, что в литературе, кроме того, представлен целый ряд работ, посвященных анализу и синтезу моделей, механизмов безопасности в распределенных системах. В. Варадхараджаном (Vijay Varadharajan) в 1990 г. предложена модель безопасности сети, основанная на сочетании дискреционного, мандатного и ролевого принципа разграничения доступа. Множество объектов системы включает помимо информационных объектов и множество сетевых компонент, имеющих в том числе свою классификацию. В состав множества операций включены операции по установлению связи субъектов с удаленными компонентами сети. Выполнение операций сопровождается прохождением системой двух фаз – фазы доступа к системе и фазы установления связи. На основе дискреционных, мандатных и ролевых механизмов рассматриваются условия и ограничения, накладываемые на операции, при которых обеспечивается безопасность сети. При этом, однако, модель Варадхараджана не охватывает доменно-групповой архитектуры современных компьютерных сетей.

¹ так называемая "Красная книга".

2.8.3. Зональная модель разграничения доступа к информации в распределенных компьютерных системах

Практика развития и совершенствования программных средств, обеспечивающих создание крупных корпоративных сетей, опережая развитие теоретических моделей, привела к использованию зонального подхода в организации системы разграничения доступа. Подобный подход, в частности, реализуется в архитектуре распределенных информационных сетей на базе ОС Windows NT/2000 и именуется доменно-групповой архитектурой безопасности сети.

Очевидно, как и по другим моделям разграничения доступа, исходным мотивом для данных решений послужил анализ наработанных во внекомпьютерной сфере подходов к разграничению доступа, в частности, организационно-режимных схем управления доступом на территорию и в помещения сотрудников предприятий и учреждений. Эти схемы характеризуются рубежно-иерархическим принципом построения и дифференциацией режимных мер путем выделения так называемых *зон* – открытых зон, режимных (закрытых) зон, производственно-технологических, складских зон и т. п. Каждая зона характеризуется своим режимом (политикой), включающим регламентацию сотрудников, работающих внутри зоны, процедуры вхождения в зону и расположения в зоне ресурсов (материальных, документальных), а также целый ряд других вопросов. Доменно-групповая архитектура сетей Windows NT/2000 как раз и призвана воспроизвести в корпоративной сети предприятия подобные организационно-режимные схемы.

Однако, насколько известно, формальных спецификаций модели зонального разграничения доступа в специальной литературе не представлено.

Основываясь на анализе и обобщении зонально-режимных схем организации и управления доступом сотрудников на территорию и в помещения предприятий и учреждений, можно сформулировать следующие *положения зональной модели разграничения доступа*, отчасти коррелирующей с решениями, лежащими в основе доменно-групповой архитектуры сетей на базе ОС Windows NT/2000.

1. Распределенная КС представляется совокупностью следующих наборов сущностей:

- множества объектов (информационных) доступа $O(o_1, o_2, \dots, o_M)$;
- множества пользователей $U(u_1, u_2, \dots, u_N)$;
- множества физических объектов системы $V(v_1, v_2, \dots, v_L)$ – вычислительных установок (рабочие станции, серверы), принтеров, коммуникационного оборудования и т. п.;
- множества зон системы $Z(z_1, z_2, \dots, z_K)$.

Определение 2.8.4. Зоной в распределенной КС называется совокупность подмножества пользователей, подмножества объектов доступа и подмножества физических объектов, обособленных в локальный сегмент с отдельной (внутризональной) политикой безопасности.

Нетрудно видеть, что зона в *определении 2.8.4* представляет разновидность локального сегмента, субъекты и объекты которого обособляются на основе локализации в пределах выделенного подмножества физических объектов.

2. Внутризональная политика безопасности в соответствии с *предположением 2.8.1* реализуется *внутризональным монитором безопасности*, который обеспечивает весь набор функций безопасности (аутентификация и порождение первичных субъектов доступа пользователей зоны, управление доступом, аудит процессов).

Определение 2.8.5. *Внутризональным монитором безопасности называется системный субъект (процесс), реализующий в отношении объектов зоны $z \in \mathcal{Z}$ разрешенное множество доступов $P_L(z)$, которое в общем виде является объединением внутризональных доступов, регламентированных правилами (критериями) внутризональной политики, и удаленных доступов пользователей зоны к объектам других зон, пользователей других зон к объектам данной зоны, разрешенных по правилам (критериям, процедурам) межзональной политики безопасности:*

$$P_L(z) = P^{\text{in}}_L(z) \cup P^{\text{out}}_L(z), \quad (2.8.1)$$

где $P^{\text{in}}_L(z)$ – множество безопасных внутризональных доступов;

$P^{\text{out}}_L(z) = P^{\text{out}}_L(z \rightarrow) \cup P^{\text{out}}_L(z \leftarrow)$ – множество безопасных удаленных доступов для зоны $z \in \mathcal{Z}$, являющееся объединением множества удаленных доступов пользователей зоны z к объектам других зон – $P^{\text{out}}_L(z \rightarrow)$ и множества удаленных доступов пользователей других зон к объектам зоны z – $P^{\text{out}}_L(z \leftarrow)$.

Ввиду того, что зона может быть образована на нескольких вычислительных установках, возможны два подхода реализации единого внутризонального монитора безопасности:

- по принципу "Клиент-сервер";
- по принципу реплицирования.

Клиент-серверный подход, обуславливает выделение среди физических объектов зоны одного или нескольких *серверов*, на котором(ых) для всех субъектов пользователей, работающих на других вычислительных установках, функционирует процесс (системный субъект) единого монитора безопасности. Важным аспектом при этом является организация и размещение на сервере ассоциированного с монитором безопасности информационного объекта, в котором находятся данные по внутризональной поли-

тике безопасности. Основной частью этих данных является идентификационно-аутентификационная информация пользователей, а также возможно другие сведения, необходимые для реализации соответствующей политики безопасности (привилегии, уровни доверия-допуска, разрешенная тематика, назначенные роли пользователей зоны и т. д.)¹. Отсюда аутентификация и порождение первичных субъектов пользователей зоны осуществляются субъектами монитора безопасности (системными процессами ядра КС) на сервере зоны.

При втором подходе монитор безопасности может функционировать независимо на каждой вычислительной установке системы, но ассоциированный с ним информационный объект, содержащий данные по внутризональной политике безопасности, представляет систему реплик одной общей базы данных. Реплики базы учетных записей пользователей размещены на всех вычислительных установках зоны. Изменения, вносимые администратором системы в реплику на любой вычислительной установке, автоматически распространяются (реплицируются) на все вычислительные установки системы².

С формальной точки зрения принцип программно-технической реализации внутризонального монитора безопасности не имеет значения. Принципиальным же является то, что внутризональный монитор безопасности должен обеспечивать полный набор функций безопасности, основной из которых является разграничение доступа любых субъектов к любым объектам своей зоны.

3. Зональная структура системы определяется отображениями множеств сущностей, выражаемыми следующими функциями:

$f_{\text{phys}} : V \rightarrow Z$ – значением функции $z = f_{\text{phys}}(v)$ является зона $z \in Z$, в которой находится (которой принадлежит) физический объект $v \in V$;

$f_{\text{user}} : U \rightarrow Z$ – значением функции $z = f_{\text{user}}(u)$ является зона $z \in Z$, в которой уполномочен (зарегистрирован) для работы пользователь $u \in U$;

$f_{\text{object}} : O \rightarrow V$ – значением функции $v = f_{\text{object}}(o)$ является физический объект $v \in V$, в котором находится (физически размещается) объект $o \in O$.

Следует отметить, что совокупность функций $v = f_{\text{object}}(o)$ и $z = f_{\text{phys}}(v)$ реализует функцию $z = f_{\text{object}}(o)$, определяющую принадлежность объекта доступа $o \in O$ к определенной зоне $z \in Z$, что можно трактовать как "зо-

¹ Чаще всего подобный объект называют "базой учетных записей" пользователей.

² Данный процесс называется "синхронизацией реплик".

нальную окрашенность"¹ всех объектов доступа зонально-распределенной системы.

4. На множестве зон системы $\mathcal{Z}$ определяется частичный нестрогий порядок, устанавливающий систему межзональных доверительных отношений:

$F_{\mathcal{Z}\mathcal{Z}}: \mathcal{Z} \times \mathcal{Z}$ - отношение, определяющее приоритет доверия одних зон безопасности по отношению к другим и задающее оператор доминирования $\leq$, такое, что

если для $z_1, z_2 \in \mathcal{Z}$, $z_1 = z_2$, то между зоной z_1 и зоной z_2 установлены отношения двустороннего доверия (зоны z_1, z_2 доверяют друг другу, иначе говоря, принципиально возможны удаленные доступы пользователей зоны z_1 к объектам зоны z_2 , и, наоборот, пользователей зоны z_2 к объектам зоны z_1)² –

$$P^{\text{out}}_L(z_1 \rightarrow z_2) \neq \emptyset \wedge P^{\text{out}}_L(z_1 \leftarrow z_2) \neq \emptyset ; \quad (2.8.2)$$

если для $z_1, z_2 \in \mathcal{Z}$, $z_1 > z_2$, то между зоной z_1 и зоной z_2 установлены отношения одностороннего доверия (зона z_2 доверяет зоне z_1 , но зона z_1 не доверяет зоне z_2 , иначе говоря, принципиально возможны удаленные доступы пользователей зоны z_1 к объектам зоны z_2 , но удаленные доступы пользователей зоны z_2 к объектам зоны z_1 принципиально невозможны) –

$$P^{\text{out}}_L(z_1 \rightarrow z_2) \neq \emptyset \wedge P^{\text{out}}_L(z_1 \leftarrow z_2) = \emptyset ; \quad (2.8.3)$$

если для $z_1, z_2 \in \mathcal{Z}$, $z_1 \neq z_2$, то отношения доверия между зонами z_1, z_2 не установлены (зоны z_1, z_2 не доверяют друг другу, иначе говоря, принципиально невозможны удаленные доступы пользователей зоны z_1 к объектам зоны z_2 и удаленные доступы пользователей зоны z_2 к объектам зоны z_1)³ –

$$P^{\text{out}}_L(z_1 \rightarrow z_2) = \emptyset \wedge P^{\text{out}}_L(z_1 \leftarrow z_2) = \emptyset . \quad (2.8.4)$$

Отметим, что отношение частичного нестрогого порядка на множестве $\mathcal{Z}$ адекватно воспроизводит доверительные отношения, так как обладает свойством рефлексивности, антисимметричности и транзитивности:

$$\forall z \in \mathcal{Z}: z \leq z ,$$

¹ иначе говоря - любой объект может существовать только в своей зоне и его перемещение, транспортировка не предусмотрены, т. к. по определению 2.8.2 операции копирования, перемещения при удаленном доступе осуществляются через создание на основе межсегментного потока в зоне-приемнике **нового объекта**.

² знак "=" означает в данном случае неразличимость (равность) зон по отношению доминирования $\leq$.

³ знак " $\neq$ " означает несравнимость зон по приоритету доверия.

$$\forall z_1, z_2 \in \mathcal{Z}: (z_1 \leq z_2 \wedge z_2 \leq z_1) \Rightarrow z_1 = z_2,$$

$$\forall z_1, z_2, z_3 \in \mathcal{Z}: (z_1 \leq z_2 \wedge z_2 \leq z_3) \Rightarrow z_1 \leq z_3.$$

Действительно, любая зона доверяет самой себе, т.е. внутризональные доступы не запрещены (рефлексивность). Если одна зона доверяет другой (возможны удаленные доступы пользователей первой зоны к объектам во второй зоне) и, одновременно, другая зона доверяет первой (возможны удаленные доступы пользователей второй зоны к объектам первой зоны), то по определению это означает наличие отношений двустороннего доверия (антисимметричность). И, наконец, если одна зона доверяет другой, а другая доверяет третьей, то принципиально запрещать удаленные доступы пользователей первой зоны к объектам третьей зоны не имеет смысла, так как по цепочке удаленных доступов информация может быть перенесена из третьей зоны в первую (транзитивность).

5. Процессы доступа пользователей к объектам системы организуются в две фазы:

- "вхождение" пользователя в зону и порождение (первичного) субъекта доступа;
- запрос на доступ субъекта пользователя у внутризонального монитора безопасности к объекту зоны и получение доступа (осуществление потока) в случае удовлетворения запрашиваемого доступа зональной политики безопасности.

Для формализации представления процессов внутризонального доступа и введения дополнительного аспекта внутризональной политики будем выделять на множестве физических объектов зоны $v^z \in V^z$ (где $V^z = f^{-1}_{\text{phys}}(z)$) подмножество вычислительных установок зоны – $V^{zu} \subseteq V^z$. На этой основе устанавливается отображение подмножества пользователей $u^z \in U^z$ (где $U^z = f^{-1}_{\text{user}}(z)$), уполномоченных (зарегистрированных) для работы в зоне $z \in \mathcal{Z}$, на множество вычислительных установок зоны V^{zu}_v :

$F^{zu}_{UV}: U^z \times V^z$ – отображение, определяющее для каждого пользователя зоны вычислительные установки $v^{zu} \in V^{zu}_v$ (где $V^{zu} = F^{zu}_{UV}(u^z)$, $u^z \in U^z$, $V^{zu} \subseteq V^z$), на которых пользователь может осуществлять "вхождение" в свою зону.

Определение 2.8.6. *Процедурой вхождения пользователя $u \in U$ в "свою" зону $z = f_{\text{user}}(u)$ называется процесс идентификации/аутентификации и порождения под управлением внутризонального монитора безопасности первичного субъекта доступа, осуществляемый пользователем на одной из вычислительных установок зоны $v^{zu} \in V^z$ при выполнении следующего условия:*

$$u \in U^z \wedge v^{zu} \in V^{zu}, \quad (2.8.5)$$

$$\text{где } U^z = f_{\text{user}}^{-1}(z);$$

$$V^{zu} = F_{UV}^z(u^z).$$

Отметим, что субъекты доступа пользователя, инициализируемые в момент вхождения на различных вычислительных установках доступа равнозначны по правам доступа, если иное не предусмотрено ограничениями внутризональной политики безопасности, регламентирующей множество безопасных внутризональных доступов (по дискреционному, мандатному, тематическому или ролевому принципу).

Вместе с тем, общим правилом, существенным с точки зрения безопасности, является следующее ограничение.

Предположение 2.8.3. *В любой момент времени любой пользователь может осуществлять процедуру вхождения и работать в своей зоне только на одной вычислительной установке.*

Обоснованность предположения 2.8.3 вытекает из самого понятия зоны, предполагающего множественность и физическую распределенность вычислительных установок зоны. Отсюда, если предоставлять возможность одновременного (точнее параллельного) вхождения пользователя в зону сразу с нескольких вычислительных установок, то, вообще говоря, для внутризонального монитора безопасности будут существовать несколько различных внешних факторов, порождающих субъекты доступа с одинаковыми правами, что может привести к проблемам с идентификацией/аутентификацией пользователя в данной зоне.

6. Для формализации удаленного (межзонального) доступа введем следующее отображение:

$F_{UZ}^{\text{out}}: U \times Z$ - отображение множества пользователей на множество "не своих" зон $z' \in Z$ (где $Z = (Z \setminus f_{\text{user}}(u))$), которые доверяют зоне пользователя ($z' \leq f_{\text{user}}(u)$) и в которых данный удаленный для этих зон пользователь может порождать субъекты доступа и осуществлять тем самым удаленные (межзональные) доступы к объектам соответствующих зон.

Определение 6.7. *Процедурой вхождения удаленного пользователя $u \in U$ в "не свою" зону $z' \in Z \setminus f_{\text{user}}(u)$ называется процесс идентификации/аутентификации и порождения под управлением внутризонального монитора безопасности зоны z' и телекоммуникационных субъектов зон $z = f_{\text{user}}(u)$ и z' первичного субъекта удаленного доступа пользователя u в "не своей" зоне z' при выполнении следующего условия:*

$$z \notin Z \wedge z \geq z' \wedge z' \in F_{UZ}^{\text{out}}(u), \quad (2.8.6)$$

$$\text{где } Z = Z \setminus f_{\text{user}}(u);$$

$$z = f_{\text{user}}(u).$$

Процедуры удаленной¹ идентификации/аутентификации требуют определенной последовательности действий пользователя (претендента) и монитора безопасности удаленной зоны (верификатора). Указанные последовательности действий образуют те или иные протоколы аутентификации, для реализации которых в общем виде не требуется защиты канала связи.

Вместе с тем, совершенно очевидно, что последующие в результате удаленных доступов межзональные потоки информации требуют обеспечения безопасности канала связи, который в большинстве теоретических моделей и практических систем представляет собой общий неразделяемый ресурс. На этом основании введем следующее предположение.

Предположение 2.8.4. *Зонально-распределенная система строится на безопасном канале связи, который удовлетворяет следующим требованиям:*

- устойчивости к несанкционированному раскрытию или модификации передаваемой ценной информации (безопасность связи);
- невозможности отказов от доставки сообщения, неправильной доставки, доставки ошибочных данных (надежность связи);
- невозможности изменений в критической информации (имитозащита);
- отсутствия скрытых каналов утечки информации за счет модуляции параметров канала.

С учетом определений 2.8.2 и 2.8.7 введем понятие субъекта удаленного доступа.

Определение 2.8.8. *Субъектом удаленного доступа $s'_m(z')$ называется активная сущность (процесс) в зонально-распределенной системе, инициализируемая пользователем u , работающим в зоне z , на основе процедуры вхождения в зону z' в виде совокупности действующих по цепочке субъектов " $s_m(z) - s^{(z)}_t - s^{(z')}_t$ ", осуществляющая информационный поток между зоной z и зоной z' по цепочке объектов " $o_i(z) - o^{(z)}_t - o^{(z')}_t - o_j(z')$ ",*

где $s_m(z)$ – субъект доступа, инициализированный в зоне z пользователем u , причем $z = f_{\text{user}}(u)$;

$s^{(z)}_t, s^{(z')}_t$ – системные телекоммуникационные субъекты зон z и z' , соответственно;

$o^{(z)}_t, o^{(z')}_t$ – объекты, ассоциированные с телекоммуникационными субъектами $s^{(z)}_t, s^{(z')}_t$ зон z и z' , соответственно;

$o_i(z)$ – объект зоны z ($f_{\text{object}}(o_i) = z$), в том числе не обязательно ассоциированный с субъектом $s_m(z)$;

$o_j(z')$ – объект зоны z' ($f_{\text{object}}(o_j) = z'$).

¹ на удаленной вычислительной установке.

7. Общесистемная политика безопасности складывается из совокупности политики внутризонального и межзонального разграничения доступа, регламентирующей множество разрешенных (легальных) доступов:

$$P_L = P^{\text{in}}_L(z_1) \cup P^{\text{in}}_L(z_2) \cup \dots \cup P^{\text{in}}_L(z_K) \cup P^{\text{out}}_L(z_1) \cup P^{\text{out}}_L(z_2) \cup \dots \cup P^{\text{out}}_L(z_K) \quad (2.8.7)$$

При этом:

$$P^{\text{out}}_L(z_i) = \bigcup_{\substack{i \neq k \\ k=1}}^K P^{\text{out}}_L(z_i \rightarrow z_k) \cup \bigcup_{\substack{i \neq k \\ k=1}}^K P^{\text{out}}_L(z_i \leftarrow z_k). \quad (2.8.8)$$

Справедливо следующее утверждение.

Утверждение 2.8.1. (Теорема безопасности зонально-распределенной системы). Совокупность внутризональных мониторов безопасности, санкционирующих не выходящие за множество (2.8.1) доступы к объектам своих зон субъектам, которые порождаются на основе процедур вхождения по определениям 2.8.6, 2.8.7 и 2.8.8 с учетом предположений 2.8.2, 2.8.3, 2.8.4, реализует общезональный монитор безопасности и общезональное множество безопасных доступов (2.8.7).

Доказательство.

В соответствии с предположениями 2.8.1, 2.8.2, а также с учетом предположения 2.8.3, регламентирующих процедуры вхождения пользователей в "свои" зоны, любые субъекты любых пользователей в своих зонах находятся под полным контролем соответствующих внутризональных мониторов безопасности, санкционирующих по условиям теоремы только подмножества безопасных (разрешенных) доступов (2.8.1). Иначе говоря, внутризональные мониторы безопасности, организующие по определению 2.8.6 процедуры вхождения пользователей в свои зоны, гарантируют отсутствие в своих зонах субъектов, осуществляющих доступы вне подмножеств разрешенных внутризональных доступов.

Что касается субъектов доступа к объектам зон от удаленных пользователей, то они порождаются по определению 2.8.8 и также находятся под полным контролем соответствующих внутризональных мониторов безопасности, так как процедурами вхождения в зоны пользователей других зон (удаленных пользователей) управляют мониторы безопасности зоны вхождения. При этом процедуры вхождения удаленных пользователей и инициализация в зонах их субъектов производится при выполнении условия (2.8.6):

- зона вхождения доверяет зоне удаленного пользователя;
- пользователь уполномочен работать в данной зоне, входящей в подмножество доверенных зон.

Отсюда, с учетом условия утверждения относительно соотношения (2.8.1), санкционирование внутризональными мониторами безопасности

доступов субъектов удаленных пользователей вызывает только те доступы, которые не могут выйти за множество доступов, регламентируемых доверительными отношениями (2.8.2), (2.8.3) и (2.8.4).

С учетом *предположения 2.8.4* (безопасности каналов связи) не могут также возникнуть и любые другие вне рамок доверительных отношений доступы (потoki) при прохождении информации через канал связи.

Объединяя множества доступов, регламентируемых каждым внутризональным монитором безопасности по соотношению (2.8.1) в отдельности, получаем общезональное множество разрешенных доступов, выражаемое соотношением (2.8.7).

Таким образом, совокупность внутризональных мониторов безопасности, работающих на основе *предположений 2.8.1, 2.8.2, 6.3* и по *определениям 2.8.6, 2.8.7 и 2.8.8* реализует общезональный монитор безопасности, обеспечивающий единую и согласованную общезональную политику безопасности. ■

Заканчивая рассмотрение зональной модели, отметим два обстоятельства.

Утверждение 2.8.1 доказывает общезональную безопасность, вообще говоря, на основе априорно заданной внутризональной безопасности и определенных процедур, отношений, регламентаций по взаимодействию зон (локальных сегментов). Отсюда следует, что изъяны и бреши в системе безопасности зонально-распределенной КС могут возникнуть либо на основе изъянов внутризональной безопасности, либо на основе ошибок при реализации правил и регламентаций межзонального взаимодействия.

В этом плане представляет также интерес анализ выполнения положений и ограничений представленной зональной модели безопасности по отношению к модели безопасности, реализуемой ОС Windows NT/2000. Можно отметить несколько изъянов в системе безопасности Windows NT/2000.

Первое – канал связи в сетях на основе Windows NT/2000 не защищен в смысле *предположения 2.8.4*.

Второе – процедуры вхождения удаленных пользователей в "не свои" домены (зоны) не требуют дополнительной аутентификации. Иначе говоря, мониторы безопасности доменов доверяют друг другу по процедурам аутентификации пользователей, что с учетом незащищенности каналов связи создает ряд сетевых угроз безопасности.

И третье (изъян файловой системы NTFS) – информационные объекты не имеют зональной (доменной) окрашенности. Любой файл, каталог или диск отдельной вычислительной установки может управляться и контролироваться любым действующим на данной вычислительной установке системным процессом ядра Windows NT/2000 с правами администратора, т. к. вся информация о доступе находится в списках доступа вместе с объ-

ектами без привязки к конкретному монитору безопасности (монитору ссылок). Иначе говоря, нет однозначной связи (привязки) между внутризональным монитором безопасности (точнее базой учетных записей) и объектами доступа вычислительной установки и домена. В результате просматривать объекты файловой системы NTFS может любой субъект (процесс) с правами администратора, запущенный на соответствующей вычислительной установке.

Представленная зональная модель не накладывает никаких ограничений на политику безопасности внутри различных зон и, в том числе, не требует одинаковости внутризональных политик безопасности. Это означает, что субъекты удаленных пользователей получают доступ к объектам по правилам и регламентациям политики тех зон, в которых находятся объекты доступа. При этом информация по разрешенным доступам удаленных пользователей (списки доступа, уровни допуска, разрешенная тематика или роли) находится в объекте, располагаемом в зоне объектов доступа, т. е. под управлением того монитора безопасности, который обеспечивает процедуру вхождения и порождение субъектов доступа удаленных пользователей. В результате обеспечивается максимально возможная гибкость в общесистемной политике безопасности. Одни зоны могут основываться на дискреционной, другие на мандатной, третьи на тематической, а четвертые на ролевой политике разграничения доступа. При этом на основе системы доверительных отношений и определенных регламентаций межзонального доступа данные зоны могут взаимодействовать и совместно функционировать как единое целое.

Данное утверждение может показаться странным для случая, когда одна зона основана на дискреционной политике, а другая на мандатной политике. Удаленные доступы субъектов из первой зоны к классифицированным по грифам секретности объектам другой зоны могут вызывать недопустимые потоки "сверху вниз". Данная проблема удаленного доступа в мандатных системах была отмечена еще в ранних работах по анализу модели Белла-ЛаПадулы. Вместе с тем, в рамках зональной политики с доверительными отношениями она легко разрешается. В частности, определение доверительных отношений между зонами, часть из которых поддерживает мандатную политику, можно основывать на отображении самих зон на решетку уровней безопасности. Присвоение уровня безопасности какой-либо зоне с дискреционной политикой автоматически присваивает с точки зрения других зон с мандатной политикой всем субъектам данной зоны соответствующий допуск, а всем объектам соответствующий гриф секретности. Зоны же с исходной мандатной политикой в этом случае рассматриваются как групповые объекты с уровнями безопасности, задаваемыми гранями множества уровней безопасности субъектов и объектов зоны. Доверительные отношения соответственно задаются линейным по-

рядком на решетке уровней безопасности по известным правилам **NRU** и **NWD**. Заметим также, что при одинаковости уровней безопасности всех субъектов и объектов зоны с учетом использования в модели Белла-ЛаПадулы матрицы доступа, разграничивающей доступ внутри классов безопасности, сама политика мандатного доступа вырождается в дискреционную политику. Поэтому утверждение о возможности удаленных доступов между зоной с дискреционной политикой и зоной с мандатной политикой с учетом данного замечания имеет вполне обоснованный и непротиворечивый смысл.

III. МЕТОДЫ АНАЛИЗА И ОЦЕНКИ ЗАЩИЩЕННОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

3.1. ТЕОРЕТИКО-ГРАФОВЫЕ МОДЕЛИ КОМПЛЕКСНОЙ ОЦЕНКИ ЗАЩИЩЕННОСТИ

Большое разнообразие методов и механизмов обеспечения компьютерной безопасности (см. п. 1.1.3), принцип разумной достаточности при обеспечения компьютерной безопасности (см. п.1.1.2) предопределяют необходимость формальных моделей и методик анализа эффективности систем защиты. В литературе и практике данные вопросы характеризуются термином "комплексные оценки защищенности" включая агрегированную оценку (в комплексе) эффективности методов и механизмов как программно-технического, так и нормативно-организационного характера для защиты активов организации.

Решение подобных задач основывается на теоретико-графовом подходе, предложенном еще в начале 70-х гг. в виде т.н. "модели систем с полным перекрытием [угроз безопасности]".

Система защиты в рамках данной формализации представляется дольным графом, вариант которого представлен на рис. 3.1.

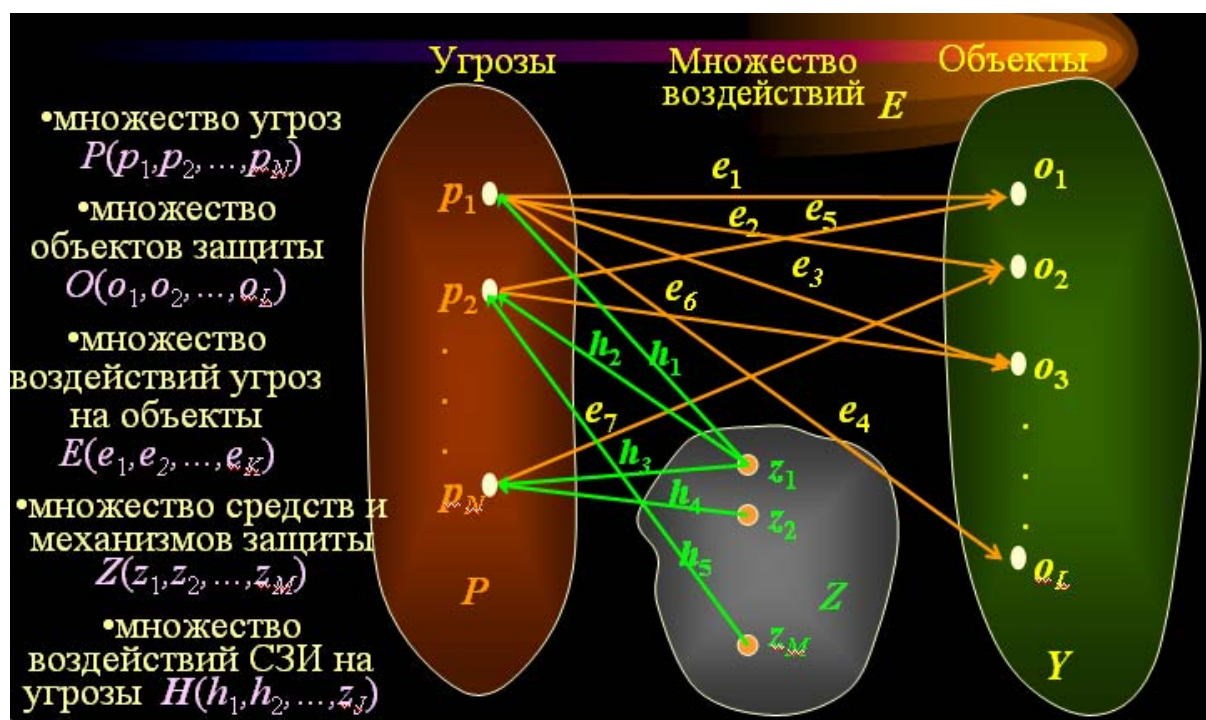


Рис.3.1. Модель системы защиты в виде 3-х дольного графа

Каждое ребро графа $G(P, O, Z, E, H)$ специфицирует воздействие конкретной угрозы на конкретный актив (объект) или устранение (нейтрализацию) определенным защитным механизмом угрозы безопасности. При этом от каждой угрозы м.б. несколько воздействий на различные объекты

и каждый объект м.б. подвергнут нескольким угрозам (связь "многие-ко-многим").

Для получения и оперирования количественными параметрами граф $G(P, O, Z, E, H)$ взвешивается и эквивалентно представляется следующей совокупностью векторов и матриц:

вектор $P(p_1, p_2, \dots, p_N)$, где p_i – вероятность осуществления соответствующей угрозы;

вектор $O(o_1, o_2, \dots, o_L)$, где o_i – стоимость соответствующего объекта защиты;

$N \times L$ матрица $E\{e_{ij}\}$, где $e_{ij} = 1$ при воздействии i -й угрозы на j -й объект, и $e_{ij} = 0$ в противном случае;

вектор $Z(z_1, z_2, \dots, z_M)$, где z_i – стоимость соответствующего способа или средства защиты;

$N \times M$ матрица $H\{h_{ij}\}$, где h_{ij} – вероятность устранения (или степень снижения ущерба) i -й угрозы от применения j -го средства защиты.

В рамках данной формализации возможно решение таких важных практических задач, как анализ технико-экономической или тактико-технической эффективности систем защиты.

Анализ технико-экономической эффективности системы защиты основывается на определении количественного критерия, отражающего влияние реализованной системы защиты (принятых) на снижение опасности от воздействия угроз, или иначе на величину ущерба от угроз безопасности.

В качестве такого критерия технико-экономической эффективности, в частности, можно использовать следующий показатель:

$$\mathcal{E} = \frac{U - U'}{\sum_{i=1}^L o_i + \sum_{k=1}^M z_k}, \quad (3.1.1)$$

где U – оценка величины ущерба от угроз безопасности при отсутствии защитных мер и механизмов;

U' – оценка величины остаточного ущерба при реализации защитных мер и механизмов.

Расчет величин U и U' осуществляется по следующим соотношениям:

$$U = \sum_{i=1}^L o_i (1 - \prod_{j=1}^N e_{ij} (1 - p_j)) \quad (3.1.2)$$

$$U' = \sum_{i=1}^L o_i (1 - \prod_{j=1}^N e_{ij} (1 - p_j (1 - \prod_{k=1}^M (1 - h_{jk})))) \quad (3.1.3)$$

Задачи тактико-технического анализа эффективности систем защиты заключаются в определении остаточной вероятности реализации всех возможных угроз без учета стоимости объектов и мер защиты, что м.б. вычислено по следующему соотношению:

$$P'_{преод} = 1 - \prod_{i=1}^N (1 - p_i (1 - \prod_{k=1}^M (1 - h_{ik}))) \quad (3.1.4).$$

Отмечая практическую значимость приведенных соотношений, следует заметить, что их использование влечет необходимость применения методов оценивания угроз безопасности, рассмотренных в п.1.2.2, в частности методов экспертных оценок вероятности реализации угроз и величин h_{ij} степени (вероятности) устранения угроз в зависимости от принятых мер и механизмов защиты.

3.2. МЕТОДЫ АНАЛИЗА И ОПТИМИЗАЦИИ ИНДИВИДУАЛЬНО-ГРУППОВЫХ СИСТЕМ РАЗГРАНИЧЕНИЯ ДОСТУПА

3.2.1. Теоретико-графовая модель системы индивидуальных назначений доступа к иерархически организованным объектам

При синтезе (проектировании) системы назначений прав доступа в рамках политики индивидуально-группового доступа имеем исходные параметры, которые можно представить в виде двудольного графа (рис.3.2).



Рис.3.2. Двудольный граф требуемых пользователям (субъектам) назначений доступа

При организации объектов доступа в иерархические структуры (каталоги, подкаталоги и т.д.) граф перестает быть двудольным. При этом права доступа передаются от каталогов к вложенным в них объектам (по наследству). Таким образом, назначения прав доступа можно осуществлять как напрямую на объект, так и через назначения доступа к каталогу, в котором находится объект. Соответственно итоговые права доступа субъекта к объекту складываются из прав по прямым назначениям и прав, получаемых по наследованию от прав субъекта к каталогам, в которых находится объект доступа.

В результате возникает вариативность наделения пользователей требуемыми правами доступа к объектам системы. Ситуация иллюстрируется на рис.3.3.

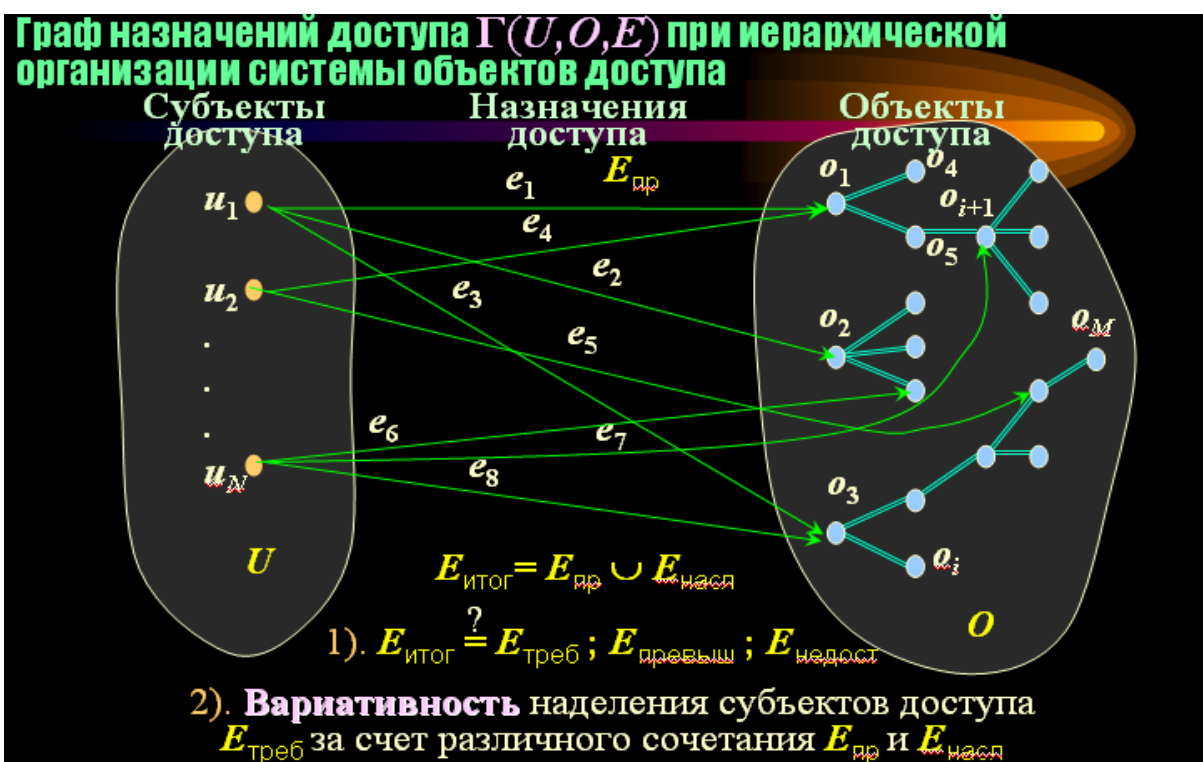


Рис.3.3. Граф назначений доступа при иерархической организации объектов

Вариативность еще более усиливается при использовании технологий рабочих групп пользователей для наделения их требуемыми одинаковыми правами доступа групповым образом – см. рис.3.4.



Рис.3.4. Граф индивидуально-групповых назначений доступа при иерархической организации системы объектов доступа

Граф $\Gamma(U, G, O, E)$ эквивалентно м.б. представлен совокупностью следующих матриц:

- $\mathbf{R}^n$ – $(N \times M \times K)$ -массив разрешенных для субъектов методов доступа к объектам по индивидуальным назначениям (k - номер метода доступа, $r_{nmk}=1$, если доступ разрешен, $r_{nmk}=0$, если доступ не разрешен, $r_{nmk}=-1$, если доступ явно запрещен);
- $\mathbf{R}^r$ – $(L \times M \times K)$ -массив разрешенных рабочим группам методов доступа к объектам по непосредственным групповым назначениям;
- $\mathbf{H}$ – квадратная $(M \times M)$ матрица смежности объектов доступа;
- $\mathbf{H}^r$ – квадратная $(L \times L)$ матрица смежности рабочих групп, аналогичная по смыслу матрице $\mathbf{H}$ ($h_{ij}^r=1$, если i -я рабочая группа содержит j -ую рабочую групп, $h_{ij}^r=0$, если не содержит);
- $\mathbf{W}$ – прямоугольная $(N \times L)$ матрица вхождения пользователей в рабочие группы ($w_{ij}=1$, если i -й пользователь входит в состав j -й рабочей группы; $w_{ij}=0$, в противном случае).

На основе данной формализации можно получить ряд соотношений, позволяющих вычислять итоговые права доступа с учетом сочетания прав по прямым назначениям, и прав получаемых по иерархической вложенности объектов:

$$\mathbf{R}^n|_{\text{итог}} = \mathbf{R}|_{\text{пр}} \otimes (\mathbf{H}^S + \mathbf{I}), \quad (3.2.1)$$

где $\mathbf{H}^S = \mathbf{H} + \mathbf{H}^2 + \dots + \mathbf{H}^n$,

$\otimes$ - модифицированная операция матричного умножения на основе ассиметричной дизъюнкции:

$$\mathbf{R}^n|_{\text{итог}})_{ij} = r_{i1}(h_{1j}^S + \delta_{1j}) \vee' r_{i2}(h_{2j}^S + \delta_{2j}) \vee' \dots \vee' r_{iM}(h_{Mj}^S + \delta_{Mj}),$$

$\mathbf{I}$ – единичная матрица;

δ_{ij} – символ Кронекера.

При этом ассиметричность операции $\vee'$ обусловлена необходимостью обеспечения приоритета прямых явных запретов на доступ, когда $r_{nm} = -1$. Результаты операции $r_{ij} \vee' r_{ji}$ вычисляются по следующей таблице.

r_{ij}	r_{ji}	$r_{ij} \vee' r_{ji}$
1	0	1
1	1	1
1	-1	-1
0	0	0
0	1	1
0	-1	-1
-1	0	-1
-1	1	1
-1	-1	-1

Итоговые групповые права доступа, соответственно можно определять по следующему соотношению:

$$\mathbf{R}^{\Gamma}_{\text{итог}} = \mathbf{W} \otimes (((\mathbf{H}^{\Gamma^S} + \mathbf{I})^T \otimes \mathbf{R}^{\Gamma}) \otimes (\mathbf{H}^S + \mathbf{I})) \quad (3.2.2)$$

И, наконец итоговые индивидуально-групповые права доступа определяются следующим соотношением:

$$\begin{aligned} \mathbf{R}^{\Pi\Gamma}_{\text{итог}} &= \mathbf{R}^{\Pi}_{\text{итог}} \oplus \mathbf{R}^{\Gamma}_{\text{итог}} = \\ &= (\mathbf{R}_{\text{пр}} \otimes (\mathbf{H}^S + \mathbf{I})) \oplus (\mathbf{W} \otimes (((\mathbf{H}^{\Gamma^S} + \mathbf{I})^T \otimes \mathbf{R}^{\Gamma}) \otimes (\mathbf{H}^S + \mathbf{I}))) \end{aligned} \quad (3.3.3)$$

где $\oplus$ - означает операция сложения, аналогичную по смыслу операции асимметричной дизъюнкции (с приоритетом прав по индивидуальным назначениям), результаты которой вычисляются по следующей таблице.

$r^{\Pi}_{\text{итог}}$	$r^{\Gamma}_{\text{итог}}$	$r^{\Pi}_{\text{итог}} \oplus r^{\Gamma}_{\text{итог}}$
1	0	1
1	1	1
1	-1	1
0	0	0
0	1	1
0	-1	-1
-1	0	-1
-1	1	-1
-1	-1	-1

Приведенные соотношения дают возможность вычислять итоговые права доступа, что особенно важно при проектировании сложных систем с большим количеством пользователей и объектов доступа.

Кроме того, представленная теоретико-графовая формализация позволяет ввести некоторые количественные параметры, индицирующие эффективность, оптимальность установленной системы индивидуально-групповых назначений доступа.

В частности, с т.зр. обеспечения конфиденциальности данных, очевидно чем менее будет случаев дублирования прав, т.е. получения одного и того же права доступа несколько раз (по индивидуальному непосредственному назначению к объекту, по иерархической вложенности, через групповые назначения и т.д.), тем очевидно система назначения прав доступа построена более оптимально.

Для анализа системы индивидуально-группового разграничения доступа по данному критерию можно использовать коэффициент дублирования) прав доступа –

$$\mathcal{K}_{\text{дубл}} = \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^M k_{ijk} \quad , \quad (3.3.4)$$

где $k_{ijk} = \frac{1}{L+1} (k_{ijk}^{\text{н}} + k_{ijk}^{\text{г}})$ – коэффициент дублирования права доступа i -го субъекта к j -му объекту по k -му методу доступа;

$k_{ijk}^{\text{н}}$ – коэффициент дублирования права доступа i -го субъекта к j -му объекту по k -му методу доступа по индивидуальным назначениям за счет иерархической структуры системы объектов доступа;

$k_{ijk}^{\text{г}}$ – коэффициент дублирования права доступа i -го субъекта к j -му объекту по k -му методу доступа по групповым назначениям и, в том числе, за счет иерархической структуры системы объектов доступа;

K, N, M, L – количество методов, субъектов, объектов доступа и рабочих групп, соответственно.

Коэффициенты $k_{ijk}^{\text{н}}$ вычисляются на основе матричного соотношения¹ $\mathbf{R}_{\text{к|итог}}^{\text{н}} = \mathbf{R}_{\text{к|гр}} \cdot (\mathbf{H}^{\text{S}} + \mathbf{I})$ и накапливают количество разрешений на доступ по k -му методу за счет назначений напрямую к объекту и по структурной вложенности объектов. Таким образом,

$$k_{ijk}^{\text{н}} = (\mathbf{R}_{\text{к|гр}}^{\text{н}} \cdot (\mathbf{H}^{\text{S}} + \mathbf{I}))_{ij}, \quad (3.3.5)$$

где $\mathbf{R}_{\text{к|гр}}^{\text{н}}$ – массив прав доступа по индивидуальным прямым назначениям субъектам-пользователям на доступ к объектам системы по k -му методу доступа.

Аналогичным образом получаем выражение для вычисления коэффициентов $k_{ijk}^{\text{г}}$ как элементов матрицы $\mathcal{K}_{\text{г}}^{\text{г}}$

$$\mathcal{K}_{\text{г}}^{\text{г}} = \mathbf{W} \cdot (((\mathbf{H}^{\text{rS}} + \mathbf{I})^{\text{T}} \cdot \mathbf{R}_{\text{г}}^{\text{г}}) \cdot (\mathbf{H}^{\text{S}} + \mathbf{I})). \quad (3.3.6)$$

Таким образом, рассматривая различные варианты создания системы рабочих групп, иерархической организации (расположения) объектов доступа, сочетания индивидуальных и групповых назначений доступа, обеспечивающих требуемый набор прав доступа для пользователей системы, администратор/проектировщик, вычисляя по выражениям (3.3.5, 3.3.6) коэффициент дублирования $\mathcal{K}_{\text{дубл}}$ прав доступа, имеет возможность проанализировать и выбрать наиболее оптимальный вариант системы индивидуально-группового доступа.

¹ В данном случае используются обычные матричные операции произведения и сложения.

3.2.2. Пространственно-векторная модель и характеристики системы рабочих групп пользователей

В процессах администрирования системы индивидуально-группового доступа в рабочие группы могут включаться, добавляться или удаляться те или иные пользователи, корректироваться групповые и индивидуальные назначения доступа. На основе величины $\mathcal{K}_{\text{дубл}}$ можно анализировать влияние производимых изменений в системе индивидуально-группового доступа на общую эффективность системы.

Однако величина $\mathcal{K}_{\text{дубл}}$ показывает агрегированный эффект производимых изменений. Но администратору-проектировщику может потребоваться анализ в отношении отдельных частей системы индивидуально-группового разграничения доступа, например, в отношении определенной части рабочих групп. В частности, исходя из принципа минимизации субъектов доступа, могут возникать задачи анализа близости групп между собой по составу пользователей и системе групповых назначений с тем, чтобы наиболее близкие и похожие группы агрегировать, объединять в одну общую группу пользователей.

Решение подобных задач может быть основано на пространственно-векторной модели рабочих групп.

Рабочую группу в многомерном пространстве доступа, измерения которого образуются набором субъектов-пользователей $U(u_1, u_2, \dots, u_N)$, набором объектов доступа $O(o_1, o_2, \dots, o_M)$ и набором методов доступа $R(r_1, r_2, \dots, r_K)$, можно представлять вектором, координаты которого, соответственно, определяют пользователей, входящих в данную группу, и групповые назначения к объектам системы для данной группы.

На этой основе можно ввести понятие и количественные параметры близости рабочих групп.

Определение 3.2.1. Близостью рабочих групп будем называть расстояние в пространстве доступа, образуемом набором пользователей, набором объектов доступа и набором прав доступа, между векторами, идентифицирующими включение пользователей в рабочие группы и групповые назначения доступа.

Для введения количественных параметров (мер) близости более удобным в практическом плане является представление рабочей группы g совокупностью бинарных векторов $(\overrightarrow{G^u}, \overrightarrow{G^o_1}, \overrightarrow{G^o_2}, \dots, \overrightarrow{G^o_k})$, определяющих включение пользователей в данную рабочую группу ($\overrightarrow{G^u}$) и установленные для нее групповые назначения доступа ($\overrightarrow{G^o_1}, \overrightarrow{G^o_2}, \dots, \overrightarrow{G^o_k}$).

В рамках векторно-матричной формализации системы индивидуально-группового доступа векторы $\overrightarrow{G^u}$ и $\overrightarrow{G^o_k}$ можно определить столбцами матрицы $\mathbf{W}$ (см. п.3.2.1) и строками матриц $\mathbf{R}_k^r$, соответственно.

Тогда количественную меру близости можно основывать на скалярном произведении векторов, представляющих соответствующие группы. В результате мера близости между рабочими группами g_i и g_j может выражаться следующей величиной

$$\sigma_{ij}^r = \frac{1}{2N} \sum_{n=1}^N (w_{ni} \cdot w_{nj} + \bar{w}_{ni} \cdot \bar{w}_{nj}) + \frac{1}{2KM} \sum_{k=1}^K \sum_{m=1}^M (r_{imk}^r \cdot r_{jmk}^r + \bar{r}_{imk}^r \cdot \bar{r}_{jmk}^r) \quad (3.3.7)$$

где $\bar{w}$ – дополнение величины w .

Использование в выражении (3.3.7) не только элементов w матрицы $\mathbf{W}$ и r матриц $\mathbf{R}_k$, но их дополнений $\bar{w}$ и $\bar{r}$, обеспечивает определение похожести как по одинаковости вхождения, так и по одинаковости невхождения пользователей, и, аналогично, как по одинаковости назначений доступа, так и по одинаковости отсутствия назначений доступа к тем или иным объектам. В результате величины σ_{ij}^r характеризуют в диапазоне $[0,1]$ степени взаимной близости рабочих групп, составляя элементы матрицы попарных расстояний между рабочими группами в пространстве индивидуально-группового доступа

$$\Omega^r = \begin{pmatrix} \sigma_{11}^r & \sigma_{12}^r & \dots & \sigma_{1L}^r \\ \sigma_{21}^r & \sigma_{22}^r & \dots & \sigma_{2L}^r \\ \dots & & & \\ \sigma_{L1}^r & \sigma_{L2}^r & \dots & \sigma_{LL}^r \end{pmatrix}. \quad (3.3.8)$$

Диагональные элементы σ_{ii}^r матрицы $\mathbf{W}^r$ по соотношению (9.33) равны единице, что означает полную похожесть рабочих групп на самих себя. Нетрудно также видеть, что $\sigma_{ij}^r = \sigma_{ji}^r$. Таким образом, матрица Ω^r симметрична и рефлексивна.

Основываясь на анализе элементов матрицы Ω^r , можно применять пороговые методики определения и слияния наиболее близких групп, что в совокупности с использованием параметра $\mathcal{K}_{\text{дубл}}$ позволяет более эффективно оптимизировать систему индивидуально-группового доступа.

Не менее интересна и важна обратная задача – в пространстве доступа, измерения которого задаются методами $\mathbf{R}$ ($r_1, r_2, \dots, r_K$) и объектами доступа $\mathbf{O}$ ($o_1, o_2, \dots, o_M$), проанализировать близость векторов, характеризующих индивидуальные потребности в получении прав доступа пользователей, и на этой основе в пространстве доступа выделить "сгущения" пользователей, создав по этим сгущениям систему рабочих групп.

Как и рабочие группы, индивидуальных пользователей в пространстве доступа можно представлять системой векторов по требуемым назначе-

ниям доступа – $(\overrightarrow{R_{\text{треб}_{i1}}}, \overrightarrow{R_{\text{треб}_{i2}}}, \dots, \overrightarrow{R_{\text{треб}_{iK}}})$, и, соответственно, в качестве меры близости использовать следующие величины

$$\sigma^u_{ij} = \frac{1}{KM} \sum_{k=1}^K \sum_{m=1}^M (r_{\text{треб}_{imk}} \cdot r_{\text{треб}_{jmk}} + \bar{r}_{\text{треб}_{imk}} \cdot \bar{r}_{\text{треб}_{jmk}}), \quad (3.3.9)$$

где $r_{\text{треб}_{imk}}$ – элементы трехмерного массива $\mathbf{R}_{\text{треб}}$ требуемых пользователям прав доступа.

Элементы σ^u_{ij} составляют матрицу Ω^u попарной близости пользователей по требуемым правам доступа, анализируя которую, можно определять группы наиболее близких по потребностям в доступе пользователей и на этой основе формировать систему рабочих групп однотипного доступа.

Решение данной задачи может быть основано на алгоритмах автоматической классификации.

Существо алгоритмов автоматической классификации заключается в построении по множеству классифицируемых объектов их классов (групп), объединяющих объекты по критерию схожести (близости). Так как образы классов, их количество и прочие характеристики предварительно не известны, то решается двуединая задача – строятся, распознаются классы, и одновременно осуществляется отнесение объектов классификации к конкретным классам.

Исходная детерминированность природы параметров доступа, на основе которых вычисляются элементы матрицы Ω^u , обуславливает геометрический способ классификации, основанный на гипотезе компактности классов в пространстве признаков объектов.

При этом возможен эвристический и оптимизационный подход.

Оптимизационный подход требует задания показателя качества классификации, ограничений по классификации и механизма поиска. Показателем качества классификации в нашем случае может являться минимизация целевого параметра $\mathcal{K}_{\text{дубл}}$, определяемого с учетом групповых назначений по формируемым группам (классам) субъектов. Ограничения могут касаться количества групп, количества субъектов в группах и т. п. Механизм поиска оптимального разбиения субъектов по группам может строиться как на использовании матрицы близости субъектов Ω^u , так и на основе различных вариантов перебора способов объединения субъектов с целью достижения минимума коэффициента $\mathcal{K}_{\text{дубл}}$.

Эвристические алгоритмы основываются на опыте и интуиции человека без явного выражения и максимизации параметра качества классификации. Многие из таких алгоритмов трактуют матрицу близости объектов в качестве матрицы смежности полного взвешенного графа, вершинами которого являются объекты классификации, а веса ребер выражают степень близости объектов классификации – см. рис.3.5.

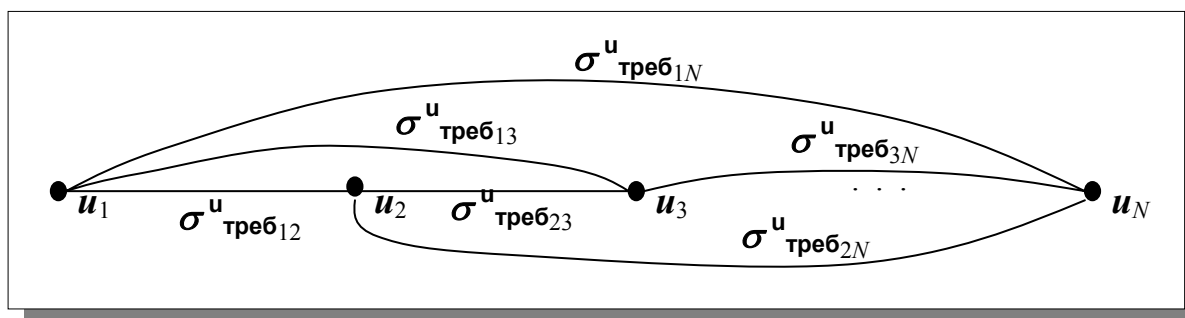


Рис. 3.5. Граф близости субъектов доступа (пользователей)

Известен ряд алгоритмов на графах, позволяющих строить классификационные группировки объектов близости на основе пороговых подходов, которые можно применять и для построения рабочих групп субъектов доступа. В частности, в алгоритме "Граф" по некоторому первоначальному порогу близости Δ_0 отбрасываются те ребра, вес которых меньше Δ_0 . Далее производятся аналогичные шаги с последовательным увеличением первоначального порога до тех пор, пока исходный полный граф не распадется на несколько изолированных подграфов, соответствующих наиболее близким между собой субъектам доступа.

Таким образом, имея вычисленные значения элементов матрицы Ω^u и рассматривая ее как матрицу смежности графа субъектов доступа (пользователей), можно использовать те или иные алгоритмы автоматической классификации, которые могут обеспечить поддержку процессов проектирования системы рабочих групп пользователей.

УПРАЖНЕНИЯ

I. По модели HRU

Задача № 1.1.

Пусть имеется два субъекта: s_1 (доверенный пользователь, *admin*) и s_2 (обычный пользователь, *user*).

Пусть имеется два каталога (объекты) o_1 и o_2 , владельцами которых являются пользователи s_1 и s_2 , соответственно. В каталоге имеется объект o_3 с секретной информацией.

Права доступа в системе заданы исходным состоянием матрицы доступа:

	o_1 - secret	o_2 - no secret	o_3 - secret
s_1	<i>own, r, w, e</i>	<i>r, w, e</i>	<i>own, r, w, e</i>
s_2	-	<i>own, r, w, e</i>	-

Классический сценарий атаки с помощью троянской программы в системах, функционирующих на основе модели **HRU**, заключается в следующем.

1-й шаг. Субъект-злоумышленник s_2 создает в своем каталоге o_2 файл троянской программы $o_{тр}$, дает на него права *чтения* r (*read*), *записи* w (*write*) и *запуска* e (*execute*) для субъекта s_1 , объявляет о каких-либо полезных свойствах и возможностях программы $o_{тр}$ и ожидает запуска доверенным пользователем s_1 троянской программы. Команда перехода и соответствующее изменение матрицы доступа выглядят следующим образом.

Command "создать файл" ($s_2, o_{тр}$):

```

if "write" ∈ [ $s_2, o_2$ ]
then
  Create object  $o_{тр}$ ;
  Enter {"own", "read", "write", "execute"} into [ $s_2, o_{тр}$ ];
end if
if {"read", "write"} ⊆ [ $s_1, o_2$ ]
then
  Enter {"read", "write", "execute"} into [ $s_1, o_{тр}$ ];
end if

```

end command

	o_1 - secret	o_2 - no secret	o_3 - secret	$o_{тр}$ - trojan
s_1	<i>own, r, w, e</i>	<i>r, w, e</i>	<i>own, r, w, e</i>	<i>r, w, e</i>
s_2	-	<i>own, r, w, e</i>	-	<i>own, r, w, e</i>

2-й шаг. Доверенный субъект s_1 запускает троянскую программу $o_{тр}$, которая автоматически приобретает его права доступа.

Command "запустить файл" ($s_1, o_{тр}$):

```

if {"read", "write", "execute"} ⊆ [ $s_1, o_{тр}$ ]
then
  Create subject  $s_{тр}$ ;
  Enter {"read", "write", "execute"} into [ $s_{тр}, o_2$ ];
  Enter {"read", "write", "execute"} into [ $s_{тр}, o_{тр}$ ];
end if
if {"own", "read", "write", "execute"} ⊆ [ $s_1, o_1$ ] and {"own", "read", "write", "execute"} ⊆ [ $s_1, o_3$ ]
then

```

```

    Enter {"read","write","execute"} into [sтр,o1];
    Enter {"read","write","execute"} into [sтр,o3];
end if
end command

```

	<i>o</i> ₁ - secret	<i>o</i> ₂ - no secret	<i>o</i> ₃ - secret	<i>o</i> _{тр} - trojan
<i>s</i> ₁	own,r,w,e	r,w,e	own,r,w,e	r,w,e
<i>s</i> ₂	-	own,r,w,e	-	own,r,w,e
<i>s</i> _{тр}	r,w,e	r,w,e	r,w,e	r,w,e

3-й шаг. На основе скрытых (недекларируемых) возможностей троянская программа *s*_{тр} копирует содержимое секретного файла *o*₃ в несекретный каталог *o*₂, обеспечивая возможность недоверенному пользователю ознакомиться с секретной информацией, прямо ему недоступной.

Command "скопировать файл *o*₃ программой *s*_{тр} в *o*₂" (*s*_{тр},*o*₃,*o*₂):

if "read" ∈ [*s*_{тр},*o*₃] and "write" ∈ [*s*_{тр},*o*₂]

then

Create object *o*';

Enter {"own", "read", "write", "execute"} into [*s*_{тр},*o*'];

Enter "read" into [*s*₂,*o*'];

Read (*s*_{тр},*o*₃);

Write (*s*_{тр},*o*');

end if

Destroy subject *s*_{тр};

end command

	<i>o</i> ₁ - secret	<i>o</i> ₂ - no secret	<i>o</i> ₃ - secret	<i>o</i> _{тр} - trojan	<i>o</i> ' ≡ <i>o</i> ₃ - secret
<i>s</i> ₁	own,r,w,e	r,w,e	own,r,w,e	r,w,e	-
<i>s</i> ₂	-	own,r,w,e	-	own,r,w,e	r

Задание. Построить сценарий аналогичной атаки в том случае, когда доверенный пользователь *s*₁ в исходном состоянии имеет на каталог *o*₂ только права чтения *r*. Отобразите соответствующие последовательности команд перехода и изменений матрицы доступа.

Исходное состояние матрицы доступа.

	<i>o</i> ₁ - secret	<i>o</i> ₂ - no secret	<i>o</i> ₃ - secret
<i>s</i> ₁	own,r,w,e	r	own,r,w,e
<i>s</i> ₂	-	own,r,w,e	-

Решение.

Возможны два варианта.

1-й вариант.

1-й шаг. Пользователь-злоумышленник *s*₂, являясь владельцем каталога *o*₂, дает на него недостающие права доверенному пользователю *s*₁.

Command "дать права на каталог от владельца" (*s*₁,*s*₂,*o*₂):

if "own" ∈ [*s*₂,*o*₂]

then

Enter {"write","execute"} into [*s*₁,*o*₂];

end if

end command

	<i>o</i> ₁ - secret	<i>o</i> ₂ - no secret	<i>o</i> ₃ - secret
<i>s</i> ₁	own,r,w,e	r,w,e	own,r,w,e
<i>s</i> ₂	-	own,r,w,e	-

Последующие шаги аналогичны, описанным выше.

2-й вариант.

1-й шаг. Если по каким-либо соображениям пользователь-злоумышленник s_2 , не может давать доверенному пользователю права *записи* w и *запуска* e на каталог o_2 , то под каким-либо обоснованным предлогом он создает новый каталог o_4 , и, являясь его владельцем, дает на него доверенному пользователю s_1 права *чтения* r , *записи* w и *запуска* e .

Command "создать каталог" (s_2, o_4):

Create object o_4 ;

Enter {"own", "read", "write", "execute"} *into* [s_2, o_4];

Enter {"read", "write", "execute"} *into* [s_1, o_4];

end command

	o_1 - secret	o_2 - no secret	o_3 - secret	o_4 - no secret
s_1	own, r, w, e	r, w, e	own, r, w, e	r, w, e
s_2	-	own, r, w, e	-	own, r, w, e

В дальнейшем файл троянской программы $o_{тр}$ создается в каталоге o_4 и последующие шаги аналогичны, описанным выше.

II. По модели ТАМ

Задача № 2.1

Пусть в системе, функционирующей на основе модели с типизованной матрицей доступа, имеется три типа сущностей (субъектов и объектов доступа) – u , ω и v .

Пусть в начальном состоянии системы имеется субъект s_1 типа u – ($s_1: u$).

Осуществляется переход системы в новое состояние посредством следующей команды:

$\alpha(s_1: u, s_2: \omega, o_1: v)$:

Create object o_1 *of type* v ;

Inter r *into* [s_1, o_1];

Create subject s_2 *of type* ω ;

Inter r' *into* [s_2, o_1];

Create subject s_3 *of type* u ;

Inter r'' *into* [s_3, o_1];

end α

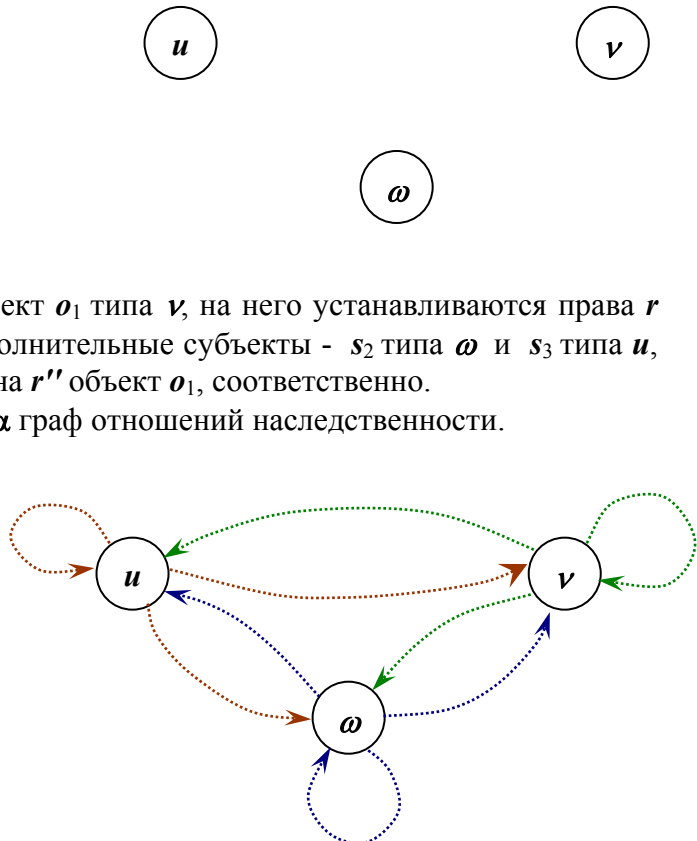
при выполнении которой создается объект o_1 типа v , на него устанавливаются права r для субъекта s_1 , инициализируются дополнительные субъекты – s_2 типа ω и s_3 типа u , им предоставляются права доступа r' и на r'' объект o_1 , соответственно.

Задание. Построить по команде α граф отношений наследственности.

Решение.

В команде α три дочерних типа v , ω и u по отношению к типам, задействованным в теле команды – u , ω , v . Поэтому в графе отношений наследственности возникают следующие дуги – (u, v), (u, ω), (u, u), (ω, v), (ω, ω), (ω, u), (v, v), (v, ω), (v, u).

Следует обратить внимание на то, что на Графе отношений наследст-



венности в результате выполнения команды перехода α возникло несколько циклов длиной 2 и более – $(u \rightarrow \omega \rightarrow v \rightarrow u)$, $(u \rightarrow \omega \rightarrow u)$, $(\omega \rightarrow v \rightarrow \omega)$, $(u \rightarrow v \rightarrow u)$.

Задача № 2.2.

Пусть в системе, функционирующей на основе модели с типизованной матрицей доступа **TAM**, имеется два субъекта доступа: субъект s_1 типа a – $(s_1: a)$ доверенного пользователя (*admin*); субъект s_2 типа u – $(s_2: u)$ обычного пользователя (*user*); а также три объекта доступа: каталог o_1 типа v (*secret*) – $(o_1: v)$, владельцем которого является пользователь s_1 ("own" $\in r_{s_1, o_1}$), несекретный каталог o_2 типа η (*no secret*) – $(o_2: \eta)$, владельцем которого является пользователь s_2 ("own" $\in r_{s_2, o_2}$), секретный файл o_3 типа v – $(o_3: v)$ в каталоге o_1 , владельцем которого также является пользователь s_1 ("own" $\in r_{s_1, o_3}$). Пользователь s_1 имеет также права *чтения*, *записи* и *запуска* на объект o_2 ($\{"read", "write", "execute"\} \subseteq r_{s_1, o_2}$).

В исходном состоянии Графа наследственности имеется четыре вершины.

Задание. Построить Граф отношений наследственности по сценарию атаки троянским конем со стороны пользователя s_2 на секретный файл o_3 .

Решение.

1-й шаг. Субъект-злоумышленник s_2 создает в своем каталоге o_2 файл троянской программы $o_{тр}$ типа η , дает на него права *чтения* r , *записи* w и *запуска* e (*execute*) для субъекта s_1 , объявляет о каких-либо полезных свойствах и возможностях программы $o_{тр}$ и ожидает запуска доверенным пользователем s_1 троянской программы. Команда перехода в нотации модели **TAM** выглядит следующим образом.

$\alpha_1(s_1:a, s_2:u, o_2:\eta, o_{тр}:\eta):$

if "write" $\in r_{s_2, o_2}$

then

Create object $o_{тр}$ **of type** η ;

Enter {"own", "read", "write", "execute"} **into** $r_{s_2, o_{тр}}$;

end if

if {"read", "write"} $\subseteq r_{s_1, o_2}$

then

Enter {"read", "write", "execute"} **into** $r_{s_1, o_{тр}}$;

end if

end α_1

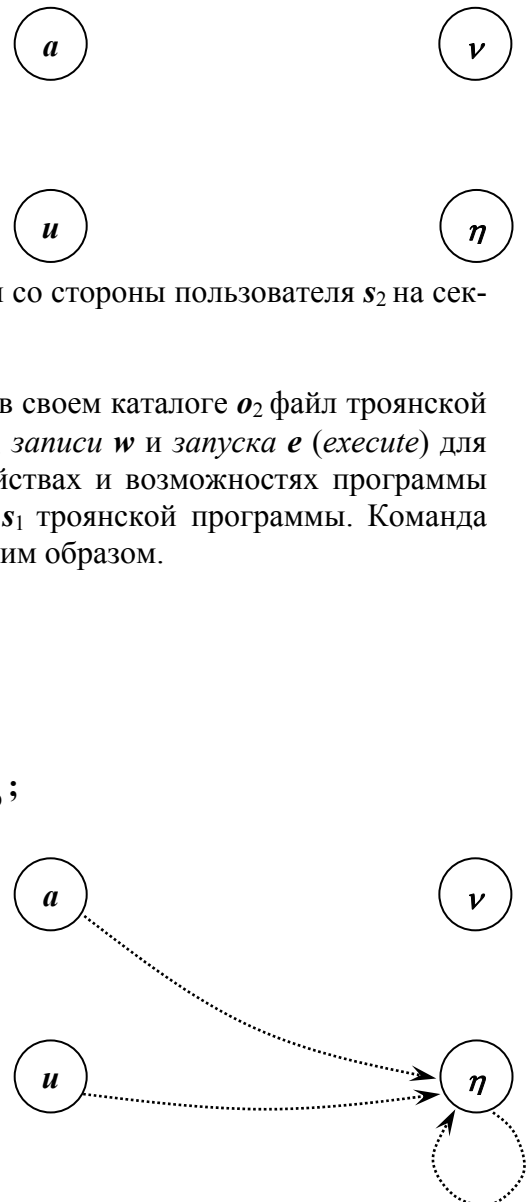
В команде α_1 дочерним типом является тип η по отношению к типам a , u и η , входящим в тело команды. Поэтому в результате перехода системы в новое состояние по команде α_1 в Графе отношений наследственности появляются следующие дуги – (a, η) , (u, η) и (η, η) .

2-й шаг. Доверенный субъект s_1 запускает троянскую программу $o_{тр}$, которая автоматически приобретает его права доступа. **Command** "запустить файл" $(s_1, o_{тр}):$

$\alpha_2(s_1:a, s_{тр}:\eta, o_1:v, o_2:\eta, o_3:v, o_{тр}:\eta):$

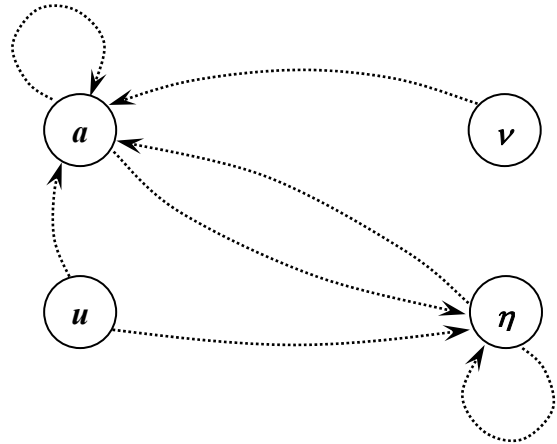
if {"read", "write", "execute"} $\subseteq r_{s_1, o_{тр}}$

then



Create subject $s_{\text{тр}}$ of type a ;
Enter {"read", "write", "execute"} **into** $r_{s_{\text{тр}}, o_2}$;
 end if
 if {"own", "read", "write", "execute"} $\subseteq r_{s_1, o_1}$ and {"own", "read", "write", "execute"} $\subseteq r_{s_1, o_3}$
 then
 Enter {"read", "write", "execute"} **into** $r_{s_{\text{тр}}, o_1}$;
 Enter {"read", "write", "execute"} **into** $r_{s_{\text{тр}}, o_3}$;
 end if
end α_2

В команде α_2 дочерним типом является тип a по отношению к типам a , η , u и v , входящим в тело команды. Поэтому в результате перехода системы в новое состояние по команде α_2 в Графе отношений наследственности появляются следующие дуги – (a, a) , (η, a) , (u, a) и (v, a) .

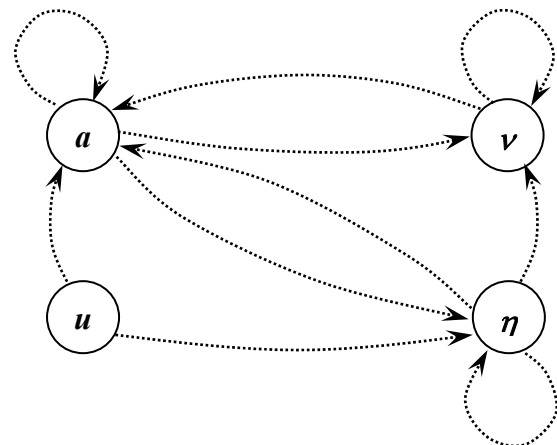


Следует заметить, что уже на данном шаге в Графе отношений наследственности появляется цикл длиной два – $(a \rightarrow \eta)$, отражающий угрозу переноса в объекты несекретного типа η информации от доверенного пользователя s_1 .

3-й шаг. На основе скрытых (недекларируемых) возможностей троянская программа $s_{\text{тр}}$ типа a копирует содержимое секретного файла o_3 типа v в несекретный каталог o_2 типа η , обеспечивая возможность недоверенному пользователю s_2 типа u ознакомиться с секретной информацией типа v , прямо ему недоступной.

$\alpha_3(s_{\text{тр}}:a, o_3:v, o_2:\eta, o':v)$:
 if "read" $\in r_{s_{\text{тр}}, o_3}$ and "write" $\in r_{s_{\text{тр}}, o_2}$
 then
 Create object o' of type v ;
 Enter {"own", "read", "write", "execute"} **into** $r_{s_{\text{тр}}, o'}$;
 Enter "read" **into** r_{s_2, o_1} ;
 Read ($s_{\text{тр}}$ of type a , o_3 of type v);
 Write ($s_{\text{тр}}$ of type a , o' of type v);
 end if
Destroy subject $s_{\text{тр}}$ of type a ;
end α_3

В команде α_3 дочерним типом является тип v по отношению к типам a , η , и v , входящим в тело команды. Поэтому в результате перехода системы в новое состояние по команде α_3 в Графе отношений наследственности появляются следующие дуги – (a, v) , (η, v) и (v, v) .



На графе отношений наследственности появляется еще два цикла, длиной 2 – $(a \rightarrow v \rightarrow a)$, и длиной 3 – $(a \rightarrow \eta \rightarrow v \rightarrow a)$.

Задача № 2.3.

В системе, функционирующей на основе модели с типизованной матрицей доступа **ТАМ** и по условиям задачи 2.2, предложить возможное разрешение проблемы атак

тройным конем на основе ограничений на команды переходов по соотношению дочерних и родительских типов. Дайте физическое обоснование решения и подтвердите его на Графе отношений наследственности.

Решение.

Возможны два варианта.

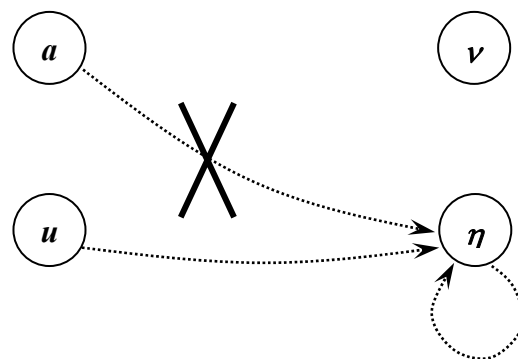
1-й вариант.

Вводится запрет на команды переходов $\alpha_{a\eta}$ в которых тип η (*no secret*) является дочерним для типа a (*admin*).

Это означает, что если при переходе в новое состояние создается объект типа η , то в команде перехода $\alpha_{a\eta}$ не могут быть задействованы субъекты типа a . Тем самым злоумышленник не может, создавая несекретный объект типа η , одновременно давать на него права доступа доверенному пользователю типа a .

Таким образом, вторая часть команды α_1 невозможна.

На Графе отношении наследственности такие ограничения на первом шаге приведут к редукции дуги (a, η) , что, в свою очередь устраняет возможность образования цикла $(a \rightarrow v \rightarrow a)$.



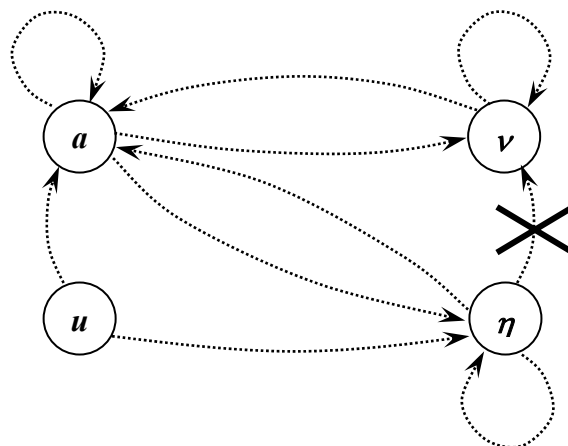
2-й вариант.

Вводится запрет на команды переходов $\alpha_{\eta v}$ в которых тип v является дочерним для типа η .

Это означает, что если при переходе в новое состояние создается секретный объект типа v , то при этом в команде перехода $\alpha_{\eta v}$ не могут быть задействованы несекретные сущности типа η . Тем самым создание секретного файла типа v в несекретном каталоге типа η невозможно.

Таким образом, команда α_3 невозможна.

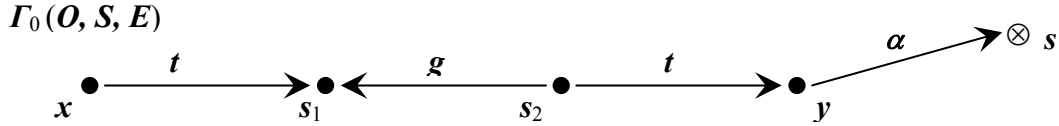
На Графе отношений наследственности такие ограничения приводят к редукции дуги (η, v) , что, в свою очередь устраняет возможность образования цикла $(a \rightarrow \eta \rightarrow v \rightarrow a)$.



III. По модели TAKE-GRANT

Задача № 3.1

Пусть имеется система субъектов и объектов доступа, представленная Графом доступов $\Gamma_0(O, S, E)$, в которой сущности x и y связаны tg -путем.

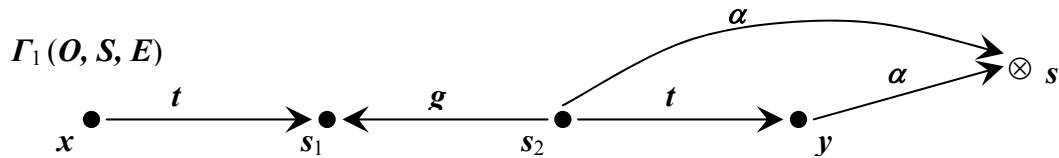


Задание: построить систему команд перехода передачи субъекту x прав доступа α на объект s от субъекта y .

Решение

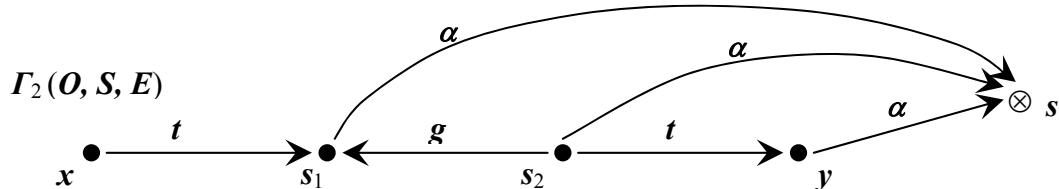
1-й шаг. Субъект s_2 на основе своего права t ("брать") на субъект y берет у него право α на объект s — $\vdash takes(\alpha, s_2, y, s)$.

Преобразование Графа доступов $\Gamma_0(O, S, E)$ в новый Граф $\Gamma_1(O, S, E)$ доступов выглядит следующим образом



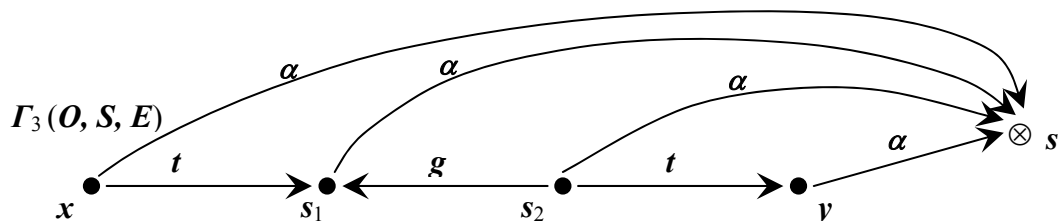
2-й шаг. Субъект s_2 предоставляет на основе права g ("давать") субъекту s_1 свое право α на объект s — $\vdash grants(\alpha, s_2, s_1, s)$.

Преобразование Графа доступов $\Gamma_1(O, S, E)$ в новый Граф $\Gamma_2(O, S, E)$ доступов выглядит следующим образом



3-й шаг. Субъект x берет на основе своего права t ("брать") на субъект s_1 имеющееся у него право α на объект s — $\vdash takes(\alpha, x, s_1, s)$.

Преобразование Графа доступов $\Gamma_2(O, S, E)$ в новый Граф $\Gamma_3(O, S, E)$ доступов выглядит следующим образом

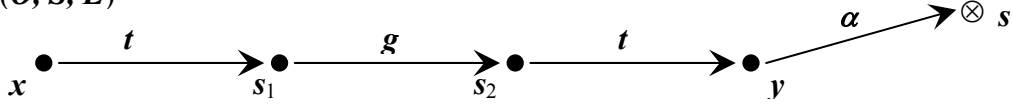


В итоге субъект x через tg -путь получает у субъекта y права доступа α на объект s .

Задача № 3.2

Для иллюстрации возможности передачи прав доступа по **tg**-пути независимо от направления прав **t** и **g**, изменяются в условиях задачи 3.1 направление права между субъектами s_1 и s_2 .

$\Gamma_0(O, S, E)$

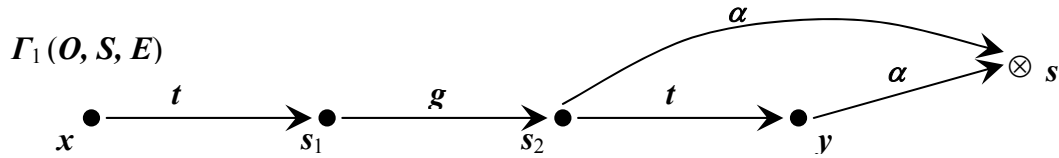


Задание: построить систему команд перехода передачи субъекту x прав доступа α на объект s от субъекта y .

Решение

1-й шаг. Аналогичен в решении задачи 3.1 (субъект s_2 на основе своего права **t** ("брать") на субъект y берет у него право α на объект s – $\vdash \text{takes}(\alpha, s_2, y, s)$.

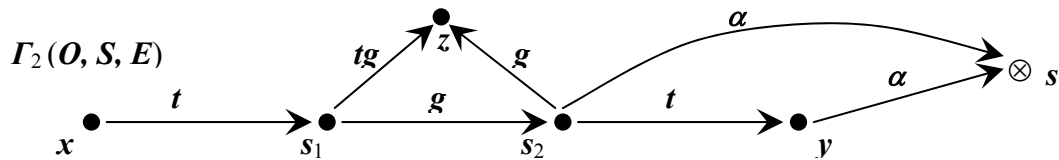
Преобразование Графа доступов $\Gamma_0(O, S, E)$ в новый Граф $\Gamma_1(O, S, E)$ доступов выглядит следующим образом



2-й шаг. Субъект s_1 создает субъект z с правами на него **tg** и предоставляет на него право **g** ("давать") субъекту s_2 .

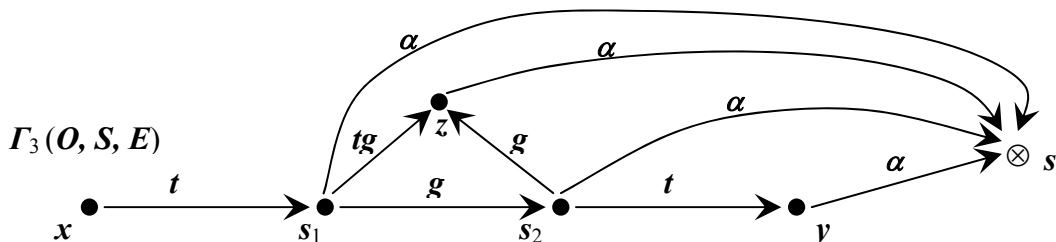
$\vdash \text{create}(tg, s_1, z)$
 $\vdash \text{grants}(g, s_1, s_2, z)$

Преобразование Графа доступов $\Gamma_1(O, S, E)$ в новый Граф $\Gamma_2(O, S, E)$ доступов выглядит следующим образом



3-й шаг. Субъект s_2 предоставляет имеющееся у него право α на объект s субъекту z , а субъект s_1 , используя свое право **t** на субъект z берет у него право α на объект s – $\vdash \text{grants}(\alpha, s_2, z, s)$
 $\vdash \text{takes}(g, s_1, s_2, z)$

Преобразование Графа доступов $\Gamma_2(O, S, E)$ в новый Граф $\Gamma_3(O, S, E)$ доступов выглядит следующим образом

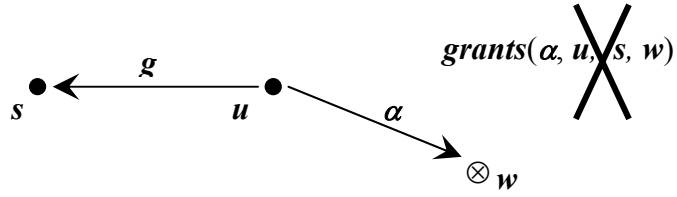


4-й шаг. Аналогичен 3-му шагу в решении задачи 3.1.

Задача № 3.3. Похищение прав доступа

Пусть имеется система субъектов и объектов доступа, представленная Графом доступов $\Gamma_0(O, S, E)$.

$\Gamma_0(O, S, E)$



Установленная для системы политика безопасности запрещает любым субъектам (владельцам) предоставлять право α на "свои" объекты другим субъектам (но не запрещает субъектам, которые владеют правами t ("брать") на какие-либо субъекты брать у них права на их объекты).

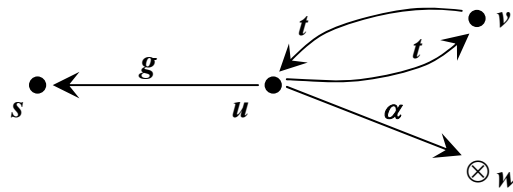
Кроме субъекта s , субъект u может быть связан tg -путем с другими субъектами.

Задание: построить систему команд получения субъектом s прав доступа α на объект w от субъекта u , при условии того, что команда $grants(\alpha, u, s, w)$ не может быть задействована.

Решение

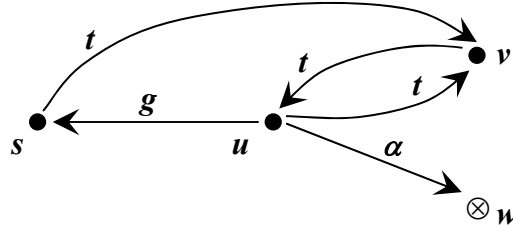
1-й шаг. Пусть найдется субъект v , имеющий право t на объект u . При этом субъект u также имеет право t на субъект v .

$\Gamma_1(O, S, E)$

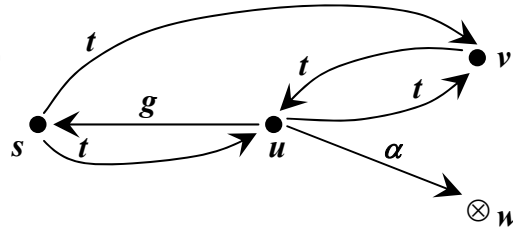


2-й шаг. Субъект u предоставляет субъекту s право t на субъект v – $\vdash grants(t, u, s, v)$.

$\Gamma_2(O, S, E)$



$\Gamma_3(O, S, E)$

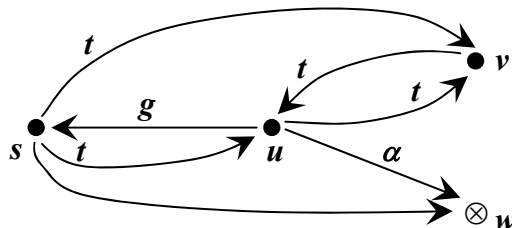


v).

3-й шаг. Субъект s берет у субъекта v право t на субъект u – $\vdash takes(t, s, v, u)$.

4-й шаг. Субъект s берет (похищает) у субъекта u право α на объект w – $\vdash takes(\alpha, s, u, w)$.

$\Gamma_4(O, S, E)$



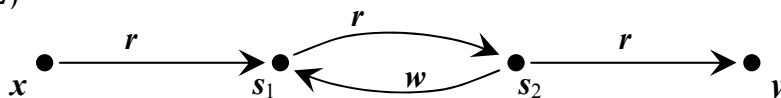
u, w).

IV. По расширенной модели TAKE-GRANT

Задача № 4.1.

Пусть имеется система субъектов и объектов доступа, представленная Графом доступов $\Gamma_0(O, S, E)$,

$$\Gamma_0(O, S, E)$$



Пусть неявные каналы чтения, генерируемые различными командами "де-факто" имеют следующую стоимость: - $r_{spy} = 1$, $r_{post} = 2$, $r_{find} = 3$ и $r_{pass} = 4$.

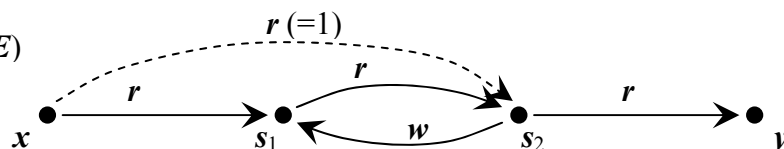
Задание: Применяя команды "де-факто" сгенерировать все возможные неявные каналы чтения субъектом x информации из субъекта y , и сравнить их стоимость.

Решение

1-й вариант

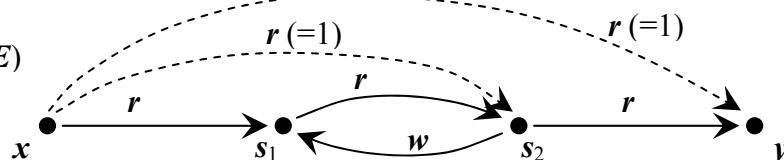
1-й шаг. $\vdash_{spy}(x, s_1, s_2)$.

$$\Gamma_1(O, S, E)$$



2-й шаг. $\vdash_{spy}(x, s_2, s_y)$.

$$\Gamma_2(O, S, E)$$

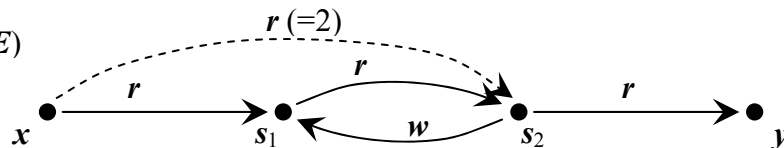


Общая стоимость затрат на реализацию неявного канала по первому варианту - $1 + 1 = 2$.

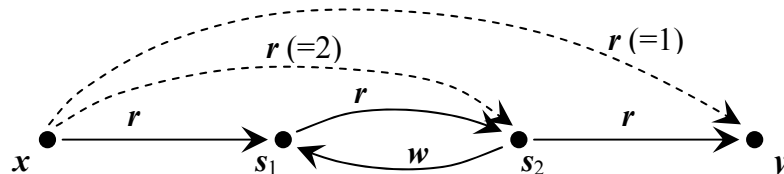
2-й вариант

1-й шаг. $\vdash_{post}(x, s_1, s_2)$.

$$\Gamma_1(O, S, E)$$



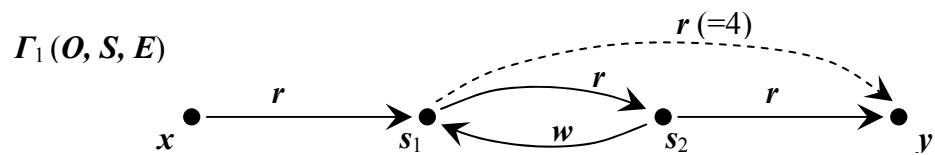
2-й шаг. $\vdash_{spy}(x, s_2, s_y)$.



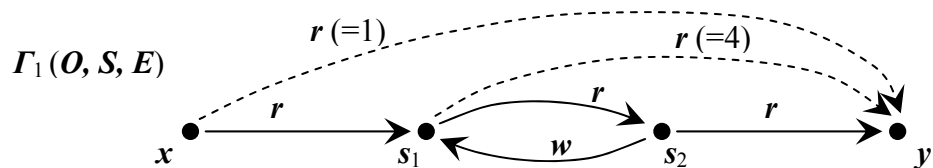
Общая стоимость затрат на реализацию неявного канала по второму варианту - $2 + 1 = 3$.

3-й вариант

1-й шаг. $\vdash_{pass}(s_1, s_2, y)$.



2-й шаг. $\vdash_{spy}(x, s_1, y)$.



Общая стоимость затрат на реализацию неявного канала по третьему варианту - $4 + 1 = 5$.

V. По модели Белла-ЛаПадуллы

Пусть имеется мандатная система доступа $\mathcal{D}(v_0, Q, F_T)$, в которой решетка уровней безопасности Λ_L является линейной и имеет три уровня – l_1, l_2, l_3 ; $l_1 > l_2 > l_3$; $l_1 > l_3$.

Пусть имеется следующая система субъектов (пользователей) доступа:

u_1 – администратор системы;

u_2 – руководитель предприятия;

u_3 – делопроизводитель;

u_4 – user, т.е. рядовой непривилегированный пользователь.

Пусть имеется следующая система объектов доступа:

o_1 – системное ПО;

o_2 – документ "Стратегия выхода предприятия на новые рынки сбыта продукции";

o_3 – документ "Приказ о поощрении работников по случаю Дня Предприятия";

o_4 – АИС "Борей" (прием, обработка и исполнение заказов клиентов) (ПО и БД).

Задача № 5.1.

Обосновать и составить систему уровней допусков пользователей, грифов секретности объектов доступа и матрицу доступа $A[u, o]$.

Решение

1. Начинать необходимо с установки уровней безопасности объектов доступа, чтобы на этой основе с учетом правил *NRU* и *NWD* определить уровни допуска субъектов доступа.

Очевидно, что наиболее конфиденциальная информация содержится в объекте o_2 . Таким образом $f_L(o_2) = l_1$.

Также очевидно, что в системном ПО каких-либо секретов предприятия (кроме самого факта использования конкретного ПО) не содержится. Таким образом $f_L(o_1) = l_3$.

Объект o_3 коммерческих, производственных и др. секретов также не содержит, кроме персональных данных работников предприятия, и по своей сути поэтому должен быть доступен всем работникам предприятия. Поэтому $f_L(o_3) = l_3$.

АИС "Борей" содержит коммерческие секреты предприятия, но, конечно же, в отличие от объекта o_2 , не самого высокого уровня. Поэтому $f_L(o_4) = l_2$.

2. Основываясь на анализе функций и полномочий пользователей, с учетом правил *NRU* и *NWD* устанавливаем уровни допуска субъектов доступа.

Поскольку функции и полномочия администратора системы заключаются, прежде всего, в установке и сопровождении ПО, то он должен иметь возможность вносить при необходимости изменения в ПО. Учитывая правило *NWD*, получаем $f_L(u_1) = l_3$.

Наивысшие полномочия у руководителя предприятия, поэтому $f_L(u_2) = l_1$.

Делопроизводитель должен иметь возможность готовить конфиденциальные документы (уровней l_1 и l_2). Учитывая правило *NWD*, получаем $f_L(u_3) = l_2$.

У непривилегированного пользователя соответственно наименьшие полномочия, поэтому $f_L(u_4) = l_3$.

3. Непосредственное применение правил *NRU* и *NWD* дает следующую матрицу доступа

$$A'[u, o] =$$

	o_1	o_2	o_3	o_4	
	r, w	w	r, w	w	u_1
	r	r, w	r	r	u_2
	r	w	r	r, w	u_3
	r	w	r, w	w	u_4

4. Очевидно, что права некоторых пользователей по правилам *NRU* и *NWD* (матрица $A'[u,o]$) являются избыточными.

Так, для пользователя u_1 (администратор) и для пользователя u_4 не являются необходимыми и оправданными права записи в объекты o_2 , o_3 и o_4 .

Для пользователя u_3 (делопроизводитель) не нужны права записи в объект o_2 (после того, как документ утвержден и введен в действие).

Таким образом получаем матрицу доступа, "уточняющую" и корректирующую права доступа, получаемые пользователями по правилам *NRU* и *NWD*.

$$A[u,o] =$$

o_1	o_2	o_3	o_4	
r,w	—	r	—	u_1
r	r,w	r	r	u_2
r	—	r	r,w	u_3
r	—	r	—	u_4

Задача № 5.2.

Составить и обосновать систему допусков и грифов секретности для двух состояний системы:

Состояние I – Подготовка (разработка) документа o_2 .

Состояние II – Документ o_2 утвержден и введен в действие.

Является ли переход системы из состояния I в состояние II безопасным по МакЛину?

Решение

1. Состояние II описано в результатах решения задачи 5.1.

2. Документ готовится по заданию руководителя (пользователь u_2) делопроизводителем (пользователь u_3). Поскольку уже на этапе подготовки он может содержать наиболее конфиденциальные сведения, то $f_L(o_2) = l_1$. Тогда для того, чтобы пользователь u_3 мог вносить в него данные, по правилам *NRU* и *NWD* необходимо $f_L(u_3) = l_1$.

Однако уровень допуска $f_L(u_3) = l_1$ по правилам *NRU* и *NWD* автоматически приведет к невозможности (запрету) для делопроизводителя (пользователь u_3) исполнять документы по заказам клиентов (запись в объект o_4). В результате для обеспечения непрерывности делового цикла в состоянии I необходимо также $f_L(o_4) = l_1$.

Матрица доступа в состоянии I выглядит следующим образом:

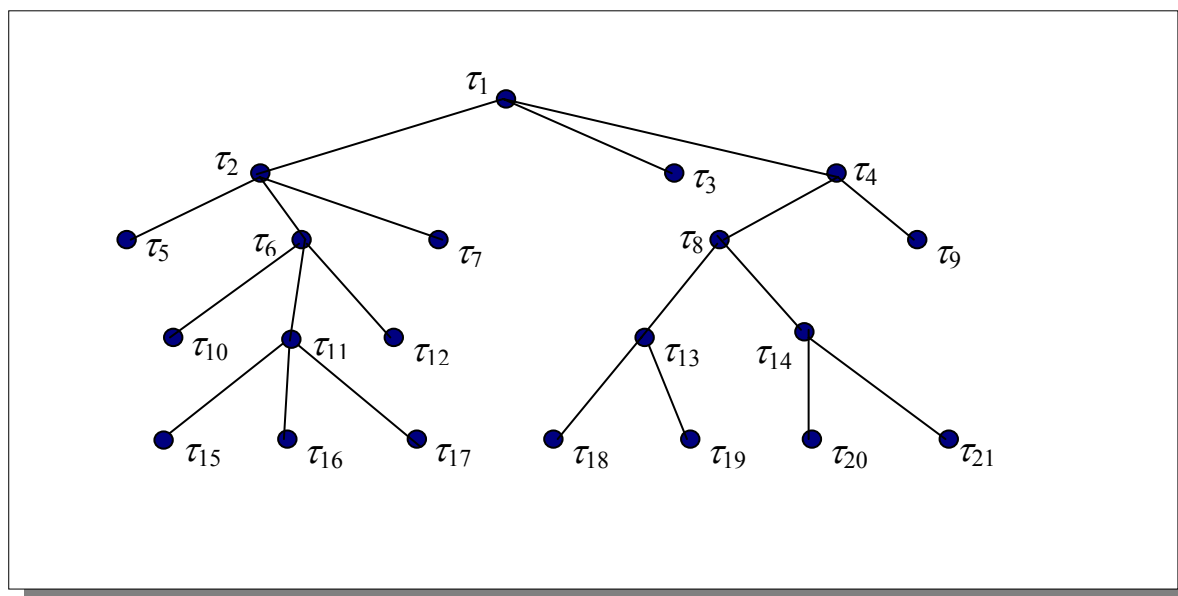
$$A_I[u,o] =$$

o_1	o_2	o_3	o_4	
r,w	—	r	—	u_1
r	r,w		r	u_2
r	r,w	r	r,w,r	u_3
r	—	r	—	u_4

3. Поскольку при переходе из состояния I в состояние II изменяются сразу два аспекта безопасности – одновременно снижается уровень допуска пользователя u_3 и гриф секретности объекта o_4 , – то по условиям Теоремы безопасности МакЛина такой переход небезопасен. В частности в состоянии I пользователь одновременно работая по правам r,w с объектами o_2 , o_4 может перенести конфиденциальную информацию из объекта o_2 в объект o_4 , который при переходе в состояние II станет доступным по чтению пользователям с уровнем допуска l_2 , и тем самым создадутся условия информационного потока "сверху-вниз".

VI. По модели тематического разграничения доступа на основе иерархических рубрикаторов

Пусть имеется иерархический тематический рубрикатор. Для тематической классификации сущностей системы (субъектов и объектов доступа) использованы мульти-рубрики:



$\mathcal{T}^M_1 = \{\tau_5, \tau_6\}$; $\mathcal{T}^M_2 = \{\tau_{11}, \tau_{12}\}$; $\mathcal{T}^M_3 = \{\tau_3, \tau_8\}$; $\mathcal{T}^M_4 = \{\tau_6, \tau_8\}$; $\mathcal{T}^M_5 = \{\tau_{12}, \tau_{13}\}$; $\mathcal{T}^M_6 = \{\tau_{13}, \tau_9\}$; $\mathcal{T}^M_7 = \{\tau_{15}, \tau_{16}, \tau_{14}\}$.

Задача № 6.1.

Определить отношения доминирования (уже, шире, несравнимо) между следующими мультирубриками:

$\mathcal{T}^M_2$ и $\mathcal{T}^M_1$; $\mathcal{T}^M_7$ и $\mathcal{T}^M_4$; $\mathcal{T}^M_5$ и $\mathcal{T}^M_3$; $\mathcal{T}^M_6$ и $\mathcal{T}^M_4$.

Решение.

$\mathcal{T}^M_2 \leq \mathcal{T}^M_1$; $\mathcal{T}^M_7 \leq \mathcal{T}^M_4$; $\mathcal{T}^M_5 \diamond \mathcal{T}^M_3$; $\mathcal{T}^M_6 \diamond \mathcal{T}^M_4$

Задача № 6.2.

Построить объединение следующих мультирубрик:

$\mathcal{T}^M_3 \cup_m \mathcal{T}^M_5$; $\mathcal{T}^M_4 \cup_m \mathcal{T}^M_6$; $\mathcal{T}^M_2 \cup_m \mathcal{T}^M_3$; $\mathcal{T}^M_1 \cup_m \mathcal{T}^M_2$; $\mathcal{T}^M_4 \cup_m \mathcal{T}^M_7$.

Решение.

$\mathcal{T}^M_3 \cup_m \mathcal{T}^M_5$

1. Шаг. Производим теоретико-множественное объединение наборов рубрик соответствующих мультирубрик. Получаем первый промежуточный объединенный набор - $\{\tau_3, \tau_8, \tau_{12}, \tau_{13}\}$;

В объединенном наборе рубрика τ_{13} доминируется (ниже по иерархии) рубрикой τ_8 и поэтому исключается из объединенного набора - $\{\tau_3, \tau_8, \tau_{12}\}$;

2. Шаг. Проверяем второй промежуточный набор - $\{\tau_3, \tau_8, \tau_{12}\}$, на предмет наличия полного набора сыновей какой-либо рубрики. Таковых нет.

3. Результат - $T^M_3 \cup_M T^M_5 = \{\tau_3, \tau_8, \tau_{12}\}$. Следует обратить внимание, что результат объединения мультирубрик сам является мультирубрикой, т.е. в нем нет подчиненных рубрик и полного набора сыновей какой-либо рубрики.

$$T^M_4 \cup_M T^M_6$$

1. Шаг. $\{\tau_6, \tau_8, \tau_{13}, \tau_9\}$. τ_{13} доминируется τ_8 . $\{\tau_6, \tau_8, \tau_9\}$.

2. Шаг. Рубрики τ_8, τ_9 является полным набором сыновей рубрики τ_4 . Поэтому во втором промежуточном наборе вместо рубрик τ_8, τ_9 ставим. $\{\tau_4, \tau_6\}$.

3. Результат - $T^M_4 \cup_M T^M_6 = \{\tau_4, \tau_6\}$.

$$T^M_2 \cup_M T^M_3$$

1. Шаг. $\{\tau_{11}, \tau_{12}, \tau_3, \tau_8\}$. Доминируемых рубрик нет.

2. Шаг. $\{\tau_{11}, \tau_{12}, \tau_3, \tau_8\}$. Полных наборов сыновей нет.

3. Результат - $T^M_2 \cup_M T^M_3 = \{\tau_{11}, \tau_{12}, \tau_3, \tau_8\}$.

$$T^M_1 \cup_M T^M_2$$

1. Шаг. $\{\tau_5, \tau_6, \tau_{11}, \tau_{12}\}$. τ_{11}, τ_{12} доминируются τ_6 . $\{\tau_5, \tau_6\}$.

2. Шаг. $\{\tau_5, \tau_6\}$. Полных наборов сыновей нет.

3. Результат - $T^M_1 \cup_M T^M_2 = \{\tau_5, \tau_6\} = T^M_1$. Следует обратить внимание, что объединенная мультирубрика всегда доминирует над объединяемыми и является их наименьшей верхней границей.

$$T^M_4 \cup_M T^M_7$$

1. Шаг. $\{\tau_6, \tau_8, \tau_{15}, \tau_{16}, \tau_{14}\}$. τ_{14} доминируются τ_8 . τ_{15}, τ_{16} доминируются τ_6 . $\{\tau_6, \tau_8\}$.

2. Шаг. $\{\tau_6, \tau_8\}$. Полных наборов сыновей нет.

3. Результат - $T^M_4 \cup_M T^M_7 = \{\tau_6, \tau_8\} = T^M_4$.

Задача № 6.3.

Построить пересечение следующих мультирубрик:

$$T^M_3 \cap_M T^M_5; T^M_4 \cap_M T^M_6; T^M_2 \cap_M T^M_3; T^M_1 \cap_M T^M_2; T^M_4 \cap_M T^M_7.$$

Решение.

$$T^M_3 \cap_M T^M_5$$

1. Шаг. $\{\tau_3, \tau_8\} \leftrightarrow \{\tau_{12}, \tau_{13}\}$. В наборе мультирубрики $T^M_3 = \{\tau_3, \tau_8\}$ оставляем те рубрики, которые доминируются (ниже по иерархии) какими-либо рубриками мультирубрики $T^M_5 = \{\tau_{12}, \tau_{13}\}$. Таковых нет. Таким образом, первый набор - $\emptyset$.

2. Шаг. $\{\tau_3, \tau_8\} \leftrightarrow \{\tau_{12}, \tau_{13}\}$. В наборе мультирубрики $T^M_5 = \{\tau_{12}, \tau_{13}\}$ оставляем те рубрики, которые доминируются (ниже по иерархии) какими-либо рубриками мультирубрики $T^M_3 = \{\tau_3, \tau_8\}$. Таковой рубрикой является τ_{13} (доминируется рубрикой τ_8). Таким образом, второй набор - $\{\tau_{13}\}$.

3. Шаг. Производим теоретико-множественное объединение первого и второго набора. Результат - $T^M_3 \cup_M T^M_5 = \{\tau_{13}\}$.

$$T^M_4 \cap_M T^M_6$$

1. Шаг. $\{\tau_6, \tau_8\} \leftrightarrow \{\tau_{13}, \tau_9\}$. От набора первой мультирубрики - $\emptyset$

2.Шаг. $\{\tau_6, \tau_8\} \leftrightarrow \{\tau_{13}, \tau_9\}$. От набора второй мультирубрики - $\{\tau_{13}\}$.

3.Шаг. Результат - $\mathcal{T}_4^M \cap_M \mathcal{T}_6^M = \{\tau_{13}\}$.

$$\mathcal{T}_2^M \cap_M \mathcal{T}_3^M$$

1.Шаг. $\{\tau_{11}, \tau_{12}\} \leftrightarrow \{\tau_3, \tau_8\}$. От набора первой мультирубрики - $\emptyset$.

2.Шаг. $\{\tau_{11}, \tau_{12}\} \leftrightarrow \{\tau_3, \tau_8\}$. От набора второй мультирубрики - $\emptyset$.

3.Шаг. Результат - $\mathcal{T}_4^M \cap_M \mathcal{T}_6^M = \emptyset$.

$$\mathcal{T}_1^M \cap_M \mathcal{T}_2^M$$

1.Шаг. $\{\tau_5, \tau_6\} \leftrightarrow \{\tau_{11}, \tau_{12}\}$. От набора первой мультирубрики - $\emptyset$.

2.Шаг. $\{\tau_5, \tau_6\} \leftrightarrow \{\tau_{11}, \tau_{12}\}$. От набора второй мультирубрики - $\{\tau_{11}, \tau_{12}\}$.

3.Шаг. Результат - $\mathcal{T}_1^M \cap_M \mathcal{T}_2^M = \{\tau_{11}, \tau_{12}\}$. Следует обратить внимание, что пересечение мультирубрик является мультирубрикой и их наибольшей нижней границей.

$$\mathcal{T}_4^M \cap_M \mathcal{T}_7^M$$

1.Шаг. $\{\tau_6, \tau_8\} \leftrightarrow \{\tau_{15}, \tau_{16}, \tau_{14}\}$. От набора первой мультирубрики - $\emptyset$.

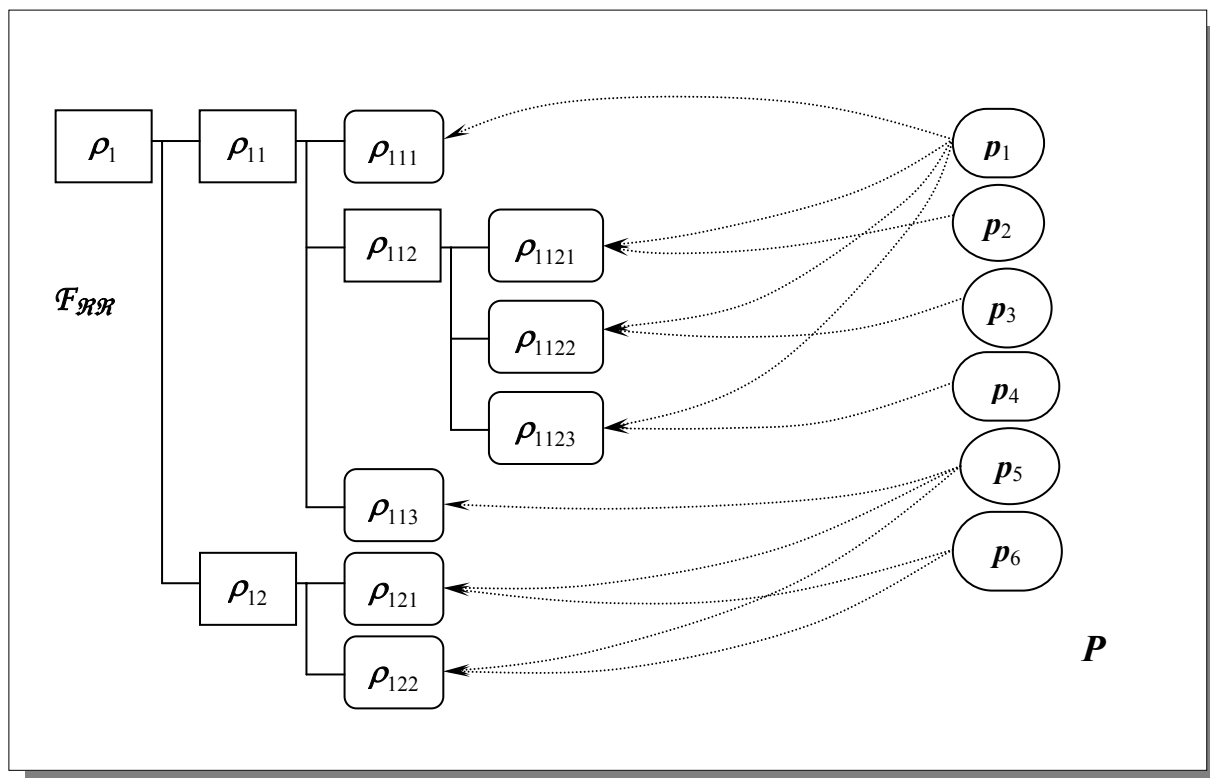
2.Шаг. $\{\tau_6, \tau_8\} \leftrightarrow \{\tau_{15}, \tau_{16}, \tau_{14}\}$. От набора второй мультирубрики - $\{\tau_{15}, \tau_{16}, \tau_{14}\}$.

3.Шаг. Результат - $\mathcal{T}_4^M \cap_M \mathcal{T}_7^M = \{\tau_{15}, \tau_{16}, \tau_{14}\}$.

VII. По модели ролевого доступа при иерархически организованной системе ролей

Задача № 7.1.

Пусть имеется система иерархически организованных ролей $\mathcal{R}$ ($\rho \in \mathcal{R}$), представленная на рис. Ролям назначены полномочия из конечного множества P ($p \in P$).



Определить тип наделения ролей полномочиями (листовой таксономический, листовой нетаксономический, иерархически охватный).

Определить полномочия роли ρ_{11} .

Решение.

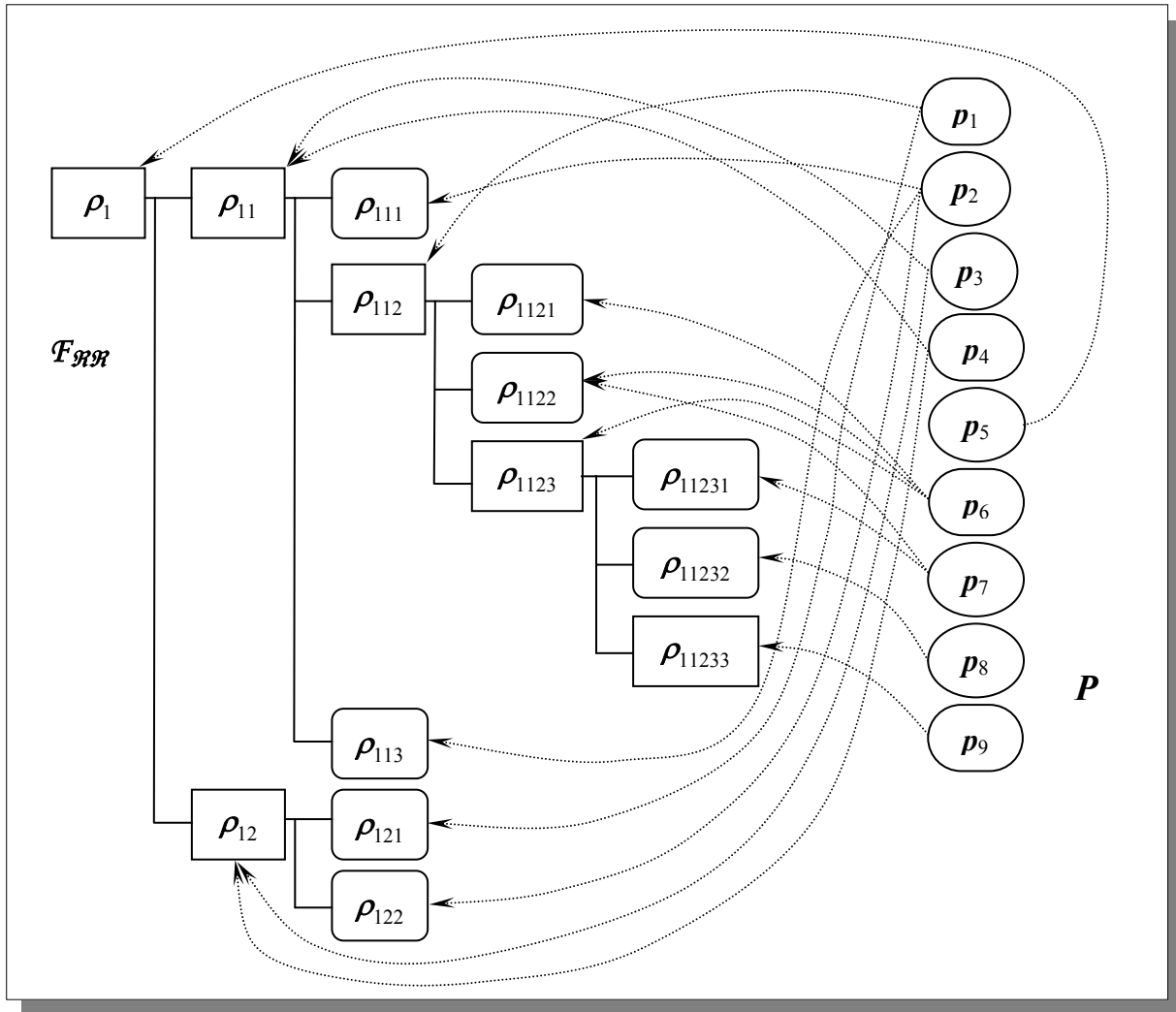
Ввиду того, что полномочия непосредственно назначаются только листовым ролям, то имеем листовой подход.

Так как, одно и то же полномочие, например p_1 , назначается сразу нескольким ролям, то имеем нетаксономический листовой подход.

$$\begin{aligned}
 P_{\rho_{11}} &= P_{\rho_{111}} \cup P_{\rho_{112}} \cup P_{\rho_{113}} = p_1 \cup (P_{\rho_{1121}} \cup P_{\rho_{1122}} \cup P_{\rho_{1123}}) \cup p_5 = \\
 &= p_1 \cup ((p_1 \cup p_2) \cup (p_1 \cup p_3) \cup (p_1 \cup p_4)) \cup p_5 = p_1 \cup p_2 \cup p_3 \cup p_4 \cup p_5
 \end{aligned}$$

Задача № 7.2.

Пусть имеется система иерархически организованных ролей $\mathcal{R}$ ($\rho \in \mathcal{R}$), представленная на рис. Ролям назначены полномочия из конечного множества P ($p \in P$).



Определить тип наделения ролей полномочиями (листовой таксономический, листовой нетаксономический, иерархически охватный).

Определить полномочия роли ρ_{11} .

Решение.

Ввиду того, что полномочия непосредственно назначаются и листовым и узловым ролям, то имеем иерархически охватный подход.

$$\begin{aligned}
 P_{\rho_{11}} &= p_3 \cup p_4 \cup (P_{\rho_{111}} \cup P_{\rho_{112}} \cup P_{\rho_{113}}) = p_3 \cup p_4 \cup (p_2 \cup (p_1 \cup (P_{\rho_{1121}} \cup P_{\rho_{1122}} \cup P_{\rho_{1123}}))) \cup p_2) \\
 &= \\
 &= p_3 \cup p_4 \cup (p_2 \cup (p_1 \cup (p_6 \cup (p_6 \cup p_7) \cup (p_6 \cup (p_7 \cup p_8 \cup p_9)))) \cup p_2) = \\
 &= p_1 \cup p_2 \cup p_3 \cup p_4 \cup p_6 \cup p_7 \cup p_8 \cup p_9
 \end{aligned}$$

Задача № 7.3.

Пусть имеется система иерархически организованных ролей $\mathcal{R}$ ($\rho \in \mathcal{R}$), представленная на рис. Ролям на основе иерархически охватного подхода назначены полномочия из конечного множества P ($p \in P$).

При предположении, что определенные полномочия могут быть назначены только ролям определенного уровня иерархии, определить возможный порядок (отношение доминирования) на множестве полномочий.

Решение.

Будем исходить из того, что одно полномочие p_1 доминирует над другим полномочием ($p_2 \leq p_1$), если роль, которой назначено второе полномочие p_2 , в иерархии ролей находится в подчинении первой роли p_1 (прямо или по дереву ролей) - $p_2 \leq p_1$.

Тогда определение порядка полномочий будем производить от полномочий корневой роли и т.д. вниз по иерархии ролей.

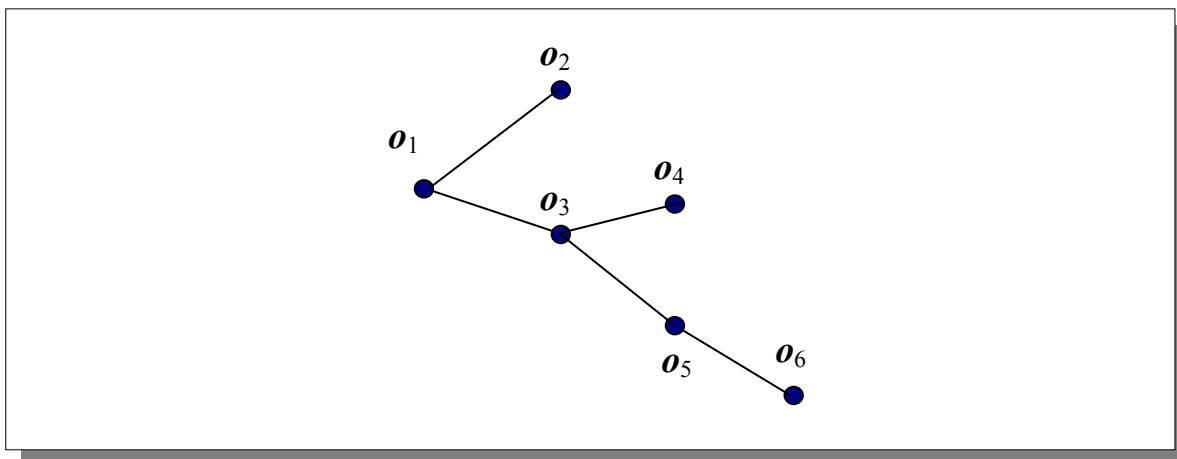
Первый уровень полномочий	-	p_5 .
Второй уровень полномочий	-	p_3, p_4 .
Третий уровень полномочий	-	p_1, p_2 .
Четвертый уровень полномочий	-	p_6, p_7 .
Пятый уровень полномочий	-	p_7, p_8, p_9 .

Следует обратить внимание, что порядок на множестве полномочий частичный и нелинейный. Так полномочие p_7 назначено одновременно ролям p_{1122} и p_{11231} .

VIII. По модели анализа индивидуально-групповых систем назначения доступа к иерархически организованным объектам доступа

Задача № 8.1.

Пусть имеется иерархически организованная система объектов доступа.



Составить матрицу смежности объектов доступа $\mathbf{H}$ (строка – куда; столбец – кто входит; диагональные элементы равны 0) и матрицу итоговой достижимости $\mathbf{H}^S$ (за один шаг, за два шага и т.д.).

Решение.

$$\mathbf{H} =$$

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	1	1	0	0	0
o_2	0	0	0	0	0	0
o_3	0	0	0	1	1	0
o_4	0	0	0	0	0	0
o_5	0	0	0	0	0	1
o_6	0	0	0	0	0	0

$$\mathbf{H}^2 =$$

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	0	0	1	1	0
o_2	0	0	0	0	0	0
o_3	0	0	0	0	0	1
o_4	0	0	0	0	0	0
o_5	0	0	0	0	0	0
o_6	0	0	0	0	0	0

$$\mathbf{H}^3 =$$

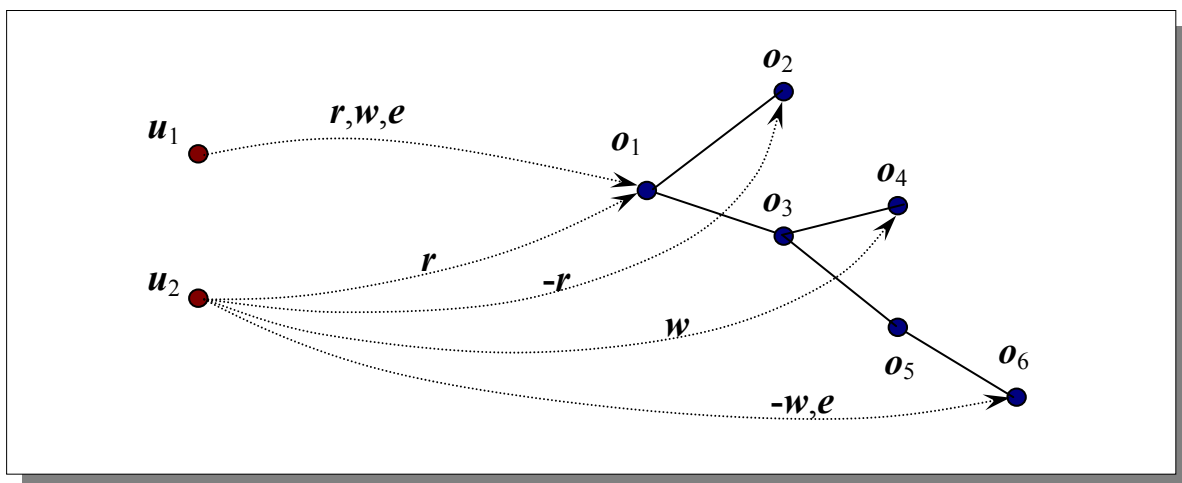
	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	0	0	0	0	1
o_2	0	0	0	0	0	0
o_3	0	0	0	0	0	0
o_4	0	0	0	0	0	0
o_5	0	0	0	0	0	0
o_6	0	0	0	0	0	0

$$\mathbf{H}^S = \mathbf{H} + \mathbf{H}^2 + \mathbf{H}^3 =$$

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	1	1	1	1	1
o_2	0	0	0	0	0	0
o_3	0	0	0	1	1	1
o_4	0	0	0	0	0	0
o_5	0	0	0	0	0	1
o_6	0	0	0	0	0	0

Задача № 8.2.

Пусть имеется иерархически организованная система объектов доступа и два пользователя u_1 и u_2 . Назначения доступа показаны на рис.



Определить итоговые права доступа.

Решение.

Имеем следующие матрицы прав доступа по непосредственным назначениям.

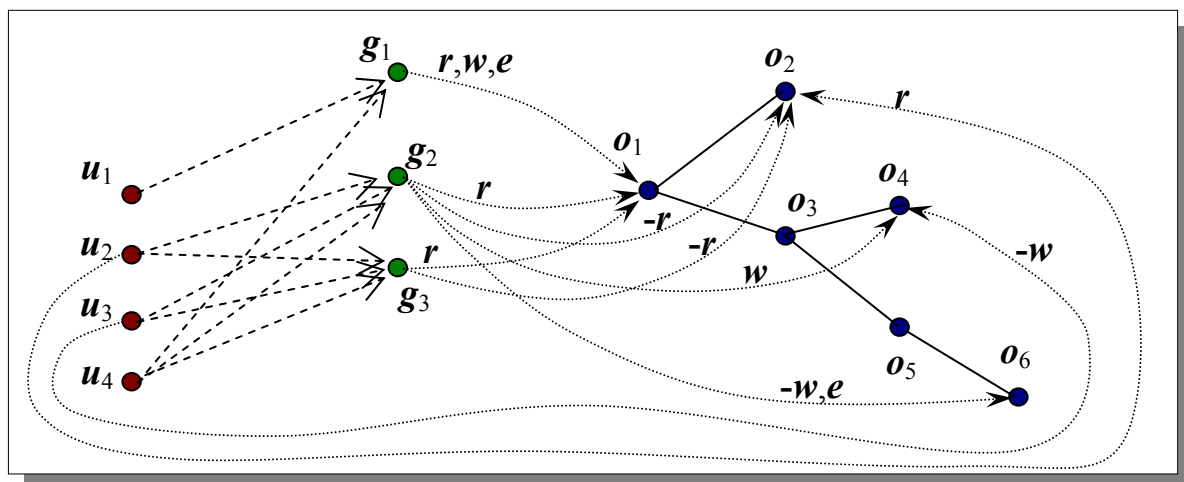
$$\mathbf{R}_r = \begin{array}{c|cccccc} & o_1 & o_2 & o_3 & o_4 & o_5 & o_6 \\ \hline u_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_2 & 1 & -1 & 0 & 0 & 1 & 0 \end{array}$$

$$\mathbf{R}_w = \begin{array}{c|cccccc} & o_1 & o_2 & o_3 & o_4 & o_5 & o_6 \\ \hline u_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_2 & 0 & 0 & 0 & 1 & 0 & -1 \end{array}$$

$$\mathbf{R}_e = \begin{array}{c|cccccc} & o_1 & o_2 & o_3 & o_4 & o_5 & o_6 \\ \hline u_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_2 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Задача № 8.3.

Пусть имеется иерархически организованная система объектов доступа, четыре пользователя u_1, u_2, u_3 и u_4 , объединенных в три рабочих группы g_1, g_2 и g_3 . Вхождение пользователей в рабочие группы, групповые и индивидуальные назначения доступа показаны на рис.



Задание. Определите общий коэффициент дублирования прав доступа в системе по чтению и коэффициент дублирования прав доступа по чтению для пользователя u_2 .

Решение.

$$\mathcal{K}_{\text{дубл}} = (\mathcal{K}_{\text{k}}^{\text{и}} + \mathcal{K}_{\text{k}}^{\text{гр}}) / N * M$$

$$k_{ijk}^{\text{и}} = (R_{\text{k|гр}}^{\text{и}} \cdot (H^S + I))_{ij}$$

$$\mathcal{K}_{\text{k}}^{\text{гр}} = W \cdot (((H^{\text{гS}} + I)^T \cdot R_{\text{k}}^{\text{г}}) \cdot (H^S + I)) = W \cdot (R_{\text{k}}^{\text{г}} \cdot (H^S + I))$$

$$(H^{\text{гS}} + I) = I$$

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	1	1	1	1	1	1
o_2	0	1	0	0	0	0
o_3	0	0	1	1	1	1
o_4	0	0	0	1	0	0
o_5	0	0	0	0	1	1
o_6	0	0	0	0	0	1

$H^S + I$

	o_1	o_2	o_3	o_4	o_5	o_6
u_1	0	0	0	0	0	0
u_2	0	1	0	0	0	0
u_3	0	0	0	0	0	0
u_4	0	0	0	0	0	0

$R_{\text{г}}^{\text{и}}|_{\text{пр}}$
чтение

	o_1	o_2	o_3	o_4	o_5	o_6
u_1	0	0	0	0	0	0
u_2	0	1	0	0	0	0
u_3	0	0	0	0	0	0
u_4	0	0	0	0	0	0

$R_{\text{г}}^{\text{и}}|_{\text{итог}}$
чтение

	o_1	o_2	o_3	o_4	o_5	o_6
g_1	1	0	0	0	0	0
g_2	1	-1	0	-1	0	-1
g_3	1	-1	0	0	0	0

$R_{\text{г}}^{\text{г}}|_{\text{пр}}$
чтение

	o_1	o_2	o_3	o_4	o_5	o_6
g_1	1	1	1	1	1	1
g_2	1	0	1	0	1	0
g_3	1	0	1	1	1	1

$R_{\text{г}}^{\text{г}}|_{\text{итог}}$
чтение

	o_1	o_2	o_3	o_4	o_5	o_6
u_1	1	1	1	1	1	1
u_2	2	0	2	1	2	1
u_3	2	0	2	1	2	1
u_4	2	1	2	2	2	2

$R_{\text{г}}^{\text{и}}|_{\text{итог}}$
чтение

$\mathcal{K}_{\text{k}}^{\text{и}} = 1,375$

	o_1	o_2	o_3	o_4	o_5	o_6
u_1	1	1	1	1	1	1
u_2	2	1	2	1	2	1
u_3	2	0	2	1	2	1
u_4	2	1	2	2	2	2

$R_{\text{г}}^{\text{и}}|_{\text{итог}} + R_{\text{г}}^{\text{г}}|_{\text{итог}}$
чтение

$\mathcal{K}_{\text{дубл}} = 1,416666667$

$\mathcal{K}_{\text{дубл}}^{\text{и}} = 1,5$

ЛИТЕРАТУРА

1. **Хоффман Л.** Современные методы защиты информации. М.:Сов.радио, 1980. – 264с.
2. **Грушо А.А.,Тимонина Е.Е.** Теоретические основы защиты информации. М.:Яхтсмен, 1996. - 192с.
3. **Теория и практика обеспечения информационной безопасности /** Под ред. П.Д. Зегжды. М.:Яхтсмен, 1996. - 302с
4. **Прокопьев И.В., Шрамков И.Г., Щербаков А.Ю.** Введение в теоретические основы компьютерной безопасности : Уч. пособие. М., 1998.- 184с.
5. **Зегжда Д.П.,Ивашко А.М.** Основы безопасности информационных систем. - М.:Горячая линия - Телеком, 2000. - 452с.
6. Теоретические основы компьютерной безопасности: Учеб. пособие для вузов / **П.Н. Девянин, О.О.Михальский, Д.И.Правиков** и др.- М.: Радио и Связь, 2000. - 192с.
8. **Щербаков А.Ю.** Введение в теорию и практику компьютерной безопасности. М.: издатель Молгачев С.В.- 2001- 352 с.
9. **Гайдамакин Н.А.** Разграничение доступа к информации в компьютерных системах. - Екатеринбург: изд-во Урал. Ун-та, 2003. – 328 с.
10. **Корт С.С.** Теоретические основы защиты информации: Учебное пособие. – М.: Гелиос АРВ, 2004. – 240 с.
- 11.**Девянин П.Н.** Модели безопасности компьютерных систем: Учеб. пособие. – М.: Изд.центр «Академия», 2005. – 144 с.