МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.94

ОБЗОРНЫЕ ЛЕКЦИИ ПО МОДЕЛЯМ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

П. Н. Девянин

Институт криптографии, связи и информатики, г. Москва, Россия

E-mail: peter_devyanin@hotmail.com

Приводится содержание трех обзорных лекций по дисциплине «Модели безопасности компьютерных систем». Рассматриваются основные свойства классических моделей дискреционного, мандатного, ролевого управления доступом, безопасности информационных потоков и изолированной программной среды, а также моделей безопасности логического управления доступом и информационными потоками (ДП-моделей). Формулируются наиболее интересные задачи для практических занятий по дисциплине.

Ключевые слова: *компьютерная безопасность, модели безопасности, ДП-модели.*

Введение

В соответствии с проектом ГОС ВПО третьего поколения модели безопасности компьютерных систем (КС) преподаются в рамках следующих дисциплин:

— «Теоретические основы компьютерной безопасности» для магистров;
— «Модели безопасности компьютерных систем» специальности «Компьютерная безопасность» квалификации специалист;
— «Безопасность информационных и аналитических систем» специальности «Информационно-аналитические системы безопасности» квалификации специалист;
— «Основы информационной безопасности» для бакалавров и других специальностей направления 090000 — «Информационная безопасность» квалификации специалист.

Исследование формальных моделей, особенно моделей безопасности логического управления доступом и информационными потоками в КС, создает предпосылки для развития теории компьютерной безопасности. С применением формальных моделей возможна разработка научно-обоснованных подходов, использование которых обеспечит гарантии выполнения требований безопасности в существующих или перспективных защищенных КС.

Основными классическими моделями, ориентированными на защиту от угроз конфиденциальности и целостности информации, а также от угрозы раскрытия параметров КС, являются модели дискреционного управления доступом, изолированной программной среды (ИПС), мандатного управления доступом, безопасности информационных потоков, ролевого управления доступом [1 – 3]. Кроме того, в настоящее

время активно развивается семейство моделей безопасности логического управления доступом и информационными потоками (сокращенно ДП-моделей).

Классические модели и ДП-модели построены с применением следующих основных понятий: субъект (*subject*), объект (*object*), контейнер (*container*), сущность (*entity*), доступ (*access*), право доступа (*access right*), информационный поток по памяти (*information flow by memory*) или по времени (*information flow by time*). При этом в современной теории компьютерной безопасности наибольшее развитие в области формального моделирования безопасности КС получил подход, заключающийся в представлении исследуемой КС в виде абстрактной системы (автомата), каждое состояние которой описывается доступами, реализуемыми субъектами к сущностям, а переходы КС из состояния в состояние описываются командами или правилами преобразования состояний, выполнение которых, как правило, инициируется субъектами. В основе данного подхода используется аксиома 1, позволяющая выделить элементы КС, необходимые для анализа ее безопасности.

Аксиома 1 (основная аксиома компьютерной безопасности). Все вопросы безопасности информации в КС описываются доступами субъектов к сущностям.

Классификация и взаимосвязь рассматриваемых моделей безопасности КС приведена на рис. 1. Все они войдут в разрабатываемое автором учебное пособие «Модели безопасности компьютерных систем», второе издание которого планируется в 2010 г.

1. Модели дискреционного управления доступом и изолированной программной среды

Модель Харрисона — Руззо — Ульмана (ХРУ) является одной из первых формальных моделей. В модели произвольная КС с дискреционным управлением доступом описывается множеством матриц доступов, каждая из которых соответствует состоянию КС, и командами преобразования матриц доступов. Каждая из команд задается множеством параметров, условием выполнения и конечной последовательностью примитивных операторов, преобразующих матрицу доступов. Применение команды переводит КС из состояния в последующее состояние.

В модели ХРУ анализируются условия, при выполнении которых возможна проверка безопасности КС. При этом используется следующее определение.

Определение 1. Начальное состояние системы ХРУ называется безопасным относительно некоторого права доступа r, когда невозможен переход системы в состояние, в котором право доступа r появилось в ячейке матрицы доступов, до этого r не содержавшей.

Основным фактом, который доказывается в модели ХРУ, является следующая теорема.

Теорема 1. Задача проверки безопасности произвольных систем ХРУ алгоритмически неразрешима.

При доказательстве теоремы 1 используется факт, обоснованный в рамках теории машин Тьюринга: не существует алгоритма проверки для произвольной машины Тьюринга и произвольного начального слова, остановится ли машина Тьюринга в конечном состоянии или нет. При этом строится гомоморфизм произвольной машины Тьюринга в соответствующую ей систему ХРУ, для чего все элементы и команды машины Тьюринга задаются в виде элементов и команд системы ХРУ. Например, внешний и внутренний алфавиты машины Тьюринга описываются множеством прав доступа системы ХРУ, пройденный участок ленты машины Тьюринга представляется

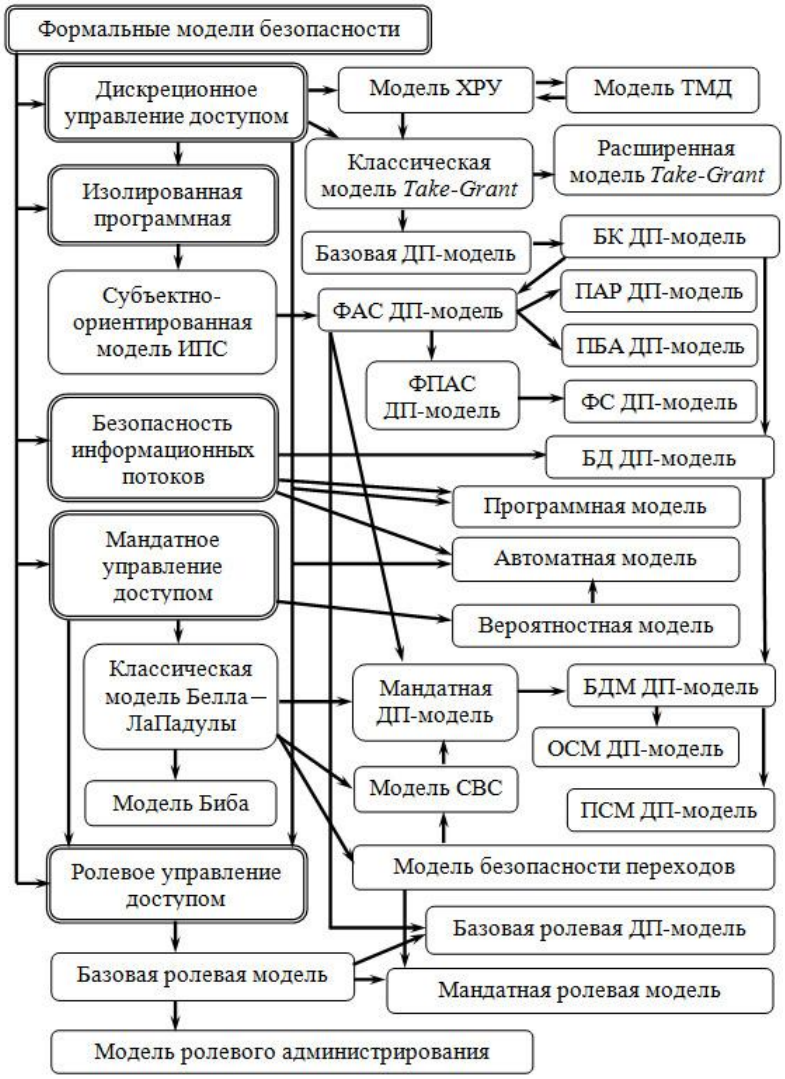

Рис. 1. Классификация и взаимосвязь положений моделей безопасности КС

матрицей доступов (рис. 2), команды машины Тьюринга задаются командами системы ХРУ. Таким образом, остановке машины Тьюринга (переходу в конечное состояние q_0) соответствует утечка права доступа q_0 в системе ХРУ. Значит, задача проверки безопасности произвольных систем ХРУ алгоритмически неразрешима.

Следует отметить, что в современных КС в большинстве случаев при определении безопасного состояния рассматривается утечка права доступа r не в произвольную, а в заданную ячейку матрицы доступов. Например, появление у администратора безопасности прав доступа на чтение или модификацию конфигурационных файлов ОС, как правило, не будет являться нарушением безопасности, однако получение данного права доступа нарушителем является нарушением безопасности. Таким образом, при доказательстве теоремы 1 следует рассмотреть вопрос о существовании для произвольной системы ХРУ алгоритма проверки возможности утечки права доступа в заданную ячейку матрицы доступов. При этом целесообразно использовать следующее рассуждение.

Каждой системе ХРУ поставим в соответствие другую систему ХРУ, в которой в каждую матрицу доступов M первой системы добавлен специальный субъект s_0, обладающий сам к себе специальным правом доступа $active \in M[s_0, s_0]$; в каждую ко-

	s_1	s_2	s_3	…	s_i	…	s_{n-1}	s_n
s_1	a_{s1}, *left*	*own*						
s_2		a_{s2}	*own*					
s_3			a_{s3}					
…								
s_i					a_{si}, q_{ij}			
…								
s_{n-1}							$a_{s\,n-1}$	*own*
s_n								a_{sn}, *right*

Рис. 2. Заполнение матрицы доступов системы ХРУ

манду первой системы, содержащую примитивный оператор внесения права доступа r, во второй системе добавлен параметр для обозначения субъекта, который в условии команды проверяется на наличие у него самого к себе права доступа *active*, также в команду добавлен примитивный оператор внесения права доступа r в ячейку $M[s_0, s_0]$. Таким образом, утечка права доступа r в первой системе происходит тогда и только тогда, когда во второй системе происходит утечка права доступа r в ячейку $M[s_0, s_0]$. Следовательно, если бы для произвольной системы ХРУ существовал алгоритм проверки возможности утечки права доступа в заданную ячейку матрицы доступов, то существовал бы алгоритм проверки возможности утечки права доступа в произвольную ячейку.

Развитием модели ХРУ является модель ТМД, позволяющая за счет введения типов субъектов и объектов расширить класс КС, для которых существует алгоритм проверки безопасности. Очевидно, что система модели ТМД является обобщением системы модели ХРУ, которую можно рассматривать как частный случай системы модели ТМД с одним единственным типом для всех объектов и субъектов. С другой стороны, следует обратить внимание на то, что любую систему модели ТМД можно выразить через систему модели ХРУ, введя для обозначения типов специальные права доступа и заменив проверку типов в командах проверкой наличия соответствующих прав доступа. Описание алгоритма построения для произвольной системы модели ТМД эквивалентной ей системы модели ХРУ является интересной практической задачей.

В модели ТМД обосновывается, что алгоритм проверки безопасности существует для ациклических монотонных систем модели ТМД, при этом используются следующие определения и доказывается теорема.

Определение 2. Система монотонной ТМД (МТМД) — система ТМД, в командах которой отсутствуют немонотонные примитивные операторы вида «удалить»… и «уничтожить»…

Определение 3. Каноническая форма системы МТМД (КФМТМД) — система МТМД, в которой команды, содержащие примитивные операторы вида «создать»..., не содержат условий и примитивных операторов вида «внести»...

Определение 4. Система МТМД (КФМТМД) называется ациклической (АМТМД или соответственно АКФМТМД) тогда и только тогда, когда ее граф создания (определение графа создания приводится в [2]) не содержит циклов.

Теорема 2. Для каждой системы МТМД существует эквивалентная ей система КФМТМД.

В доказательстве теоремы 2 приводится алгоритм построения для произвольной системы МТМД эквивалентной ей системы КФМТМД. В алгоритме особо рассматриваются специальным образом активизированные объекты и субъекты системы. При этом при построении из системы МТМД ее канонической формы в нее добавляется специальный субъект s_{active}, имеющий специальный тип t_{active}, который будет обладать специальным правом доступа *active* только к активизированным объектам. Кроме того, в параметры каждой команды системы КФМТМД, не содержащей примитивные операторы вида «создать»..., добавляется параметр с типом t_{active}. Таким образом, наличие специального права доступа *active* у субъекта s_{active} к объекту может быть использовано для обозначения объекта как активизированного.

В результате обосновывается теорема.

Теорема 3. Существует алгоритм проверки безопасности систем АМТМД.

Алгоритм проверки безопасности систем АМТМД состоит из следующих трех шагов:

1. Для системы АМТМД построить эквивалентную ей систему АКФМТМД.

2. Используя команды, содержащие только примитивные операторы «создать»... и не содержащие условий, перейти из начального состояния системы в некоторое «развернутое» состояние, обеспечивающее минимально необходимый и достаточный для распространения прав доступа состав объектов.

3. Используя команды, не содержащие примитивные операторы «создать»..., перейти из «развернутого» состояния в «замкнутое» состояние, в котором дальнейшее применение таких команд не приводит к изменениям в матрице доступов.

В модели *Take-Grant* условия передачи прав доступа и реализации информационных потоков рассматриваются с использованием графов доступов, что позволяет добиться большей наглядности исследуемых положений модели. Модель *Take-Grant* изучается в два этапа. На первом этапе рассматривается классическая модель *Take-Grant*, в которой анализируются алгоритмически проверяемые условия передачи прав доступа. На втором этапе рассматривается расширенная модель *Take-Grant*, в которой анализируются условия реализации в КС информационных потоков.

В классической модели *Take-Grant* описываются четыре де-юре правила преобразования графов доступов $take(\alpha, x, y, z)$, $grant(\alpha, x, y, z)$, $create(\alpha, x, y)$, $remove(\alpha, x, y)$ и используются следующие основные определения.

Определение 5. Пусть $x, y \in O_0$, $x \neq y$ — различные объекты графа доступов $G_0 = (S_0, O_0, E_0)$ и α — множество прав доступа. Определим предикат $can\_share(\alpha, x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют графы $G_1 = (S_1, O_1, E_1), \ldots, G_N = (S_N, O_N, E_N)$ и правила $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, \alpha) \subset E_N$.

Определение 6. Островом в произвольном графе доступов G_0 называется его максимальный tg-связный подграф, состоящий только из вершин субъектов.

Определение 7. Мостом в графе доступов G_0 называется проходящий через вершины-объекты tg-путь, концами которого являются вершины-субъекты и словарная запись имеет вид $\overrightarrow{t^*}$, $\overleftarrow{t^*}$, $\overrightarrow{t^*}\,\overrightarrow{g}\,\overleftarrow{t^*}$, $\overrightarrow{t^*}\,\overleftarrow{g}\,\overleftarrow{t^*}$, где символ «$*$» означает многократное (в том числе нулевое) повторение.

Определение 8. Начальным пролетом моста в графе доступов G_0 называется tg-путь, началом которого является вершина-субъект, концом — объект, проходящий через вершины-объекты, словарная запись которого имеет вид $\overrightarrow{t^*}\,\overrightarrow{g}$.

Определение 9. Конечным пролетом моста в графе доступов G_0 называется tg-путь, началом которого является вершина-субъект, концом — объект, проходящий через вершины-объекты, словарная запись которого имеет вид $\overrightarrow{t^*}$.

Алгоритмически проверяемые необходимые и достаточные условия истинности предиката $can\_share(\alpha, x, y, G_0)$ обосновываются в следующей теореме.

Теорема 4. Пусть $G_0 = (S_0, O_0, E_0)$ — произвольный граф доступов, $x, y \in O_0$, $x \neq y$. Предикат $can\_share(\alpha, x, y, G_0)$ истинен тогда и только тогда, когда или $(x, y, \alpha) \subset E_0$, или выполняются условия 1–3:

1. Существуют объекты $s_1, \ldots, s_m \in O_0$: $(s_i, y, \gamma_i) \subset E_0$ для $i = 1, \ldots, m$ и $\alpha = \gamma_1 \cup \ldots \cup \gamma_m$.

2. Существуют субъекты $x'_1, \ldots, x'_m, s'_1, \ldots, s'_m \in S_0$:

а) $x = x'_i$ или x'_i соединен с x начальным пролетом моста в графе G_0, где $i = 1, \ldots, m$;

б) $s_i = s'_i$ или s'_i соединен с s_i конечным пролетом моста в графе G_0, где $i = 1, \ldots, m$.

3. В графе G_0 для каждой пары $(x'_i,\ s'_i)$, где $i = 1, \ldots, m$, существуют острова $I_{i,1}, \ldots, I_{i,u_i}$, где $u_i \geqslant 1$, такие, что $x'_i \in I_{i,1}$, $s'_i \in I_{i,u_i}$, и существуют мосты между островами $I_{i,j}$ и $I_{i,j+1}$, где $j = 1, \ldots, u_i - 1$.

Доказательство достаточности выполнения условий теоремы для истинности предиката $can\_share(\alpha, x, y, G_0)$ является конструктивным и состоит в обосновании возможности передачи прав доступа по пролетам мостов, мостам и островам (фрагмент доказательства приведен на рис. 3). Формальное доказательство необходимости выполнения условий теоремы для истинности предиката $can\_share(\alpha, x, y, G_0)$ является достаточно трудоемким.

Однако интересной задачей, реализующей наглядный подход к доказательству необходимости выполнения условий теоремы для истинности предиката $can\_share(\alpha, x, y, G_0)$, является обоснование того, что не существует путей между двумя субъектами, отличных от мостов и проходящих через вершины-объекты, по которым возможна передача прав доступа. При ее решении следует рассмотреть все пути (с учетом симметрии) длины 2 между двумя субъектами, проходящие через вершины-объекты, по которым возможна передача прав доступа (рис. 4,*а*) и невозможна передача прав доступа (рис. 4,*б*). Любой путь, соединяющий двух субъектов, проходящий через объекты, по которому возможна передача прав доступа, не должен содержать фрагменты, приведенные на рис. 4,*б*. В то же время очевидно, что любой такой путь, состоящий только из фрагментов, приведенных на рис. 4,*а*, является мостом.

Для того чтобы обеспечить лучшее усвоение способов применения определения моста при анализе путей передачи прав доступа, целесообразно рассмотреть графы

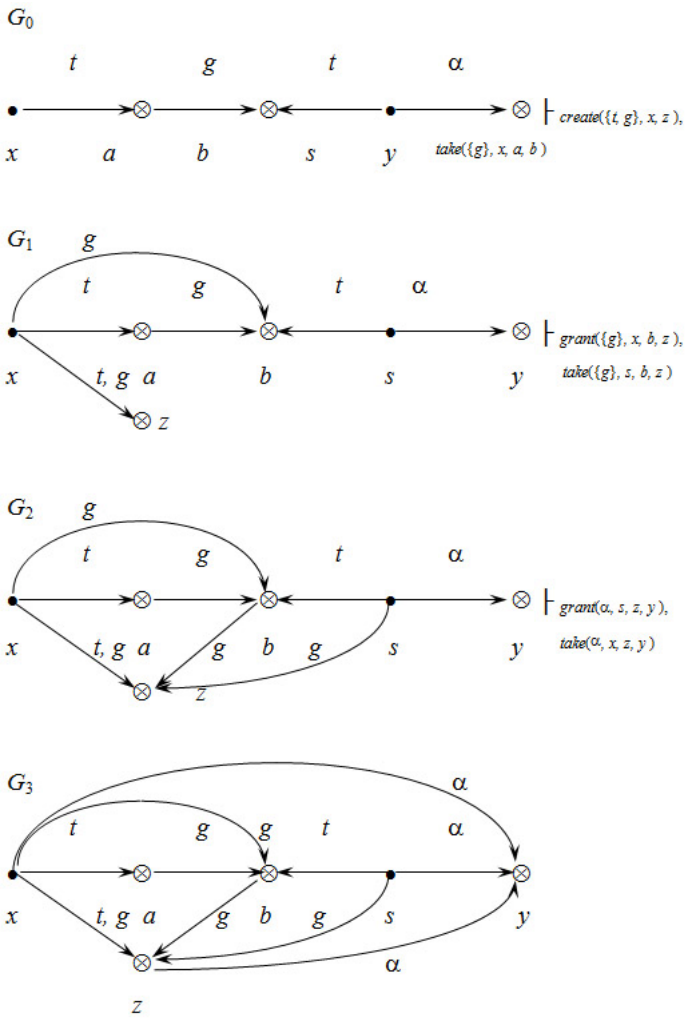

Рис. 3. Пример передачи прав доступа по мосту

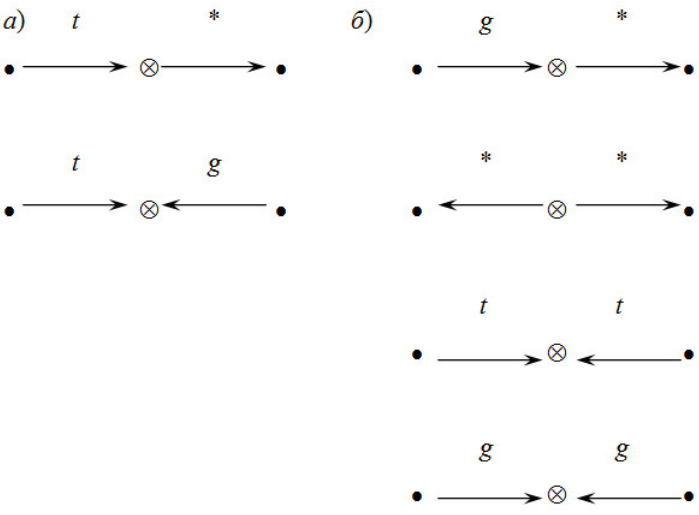

Рис. 4. Виды путей длины 2 ($*$ — или t, или g)

доступов, представленные на рис. 5, с использованием которых следует показать, что мостами могут являться tg-пути, неоднократно проходящие через один и тот же объект.

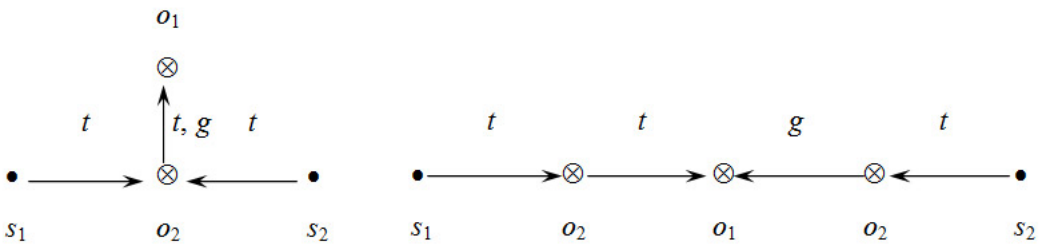

Рис. 5. «Т-образный» мост

Также представляет интерес решение задачи на непосредственную проверку условий теоремы 4 (рис. 6).

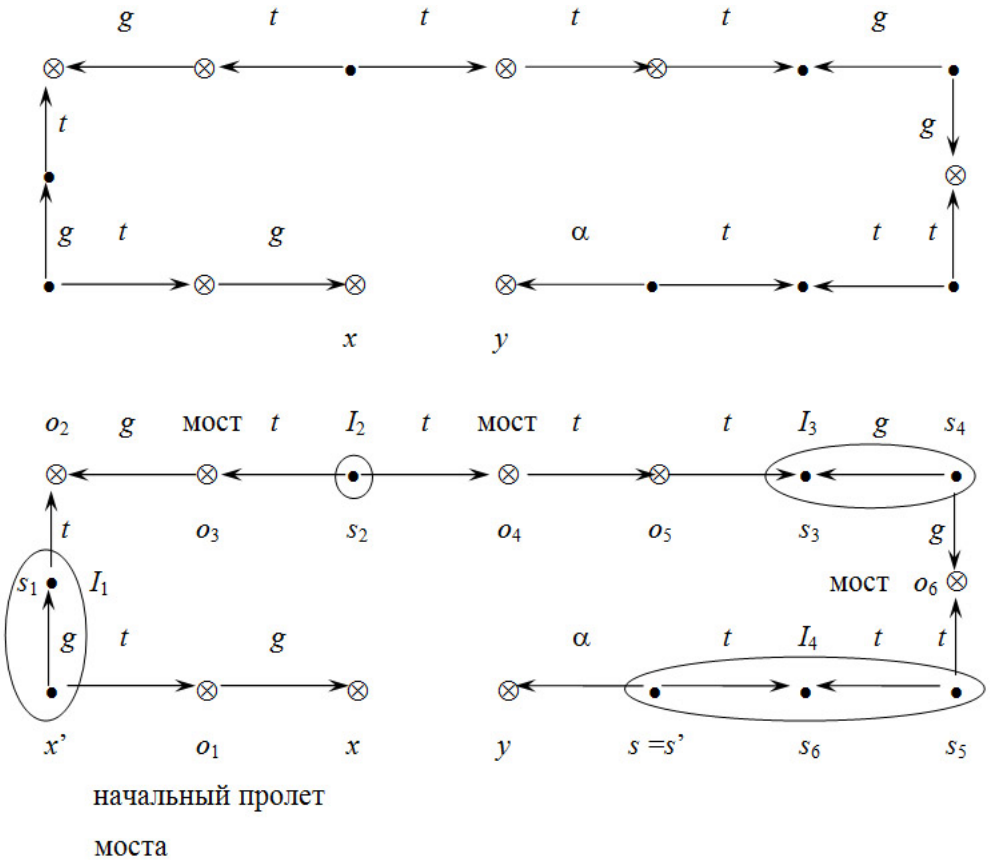

Рис. 6. Проверка условий теоремы 4

В расширенной модели *Take-Grant* кроме де-юре правил рассматриваются четыре де-факто правила преобразования графов доступов $spy(x, y, z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$ и используется следующее определение.

Определение 10. Пусть $x, y \in O_0$, $x \neq y$ — различные объекты графа доступов и информационных потоков $G_0 = (S_0, O_0, E_0 \cup F_0)$. Определим предикат $can\_write(x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют графы $G_1 = (S_1, O_1, E_1 \cup F_1)$, ..., $G_N = (S_N, O_N, E_N \cup F_N)$ и де-юре или де-факто правила $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, w) \in F_N$.

Алгоритмически проверяемые необходимые и достаточные условия истинности предиката $can\_write(x, y, G_0)$ обосновываются в следующей теореме.

Теорема 5. Пусть $G_0 = (S_0,\ O_0,\ E_0 \cup F_0)$ — граф доступов и информационных потоков, $x, y \in O_0$, $x \neq y$. Тогда предикат $can\_write(x, y, G_0)$ истинен тогда и только тогда, когда существуют объекты $o_1, \ldots, o_m \in O_0$, где $o_1 = x$, $o_m = y$, такие, что или $m = 2$ и $(x, y, w) \in F_0$, или для $i = 1, \ldots, m-1$ выполняется одно из условий:

— $o_i \in S_0$ и или истинен предикат $can\_share(\{w\}, o_i, o_{i+1}, G_0)$, или $(o_i, o_{i+1}, w) \in \in E_0 \cup F_0$;
— $o_{i+1} \in S_0$ и или истинен предикат $can\_share(\{r\}, o_{i+1}, o_i, G_0)$, или $(o_{i+1}, o_i, r) \in \in E_0 \cup F_0$;
— $o_i, o_{i+1} \in S_0$ и или истинен предикат $can\_share(\alpha, o_i, o_{i+1}, G_0)$, или истинен предикат $can\_share(\alpha, o_{i+1}, o_i, G_0)$, где $\alpha \in \{t, g\}$, или существует объект $o'_i \in O_0$, такой, что либо истинны предикаты $can\_share(\{t\}, o_i, o'_i, G_0)$, $can\_share(\{g\}, o_{i+1}, o'_i, G_0)$, либо истинны предикаты $can\_share(\{g\}, o_i, o'_i, G_0)$, $can\_share(\{t\}, o_{i+1}, o'_i, G_0)$.

Наиболее интересным при доказательстве теоремы 5 является обоснование возможности реализации информационного потока между двумя субъектами в случае, когда выполнено третье условие теоремы. На рис. 7 приведен пример обоснования возможности реализации информационного потока между двумя субъектами для одного из случаев.

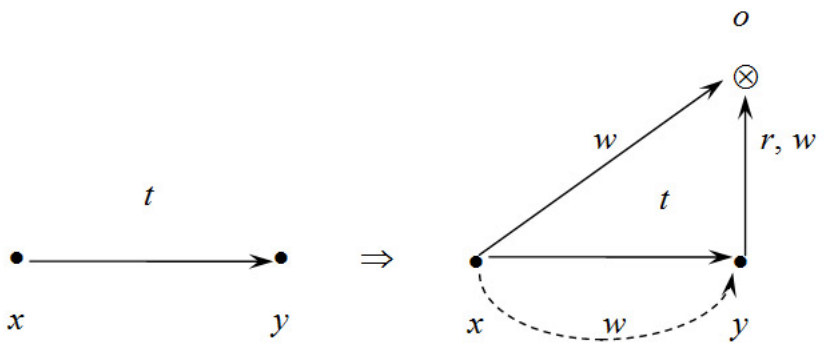

Рис. 7. Реализация информационного потока на запись

Для лучшего усвоения положений расширенной модели *Take-Grant* целесообразно решить задачи на применение теоремы 5 с целью проверки условий истинности предиката $can\_write(x, y, G_0)$ (рис. 8).

Кроме анализа условий реализации информационных потоков, в рамках расширенной модели *Take-Grant* рассматривается алгоритм построения замыкания графа доступов и информационных потоков. При этом используются следующие определения.

Определение 11. Пусть $G = (S, O, E \cup F)$ — граф доступов и информационных потоков, такой, что для каждого субъекта $s \in S$ существует объект $o \in O$, такой, что выполняется условие $(s, o, \{t, g, r, w\}) \subset E$. Тогда замыканием (или де-факто-замыканием) графа G называется граф доступов и информационных потоков $G^* = (S, O, E^* \cup F^*)$, полученный из G применением последовательности правил *take*, *grant* и де-факто правил. При этом применение к графу G^* указанных правил не приводит к появлению в нем новых ребер.

Определение 12. Пусть $G = (S, O, E \cup F)$ — граф доступов и информационных потоков, такой, что для каждого субъекта $s \in S$ существует объект $o \in O$, такой,

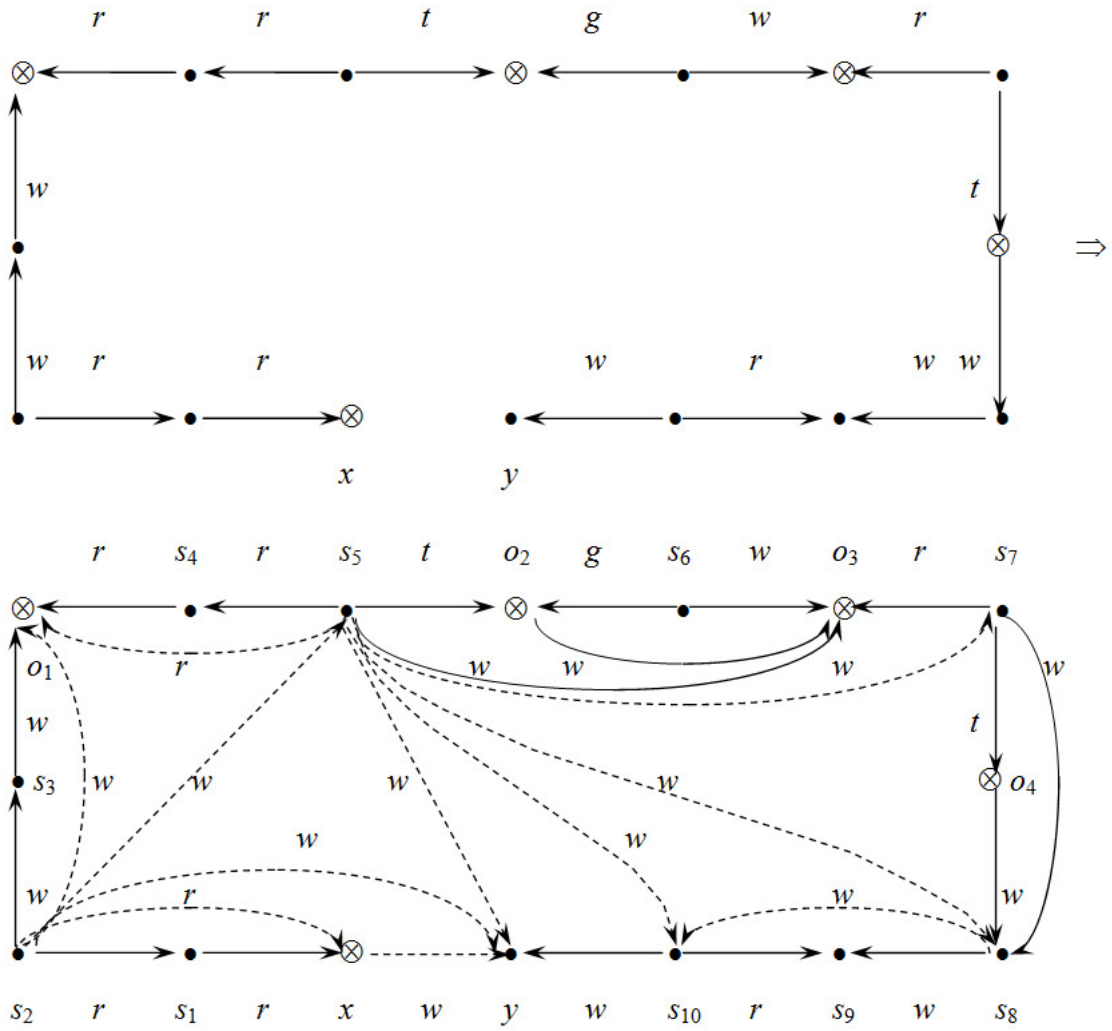

Рис. 8. Проверка выполнения условий теоремы 5

что выполняется условие $(s, o, \{t, g, r, w\}) \subset E$. Тогда tg-замыканием графа G называется граф доступов и информационных потоков $G^{tg} = (S, O, E^{tg} \cup F)$, полученный из G применением последовательности правил *take* или *grant*. При этом каждое ребро $(o_1, o_2, \alpha) \in E^{tg} \setminus E$ имеет вид (o_1, o_2, t) или (o_1, o_2, g), и применение к графу G^{tg} правил *take* или *grant* не приводит к появлению в нем новых ребер указанного вида.

Определение 13. Пусть $G = (S, O, E \cup F)$ — граф доступов и информационных потоков, такой, что для каждого субъекта $s \in S$ существует объект $o \in O$, для которого выполняется условие $(s, o, \{t, g, r, w\}) \subset E$. Тогда де-юре-замыканием графа G называется граф доступов и информационных потоков $G^{\text{de jure}} = (S, O, E^{\text{de jure}} \cup F)$, полученный из G применением последовательности правил *take* или *grant*. При этом применение к графу $G^{\text{de jure}}$ правил *take* или *grant* не приводит к появлению в нем новых ребер.

Таким образом, схематично алгоритм построения замыкания графа доступов и информационных потоков состоит из следующих трех шагов:

1. Построение tg-замыкания.
2. Построение де-юре-замыкания.
3. Построение замыкания (де-факто-замыкания).

При этом наиболее интересным является алгоритм построения tg-замыкания графа доступов и информационных потоков $G = (S, O, E \cup F)$, состоящий из следующих пяти шагов:

1. Для каждого $s \in S$ выполнить правило $create(\{t, g, r, w\}, s, o)$; при этом создаваемые объекты занести в множество O, создаваемые ребра занести в множество E.

2. Инициализировать: $L = \{(x, y, \alpha) \in E : \alpha \in \{t, g\}\}$ — список ребер графа доступов и информационных потоков и $N = \varnothing$ — множество вершин.

3. Выбрать из списка L первое ребро (x, y, α). Занести x и y в множество N. Удалить ребро (x, y, α) из списка L.

4. Для всех вершин $z \in N$ проверить возможность применения правил $take$ или $grant$ на тройке вершин x, y, z с использованием ребра (x, y, α), выбранного на шаге 3. Если в результате применения правил $take$ или $grant$ появляются новые ребра вида (a, b, β), где $\{a, b\} \subset \{x, y, z\}$ и $\alpha \in \{t, g\}$, занести их в конец списка L и множество E.

5. Если список L не пуст, перейти на шаг 3.

Пример применения алгоритма построения tg-замыкания приведен на рис. 9.

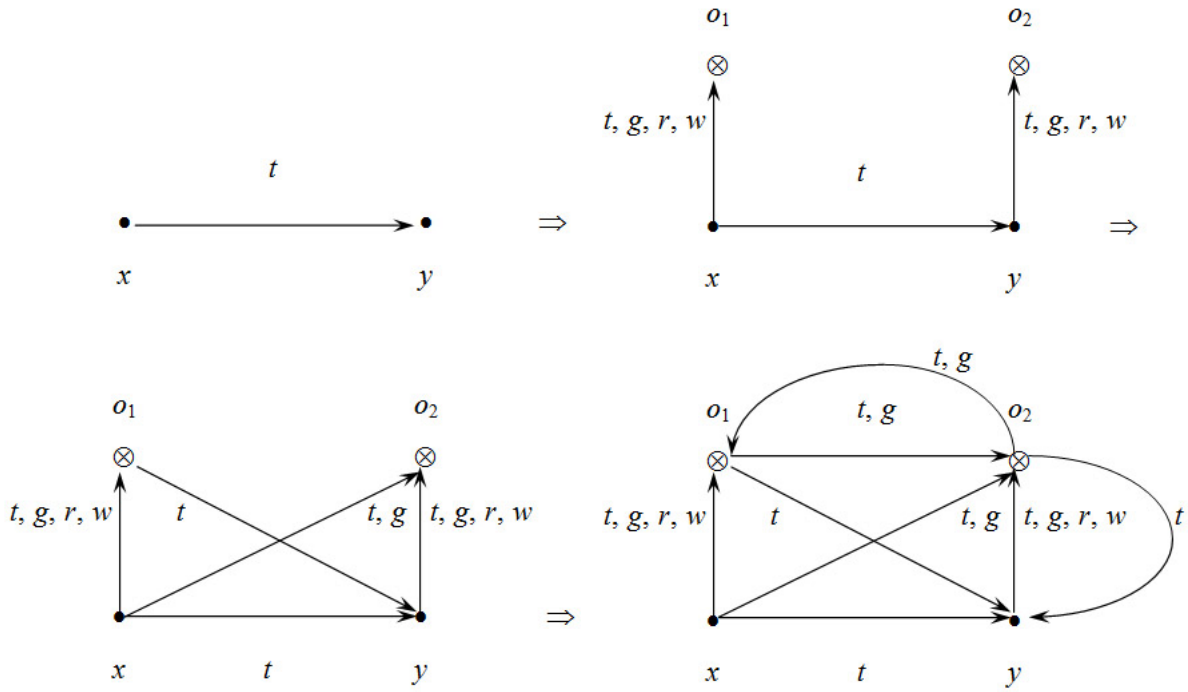

Рис. 9. Пример применения алгоритма построения tg-замыкания

При завершении изучения положений моделей КС с дискреционным управлением доступом целесообразно решить практическую задачу построения для системы *Take-Grant* двух эквивалентных ей систем ХРУ и ТМД. При этом следует особо рассмотреть отличия полученных систем ХРУ и ТМД. В том числе следует обратить внимание на то, что для обозначения субъектов модели *Take-Grant* в модели ХРУ потребуется использование специального права доступа, а в модели ТМД для этого достаточно использовать специальные типы. Также целесообразно дать ответ на вопрос, каким ограничениям должна соответствовать система, построенная в соответствии с положениями модели *Take-Grant*, чтобы эквивалентная ей система ТМД была монотонной и ациклической.

В основанной на дискреционном управлении доступом субъектно-ориентированной модели ИПС рассматриваются вопросы определения порядка безопасного взаимодействия субъектов системы, описания и обоснования необходимых условий реализации ИПС, которая обеспечивает выполнение в КС требований априорно заданной политики безопасности. При этом используются следующие определения и обозначения.

Определение 14. Объект o в момент времени t функционально ассоциирован с субъектом s, когда состояние объекта o повлияло на вид преобразования данных, реализуемого субъектом s в следующий момент времени $t+1$.

Определение 15. Монитор обращений (МО) — субъект, активизирующийся при возникновении любого информационного потока между объектами КС. Монитор безопасности объектов (МБО) — МО, который разрешает только потоки, не принадлежащие множеству запрещенных информационных потоков.

Введём следующие обозначения:
$[s]_t$ — множество объектов, ассоциированных с субъектом s в момент времени t;
$o[t]$ — состояние объекта o в момент времени t;
$Stream(s,o) \to o'$ — поток информации от объекта o к объекту o';
$Create(s,o) \to s'$ — операция порождения субъектов (из объекта o порожден субъект s' при активизирующем воздействии субъекта s).

МБО фактически является механизмом реализации политики безопасности в КС. При изменении функционально ассоциированных с субъектами или с МБО объектов могут измениться и свойства субъектов или самого МБО, и, как следствие, могут возникнуть потоки, принадлежащие множеству запрещенных информационных потоков (рис. 10).

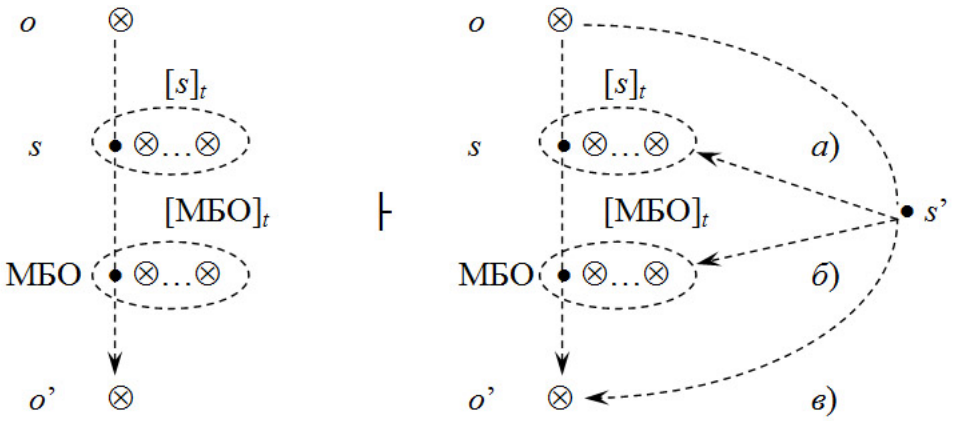

Рис. 10. Возможные пути обхода политики безопасности нарушителем s': *а* — изменение функциональности субъекта s; *б* — изменение функциональности МБО; *в* — реализация информационного потока в обход МБО

Для противодействия возможности реализации рассмотренных запрещенных информационных потоков в рамках модели предлагается обеспечить корректность субъектов системы друг относительно друга и замкнутость программной среды. При этом используются следующие определения.

Определение 16. Субъекты s и s' называются корректными относительно друг друга, когда в любой момент времени отсутствует поток (изменяющий состояние объ-

екта) между любыми объектами o и o', ассоциированными соответственно с субъектами s и s'. Таким образом, выполняется следующее условие:

$o \in [s]_t$, $o' \in [s']_t$, где $t \geqslant 0$, не существует субъекта s'', такого, что или $Stream(s'', o) \to o'$, или $Stream(s'', o') \to o$.

Определение 17. Корректные относительно друг друга субъекты s и s' называются абсолютно корректными, когда множества ассоциированных объектов указанных субъектов не имеют пересечения. Таким образом, выполняется условие

$$[s]_t \cap [s']_t = \varnothing, \text{ где } t \geqslant 0.$$

Определение 18. Монитор порождения субъектов (МПС) — субъект, активизирующийся при любом порождении субъектов. Монитор безопасности субъектов (МБС) — МПС, который разрешает порождение субъектов только для фиксированного множества пар активизирующих субъектов и объектов-источников.

Определение 19. КС называется замкнутой по порождению субъектов (обладает замкнутой программной средой), когда в ней действует МБС.

Определение 20. Программная среда называется изолированной (абсолютно изолированной), когда она является замкнутой по порождению субъектов (в ней действует МБС) и субъекты из порождаемого множества корректны (абсолютно корректны) относительно друг друга, МБС и МБО.

Определение 21. Операция порождения субъекта $Create(s, o) \to s'$ называется порождением с контролем неизменности объекта-источника, когда для любого момента времени $t > t_0$, в который активизирована операция порождения объекта $Create$, порождение субъекта s' возможно только при тождественности объектов $o[t_0]$ и $o[t]$.

В субъектно-ориентированной модели ИПС обосновывается следующая лемма.

Лемма 1 (базовая теорема ИПС). Если с момента времени t_0 в абсолютной ИПС действует только порождение субъектов с контролем неизменности объекта-источника и все порождаемые субъекты абсолютно корректны относительно друг друга и существующих субъектов (в том числе МБО и МБС), то в любой момент времени $t > t_0$ программная среда также остается абсолютной ИПС.

Для учета влияния субъектов в КС на систему защиты целесообразно рассматривать расширенную схему взаимодействия элементов системы. При этом должна быть особо подчеркнута роль МБС при порождении субъектов из объектов (рис. 11). Взаимодействие субъектов и объектов при порождении потоков уточняется введением ассоциированных с субъектами объектов. Объект управления МБО (ОУ_O) содержит информацию о разрешенных значениях отображения $Stream$ (о разрешенных и запрещенных информационных потоках) и функционально ассоциирован с МБО, и объект управления МБС (ОУ_S) содержит информацию о значениях отображения $Create$ (об элементах множества разрешенных объектов-источников) и функционально ассоциирован с МБС. Таким образом, рассмотренная концепция ИПС является расширением классического подхода к реализации ядра безопасности КС.

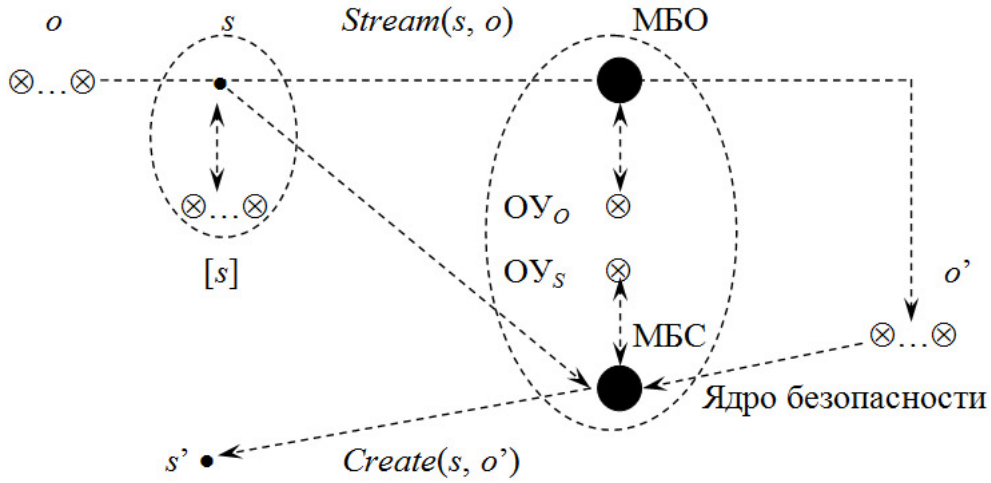

Рис. 11. Ядро безопасности с учетом контроля порождения субъектов

2. Модели мандатного или ролевого управления доступом и безопасности информационных потоков

Мандатное управление доступом чаще всего описывают в терминах, понятиях и определениях свойств классической модели Белла — ЛаПадулы и ее основных интерпретаций.

Классическая модель Белла — ЛаПадулы является автоматной моделью, в которой моделируемая КС представляется абстрактной системой, каждое состояние которой описывается с использованием:

— множества текущих доступов субъектов к объектам системы;
— функций, задающих для каждого субъекта уровень доступа и текущий уровень доступа, для каждого объекта его уровень конфиденциальности;
— матрицы доступов, позволяющей в дополнение к мандатному управлению доступом использовать дискреционное управление доступом.

Описывается множество действий системы, задающих правила перехода системы из состояния в состояние.

С использованием уровней доступа субъекта и уровня конфиденциальности объекта, видов доступа и матрицы доступов в модели Белла — ЛаПадулы задаются три базовых свойства безопасности: ss-свойство, $*$-свойство и ds-свойство, которыми должен обладать каждый безопасный доступ субъекта к объекту.

Определение 22. Доступ (s, o, r) обладает ss-свойством относительно $f = (f_s, f_o, f_c)$, где f_s — функция уровней доступа субъектов, f_o — функция уровней конфиденциальности объектов, f_c — функция текущих уровней доступа субъектов, когда выполняется одно из условий:

— $r \in \{execute, append\}$;
— $r \in \{read, write\}$ и $f_s(s) \geqslant f_o(o)$.

Состояние системы (b, m, f), где b — множество текущих доступов, m — матрица доступов, обладает ss-свойством, когда в нём все доступы обладают ss-свойством относительно f.

Определение 23. Доступ (s, o, r) обладает $*$-свойством относительно $f = (f_s, f_o, f_c)$, когда выполняется одно из условий:

— $r = execute$;
— $r = append$ и $f_o(o) \geqslant f_c(s)$;
— $r = read$ и $f_c(s) \geqslant f_o(o)$;
— $r = write$ и $f_c(s) = f_o(o)$.

Состояние системы $(b,\ m,\ f)$ обладает $*$-свойством, когда в нём все доступы обладают $*$-свойством относительно f.

Определение 24. Состояние системы $(b,\ m,\ f)$ обладает ds-свойством, когда в нём для каждого доступа $(s,\ o,\ r)$ выполняется условие $r \in m[s, o]$.

В безопасном состоянии все доступы должны быть безопасными. Система является безопасной, когда все состояния на всех возможных траекториях ее функционирования являются безопасными.

Таким образом, на безопасных траекториях функционирования система, построенная в соответствии со свойствами классической модели Белла — ЛаПадулы, должна разрешать получение субъектами только безопасных доступов к объектам.

С использованием требования соответствия ss-свойству всех доступов состояния системы уровнем доступа субъекта ограничивается максимальный уровень конфиденциальности объектов, к которым субъект потенциально может получить доступ на чтение. Основным свойством безопасности является $*$-свойство. С использованием требования соответствия $*$-свойству всех доступов состояния системы предотвращается возможность реализации субъектом информационного потока от объекта с высоким уровнем конфиденциальности к объекту с низким уровнем конфиденциальности, при этом используется проверка текущего уровня доступа субъекта. Свойство дискреционной безопасности (ds-свойство), в котором используется матрица доступов, как правило, подробно не рассматривается.

Алгоритмически проверяемые условия выполнения в системе свойств безопасности формулируются и обосновываются в имеющих схожие формулировки теоремах $A1$, $A2$ и $A3$. В качестве примера приведена формулировка теоремы $A1$.

Теорема 6 (теорема $A1$). Система обладает ss-свойством для любого начального состояния z_0, обладающего ss-свойством, тогда и только тогда, когда для каждого действия $(q,\ d,\ (b^*,\ m^*,\ f^*),\ (b,\ m,\ f))$, где q — запрос системе, d — ответ системы, $(b,\ m,\ f)$ — исходное состояние, $(b^*,\ m^*,\ f^*)$ — последующее состояние, выполняются следующие условия:

1. Каждый доступ $(s,\ o,\ r) \in b^* \setminus b$ обладает ss-свойством относительно f^*.

2. Если доступ $(s,\ o,\ r) \in b$ и не обладает ss-свойством относительно f^*, то $(s,\ o,\ r) \notin b^*$.

При изучении классической модели Белла — ЛаПадулы целесообразно учесть, что она разрабатывалась для обеспечения безопасности конкретной защищенной КС *Multics*. Поэтому некоторые элементы (например, права доступа *append, execute*, иерархия объектов, функция текущего уровня доступа субъектов) реализованы в модели только для обеспечения соответствия условиям функционирования КС *Multics* и не являются необходимыми для моделирования произвольной КС с мандатным управлением доступом. Поэтому рассмотрение интерпретаций модели Белла — ЛаПадулы (например, интерпретации «*read-write*» или интерпретации «безопасность переходов») и особенно описанных в них свойств безопасности позволит лучше изучить наиболее существенные и важные для теории элементы модели Белла — ЛаПадулы. Для того

чтобы лучше исследовать свойства классической модели Белла — ЛаПадулы и оценить сложность практической разработки формальной модели управления доступом для реальной КС, уместно решить задачу: построить пример системы Белла — ЛаПадулы с двумя субъектами и двумя объектами (большее число элементов системы не повлияет, по сути, на ход построения, но сделает его существенно более трудоемким).

Самым интересным при изучении модели Белла — ЛаПадулы является анализ ее недостатков, в том числе:

— отсутствие логической увязки условий выполнения системой свойств безопасности, данных в определениях, с заложенными в модель условиями их проверки, необходимость и достаточность которых доказывается в базовой теореме безопасности (теореме БТБ);
— модель «статична», то есть в ней отсутствует описание правил перехода системы из состояния в состояние, а также «статично» анализируются условия возникновения информационных потоков;
— реализация в КС только свойств безопасности, описанных в модели, не позволяет обеспечить защиту от возможности возникновения запрещенных информационных потоков (особенно информационных потоков по времени).

Первый недостаток легко иллюстрируется на примере некорректной интерпретации модели Белла — ЛаПадулы и имеет в основном теоретическое значение.

Второй недостаток можно рассмотреть на следующих примерах.

Пример 1. В классической модели Белла — ЛаПадулы для безопасности системы требуется безопасность каждого состояния на любой реализации (траектории) системы. Рассмотрим систему, в которой при переходе из любого состояния в последующее уровень доступа каждого субъекта устанавливается максимальным, а уровень конфиденциальности каждого объекта — минимальным. Такая система будет безопасна в смысле определений свойств безопасности классической модели Белла — ЛаПадулы (любой доступ любого субъекта к любому объекту будет, очевидно, удовлетворять *ss*-свойству и $*$-свойству). В то же время реализация такой системы вряд ли имеет какой-либо практический смысл.

Пример 2. Используем интерпретацию *low-watermark* классической модели Белла — ЛаПадулы, в которой строится пример системы с мандатным управлением доступом. В ней описываются три правила перехода системы из состояния в состояние: *Read*, *Write*, *Reset*, и доказывается безопасность системы в смысле определений свойств безопасности классической модели Белла — ЛаПадулы. Наиболее важным в примере (на что следует обратить внимание) является то, что наличие или отсутствие в определении результатов применения правила *Write* требования стирания информации в объекте в случае реального понижения его уровня конфиденциальности не влияет на логику доказательства безопасности системы. Таким образом, отсутствие в классической модели Белла — ЛаПадулы описаний правил перехода системы из состояния в состояние может привести к тому, что в формально безопасной КС будет возможна реализация запрещенных информационных потоков.

Пример 3. В интерпретации «безопасность переходов» делается попытка устранить второй недостаток модели Белла — ЛаПадулы путем введения в модель функции переходов, с использованием которой на моделируемую КС накладывается существенное ограничение: за один шаг работы системы (переход из состояния в состояние) вносится только одно изменение в один из параметров, используемый при определении

свойств безопасности. То есть либо изменяется на один элемент множество текущих доступов, либо одно из значений одной из функций, при этом остальные параметры остаются неизменными. С использованием интерпретации «безопасность переходов» можно разработать новые ограничения на функцию переходов, учитывающие особенности реализации механизмов защиты в современных КС (например, механизмов администрирования параметров безопасности).

Условия реализации информационных потоков анализируются в модели Белла — ЛаПадулы с использованием определения $*$-свойства. При этом предполагается, что информационный поток возникает в случае, когда в состоянии КС найдется субъект, который имеет доступ на чтение к одному объекту и доступ на запись к другому объекту. В то же время функциональность субъекта может быть реализована таким образом, что наличие описанных двух доступов не приводит к реализации информационного потока.

Третий недостаток модели Белла — ЛаПадулы имеет наибольшее практическое и теоретическое значение. Примеры запрещенных информационных потоков по времени, возникающих в результате реализации субъектами доступов к объектам, впервые были описаны самими авторами модели. В то же время в связи с постоянным усложнением современных КС, появлением в них новых функциональных возможностей, которые могут быть эффективно использованы нарушителем для создания запрещенных информационных потоков по памяти и по времени, условия реализации информационных потоков продолжают исследоваться в теории компьютерной безопасности.

При изучении модели Белла — ЛаПадулы условия реализации в КС запрещенных информационных потоков целесообразно рассмотреть на примерах.

Пример 4. Классический пример реализации запрещенного информационного потока по времени от объекта с высоким уровнем конфиденциальности к объекту с низким уровнем конфиденциальности.

Используем обозначения:

f_1 — объект-файл с уровнем конфиденциальности *High*, который может содержать запись «A» или запись «B»;

f_2 — объект-файл с уровнем конфиденциальности $Low < High$;

s_1 — субъект с высоким уровнем доступа *High*, работающий по программе:

Process $s_1(f_1 : file)$
  Open f_1 *for read*;
  While $f_1 =$ «A» *Do End*;
  Close f_1;
End.

s_2 — субъект-нарушитель с низким уровнем доступа *Low*, работающий по программе:

Process $s_2(f_1 : file, f_2 : file)$
  Open f_2 *for write*;
  Start $s_1(f_1)$;
  If (*Stop* s_1) *Then*
    Write «B» *to* f_2;
  Else
    Write «A» *to* f_2;
  End If

Close f_2;
End.

Схема реализации информационного потока по времени приведена на рис. 12.

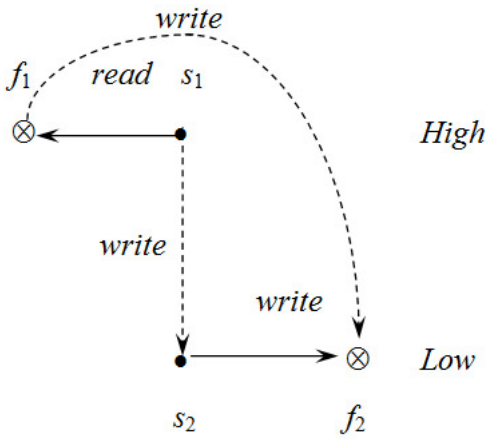

Рис. 12. Пример реализации информационного потока по времени с использованием «зависания» субъекта

Субъект-нарушитель s_2 действует одновременно с субъектом s_1, проверяя его состояние. При этом в зависимости от результата работы с конфиденциальным объектом-файлом f_1 субъекта s_1, который либо сразу завершит работу, либо «зависнет», субъект s_2 записывает данные в объект-файл f_2 с низким уровнем конфиденциальности.

Пример 5. Пример реализации информационного потока по времени с использованием совместного доступа кооперирующих субъектов к файловой директории. Используем следующие обозначения:

f_1 — объект-файл с уровнем конфиденциальности *High*, который может содержать запись «A» или запись «B»;

f_2 — объект-файл с уровнем конфиденциальности $Low < High$;

s_1 — субъект с высоким уровнем доступа *High*;

s_2 — субъект с низким уровнем доступа *Low*;

dir — объект-директория с уровнем конфиденциальности *Low*;

file — объект-файл в директории *dir* с уровнем конфиденциальности *Low*.

Субъекты s_1 и s_2 кооперируют друг с другом. В согласованный момент времени субъект s_1 считывает из f_1 его содержимое. Если оно «A», то он ничего не делает, если «B», то он открывает объект *file* на чтение (не нарушая свойств безопасности модели Белла — ЛаПадулы). В тот же момент времени субъект s_2 пытается удалить директорию *dir*. Если ему это удается, то он записывает в файл f_2 запись «A», если нет, то запись «B» (большинство современных КС не разрешат субъекту удалить директорию в случае, когда доступ к ее содержимому имеет другой субъект).

Таким образом, возможна реализация запрещенного информационного потока по времени, схема которого приведена на рис. 13.

Развитием модели Белла — ЛаПадулы является построенная на ее основе модель систем военных сообщений (СВС). Модель СВС ориентирована на анализ безопасности электронных почтовых систем. В то же время свойства модели СВС могут быть

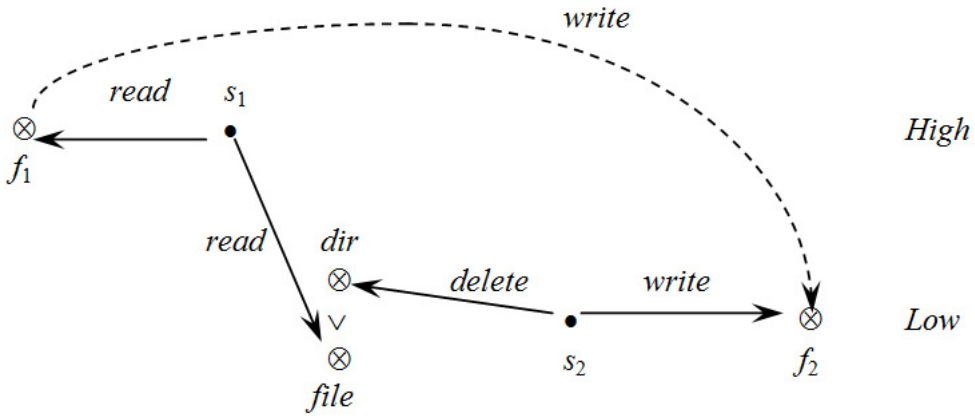

Рис. 13. Пример реализации информационного потока по времени с использованием совместного доступа кооперирующих субъектов к файловой директории

использованы как основа для построения произвольной КС с мандатным управлением доступом.

В отличие от модели Белла — ЛаПадулы в модели СВС рассматривается типичная для современных КС иерархическая структура сущностей (объектов и контейнеров), при этом определяются условия получения субъектом доступа к сущностям, находящимся внутри сущности-контейнера.

Модель СВС является более сложной для изучения, чем модель Белла — ЛаПадулы, ее описание включает значительный объем обозначений, определений, свойств. Целесообразно обратить внимание на структуру описания модели СВС — она состоит из двух частей. В первой части приводится неформальное описание модели, которое ориентировано на широкий круг пользователей КС. Во второй части дается предназначенное для специалистов по теории компьютерной безопасности формальное описание модели. Поэтому после изучения модели следует решить задачу: установить соответствие между неформальным и формальным описаниями модели.

Определение ss-свойства в модели СВС соответствует определению ss-свойства в модели Белла — ЛаПадулы. Для определения $*$-свойства безопасности используется понятие потенциальной модификации с источником, основанное на понятии информационного потока по памяти (возникающего при переходе системы из состояния в состояние) от сущности-источника к модифицируемой сущности. Дискреционное свойство безопасности (ds-свойство) реализовано в модели с использованием списков доступа. Таким образом, после изучения модели СВС следует дать ответ на вопрос: где в определениях свойств модели СВС реализованы свойства безопасности (ss-свойство, $*$-свойство и ds-свойство) классической модели Белла-ЛаПадулы.

Самым сложным в модели СВС является понятие потенциальной модификации с источником. Его целесообразно проиллюстрировать на примере.

Пример 6. Рассмотрим решение задачи с условием: описать потенциальную модификацию булевой сущности по ссылке r с источником булевой сущностью по ссылке y при запросе $i =< \text{and}, r, y >$ на вычисление функции $V(r_{s^*}) = V(r_s)$ and $V(y_s)$, при этом используя следующие обозначения:

V — функция значения сущности;

r_s — сущность по ссылке r в состоянии s;

$s \sim^{\rho} s_1$ — эквивалентность состояний s и s_1 относительно множества ссылок ρ;

$(u,\ i,\ s)$ — запрос i пользователя с идентификатором u в состоянии s;

s^* — состояние, последующее за состоянием s.

В состоянии s возможно, что значение сущности по ссылке r равно 0, и при потенциальной модификации при любом значении сущности по ссылке y изменение значения сущности по ссылке r не произойдет. Поэтому следует рассмотреть эквивалентное относительно множества ссылок $\{r\}$ состояние s_1 (рис. 14), отличающееся от состояния s только значением сущности по ссылке r.

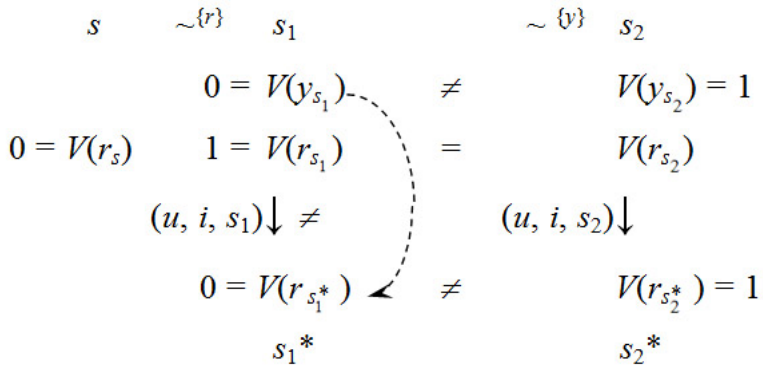

Рис. 14. Пример потенциальной модификации с источником

В случае, когда значение сущности по ссылке y в состоянии s_1 равно 0, при выполнении запроса $(u,\ i,\ s_1)$ произойдет изменение (модификация) значения сущности по ссылке r (значение сущности $r_{s_1^*}$ станет равным 0). Чтобы показать, что сущность по ссылке y является источником потенциальной модификации, следует перейти к эквивалентному относительно множества ссылок $\{y\}$ состоянию s_2, которое отличается от состояния s_1 только значением сущности по ссылке y. В результате применения запроса $(u,\ i,\ s_2)$ значение сущности по ссылке r в состоянии s_2^* будет отличаться от ее значения в состоянии s_1^*. Таким образом, при потенциальной модификации сущности по ссылке r с источником сущностью по ссылке y реализуется информационный поток по памяти от сущности по ссылке y к сущности по ссылке r.

Использование понятия потенциальной модификации с источником позволяет более точно установить факт реализации в КС информационного потока. Однако для этого требуется осуществить анализ функциональности субъектов КС, что труднореализуемо на практике. Более простой является проверка выполнения в КС $*$-свойства безопасности, описанного в модели Белла — ЛаПадулы.

Моделирование КС с ролевым управлением доступом является в настоящее время одним из самых динамично развивающихся направлений компьютерной безопасности. Основой всех ролевых моделей, как правило, является базовая модель ролевого управления доступом ($RBAC$). В ней используются следующие основные элементы:

U — множество пользователей;

R — множество ролей;

P — множество прав доступа к объектам КС;

S — множество сессий пользователей;

$PA : R \to 2^P$ — функция, задающая для каждой роли множество прав доступа;

$UA : U \to 2^R$ — функция, задающая для каждого пользователя множество ролей, на которые он может быть авторизован;

$user : S \to U$ — функция, задающая для каждой сессии пользователя, от имени которого она активизирована;

$roles : S \to 2^R$ — функция, задающая для пользователя множество ролей, на которые он авторизован в данной сессии.

Для обеспечения возможности большего соответствия реальным КС, каждый пользователь которых занимает определенное положение в служебной иерархии, на множестве ролей реализуется иерархическая структура.

Определение 25. Иерархией ролей в базовой модели ролевого управления доступом называется заданное на множестве ролей R отношение частичного порядка $\leqslant$. При этом по определению выполняется условие: для пользователя $u \in U$, если роли $r, r' \in R$, $r \in UA(u)$ и $r' \leqslant r$, то $r' \in UA(u)$.

Другим важным механизмом базовой модели ролевого управления доступом являются ограничения, накладываемые на множества ролей, на которые может быть авторизован пользователь или на которые он авторизуется в течение одной сессии. Данный механизм также необходим для широкого использования ролевого управления доступом, так как обеспечивает большее соответствие используемым в существующих КС технологиям обработки информации.

Общая структура элементов базовой модели ролевого управления доступом приведена на рис. 15.

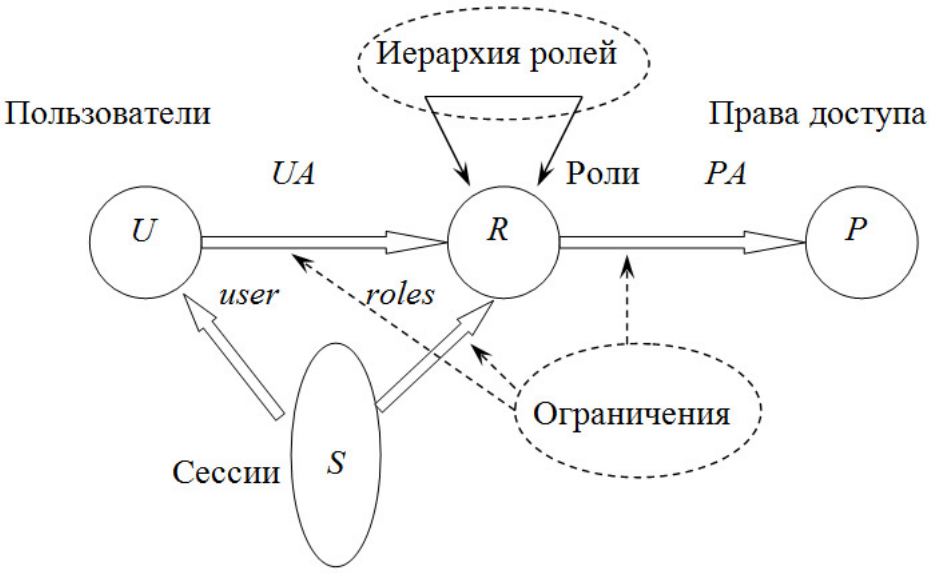

Рис. 15. Структура элементов базовой модели ролевого управления доступом

На основе базовой модели ролевого управления доступом $RBAC$ создана модель ролевого администрирования. В дополнение к используемым элементам базовой модели в этой модели рассматриваются следующие элементы:

AR — множество административных ролей ($AR \cap R = \varnothing$);

AP — множество административных прав доступа ($AP \cap P = \varnothing$);

$APA : AR \to 2^{AP}$ — функция, задающая для каждой административной роли множество административных прав доступа;

$AUA : U \to 2^{AR}$ — функция, задающая для каждого пользователя множество административных ролей, на которые он может быть авторизован.

Кроме того, переопределяется функция:

$roles : S \to 2^R \cup 2^{AR}$ — функция, задающая для пользователя множество ролей, на которые он авторизован в данной сессии.

Как и в базовой модели, в модели администрирования ролевого управления доступом реализуются иерархия административных ролей и механизмы ограничений.

Определение 26. Иерархией административных ролей в модели администрирования ролевого управления доступом называется заданное на множестве ролей AR отношение частичного порядка $\leqslant$. При этом выполняется условие: для $u \in U$, если $r, r' \in AR$, $r \in AUA(u)$ и $r' \leqslant r$, то $r' \in AUA(u)$.

При этом административные роли по своему назначению могут быть разделены на три группы:

— администрирование множеств авторизованных ролей пользователей;
— администрирование множеств прав доступа, которыми обладают роли;
— администрирование иерархии ролей.

Общая структура элементов модели администрирования ролевого управления доступом приведена на рис. 16.

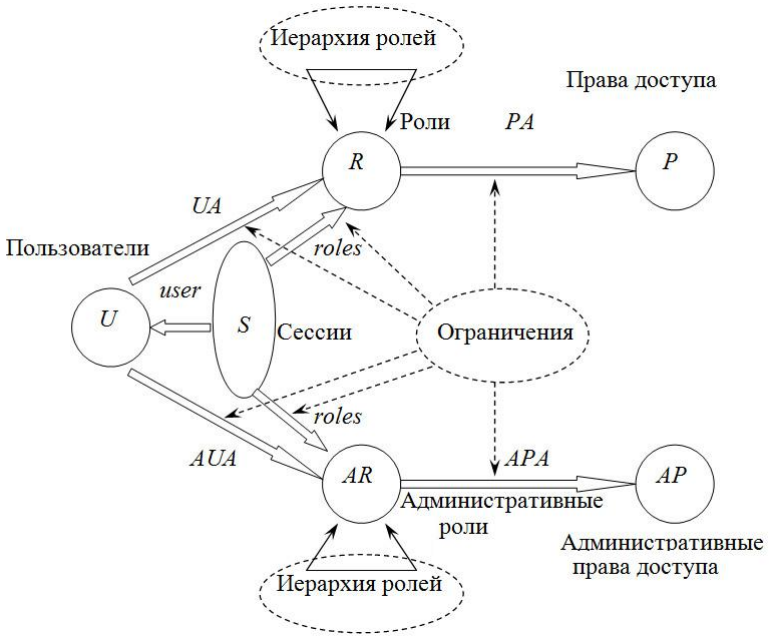

Рис. 16. Структура элементов модели администрирования ролевого управления доступом

Развитием базовой модели ролевого управления доступом $RBAC$ является модель мандатного ролевого управления доступом. В дополнение к используемым элементам базовой модели в этой модели рассматриваются следующие элементы:

O — множество объектов;
$(L, \leqslant)$ — решетка уровней конфиденциальности;
$c : U \to L$ — функция уровней доступа пользователей;
$c : O \to L$ — функция уровней конфиденциальности объектов;
$A = \{read, write\}$ — виды доступа;
$R = \{x\_read | x \in L\} \cup \{x\_write | x \in L\}$ — множество ролей;
$P = \{(o, read) | o \in O\} \cup \{(o, write) | o \in O\}$ — множество прав доступа.

На множестве ролей R задается иерархия. При этом иерархии ролей на множествах $\{x\_read|x \in L\}$ и $\{x\_write|x \in L\}$ независимы.

Определение 27. Иерархией на множестве ролей R в соответствии с требованиями либерального мандатного управления доступом называется отношение частичного порядка $\leqslant$, где для ролей $r, r' \in R$ справедливо неравенство $r \leqslant r'$, когда выполняется одно из условий:

— $r = x\_read, r' = x'\_read$ и $x \leqslant x'$;
— $r = x\_write, r' = x'\_write$ и $x' \leqslant x$.

Для задания мандатного управления доступом в рамках ролевого управления доступом используется следующее определение.

Определение 28. Модель ролевого управления доступом соответствует требованиям либерального мандатного управления доступом, когда иерархия на множестве ролей R соответствует требованиям определения 27, и выполняются ограничения:

— ограничение функции UA: для каждого пользователя $u \in U$ роль $x\_read = \oplus(UA(u) \cap \{y\_read|y \in L\}) \in UA(u)$ (здесь $x = c(u)$, $\oplus$ — наименьшая верхняя граница) и $\{y\_write|y \in L\} \subset UA(u)$;
— ограничение функции $roles$: для каждой сессии $s \in S$ справедливо равенство $roles(s) = \{y\_read|y \in L, y \leqslant x\} \cup \{x\_write\}$;
— ограничение функции PA: должны выполняться условия
  - для каждого $x \in L$ верно: $(o, read) \in PA(x\_read)$ тогда и только тогда, когда $(o, write) \in PA(x\_write)$;
  - для каждого доступа $(o, read) \in P$ существует единственная роль x_read, такая, что $(o, read) \in PA(x\_read)$ (здесь $x = c(o)$).

В рамках модели мандатного ролевого управления доступом даётся определение информационного потока и обосновывается, что невозможна реализация запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Определение 29. Будем считать, что существует информационный поток от объекта $o \in O$ к объекту $o' \in O$ тогда и только тогда, когда существуют роли $r, r' \in R$, сессия $s \in S$, такие, что $(o, read) \in PA(r)$, $(o', write) \in PA(r')$ и $r, r' \in roles(s)$.

Теорема 7. Если модель ролевого управления доступом соответствует требованиям либерального мандатного управления доступом, то в ней для любых объектов $o, o' \in O$, таких, что $c(o) > c(o')$, невозможно возникновение информационного потока от o к o'.

При анализе модели мандатного ролевого управления доступом целесообразно ответить на вопрос о том, каким образом в определениях данной модели реализованы ss-свойство и $*$-свойство, заданные в классической модели Белла — ЛаПадулы.

Часто в рассматриваемых моделях наряду с исследованием условий передачи прав доступа анализируется безопасность информационных потоков. В то же время существуют модели, в основном ориентированные только на анализ условий реализации информационных потоков.

В рамках автоматной модели безопасности информационных потоков анализируются условия информационного невлияния групп пользователей распределенной КС.

В программной модели контроля информационных потоков предлагается механизм построения системы защиты, реализующей дискреционное управление доступом для

командных интерпретаторов. При этом анализируются подходы, позволяющие учитывать зависимость выходного значения программы от её входных параметров.

В вероятностной модели безопасности информационных потоков исследуются КС с мандатным управлением доступом. При этом в модели рассматривается ряд способов определения возможных информационных потоков между высокоуровневыми и низкоуровневыми компонентами системы, основанных на вероятностных понятиях информационной невыводимости и информационного невлияния.

3. Семейство ДП-моделей и направления его развития

Следует отметить, что, как правило, в рассмотренных классических моделях используются оригинальные определения основных элементов и механизмов КС и часто не учитываются следующие существенные особенности функционирования современных КС:

— возможность кооперации части субъектов при передаче прав доступа и создании информационных потоков;
— возможность реализации в КС доверенных и недоверенных субъектов с различными условиями функционирования;
— возможность противодействия доверенными субъектами КС передаче прав доступа или созданию информационных потоков недоверенными субъектами;
— различие условий реализации в КС информационных потоков по памяти и по времени;
— наличие в КС иерархической структуры сущностей и возможность ее использования при создании информационных потоков по времени;
— возможность изменения функциональности субъекта при реализации информационного потока по памяти на функционально ассоциированные с ним сущности или от параметрически ассоциированных с ним сущностей;
— необходимость в ряде случаев определения различных правил управления доступом и информационными потоками для распределенных компонент КС.

С целью обеспечения возможности теоретического анализа условий утечки прав доступа и реализации запрещенных информационных потоков по памяти или по времени с учетом приведенных существенных особенностей современных КС построено семейство формальных моделей безопасности логического управления доступом и информационными потоками (семейство ДП-моделей) [4]. Первоначально в состав этого семейства вошло десять ДП-моделей КС с дискреционным или мандатным управлением доступом.

Основой всех моделей семейства является базовая ДП-модель, построенная с применением положений расширенной модели *Take-Grant*, модели Белла — ЛаПадулы, модели СВС и субъектно-ориентированной модели ИПС. При этом использован классический подход, состоящий в том, что каждая моделируемая КС представляется абстрактной системой, каждое состояние которой описывается графом доступов, а любой переход системы из состояния в состояние осуществляется в результате применения одного из правил преобразования графов доступов.

В рамках базовой ДП-модели рассматриваются следующие основные элементы:

$E = O \cup C$ — множество сущностей, где O — множество объектов, C — множество контейнеров;

$S \subset E$ — множество субъектов;

$R_r = \{read_r, write_r, append_r, execute_r, own_r\}$ — множество видов прав доступа;

$R_a = \{read_a, write_a, append_a\}$ — множество видов доступа;

$R_f = \{write_m, write_t\}$ — множество видов информационных потоков, где $write_m$ — информационный поток по памяти на запись в сущность, $write_t$ — информационный поток по времени на запись в сущность;

$R_{raf} = R_r \cup R_a \cup R_f$ — множество видов прав доступа, видов доступа и видов информационных потоков.

Определение 30. Иерархией сущностей называется заданное на множестве сущностей E отношение частичного порядка $\leqslant$, удовлетворяющее условию: если для сущности $e \in E$ существуют сущности $e_1, e_2 \in E$, такие, что $e \leqslant e_1$, $e \leqslant e_2$, то $e_1 \leqslant e_2$ или $e_2 \leqslant e_1$. В случае, когда для двух сущностей $e_1, e_2 \in E$ выполняются условия $e_1 \leqslant e_2$ и $e_1 \neq e_2$, будем говорить, что сущность e_1 содержится в сущности-контейнере e_2, и будем использовать обозначение $e_1 < e_2$.

Определение 31. Определим $H : E \to 2^E$ — функцию иерархии сущностей, сопоставляющую каждой сущности $c \in E$ множество сущностей $H(c) \subset E$ и удовлетворяющую следующим условиям:

1. Если сущность $e \in H(c)$, то $e < c$ и не существует сущности-контейнера $d \in C$, такой, что $e < d$, $d < c$.

2. Для любых сущностей $e_1, e_2 \in E$, $e_1 \neq e_2$, по определению выполняются равенство $H(e_1) \cap H(e_2) = \varnothing$ и условия:

— если $o \in O$, то выполняется равенство $H(o) = \varnothing$;
— если $e_1 < e_2$, то или $e_1, e_2 \in E \setminus S$, или $e_1, e_2 \in S$;
— если $e \in E \setminus S$, то $H(e) \subset E \setminus S$;
— если $s \in S$, то $H(s) \subset S$.

Определение 32. Пусть определены множества S, E, $R \subset S \times E \times R_r$, $A \subset S \times E \times R_a$, $F \subset E \times E \times R_f$ и H. Определим $G = (S, E, R \cup A \cup F, H)$ — конечный помеченный ориентированный граф без петель, где элементы множеств S, E являются вершинами графа, элементы множества $R \cup A \cup F$ — ребрами графа. Назовем $G = (S, E, R \cup A \cup F, H)$ графом прав доступа, доступов и информационных потоков или, сокращенно, графом доступов. При этом в графе доступов используются следующие обозначения:

— вершины из множества S в графе доступов обозначаются «•»;
— вершины из множества $E \setminus S$ в графе доступов обозначаются «⊗»;
— каждое ребро графа доступов помечено одним из элементов множества R_{raf};
— каждое ребро из множества R обозначается стрелкой вида рис. 17,*а*;
— каждое ребро из множества A обозначается стрелкой вида рис. 17,*б*;
— каждое ребро из множества F, помеченное $write_m$, обозначается стрелкой вида рис. 17,*в*, каждое ребро из множества F, помеченное $write_t$, обозначается стрелкой вида рис. 17,*г*.

Также в базовой ДП-модели используются следующие обозначения:

$\Sigma(G^*, OP)$ — система, при этом каждое состояние системы представляется графом доступов; G^* — множество всех возможных состояний; OP — множество правил преобразования состояний вида: $take\_right(\alpha_r, x, y, z)$, $grant\_right(\alpha_r, x, y, z)$, $own\_take(\alpha_r, x, y)$, $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $rename\_entity(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$ — монотонные правила, $remove\_right(\alpha_r, x, y, z)$, $own\_remove(\alpha_r, x,$

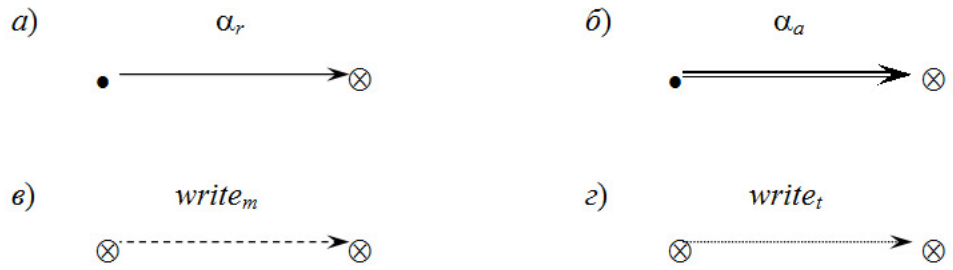

Рис. 17. Обозначения ребер графа доступов

$y)$, $delete\_entity(x, y, z)$ — немонотонные правила; $G \vdash_{op} G'$ — переход системы $\Sigma(G^*, OP)$ из состояния G в состояние G' с использованием правила преобразования состояний $op \in OP$;

$\Sigma(G^*, OP, G_0)$ — система $\Sigma(G^*, OP)$ с начальным состоянием G_0.

Пример применения правила преобразования состояний приведен на рис.18.

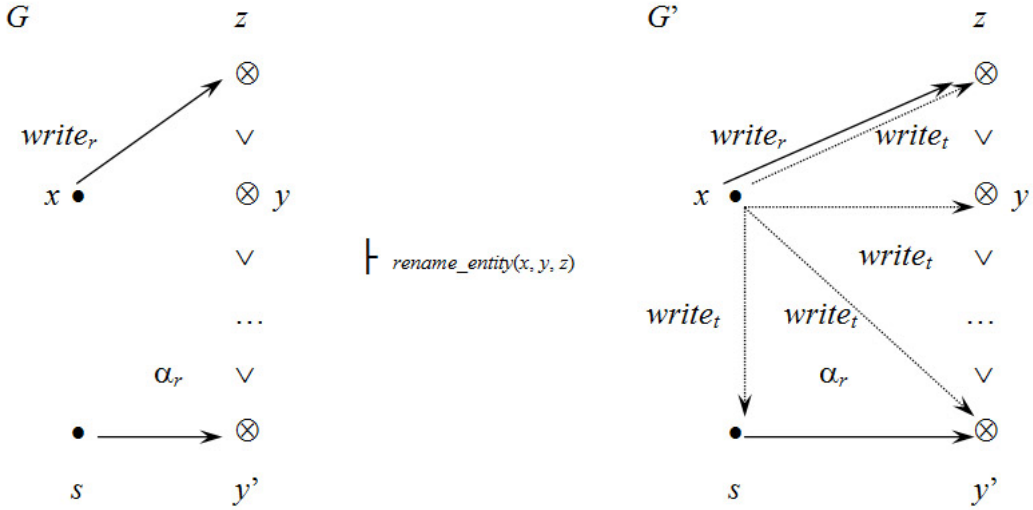

Рис. 18. Пример применения правила $rename\_entity(x, y, z)$

В рамках базовой ДП-модели обосновываются необходимые и достаточные условия передачи в КС прав доступа или реализации информационных потоков по памяти или по времени. Например, при анализе условий возникновения информационных потоков по памяти используются следующие определения и обосновывается следующая теорема.

Определение 33. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$ и субъект $x \in S_0$, сущность $y \in E_0$, где $x \neq y$, и пусть $\alpha \in R_r$ — некоторое право доступа. Определим предикат $can\_share(\alpha, x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, \alpha) \in R_N$.

Определение 34. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, где $x \neq y$. Определим предикат $can\_write\_memory(x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют состоя-

ния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, write_m) \in F_N$.

Теорема 8. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, где $x \neq y$. Предикат $can\_write\_memory(x, y, G_0)$ истинен тогда и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geqslant 2$, таких, что выполняется одно из условий:

1) $m = 2$ и $(x, y, write_m) \in F_0$.

2) Для каждого $i = 1, \ldots, m-1$ выполняется одно из условий:

— $e_i \in S_0$ и или $(e_i, e_{i+1}, write_m) \in F_0$, или истинен предикат $can\_share(\alpha_r, e_i, e_{i+1}, G_0)$, где $\alpha_r \in \{write_r, append_r\}$;
— $e_{i+1} \in S_0$ и истинен предикат $can\_share(read_r, e_{i+1}, e_i, G_0)$;
— $e_i, e_{i+1} \in S_0$ и или истинен предикат $can\_share(own_r, e_i, e_{i+1}, G_0)$, или истинен предикат $can\_share(own_r, e_{i+1}, e_i, G_0)$.

Для анализа КС, в которых все субъекты являются либо доверенными, либо недоверенными, когда доверенные субъекты не кооперируют с недоверенными при передаче прав доступа или реализации информационных потоков, построена ДП-модель без кооперации доверенных и недоверенных субъектов (БК ДП-модель). На ее основе строится ДП-модель с блокирующими доступами доверенных субъектов (БД ДП-модель), в рамках которой анализируются условия реализации в КС запрещенных информационных потоков по времени для случая, когда доверенные субъекты препятствуют использованию недоверенными субъектами иерархии сущностей для создания таких информационных потоков.

На основе БК ДП-модели строится ДП-модель с функционально ассоциированными с субъектами сущностями (ФАС ДП-модель). При этом используются определения.

Определение 35. Пусть $G = (S, E, R \cup A \cup F, H)$ — состояние системы $\Sigma(G^*, OP)$. Назовем сущность $e \in E$ функционально ассоциированной с субъектом $s \in S$ в состоянии G, когда данные в сущности e влияют на вид преобразования данных, реализуемого субъектом s в состоянии G. Всегда по определению субъект s как сущность функционально ассоциирован сам с собой. При этом по определению для каждого субъекта s в любом состоянии системы задано множество функционально ассоциированных с ним сущностей ($[s] \subset E$ — множество всех сущностей, функционально ассоциированных с субъектом s, и выполняется условие $s \in [s]$).

Определение 36. Доверенного субъекта y назовем функционально корректным, когда в множество функционально ассоциированных с ним сущностей $[y]$ не входят недоверенные субъекты.

Определение 37. Доверенного субъекта y назовем корректным относительно сущности e, не являющейся доверенным субъектом, когда субъект y не реализует информационный поток по памяти от сущности e к сущности e', функционально ассоциированной с некоторым доверенным субъектом y'. При этом по определению каждый доверенный субъект корректен относительно другого доверенного субъекта.

К правилам преобразования состояний базовой и БК ДП-моделей в ФАС ДП-модели добавлено правило $control(x, y, z)$, которое позволяет недоверенному субъекту получить право доступа владения к доверенному субъекту с использованием реализации недоверенным субъектом информационного потока по памяти к сущности, функционально ассоциированной с доверенным субъектом. Таким образом, в рамках

ФАС ДП-модели правила преобразования состояний обладают следующими свойствами (рис. 19):

— права доступа могут быть применены для получения прав доступа, доступов и реализации информационных потоков по памяти или по времени;
— информационные потоки по памяти могут быть использованы для получения прав доступа, для реализации информационных потоков по памяти или по времени;
— информационные потоки по времени могут быть использованы только для реализации новых информационных потоков по времени;
— доступы не применяются для получения новых элементов состояний.

Рис. 19. Зависимость условий и результатов применения правил преобразования состояний ФАС ДП-модели

На основе ФАС ДП-модели построена ДП-модель для политики безопасного администрирования (ПБА ДП-модель). Такая модель позволяет анализировать условия обеспечения защиты распределенных КС:

— от захвата нарушителем (недоверенным субъектом) контроля над компьютером, на котором нарушитель разместил свои ресурсы (запустил процесс или разместил файлы);
— от захвата нарушителем прав доступа и привилегий доверенного субъекта, обратившегося (с целью получить данные файлов локально или по сетевым коммуникационным каналам) к компьютеру, на котором разметил свои ресурсы нарушитель.

Кроме ПБА ДП-модели, на основе ФАС ДП-модели построена ДП-модель для политики абсолютного разделения административных и пользовательских полномочий (ПАР ДП-модель). Эта модель направлена на исследование условий обеспечения безопасности рабочих станций пользователей распределенных КС для случая, когда при наличии на рабочей станции активных недоверенных субъектов возможно блокирова-

ние использования любыми субъектами административных прав доступа или привилегий.

Для анализа безопасности КС с мандатным управлением доступом на основе ФАС ДП-модели построена мандатная ДП-модель, в которой дополнительно используются следующие обозначения и определения:

$(L, \leqslant)$ — решетка линейно упорядоченных уровней доступа и конфиденциальности;

$ES \subset E \setminus S$ — множество сущностей, которые могут быть применены для создания новых субъектов (в отличие от дискреционных ДП-моделей, в мандатной ДП-модели субъект может создать субъекта из сущности, когда сущность принадлежит множеству ES);

$f_s : S \to L$ — функция, задающая уровень доступа каждого субъекта;

$f_e : E \setminus S \to L$ — функция, задающая уровень конфиденциальности каждой сущности системы, не являющейся субъектом, при этом если для двух сущностей $e_1, e_2 \in E$ выполняется неравенство $e_1 \leqslant e_2$ (сущность e_1 содержится в контейнере e_2), то по определению выполняется условие $f_e(e_1) \leqslant f_e(e_2)$;

$CCR : E \setminus S \to \{true, false\}$ — функция, задающая способ доступа к сущностям, не являющимся субъектами, внутри контейнеров. Если сущность $e \in E$ является контейнером и доступ к сущностям, содержащимся внутри контейнера e, разрешен без учета уровня конфиденциальности контейнера e, то по определению выполняется равенство $CCR(e) = false$, в противном случае выполняется равенство $CCR(e) = true$. При этом по определению для каждой сущности $e \in E$, являющейся объектом, выполняется условие $CCR(e) = false$.

Определение 38. В состоянии $G = (S, E, R \cup A \cup F, H, (f_s, f_e), CCR)$ системы $\Sigma(G^*, OP)$ доступ $(s, e, \alpha) \in A$, где $s \in S$, $e \in E \setminus S$, $\alpha \in R_a$, обладает ss-свойством, когда выполняются условия:

- $f_s(s) \geqslant f_e(e)$;
- для каждой сущности-контейнера $e' \in E \setminus S$, такой, что $e < e'$ и $CCR(e') = true$, выполняется неравенство $f_s(s) \geqslant f_e(e')$.

Определение 39. В состоянии $G = (S, E, R \cup A \cup F, H, (f_s, f_e), CCR)$ системы $\Sigma(G^*, OP)$ доступы $(s, e_1, read_a)$, $(s, e_2, \alpha) \in A$, где $s \in S$, $e_1, e_2 \in E \setminus S$, $\alpha \in \{write_a, append_a\}$, обладают $*$-свойством, когда выполняется условие $f_e(e_1) \geqslant f_e(e_2)$.

Определение 40. Состояние системы $\Sigma(G^*, OP)$ обладает ss-свойством или $*$-свойством, когда в состоянии все доступы обладают ss-свойством или $*$-свойством соответственно.

Определение 41. Состояние системы $\Sigma(G^*, OP)$ называется безопасным в смысле Белла — ЛаПадулы, когда оно обладает ss-свойством и $*$-свойством. Система $\Sigma(G^*, OP)$ называется безопасной в смысле Белла — ЛаПадулы, когда все состояния системы на всех конечных траекториях ее функционирования безопасны в смысле Белла — ЛаПадулы.

В рамках мандатной ДП-модели обоснована следующая теорема.

Теорема 9 (теорема БТБ-ДП). Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0, (f_{s0}, f_{e0}), CCR_0)$ — начальное состояние системы $\Sigma(G^*, OP, G_0)$, являющееся безопасным в смысле Белла — ЛаПадулы, и $A_0 = F_0 = \varnothing$. Тогда система $\Sigma(G^*, OP, G_0)$ является безопасной в смысле Белла — ЛаПадулы.

При этом показано, что обеспечения безопасности состояний и функции переходов в смысле Белла — ЛаПадулы (в смысле определений ss- и $*$-свойств безопасности) недостаточно для противодействия возможности реализации запрещенных информационных потоков двух видов:

— информационных потоков по времени от сущностей с высоким уровнем конфиденциальности информации к сущностям с низким уровнем конфиденциальности информации;
— информационных потоков по памяти от недоверенных субъектов с низким уровнем доступа к сущностям, функционально ассоциированным с субъектами с высоким уровнем доступа.

Пример информационного потока по времени первого вида от сущности x с высоким уровнем конфиденциальности к сущности y с низким уровнем конфиденциальности, реализуемого двумя кооперирующими субъектами s_x и s_y с использованием находящихся в одной иерархии сущностей z_x и z_y, приведен на рис. 20.

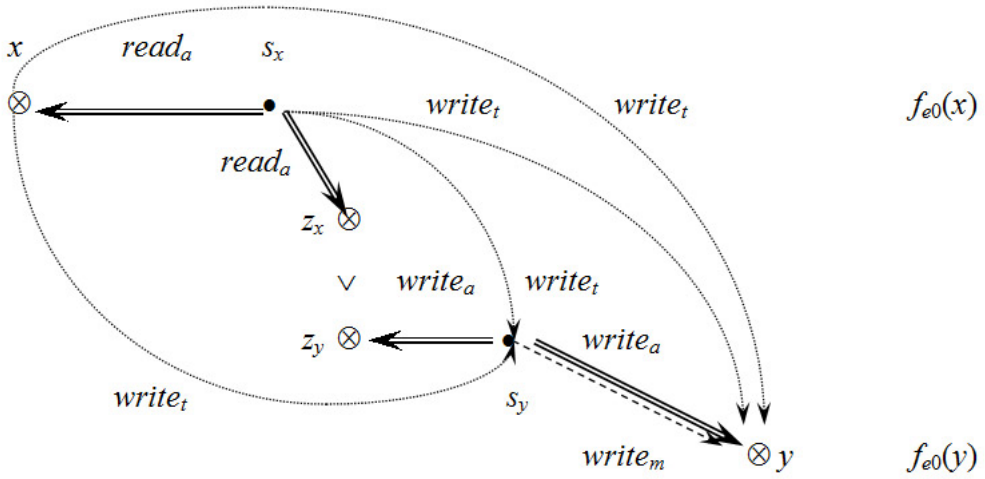

Рис. 20. Пример реализации запрещенного информационного потока по времени

Для анализа условий, выполнение которых в КС с мандатным управлением доступом позволит предотвратить возможность реализации этих запрещенных информационных потоков, на основе мандатной, БД и ФАС ДП-моделей построены следующие три ДП-модели:

— мандатная ДП-модель с блокирующими доступами доверенных субъектов (БДМ ДП-модель), в которой анализируются условия реализации в КС запрещенных информационных потоков для случая, когда доверенные субъекты препятствуют использованию недоверенными субъектами иерархии сущностей для создания информационных потоков по времени;
— мандатная ДП-модель с отождествлением порожденных субъектов (ОСМ ДП-модель), в которой при определении $*$-свойства безопасности рассматриваются доступы недоверенного субъекта вместе со всеми доступами порожденных им субъектов;
— мандатная ДП-модель КС, реализующих политику строгого мандатного управления доступом (ПСМ ДП-модель), в рамках которой недоверенному субъекту разрешается получать доступы только к сущностям с уровнем конфиденциальности, совпадающим с его уровнем доступа.

С использованием БД, ФАС, ПБА, ПАР, мандатной, БДМ, ОСМ и ПСМ ДП-моделей описаны и теоретически обоснованы пять методов, реализация которых в КС с дискреционным или мандатным управлением доступом позволяет предотвратить возможность возникновения некоторых видов запрещенных информационных потоков по памяти или по времени.

Развитие семейства ДП-моделей в настоящее время осуществляется по двум направлениям:

— построение ДП-моделей типовых практически значимых или перспективных КС;
— построение ДП-моделей, содержащих элементы, принципиально новые по сравнению с элементами уже разработанных ДП-моделей.

По первому направлению построена ДП-модель веб-системы [5], в рамках которой осуществлен анализ защищенности типовых веб-систем от угроз реализации атаки вида межсайтового скриптинга (*Cross Site Scripting, XSS*). Граф доступов, описывающий начальное состояние рассматриваемой системы, приведен на рис. 21. При этом использованы следующие обозначения и определения:

s_i — доверенный субъект-интерфейс веб-системы;

s_{h1}, s_{h2} — доверенные субъекты-процессы $HTTP$-сервера, реализующие передачу данных между субъектами рабочих станций пользователей и субъектом-интерфейсом веб-системы;

e_{i1}, e_{i2} — сущности-порты, используемые для обмена данными между субъектами-процессами $HTTP$-сервера и субъектом-интерфейсом веб-системы;

e_{h1}, e_{h2} — сущности-порты, используемые для обмена данными между субъектами-процессами $HTTP$-сервера и субъектами рабочих станций пользователей;

s_u — доверенный субъект-пользователь веб-системы;

e_u — сущность-скрипт, функционально ассоциированная с субъектом-браузером пользователя веб-системы;

s_a — недоверенный субъект-нарушитель;

$N_S = \{s_a\}$ — множество недоверенных субъектов;

$L_S = \{s_i, s_{h1}, s_{h1}, s_u\}$ — множество доверенных субъектов.

Определение 42. Субъектом-нарушителем в веб-системе является недоверенный субъект, имеющий возможность осуществить запрос к интерфейсу веб-системы с параметрами, содержащими данные (в том числе исполняемые скрипты), которые помещаются веб-системой в страницу в формате $HTML$ и передаются доверенному субъекту-браузеру целевого пользователя веб-системы.

Определение 43. Нарушением безопасности веб-системы является получение субъектом-нарушителем контроля (права доступа владения) над доверенным субъектом-браузером пользователя.

Определение 44. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа, когда при ее реализации используются монотонные правила преобразования состояний, и доверенные субъекты (где $u \in L_S$ — доверенный субъект; $x \in N_S$ — недоверенный субъект; $e, e' \in E$ — сущности; $r \in R_r$ — право доступа):

— не инициируют выполнения следующих правил преобразования состояний: $take\_right(\alpha_r, u, x, e)$, $grant\_right(\alpha_r, u, x, e)$, $control(x, y, z)$ (доверенные субъекты не

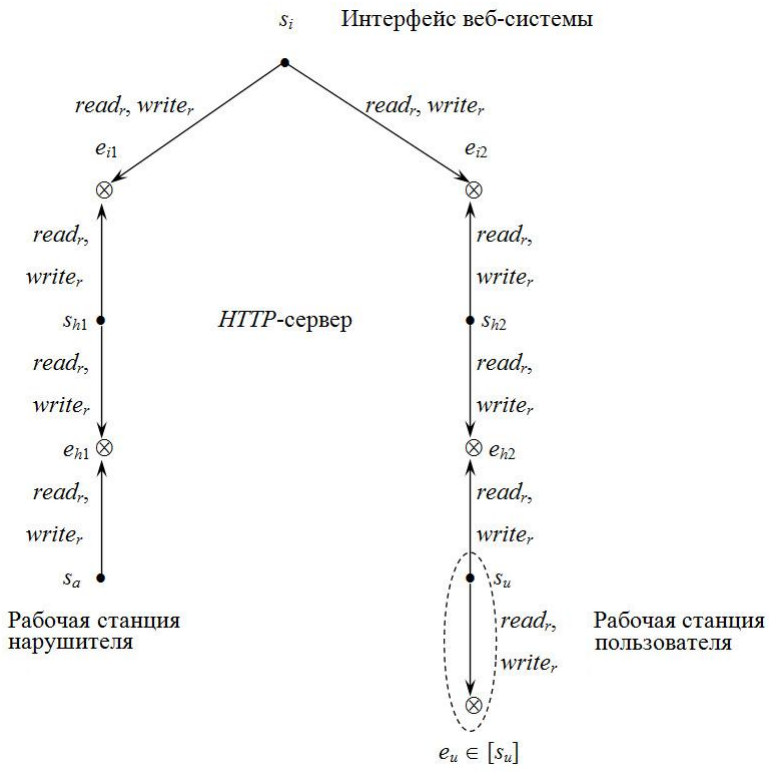

Рис. 21. Начальное состояние системы в рамках ДП-модели веб-системы

дают недоверенным субъектам права доступа к сущностям; не берут у недоверенных субъектов права доступа к сущностям; используя информационные потоки по памяти к сущностям, не получают право доступа владения к субъектам);

— могут выполнять монотонные правила преобразования состояний $own\_take(\alpha_r, u, e)$, $create\_entity(u, e, e')$, $create\_subject(u, e, e')$, $rename\_entity(u, e, e')$, $access\_read(u, e)$, $access\_write(u, e)$, $access\_append(u, e)$, $find(u, e, e')$, $post(u, e, e')$, $pass(u, e, e')$ с условиями и результатами применения, заданными в БК ДП-модели.

Определение 45. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков, когда она является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и при ее реализации используются правила преобразования состояний:

— $take\_right(\alpha_r, x, y, z)$, $grant\_right(\alpha_r, x, y, z)$, $own\_take(\alpha_r, x, y)$ с условиями и результатами применения, заданными в базовой ДП-модели;
— $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $rename\_entity(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$ с условиями и результатами применения, заданными в БК ДП-модели;
— $control(x, y, z)$ с условиями и результатами применения, заданными в ФАС ДП-модели.

При этом в состояниях траектории по определению отсутствуют информационные потоки по времени, исходящие из доверенных субъектов.

Определение 46. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$ и недоверенный субъект $x \in N_S \cap S_0$, субъект $y \in S_0$, где $x \neq y$. Определим предикат $can\_share\_own(x, y, G_0, L_S)$, который будет истинным тогда и только тогда,

когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков, и $(x, y, own_r) \in R_N$.

В результате анализа безопасности веб-системы обоснована следующая теорема.

Теорема 10. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — начальное состояние системы $\Sigma(G^*, OP, G_0)$, в котором субъект-интерфейс веб-системы s_i является некорректным относительно сущности e_{i1}. Тогда предикат $can\_share\_own(s_a, s_u, G_0, L_S)$ является истинным, и возможно нарушение безопасности веб-системы в смысле определения 43.

Аналогичный подход [6], использующий ФАС ДП-модель, можно применить для анализа защищенности веб-систем на основе СУБД от атак с применением SQL-инъекции.

По второму направлению развития ДП-моделей в [7] рассмотрен случай, когда в КС могут существовать параметрически ассоциированные с субъектами сущности, реализация от которых информационных потоков по памяти к недоверенным субъектам позволяет им получить контроль над другими субъектами системы, в том числе доверенными. Например, возможно получение субъектом-нарушителем права доступа на чтение к конфигурационному файлу операционной системы, в котором хранится пароль или хеш-образ пароля доверенного субъекта. Наличие такого права позволяет субъекту-нарушителю получить контроль над доверенным субъектом. В данном случае конфигурационный файл будет являться сущностью, параметрически ассоциированной с доверенным субъектом. Таким образом, на основе ФАС ДП-модели построена ДП-модель с функционально или параметрически ассоциированными с субъектами сущностями (ФПАС ДП-модель). При этом в ФПАС ДП-модели добавлено правило преобразования состояний $know(x, y)$, позволяющее недоверенному субъекту x при условии реализации им к себе информационных потоков по памяти от всех сущностей, параметрически ассоциированных с субъектом y, получить контроль (право доступа владения own_r) к субъекту y.

Пример применения правила $know(x, y)$ приведен на рис. 22. При этом использовано обозначение:

$]y[\in E \setminus S$ — множество сущностей, параметрически ассоциированных с субъектом $y \in S$.

В рамках ФПАС ДП-модели выполнен теоретический анализ условий утечки прав доступа и реализации запрещенных информационных потоков по памяти.

Также по второму направлению на основе ФАС и ФПАС ДП-моделей в [8] построена ДП-модель файловых систем (ФС ДП-модель). В данной модели анализируется характерный для файловых систем новый вид доверенных субъектов — потенциальных доверенных субъектов. Из потенциальных доверенных субъектов могут быть созданы доверенные субъекты, реализующие доступ к сущностям, защищенным механизмами файловых систем (например, механизмами файловой системы EFS в среде ОС семейства *Windows XP*/2003/*Vista*). Создать доверенного субъекта из потенциального доверенного субъекта могут доверенные или недоверенные субъекты, имеющие доступ на чтение к сущностям, параметрически ассоциированным с потенциальными доверенными субъектами. Таким образом, в ФС ДП-модели используется следующее определение.

Определение 47. В рамках ФС ДП-модели по определению выполняются следующие условия.

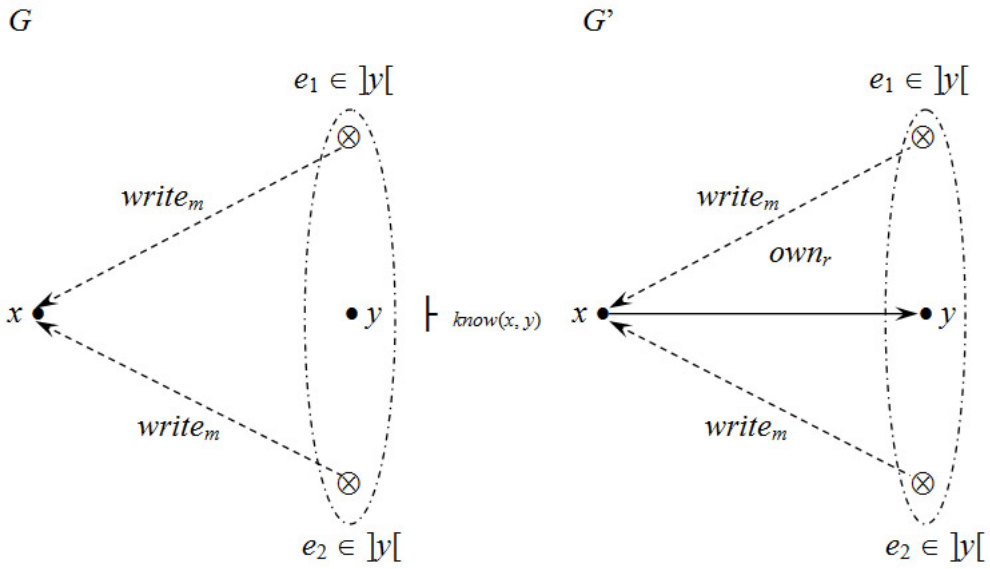

Рис. 22. Пример применения правила $know(x, y)$, где справедливо равенство $]y[= \{e_1, e_2\}$

1. В множестве сущностей выделено подмножество сущностей, защищенных ФС и не являющихся субъектами.

2. В множестве доверенных субъектов выделено подмножество субъектов, обладающих правами доступа и реализующих доступ к сущностям, защищенным ФС, и кодирование в них данных в случае, когда оно осуществляется ФС. Эти доверенные субъекты реализуют информационные потоки по памяти между каждой сущностью, защищенной ФС, и соответствующей ей сущностью-образом, не являющейся субъектом.

3. Доверенные или недоверенные субъекты, не реализующие доступ к сущностям, защищенным ФС, не обладают правами доступа и не могут получать доступ к этим сущностям. При этом они могут обладать правами доступа или получать доступ к сущностям-образам сущностей, защищенных ФС.

4. В каждом состоянии системы кроме множества субъектов анализируется множество потенциальных доверенных субъектов (доверенных субъектов, которые могут быть созданы в процессе функционирования системы для реализации доступа к сущностям, защищенным ФС).

5. Кроме возможности создания новых субъектов из сущностей, недоверенный субъект может создать доверенного субъекта в случае, когда недоверенный субъект реализовал к себе информационные потоки по памяти от всех сущностей, параметрически ассоциированных с потенциальным доверенным субъектом. При этом недоверенный субъект получает контроль над созданным доверенным субъектом.

6. Каждый доверенный субъект не обладает правами доступа ко всем сущностям.

7. Доверенные субъекты, не реализующие доступ к сущностям, защищенным ФС, в процессе функционирования системы не получают новые доступы к сущностям и не участвуют в создании информационных потоков к или от сущностей, защищенных ФС.

8. Не рассматриваются информационные потоки по времени, право доступа и доступ на запись в конец сущности.

9. В начальном состоянии системы недоверенные субъекты не реализуют доступы к сущностям, к ним не имеют доступы другие субъекты, и отсутствуют информационные потоки по памяти с участием недоверенных субъектов.

Кроме обозначений ФПАС ДП-модели, в ФС ДП-модели используются следующие обозначения:

$FSE \subset E \setminus S$ — множество сущностей, защищенных ФС;

$fs : FSE \to E \setminus S$ — инъективная функция, которая ставит в соответствие каждой сущности, защищенной ФС, соответствующую ей сущность-образ;

PS — множество потенциальных доверенных субъектов, реализующих доступ к сущностям из множества FSE;

$FSS \subset L_S \cap S$ — множество доверенных субъектов, реализующих доступ к сущностям из множества FSE.

При этом в ФС ДП-модель добавлено правило преобразования состояний $potential\_subject(x, y, z)$, позволяющее недоверенному субъекту создать доверенного субъекта из потенциального доверенного субъекта (рис. 23).

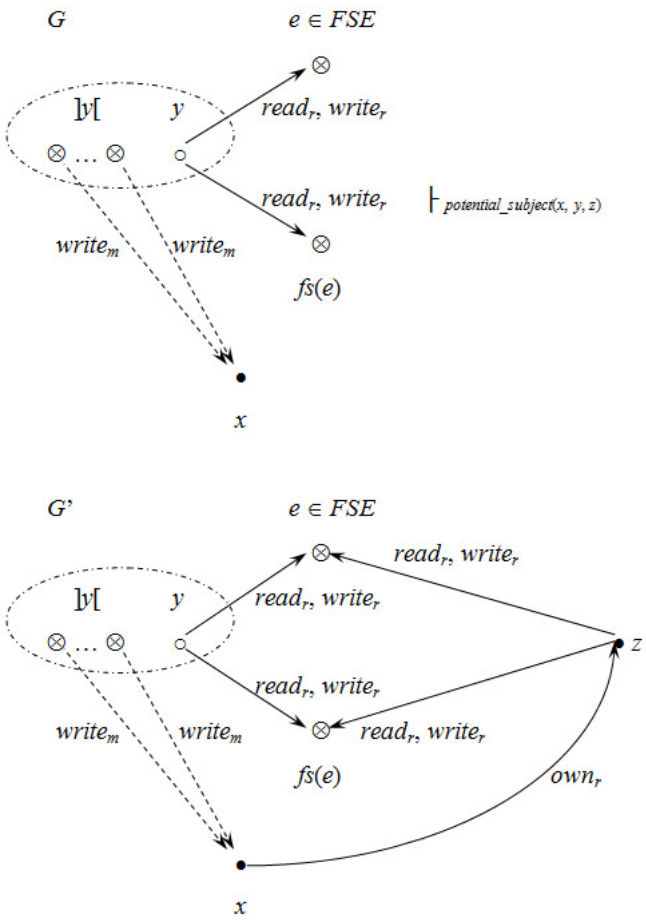

Рис. 23. Пример применения правила $potential\_subject(x, y, z)$

В рамках ФС ДП-модели проведён анализ условий реализации информационных потоков. В том числе приведены и обоснованы достаточные условия, при выполнении которых в КС возможна реализация запрещенных информационных потоков по памяти от сущностей, защищенных механизмами файловых систем.

Теорема 11. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, где $x \neq y$. Предикат $can\_write\_memory(x, y, G_0)$ является истинным в случае, когда существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geqslant 2$, таких, что для каждого $i = 1, \ldots, m-1$ выполняется одно из условий:

1) $e_i \in L_s \cap S_0$ и или $(e_i, e_{i+1}, write_m) \in F_0$, или $(e_i, e_{i+1}, write_a) \in A_0$.

2) $e_i \in FSS_0 \cup (N_S \cap S_0)$ и или $(e_i, e_{i+1}, write_m) \in F_0$, или истинен предикат $can\_share(write_r, e_i, e_{i+1}, G_0)$.

3) $e_{i+1} \in L_S \cap S_0$ и $(e_{i+1}, e_i, read_a) \in A_0$.

4) $e_{i+1} \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $can\_share(read_r, e_{i+1}, e_i, G_0)$.

5) $e_i \in N_S \cap S_0$, $e_{i+1} \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $can\_share\_own(e_i, e_{i+1}, G_0)$.

6) $e_{i+1} \in N_S \cap S_0$, $e_i \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $can\_share\_own(e_{i+1}, e_i, G_0)$.

Развитием семейства ролевых моделей $RBAC$ и семейства ДП-моделей КС с дискреционным или мандатным управлением доступом является базовая ролевая ДП-модель (БР ДП-модель) [9, 10]. Данная модель ориентирована на анализ в КС с ролевым управлением доступом условий передачи прав доступа ролей и реализации информационных потоков по памяти и по времени. В ней кроме обозначений дискреционных ДП-моделей используются следующие обозначения:

U — множество пользователей;

L_U — множество доверенных пользователей;

N_U — множество недоверенных пользователей;

$S \subseteq E$ — множество субъект-сессий пользователей;

L_S — множество доверенных субъект-сессий;

N_S — множество недоверенных субъект-сессий;

R — множество ролей;

AR — множество административных ролей;

$UA : U \to 2^R$ — функция авторизованных ролей пользователей;

$AUA : U \to 2^{AR}$ — функция авторизованных административных ролей пользователей;

$PA : R \to 2^P$ — функция прав доступа ролей;

$user : S \to U$ — функция принадлежности субъект-сессии пользователю;

$roles : S \to 2^R \cup 2^{AR}$ — функция текущих ролей субъект-сессий;

$can\_manage\_rights : AR \to 2^R$ — функция администрирования прав доступа ролей;

$H_R : R \to 2^R$ — функция иерархии ролей;

$H_{AR} : AR \to 2^{AR}$ — функция иерархии административных ролей;

$G = (PA, user, roles, A, F, H_E)$ — состояние системы;

$fa : U \times E \to 2^E \cup 2^U$ — функция, задающая множества сущностей, функционально ассоциированных с субъект-сессией, при ее создании пользователем (или от имени пользователя другой субъект-сессией) из сущности;

$y(E) \subset L_S \times E$ — множество пар вида (доверенная субъект-сессия, сущность), относительно которых корректна доверенная субъект-сессия y.

В КС с ролевым управлением доступом право доступа к сущности может быть получено субъект-сессией только через обладание ролью, содержащей данное право.

Реализация субъект-сессией информационного потока по памяти на сущность, функционально ассоциированную с другой субъект-сессией, позволит первой субъект-сессии получить контроль над второй субъект-сессией, включая возможность использовать права доступа ее ролей. При этом множество текущих ролей первой субъект-сессии, как правило, останется неизменным (кроме того, изменение множества текущих ролей может быть заблокировано реализованным в КС механизмом статических и, особенно, динамических ограничений). Таким образом, в рамках БР ДП-модели кроме ролей, прав доступа ролей и возможностей осуществить действия, которыми явно обладают субъект-сессии, рассматриваются фактические роли, фактические права доступа ролей и фактические возможности осуществить действия, которыми обладают субъект-сессии за счет получения контроля над другими субъект-сессиями. При этом используются следующие обозначения:

$de\_facto\_roles : S \to 2^{R \cup AR}$ — функция фактических текущих ролей субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s_1 \in S$ выполняется равенство

$de\_facto\_roles(s_1) = roles(s_1) \cup \{r \in R \cup AR : \exists s_2 \in S\ [(s_1, s_2, own_a) \in A\ \&\ \&\ r \in roles(s_2)]\}$;

$de\_facto\_rights : S \to 2^P$ — функция фактических текущих прав доступа субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s \in S$ выполняется равенство

$de\_facto\_rights(s) = \{p \in P : \exists r \in de\_facto\_roles(s)\ [p \in PA(r)]\}$;

$de\_facto\_actions : S \to 2^P \times 2^R$ — функция фактических возможных действий субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s_1 \in S$ выполняется равенство

$de\_facto\_actions(s_1) = (PA(roles(s_1)) \times can\_manage\_rights(roles(s_1) \cap AR)) \cup \cup \{(p, r) \in P \times R : \exists s_2 \in S\ \exists (s_1, s_2, own_a) \in A\ [r \in can\_manage\_rights(roles(s_2) \cap \cap AR)\ \&\ p \in PA(roles(s_2))]\}$.

В БР ДП-модели определены следующие правила преобразования состояний:

— монотонные: $take\_role(x, r)$, $grant\_right(x, r, (y, \alpha_r))$, $create\_entity(x, r, y, z)$, $create\_first\_session(u, r, y, z)$, $create\_session(x, w, r, y, z)$, $rename\_entity(x, y, z)$, $control(x, y, z)$, $access\_own(x, y)$, $take\_access\_own(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$, $take\_flow(x, y)$;

— немонотонные: $remove\_role(x, r)$, $remove\_right(x, r, (y, \alpha_r))$, $delete\_entity(x, y, z)$.

Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели показана на рис. 24.

По аналогии с базовой ДП-моделью может быть доказано, что в рамках БР ДП-модели при анализе условий передачи прав доступа, реализации информационных потоков по памяти или по времени можно обойтись применением только монотонных правил преобразования состояний.

В настоящее время в рамках БР ДП-модели не удалось завершить исследование КС с ролевым управлением доступом, на условия функционирования которых не наложено ограничений. В связи с этим обстоятельством анализ необходимых и достаточных условий передачи прав доступа выполнен для двух случаев. Первый случай — когда в КС существуют только две субъект-сессии двух пользователей. Второй случай — когда в КС функционирует произвольное число субъект-сессий, и они не получают

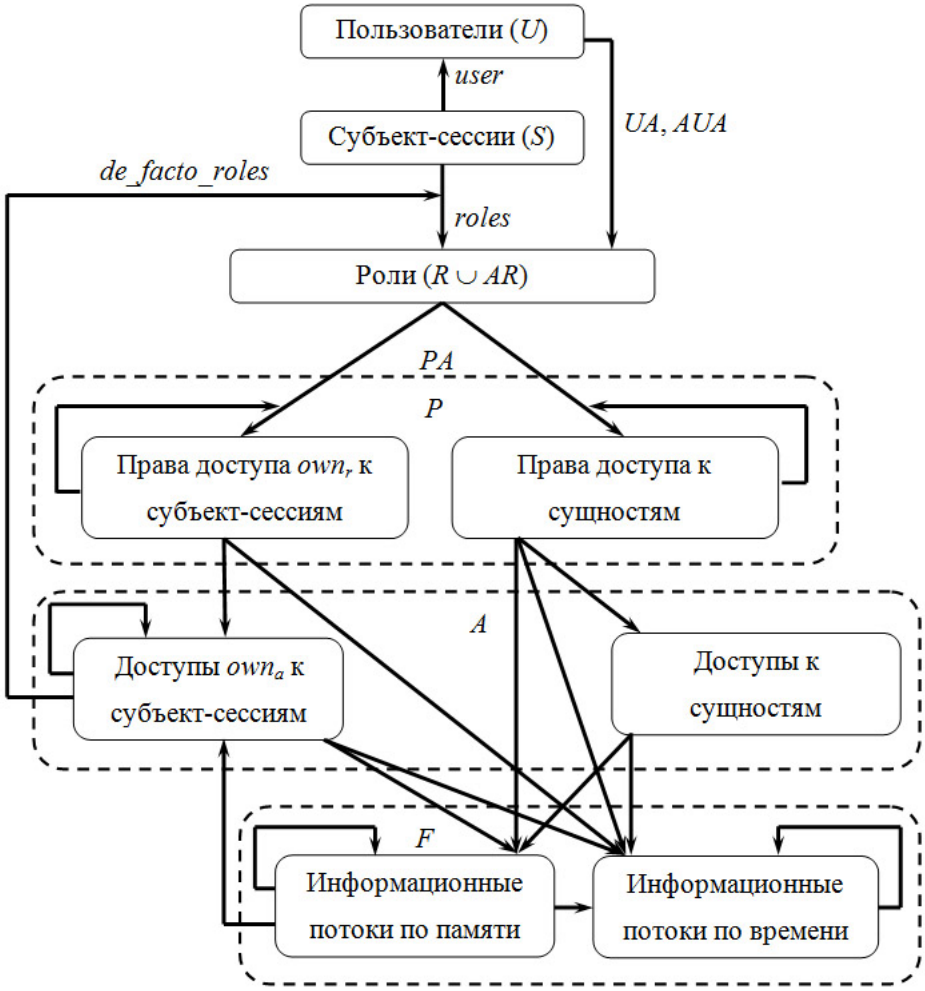

Рис. 24. Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели

доступа владения друг к другу с использованием информационных потоков по памяти к функционально ассоциированным с субъект-сессиями сущностям.

В дальнейшем, в том числе в ходе учебного процесса (например, при выполнении курсовых и дипломных работ), целесообразно развитие и исследование ролевых ДП-моделей по следующим направлениям:

— разработка моделей, адекватных условиям функционирования существующих КС;
— построение ДП-моделей существующих КС для формального анализа их безопасности;
— анализ условий кооперации доверенных и недоверенных субъект-сессий при реализации информационных потоков;
— построение алгоритмов замыкания графа доступов, описывающего состояние системы;
— анализ условий передачи прав доступа ролей для случая произвольного числа субъект-сессий, а также анализ условий возникновения информационных потоков по памяти или по времени;
— исследование способов использования динамических ограничений;

— исследование возможности реализации мандатного ролевого управления доступом с применением динамических ограничений, не позволяющих порождать запрещенные информационные потоки по времени.

Заключение

При преподавании моделей безопасности КС целесообразно учесть, что **бакалавру** следует изучить как минимум следующие модели:

— модель ХРУ;
— классическую модель *Take-Grant*;
— субъектно-ориентированную модель ИПС;
— классическую модель Белла — ЛаПадулы;
— базовую модель ролевого управления доступом.

Дополнительно **специалисту** (особенно по специальностям «Компьютерная безопасность» и «Информационно-аналитические системы безопасности») и **магистру** целесообразно изучить следующие модели:

— модель ТМД;
— расширенную модель *Take-Grant*;
— интерпретации модели Белла — ЛаПадулы;
— модель СВС;
— модели безопасности информационных потоков;
— модель администрирования ролевого управления доступом и модель мандатного ролевого управления доступом;
— базовую ДП-модель, БК ДП-модель, ФАС ДП-модель и ФПАС ДП-модель.

Кроме того, в случае проведения специалистом или магистром теоретических исследований в области безопасности логического управления доступом или информационными потоками в КС целесообразно изучение других ДП-моделей, соответствующих выбранному направлению исследований.

ЛИТЕРАТУРА

1. *Bishop M.* Computer Security: art and science. ISBN 0-201-44099-7. 2002. 1084 p.
2. *Девянин П. Н.* Модели безопасности компьютерных систем: учеб. пособие для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2005. 144 с.
3. *Щербаков А. Ю.* Современная компьютерная безопасность. Теоретические основы. Практические аспекты: учеб. пособие. М.: Книжный мир, 2009. 352 с.
4. *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006. 176 с.
5. *Назаров И. О.* Анализ безопасности веб-систем, в условиях реализации уязвимости класса межсайтового скриптинга // Проблемы информационной безопасности. Компьютерные системы / под ред. П. Д. Зегжды. СПб.: СПбГТУ, 2007. Вып. 2. С. 105–117.
6. *Назаров И. О.* Обеспечение безопасности управления доступом и информационными потоками в веб-системе на основе СУБД // Вестник Казанского государственного технического университета им. А. Н. Туполева. Казань: КГТУ им. А. Н. Туполева, 2008. Вып. 2. С. 56–59.
7. *Колегов Д. Н.* ДП-модель компьютерной системы с функционально и параметрически ассоциированными с субъектами сущностями // Вестник Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнева. 2009. Вып. 1(22). Ч. 1. С. 49–54.

8. *Буренин П. В.* Подходы к построению ДП-модели файловых систем // Прикладная дискретная математика. 2009. № 1(3). С. 93–112.
9. *Девянин П. Н.* Базовая ролевая ДП-модель // Прикладная дискретная математика. 2008. № 1(1). С. 64–70.
10. *Девянин П. Н.* Анализ условий получения доступа владения в рамках базовой ролевой ДП-модели без информационных потоков по памяти // Прикладная дискретная математика. 2009. № 3(5). С. 69–84.