МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/3/6

УДК 004.94

ПОДХОДЫ К ПОСТРОЕНИЮ ДП-МОДЕЛИ ФАЙЛОВЫХ СИСТЕМ

П. В. Буренин

ООО «Твест», г. Тверь

E-mail: troy1f4@mail.ru

В статье приводятся подходы к созданию ДП-модели файловых систем, в основе которой используется семейство ДП-моделей компьютерных систем с дискреционным управлением доступом. В рамках ДП-модели рассматриваются специфичные для файловых систем условия функционирования субъектов, условия передачи прав доступа и реализации информационных потоков, а также обосновываются достаточные условия реализации в файловых системах запрещенных информационных потоков по памяти.

Ключевые слова: *компьютерная безопасность, файловые системы, ДП-модель.*

С целью обеспечения возможности анализа условий получения недоверенными субъектами контроля над доверенными субъектами, реализующими механизмы защиты файловых систем (ФС), или условий создания недоверенными субъектами информационных потоков по памяти в обход механизмов защиты ФС построим на основе ДП-модели с функционально-ассоциированными с субъектами сущностями (ФАС ДП-модели, [1]) и разработанной Д. Н. Колеговым модели с функционально- и параметрически-ассоциированными с субъектами сущностями с дискреционным управлением доступом (ФПАС ДП-модели) ДП-модель файловых систем (или, сокращенно, ФС ДП-модель).

При этом для построения ФС ДП-модели в ФАС и ФПАС ДП-модели внесены изменения, позволяющие учитывать существенные особенности реализации механизмов защиты современных ФС. Таким образом, в дальнейшем используем следующее предположение.

Предположение 1. В рамках ФС ДП-модели выполняются следующие условия.

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

Условие 2. Во 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.

Условие 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.

Условие 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).

Условие 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.

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

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

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

Условие 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.

В основе ФС ДП-модели использован классический подход (используемый, в том числе, в семействе ДП-моделей КС с дискреционным, мандатным или ролевым управлением доступом), состоящий в том, что каждая моделируемая КС представляется абстрактной системой, каждое состояние которой представляется графом доступов, каждый переход системы из состояния в состояние осуществляется в результате применения одного из правил преобразования графов доступа.

В рамках предположения 1 используем следующие обозначения и определения ФАС и ФПАС ДП-моделей:

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

— $S \subset E$ — множество субъектов;

— $[s] \subset E$ — множество всех сущностей, функционально ассоциированных с субъектом s (при этом по определению выполняется условие $s \in [s]$, и для каждого субъекта множество сущностей, функционально с ним ассоциированных, не изменяется в процессе функционирования системы);

— $]s[\subset E$ — множество всех сущностей, параметрически ассоциированных с субъектом и потенциальным субъектом s (при этом по определению для каждого субъекта множество сущностей, параметрически с ним ассоциированных, не изменяется в процессе функционирования системы);

— L_S — множество доверенных субъектов;

— N_S — множество недоверенных субъектов, при этом по определению выполняется равенство $L_S \cap N_S = \varnothing$;

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

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

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

Определение 1. 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 $e \leqslant e_2$, $e \leqslant e_1$, 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 2. 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:

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

Условие 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 $H(s) \subset S$.

В рамках предположения 1 в ФС ДП-модели дополнительно используем 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;

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

Будем считать, что в дальнейшем выполняется следующее предположение.

Предположение 2. В рамках ФС ДП-модели выполняются следующие условия.

Условие 1. Каждый доверенный субъект из множества FSS является функционально корректным, корректным относительно любой сущности и может обладать только правами доступа на чтение и запись к сущностям из множества FSE и соответствующим им сущностям-образам.

Условие 2. Каждый потенциальный доверенный субъект из множества PS может обладать только правами доступа на чтение и запись к сущностям из множества FSE и соответствующим им сущностям-образам и не может реализовать доступы к любым сущностям или информационные потоки.

Условие 3. Из потенциального доверенного субъекта из множества PS может быть создан только доверенный субъект из множества FSS. При этом множество PS не изменяется в процессе функционирования системы.

Условие 4. Для каждого доверенного субъекта из множества FSS или потенциального доверенного субъекта из множества PS множество параметрически ассоциированных с ним сущностей не пусто (для каждого $s \in FSS \cup PS$ справедливо неравенство $]s[\neq \varnothing$). Для каждого доверенного субъекта из множества FSS множество функционально ассоциированных с ним сущностей состоит только из самого субъекта (для каждого $s \in FSS$ справедливо равенство $[s] = \{s\}$), и невозможно получение к нему права доступа владения с использованием реализованного к нему информационного потока по памяти.

Условие 5. Для каждой сущности из множества FSE существует доверенный субъект из множества FSS или потенциальный доверенный субъект из множества PS, обладающий правами доступа на чтение и запись к сущности и к соответствующей ей сущности-образу (для каждой $e \in FSE$ существует субъект $s \in FSS \cup PS$, обладающий правами доступа $(s, e, read_r)$, $(s, e, write_r)$, $(s, fs(e), read_r)$, $(s, fs(e), write_r)$).

Условие 6. Каждый доверенный субъект, не входящий во множество FSS, обладает всеми правами доступа ко всем сущностям, не входящим во множества FSE, FSS и $\{e \in E$: существует $s \in S$ и $e \in]s[\}$ (множество сущностей, параметрически ассоциированных с субъектами).

В рамках предположений 1 и 2 дадим определение состояния системы.

Определение 3. Пусть определены множества S, PS, E, $R \subseteq (S \cup PS) \times E \times R_r$, $A \subseteq S \times E \times R_a$, $F \subseteq E \times E \times R_f$ и функция иерархии сущностей H. Определим $G{=}\ (S, E, R \cup A \cup F, H)$ — конечный помеченный ориентированный граф, без петель, где элементы множеств S, PS, 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 будем использовать следующие обозначения:

— вершины из множества S (соответствующие субъектам) в графе доступов будут обозначаться «•»;
— вершины из множества PS (соответствующие потенциальным субъектам) в графе доступов будут обозначаться «○»;
— вершины из множества $E \setminus S$ (соответствующие сущностям, не являющимся субъектами) в графе доступов будут обозначаться «⊗»,
— каждое ребро графа доступов помечено одним из элементов множества $R_r \cup R_a \cup R_f$;
— каждое ребро из множества R будет обозначаться стрелкой вида, представленного на рис. 1,*а*;
— каждое ребро из множества A будет обозначаться стрелкой вида, представленного на рис. 1,*б*;
— каждое ребро из множества F, помеченное $write_m$, будет обозначаться стрелкой вида, представленного на рис. 1,*в*.

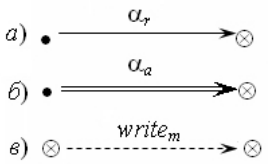

Рис. 1. Обозначения ребер графа доступов: *а* — ребро из множества R, помеченное $\alpha_r \in R_r$; *б* — ребро из множества A, помеченное $\alpha_a \in R_a$; *в* — ребро из множества F, помеченное $write_m$

Используем также обозначения:
$\Sigma(G^*, OP)$ — система, при этом:

— каждое состояние системы представляется графом доступов;
— G^* — множество всех возможных состояний;
— OP — множество правил преобразования состояний, определенных в таблице.

$G \vdash_{op} G'$ — переход системы $\Sigma(G^*, OP)$ из состояния G в состояние G' с использованием правила преобразования состояний $op \in OP$.

Если для системы $\Sigma(G^*, OP)$ определено начальное состояние, то будем использовать обозначение:

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

При анализе правил преобразования состояний и траекторий функционирования системы, в результате реализации которых возникают запрещенные информационные потоки по памяти, применим подход, аналогичный использованному в рамках ФАС ДП-модели и ФПАС ДП-модели. Кроме того, в соответствии с условием 7 предположения 1 доверенные субъекты, не входящие во множество *FSS*, в процессе функционирования системы не реализуют новые доступы к сущностям. В противном случае, любой недоверенный субъект с помощью доверенных субъектов мог бы реализовать к себе информационный поток по памяти от любой сущности, защищенной ФС, для которой в системе существует соответствующий ей доверенный субъект из множества *FSS*. Таким образом, будем считать, что в дальнейшем выполняется следующее предположение.

Предположение 3. В процессе функционирования системы доверенные субъекты:

- не дают недоверенным субъектам права доступа к сущностям;
- не берут у недоверенных субъектов права доступа к сущностям;
- не получают прав доступа владения к другим субъектам;
- не создают доверенных субъектов из потенциальных доверенных субъектов.

Кроме того, доверенные субъекты, не входящие во множество *FSS*, не реализуют новые доступы к сущностям.

На основе предположения 3 дадим определение.

Определение 4. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков по памяти, если при ее реализации используются монотонные правила преобразования состояний, и:

- доверенные субъекты не инициируют выполнения следующих правил преобразования состояний: $take\_right(\alpha_r, u, x, e)$, $grant\_right(\alpha_r, u, x, e)$, $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $potential\_subject(u, ps, y)$, $control(u, y, e)$, $know(u, y)$, $access\_read(u, e)$, $access\_write(u, e)$;
- доверенные субъекты могут инициировать выполнение правил преобразования состояний: $own\_take(\alpha_r, u, e)$, $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $find(u, e, e')$, $post(u, e, e')$, $pass(u, e, e')$,

где $u, y \in L_S$ — доверенные субъекты, $x \in N_S$ — недоверенный субъект, $ps \in PS$ — потенциальный доверенный субъект, e, e' — сущности, $\alpha_r \in R_r$ — право доступа.

Таким образом, в рамках предположений 1 и 2 в дальнейшем будем рассматривать траектории без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков по памяти. При этом по сравнению с ФАС ДП-моделью и ФПАС ДП-моделью в ФС ДП-модели (таблица):

- заданы без использования информационных потоков по времени, права доступа $append_r$ и доступа $append_a$ условия и результаты применения монотонных правил преобразования состояний: $take\_right(\alpha_r, x, y, z)$, $grant\_right(\alpha_r, x, y, z)$, $own\_take(\alpha_r, x, y)$, $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $control(x, y, z)$, $know(x, y)$, $access\_read(x, y)$, $access\_write(x, y)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$;

— для обеспечения возможности создания доверенными субъектами информационных потоков по памяти при наличии у них доступов к сущностям изменены условия применения правил *find*(x, y, z), *post*(x, y, z), *pass*(x, y, z);
— не рассматриваются правила *flow*(x, y, y', z) и *rename_entity*(x, y, z), так как в результате их применения только информационные потоки по времени;
— не рассматривается правило *access_append*(x, y), так как оно используется для получения к сущности доступа $append_a$ с применением права доступа $append_r$;
— добавлено новое правило *potential_subject*(x, y, z), позволяющее недоверенному субъекту создать доверенного субъекта из потенциального доверенного субъекта.

Правила преобразования состояний ФС ДП-модели

Правило	Исходное состояние $G = (S, E, R \cup A \cup F, H)$	Результирующее состояние $G' = (S', E', R' \cup A' \cup F', H')$
1	2	3
take_right(α_r, x, y, z)	$x \in N_S \cap S$, $y \in S$, $z \in E \setminus FSE$, $x \neq z$, $\alpha_r \in R_r$, $(x, y, own_r) \in R$, $(y, z, \alpha_r) \in R$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(x, z, \alpha_r)\}$
grant_right(α_r, x, y, z)	$x \in N_S \cap S$, $y \in S$, $z \in E \setminus FSE$, $y \neq z$, $\alpha_r \in R_r$, $(x, y, own_r) \in R$, $(x, z, \alpha_r) \in R$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(y, z, \alpha_r)\}$
own_take(α_r, x, y)	$x \in S$, $y \in E$, $\alpha_r \in R_r$, $(x, y, own_r) \in R$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(x, y, \alpha_r)\}$
create_entity(x, y, z)	$x \in S$, $y \notin E$, $z \in E \setminus S$, $(x, z, write_r) \in R$	$S' = S$, $E' = E \cup \{y\}$, $A' = A$, $F' = F$, $H'(z) = H(z) \cup \{y\}$, $H'(y) = \varnothing$, для $e \in E \setminus \{z\}$ выполняется равенство $H'(e) = H(e)$, $R' = R \cup \{(x, y, own_r)\}$
create_subject(x, y, z)	$x \in S$, $y \in E$, $z \notin E$, $(x, y, execute_r) \in R$	$S' = S \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A$, $F' = F$, $H'(x) = H(x) \cup \{z\}$, $H'(z) = \varnothing$, для $e \in E \setminus \{x\}$ выполняется равенство $H'(e) = H(e)$, $R' = R \cup \{(x, z, own_r)\}$
potential_subject(x, y, z)	$x \in N_S \cap S$, $y \in PS$, $z \notin E$, и для каждой $e \in E$ такой, что $e \in]y[$, существует $(e, x, write_m) \in F$	$S' = S \cup \{z\}$, $FSS' = FSS \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A$, $F' = F$, $H'(x) = H(x) \cup \{z\}$, $H'(z) = \varnothing$, для $e \in E \setminus \{x\}$ выполняется равенство $H'(e) = H(e)$, $R' = R \cup \{(x, z, own_r)\} \cup \{(z, e, \alpha_r)$: $(y, e, \alpha_r) \in R\}$

Продолжение таблицы

1	2	3
$know(x, y)$	$x \in N_S \cap S$, $y \in S$, $x \neq y$, и для каждой $e \in E$ такой, что $e \in]y[$, существует $(e, x, write_m) \in F$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(x, y, own_r)\}$
$control(x, y, z)$	$x \in N_S \cap S$, $y \in S$, $x \neq y$, $z \in E$, $z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(x, y, own_r)\}$
$access\_write(x, y)$	$x \in FSS \cup (N_S \cap S)$, $y \in E$, $(x, y, write_r) \in R$	$S' = S$, $E' = E$, $R' = R$, $H' = H$, $A' = A \cup \{(x, y, write_a)\}$, $F' = F \cup \{(x, y, write_m)\}$
$access\_read(x, y)$	$x \in FSS \cup (N_S \cap S)$, $y \in E$, $(x, y, read_r) \in R$	$S' = S$, $E' = E$, $R' = R$, $H' = H$, $A' = A \cup \{(x, y, read_a)\}$, $F' = F \cup \{(y, x, write_m)\}$
$control(x, y, z)$	$x \in N_S \cap S$, $y \in S$, $x \neq y$, $z \in E$, $z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$	$S' = S$, $E' = E$, $A' = A$, $H' = H$, $F' = F$, $R' = R \cup \{(x, y, own_r)\}$
$find(x, y, z)$	$x, y \in S$, $z \in E$, $x \neq z$, и либо $x = y$, $x \in L_S \cap S$ и $(x, z, write_a) \in A$, либо $x \neq y$ и $\{(x, y, \alpha), (y, z, \beta)\} \subset R \cup A \cup F$, где если $x \in L_S \cap S$, то $\alpha \in \{write_a, write_m\}$, если $x \in N_S \cap S$, то $\alpha \in \{write_r, write_m\}$, если $y \in L_S \cap S$, то $\beta = \{write_a, write_m\}$, если $y \in N_S \cap S$, то $\beta \in \{write_r, write_m\}$	$S' = S$, $E' = E$, $R' = R$, $A' = A$, $H' = H$, $F' = F \cup \{(x, z, write_m)\}$
$post(x, y, z)$	$x, z \in S$, $y \in E$, $x \neq z$, $\{(x, y, \alpha), (z, y, \beta)\} \subset R \cup A \cup F$, где если $x \in L_S \cap S$, то $\alpha \in \{write_a, write_m\}$, если $x \in N_S \cap S$, то $\alpha \in \{write_r, write_m\}$, если $z \in L_S \cap S$, то $\beta = read_a$, если $z \in N_S \cap S$, то $\beta = read_r$	$S' = S$, $E' = E$, $R' = R$, $A' = A$, $H' = H$, $F' = F \cup \{(x, z, write_m)\}$

Окончание таблицы

1	2	3
$pass(x, y, z)$	$y \in S$, $x, z \in E$, $x \neq z$, и либо $y = z$, $y \in L_S \cap S$ и $(y, x, read_a) \in A$, либо $y \neq z$ и $\{(y, x, \beta), (y, z, \alpha)\} \subset R \cup A \cup F$, где если $y \in L_S \cap S$, то $\alpha \in \{write_a, write_m\}$, $\beta = read_a$, если $y \in N_S \cap S$, то $\alpha \in \{write_r, write_m\}$, $\beta = read_r$	$S' = S$, $E' = E$, $R' = R$, $A' = A$, $H' = H$, $F' = F \cup \{(x, z, write_m)\}$

Правило преобразования состояний $potential\_subject(x, y, z)$ позволяет недоверенному субъекту x, реализовавшему к себе информационные потоки по памяти от всех сущностей, параметрически ассоциированных с потенциальным доверенным субъектом y, создать соответствующего y доверенного субъекта z (рис. 2). При этом субъект z получает все права доступа субъекта y, субъект x получает доступ владения own_r к субъекту z.

При анализе условий реализации запрещенных информационных потоков по памяти будем использовать следующие определения.

Определение 5. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — начальное состояние системы $\Sigma(G^*, OP, G_0)$. Запрещенным информационным потоком по памяти является информационный поток от сущности $e \in FSE$, защищенной ФС, к недоверенному субъекту $x \in N_S \cap S_0$ в случае, когда в начальном состоянии G_0 субъект x не имеет прав доступа $read_r$ или own_r к сущности-образу $fs(e) \in E$, соответствующей сущности e.

Определение 6. В рамках ФС ДП-модели будем говорить, что система $\Sigma(G^*, OP, G_0)$ является безопасной в случае, когда невозможен переход системы в состояние, в котором реализуется запрещенный информационный поток по памяти, удовлетворяющий условиям определения 5.

Определение 7. Нарушитель в рамках ФС ДП-модели — любой недоверенный субъект системы.

Примеры разрешенных и запрещенных информационных потоков по памяти приведены на рис. 3.

Анализ траекторий системы без получения недоверенными субъектами прав доступа владения к доверенным субъектам.

Рассмотрим частный случай, когда при реализации запрещенных информационных потоков по памяти недоверенные субъекты не применяют правила вида $control(x, y, z)$, $know(x, y)$ или $potential\_subject(x, y, z)$ для получения прав доступа владения к доверенным субъектам системы. Дадим определение.

Определение 8. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам, если она является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных

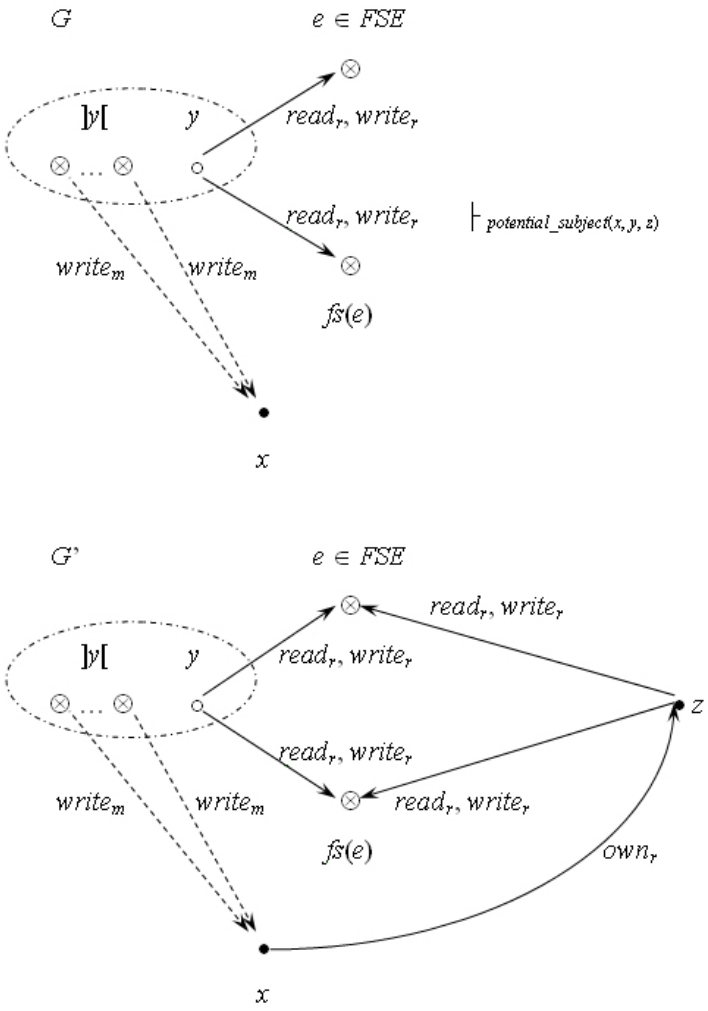

Рис. 2. Пример применения правила *potential_ subject*(x, y, z)

потоков по памяти, и при ее реализации недоверенные субъекты не инициируют выполнение правил вида *control*(x, y, z), *know*(x, y) и *potential_ subject*(x, y, z).

Определение 9. Назовем состояние G системы $\Sigma(G^*, OP)$ безопасным относительно прав доступа, когда в нем недоверенные субъекты не обладают правами доступа к доверенным субъектам.

Определение 10. 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 *simple_ can_ write_ memory*(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 траектория $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(x, y, write_m) \in F_N$.

В рамках ФС ДП-модели с учетом предположений 1–3 воспользуемся определенными и обоснованными в БК ДП-модели и классической модели *Take-Grant* необходимыми и достаточными условиями истинности предиката *can_ share*(α, x, y, G_0). При этом доверенные субъекты с учетом предположения 3 могут рассматриваться как объекты модели *Take-Grant*.

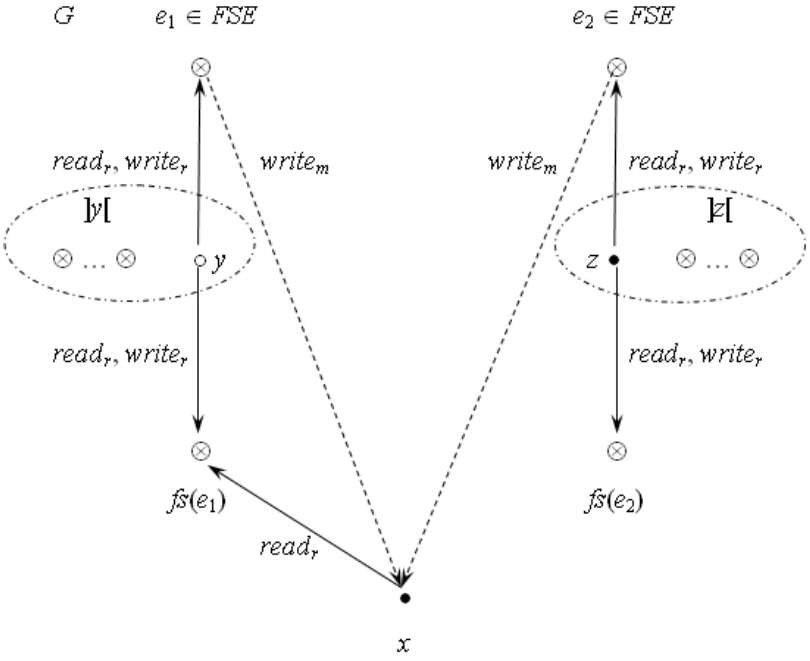

Рис. 3. Примеры информационных потоков по памяти: $(e_1,\ x,\ write_m)$ — разрешенный информационный поток; $(e_2,\ x,\ write_m)$ — запрещенный информационный поток

Определение 11. Пусть $G_0 = (S_0,\ E_0,\ R_0 \cup A_0 \cup F_0,\ H_0)$ — состояние системы $\Sigma(G^*,\ OP)$, в котором существуют субъект $x \in S_0$ и сущность $y \in E_0$, где $x \neq y$, и пусть право доступа $\alpha \in R_r$. Определим предикат $simple\_can\_share(\alpha,\ x,\ y,\ G_0)$, 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 $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам, и $(x,\ y,\ \alpha) \in R_N$.

Определение 12. Мостом в состоянии G между двумя недоверенными субъектами называется путь в графе-состоянии, удовлетворяющий одному из условий:

— субъекты соединены ребром, без учета направления помеченным правом доступа own_r;

— субъекты соединены путем, проходящим через доверенных субъектов, словарная запись которого имеет вид: $\overrightarrow{own_r}^*\ \overleftarrow{own_r}^*$, где символ «*» означает многократное, в том числе нулевое, повторение.

Определение 13. Пролетом моста в состоянии G называется путь с началом в недоверенном субъекте и концом в доверенном субъекте, проходящий через доверенных субъектов, словарная запись которого имеет вид: $\overrightarrow{own_r}^*$, где символ «*» означает многократное повторение.

Определение 14. Два недоверенных субъекта x и y в состоянии G являются own-связанными, когда существует последовательность недоверенных субъектов $s_1,\ \ldots,\ s_n$, где $n \geqslant 2$, таких, что $s_1 = x$, $s_n = y$, и каждая пара $s_i,\ s_{i+1}$ соединена мостом, где $1 \leqslant i < n$.

Утверждение 8. Пусть $G_0 = (S_0,\ E_0,\ R_0 \cup A_0 \cup F_0,\ H_0)$ — состояние системы $\Sigma(G^*,\ OP)$, в котором существуют субъект $x \in S_0$, сущность $y \in E_0$, где $x \neq y$, и пусть право доступа $\alpha \in R_r$. Предикат $simple\_can\_share(\alpha,\ x,\ y,\ G_0)$ является истинным тогда и только тогда, когда выполняются условия.

Условие 1. Существует субъект $s \in S_0$ такой, что или $(s,\ y,\ \alpha) \in R_0$, или $(s,\ y,\ own_r) \in R_0$.

Условие 2. Существуют недоверенные субъекты $x',\ s' \in N_S \cap S_0$ такие, что выполняются условия:

— или $x = x'$, или x' соединен с x пролетом моста;
— или $s = s'$, или s' соединен с s пролетом моста;
— x' и s' являются *own*-связанными.

Доказательство. Справедливость данного утверждения обосновывается в рамках классической модели *Take-Grant*. ■

Определим и обоснуем алгоритмически проверяемые необходимые и достаточные условия истинности предиката $simple\_can\_write\_memory(x,\ y,\ G_0)$.

Теорема 1. 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 $simple\_can\_write\_memory(x,\ y,\ G_0)$ является истинным 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 для каждого i =1, ..., $m-1$ выполняется одно из условий.

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

Условие 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 $simple\_can\_share(write_r,\ e_i,\ e_{i+1},\ G_0)$.

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

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

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

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

Доказательство. Докажем достаточность выполнения условий теоремы для истинности предиката $simple\_can\_write\_memory(x,\ y,\ G_0)$.

Пусть существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geqslant 2$, таких, что выполняются условия теоремы. Выполним доказательство индукцией по длине m последовательности сущностей.

Пусть m= 2. Возможны шесть случаев.

Первый случай: $x \in L_S \cap S_0$ и или $(x,\ y,\ write_m) \in F_0$, или $(x,\ y,\ write_a) \in A_0$. Если $(x,\ y,\ write_m) \in F_0$, то предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен. Если $(x, y, write_a) \in A_0$, то положим $op_1 = find(x,\ x,\ y)$, $G_0 \vdash_{op_1} G_1$. Тогда $(x,\ y,\ write_m) \in F_1$ и предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен.

Второй случай: $x \in FSS_0 \cup (N_S \cap S_0)$ и или $(x,\ y,\ write_m) \in F_0$, или истинен предикат $simple\_can\_share(write_r,\ x,\ y,\ G_0)$. Если $(x,\ y,\ write_m) \in F_0$, то предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен. Если истинен предикат $simple\_can\_share(write_r,\ x,\ y,\ G_0)$, то по определению 11 существуют состояния G_1, ..., G_N и правила преобразования состояний op_1, ..., 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 получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(x,\ y,\ write_r) \in R_N$. Пусть $op_{N+1} = access\_write(x,\ y)$ и $G_N \vdash_{op_{N+1}} G_{N+1}$. Тогда $(x,\ y,\ write_m) \in F_{N+1}$ и предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен.

Третий случай: $y \in L_S \cap S_0$ и $(y,\ x,\ read_a) \in A_0$. Положим $op_1 = pass(x,\ y,\ y)$, $G_0 \vdash_{op_1} G_1$. Тогда $(x,\ y,\ write_m) \in F_1$ и предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен.

Четвертый случай: $y \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(read_r, y, x, G_0)$. Тогда по определению 11 существуют состояния $G_1,\ \ldots,\ G_N$ 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 получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(y, x, read_r) \in R_N$. Пусть $op_{N+1} = access\_read(y, x)$ и $G_N \vdash_{op_{N+1}} G_{N+1}$. Тогда $(x, y, write_m) \in F_{N+1}$ и предикат $simple\_can\_write\_memory(x, y, G_0)$ истинен.

Пятый случай: $x \in N_S \cap S_0$, $y \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(own_r, x, y, G_0)$. По определению 11 существуют состояния $G_1, \ldots, G_N$ 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 получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(x, y, own_r) \in R_N$. Воспользуемся предположением базовой ДП-модели, согласно которому для любого субъекта в любом состоянии системы существует сущность-контейнер, в составе которой он может создать новую сущность. Следовательно, существует сущность-контейнер $e \in E_N$, в составе которой субъект x может создать новую сущность. Пусть $op_{N+1} = create\_entity(x, z, e)$, $op_{N+2} = own\_take(write_r, x, z)$, $op_{N+3} = own\_take(read_r, x, z)$, $op_{N+4} = grant\_right(read_r, x, y, z)$, $op_{N+5} = access\_read(y, z)$, $op_{N+6} = post(x, z, y)$, и $G_N \vdash_{op_{N+1}} G_{N+1} \vdash_{op_{N+2}} \ldots \vdash_{op_{N+6}} G_{N+6}$. Тогда $(x, y, write_m) \in F_{N+6}$ и предикат $simple\_can\_write\_memory(x, y, G_0)$ истинен. На рис. 4 приведена последовательность преобразований состояний, при этом показаны только ребра графов-состояний, которые необходимы для применения правил.

Шестой случай: $y \in N_S \cap S_0$, $x \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(own_r,\ y,\ x,\ G_0)$. Доказательство для шестого случая осуществляется аналогично доказательству для пятого случая.

Докажем индуктивный шаг. Пусть $m > 2$ и утверждение теоремы верно для всех последовательностей сущностей длины $k < m$. Докажем, что утверждение теоремы верно для всех последовательностей сущностей длины m.

Пусть $e_1,\ \ldots,\ e_m \in E_0$ — последовательность сущностей, где $e_1 = x$, $e_m = y$. Возможны два случая: $x \in S_0$ или $x \in E_0 \setminus S_0$.

Первый случай: $x \in S_0$. Положим $e_{m-1} = z$. Тогда по предположению индукции существуют состояния $G_1,\ \ldots,\ G_N$ 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 получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(x,\ z,\ write_m) \in F_N$.

Если $z \in S_0$, то по предположению индукции существуют состояния $G_{N+1}, \ldots, G_{N+K}$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, где $K \geqslant 0$, такие, что $G_N \vdash_{op_{N+1}} G_{N+1} \vdash_{op_{N+2}} \ldots \vdash_{op_{N+K}} G_{N+K}$ является траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(z,\ y,\ write_m) \in F_{N+K}$. Пусть $op_{N+K+1} = find(x,\ z,\ y)$, тогда $G_{N+K} \vdash_{op_{N+K+1}} G_{N+K+1}$ и $(x, y, write_m) \in F_{N+K+1}$, следовательно, предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен.

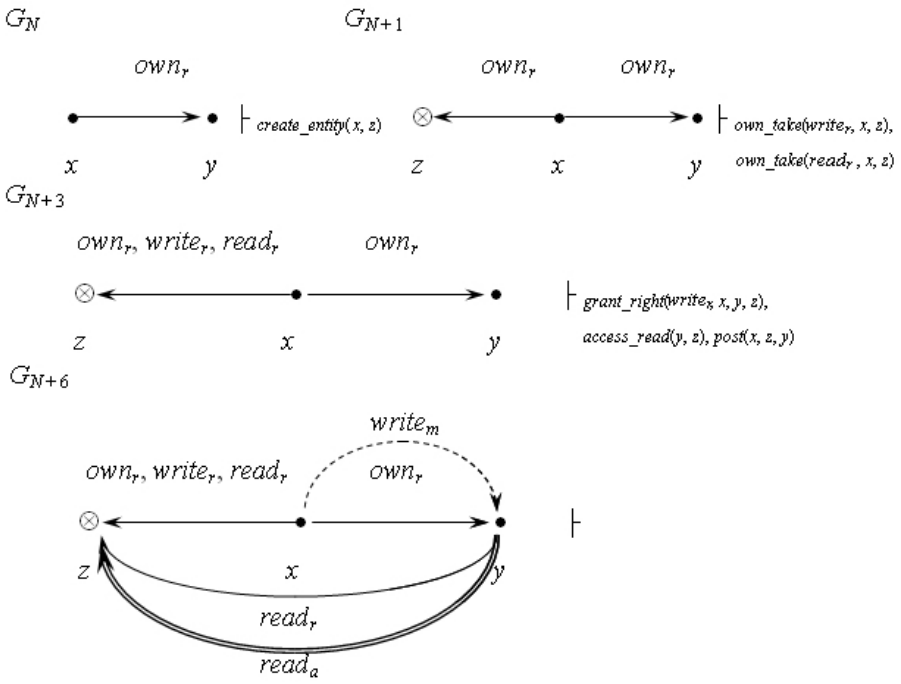

Рис. 4. Случай $x \in N_S \cap S_0$, $y \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $can\_share(own_r, x, y, G_0)$

Если $z \in E_0 \backslash S_0$, то выполняется одно из следующих условий:

— $y \in L_S \cap S_0$ и $(y, z, read_a) \in A_0$;
— $y \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(read_r, y, z, G_0)$.

Если $y \in L_S \cap S_0$ и $(y, z, read_a) \in A_0$, то пусть $M = N$.

Если $y \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(read_r, y, z, G_0)$, то по определению 11 существуют состояния $G_{N+1}, \ldots, G_{N+K}$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, где $K \geqslant 0$, такие, что $G_N \vdash_{op_{N+1}} G_{N+1} \vdash_{op_{N+2}} \ldots \vdash_{op_{N+K}} G_{N+K}$ является траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(y, z, read_r) \in R_{N+K}$. Пусть $op_{N+K+1} = access\_read(y, z)$ и $M = N + K + 1$.

Положим $op_{M+1} = post(x, z, y)$, $G_M \vdash_{op_{M+1}} G_{M+1}$. Тогда $(x, y, write_m) \in F_{M+1}$ и предикат $simple\_can\_write\_memory(x, y, G_0)$ истинен.

Второй случай: $x \in E_0 \backslash S_0$. Положим $e_2 = z$. Тогда по условию теоремы $z \in S_0$ и по предположению индукции существуют состояния $G_1, \ldots, G_N$ 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 получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(z, y, write_m) \in F_N$. При этом выполняется одно из следующих условий:

— $z \in L_S \cap S_0$ и $(z, x, read_a) \in A_0$;
— $z \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(read_r, z, x, G_0)$.

Если $z \in L_S \cap S_0$ и $(z, x, read_a) \in A_0$, то пусть $M = N$.

Если $z \in FSS_0 \cup (N_S \cap S_0)$ и истинен предикат $simple\_can\_share(read_r, z, x, G_0)$, то по определению 11 существуют состояния $G_{N+1}, \ldots, G_{N+K}$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, где $K \geqslant 0$, такие, что $G_N \vdash_{op_{N+1}} G_{N+1} \vdash_{op_{N+2}} \ldots \vdash_{op_{N+K}} G_{N+K}$ является траекторией без получения недоверенными субъектами прав

доступа владения к доверенным субъектам и $(z,\ x,\ read_r) \in R_{N+K}$. Пусть $op_{N+K+1} = access\_read(z,\ x)$ и $M = N + K + 1$.

Положим $op_{M+1} = pass(x,\ z,\ y)$, $G_M \vdash_{op_{M+1}} G_{M+1}$. Тогда $(x,\ y,\ write_m) \in F_{M+1}$ и предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$ истинен.

Индуктивный шаг доказан. Доказательство достаточности выполнения условий теоремы для истинности предиката $simple\_can\_write\_memory(x,\ y,\ G_0)$ выполнено.

Докажем необходимость выполнения условия теоремы для истинности предиката $simple\_can\_write\_memory(x,\ y,\ G_0)$.

Пусть истинен предикат $simple\_can\_write\_memory(x,\ y,\ G_0)$, при этом по определению 10 существуют состояния $G_1,\ \ldots,\ G_N$ и правила преобразования состояний $op_1,\ \ldots,\ op_N$, где $N \geqslant 0$, такие, что траектория $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без получения недоверенными субъектами прав доступа владения к доверенным субъектам и $(x,\ y,\ write_m) \in F_N$. Среди всех этих последовательностей выберем ту, у которой длина N является минимальной. Проведем доказательство индукцией по длине N последовательности преобразований состояний.

Пусть $N = 0$, тогда $(x,\ y,\ write_m) \in F_0$, $m{=}\ 2$ и условие 1 или 2 теоремы выполнено.

Пусть $N > 0$ и утверждение теоремы верно для всех последовательностей преобразований состояний длины $l < N$. Тогда $(x,\ y,\ write_m) \notin F_{N-1}$ и существует правило преобразования состояний op_N такое, что $G_{N-1} \vdash_{op_N} G_N$ и $(x,\ y,\ write_m) \in F_N$.

Из определения правил преобразования состояний следует, что возможны семь случаев:

— $x \in FSS_0 \cup (N_S \cap S_0)$, $(x, y, write_r) \in R_{N-1}$, $op_N = access\_write(x, y)$;
— $y \in FSS_0 \cup (N_S \cap S_0)$, $(y, x, read_r) \in R_{N-1}$ и $op_N = access\_read(y, x)$;
— $x \in L_S \cap S_0$, $(x, y, write_a) \in A_{N-1}$, $op_N = find(x, x, y)$;
— $x \in S_0$ и существует субъект $z \in S_{N-1}$ такой, что $op_N = find(x, z, y)$, $\{(x, z, \alpha), (z, y, \beta)\} \subset R_{N-1} \cup A_{N-1} \cup F_{N-1}$, где если $x \in L_S \cap S_0$, то $\alpha \in \{write_a, write_m\}$, если $x \in N_S \cap S_0$, то $\alpha \in \{write_r, write_m\}$, если $z \in L_S \cap S_{N-1}$, то $\beta = \{write_a, write_m\}$, если $z \in N_S \cap S_{N-1}$, то $\beta \in \{write_r, write_m\}$;
— $x, y \in S_0$ и существует сущность $z \in E_{N-1}$ такая, что $op_N = post(x, z, y)$, $\{(x, z, \alpha), (y, z, \beta)\} \subset R_{N-1} \cup A_{N-1} \cup F_{N-1}$, где если $x \in L_S \cap S_0$, то $\alpha \in \{write_a, write_m\}$, если $x \in N_S \cap S_0$, то $\alpha \in \{write_r, write_m\}$, если $y \in L_S \cap S_0$, то $\beta = read_a$, если $y \in N_S \cap S_0$, то $\beta = read_r$;
— $y \in L_S \cap S_0$, $(y, x, read_a) \in A_{N-1}$, $op_N = pass(x, y, y)$;
— $x, y \in E_0$ и существует субъект $z \in S_{N-1}$, $op_N = pass(x, z, y)$, $\{(z, x, \beta), (z, y, \alpha)\} \subset R_{N-1} \cup A_{N-1} \cup F_{N-1}$, где если $z \in L_S \cap S_{N-1}$, то $\alpha \in \{write_a, write_m\}$, $\beta = read_a$, если $z \in N_S \cap S_{N-1}$, то $\alpha \in \{write_r, write_m\}$, $\beta = read_r$.

В первом случае истинен предикат $simple\_can\_share(write_r,\ x,\ y,\ G_0)$, и условие 2 теоремы выполнено.

Во втором первом случае истинен предикат $simple\_can\_share(read_r,\ y,\ x,\ G_0)$, и условие 4 теоремы выполнено.

Третий случай: $x \in L_S \cap S_0$, $(x,\ y,\ write_a) \in A_{N-1}$, $op_N = find(x,\ x,\ y)$. Предположим $(x,\ y,\ write_a) \notin A_0$, тогда по предположению 3 доверенный субъект $x \in FSS_0$ и существует $0 \leqslant M < N$ такое, что $op_M = access\_write(x,\ y)$. Следовательно, выполняется условие $(x,\ y,\ write_m) \in F_M$, противоречие с минимальностью N. Значит, $(x,\ y,\ write_a) \in A_0$, и условие 1 теоремы выполнено.

В четвертом случае $op_N = find(x,\ z,\ y)$, $\{(x,\ z,\ \alpha),\ (z,\ y,\ \beta)\} \subset R_{N-1} \cup A_{N-1} \cup F_{N-1}$, из минимальности N следует, что $z \in S_0$, и выполняются условия:

— если $x \in L_S \cap S_0$, то $\alpha \in \{write_a, write_m\}$;
— если $x \in N_S \cap S_0$, то $\alpha \in \{write_r, write_m\}$;
— если $z \in L_S \cap S_0$, то $\beta = \{write_a, write_m\}$;
— если $z \in N_S \cap S_0$, то $\beta \in \{write_r, write_m\}$.

Пусть $x \in L_S \cap S_0$. Если $(x, z, write_a) \in A_{N-1}$, то по аналогии с третьим случаем получаем, что $(x, z, write_a) \in A_0$. Если $x \in L_S \cap S_0$ и $(x, z, write_m) \in F_{N-1}$, то истинен предикат $simple\_can\_write\_memory(x, z, G_0)$ с длиной последовательности преобразований состояний меньше N. Следовательно, по предположению индукции существует последовательность сущностей, удовлетворяющая условиям теоремы.

Пусть $x \in N_S \cap S_0$. Если $(x, z, write_r) \in R_{N-1}$, то истинен предикат $simple\_can\_share(write_r, x, z, G_0)$. Если $(x, z, write_m) \in F_{N-1}$, то истинен предикат $simple\_can\_write\_memory(x, z, G_0)$ с длиной последовательности преобразований состояний меньше N. Следовательно, по предположению индукции существует последовательность сущностей, удовлетворяющая условиям теоремы.

Аналогично рассматриваются условия $z \in L_S \cap S_0$ и $z \in N_S \cap S_0$. Таким образом, объединяя последовательности сущностей для каждого из возможных сочетаний пар условий, получаем, что в четвертом случае выполняются условия теоремы.

В пятом случае существует сущность $z \in E_{N-1}$ такая, что $op_N = post(x, z, y)$, $\{(x, z, \alpha), (y, z, \beta)\} \subset R_{N-1} \cup A_{N-1} \cup F_{N-1}$, и выполняются условия:

— если $x \in L_S \cap S_0$, то $\alpha \in \{write_a, write_m\}$;
— если $x \in N_S \cap S_0$, то $\alpha \in \{write_r, write_m\}$;
— если $y \in L_S \cap S_0$, то $\beta = read_a$;
— если $y \in N_S \cap S_0$, то $\beta = read_r$.

Если $z \in E_0$, то шаг индукции обосновывается аналогично четвертому случаю. Если $z \notin E_0$, то существуют $1 \leqslant M < N$, субъект $s \in S_{M-1}$, сущность-контейнер $e \in E_{M-1}$ такие, что $op_M = create\_entity(s, z, e)$. Из минимальности N следует, что выполняются условия $s \in S_0$ и сущность-контейнер $e \in E_0$. Из всех последовательностей преобразований выберем ту, в которой $M = 1$. Рассмотрим состояние G_1, где $G_0 \vdash_{op_1} G_1$, $S_1 = S_0$, $E_1 = E_0 \cup \{z\}$, $R_1 = R_0 \cup \{(s, z, own_r)\}$, $A_1 = A_0$, $F_1 = F_0$. Так как $(x, z, \alpha) \in R_{N-1} \cup A_{N-1} \cup F_{N-1}$, то истинен предикат $simple\_can\_write\_memory(x, z, G_1)$ с длиной последовательности состояний меньше N. Таким образом, получаем, что существуют сущности $e_1, \ldots, e_k$, где $e_1 = x$, $e_k = s$ и $k \geqslant 2$, удовлетворяющие условиям теоремы в состоянии G_1, а следовательно, в состоянии G_0. Аналогично получаем, что существуют сущности $e_k, \ldots, e_{k+m}$, где $e_k = s$, $e_{k+m} = y$ и $m \geqslant 1$, удовлетворяющие условиям теоремы в состоянии G_0. Значит, в пятом случае условия теоремы выполнены.

Седьмой случай рассматривается аналогично третьему случаю.

Восьмой случай рассматривается аналогично четвертому и пятому случаям.

Индуктивный шаг доказан. Обоснована необходимость выполнения условий теоремы для истинности предиката $simple\_can\_write\_memory(x, y, G_0)$. ■

Рассмотрим условия истинности предиката $simple\_can\_write\_memory(e, x, G_0)$ для случая, когда сущность x является недоверенным субъектом и сущность e защищена ФС.

Следствие 1. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — начальное состояние системы $\Sigma(G*, OP, G_0)$, безопасное относительно прав доступа, в котором существуют сущность $e \in FSE_0$, недоверенный субъект $x \in N_S \cap S_0$. Предикат $simple\_can\_$

$\_write\_memory(e,\ x,\ G_0)$ (недоверенный субъект x пытается реализовать запрещенный информационный поток по памяти от сущности e) является истинным тогда и только тогда, когда существует доверенный субъект $s \in FSS_0$ такой, что он обладает правами доступа на чтение и запись к сущностям e и $fs(e)$, и существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = fs(e)$ и $m \geqslant 2$, таких, что для каждого $i = 1, \ldots, m-1$ выполняется одно из условий.

Условие 1. $e_i \in L_S \cap S_0$, $e_{i+1} \in E_0 \setminus N_S$ и или $(e_i,\ e_{i+1},\ write_m) \in F_0$, или $(e_i,\ e_{i+1},\ write_a) \in A_0$.

Условие 2. $e_i \in N_S \cap S_0$, $e_{i+1} \in E_0 \setminus L_S$ и истинен предикат $simple\_can\_share(write_r,\ e_i,\ e_{i+1},\ G_0)$.

Условие 3. $e_{i+1} \in L_S \cap S_0$, $e_i \in E_0 \setminus N_S$ и $(e_{i+1},\ e_i,\ read_a) \in A_0$.

Условие 4. $e_{i+1} \in N_S \cap S_0$, $e_i \in E_0 \setminus L_S$ и истинен предикат $simple\_can\_share(read_r,\ e_{i+1},\ e_i,\ G_0)$.

Условие 5. $e_i, e_{i+1} \in N_S \cap S_0$ и истинен предикат $simple\_can\_share(own_r, e_i, e_{i+1}, G_0)$.

Условие 6. $e_i,\ e_{i+1} \in N_S \cap S_0$ и истинен предикат $simple\_can\_share(own_r, e_{i+1}, e_i, G_0)$.

Доказательство. В соответствии с предположениями 1 и 2 только доверенные субъекты из множества FSS_0 могут обладать правами доступа к сущностям, защищенным ФС. Значит, субъекты, не входящие во множество FSS_0, могут обладать правами доступа, реализовывать информационные потоки или доступы только к сущностям-образам из множества $fs(FSE_0)$. При этом для каждой сущности из множества FSE_0 всегда существует доверенный субъект из множества FSS_0 или потенциальный доверенный субъект из множества PS_0, обладающий правами доступа на чтение и запись к сущности и к соответствующей ей сущности-образу (для каждой $e \in FSE_0$ существует субъект $s \in FSS_0 \cup PS_0$, обладающий правами доступа $(s, e, read_r)$, $(s, e, write_r)$, $(s, fs(e), read_r)$, $(s, fs(e), write_r)$).

В соответствии с предположением 1 в начальном состоянии отсутствуют доступы к недоверенным субъектам и информационные потоки с участием недоверенных субъектов.

Так как начальное состояние безопасно относительно прав доступа, то для любых доверенного субъекта $s_1 \in FSS_0$ и субъекта $s_2 \in FSS_0 \cup (N_S \cap S_0)$ предикаты $simple\_can\_share(write_r, s_1, s_2, G_0)$, $simple\_can\_share(write_r, s_2, s_1, G_0)$, $simple\_can\_share(read_r, s_1, s_2, G_0)$, $simple\_can\_share(read_r, s_2, s_1, G_0)$, $simple\_can\_share(own_r, s_1, s_2, G_0)$ и $simple\_can\_share(own_r, s_2, s_1, G_0)$ являются ложными.

Таким образом, утверждение следствия следует из теоремы 1. ■

В соответствия с утверждением следствия 1, если начальное состояние системы безопасное относительно прав доступа, то необходимым условием истинности предиката $simple\_can\_write\_memory(e,\ x,\ G_0)$, где сущность $e \in FSE_0$, недоверенный субъект $x \in N_S \cap S_0$, является наличие доверенного субъекта $s \in FSS_0$ такого, что он обладает правами доступа на чтение и запись к сущностям e и $fs(e)$.

Следствие 2. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — начальное состояние системы $\Sigma(G*, OP, G_0)$, безопасное относительно прав доступа, в котором отсутствуют доступы или информационные потоки к или от доверенных субъектов, не существует недоверенных субъектов $x \in N_S \cap S_0$ и сущностей $e \in FSE_0$ таких, что выполняются условия или $(x,\ fs(e),\ read_r) \in R_0$, или $(x,\ fs(e),\ own_r) \in R_0$. Пусть также в системе могут реализовываться только траектории без получения недоверенными субъектами прав доступа владения к доверенным субъектам. Тогда система является безопасной.

Доказательство. Утверждение следует из теоремы 1 и следствия 1. ■

Таким образом, в следствии 2 обоснованы достаточные условия безопасности системы для случая, когда в ней могут реализовываться только траектории без получения недоверенными субъектами прав доступа владения к доверенным субъектам.

Анализ траекторий системы с возможным получением недоверенными субъектами прав доступа владения к доверенным субъектам.

Рассмотрим общий случай, когда при реализации запрещенных информационных потоков по памяти недоверенные субъекты могут применять правила вида $control(x, y, z)$, $know(x, y)$ или $potential\_subject(x, y, z)$ для получения прав доступа владения к доверенным субъектам системы. Дадим определение.

Определение 15. 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 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 траектория $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков по памяти и $(x, y, write_m) \in F_N$.

Так как при использовании правил вида $control(x, y, z)$, $know(x, y)$ или $potential\_subject(x, y, z)$ права доступа используются для реализации информационных потоков по памяти, а информационные потоки по памяти могут быть использованы для получения прав доступа, то необходимые условия реализации запрещенных информационных потоков по памяти должны быть определены рекурсивно, что является труднореализуемой задачей. Таким образом, определим и обоснуем достаточные условия реализации запрещенных информационных потоков по памяти. Дадим определения.

Определение 16. 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 траектория $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков по памяти и $(x, y, own_r) \in R_N$.

Определение 17. 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 или потенциальный доверенный субъект $y \in S_0 \cup PS$, где $x \neq y$. Определим предикат $directly\_can\_share\_own(x, y, G_0)$, который будет истинным тогда и только тогда, когда существует последовательность субъектов $s_1, \ldots, s_m \in S_0 \cup PS$, где $s_1 = x$, $s_m = y$ и $m \geqslant 2$, таких, что для каждого $i = 1, \ldots, m-1$ справедливо неравенство $s_i \neq s_{i+1}$ и выполняется одно из условий:

- $s_{i+1} \notin FSS_0$ и $s_i \in [s_{i+1}]$ (каждый доверенный субъект из множества FSS_0 по предположению 2 является функционально корректным);
- истинен предикат $simple\_can\_share(own_r, s_i, s_{i+1}, G_0)$;
- $s_{i+1} \notin FSS_0$ и существует сущность $e \in [s_{i+1}]$ такая, что истинен предикат $simple\_can\_write\_memory(s_i, e, G_0)$ (по предположению 2 невозможно получение

к доверенному субъекту из множества FSS_0 права доступа владения с использованием реализованного к нему информационного потока по памяти);

— для каждой сущности $e \in]s_{i+1}[$ истинен предикат $simple\_can\_write\_memory(s_i, e, G_0)$.

Определим и обоснуем достаточные условия истинности предиката $can\_share\_own(x, y, G_0)$.

Утверждение 9. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$, пусть также существуют недоверенный субъект $x \in N_S \cap S_0$, субъект или потенциальный доверенный субъект $y \in S_0 \cup PS$, где $x \neq y$. Предикат $can\_share\_own(x, y, G_0)$ является истинным в случае, когда существует последовательность субъектов $s_1, \ldots, s_m \in S_0 \cup PS$, где $s_1 = x$, $s_m = y$ и $m \geqslant 2$, таких, что выполняется одно из условий.

Условие 1. $m = 2$ и истинен предикат $directly\_can\_share\_own(x, y, G_0)$.

Условие 2. $m > 2$ и для каждого i =1, ..., $m - 2$ выполняется одно из условий:

— $s_i, s_{i+1} \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(s_i, s_{i+1}, G_0)$, $directly\_can\_share\_own(s_{i+1}, s_{i+2}, G_0)$;

— $i < m - 2$,$s_i, s_{i+2} \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(s_i, s_{i+1}, G_0)$, $directly\_can\_share\_own(s_{i+2}, s_{i+1}, G_0)$;

— $i < m - 2$, $s_{i+1}, s_{i+2} \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(s_{i+1}, s_i, G_0)$, $directly\_can\_share\_own(s_{i+2}, s_{i+1}, G_0)$;

— $s_{i+1} \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(s_{i+1}, s_i, G_0, L_S)$, $directly\_can\_share\_own(s_{i+1}, s_{i+2}, G_0, L_S)$.

Доказательство. Доказательство теоремы выполняется аналогично доказательству в рамках ФПАС ДП-модели достаточности условий истинности предиката $can\_share\_own(x, y, G_0)$. ■

Определим и обоснуем достаточные условия истинности предиката $can\_write\_memory(x, y, G_0)$.

Теорема 2. 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, ..., $m - 1$ выполняется одно из условий.

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

Условие 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 $can\_share(write_r, e_i, e_{i+1}, G_0)$.

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

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

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

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

Доказательство. Доказательство осуществляется аналогично обоснованию достаточности условий теоремы 1 для истинности предиката $simple\_can\_write\_memory(x, y, G_0)$. ■

Рассмотрим условия истинности предиката $can\_write\_memory(e, x, G_0)$ для случая, когда сущность x является недоверенным субъектом и сущность e защищена ФС.

Следствие 3. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущность $e \in FSE_0$, недоверенный субъект $x \in N_S \cap S_0$, и выполняются условия сущность $fs(e) \notin FSE_0$, и она не является параметрически ассоциированной ни с одним субъектом. Пусть также существует доверенный субъект $z \in FSS_0$ такой, что он обладает правами доступа на чтение и запись к сущностям e и $fs(e)$, и существует доверенный субъект $z' \in (L_S \cap S_0) \setminus FSS_0$ такой, что истинен предикат $can\_share\_own(x, z', G_0)$. Тогда предикат $can\_write\_memory(e, x, G_0)$ является истинным.

Доказательство. Так как выполняется условие $fs(e) \notin FSE_0 \cup FSS_0 \cup \{e \in E_0$: существует $s \in S_0$ и $e \in]s[\}$, то по предположению 2 справедливо условие $(z', fs(e), read_r) \in R_0$. Следовательно, истинен предикат $can\_share(read_r, x, fs(e), G_0)$. Кроме того, по предположению 2 являются истинными предикаты $can\_share(read_r, z, e, G_0)$ и $can\_share(write_r, z, fs(e), G_0)$. Таким образом, выполнены условия теоремы 2, и истинен предикат $can\_write\_memory(e, x, G_0)$. Следствие доказано. ■

Пример выполнения условия следствия 3 приведен на рис. 5.

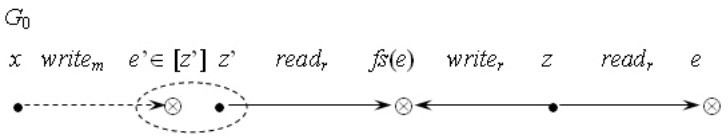

Рис. 5. Пример выполнения условия следствия 3, где $z \in FSS_0$, $e \in FSE_0$

Следствие 4. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущность $e \in FSE_0$ и недоверенный субъект $x \in N_S \cap S_0$. Пусть также существует доверенный субъект или потенциальный доверенный субъект $y \in FSS_0 \cup PS$ такой, что он обладает правами доступа на чтение и запись к сущностям e и $fs(e)$, и выполняется условие: для каждой сущности $e' \in]y[$ истинен предикат $simple\_can\_write\_memory(x, e', G_0)$. Тогда предикат $can\_write\_memory(e, x, G_0)$ является истинным.

Доказательство. По определению 17 истинен предикат $directly\_can\_share\_own(x, y, G_0)$. Следовательно, по утверждению 2 истинен предикат $can\_share\_own(x, y, G_0)$. Таким образом, выполнены условия теоремы 2 и истинен предикат $can\_write\_memory(e, x, G_0)$. Следствие доказано. ■

Рассмотрим достаточные условия, при выполнении которых является ложным предикат $can\_write\_memory(e, x, G_0)$, где сущность e защищена ФС, и субъект x является недоверенным.

Утверждение 10. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние системы $\Sigma(G*, OP)$, в котором существуют сущность $e \in FSE_0$ и недоверенный субъект $x \in N_S \cap S_0$. Пусть также выполнены условия.

Условие 1. Не существует доверенных субъектов таких, что они обладает правами доступа на чтение и запись к сущностям e и $fs(e)$.

Условие 2. Для каждого потенциального доверенного субъекта $y \in PS$ такого, что он обладает правами доступа на чтение и запись к сущностям e и $fs(e)$, и для любого

субъекта $x' \in S_0$ существует сущность $e_y \in]y[$ такая, что выполняются условия $(e_y, x', write_m) \notin F_0$, $(x', e_y, write_a) \notin A_0$, $(x', e_y, write_r) \notin R_0$, $(x', e_y, own_r) \notin R_0$.

Тогда предикат $can\_write\_memory(e, x, G_0)$ является ложным.

Доказательство. Так как по условию 1 не существует доверенных субъектов таких, что они обладает правами доступа на чтение и запись к сущностям e и $fs(e)$, то по предположениям 1 и 2 для реализации информационного потока от сущности e необходимо создание такого доверенного субъекта $y' \in FSS$ из потенциального доверенного субъекта $y \in PS$ с использованием правила $potential\_subject(x', y, y')$, где $x' \in N_S \cap S_0$. По условию 2 не существует недоверенного субъекта $x' \in N_S \cap S_0$, удовлетворяющего условиям применения правила $potential\_subject(x', y, y')$. Следовательно, предикат $can\_write\_memory(e, x, G_0)$ является ложным. Утверждение доказано. ■

Таким образом, описаны и обоснованы достаточные условия, при выполнении которых в системе невозможна реализация запрещенного информационного потока по памяти от сущности, защищенной ФС.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006. 176 с.