МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/1/11

УДК 004.94

БАЗОВАЯ РОЛЕВАЯ ДП-МОДЕЛЬ

П.Н. Девянин

Институт криптографии, связи и информатики Академии ФСБ России, г. Москва

E-mail: peter_devyanin@hotmail.com

На основе ролевых моделей (*RBAC*) и ДП-моделей компьютерных систем с дискреционным или мандатным управлением доступом строится базовая ролевая ДП-модель. С учетом фактических ролей, прав доступа и возможных действий недоверенных сессий анализируются правила преобразования состояний системы. Обосновываются условия передачи прав доступа ролей сессиями пользователей.

Ключевые слова: *компьютерная безопасность*, *математические модели безопасности*, *ролевая модель.*

Ролевое управление доступом (РУД) [1 – 3] является современным, эффективным механизмом защиты компьютерных систем (КС). С использованием иерархии ролей возможно обеспечение управления доступом, точно соответствующего должностным обязанностям пользователей КС. При этом используемые в РУД механизмы статических и динамических ограничений позволяют реализовать на основе РУД дискреционное или мандатное управление доступом.

Модифицируем определения семейства ролевых моделей *RBAC* [1 – 3], базовой ДП-модели, БК ДП-модели, ФАС ДП-модели и мандатной ДП-модели [4] для обеспечения возможности анализировать условия реализации информационных потоков по памяти и по времени в КС с РУД. Модифицированную ДП-модель будем называть базовой ролевой ДП-моделью (или, сокращенно, БР ДП-моделью). При этом используем следующие обозначения:

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

U – множество пользователей, при этом пользователи по определению не являются сущностями;

L_U – множество доверенных пользователей;

N_U – множество недоверенных пользователей, при этом выполняются равенства: $L_U \cup N_U = U$, $L_U \cap N_U = \varnothing$;

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

L_S – множество доверенных субъект-сессий;

N_S – множество недоверенных субъект-сессий, при этом выполняется равенство $L_S \cap N_S = \varnothing$;

R – множество ролей;

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

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

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

$R_f = \{write_m, write_t\}$ – множество видов информационных потоков;

$R_{raf} = R_r \cup R_a \cup R_f$ – множество видов прав доступа, видов доступа и видов информационных потоков, при этом множества R_r, R_a, R_f попарно не пересекаются;

$P \subset E \times R_r$ – множество прав доступа к сущностям;

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, задающая для каждого субъект-сессии пользователя, от имени которого она активизирована;

roles: $S \to 2^R \cup 2^{AR}$ – функция текущих ролей субъект-сессий, задающая для пользователя роли, на которые авторизован активизированный от его имени данный субъект-сессия, при этом в каждом состоянии системы для каждого субъект-сессии $s \in S$ выполняется включение $roles(s) \subseteq UA(user(s)) \cup AUA(user(s))$;

can_manage_rights: $AR \to 2^R$ – функция администрирования прав доступа ролей, определяющая для каждой административной роли множество ролей, для которых разрешено включать или удалять права доступа во множества их прав доступа с использованием данной административной роли.

Определение 1. Иерархией сущностей называется заданное на множестве сущностей E отношение частичного порядка «≤», удовлетворяющее условию: если для сущности $e \in E$ имеются сущности $e_1, e_2 \in E$, такие, что $e \leq e_2$, $e \leq e_1$, то $e_1 \leq e_2$ или $e_2 \leq e_1$. В случае, когда для двух сущностей $e_1, e_2 \in E$ выполняются условия $e_1 \leq e_2$ 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: $E \to 2^E$ – функцию иерархии сущностей, сопоставляющую каждой сущности $c \in E$ множество сущностей $H_E(c) \subset E$ и удовлетворяющую следующим условиям.

Условие 1. Если сущность $e \in H_E(c)$, то $e < c$ и не найдется 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 $H_E(e_1) \cap H_E(e_2) = \varnothing$ и условия:

- если $o \in O$, то выполняется равенство $H_E(o) = \varnothing$;
- 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 $H_E(e) \subset E \setminus S$;
- если $s \in S$, то $H_E(s) \subset S$.

Определение 3. Иерархией ролей в БР ДП-модели называется заданное на множестве ролей R отношение частичного порядка «≤». При этом по определению выполняется условие: для пользователя $u \in U$, если роли $r, r' \in R$, такие, что $r \in UA(u)$ и $r' \leq r$, то выполняются условия $r' \in UA(u)$. В случае, когда для двух ролей $r_1, r_2 \in R$ выполняются условия $r_1 \leq r_2$ и $r_1 \neq r_2$, будем использовать обозначение $r_1 < r_2$.

Определение 4. Определим H_R: $R \to 2^R$ – функцию иерархии ролей, сопоставляющую каждой роли $r \in R$ множество ролей $H_R(r) \subset R$ и удовлетворяющую условию: если роль $r' \in H_R(r)$, то $r' < r$ и не существует роли $r'' \in R$, такой, что $r' < r''$, $r'' < r$.

Определение 5. Иерархией административных ролей в БР ДП-модели называется заданное на множестве ролей AR отношение частичного порядка «≤». При этом по определению выполняется условие: для пользователя $u \in U$, если административные роли $r, r' \in AR$, такие, что $r \in AUA(u)$ и $r' \leq r$, то выполняются условия $r' \in AUA(u)$. В случае, когда для двух ролей $r_1, r_2 \in AR$ выполняются условия $r_1 \leq r_2$ и $r_1 \neq r_2$, будем использовать обозначение $r_1 < r_2$.

Определение 6. Определим H_{AR}: $AR \to 2^{AR}$ – функцию иерархии административных ролей, сопоставляющую каждой роли $r \in AR$ множество ролей $H_{AR}(r) \subset AR$ и удовлетворяющую условию: если роль $r' \in H_{AR}(r)$, то $r' < r$ и не существует роли $r'' \in AR$, такой, что $r' < r''$, $r'' < r$.

Определение 7. Пусть определены: множества пользователей U, сущностей E, субъект-сессий S, прав доступа к сущностям P, доверенных пользователей L_U, доверенных субъект-сессий L_S, множество доступов субъект-сессий к сущностям $A \subseteq S \times E \times R_a$, множество информационных потоков $F \subseteq E \times E \times R_f$, функции авторизованных ролей пользователей UA, авторизованных административных ролей пользователей AUA, прав доступа ролей PA, принадлежности субъект-сессии пользователю *user*, текущих ролей субъект-сессии *roles*, иерархии ролей H_R, иерархии административных ролей H_{AR}, иерархии сущностей H_E. Определим $G = (UA, AUA, PA, user, roles, A, F, H_R, H_{AR}, H_E, L_U, L_S)$ – состояние системы.

Используем обозначения:

$\Sigma(G^*, OP)$ – система, при этом: G^* – множество всех возможных состояний, OP – множество правил преобразования состояний G Ã $_{op}$ 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.

БР ДП-модель предназначена для анализа условий реализации в КС с РУД информационных потоков, и в ее рамках не предполагается исследовать вопросы администрирования множества ролей, иерархии ролей, иерархии административных ролей, множеств авторизованных ролей пользователей, параметров ограничений. Таким образом, будем использовать следующее предположение.

Предположение 1. В рамках БР ДП-модели на траекториях функционирования системы не изменяются: значения множеств U, L_U, R, функции UA, AUA, H_R, H_{AR}. В БР ДП-модели не используются динамические ограничения.

В БР ДП-модели пользователи или субъект-сессии могут быть доверенными или недоверенными. При этом в отличие от доверенных субъектов систем с дискреционным управлением доступом доверенные пользователи или субъект-сессии могут не обладать ролями, включающими все права доступа ко всем сущностям системы. Чтобы не усложнять описание правил преобразования состояний системы в рамках БР ДП-модели, целесообразно использовать следующие предположение и определение.

Предположение 2. Каждый пользователь или субъект-сессия системы $\Sigma(G^*, OP)$ вне зависимости от имеющихся у нее авторизованных ролей является либо доверенной, либо недоверенной. Доверенные поль-

зователи или субъект-сессии не создают новых субъект-сессий. Каждый недоверенный пользователь или субъект-сессия могут создать только недоверенную субъект-сессию.

Определение 8. Доверенную субъект-сессию назовем корректной относительно информационных потоков по времени, если она не участвует в их реализации.

Используем обозначения:

$LF_S \subset L_S$ – множество доверенных субъект-сессий корректных относительно информационных потоков по времени, $NF_S \subset L_S$ – множество доверенных субъект-сессий некорректных относительно информационных потоков по времени, при этом по определению выполняются равенства $LF_S \cap NF_S = \varnothing$, $LF_S \cup NF_S = L_S$.

В рамках предположений 1 и 2 будем использовать следующее сокращенное обозначение для состояния системы $G = (PA, user, roles, A, F, H_E)$.

Предположение 3. Только информационный поток по памяти к сущности, функционально ассоциированной с субъект-сессией, приводит к изменению вида преобразования данных, реализуемого этой субъект-сессией. Функционально ассоциированными с субъект-сессией являются сущности, от которых зависит вид преобразования данных, реализуемого субъект-сессией в данном или некотором последующем состоянии системы $\Sigma(G^*, OP)$. Множество сущностей, функционально ассоциированных с субъект-сессией, не изменяется в процессе функционирования системы.

В существующих КС при создании субъект-сессии множество функционально-ассоциированных с ней сущностей может задаваться в зависимости от многих параметров (например, в зависимости от пользователя, создающего субъект-сессию, от сущности, из которой создается субъект-сессия, или от компьютера, на котором создается субъект-сессия). Для упрощения описания правил преобразования состояний в дальнейшем будем использовать следующее предположение.

Предположение 4. При создании новой субъект-сессии s множество функционально ассоциированных с ней сущностей задается только в зависимости от сущности, из которой создается субъект-сессия s, и пользователя, который либо создает субъект-сессию s, либо от имени которого другая субъект-сессия создает субъект-сессию s.

Используем следующие обозначения:

$[s] \subset E \cup U$ – множество сущностей, функционально ассоциированных с субъект-сессией s (при этом по определению выполняется условие $s \in [s]$), и пользователей, каждый из которых может создать субъект-сессию, являющуюся функционально ассоциированной сущностью с субъект-сессией s.

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.

Определение 9. Доверенную субъект-сессию y назовем функционально корректной, если во множество функционально ассоциированных с ней сущностей $[y]$ не входят недоверенные субъект-сессии.

Определение 10. Доверенную субъект-сессию y назовем корректной относительно доверенной субъект-сессии y' и сущности e (не являющейся доверенной субъект-сессией), если субъект-сессия y не реализует информационный поток по памяти от сущности e к сущности e', функционально ассоциированной с доверенной субъект-сессией y'.

Используем обозначение:

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

Предположение 5. Субъект-сессии могут иметь друг к другу только доступ владения own_a. Роли могут обладать к субъект-сессиям только правом доступа владения own_r. Если субъект-сессия s_1 реализовала информационный поток по памяти от себя к сущности, функционально ассоциированной с другой субъект-сессией s_2, то субъект-сессия s_1 получает: доступ владения own_a к субъект-сессии s_2, возможность использовать роли из множества ролей $roles(s_2)$ (при этом субъект-сессия s_1 не может изменить множество текущих ролей $roles(s_2)$), возможность получать доступ владения own_a к субъект-сессиям, доступом владения к которым обладает субъект-сессия s_2, возможность использовать административные роли субъект-сессии s_2 для осуществления действий над ролями и сущностями, которые позволяют ей выполнять права доступа ролей субъект-сессии s_2.

Используем 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: $de\_facto\_roles(s_1) = roles(s_1) \cup \{r \in R \cup AR$: существует $s_2 \in S$, $(s_1, s_2, own_a) \in A$ и $r \in roles(s_2)\}$.

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: $de\_facto\_rights(s) = \{p \in P$: существует $r \in de\_facto\_roles(s)$ и $p \in PA(r)\}$.

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

В рамках БР ДП-модели используются следующие 19 правил преобразования состояний: *take_role*(x, r), *remove_role*(x, r), *grant_right*$(x, r, (y, \alpha_r))$, *remove_right*$(x, r, (y, \alpha_r))$, *create_entity*(x, r, y, z), *create_first_session*(u, r, y, z), *create_session*(x, r, y, z), *delete_entity*(x, 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). Дадим определение.

Определение 11. Монотонное правило преобразования состояний – правило преобразования состояний из множества *OP*, применение которого не приводит к удалению из состояний: ролей из множества текущих ролей субъект-сессии, прав доступа ролей к сущностям, субъект-сессий или сущностей, доступов субъект-сессий к сущностям, информационных потоков.

По определению 11 немонотонными правилами преобразования состояний будут являться: *remove_role*(x, r), *remove_right*$(x, r, (y, \alpha_r))$, *delete_entity*(x, y, z). Возможно доказать утверждение 1.

Утверждение 1. Пусть $G_0 = (PA_0, user_0, roles_0, A_0, F_0, H_{E0})$ – начальное состояние системы $\Sigma(G^*, OP, G_0)$. Пусть также существуют состояния системы $G_1, \ldots, G_N = (PA_N, user_N, roles_N, A_N, F_N, H_{EN})$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0\ \tilde{A}_{op1}\ G_1\ \tilde{A}_{op2} \ldots \tilde{A}_{opN}\ G_N$, где $N \geq 0$. Тогда существуют состояния $G'_1, \ldots, G'_M = (PA'_M, user'_M, roles'_M, A'_M, F'_M, H'_{EM})$, где $M \geq 0$, и монотонные правила преобразования состояний $op'_1, \ldots, op'_M$ такие, что $G_0\ \tilde{A}_{op'1}\ G'_1\ \tilde{A}_{op'2} \ldots \tilde{A}_{op'M}\ G'_M$ и выполняются следующие условия.

Условие 1. Верно включение $S_N \subset S'_M$, и для каждого субъект-сессии $s \in S_N$ выполняются условия: $user_N(s) = user'_M(s)$, $roles_N(s) \subset roles'_M(s)$.

Условие 2. Верно включение $E_N \subset E'_M$, и для каждой сущности $e \in E_N$ выполняется условие $H_{EN}(e) \subset H'_{EM}(e)$.

Условие 3. Для каждой роли $r \in R$ выполняется условие $PA_N(r) \subset PA'_M(r)$.

Условие 4. Верно включение $A_N \subset A'_M$.

Условие 5. Верно включение $F_N \subset F'_M$.

Таким образом, в рамках БР ДП-модели при анализе условий передачи прав доступа, реализации информационных потоков по памяти или по времени возможно использование только монотонных правил преобразования состояний. При исследовании в статье условий передачи прав доступа ролей кооперирующими субъект-сессиями пользователей не применяются следующие монотонные правила: *create_entity*(x, r, y, z), *create_session*(x, r, y, z), *rename_entity*(x, y, z), *access_read*(x, y), *flow*(x, y, y', z), *find*(x, y, z), *pass*(x, y, z). Данные правила преобразования состояний введены в БР ДП-модель для анализа условий реализации информационных потоков по времени. Следовательно, приведем в таблице условия и результаты применения только правил преобразования состояний, которые необходимы для анализа условий передачи прав доступа ролей.

Монотонные правила преобразования состояний, используемые для передачи прав доступа ролей

Правило	Исходное состояние $G = (PA, user, roles, A, F, H_E)$	Результирующее состояние $G' = (PA', user', roles', A', F', H_E')$
1	2	3
take_role(x, r)	$x \in S$, $r \in UA(user(x)) \cup AUA(user(x))$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $A' = A$, $F' = F$, $H_E' = H_E$, $roles'(x) = roles(x) \cup \{r\}$ и для $x' \in S \setminus \{x\}$ выполняется равенство $roles'(x') = roles(x')$
grant_right$(x, r, (y, \alpha_r))$	$x \in S$, $y \in E$, $(y, \alpha_r) \in P$ и $((y, own_r), r) \in de\_facto\_actions(x)$	$S' = S$, $E' = E$, $user' = user$, $roles' = roles$, $A' = A$, $H_E' = H_E$, $PA'(r) = PA(r) \cup \{(y, \alpha_r)\}$, и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S$, $x \neq s$ и $r \in de\_facto\_roles(s)\}$, если $x \in LF_S \cap S$, то $F' = F$
create_first_session (u, r, y, z)	$u \in N_U$, $y \in E$, $z \notin E$, $(y, execute_r) \in PA(UA(u))$ и $r \in can\_manage\_rights(AUA(u))$	$S' = S \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A$, $user'(z) = u$, для $s \in S$ выполняется равенство $user'(s) = user(s)$, $F' = F$, $roles'(z) = \varnothing$, для $s \in S$ выполняется равенство $roles'(s) = roles(s)$, $[z] = fa(u, y)$, $H_E'(z) = \varnothing$, для $e \in E$ выполняется равенство $H_E'(e) = H_E(e)$, $PA'(r) = PA(r) \cup \{(z, own_r)\}$, для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$
control(x, y, z)	$x, y \in S$, $x \neq y$, $z \in E$, $z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H_E' = H_E$, $A' = A \cup \{(x, y, own_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leq e\}$, если $x \in LF_S \cap S$, то $F' = F$

Продолжение таблицы

1	2	3
access_own(*x*, *y*)	$x, y \in S, x \neq y,$ $(y, own_r) \in de\_facto\_rights(x)$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, H_E' = H_E,$ $A' = A \cup \{(x, y, own_a)\},$ если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$, если $x \in LF_S \cap S$, то $F' = F$
take_access_own(*x*, *y*, *z*)	$x, y, z \in S, x \neq z,$ $\{(x, y, own_a), (y, z, own_a)\} \subset A$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, H_E' = H_E,$ $A' = A \cup \{(x, z, own_a)\},$ если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E,$ $x \neq e$ и $z \leq e\}$, если $x \in LF_S \cap S$, то $F' = F$
access_write(*x*, *y*)	$x \in S, (y, write_r) \in$ $de\_facto\_rights(x)$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, H_E' = H_E,$ $A' = A \cup \{(x, y, write_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup$ $\{(x, y, write_m)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$, если $x \in LF_S \cap S$, то $F' = F \cup \{(x, y, write_m)\}$
access_append(*x*, *y*)	$x \in S, (y, append_r) \in$ $de\_facto\_rights(x)$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, H_E' = H_E,$ $A' = A \cup \{(x, y, append_a)\},$ если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, y, write_m)\} \cup \{(x, e,$ $write_t): e \in E, x \neq e$ и $y \leq e\},$ если $x \in LF_S \cap S$, то $F' = F \cup \{(x, y, write_m)\}$
post(*x*, *y*, *z*)	$x, z \in S, y \in E, x \neq z,$ $(y, read_r) \in de\_facto\_rights(z)$ и или $(y, \alpha) \in de\_facto\_rights(x),$ где $\alpha \in \{write_r, append_r\},$ или $(x, y, \alpha) \in F,$ где $\alpha \in \{write_m, write_t\}$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, A' = A, H_E' = H_E,$ если $\alpha \neq write_t$, то $F' = F \cup \{(x, z, write_m)\},$ если $\alpha = write_t$ и $x, z \in (N_S \cup NF_S) \cap S,$ то $F' = F \cup \{(x, z,$ $write_t)\},$ если $\alpha = write_t$ и $\{x, z\} \cap (LF_S \cap S) \neq \varnothing$, то $F' = F$

Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели показана на рис. 1.

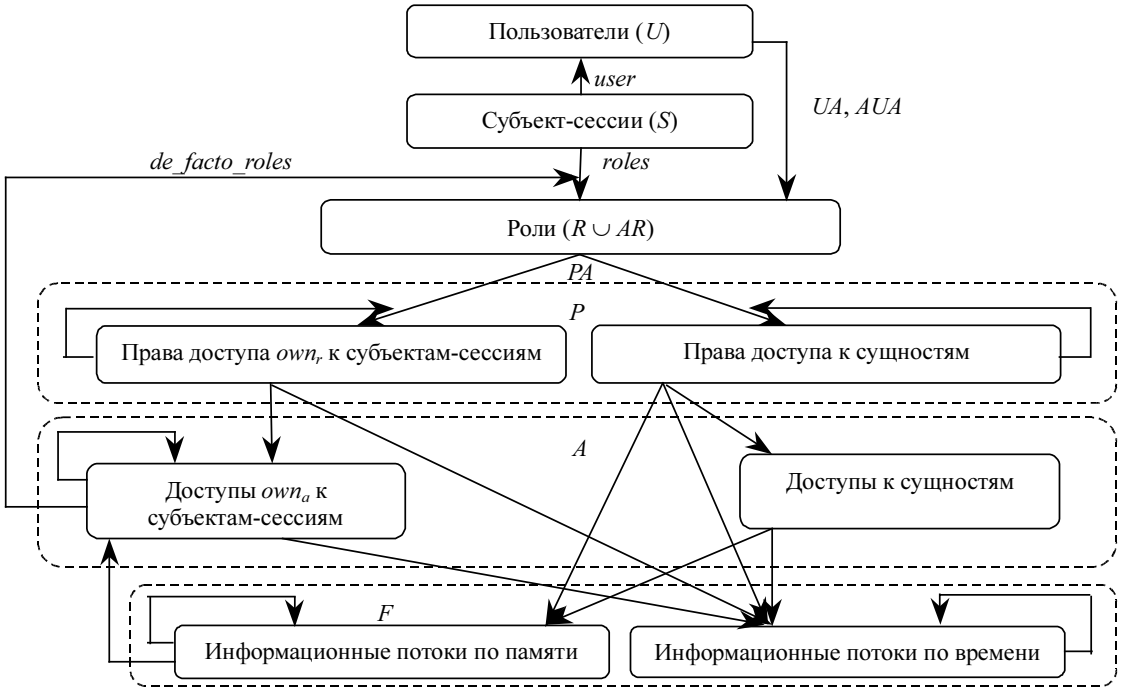

Рис. 1. Зависимость условий и результатов применения правил преобразования состояний

Проанализируем случай, когда для передачи права доступа ролей непосредственно взаимодействуют только две субъект-сессии.

Определение 12. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, если при ее реализации используются монотонные правила преобразования состояний, и доверенные субъект-сессии: не берут роли во множество текущих ролей, не дают другим ролям права доступа к сущностям, не получают доступ владения к субъект-сессиям.

Определение 13. Пусть $G_0 = (PA_0, user_0, roles_0, A_0, F_0, H_{E0})$ – состояние системы $\Sigma(G^*, OP)$, в котором существуют пользователь $u \in U$ и право доступа к сущности $(e, \alpha) \in P_0$. Определим предикат $can\_share((e, \alpha), u, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (PA_N, user_N, roles_N, A_N, F_N, H_{EN})$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \tilde{A}_{op1} G_1 \tilde{A}_{op2} \ldots \tilde{A}_{opN} G_N$ является траекторией без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, и существует субъект-сессия $x \in S_N$, такая, что $user_N(x) = u$, и право доступа к сущности $(e, \alpha) \in de\_facto\_rights_N(x)$, где $N \geq 0$.

Для упрощения записи алгоритмически проверяемых необходимых и достаточных условий истинности предиката $can\_share((y, \alpha), u, G_0)$ используем следующие определения.

Определение 14. Пусть $G = (PA, user, roles, A, F, H_E)$ – состояние системы $\Sigma(G^*, OP)$, в котором существуют недоверенный пользователь или недоверенная субъект-сессия $x \in N_U \cup (N_S \cap S)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S$. Определим предикат $directly\_access\_own(x, y, G)$, который будет истинным тогда и только тогда, когда выполняется одно из следующих условий 1 – 4.

Условие 1. Если $y \in N_U$ и $x \in N_U$, то существуют сущность $e_y \in E$ и роль $r_y \in R$, такие, что $(e_y, execute_r) \in PA(UA(y))$, $r_y \in can\_manage\_rights(AUA(y))$, и выполняется одно из условий: $(r_y \in UA(x))$, или $(x \in fa(y, e_y))$, или (существует сущность $e \in E$, такая, что $(e, \beta) \in PA(UA(x))$, где $\beta \in \{write_r, append_r, own_r\}$, и или $e \in fa(y, e_y)$, или $(e, \gamma) \in PA(UA(y))$, где $\gamma \in \{read_r, own_r\}$).

Условие 2. Если $y \in N_U$ и $x \in N_S \cap S$, то существуют сущность $e_y \in E$ и роль $r_y \in R$, такие, что $(e_y, execute_r) \in PA(UA(y))$, $r_y \in can\_manage\_rights(AUA(y))$, и выполняется одно из условий: или $(r_y \in UA(user(x)))$, или $(x \in fa(y, e_y))$, или (существует сущность $e \in E$, такая, что $(e, \beta) \in PA(UA(user(x))$, где $\beta \in \{write_r, append_r, own_r\}$, или $(x, e, write_m) \in F$, и или $e \in fa(y, e_y)$, или $(e, \gamma) \in PA(UA(y))$, где $\gamma \in \{read_r, own_r\}$).

Условие 3. Если $y \in S$ и $x \in N_U$, то выполняется одно из условий: или $((y, own_r) \in PA(UA(x)))$, или $(x \in [y])$, или (существует сущность $e \in E$, такая, что $(e, \beta) \in PA(UA(x))$, где $\beta \in \{write_r, append_r, own_r\}$, и или $e \in [y]$, или $y \in N_S \cap S$, $(e, \gamma) \in PA(UA(user(y)))$, где $\gamma \in \{read_r, own_r\}$, или $y \in L_S \cap S$, $(e, read_r) \in PA(roles(y))$, $(y, e) \notin y(E)$).

Условие 4. Если $y \in S$ и $x \in N_S \cap S$, то выполняется одно из условий: или $((y, own_r) \in PA(UA(user(x))))$, или $(x \in [y])$, или $((x, y, own_a) \in A)$, или (существует сущность $e \in E$, такая, что $(e, \beta) \in PA(UA(user(x))$, где $\beta \in \{write_r, append_r, own_r\}$, или $(x, e, write_m) \in F$, и или $e \in [y]$, или $y \in N_S \cap S$, $(e, \gamma) \in PA(UA(user(y)))$, где $\gamma \in \{read_r, own_r\}$, или $y \in L_S \cap S$, $(e, read_r) \in PA(roles(y))$, $(y, e) \notin y(E)$).

Определение 15. Пусть $G = (PA, user, roles, A, F, H_E)$ – состояние системы $\Sigma(G^*, OP)$, в котором недоверенная субъект-сессия или недоверенный пользователь $x \in N_U \cup (N_S \cap S)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S$. Определим предикат $directly\_grant\_right(x, y, G)$, который будет истинным тогда и только тогда, когда выполняется одно из следующих условий 1 – 4.

Условие 1. Если $y \in N_U$ и $x \in N_U$, то существует роль $r \in can\_manage\_rights(AUA(x)) \cap UA(y)$.

Условие 2. Если $y \in N_U$ и $x \in N_S \cap S$, то существуют роль $r \in can\_manage\_rights(AUA(user(x))) \cap UA(y)$.

Условие 3. Если $y \in S$ и $x \in N_U$, то выполняется одно из условий:

- $y \in N_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(x)) \cap UA(user(y))$;
- $y \in L_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(x)) \cap roles(y)$.

Условие 4. Если $y \in S$ и $x \in N_S \cap S$, то выполняется одно из условий:

- $y \in N_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(user(x))) \cap UA(user(y))$.
- $y \in L_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(user(x))) \cap roles(y)$.

Возможно доказать утверждения 2 и 3, в которых обосновываются достаточные условия истинности предиката $can\_share((e, \alpha), x, G_0)$ для случая, когда при передаче прав доступа ролей взаимодействуют только две субъект-сессии двух пользователей.

Утверждение 2. Пусть $G_0 = (PA_0, user_0, roles_0, A_0, F_0, H_{E0})$ – состояние системы $\Sigma(G^*, OP)$, в котором существуют недоверенный пользователь или недоверенная субъект-сессия $x \in N_U \cup (N_S \cap S_0)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S_0$ и право доступа к сущности $(e, \alpha) \in P_0$. Пусть истинен предикат $directly\_access\_own(x, y, G_0)$ и выполняется одно из условий:

- если $y \in N_U$, то $(e, \alpha) \in PA_0(UA_0(y))$;
- если $y \in N_S \cap S_0$, то $(e, \alpha) \in PA_0(UA_0(user_0(y)))$;
- если $y \in L_S \cap S_0$, то $(e, \alpha) \in PA_0(roles_0(y))$.

Тогда выполняется одно из следующих условий.

Условие 1. Если $x \in N_U$, то истинен предикат $can\_share((e, \alpha), x, G_0)$.

Условие 2. Если $x \in N_S \cap S_0$, то истинен предикат $can\_share((e, \alpha), user_0(x), G_0)$.

Утверждение 3. Пусть $G_0 = (PA_0, user_0, roles_0, A_0, F_0, H_{E0})$ – состояние системы $\Sigma(G^*, OP)$, в котором существуют недоверенный пользователь или недоверенная субъект-сессия $x \in N_U \cup (N_S \cap S_0)$, субъект-сессия

или недоверенный пользователь $y \in N_U \cup S_0$ и право доступа к сущности $(e, \alpha) \in P_0$. Пусть истинен предикат *directly_grant_right*(x, y, G_0) и выполняется одно из условий:

- если $x \in N_U$, то $(e, own_r) \in PA_0(UA_0(x))$;
- если $x \in N_S \cap S_0$, то $(e, own_r) \in PA_0(UA_0(user_0(x)))$.

Тогда выполняется одно из условий.

Условие 1. Если $y \in N_U$, то истинен предикат *can_share*$((e, \alpha), y, G_0)$.

Условие 2. Если $y \in S_0$, то истинен предикат *can_share*$((e, \alpha), user_0(y), G_0)$.

Таким образом, с применением БР ДП-модели возможен анализ условий передачи прав доступа ролей в КС с РУД. В дальнейшем возможно обоснование условий возникновения в КС с РУД информационных потоков по памяти или по времени.

ЛИТЕРАТУРА

1. *Девянин П.Н.* Модели безопасности компьютерных систем: Учеб. пособие для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2005. 144 с.
2. *Bishop M.* Computer Security: Art and Science. 2002. 1084 p.
3. *Sandhu R.* Role-Based Access Control, Advanced in Computers // Academic Press. 1998. V. 46.
4. *Девянин П.Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006. 176 с.