МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.056.52

ПРЕДСТАВЛЕНИЕ СИСТЕМЫ СЕМАНТИЧЕСКИ ОСМЫСЛЕННОГО РОЛЕВОГО УПРАВЛЕНИЯ ДОСТУПОМ В ВИДЕ ЦВЕТНОЙ СЕТИ ПЕТРИ

Н. А. Семенова

Московский государственный институт электроники и математики (ТУ), г. Москва, Россия

E-mail: natasha_sem@inbox.ru

Рассматривается механизм внесения изменений в систему семантически осмысленного ролевого управления доступом в рамках СК-РУД модели одновременно несколькими администраторами. С использованием математического аппарата сетей Петри описываются процессы перехода системы между состояниями и обосновывается условие безопасности переходов.

Ключевые слова: *ролевое управление доступом, автоматизация управления ролями, сети Петри.*

Введение

Система семантически осмысленного ролевого управления доступом основана на модели ролевого управления доступом с семантическим контекстом СК-РУД [1] и позволяет автоматизировать назначение и отзыв ролей пользователям при наступлении в системе ряда определённых событий. Классическая модель ролевого управления доступом $RBAC$ [2], расширением которой является модель СК-РУД, содержит описание процессов внесения изменений в конфигурацию в предположении, что в каждый момент времени в системе действует не более одного субъекта с административными привилегиями. В системах ролевого управления доступом в компьютерных системах с десятками тысяч пользователей и тысячами привилегий администрирование, как правило, осуществляется несколькими пользователями. Данный подход получил название «распределённого администрирования» системы. Применение классического подхода для описания функционирования таких систем может привести к конфликтам между действиями различных администраторов. Примерами таких конфликтов могут быть ситуации, когда два администратора одновременно вносят изменения в учётную запись пользователя или заявка на назначение роли зарегистрирована и назначена одному из администраторов раньше, чем другому администратору назначена заявка на отзыв роли, взаимоисключающей первую роль. Исходя из этого, целесообразно рассмотреть вопрос формального описания процессов распределённого администрирования в системах ролевого управления доступом и определения условий безопасного перехода системы между состояниями.

1. Основные понятия и определения

В данной работе используются следующие обозначения для элементов СК-РУД модели:

P — множество привилегий или полномочий;

A — множество атрибутов учетных записей пользователей;

$V = \{(v_{ij})\}$, где i — номер атрибута из A; $j \in [1, N_i]$; N_i — число возможных различных значений атрибута $a_i \in A$;

$values : A \to 2^V$ — функция, задающая для каждого атрибута множество его допустимых значений;

U — множество атрибут-пользователей (множество векторов вида $(ux_1, \ldots, ux_{na})$, где $ux_i \in values(a_i)$);

R — множество ролей системы;

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

S — множество сессий пользователей системы;

$PA : R \to 2^P$ — функция, задающая для каждой роли множество привилегий. При этом для каждой привилегии $p \in P$ существует роль $r \in R$, такая, что $p \in PA(r)$;

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

$roles : S \to 2^R$ — функция, задающая для пользователя множество ролей, на которые он авторизован в данной сессии. При этом в каждый момент времени для каждой сессии $s \in S$ выполняется условие $roles(s) \in UA(user(s))$;

CR — множество предварительных условий назначения ролей;

AP — множество административных привилегий;

$AR \subseteq 2^{AP}$ — множество административных ролей;

$APA : AR \to 2^{AP}$ — функция, задающая для каждой административной роли множество административных привилегий; при этом для каждой привилегии $p \in AP$ существует роль $r \in AR$, такая что $p \in APA(r)$;

$can\_assign : AR \to CR \times 2^R$ — функция, определяющая для каждой административной роли множество ролей, которые могут быть назначены пользователю с использованием данной административной роли при выполнении заданных предварительных условий CR;

$can\_revoke : AR \to 2^R$ — функция, определяющая для каждой административной роли множество ролей, которые могут быть отозваны у пользователя с использованием данной административной роли.

Определение 1. Две роли $r_1, r_2 \in R$ называются статически взаимоисключающими, если они не могут быть назначены пользователю одновременно: для $s \in S$ выполняется неравенство $\mid \{r_1, r_2\} \cap roles(s) \mid \leqslant 1$.

Определение 2. Две роли $r_1, r_2 \in R$ называются динамически взаимоисключающими, если они не могут быть активированы сессией пользователя одновременно: для $u \in U$ выполняется неравенство $\mid \{r_1, r_2\} \cap UA(u) \mid \leqslant 1$.

Определение 3. Иерархией ролей «по предусловию» RH [1] будем называть заданное на множестве ролей R отношение строгого порядка «$>$». При этом по определению (для пользователя $u \in U$) если $(r_1, r_2) \in RH$, $r_1 > r_2$ и $c_{r_1}(u) = \texttt{true}$, то $c_{r_2}(u) = \texttt{true}$ и $c_{r_1}(u) = c_{r_1}(z_1(u), \ldots, c_{r_2}(u), \ldots, z_k(u))$. Роль r_1 будем называть предком r_2 «по предусловию».

Определение 4. Пусть $t_i : U \to \{\texttt{false}, \texttt{true}\}$ — функция, такая, что $t_i(u)$=`true` тогда и только тогда, когда $t_i \in values(a_i)$ и для $u \in U$ выполняется условие $ux_i = t_i$, где $a_i \in A$, $1 \leqslant i \leqslant |A|$.

Пусть $ca_r : U \to \{\texttt{false}, \texttt{true}\}$ — функция, такая, что $ca_r\,(u) = ca_r\,(t_1\,(u), \ldots, t_{na}\,(u))$, $u \in U$, $r \in R$, где $ca_r(y_1, \ldots, y_{na})$ — булева функция от na переменных, $na \in \{1, \ldots, |A|\}$. Тогда $CA = \{ca_{r1}, \ldots, ca_{rt}\}$ — множество функций, определяющих, при каких ограничениях на атрибуты учетной записи пользователя ему может быть назначена некоторая роль $r_i \in R$.

Дадим основные понятия и определения теории цветных сетей Петри (ЦСП), которые потребуются для дальнейшего описания результатов исследования [3]:

Q_c — множество цветов;

TK — множество токенов. По определению токен — это кортеж вида $tk = (x_1, \ldots, x_n)$, где x_i — переменная некоторого заданного цвета. Цвета токенов могут быть простыми или составными. Множество допустимых значений для токенов составных цветов — это декартово произведение множеств допустимых значений для входящих в его состав простых цветов;

$C_{tk} : TK \to Q_c$ — функция окрашивания, задающая для каждого токена его цвет;

Pl — множество мест (элементов $pl \in Pl$, каждый из которых содержит один или несколько токенов определенного цвета);

$C_{pl} : Pl \to Q_c$ — функция, определяющая множество допустимых цветов для данного места.

Множество цветов токенов, которые могут содержаться на месте pl, называется типом места pl. Множество токенов на месте pl называется содержимым места и обозначается $tk(pl)$.

Определение 5. Пусть Pl — конечное множество мест. Пусть $N_j(pl_i)$ — количество токенов каждого из цветов на месте $pl_i \in Pl$ в момент времени j, где $1 \leqslant i \leqslant |Pl|$. Пусть в момент времени $j+1$ существует такое $1 \leqslant k \leqslant |Pl|$, что $N_{j+1}(pl_k) \neq N_j(pl_k)$. Тогда будем говорить, что количество токенов на месте pl_k изменилось в результате активации перехода. Множество переходов будем обозначать T.

Определение 6. Дуга — это пара вида (pl_i, t_j), где $pl_i \in Pl$, $t_i \in T$. Множество всех дуг будем обозначать Ac.

Каждый переход инцидентен некоторой дуге, задающей правила его активации в зависимости от токенов, расположенных на местах, инцидентных той же дуге.

Дуги могут принадлежать к одному из следующих трех типов:

— NA — нормальные дуги;
— RA — ингибиторные дуги;
— IA — только чтение.

Дугу типа i будем обозначать ac_i.

Будем различать два направления дуг:

— входящие дуги, соединяющие место $pl \in Pl$ и переход $t \in T$;
— исходящие дуги, соединяющие переход $t \in T$ и место $pl \in Pl$.

Входящие дуги могут принадлежать к любому из трёх перечисленных типов и связаны с функциями чтения или удаления токена с места pl. Исходящие дуги всегда имеют тип NA и связаны с функциями добавления токенов на место pl.

Определение 7. Для каждой дуги $ac \in Ac$ определим значение количественной функции E, задающей для каждых места $pl \in Pl$ и перехода $t \in T$, инцидентных

дуге ac, количество токенов заданных цветов для возможности активации перехода t, которое либо должно содержать место pl, когда ac — входящая дуга, либо размещается на месте pl, когда ac — исходящая дуга:

$$E(ac) = m_1 \cdot tk_1 + m_2 \cdot tk_2 + \cdots + m_n \cdot tk_n,$$

где m_i — целые неотрицательные числа; tk_i — токены цвета $i \in C_{pl}(pl)$; $n = |C_{pl}(pl)|$.

Пример 1. Количественная функция $E(ac) = 3 \cdot \langle tk_1 \rangle + 2 \cdot \langle tk_2 \rangle$ задаёт для дуги ac условие активации перехода, инцидентного данной дуге: место $pl \in Pl$, инцидентное дуге, должно содержать три токена цвета tk_1 и два токена цвета tk_2.

Примечание. Если коэффициент m_i количественной функции равен 1, то вместо $1 \cdot tk$ будем писать tk.

2. Представление системы семантически осмысленного ролевого управления доступом в виде ЦСП

2.1. Цвета токенов и мест

В работе используются следующие цвета для обозначения элементов СК-РУД модели:

- $USER\langle u \rangle$ — идентификатор пользователя. Допустимыми значениями для токенов данного цвета являются все пользователи $u \in U$;
- $ROLE\langle r \rangle$ — идентификатор роли. Допустимыми значениями для токенов данного цвета являются все роли $r \in R$;
- $ADMROLE\langle ar \rangle$ — идентификатор административной роли. Допустимые значения: $ar \in AR$;
- $SESSION\langle s \rangle$ — идентификатор сессии. Допустимые значения: $s \in S$;
- $ACOND\langle ca_r \rangle$ — идентификатор атрибут-условия назначения роли. Допустимые значения: $ca_r \in CA$;
- $COMMAND\langle command \rangle$ — идентификатор действия. Допустимые значения: $command \in \{assign\_role, revoke\_role, auto\_assign\_role, auto\_revoke\_role, take\_role, remove\_role, auto\_recalculate\}$;
- $ROLE\_ACOND\langle ca_r, r \rangle$ — сопоставление роли и правила ее автоматического назначения. Множество допустимых значений — декартово произведение $ACOND \times \times ROLE$;
- $USER\_ROLE\langle u, r \rangle$ — назначение роли пользователю. Множество допустимых значений — $USER \times ROLE$;
- $UAS\langle u, r, s \rangle$ — текущая роль пользователя в сессии. Множество допустимых значений — $USER \times ROLE \times SESSION$;
- $USER\_ADMROLE\langle u, ar \rangle$ — назначение административной роли. Множество допустимых значений — $USER \times ADMROLE$;
- $CMD\langle command, u_1, u_2, r, s \rangle$ — токен действия, выполняемого сессией s от имени пользователя u_1. Множество допустимых значений — декартово произведение $COMMAND \times USER \times USER \times ROLE \times SESSION$. Если $u_1 = u_2$, то сессия выполняет операцию над пользователем, от имени которого она активирована.

Рассмотрим цвета мест и токенов, которые могут на них содержаться:

- ETG — генератор токенов. Хранит множество действий, разрешённых в текущем состоянии системы. Для выполнения каждого перехода необходимо, чтобы в ETG содержался токен соответствующего действия. Допустимым цветом токенов для данного места является CMD;

— UA — авторизованные роли. Токен $\langle u, r\rangle$ на этом месте означает, что пользователь u авторизован на роль r. Допустимый цвет токенов $USER\_ROLE$;
— AUA — авторизованные административные роли. Токен $\langle u, ar\rangle$ на этом месте означает, что пользователь u авторизован на административную роль ar. Допустимый цвет токенов $USER\_ADMROLE$;
— UAS — активация роли сессией пользователем. Каждый токен $\langle u, r, s\rangle$ на этом месте означает, что сессия s активирована от имени пользователя u, обладающего ролью r. Допустимый цвет токенов UAS;
— RCA — правила назначения ролей. Токен $\langle ca_r, r\rangle$ означает, что роли r соответствует условие назначения ca_r. Допустимый цвет токенов $ROLE\_ACOND$.

2.2. Количественные и ограничительные функции

Количественная и ограничительная функции дуг используются для представления системы ролевого управления доступом в виде ЦСП, так как данные функции представляют возможность моделировать ограничения взаимного исключения ролей и иерархические связи между ролями, задавая пред- и постусловия переходов системы между состояниями.

Рассмотрим, каким образом происходит активация перехода, связанного с дугой заданного типа и направления.

Для возможности активации перехода t по входящей дуге $ac_{NA} = (pl, t)$ каждое место pl, соединённое с переходом t дугой ac_{NA}, должно содержать токенов заданных цветов не меньше, чем указано количественной функцией дуги $E(ac)$, т. е. не менее m_i токенов цвета tk_i. Переход в момент времени j возможен, если для всех мест pl, инцидентных дуге ac, и для всех цветов i выполняется неравенство $N_j(pl, i) \geqslant m_i$, где $N_j(pl, i)$ — количество токенов цвета i на месте pl в момент времени j, а коэффициенты m_i задаются количественной функцией дуги $E(ac)$. В результате активации перехода токены удаляются из места pl: для всех $1 \leqslant i \leqslant n$ выполняется $N_{j+1}(pl, i) = N_j(pl, i) - m_i$.

Пример 2. Необходимым условием перехода, в результате которого сессия s от имени администратора $admin$ назначит роль r пользователю u, является наличие токена цвета $CMD\langle assign\_role, admin, u, r, s\rangle$ на месте ETG. После активации перехода соответствующий токен удаляется, предотвращая повторное назначение роли. Таким образом, значение количественной функции для входящей нормальной дуги, соединяющей место ETG и переход $assign\_role$, равно $\langle CMD\rangle$.

Переход по исходящей дуге ac_{NA} в момент времени j активируется в результате завершения всех входящих переходов, инцидентных данной дуге. В результате активации перехода по исходящей дуге $ac_{NA} = (t, pl)$ соответствующее количество токенов, равное $m_1 \cdot tk_1 + m_2 \cdot tk_2 + \cdots + m_n \cdot tk_n$, добавляется на место pl: $N_{j+1}(pl, i) = N_j(pl, i) + m_i$.

Пример 3. Результатом перехода, в рамках которого сессия от имени администратора $admin$ назначает роль r пользователю u, является помещение токена цвета $USER\_ROLE\langle u, r\rangle$ на место UA, означающее, что данный пользователь авторизован на роль r. Таким образом, значение количественной функции для соответствующей исходящей нормальной дуги равно $\langle USER\_ROLE\rangle$.

Для возможности активации перехода t по входящей дуге $ac_{RA} = (pl, t)$ в момент времени j каждое место pl, соединённое с переходом t, должно содержать токенов заданного цвета и значения не меньше, чем указано количественной функцией дуги $E(ac_{RA})$. Переход в момент времени j возможен, если для всех мест, инцидентных дуге ac, и для всех цветов i выполняется неравенство $N_j(pl, i) \geqslant m_i$. В результате активации перехода количество токенов на месте pl не изменяется: $N_{j+1}(pl, i) = N_j(pl, i)$.

Пример 4. Если для некоторой роли $r \in R$ существует роль $r' \geqslant r$, то необходимым условием перехода, в результате которого сессия от имени администратора назначит роль r пользователю u, является наличие токенов цвета $USER\_ROLE\langle u, r\rangle$ на месте UA для каждой из ролей-предков r. Каждый из токенов означает, что данный пользователь авторизован на соответствующую роль. После завершения перехода токены $USER\_ROLE$ остаются на месте UA. Таким образом, значение количественной функции для дуги RA равно $\mid \{r' : r' \geqslant r\} \mid \cdot\langle USER\_ROLE\rangle$.

Для входящей дуги ингибиторного типа $ac_{IA} = (pl, t)$ все коэффициенты m_i, за исключением одного m_j, равны нулю. Для возможности активации перехода t по входящей дуге $ac_{IA} = (pl, t)$ типа IA каждое место входа pl, соединённое с переходом t, должно содержать не более m_j токенов заданного цвета j. В результате активации перехода по дуге ac_{IA} количество токенов на месте pl не изменяется: $N_{j+1}(pl, i) = N_j(pl, i)$.

Пример 5. Пусть для роли r существуют статически взаимоисключающие роли. Тогда необходимым условием перехода, в результате которого сессия от имени администратора назначит роль r пользователю u, является отсутствие токенов цвета $USER\_ROLE\langle u, r\rangle$ на месте UA, означающее, что данный пользователь не авторизован на роли, взаимоисключающие с r. Таким образом, значение количественной функции для соответствующей входящей ингибиторной дуги равно $\langle USER\_ROLE\rangle$.

Введём определение ограничительной функции дуги G.

Определение 8. Множество ограничительных функций дуги G — это множество, включающее в себя ограничения на значения токенов, которые должны находиться на местах, инцидентных входящей дуге, для того чтобы стала возможна инициация перехода, инцидентного дуге. Пусть $v_p : TK \to \{\texttt{true}, \texttt{false}\}$ — одна из следующих функций:

- $assigned_r : U \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $assigned_r(u) = \texttt{true}$ тогда и только тогда, когда $r \in UA(u)$;
- $ca_r : U \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $ca_r(u) = \texttt{true}$ тогда и только тогда, когда атрибуты пользователя $u \in U$ соответствуют атрибут-условию назначения роли $r \in R$;
- $static\_conflict\_role_r : R \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $static\_conflict\_role_r(r') = \texttt{true}$ тогда и только тогда, когда r' является статически взаимоисключающей для r;
- $dynamic\_conflict\_role_r : R \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $dynamic\_conflict\_role_r(r') = \texttt{true}$ тогда и только тогда, когда r' является динамически взаимоисключающей для r;
- $can\_assign_{ar} : R \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $can\_assign_{ar}(r) = \texttt{true}$ тогда и только тогда, когда для $r \in R$ и $ar \in AR$ выполняется условие $r \in \in can\_assign(ar)$;
- $can\_revoke_{ar} : R \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $can\_revoke_{ar}(r) = \texttt{true}$ тогда и только тогда, когда для $r \in R$ и $ar \in AR$ выполняется условие $r \in \in can\_revoke(ar)$;
- $prec\_role_r : R \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $prec\_role_r(r') = \texttt{true}$ тогда и только тогда, когда $r > r'$ в иерархии RH.

Пусть $G_{ac} : Pl \to \{\texttt{true}, \texttt{false}\}$ — функция, такая, что $G_{ac}(pl) = G_{ac}(v_p(tk_1, tk'_1), \ldots, v_p(tk_m, tk'_m))$, $ac \in Ac$, $pl \in Pl$, pl инцидентно ac, $tk_1, \ldots, tk_m$, $tk'_1, \ldots, tk'_m \in TK$, $C_{tk}(tk_1) \subseteq C_{pl}(pl)$, где $1 \leqslant i \leqslant m$ и $G_{ac}(y_1, \ldots, y_m)$ — булева функция от m перемен-

ных. Тогда $\{G_{ac}(pl_1), \ldots, G_{ac}(pl_t)\}$ — множество функций, определяющих, при каких условиях может быть активирован переход, инцидентный соответствующей дуге.

Если с некоторой входящей дугой $ac \in Ac$ связана ограничительная функция G_{ac}, то соответствующий дуге переход t может быть активирован только в том случае, если для всех мест, инцидентных данной дуге, $G_{ac}(pl) = \texttt{true}$. Количественная и ограничительная функции для i-й дуги j-го типа обозначаются как Ei_j и G_i соответственно.

Пример 6. Пусть для некоторой роли $r \in R$ существует $r' \geqslant r$. Тогда необходимым условием перехода, в результате которого сессия от имени администратора $admin$ назначит роль r_j пользователю u, является наличие на месте UA токена цвета $USER\_ROLE$, имеющего значение $\langle u, r_i\rangle$, для которого $assigned_{r_i}(u) = \texttt{true}$. После завершения перехода токен $\langle u, r_i\rangle$ остается на месте UA. Таким образом, значение ограничительной функции входящей дуги RA равно $G_{ac} = assigned_{r_i}(u)$.

Пример 7. Пусть $r_i, r_j \in R$ — две статически взаимоисключающие роли. Тогда необходимым условием перехода, в результате которого сессия от имени администратора $admin$ назначит роль r_j пользователю u, является отсутствие на месте UA токена цвета $USER\_ROLE$ со значением $\langle u, r_i\rangle$, для которого $assigned_{r_i}(u) = \texttt{true}$. Таким образом, значение ограничительной функции входящей ингибиторной дуги равно $G_a(\neg assigned_{r_i}(u))$, где $\neg$ — оператор логического отрицания.

По определению [3], цветная сеть Петри (ЦСП) — это кортеж $CPN = (Q_c, TK, Pl, T, Ac, C_{tk}, C_{pl}, G, E)$. Состояние ЦСП в каждый момент времени задается размещением токенов на каждом из Pl мест.

Определение 9. Состояние ЦСП M_n называется достижимым из состояния M_0, если существует конечная последовательность переходов $t_0, t_1, \ldots, t_{n-1} \in T$, переводящая ЦСП из состояния M_0 в состояние M_n. Начальное состояние M_0 является достижимым по определению.

Переход сети в новое состояние может быть активирован только в том случае, если хотя бы для одного из переходов $t \in T$ выполнены все предусловия, задаваемые ограничительными и количественными функциями дуг.

В данной работе для описания системы СК-РУД используются следующие функции дуг и ограничений:
$E1_{NA} = CMD\langle assign\_role, u, u', r, s\rangle$;
$E2_{RA} = E11_{RA} = E40_{RA} = E43_{RA} = E58_{RA} = USER\_ADMROLE\langle u', ra\rangle$;
$E3_{IA} = E19_{IA} = USER\_ROLE\langle u, r_c\rangle$;
$E4_{RA} = E20_{RA} = USER\_ROLE\langle u, r_{pc}\rangle$;
$E5_{IA} = E6_{NA} = E9_{NA} = E14_{IA} = E15_{IA} = E17_{NA} = E21_{IA} = E22_{NA} = E24_{NA} = = E28_{IA} = E29_{IA} = E31_{NA} = E45_{IA} = E55_{IA} = E63_{IA} = USER\langle u\rangle$;
$E7_{NA} = E8_{RA} = E12_{RA} = E16_{NA} = E23_{NA} = E26_{RA} = E30_{NA} = E33_{RA} = E41_{RA} = = E48_{RA} = E53_{IA} = E59_{RA} = USER\_ROLE\langle u, r\rangle$;
$E10_{NA} = E39_{NA} = E57_{RA} = CMD\langle revoke\_role, u, u', r, s\rangle$;
$E13_{IA} = E27_{IA} = E42_{RA} = E60_{RA} = USER\_ROLE\langle u, r_d\rangle$;
$E18_{NA} = E56_{NA} = CMD\langle auto\_assign\_role, u, system, r, s\rangle$;
$E25_{NA} = CMD\langle auto\_revoke\_role, u, system, r, s\rangle$;
$E32_{NA} = CMD\langle take\_role, u, u, r, s\rangle$;
$E34_{IA} = UAS\langle u, r_{dc}, s\rangle$;
$E35_{NA} = E37_{NA} = E38_{NA} = E61_{RA} = UAS\langle u, r, s\rangle$;
$E36_{NA} = CMD\langle remove\_role, u, u, r, s\rangle$;
$E44_{IA} = E46_{NA} = CMD\langle revoke\_role, u, u', rd, s\rangle$;

$E47_{RA} = E51_{RA} = CMD\langle auto\_recalculate, u, system, r, s\rangle$;
$E49_{IA} = E50_{NA} = CMD\langle auto\_revoke\_role, u, system, r, s\rangle$;
$E52_{RA} = ROLE\_ACOND\langle ca_r, r\rangle$;
$E54_{IA} = USER\_ROLE\langle u, r_{old}\rangle$;
$E62_{IA} = CMD\langle revoke\_role, u, u, r, s\rangle + CMD\langle auto\_revoke\_role, u, system, r, s\rangle$;
$G_2 = \langle u', r_a\rangle : can\_assign_{r_a}(r) = \texttt{true}$;
$G_3 = \langle u, r_c\rangle : static\_conflict\_role_r(r_c) = \texttt{false}$;
$G_4 = \langle u, r_{pc}\rangle : prec\_role_r(u) = \texttt{true}$;
$G_{11} = \langle u', r_a\rangle : can\_revoke_{r_a}(r) = \texttt{true}$;
$G_{13} = \langle u, r_d\rangle : prec\_role_r(r_d) = \texttt{false}$;
$G_{19} = \langle u, r_c\rangle : static\_conflict\_role_r(r_c) = \texttt{false}$;
$G_{20} = \langle u, r_{pc}\rangle : prec\_role_r(u) = \texttt{true}$;
$G_{27} = \langle u, r_d\rangle : prec\_role_r(r_d) = \texttt{false}$;
$G_{33} = \langle u, r\rangle : assigned_r(u) = \texttt{true}$;
$G_{34} = \langle u, r_{dc}, s\rangle : dynamic\_conflict\_role_r(r_{dc}) = \texttt{false}$;
$G_{40} = \langle u', r_a\rangle : can\_revoke_{r_a}(r) = \texttt{true}$;
$G_{42} = \langle u, r_d\rangle : prec\_role_r(r_d) = \texttt{true}$;
$G_{43} = \langle u', r'_a\rangle : can\_revoke_{r'_a}(r_d) = \texttt{true}$;
$G_{48} = \langle u, r\rangle : ca_r(u) = \texttt{false}$;
$G_{52} = \langle ca_r, r\rangle : ca_r(u) = \texttt{true}$;
$G_{53} = \langle u, r\rangle : ca_r(u) = \texttt{true}$;
$G_{54} = \langle u, r_{old}\rangle : ca_{r_{old}}(u) = \texttt{false}$.

3. Переходы сети Петри между состояниями

Рассмотрим переходы следующих типов:

— назначение роли ($assign\_role$): сессия s администратора u' назначает пользователю $u \in U$ роль r;
— отзыв роли ($revoke\_role$): роль r отзывается у пользователя u сессией s администратора u';
— автоматическое назначение роли ($auto\_assign\_role$): роль r назначается пользователю u по атрибут-условию ca_r;
— автоматический отзыв ($auto\_revoke\_role$): роль r отзывается у пользователя u, если он более не удовлетворяет условию ca_r;
— перерасчёт списка ролей пользователя при изменении его атрибутов ($auto\_recalculate$): в системе создаётся очередь команд на отзыв и назначение ролей в соответствии с новыми атрибутами учётной записи пользователя;
— активация ($take\_role$): активация роли r сессией s от имени пользователя u;
— деактивация ($remove\_role$): роль r удаляется из списка активных ролей для заданной сессии s пользователя u.

Переход может быть совершён в любой момент, если он разрешён в данном состоянии действующими ограничениями. Любой из данных переходов переводит систему в новое состояние. Рассмотрим подробнее переходы в системе ролевого управления доступом, представленной в виде ЦСП.

Каждый из рассматриваемых переходов в ЦСП описывается следующим образом: сначала перечисляются входящие дуги с указанием в скобках их типа и связанных количественных и ограничительных функций. Результат перехода описывается с помощью перечисления исходящих дуг с указанием их количественных функций.

3.1. Переход «Назначение роли»

Правила активации перехода:

— (Дуга с количественной функцией $E1_{NA}$.) Существует токен $\langle assign\_role, u, u', r, s\rangle$ на месте ETG, означающий, что необходимо выполнить назначение роли r пользователю u сессией от имени администратора u'.
— ($E2_{RA}$). На месте AUA должен существовать токен $\langle u', r_a\rangle$, для которого $can\_assign_{r_a}(r) = \texttt{true}$ (ограничительная функция G_2).
— ($E3_{IA}$). Место UA не содержит токенов $\langle u, r_c\rangle$, для которых $static\_conflict\_role_r(r_c) = \texttt{true}$ (G_3: пользователю не назначены роли, взаимоисключающие с ролью r).
— ($E4_{RA}$). Место UA содержит токены $\langle u, r_{pc}\rangle$, для которых $prec\_role_r(r_{pc}) = \texttt{true}$ (G_4: пользователю назначены все роли, необходимые для назначения роли r).
— ($E5_{IA}$). Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход ($E7_{NA}$) помещает токен $\langle u, r\rangle$ на место UA, что означает авторизацию пользователя u на роль r. На время выполнения перехода на место UB помещается токен $\langle u\rangle$ ($E6_{NA}$), чтобы блокировать одновременное выполнение любых других действий над пользователем u. После активации перехода команда $\langle assign, u, u', r, s\rangle$ удаляется с места ETG ($E1_{NA}$). После выполнения перехода ($E8_{RA}$) токен $\langle u\rangle$ удаляется с места UB ($E9_{NA}$).

3.2. Переход «Отзыв роли»

Правила активации перехода:

— ($E10_{NA}$). Существует токен $\langle revoke\_role, u, u', r, s\rangle$ на месте ETG, означающий, что необходимо выполнить отзыв роли r у пользователя u сессией от имени администратора u'.
— ($E11_{RA}$). На месте AUA существует токен $\langle u', r_a\rangle$, для которого $can\_revoke_{r_a}(r) = = \texttt{true}$ (G_{11}).
— ($E12_{RA}$). Место UA содержит токен $\langle u, r\rangle$ (нельзя отозвать роль, если она не назначена).
— ($E13_{IA}$). Место UA не содержит множество токенов $\{\langle u, r_d\rangle\}$, таких, что $prec\_role_r(r_d) = \texttt{true}$ (G_{13}: нельзя отозвать роль, предварительно не отозвав роли, которые от неё зависят).
— ($E14_{IA}$). Место UAS не содержит токена $\langle u, r, s\rangle$ (G_{14}: нельзя отозвать роль, если она активирована сессией пользователя).
— ($E15_{IA}$). Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никакой операции).

Результат перехода: данный переход ($E17_{NA}$) удаляет токен $\langle u, r\rangle$ с места UA, что означает отзыв роли r у пользователя u. На время выполнения перехода на место UB помещается токен $\langle u\rangle$ ($E16_{NA}$), чтобы блокировать одновременное выполнение любых других действий над пользователем u. После активации перехода команда $\langle revoke\_role, u, u', r, s\rangle$ удаляется с места ETG ($E10_{NA}$). После выполнения перехода токен $\langle u\rangle$ удаляется с места UB.

3.3. Переход «Автоматическое назначение роли»

Правила активации перехода:

— ($E18_{NA}$). Существует токен $\langle auto\_assign\_role, u, system, r, s\rangle$ на месте ETG, означающий, что необходимо выполнить назначение роли r пользователю u сессией от имени пользователя $system$.

— ($E19_{IA}$). Место UA не содержит токенов $\langle u, r_c\rangle$, для которых $static\_conflict\_role_r(r_c) = \texttt{true}$ (G_{19}: пользователю не назначены роли, взаимоисключающие с ролью r).

— ($E20_{RA}$). Место UA содержит токены $\langle u, r_{pc}\rangle$, для которых $prec\_role_r(r_{pc}) = \texttt{true}$ (G_{20}: пользователю назначены все роли, необходимые для назначения роли r).

— ($E21_{IA}$). Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход ($E23_{NA}$) помещает токен $\langle u, r\rangle$ на место UA, что означает авторизацию пользователя u на роль r. На время выполнения перехода на место UB помещается токен $\langle u\rangle$ ($E22_{NA}$), чтобы блокировать одновременное выполнение любых других действий над пользователем u. После активации перехода команда $\langle auto\_assign\_role, u, system, r, s\rangle$ удаляется с места ETG ($E18_{NA}$). После выполнения перехода токен $\langle u\rangle$ удаляется с места UB ($E24_{NA}$).

3.4. Переход «Автоматический отзыв роли»

Правила активации перехода:

— ($E25_{NA}$). Существует токен $\langle auto\_revoke\_role, u, system, r, s\rangle$ на месте ETG, означающий, что сессии от имени пользователя $system$ необходимо выполнить отзыв роли r у пользователя u.

— ($E26_{RA}$). Место UA содержит токен $\langle u, r\rangle$ (нельзя отозвать роль, если она не назначена).

— ($E27_{IA}$). Место UA не содержит множество токенов $\{\langle u, r_d\rangle\}$, таких, что $prec\_role_r(r_d) = \texttt{true}$ (нельзя отозвать роль, предварительно не отозвав роли, которые от неё зависят).

— ($E28_{IA}$). Место UAS не содержит токена $\langle u, r, s\rangle$ (G_{14}: нельзя отозвать роль, если она активирована одной из сессий пользователя).

— ($E29_{IA}$). Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход ($E31_{NA}$) удаляет токен $\langle u, r\rangle$ с места UA, что означает отзыв роли r у пользователя u. На время выполнения перехода на место UB помещается токен $\langle u\rangle$ ($E30_{NA}$), чтобы блокировать одновременное выполнение любых других действий над пользователем u. После активации перехода команда $\langle auto\_revoke\_role, u, system, r, s\rangle$ удаляется с места ETG ($E25_{NA}$). После выполнения перехода токен $\langle u\rangle$ удаляется с места UB.

3.5. Переход «Активация роли»

Правила активации перехода:

— ($E32_{NA}$). Существует токен $\langle take\_role, u, u, r, s\rangle$ на месте ETG, означающий, что сессия s от имени пользователя u активирует роль r.

— ($E33_{RA}$). Место UA содержит токен $\langle u, r\rangle$ (сессия пользователя может активировать только те роли, на которые он авторизован).

— ($E34_{IA}$). Место UAS не содержит множество токенов $\{\langle u, r_{dc}, s\rangle\}$, таких, что $dynamic\_conflict\_role_r(r_{dc}) = \texttt{true}$ (G_{34}: сессия пользователя не может активи-

ровать роль r, если для этой сессии в настоящий момент активированы роли, динамически взаимоисключающие с ролью r).

Результат перехода: данный переход ($E35_{NA}$) помещает токен $\langle u, r, s\rangle$ на место UAS, что означает активацию роли r сессией s от имени пользователя u.

3.6. Переход «Деактивация роли»

Правила активации перехода:

— ($E36_{NA}$). Существует токен $\langle remove, u, u, r, s\rangle$ на месте ETG, означающий, что сессия s пользователя u деактивирует роль r.
— ($E37_{NA}$). Место UAS содержит токен $\langle u, r, s\rangle$ (сессия пользователя не может деактивировать роль, которая не активирована для данной сессии).

Результат перехода: данный переход ($E38_{NA}$) удаляет токен $\langle u, r, s\rangle$ с места UAS, что означает деактивацию роли r сессией s от имени пользователя u.

3.7. Переход «Перерасчет зависимых ролей при удалении роли»

Правила активации перехода:

— ($E39_{NA}$). Существует токен $\langle revoke\_role, u, u', r, s\rangle$ на месте ETG, означающий, что необходимо выполнить команду отзыва роли r у пользователя u сессией от имени администратора u'.
— ($E40_{RA}$). На месте AUA существует токен $\langle u', r_a\rangle$, для которого $can\_revoke_{r_a}(r) = \texttt{true}$ (G_{40}).
— ($E41_{RA}$). Место UA содержит токен $\langle u, r\rangle$ (нельзя отозвать роль, если она не назначена).
— ($E42_{RA}$). Место UA содержит хотя бы один токен $\langle u, r_d\rangle$, такой, что $prec\_role_r(r_d) = \texttt{true}$.
— ($E43_{RA}$). На месте AUA существует токен $\langle u', r'_a\rangle$, для которого $can\_revoke_{r'_a}(r_d) = \texttt{true}$ (G_{43}).
— ($E44_{IA}$). Не существует токена $\langle revoke\_role, u, u', r_d, s\rangle$ на месте ETG.
— ($E45_{IA}$). Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход ($E46_{NA}$) добавляет токен $\langle revoke, u, u', r_d, s\rangle$ на место ETG, что означает команду на отзыв роли r_d, зависимой от роли r, у пользователя u.

3.8. Переход «Перерасчет ролей для удаления при изменении атрибутов»

Правила активации перехода:

— ($E47_{RA}$). Существует токен $\langle auto\_recalculate, u, system, r, s\rangle$ на месте ETG, означающий, что атрибуты учётной записи пользователя u были изменены и требуется перерасчет его ролей.
— ($E48_{RA}$). Место UA содержит хотя бы один токен $\langle u, r\rangle$, такой, что $ca_r(u) = \texttt{false}$.
— ($E49_{IA}$) Не существует токена $\langle auto\_revoke\_role, u, system, r, s\rangle$ на месте ETG.

Результат перехода: данный переход ($E50_{NA}$) добавляет токен $\langle auto\_revoke\_role, u, system, r, s\rangle$ на место ETG, что означает команду на автоматический отзыв роли r у пользователя u.

3.9. Переход «Перерасчет ролей для назначения при изменении атрибутов»

Правила активации перехода:

— $(E51_{RA})$. Существует токен $\langle auto\_recalculate, u, system, r, s\rangle$ на месте ETG, означающий, что атрибуты учётной записи пользователя u были изменены и требуется перерасчет его ролей.
— $(E52_{RA})$. Место RCA содержит хотя бы один токен $\langle ca_r, r\rangle$, такой, что $ca_r(u) = \texttt{true}$.
— $(E53_{IA})$. Место UA не содержит токен $\langle u, r\rangle$, такой, что $ca_r(u) = \texttt{true}$.
— $(E54_{IA})$. Место UA не содержит ни одного токена $\langle u, r_{old}\rangle$, такого, что $ca_{r_{old}}(u) = \texttt{false}$ (данное условие гарантирует, что расчет назначений ролей начнётся только после окончания отзыва всех старых ролей).
— $(E55_{IA})$. Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход $(E56_{NA})$ добавляет токен $\langle auto\_assign\_role, u, system, r, s\rangle$ на место ETG, что означает переход «Автоматическое назначение роли» r пользователю u.

3.10. Переход «Удаление роли из списка активных в результате её отзыва у пользователя»

Правила активации перехода:

— $(E57_{RA})$. Существует токен $\langle revoke\_role, u, admin, r, s\rangle$ на месте ETG, означающий, что необходимо выполнить команду отзыва роли r у пользователя u сессией от имени администратора $admin$.
— $(E58_{RA})$. На месте AUA существует токен $\langle admin, r_a\rangle$, для которого $can\_revoke_{r_a}(r) = \texttt{true}$ (G_{58}).
— $(E59_{RA})$. Место UA содержит токен $\langle u, r\rangle$ (нельзя отозвать роль, если она не назначена).
— $(E60_{RA})$. Место UA не содержит токенов $\langle u, r_d\rangle$, таких, что $prec\_role_r(r_d) = \texttt{true}$.
— $(E61_{RA})$. Место UAS содержит токен $\langle u, r, s\rangle$ (роль в данный момент времени активирована).
— $(E62_{IA})$. Не существует токенов $\langle remove\_role, u, u, r, s\rangle$ и $\langle remove\_role, u, system, r, s\rangle$ на месте ETG.
— $(E63_{IA})$. Место UB не содержит токена $\langle u\rangle$ (над пользователем в данный момент не выполняется никаких действий).

Результат перехода: данный переход $(E64_{NA})$ добавляет токен $\langle remove\_role, u, system, r, s\rangle$ на место ETG, что означает команду на деактивацию роли r, активированной пользователем u в рамках сессии s.

Таким образом, описаны переходы между состояниями в системе семантически осмысленного ролевого управления доступом, заданной в рамках СК-РУД-модели. В отличие от системы, основанной на классической модели $RBAC$, возможность осуществления перехода зависит не только от текущего состояния системы, но и от переходов, выполненных на предыдущих шагах.

4. Доказательство безопасности системы семантически осмысленного ролевого управления доступом, представленной в виде ЦСП

Представление системы ролевого управления доступа в виде ЦСП позволяет осуществить формальное доказательство безопасности системы. Назовём состояние системы безопасным в рамках СК-РУД-модели, если оно не противоречит ограничениям, на-

кладываемым на составляющие её множества и функции, и каждому пользователю $u \in U$ назначены только роли, не противоречащие условиям CR и CA.

Переформулируем определение с учётом перечисленных в п. 1 элементов СК-РУД-модели.

Определение 10. Состояние системы в рамках СК-РУД-модели называется безопасным, если в нем выполняются следующие свойства:

— P_1: у каждого пользователя активны только те роли, на которые он авторизован: в каждый момент времени для каждой сессии $s \in S$ выполняется условие $roles(s) \subseteq \subseteq UA(user(s))$ (условие определения функции $roles$);
— P_2: если $prec\_role_r(r') = \texttt{true}$, то роль r может быть назначена пользователю u только в том случае, если он также авторизован на роль r' (определение 3 иерархии «по предусловию» и определение предусловий CR [4]);
— P_3: пользователь может быть авторизован только на те атрибут-роли, атрибут-условиям которых соответствуют значения его атрибутов: для любого пользователя $u \in U$ если $r \in UA(u)$, то $ca_r(u) = \texttt{true}$ (ограничение в определении 4 атрибут-условия);
— P_4: пользователь не может быть одновременно авторизован на две статически взаимоисключающие роли: для любого пользователя $u \in U$ если $r_1, r_2 \in UA(u)$, то $static\_conflict\_role_{r_1}(r_2) = \texttt{false}$ (определение 1 статически взаимоисключающих ролей);
— P_5: у любой сессии пользователя не могут быть одновременно активированы две динамически взаимоисключающие роли: для сессии любого пользователя $s \in S$ если $r_1, r_2 \in roles(s)$, то $dynamic\_conflict\_role_{r_1}(r_2) = \texttt{false}$ (определение 2 динамически взаимоисключающих ролей).

Определение 11. Переход системы из состояния в состояние называется безопасным, если он удовлетворяет следующим ограничениям:

— T_1: переход «Активация роли» сессией пользователя может быть осуществлён только для тех ролей, на которые соответствующий пользователь авторизован в настоящий момент времени;
— T_2: переход «Назначение роли» r пользователю u сессией от имени администратора $admin$ может произойти только в том случае, если $can\_assign_r(admin) = \texttt{true}$;
— T_3: переход «Назначение роли» r пользователю u может быть осуществлён только в том случае, если $prec\_role_r(r_2) = \texttt{true}$ и пользователь в настоящий момент времени авторизован на роль r_2;
— T_4: переход «Назначение роли» r пользователю u может быть осуществлён только в том случае, если он в настоящий момент времени не авторизован на одну или более статически взаимоисключающих ролей;
— T_5: переход «Активация роли» r пользователю u может быть осуществлён только в том случае, если в настоящий момент времени в этой сессии не активны роли, динамически взаимоисключающие с r;
— T_6: переход «Отзыв роли» r у пользователя u сессией от имени администратора $admin$ может быть осуществлён только в том случае, если $can\_revoke_r(admin) = = \texttt{true}$;
— T_7: переход «Отзыв роли» r_2 у пользователя u может быть осуществлён только в том случае, если $prec\_role_r(r_2) = \texttt{true}$ и пользователь в настоящий момент времени авторизован на роль r.

Утверждение 1. Для системы ролевого управления доступом, начальное состояние M_0 которой является безопасным, все состояния M, достижимые из M_0 в результате конечного числа переходов, удовлетворяют свойству P_1. Для пользователя могут быть активными только те роли, на которые он авторизован.

Доказательство. Входящая дуга A_{33} между UA и переходом $take\_role$ даёт возможность выполнить переход $take\_role$ для пользователя u только в том случае, если токен $\langle u, r_y\rangle$ находится на месте UA (гарантирует выполнение ограничения T_1). Наличие токена $\langle u, r_y\rangle$ на месте UA означает, что пользователь u авторизован на роль r_y. Следовательно, множество ролей, активированных сессией пользователя, является подмножеством множества ролей, на которые он авторизован. ■

Утверждение 2. Для системы ролевого управления доступом, начальное состояние M_0 которой является безопасным, все состояния M, достижимые из M_0, удовлетворяют свойству P_2.

Доказательство. Рассмотрим ограничение последовательного назначения ролей, согласно которому, если некоторая роль r_x имеет зависимость от назначения подмножества ролей, то роль r_x может быть назначена пользователю только в том случае, если все роли r_y, такие, что $prec\_role_{r_x}(r_y) = \texttt{true}$, назначены данному пользователю. Предположим, что ограничение не выполняется для некоторого состояния M, достижимого из M_0. Предположим, что $\{r_1, r_2, \ldots, r_n\}$ — такие роли, что $prec\_role_{r_x}(r_i) = \texttt{true}$, $1 \leqslant i \leqslant n$. Согласно предположению, в состоянии M место UA содержит токен $\langle u, r_x\rangle$, но не содержит все токены множества $\{\langle u, r_i\rangle : 1 \leqslant i \leqslant n\}$. Предположим, что M — первое состояние, в котором роль r_x назначена пользователю. Это означает, что состояние M было достигнуто при выполнении перехода $assign\_role$ ($auto\_assign\_role$) из состояния M', в котором токены $\langle u, r_i\rangle$ также отсутствовали на месте UA. Однако дуга A_3 (A_{19}) и соответствующее ограничение G_3 (G_{19}) $prec\_role_{r_x}(r_i)$ не позволяет активировать переход $assign\_role$ ($auto\_assign\_role$), гарантируя выполнение ограничения T_3. Следовательно, состояние M не может быть достигнуто. Противоречие. ■

Утверждение 3. Для системы ролевого управления доступом, начальное состояние M_0 которой является безопасным, все состояния M, достижимые из M_0, удовлетворяют свойству P_3.

Доказательство. Рассмотрим ограничение последовательного отзыва ролей, согласно которому, если для ролей из множества $\{r_1, r_2, \ldots, r_n\}$ роль r_x входит в предварительное условие их назначения, то роль r_x может быть отозвана только в том случае, если все роли $\{r_1, r_2, \ldots, r_n\}$, такие, что $prec\_role_{r_i}(r_x) = \texttt{true}$, отозваны у пользователя. Предположим противное: ограничение не выполняется для некоторого состояния M, достижимого из M_0. Пусть $\{r_1, r_2, \ldots, r_n\}$ — такие роли, что $prec\_role_{r_i}(r_x) = \texttt{true}$, $1 \leqslant i \leqslant n$. Согласно предположению, в состоянии M место UA содержит токен $\langle u, r_x\rangle$, но не содержит всех токенов из множества $\{\langle u, r_i\rangle\}$, где $1 \leqslant i \leqslant n$. Предположим, что M — первое состояние, в котором роль r_i, $1 \leqslant i \leqslant n$, отозвана у пользователя. Это означает, что состояние M было достигнуто при выполнении перехода $revoke\_role$ ($auto\_revoke\_role$) из состояния M', в котором все токены $\langle u, r_i\rangle$ присутствовали на месте UA. Однако дуга A_{13} (A_{27}) и соответствующее ограничение G_{13} (G_{27}) $prec\_role_{r_i}(r_x)$ не позволяет активировать переход $revoke\_role$ ($auto\_revoke\_role$), гарантируя выполнение ограничения T_7. Следовательно, состояние M не может быть достигнуто. Противоречие. ■

Утверждение 4. Для системы ролевого управления доступом, начальное состояние M_0 которой является безопасным, все состояния M, достижимые из M_0, удовлетворяют свойству P_4.

Доказательство. Предположим противное: в системе существует пользователь u, для которого ограничение на запрет одновременного назначения статически взаимоисключающих ролей не выполняется, т. е. u может быть одновременно авторизован на роли r_x и r_y, такие, что $static\_conflict\_role_{r_x}(r_y) = \texttt{true}$. Следовательно, токены $\langle u, r_x\rangle$ и $\langle u, r_y\rangle$ могут одновременно находиться на месте UA. Пусть M_0 — состояние системы непосредственно перед выполнением перехода $assign\_role$, добавляющего токен $\langle u, r_y\rangle$ на место UA. Пусть также в состоянии M_0 пользователь u авторизован на роль r_x (токен $\langle u, r_x\rangle$ находится на месте UA, а токен $\langle u, r_y\rangle$ отсутствует на данном месте). Пусть r_y — первая роль, для которой не выполняется ограничение P_4. Ингибиторная дуга A_3 связана с функцией $E3 : \langle u, r_c\rangle$ и ограничением перехода G_3: $static\_conflict\_role_r(r_c) = \texttt{true}$, отсюда следует, что выполняется условие T_4 и переход $assign\_role$ не может быть активирован. Следовательно, свойство P_4 выполняется для любого состояния M, достижимого из M_0. ■

Утверждение 5. Для системы ролевого управления доступом, начальное состояние M_0 которой является безопасным, все состояния M, достижимые из M_0, удовлетворяют свойству P_5.

Доказательство. Предположим противное: в системе существует пользователь u, для которого ограничение на запрет одновременного назначения взаимоисключающих ролей не выполняется, т. е. u может одновременно активировать роли r_x и r_y, такие, что $dynamic\_conflict\_role_{r_x}(r_y) = \texttt{true}$. Следовательно, токены $\langle u, r_x, s\rangle$ и $\langle u, r_y, s\rangle$ могут одновременно находиться на месте UAS. Пусть M_0 — состояние системы непосредственно перед выполнением перехода $assign\_role$, добавляющего токен $\langle u, r_y, s\rangle$ на место UAS. Пусть также в состоянии M_0 сессия от имени пользователя u активирует роль r_x (токен $\langle u, r_x, s\rangle$ находится на месте UAS, а токен $\langle u, r_y, s\rangle$ отсутствует на данном месте). Пусть r_y — первая роль, для которой не выполняется ограничение P_5. Ингибиторная дуга A_{34} связана с функцией E_{34}: $\langle u, r_{dc}, s\rangle$ и ограничением перехода G_{34}: $dynamic\_conflict\_role_r(r_{dc}) = \texttt{true}$, отсюда следует, что выполняется условие T_5 и переход $take\_role$ не может быть активирован. Противоречие. Следовательно, свойство P_5 выполняется для любого состояния M, достижимого из M_0. ■

Теорема 1. Если начальное состояние системы M_0, представленной в виде ЦСП, безопасно, то все состояния M, достижимые из M_0 посредством безопасных переходов, безопасны.

Доказательство. Из утверждений 1–5 следует, что все безопасные переходы между состояниями сохраняют свойства P_1–P_5. Следовательно, по определению 10, любое достижимое состояние является безопасным. ■

Таким образом, теоретически обоснованы достаточные условия безопасности систем, реализующих автоматизированное назначение ролей в рамках СК-РУД-модели.

Заключение

Представление системы семантически осмысленного ролевого управления доступом в рамках модели СК-РУД в виде сети Петри позволяет адаптировать её к условиям функционирования реальных компьютерных сетей предприятий, допускающих распределённое администрирование, а также проводить анализ переходов между состоя-

ниями, состоящих из двух и более операций. Безопасность системы ролевого управления доступом, построенной на основе предложенной модели СК-РУД, подтверждается проведённым анализом переходов между состояниями.

ЛИТЕРАТУРА

1. *Семенова Н. А.* Семантическая ролевая модель управления доступом // Прикладная дискретная математика. 2012. № 2(16). С. 50–64.
2. `http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf` — National Institute of Standards and Technology, Proposed Standard for Role-Based Access Control.
3. *Котов В. Е.* Сети Петри. М.: Наука, 1984. 160 с.
4. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. М.: Горячая линия-Телеком, 2011. 320 с.