Лекция 5. Организация разграничения доступа в базы данных на основе дискретной модели

1. Общая характеристика политики дискреционного доступа
2. Пятимерное пространство Хартсона
3. Модели на основе матрицы доступа
4. Модели распространения прав доступа

1. Общая характеристика политики дискреционного доступа

Исходные понятия

Разграничение доступа к информации (данным) КС

-разделение информации АИС на объекты (части, элементы, компоненты и т. д.), и организация такой системы работы с информацией, при которой пользователи имеют доступ только и только к той части информации (к тем данным), которая им необходима для выполнения своих функциональных обязанностей или необходима исходя из иных соображений

-создание такой системы организации данных, а также правил и механизмов обработки, хранения, циркуляции данных, которые обеспечивают функциональность КС и безопасность информации (ее конфиденциальность, целостность и доступность)

Доступ к информации (данным)

-действия субъектов на объектами КС, вызывающие одно-двунаправленные информационные потоки

Методы доступы

-виды действий (операций) субъектов над объектами КС (чтение/просмотр, запись/модификация/добавление, удаление, создание,

Права доступа

-методы доступа (действия, операции), которыми обладают (наделяются, способны выполнять) субъекты над объектами КС

Политика (правила) разграничения доступа

-совокупность руководящих принципов и правил наделения субъектов КС правами доступа к объектам, а также правил и механизмов осуществления самих доступов и реализации информационных потоков

1. Общая характеристика политики дискреционного доступа

Виды политик (правил, механизмов) разграничения доступа

Политика дискреционного разграничения доступа

-разграничение доступа на основе *непосредственного* и *явного предоставления* субъектам *прав* доступа к объектам в виде троек «субъект-операция-объект»

Политика мандатного разграничения доступа

-предоставление прав доступа субъектов к объектам *неявным образом* посредством присвоения уровней (меток) безопасности объектам (*гриф конфиденциальности, уровень целостности*), субъектам (*уровень допуска/полномочий*) и организация доступа на основе соотношения «уровень безопасности субъекта-операция-уровень безопасности объекта»

Политика тематического разграничения доступа

-предоставление прав доступа субъектам к объектам *неявным образом* посредством присвоения тематических категорий объектам (*тематические индексы*) и субъектам (*тематические полномочия*) и организация доступа на основе соотношения «тематическая категория субъекта-операция-тематическая категория объекта»

Политика ролевого разграничения доступа

-агрегирование прав доступа к объектам в именованные совокупности (роли), имеющие определенный функционально-технологический смысл в предметной области КС, и наделение пользователей правом работы в КС в соответствующих ролях

Политика временного разграничения доступа

-предоставление пользователям прав работы в КС по определенному временному регламенту (по времени и длительность доступа)

-предоставление пользователям прав работы в КС по определенному временному регламенту (по времени и длительность доступа)

Политика маршрутного доступа

-предоставление пользователям прав работы в КС при доступе по определенному маршруту (*с определенных рабочих станций*)

Общая характеристика политики дискреционного доступа

- множество легальных (неопасных) доступов P_L задается явным образом внешним по отношению к системе фактором в виде указания дискретного набора троек "субъект-поток(операция)-объект";

-права доступа предоставляются (*«прописываются» в специальных информационных объектах-стуктурах, ассоциированных с монитором безопасности*), отдельно каждому пользователю к тем объектам, которые ему необходимы для работы в КС;

-при запросе субъекта на доступ к объекту монитор безопасности, обращаясь к ассоциированным с ним информационным объектам, в которых «прописана» политика разграничения доступа, определяет «легальность» запрашиваемого доступа и разрешает/отвергает доступ

Модели и механизмы реализации дискреционного разграничения доступа

Различаются:

-в зависимости от принципов и механизмов программно-информационной структуры объекта(объектов), ассоциированных с монитором безопасности, в которых хранятся «прописанные» права доступа (тройки доступа)

-в зависимости от принципа управления правами доступа, т.е. в зависимости от того – кто и как заполняет/изменяет ячейки матрицы доступа (принудительный и добровольный принцип управления доступом)

Выделяют:

-теоретико-множественные (реляционные) модели разграничения доступа (пятимерное пространство Хартсона, модели на основе матрицы доступа)

-модели распространения прав доступа (модель Харисона-Рузо-Ульмана, модель типизованной матрицы доступа, теоретико-графовая модель TAKE-GRANT)

2. Пятимерное пространство Хартсона

Элементы множества A - a_{ijkl} специфицируют:

Система защиты -пятимерное пространство на основе следующих множеств:

U - множество пользователей;

R - множество ресурсов;

E - множество операций над ресурсами;

S - множество состояний системы;

A - множество установленных полномочий.

- ресурсы
- вхождение пользователей в группы;
- разрешенные операции для групп по отношению к ресурсам;

Декартово произведение $A \times U \times E \times R \times S$ - область безопасного доступа

Запрос пользователя на доступ представляет собой *4-х мерный кортеж*:
$q = (u,e,R',s)$, где R' - требуемый набор ресурсов

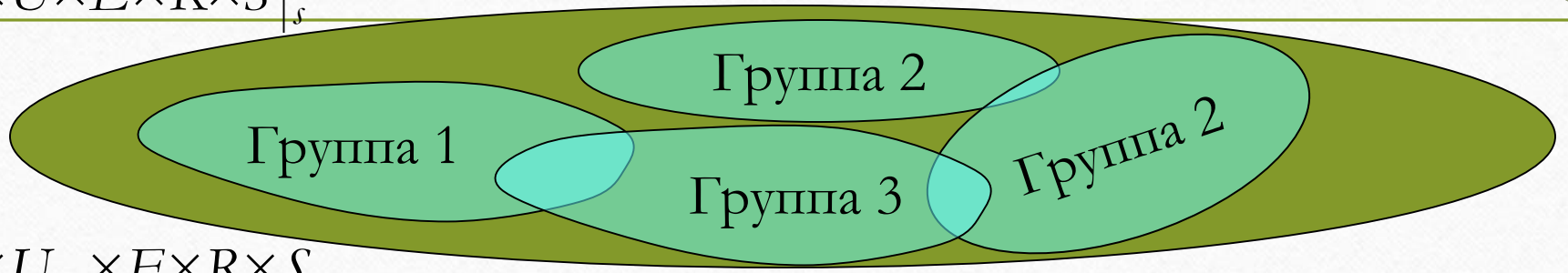
Процесс организации доступа по запросу осуществляется по следующему алгоритму:

*1.*Вызвать все вспомогательные программы для предварительного принятия решения

*2.*Определить те группы пользователей, в которые входит *u*, и выбрать из *A* те спецификации полномочий $P=F(u)$, которым соответствуют выделенные группы пользователей. Набор полномочий $P=F(u)$ определяет т.н.привилегию пользователя

2. Пятимерное пространство Хартсона

$A \times U \times E \times R \times S_{|s}$

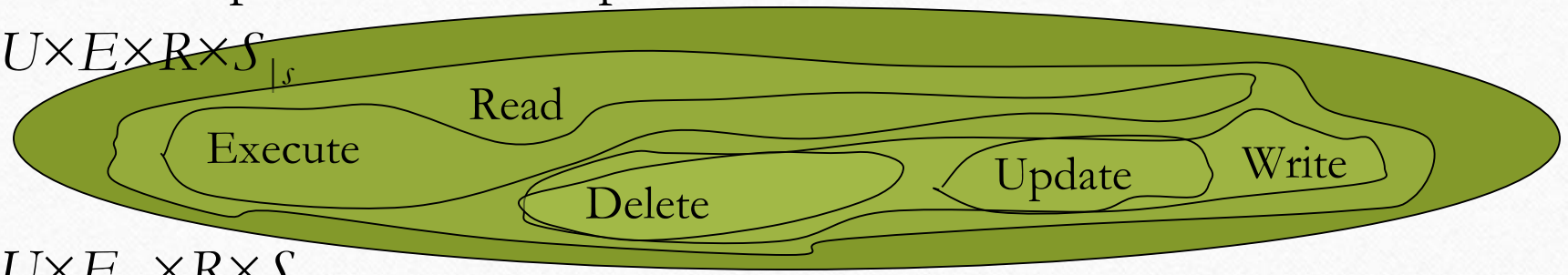


$A \times U_{|u} \times E \times R \times S$

$_{|s}$

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

$A \times U \times E \times R \times S_{|s}$



$A \times U \times E_{|e} \times R \times S$

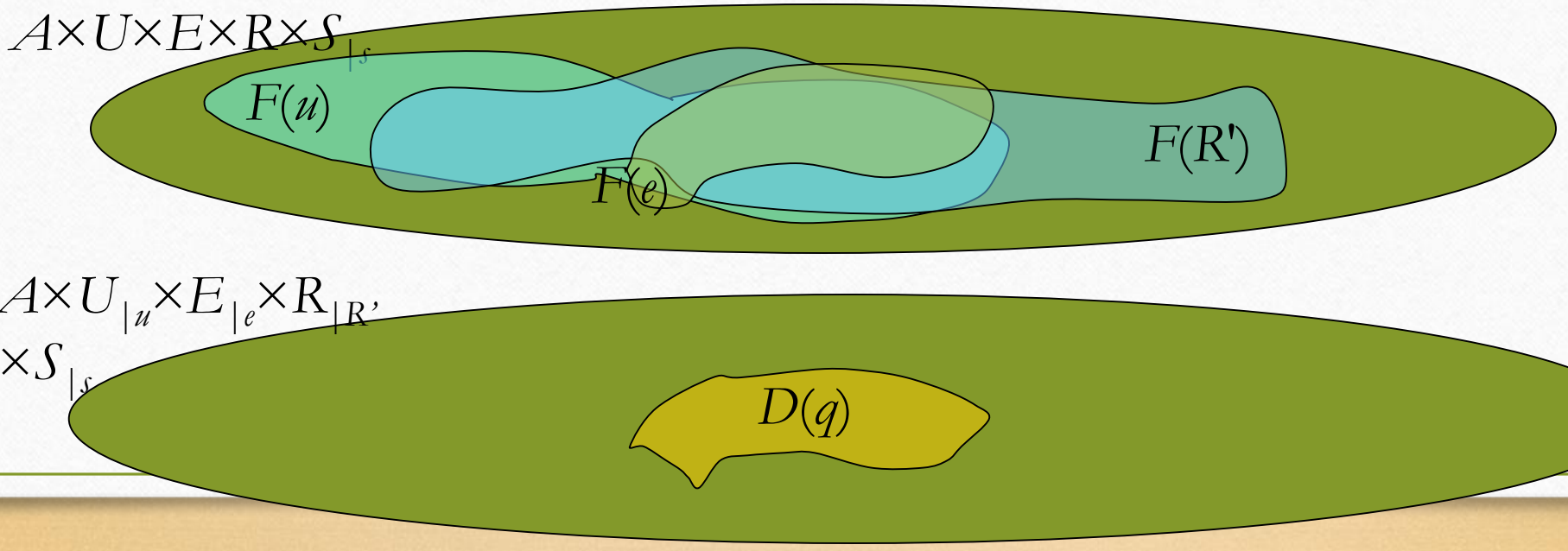
$_{|s}$

2. Пятимерное пространство Хартсона

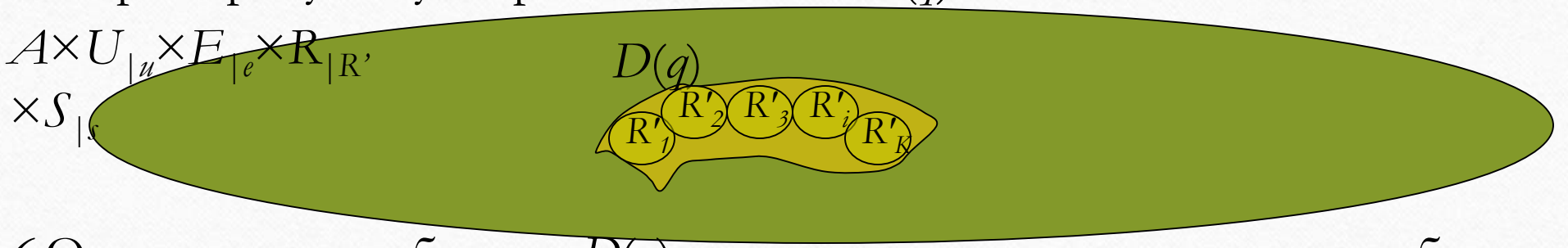
4.Определить из множества *A* набор полномочий $P=F(R')$, разрешающих доступ к набору ресурсов *R′*. Набор полномочий $P=F(R')$ определяет привилегию ресурсов.

На основе $P=F(u)$, $P=F(e)$ и $P=F(R')$ образуется т.н. домен полномочий запроса:

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



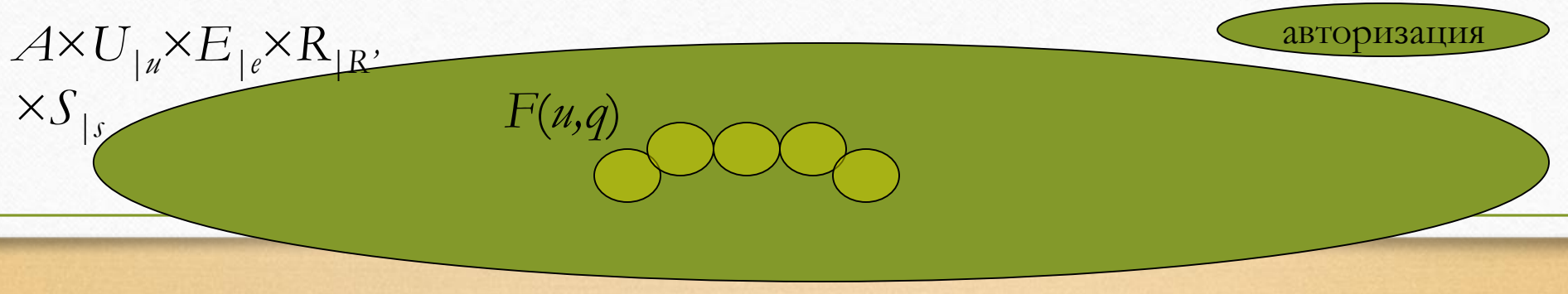
5.Убедиться, что запрашиваемый набор ресурсов R' полностью содержится в домене запроса $D(q)$, т.е. любой r из набора R' хотя бы один раз присутствует среди элементов $D(q)$.



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

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

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



2. Пятимерное пространство Хартсона

7.Вычислить условие фактического доступа (*EAC*), соответствующее запросу *q* , через операции логического ИЛИ по элементам полномочий *F*(*u*,*q*) и запрашиваемым ресурсам *r* из набора *R*′, и получить тем самым набор *R*″ - набор фактически доступных по запросу ресурсов

8.Оценить *EAC* и принять решение о доступе:

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

9.Произвести запись необходимых событий

10.Вызвать все программы, необходимые для организации доступа после "принятия решения".

11.Выполнить все вспомогательные программы, вытекающие для каждого случая по п.8.

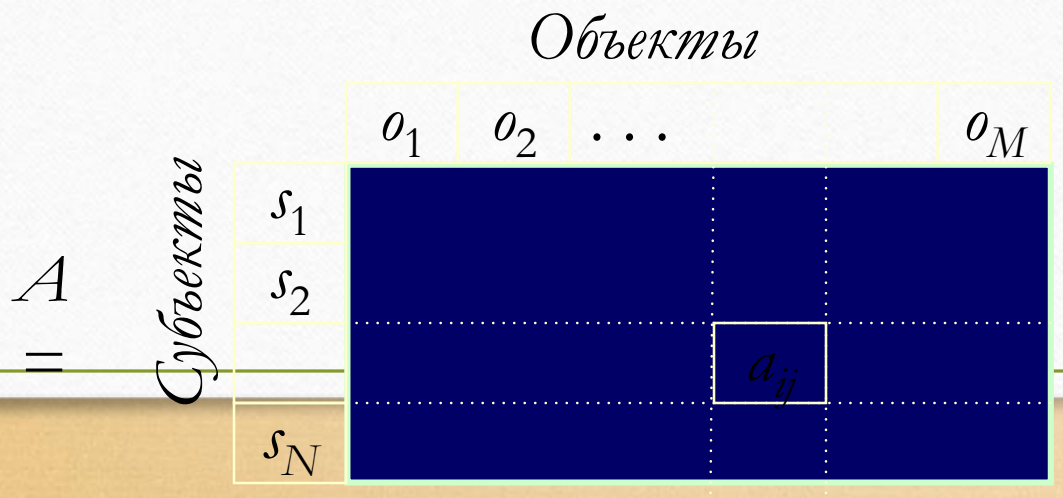
12.При положительном решении о доступе завершить физическую обработку.

Но!!! Безопасность системы в строгом смысле не доказана

3. Модели на основе матрицы доступа

Система защиты -совокупность следующих множеств:

- множество исходных объектов O $(o_1,o_2,\ldots,o_M)$
- множество исходных субъектов S $(s_1,s_2,\ldots,s_N)$, при этом $S \subseteq O$
- множество операций (действий) над объектами Op $(Op_1, Op_2, \ldots, Op_L)$
- множество прав, которые м.б. даны субъектам по отношению к объектам R $(r_1,r_2,\ldots,r_K)$ – т.н. "общие права"
- NxM матрица доступа A, в которой каждому *субъекту* соответствует *строка*, а каждому *объекту* - *столбец*. В ячейках матрицы располагаются права r соотв.субъекта над соотв. объектом в виде набора разрешенных операций Op_i



$A[s_i,oj]= a_{ij}$ - право r из R (т.е. не общее, а конкр. право)

Каждый элемент прав r_k специфицирует совокупность операций над объектом

$r_k \sim (Op_{1k}, Op_{2k}, \ldots, Op_{Jk})$

3. Модели на основе матрицы доступа

Две разновидности моделей в зависимости от того, каким образом заполняются ячейки матрицы доступа *A*. Выделяют:

- *системы с принудительным управлением доступа;*
- *системы с добровольным управлением доступом.*

Принудительное управление доступом

Жесткость, но и более четкий контроль

- вводится т.н.доверенный субъект (администратор доступа), который и определяет доступ субъектов к объектам (централизованный принцип управления)
- в таких системах заполнять и изменять ячейки матрицы доступа может только администратор

Добровольное управление доступом

Гибкость, но и сложность контроля

- вводится т.н. владение (владельцы) объектами и доступ субъектов к объекту определяется по усмотрению владельца (децентрализованный принцип управления)
- в таких системах субъекты посредством запросов могут изменять состояние матрицы доступа

Способы организации информационной структуры матрицы доступа

Централизованная единая информационная структура

Децентрализованная распределенная информационная структура

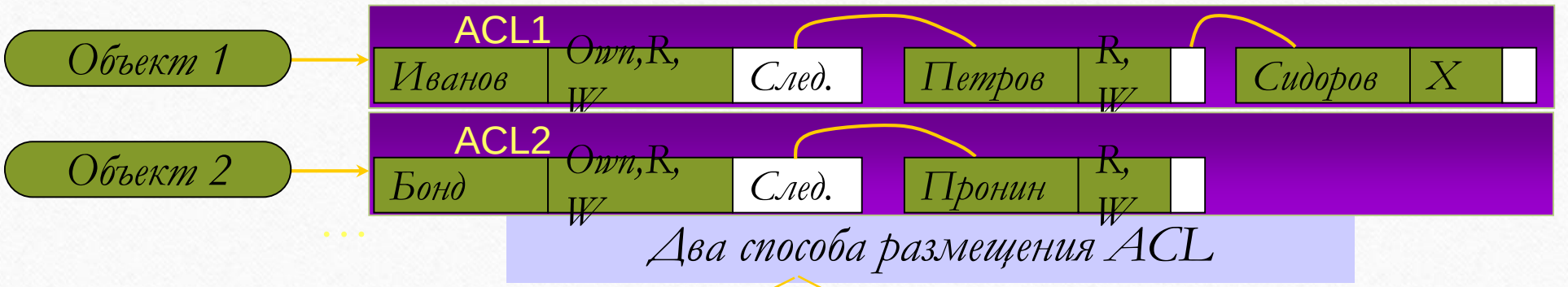
СУБД

системная таблица с назначениями доступа

Биты защиты (UNIX)

	Владелец			*Группа*			*Остальные польз-ли*		
	Чтение	*Запись*	*Выполн*	*Чтение*	*Запись*	*Выполн*	*Чтение*	*Запись*	*Выполн*
Биты	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Объект1	1	1	1	1	0	1	0	0	0
Объект2	1	1	0	1	1	0	1	0	0
…									

Списки доступа в файловой системе ОС Windows (Access Control List – ACL)



Объекты д.б. зарегистрированы в системе

Д.б. обеспечен контроль целостности ACL

Структура списков доступа на примере NTFS

С каждым объектом NTFS связан т.н. дескриптор защиты, состоящий из:

ID влад.	ID перв. гр. влад.	DACL	SACL

Список дискр. контроля доступа

Список контроля доступа

DACL – последовательность произв. кол-ва элементов контроля доступа – ACE, вида:

Allowed / Denied	ID субъекта (польз., группа)	Права доступа (отображ-е)	Флаги, атрибуты

SACL – данные для генерации сообщений аудита

Наиболее типичный представитель систем с добровольным управлением доступом - *модель Харрисона-Руззо-Ульмана*

Разработана для исследования дискреционной политики

В модели *Харрисона-Руззо-Ульмана* помимо элементарных опе-раций доступа *Read*, *Write* и т.д., вводятся также т.н. прими-тивные операции Op_k по *изменению* субъектами матрицы доступа:

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

Состояние системы *Q* изменяется при выполнении команд $C(\alpha_1, \alpha_2, \ldots)$, изменяющих состояние матрицы доступа *A*. Команды инициируются пользователями-субъектами

Структура команды

Название
[Условия] (необяз.)
Операции

```
Command α(x1,…xk)
    if r1 in A[s1,o1] and r2 in A[s2,o2] …
      then; Op2 ; …;
end
```

x_i – идентификаторы задействованных субъектов или объектов

Команды с одной операцией – монооперационные, с одним условием - моноусловные

Примеры команд -

Command "создать файл"(s, f):
  Create object f ;
  Enter "own" into (s, f) ;
  Enter "read" into (s, f) ;
  Enter "write" into (s, f) ;
end

Command «ввести право чтения"(s,s',f):
  if $own \subseteq (s, f)$;
  then
    Enter r "read" into (s', f) ;
end

A	o_1	…	o_M	A	o_1	…	o_M	A	o_1	…	o_M
s_1				s_1				s_1			
⋮				⋮				⋮			
s			own	s			own	s			own
							r, w				r
⋮				⋮				⋮			
s				s				s			
N				N				N			

Основной критерий безопасности - Состояние системы с начальной конфигурацией Q_0 безопасно по праву r, если не существует (при определенном наборе команд и условий их выполнения) последовательности запросов к системе, которая приводит к записи права r в ранее его не содержащую ячейку матрицы $A[s,o]$

Формулировка проблемы безопасности для модели Харрисона-Руззо-Ульмана:

Существует ли какое-либо достижимое состояние, в котором конкретный субъект обладает конкретным правом доступа к конкретному объекту? (т.е. всегда ли возможно построить такую последовательность запросов при некоторой исходной конфигурации когда изначально субъект этим правом не обладает?)

Харрисон, Руззо и Ульман показали :

Теорема 1. Проблема безопасности разрешима для *моно-операционных* систем, т.е. для систем, в которых запросы содержат лишь одну примитивную операцию

Док-во на основе моделирования системы машиной Тьюринга

Теорема 2. Проблема безопасности неразрешима в общем случае

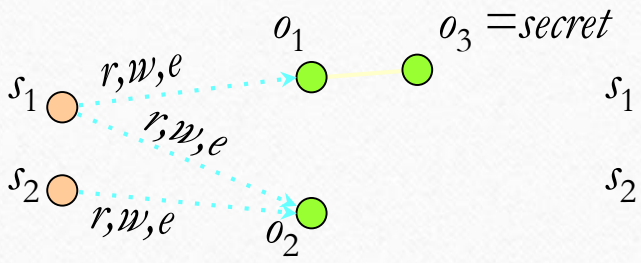
Выводы по модели Харрисона-Руззо-Ульмана:

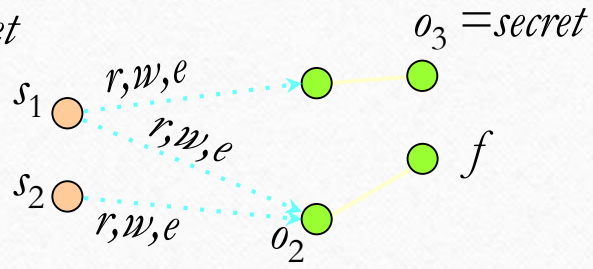
-данная модель в ее полном виде позволяет реализовать множество политик безопасности, но при этом проблема безопасности становится неразрешимой

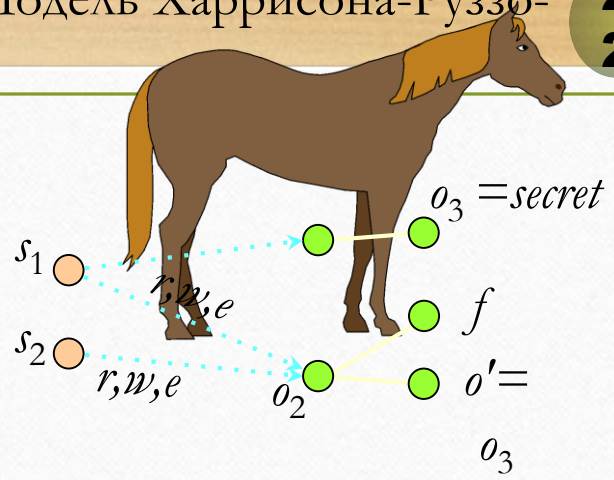
-разрешимость проблемы безопасности только для монооперационных систем приводит к слабости такой модели для реализации большинства политик безопасности (т.к. нет операции автоматического наделения своими правами дочерних объектов, ввиду чего по правам доступа они изначально не различимы)

4.Модели распространения прав доступа. 4.1.Модель Харрисона-Руззо-Ульмана (модель HRU)

Проблема «троянских» программ



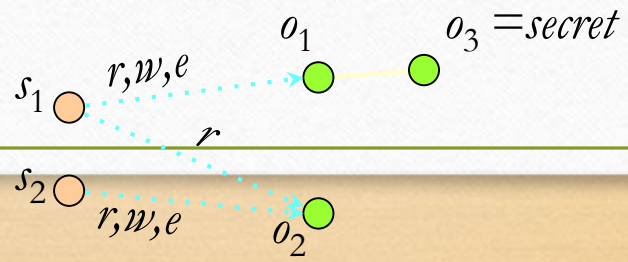




Command "создать файл“ (s_2,f):
if *write* ∈ [s_2, o_2] ;
then
Create object f ;
Enter "read" into [s_2, *f*] ;
Enter "write" into [s_2, *f*] ;
Enter "execute" into [s_2, *f*];
if *read* ∈ [s_1, o_2] ;
then
Enter "read" into [s_1, *f*] ;
if *write* ∈ [s_1, o_2] ;
then
Enter "write" into [s_1, *f*] ;
if *execute* ∈ [s_1, o_2] ;
then
Enter "execute" into [s_1, *f*] ;
end

Command “запустить файл"(s_1, *f*):
if *execute* ∈ [s_1, *f*] ;
then
Create subject f' ;
Enter "read“ into [*f'*,o_1];
Enter "read" into [*f'*,o_3];
if *write* ∈ [s_1,o_2] ;
then
Enter “write“ into [*f'*,o_2];
end

Command “скопировать файл o_3 программой *f'* в o_2“ (*f'*,o_3, o_2):
if *read* ∈ [*f'*, o_3] and
write ∈ [*f'*, o_2]
then
Create object o';
Write (*f'*, o_3 , *o'*);
if *read* ∈ [s_2, o_2] ;
then
Enter "read" into [s_2,*o'*];
end



Расширения модели HRU

Типизованная матрица доступа (Модель TAM) R. Sandhu,1992г.

Вводится фиксированное количество типов τ_k (например, "user"- пользователь, 'so"-офицер безопасности и "file"), которым могут соответствовать сущности КС (субъекты и объекты).

Command $\alpha(x_1{:}\tau_1, x_2{:}\tau_2, \ldots, x_k{:}\tau_k)$

Накладываются ограничения на условия и соответствие типов в монотонных операциях (порождающие сущности)

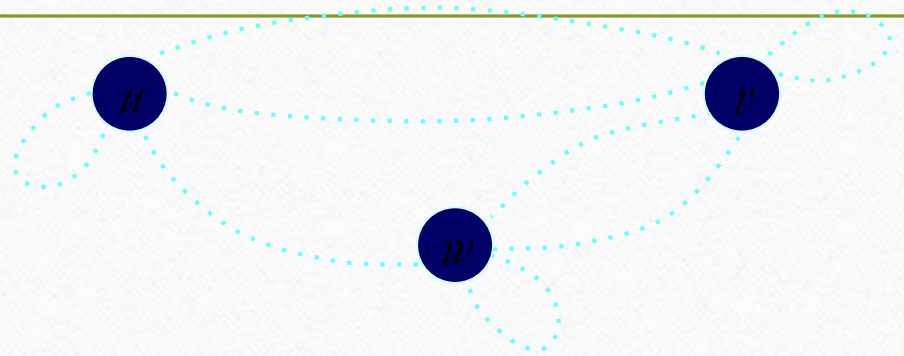
Смягчаются условия на разрешимость проблемы безопасности

Анализ проблем безопасности в модели TAM основывается на понятии родительских и дочерних типов

Определение 1. *Тип τ_k является дочерним типом в команде создания $\alpha(x_1{:}\tau_1, x_2{:}\tau_2, \ldots, x_k{:}\tau_k)$, если и только если имеет место один из следующих элементарных операторов: "Create subject x_k of type τ_k" или "Create object x_k of type τ_k". В противном случае тип τ_k является родительским типом.*

Вводится Граф отношений наследственности

Пусть имеется
три типа *u*, *v*, *w*



Функционирование системы осуществляется через последовательность следующих команд:

0-й шаг – в системе имеется субъект типа *u* - $(s_1{:}u)$

1-й шаг. $\alpha(s_1{:}u, s_2{:}w, o_1{:}v)$:
Create object o_1 *of type* v ;
Inter r *into* $[s_1, o_1]$;
Create subject s_2 *of type* w ;
Inter r' *into* $[s_2, o_1]$;
end

2-й шаг. $\alpha(s_3{:}u, o_1{:}v)$:
Create subject s_3 *of type* u ;
Inter r'' *into* $[s_3, o_1]$;
end

v – дочерний тип в команде α, в теле которой имеются еще типы *u*, *w*. Т.о. в Графе отношений наследственности возникают дуги (*u*,*v*), (*w*,*v*) и в т.ч. (*v*,*v*)

w – дочерний тип в команде α, в теле которой имеются еще типы *u*, *v*. Т.о. в Графе отношений наследственности возникают дуги (*u*,*w*), (*v*,*w*) и в т.ч. (*w*,*w*)

u – дочерний тип в команде α, в теле которой имеются еще тип *v*. Т.о. возникают дуги (*v*,*u*) и в т.ч. (*u*,*u*)

4.Модели распространения прав доступа. **4.2.**Модель типизованной матрицы доступа (модель TAM)

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

Определение 2. *Реализация MTAM является ациклической тогда и только тогда, когда ее граф отношений наследственности не содержит циклов*

Теорема 3. Проблема безопасности разрешима для *ациклических реализаций MTAM*

Джонс, Липтон, Шнайдер, 1976г.

Теоретико-графовая модель анализа распространения прав доступа в дискреционных системах на основе матрицы доступа

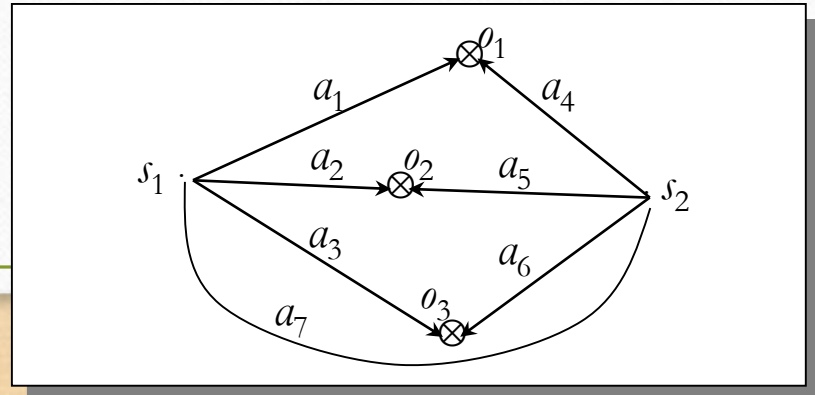
1.Также как и в модели HRU система защиты представляет совокупность следующих множеств:

- множество исходных объектов O $(o_1, o_2, \ldots, o_M)$

- множество исходных субъектов S $(s_1, s_2, \ldots, s_N)$, при этом $S \subseteq O$

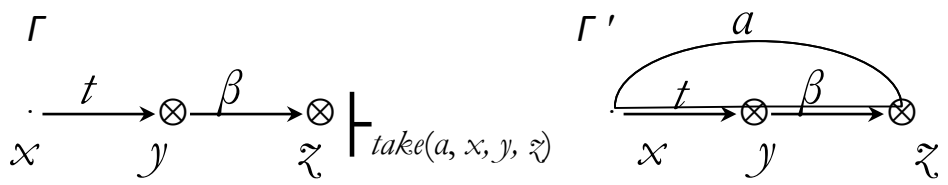
- множество прав, которые м.б. даны субъектам по отношению к объектам $(r_1, r_2, \ldots, r_K) \cup \{t, g\}$, в том числе с двумя специфическими правами – правом *take* (t – *право брать права доступа у какого-либо объекта по отношению к другому объекту*) и правом *grant* (g – *право предоставлять права доступа к определенному объекту другому субъекту*)

- множеством E установленных прав доступа (x, y, α) субъекта x к объекту y с правом α из конечного набора прав. При этом состояние системы представляется Графом доступов Γ



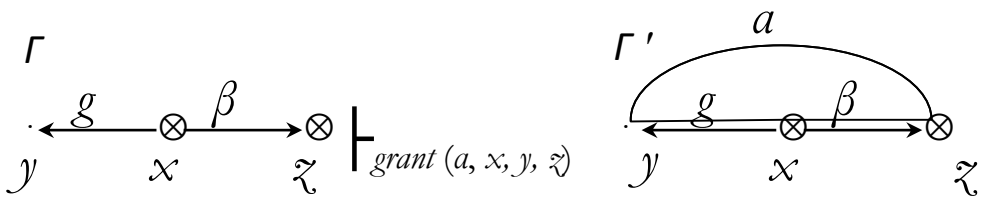
2.Состояние системы (Графа доступов) изменяется под воздействием элементарных команд 4-х видов

Команда "*Брать*" – *take*(*a*, *x*, *y*, *z*)



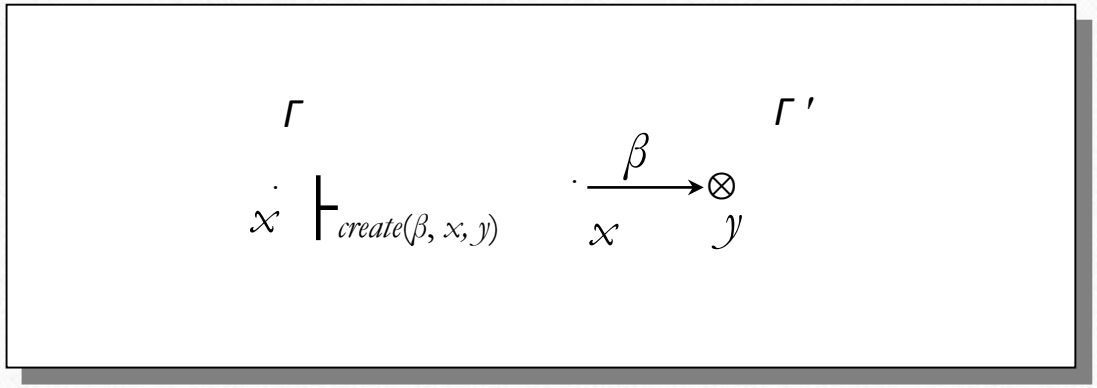
субъект x берет права доступа $a \subseteq \beta$ на объект z у объекта y (обозначения: $\vdash_c$ – переход графа Γ в новое состояние Γ' по команде c ; $x \in S$; $y, z \in O$)

Команда «*Давать*" – *grant*(*a*, *x*, *y*, *z*)



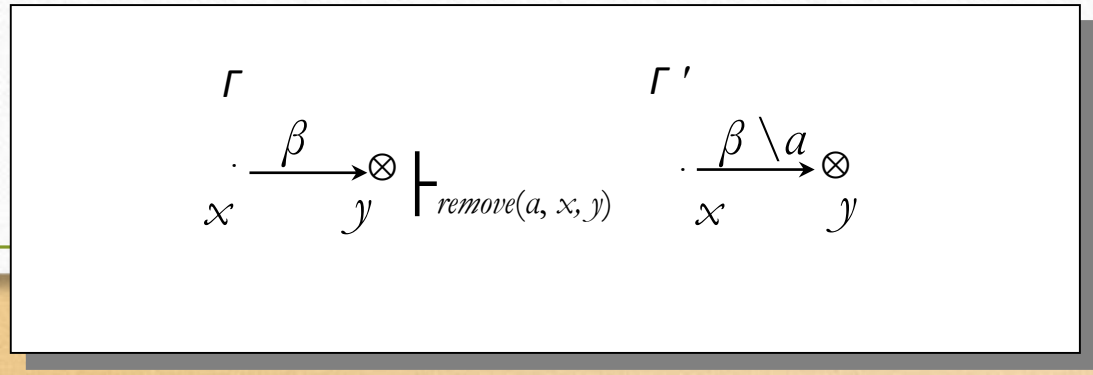
субъект x дает объекту y право $a \subseteq \beta$ на доступ к объекту z

Команда "*Создать*" – *create*(β, x, y)



субъект x создает объект y с правами доступа на него $\beta \subseteq R$ (y – новый объект, $O'=O \cup \{y\}$), в т. ч. с правами t, или g, или $\{t, g\}$.

Команда «*Изъять*" – *remove*(a, x, y)



субъект x удаляет права доступа $a \subseteq \beta$ на объект y

3. Безопасность системы рассматривается с точки зрения возможности получения каким-либо субъектом прав доступа к определенному объекту (в начальном состоянии $\Gamma_0\ (O_0, S_0, E_0)$ такие права отсутствуют) при определенной кооперации субъектов путем последовательного изменения состояния системы на основе выполнения элементарных команд. Рассматриваются две ситуации – условия санкционированного, т.е. законного получения прав доступа, и условия «похищения» прав доступа

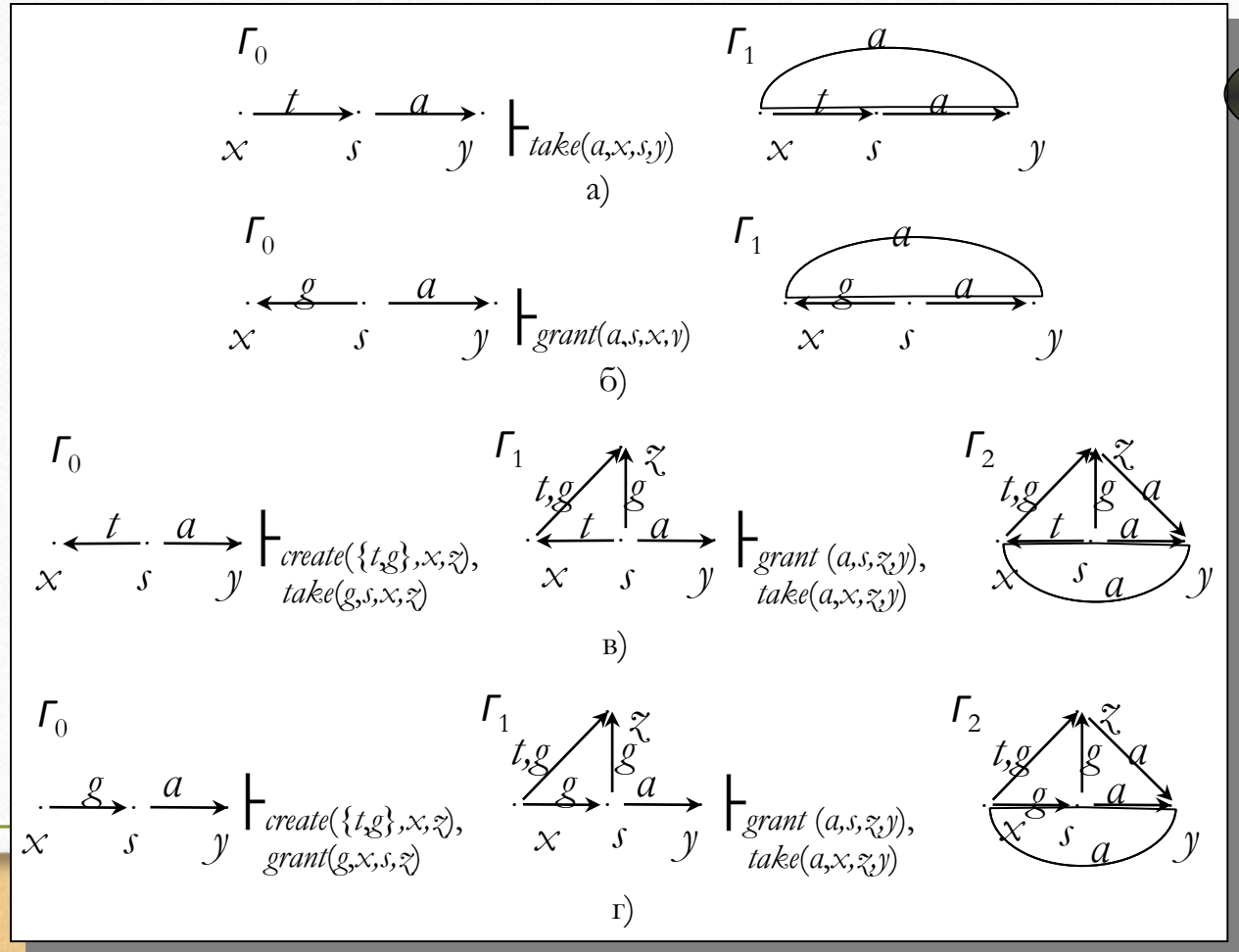
3.1. *Санкционированное получение прав доступа*

Определение 3. *Для исходного состояния системы* $\Gamma_0\ (O_0, S_0, E_0)$ *и прав доступа* $\alpha \subseteq R$ *предикат "возможен доступ*$(\alpha,\ x,\ y, \Gamma_0)$*" является истинным тогда и только тогда, когда существуют графы доступов системы* $\Gamma_1\ (O_1, S_1, E_1)$, $\Gamma_2\ (O_2, S_2, E_2)$, …, $\Gamma_N\ (O_N, S_N, E_N)$, *такие, что:*
$\Gamma_0\ (O_0, S_0, E_0) \vdash c_1\ \Gamma_1\ (O_1, S_1, E_1) \vdash c_2 \ldots \vdash c_N\ \Gamma_N\ (O_N, S_N, E_N)$ *и* $(x, y, \alpha) \in E_N$
где $c_1,\ c_2,\ \ldots,\ c_N$ *– команды переходов*

Определение 4. *Вершины графа доступов являются tg-связными* (*соединены tg-путем*), *если в графе между ними существует такой путь, что каждая дуга этого пути выражает право t или g* (*без учета направления дуг*)

Теорема 4. *В графе доступов* $\Gamma_0 (O_0, S_0, E_0)$ *, содержащем только вершины-субъекты, предикат "возможен доступ*(α, x, y, Γ_0)*" истинен тогда и только тогда, когда выполняются следующие условия:*

- *существуют субъекты* $s_1, \ldots, s_m$ *такие, что* $(s_i, y, \gamma_i) \in E_0$ *для* $i=1, \ldots, m$ *и* $\alpha = \gamma_1 \cup \ldots \cup \gamma_m$.
- *субъект x соединен в графе* Γ_0 *tg-путем с каждым субъектом* s_i *для* $i=1, \ldots, m$



Доказательство

получение прав *a* доступа субъектом *x* у субъекта *s* на объект *y* при различных вариантах непосредственной *tg*-связности

Определение 5. *Островом в произвольном графе доступов* $\Gamma(O, S, E)$ *называется его максимальный tg-связный подграф, состоящий только из вершин субъектов.*

Определение 6. *Мостом в графе доступов* $\Gamma(O, S, E)$ *называется tg–путь, концами которого являются вершины-субъекты; при этом словарная запись tg–пути должна иметь вид*

$$\vec{t}^{\,*}, \overleftarrow{t}^{\,*}, \vec{t}^{\,*}\vec{g}\overleftarrow{t}^{\,*}, \vec{t}^{\,*}\overleftarrow{g}\overleftarrow{t}^{\,*}$$

*где символ * означает многократное (в том числе нулевое) повторение.*

Определение 7. *Начальным пролетом моста в графе доступов* $\Gamma(O, S, E)$ *называется tg-путь, началом которого является вершина-субъект; при этом словарная запись tg-пути должна иметь вид*

$$\vec{t}^{\,*}\vec{g}$$

Определение 8. *Конечным пролетом моста в графе доступов* $\Gamma(O, S, E)$ *называется tg-путь, началом которого является вершина-субъект; при этом словарная запись tg-пути должна иметь вид*

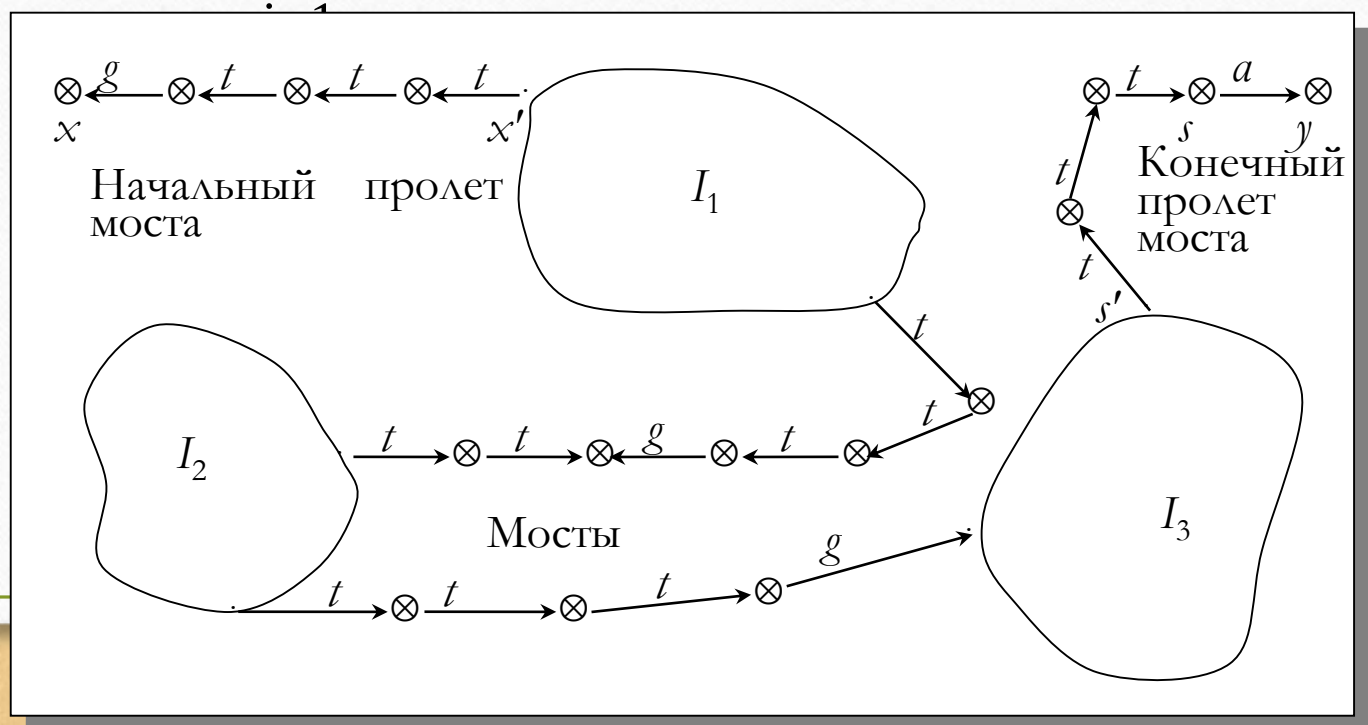
$$\vec{t}^{\,*}$$

Теорема 4. *В произвольном графе доступов* $\Gamma_0\ (O_0, S_0, E_0)$ *предикат "возможен доступ*$(\alpha,\ x,\ y, \Gamma_0)$*" истинен тогда и только тогда, когда выполняются условия*:

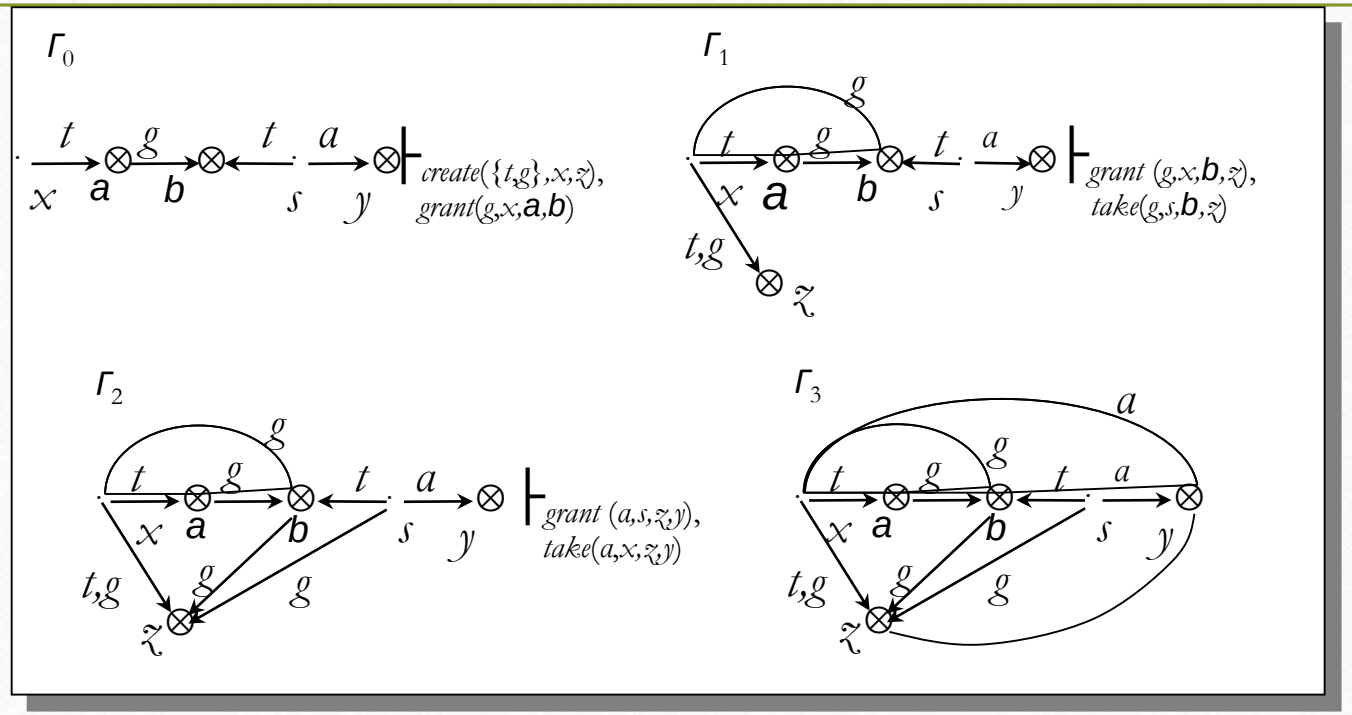
- *существуют объекты* $s_1,\ldots,s_m$ *такие, что* $(s_i, y, \gamma_i)\in E_0$ *для* $i=1,\ \ldots,\ m$ *и* $\alpha = \gamma_1 \cup \ldots \cup \gamma_m$.
- *существуют вершины-субъекты* $x_1',\ldots,x_m'$ *и* $s_1',\ldots,s_m'$ *такие, что*:

-- $x = x_i'$ *или* x_i' *соединен с* x *начальным пролетом моста для* $i=1,\ \ldots,\ m$;

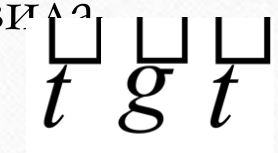
-- $s_i = s_i'$ *или* s_i' *соединен с* s_i *конечным пролетом моста для*



Пример графа доступов с возможностью передачи объекту x прав доступа a на объект y



Пример передачи прав доступа по мосту вида



3.1. *Похищение прав доступа*

Определение 9. *Для исходного состояния системы* $\Gamma_0 (O_0, S_0, E_0)$ *и прав доступа* $\alpha \subseteq R$ *предикат "возможно похищение*(α, x, y, Γ_0)*" является истинным тогда и только тогда, когда существуют графы доступов системы* $\Gamma_1 (O_1, S_1, E_1)$, $\Gamma_2 (O_2, S_2, E_2)$, ..., $\Gamma_N (O_N, S_N, E_N)$ *такие, что:*

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

где $c_1, c_2, \ldots, c_N$ *– команды переходов;*

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

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

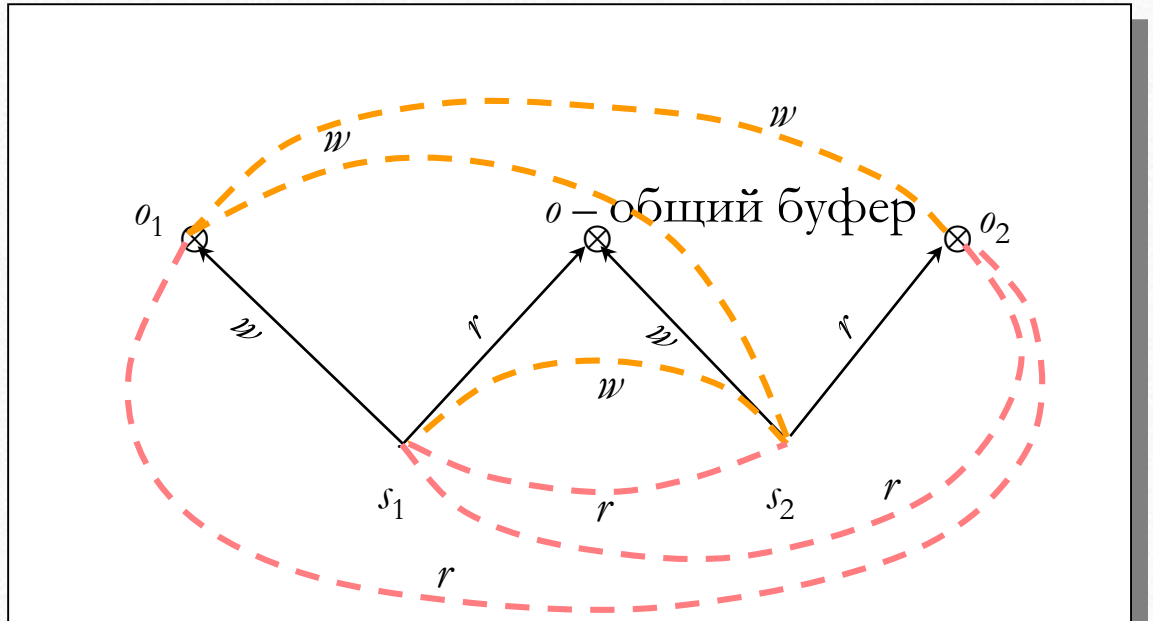
Теорема 4. *В произвольном графе доступов* $\Gamma_0 (O_0, S_0, E_0)$ *предикат "возможно похищение*(α, x, y, Γ_0)*" истинен тогда и только тогда, когда выполняются условия*:

- $(x, y, \alpha) \notin E_0$.
- *существуют субъекты* $s_1, \ldots, s_m$ *такие, что* $(s_i, y, \gamma_i) \in E_0$ *для* $i=1, \ldots, m$ *и* $\alpha = \gamma_1 \cup \ldots \cup \gamma_m$
- *являются истинными предикаты "возможен доступ*(t, x, s_i, Γ_0)*" для* $i=1, \ldots, m$.

Если политика разграничения доступа в КС запрещает субъектам, имеющим в исходном состоянии права доступа к определенным объектам, непосредственно предоставлять эти права другим субъектам, которые изначально такими правами не обладают, то, тем не менее, такие первоначально "обделенные" субъекты могут получить данные права при наличии в графе доступов возможностей получения доступа с правом t к первым субъектам

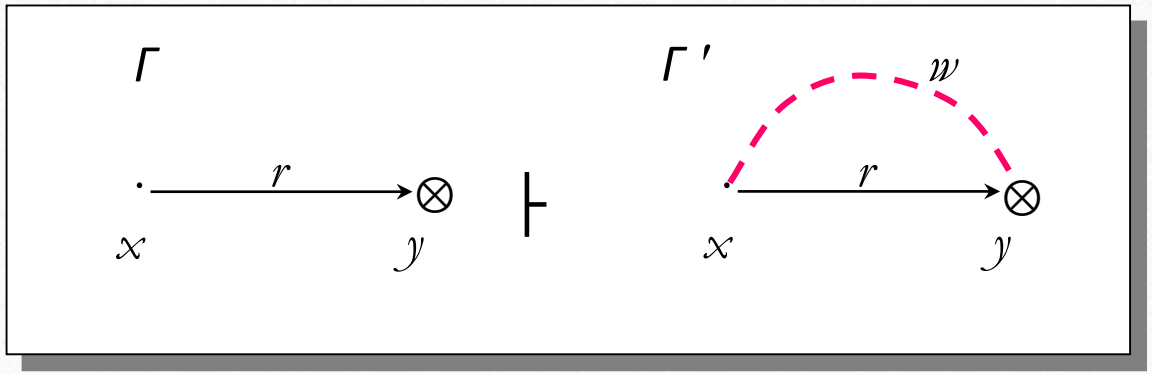
Теоретическая основа для анализа неявных (скрытых) каналов утечки информации в системах с дискреционным доступом

Определение 10. *Неявным информационным потоком между объектами системы называется процесс переноса информации между ними без их непосредственного взаимодействия* (*операции Read, Write*)



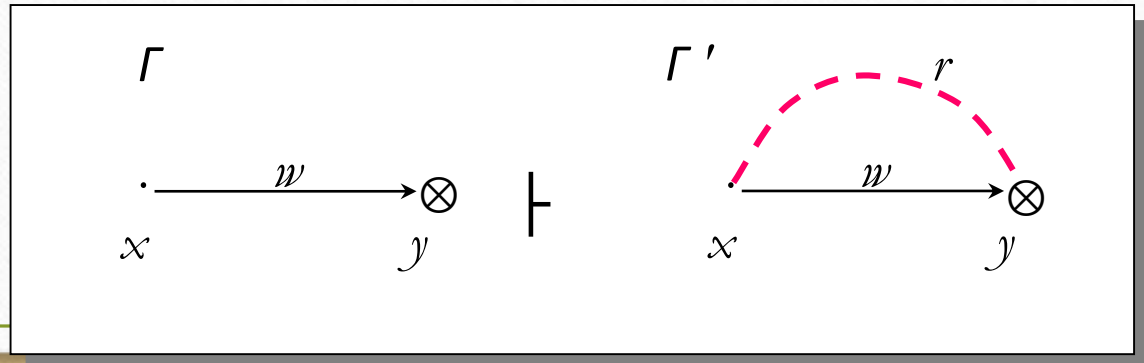
2.Состояние Графа доступов изменяется под воздействием элементарных команд 6-х видов (т.н. команды *де-факто*)

Команда без названия $a_1(x, y)$



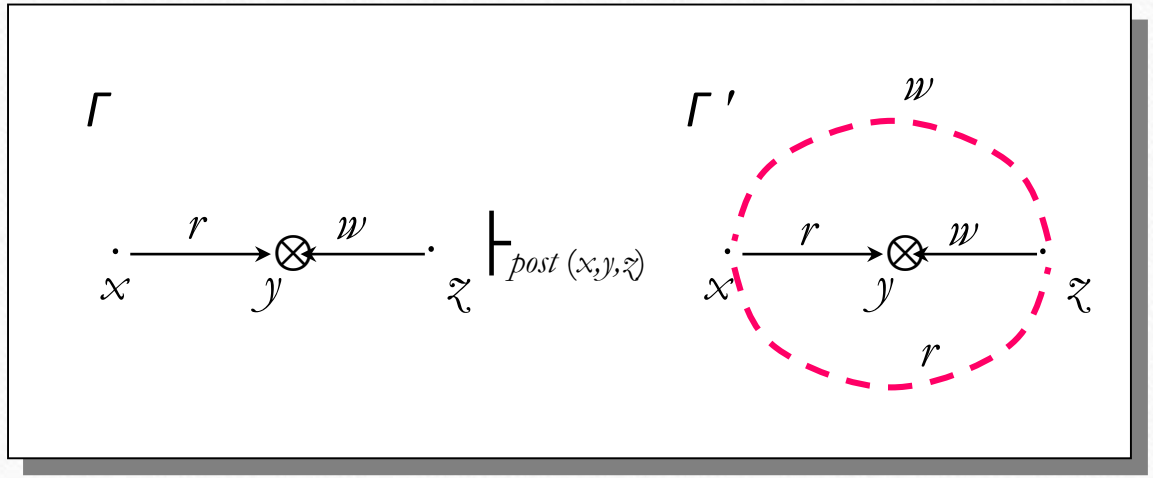
имеется неявная возможность передачи (записи) [конфиденциальной] информации из объекта y субъекту x, когда тот осуществляет доступ r к объекту y

Команда без названия $a_2(x, y)$



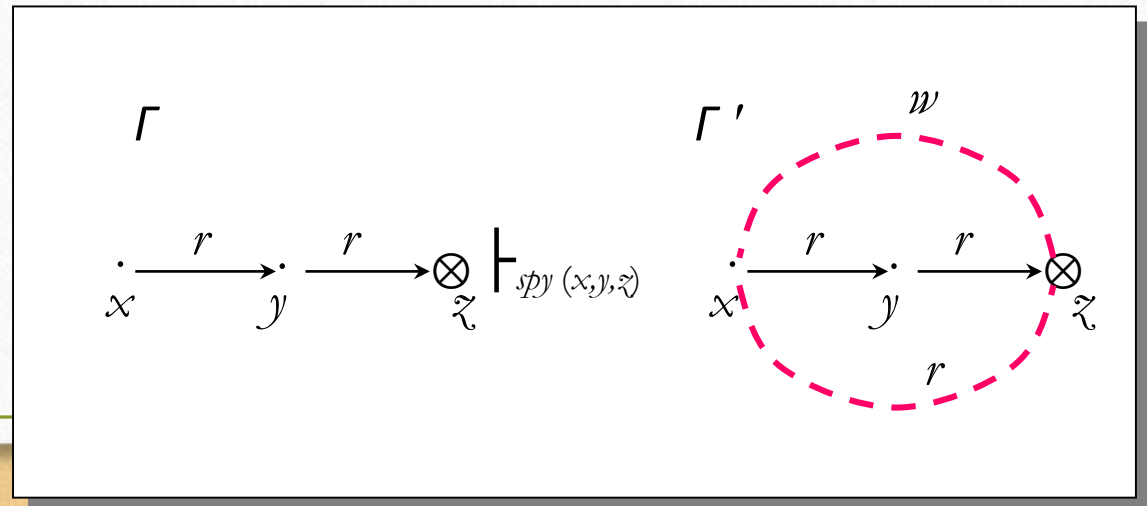
имеется неявная возможность получения (чтения) объектом y [конфиденциальной] информации от субъекта x], когда тот осуществляет доступ w к объекту y

Команда *post*(*x*, *y*, *z*)



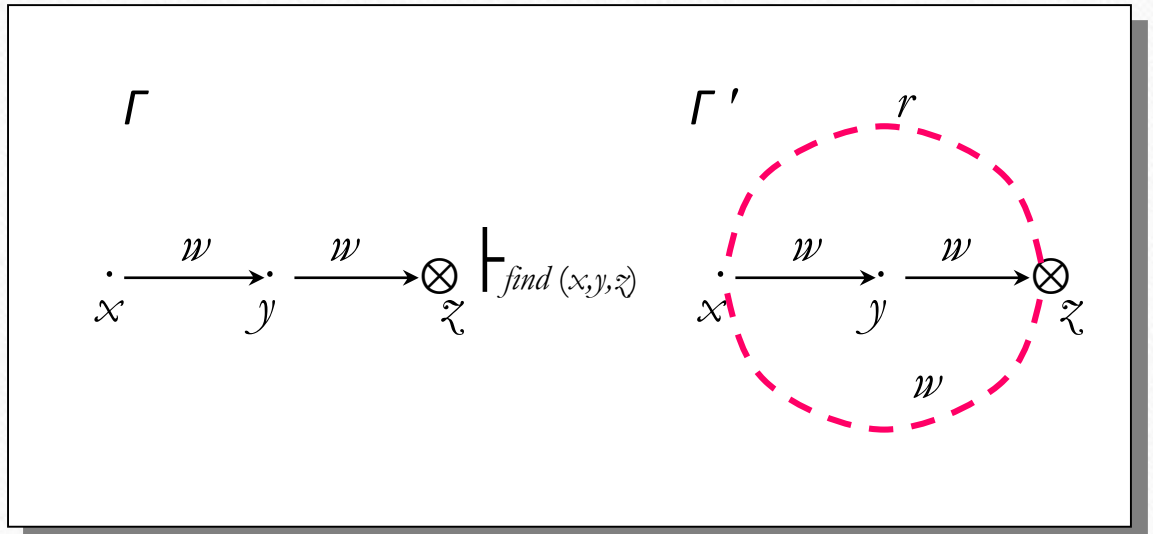
субъект *x* получает возможность чтения информации от (из) другого субъекта *z*, осуществляя доступ *r* к объекту *y*, к которому субъект *z* осуществляет доступ *w*, а субъект *z*, в свою очередь, получает возможность записи своей информации в субъект *x*

Команда *spy*(*x*, *y*, *z*)



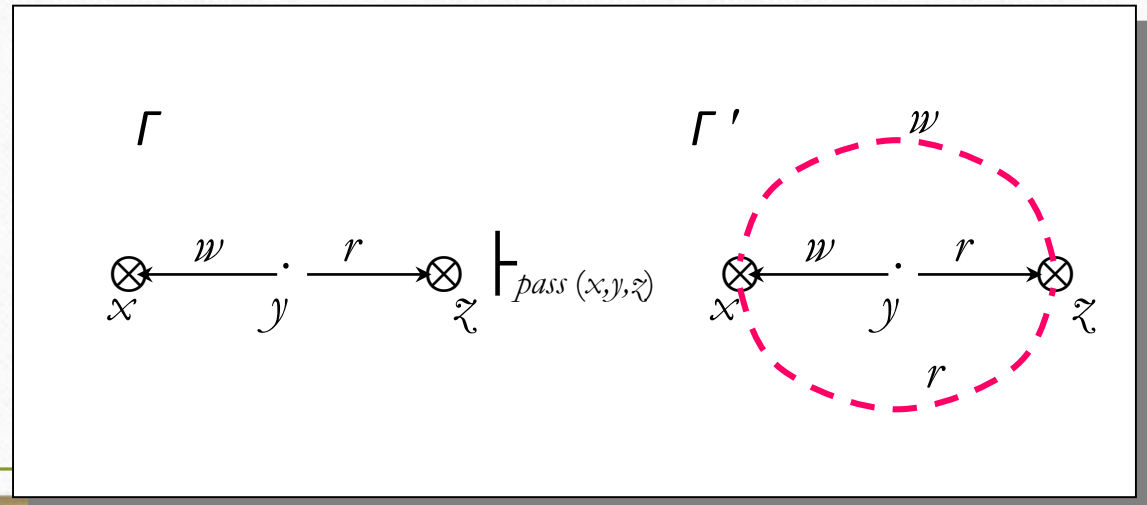
субъект *x* получает возможность чтения информации из объекта *z*, осуществляя доступ *r* к субъекту *y*, который, в свою очередь, осуществляет доступ *r* к объекту *z*, при этом также у субъекта *x* возникает возможность записи к себе информации из объекта *z*

Команда *find*(*x*, *y*, *z*)



субъект *x* получает неявн. возможность передачи (записи) конф. информации в объект *z*, осуществляя доступ *w* к субъекту *y*, который, в свою очередь, осуществляет доступ *w* к объекту *z*, при этом также у субъекта *z* возникает неявн. возможность чтения конф. информации из субъекта *x*

Команда *pass*(*x*, *y*, *z*)



при осуществлении субъектом *y* доступа *r* к объекту *z* возникает неявная возможность внесения из него конф. информации в другой объект *x*, к которому субъект *y* осуществляет дос-туп *w*, и, кроме того, возни-кает возможность получе-ния информации (чтения) объектом *x* из объекта *z*

Правила *де-юре* к мнимым дугам не применяются

3. Анализ возможности возникновения неявного информационного канала (потока) между двумя произвольными объектами (субъектами) *x* и *y* системы осуществляется на основе поиска и построения в графе доступов пути между *x* и *y*, образованного мнимы-ми дугами, порождаемыми применением команд *де-факто* к различным фрагментам исходного Графа доступов

Расширенная модель TAKE-GRANT позволяет анализировать специфические проблемы в дискреционных системах разграничения доступа:

- при допущении возможности или при наличии достоверных фактов о состоявшемся неявном информационном потоке от одного объекта(субъекта) к другому объекту(субъекту), анализировать и выявлять круг возможных субъектов-"заговорщиков" несанкционированного информационного потока

- для какой-либо пары объектов (субъектов) осуществлять анализ не только возможности неявного информационного потока, но и количественных характеристик по тому или иному маршруту:

  возможно взвешивание мнимых дуг на Графе доступов посредством оценки вероятности их возникновения

  возможны количественные сравнения различных вариантов возникновения неявного потока по длине пути на Графе доступов

- оптимизировать систему назначений доступа по критериям минимизации возможных неявных информационных потоков