IS 2150 / TEL 2810 Introduction to Security

James Joshi
Associate Professor, SIS

Lecture 6
September 27, 2011

Take Grant Model

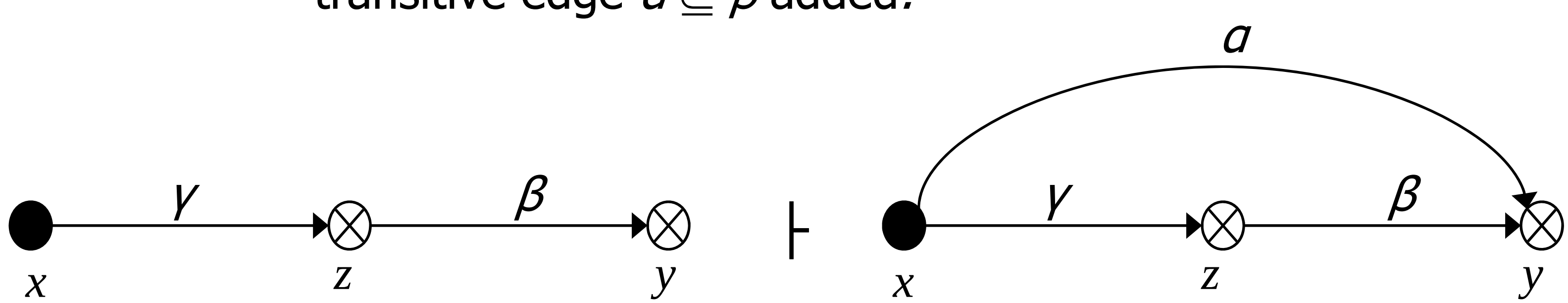
Objective

- Understand Take-Grant model
  - Specific, restricted
- Analyze
  - Right Sharing
  - Stealing/Theft
  - conspiracy

Take-Grant Protection Model

- System is represented as a directed graph
  - Subject: ● Either: ⊗
  - Object: ○
  - Labeled edge indicate the rights that the source object has on the destination object
- Four graph rewriting rules ("de jure", "by law", "by rights")
  - The graph changes as the protection state changes according to

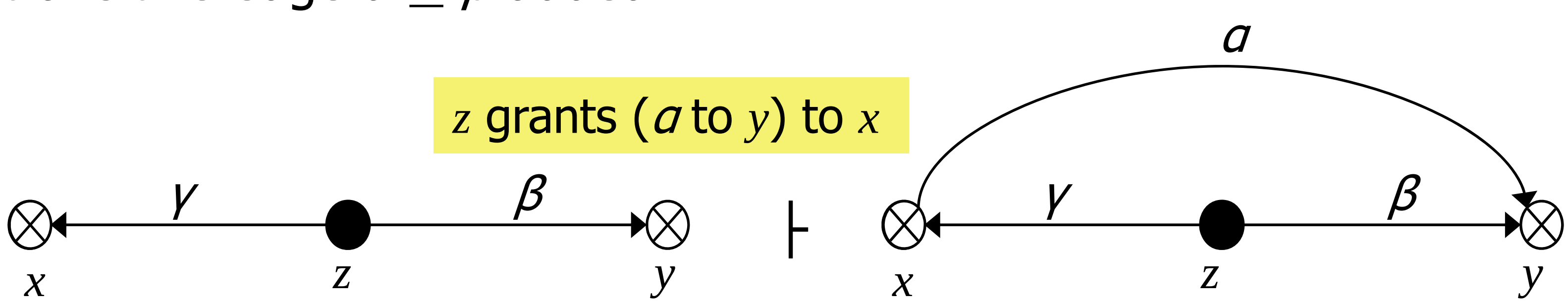
1. Take rule: if $t \in \gamma$, the take rule produces another graph with a transitive edge $\alpha \subseteq \beta$ added.



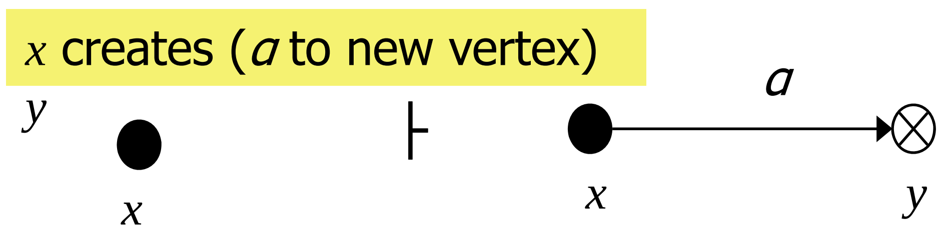
x takes (α to y) from z

Take-Grant Protection Model

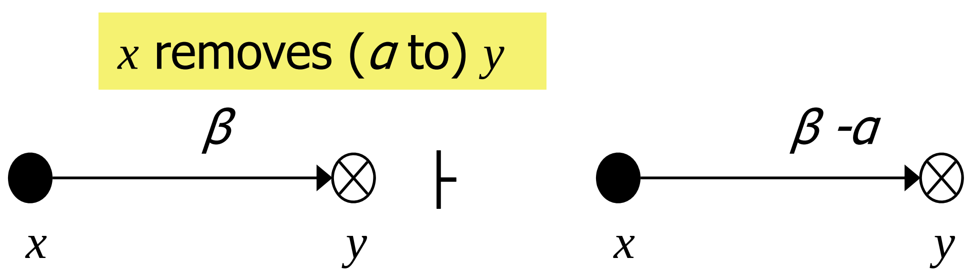
2. Grant rule: if $g \in \gamma$, the take rule produces another graph with a transitive edge $\alpha \subseteq \beta$ *added.*



3. Create rule:



4. Remove rule:

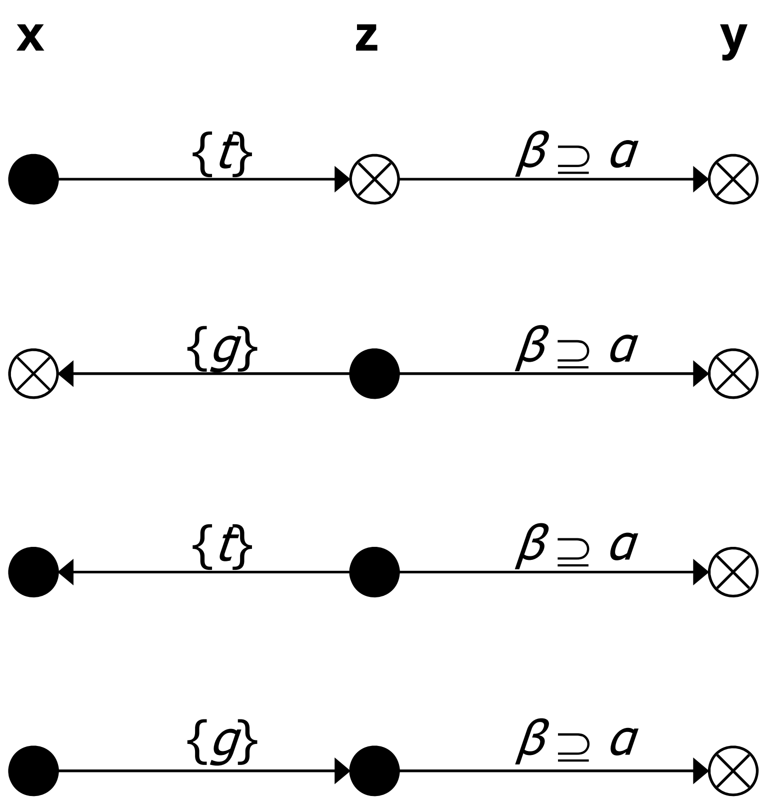


Take-Grant Protection Model: Sharing

- Given G_0, can vertex **x** obtain α rights over **y**?
  - Can_share(α, x, y, G_0) is true iff
    - $G_0 \vdash^* G_n$ using the four rules, &
    - There is an α edge from x to y in G_n
- *tg-path*: $\mathbf{v}_0, \ldots, \mathbf{v}_n$ with t or g edge between any pair of vertices $\mathbf{v}_i$, $\mathbf{v}_{i+1}$
  - Vertices *tg-connected* if *tg-path* between them
- Theorem: Any two subjects with *tg-path* of length 1 can share rights

Any two subjects with *tg-path* of length 1 can share rights

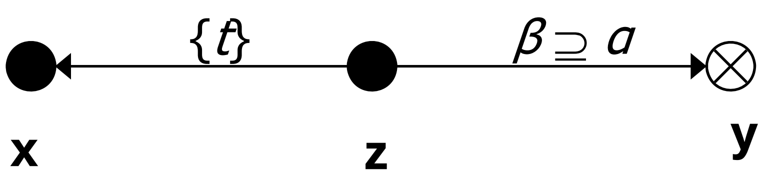
Can_share(α, x, y, G_0)



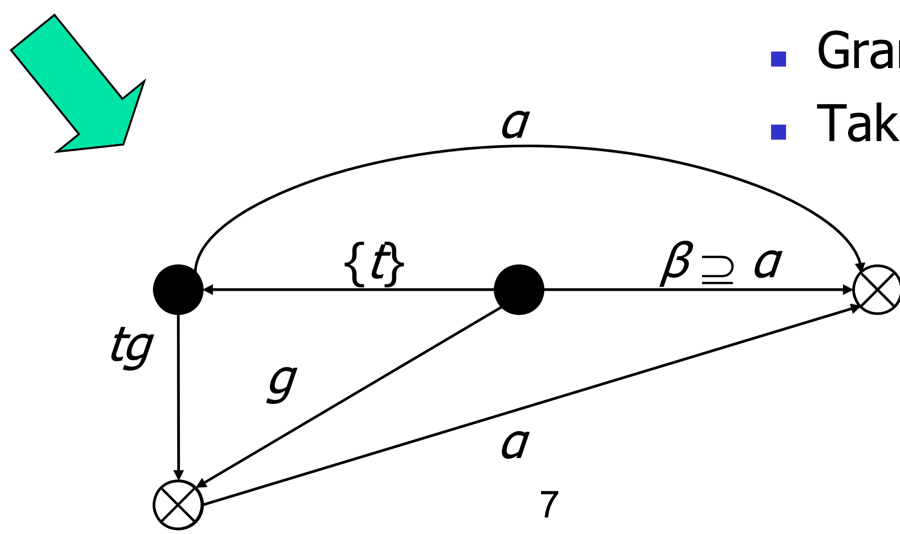
- Four possible length 1 *tg*-paths
  1. Take rule
  2. Grant rule
  3. Lemma 3.1
  4. Lemma 3.2

Any two subjects with *tg-path* of length 1 can share rights

Can_share(α, x, y, G_0)



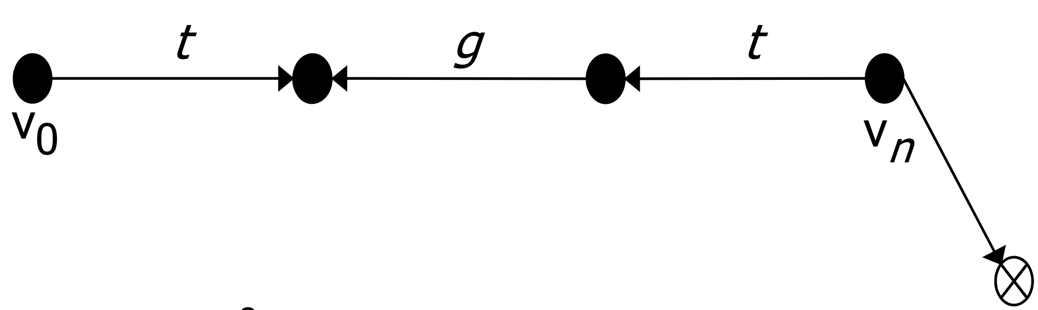
- Lemma 3.1
  - Sequence:
    - Create
    - Take
    - Grant
    - Take



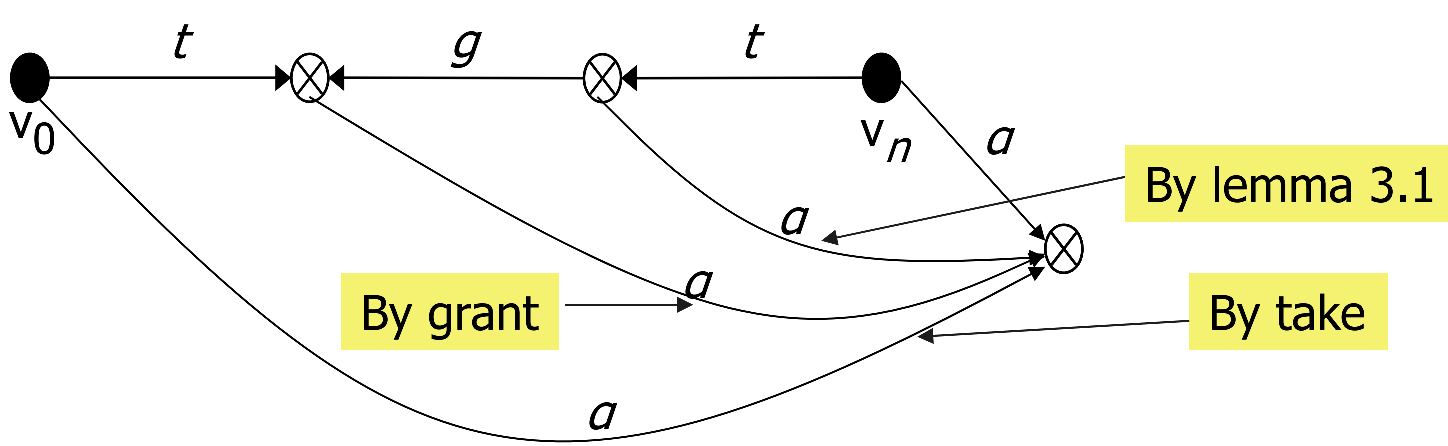
Prove Lemma 3.2

Other definitions

- Island: Maximal *tg*-connected subject-only subgraph
  - Can_share all rights in island
  - Proof: Induction from previous theorem
- Bridge: *tg*-path between subjects v_0 and v_n with edges of the following form:
  - $\vec{t}^{\,*}$,
  - $\overleftarrow{t}^{\,*}$
  - $\vec{t}^{\,*} \vec{g} \overleftarrow{t}^{\,*}$
  - $\vec{t}^{\,*} \overleftarrow{g} \overleftarrow{t}^{\,*}$



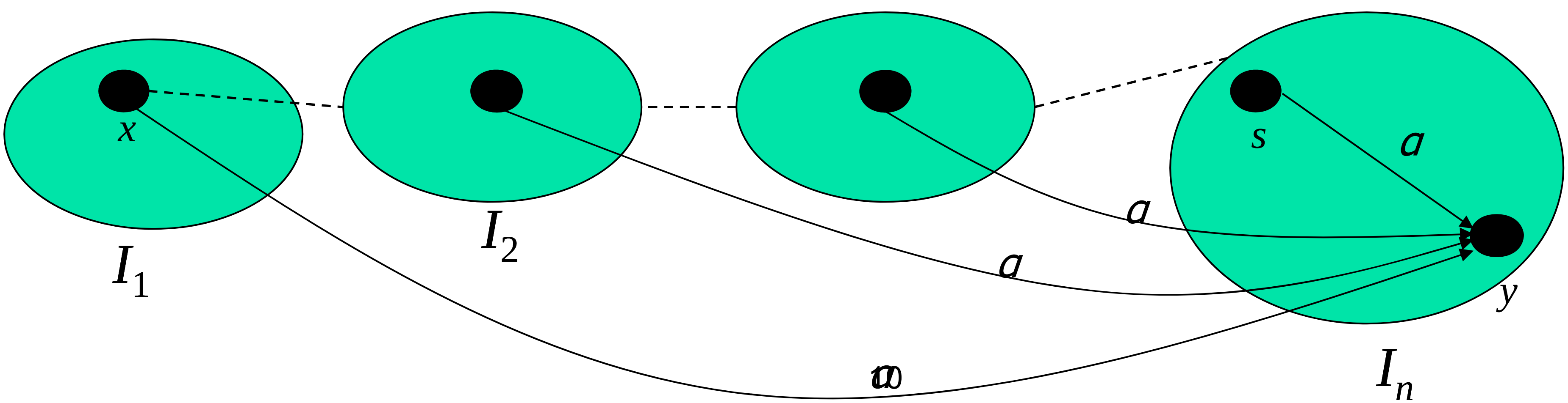
Bridge -- example



Theorem: Can_share(α,**x**,**y**,G_0) (for subjects)

- Subject_can_share(α, x, y, G_0) is true iff if x and y are subjects and
  - there is an α edge from x to y in G_0

  OR if:
  - ∃ a subject s ∈ G_0 with an s-to-y α edge, and
  - ∃ islands $I_1, \ldots, I_n$ such that $x \in I_1$, **s** ∈ I_n, and there is a bridge from I_j to I_{j+1}

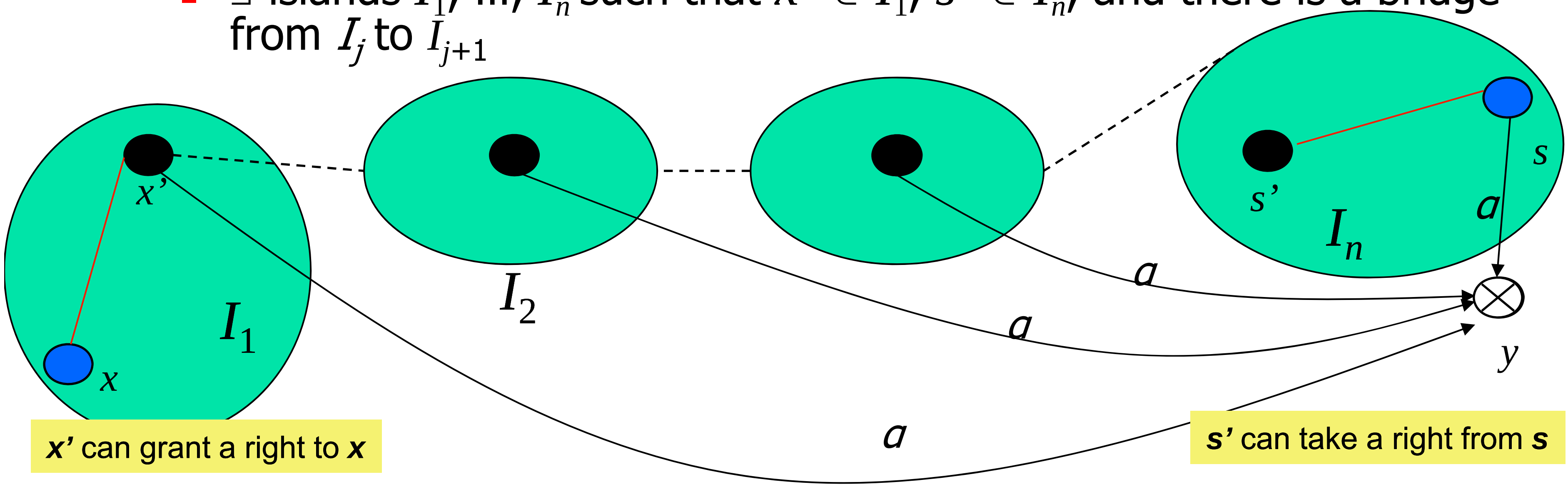


What about objects? Initial, terminal spans

- $\boldsymbol{x}$ *initially spans* to $\boldsymbol{y}$ if $\boldsymbol{x}$ is a subject and there is a *tg*-path between them with *t* edges ending in a *g* edge (i.e., $t_{\rightarrow}{}^{*}g_{\rightarrow}$)
  - $\boldsymbol{x}$ can grant a right to $\boldsymbol{y}$
- $\boldsymbol{x}$ *terminally spans* to $\boldsymbol{y}$ if $\boldsymbol{x}$ is a subject and there is a *tg*-path between them with *t* edges (i.e., $t_{\rightarrow}{}^{*}$)
  - $\boldsymbol{x}$ can take a right from $\boldsymbol{y}$

Theorem: Can_share(α,**x**,**y**,G_0)

- Can_share(α,x, y,G_0) iff there is an α edge from x to y in G_0 or if:
  - $\exists$ a vertex $s \in G_0$ with an s to y α edge,
  - $\exists$ a subject x' such that $x'=x$ or x' *initially spans* to x,
  - $\exists$ a subject s' such that $s'=s$ or s' *terminally spans* to s, and
  - $\exists$ islands I_1, …, I_n such that $x' \in I_1$, $s' \in I_n$, and there is a bridge from I_j to I_{j+1}



Theorem: Can_share(α,**x**,**y**,G_0)

- Corollary: There is an $O(|V|+|E|)$ algorithm to test can_share: Decidable in linear time!!

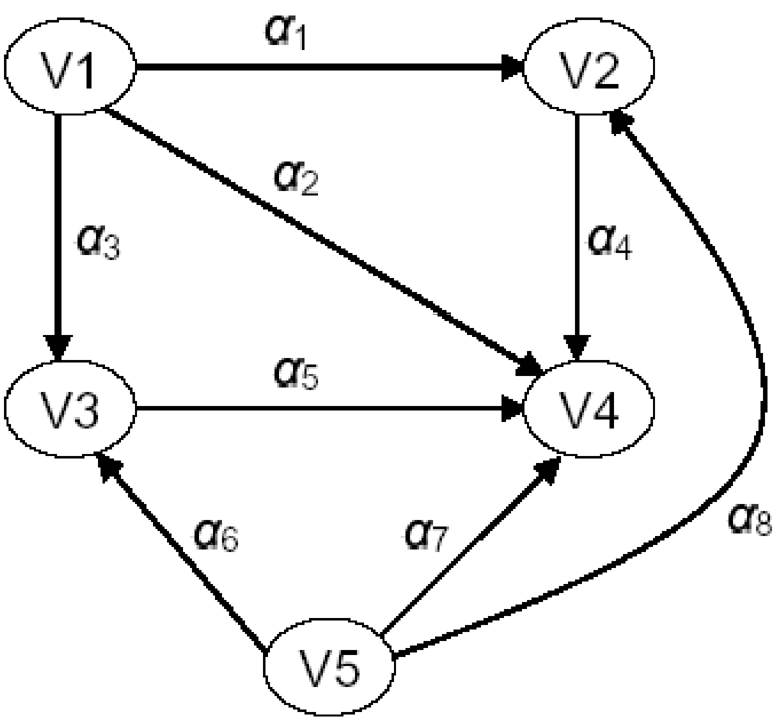
- Theorem:
  - Let G_0 contain exactly one vertex and no edges,
  - R a set of rights.
  - $G_0 \vdash^* G$ iff G is a finite directed acyclic graph, with edges labeled from R, and at least one subject with no incoming edge.
  - *Only if* part: v is initial subject and $G_0 \vdash^* G$;
    - No rule allows the deletion of a vertex
    - No rule allows an incoming edge to be added to a vertex without any incoming edges. Hence, as v has no incoming edges, it cannot be assigned any

Theorem: Can_share(α,**x**,**y**,G_0)

- *If* part : G meets the requirement
  - Assume v is the vertex with no incoming edge and apply rules

1. Perform "v creates ($\alpha \cup \{g\}$ to) new x_i" for all 2<=i <= n, and α is union of all labels on the incoming edges going into x_i in G
2. For all pairs x, y with x having α over y in G, perform "v grants (α to y) to x"
3. If β is the set of rights x has over y in G, perform "v removes (($\alpha \cup \{g\}$) - β) to y"

Example

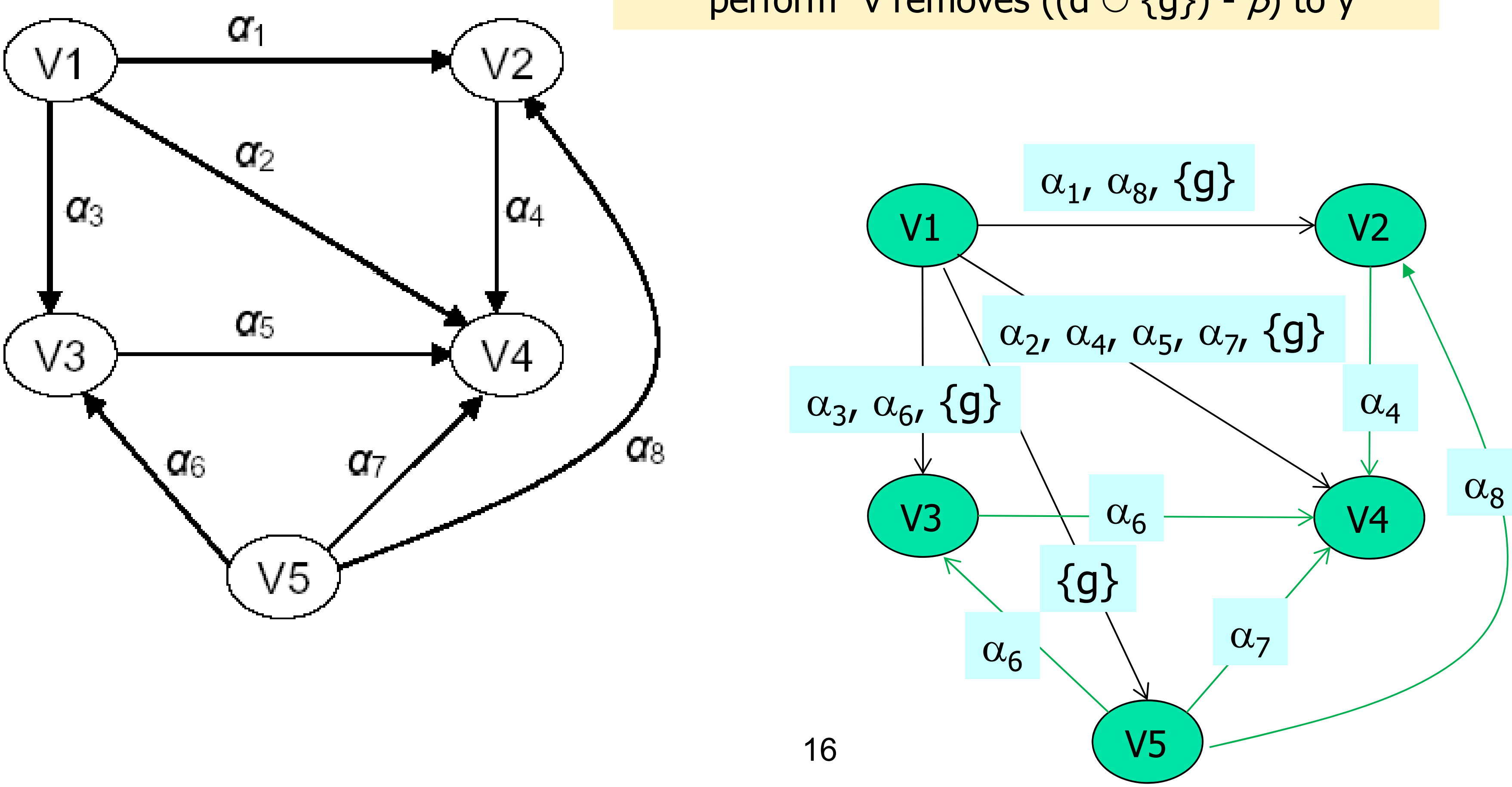


V1
V2
V3
V4
V5
α_1
α_2
α_3
α_4
α_5
α_6
α_7
α_8

Example

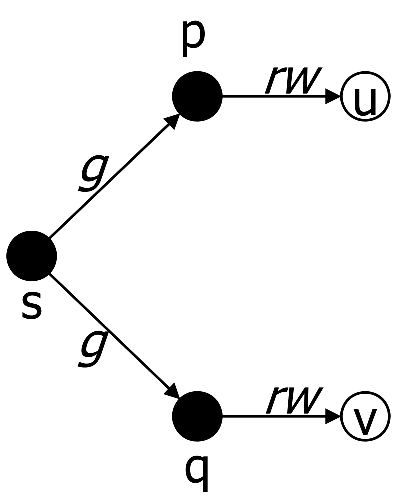
- V1 is the vertex with no incoming edge

1. Perform "v creates ($\alpha \cup \{g\}$ to) new x_i" for all $2 <= i <= n$, and α is union of all labels on the incoming edges going into x_i in G
2. For all pairs x, y with x having α over y in G, perform "v grants (α to y) to x"
3. If β is the set of rights x has over y in G, perform "v removes (($\alpha \cup \{g\}$) - β) to y"

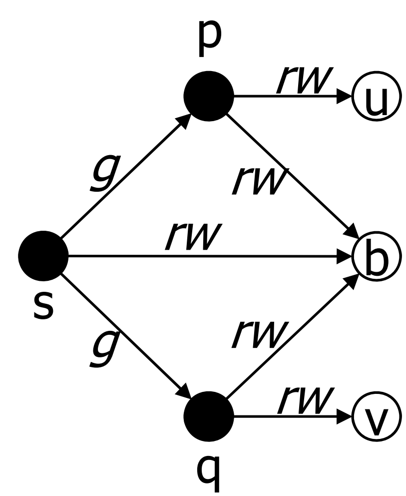


Take-Grant Model: Sharing through a Trusted Entity

- Let *p* and *q* be two processes
- Let *b* be a buffer that they share to communicate
- Let *s* be third party (e.g. operating system) that controls *b*



Witness
- S creates ({r, w}, to new object) b
- S grants ({r, w}, b) to p
- S grants ({r, w}, b) to q

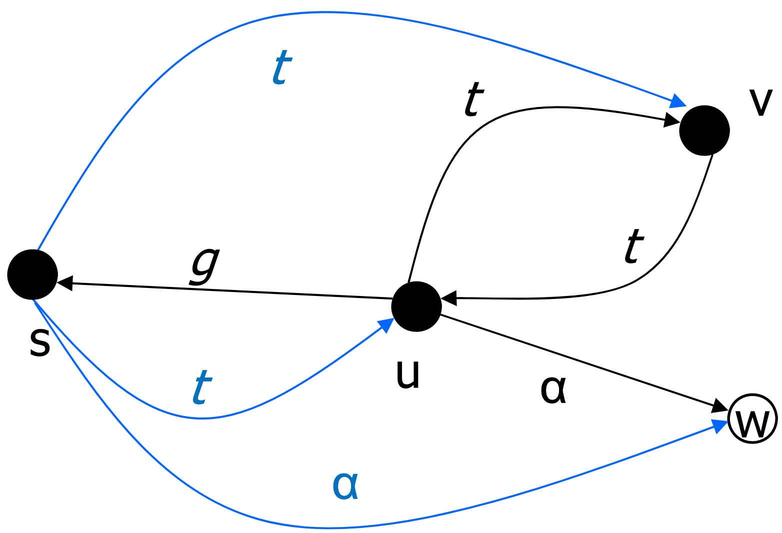


Theft in Take-Grant Model

- Can_steal(α,**x**,**y**,G_0) is true if there is no α edge from **x** to **y** in G_0 and $\exists$ sequence $G_1, \ldots, G_n$ s. t.:
  - $\exists$ α edge from **x** to **y** in G_n,
  - $\exists$ rules $\rho_1, \ldots, \rho_n$ that take $G_{i-1} \vdash \rho_i\ G_i$, and
  - $\forall$ **v**,**w** $\in G_{i-1}$, $1 \le i < n$, if $\exists$ α edge from **v** to **y** in G_0 then ρ_i is not "**v** grants (α to **y**) to **w**"

  - Disallows owners of α rights to y from transferring those rights
  - Does not disallow them to transfer other rights
  - This models a Trojan horse

A witness to theft



u grants (t to v) to s

s takes (t to u) from v

s takes (α to w) from u

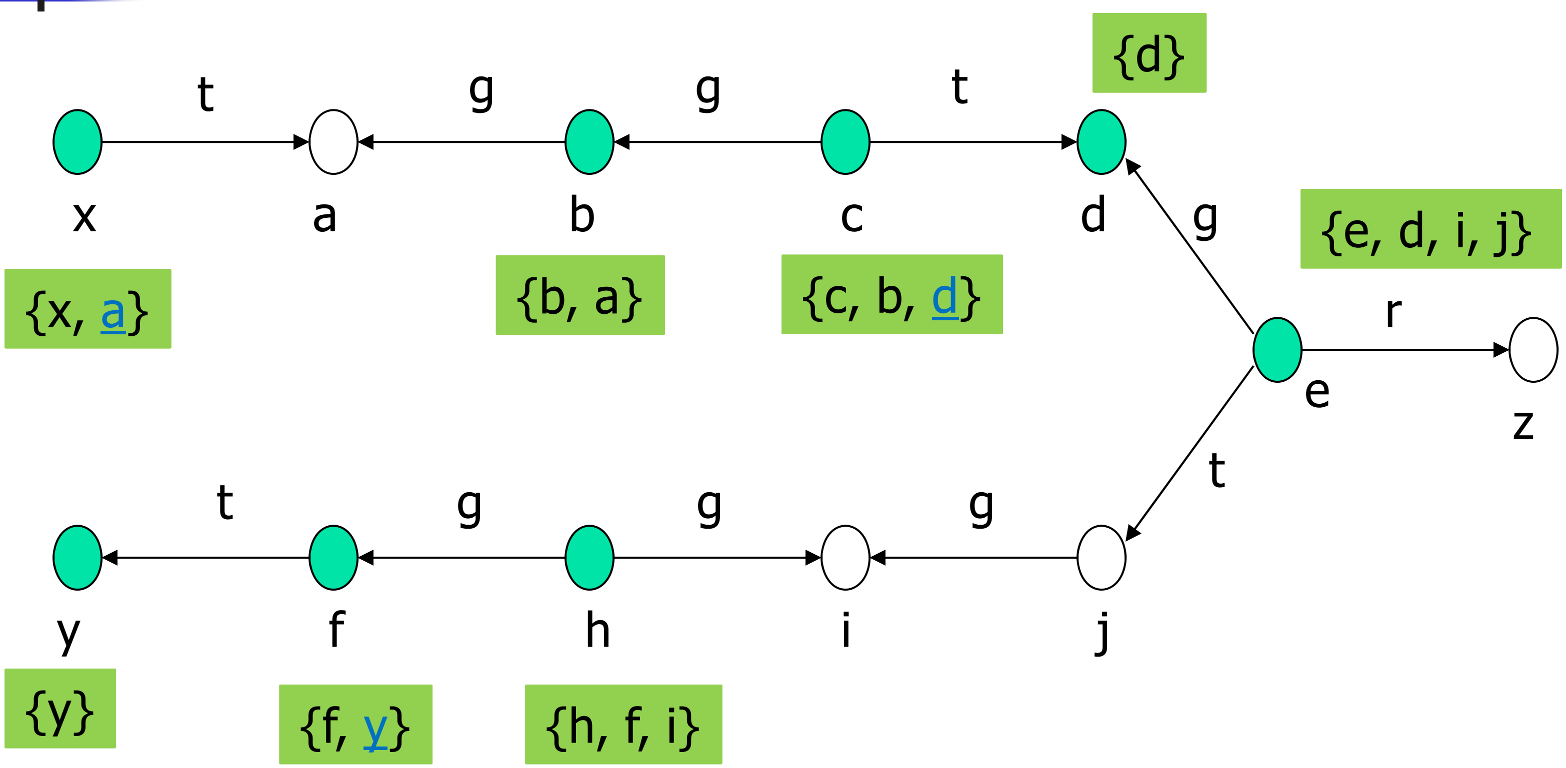
Conspiracy

- Theft indicates cooperation: which subjects are actors in a transfer of rights, and which are not?
- Next question is
  - How many subjects are needed to enable *Can_share(a,**x**,**y**,G_0)?*
- Note that a vertex y
  - Can take rights from any vertex to which it terminally spans
  - Can grant rights to any vertex to which it initially spans
- Access set A(**y**) with focus **y** (y is subject) is union of
  - set of vertices **y,**
  - vertices to which **y** initially spans, and
  - vertices to which **y** terminally spans

Conspiracy

- Deletion set $\delta(y,y')$: All $z \in A(y) \cap A(y')$ for which
  - **y** initially spans to **z** and **y'** terminally spans to **z**
  - **y** terminally spans to **z** and **y'** initially spans to z
  - **z**=**y** &
  - **z**=**y'**
- Conspiracy graph H of G_0:
  - Represents the paths along which subjects can transfer rights
  - For each subject in G_0, there is a corresponding vertex h(x) in H
  - if δ(**y**,**y'**) not empty, edge from h(**y**) to h(**y'**)

Example



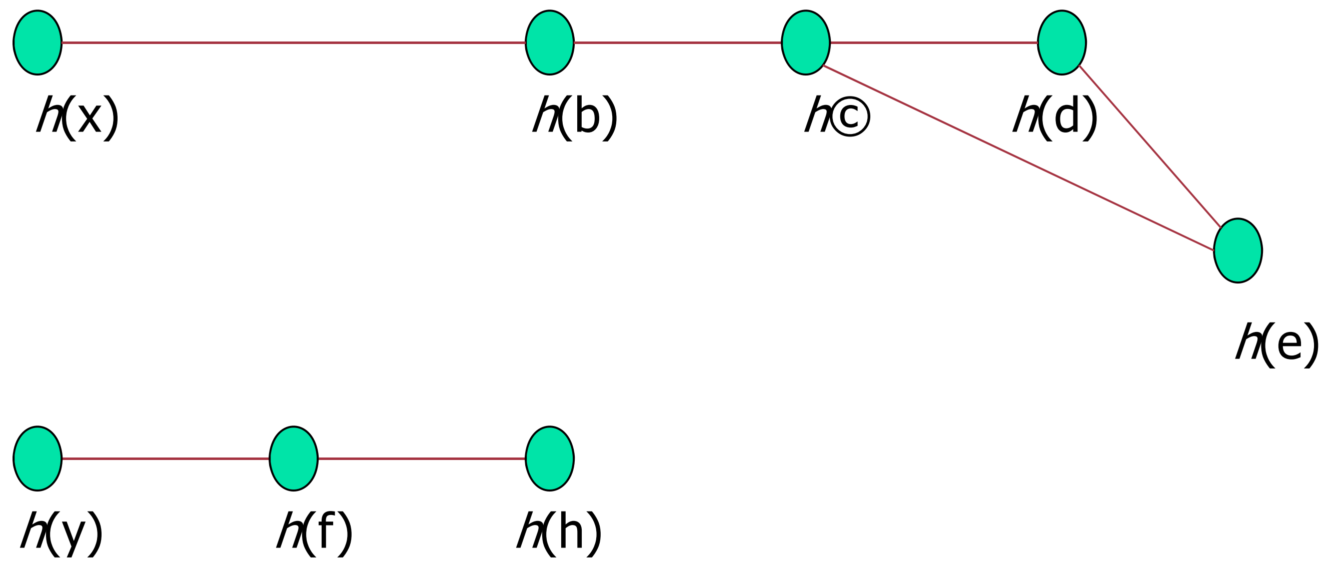
Find the conspiracy graph!!
And the witness!!

Example

Conspiracy graph H of G_0:

For each subject in G_0, there is a corresponding vertex h(x) in H

if δ(**y**,**y'**) not empty, edge from h(**y)** to h(**y')**



Theorems

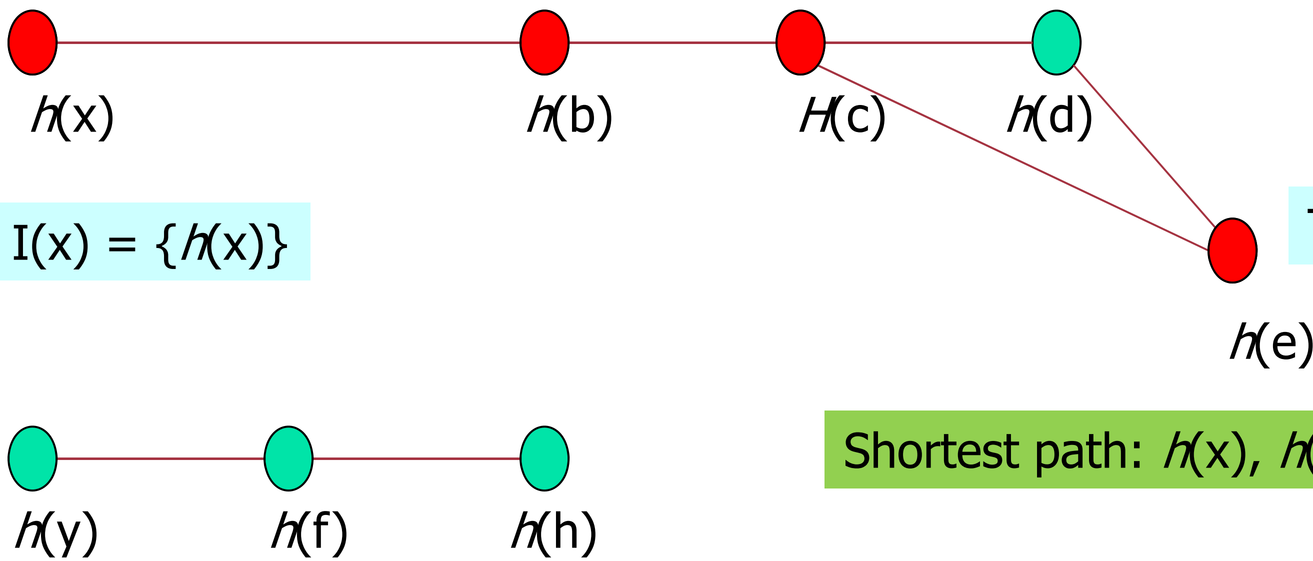
- I(p) =
  - contains the vertex h(p) and the set of all vertices h(p′) such that p′ initially spans to p
- T(q) =
  - contains the vertex h(q) and the se t of all vertices h(q′) such that q′ terminally spans to q
- Theorem 3-13:
  - Can_share(α,**x**,**y**,G_0) iff there is a path from some h(p) in I(x) to some h(q) in T(y)
- Theorem 3-14:
  - Let L be the number of vertices on a shortest path between h(p) and h(q) (as in theorem 3-13), then L conspirators are necessary and sufficient to produce a witness to Can_share(α,**x**,**y**,G_0)

Example

Conspiracy graph H of G_0:

For each subject in G_0, there is a corresponding vertex h(x) in H

if δ(**y**,**y'**) not empty, edge from h(**y**) to h(**y'**)



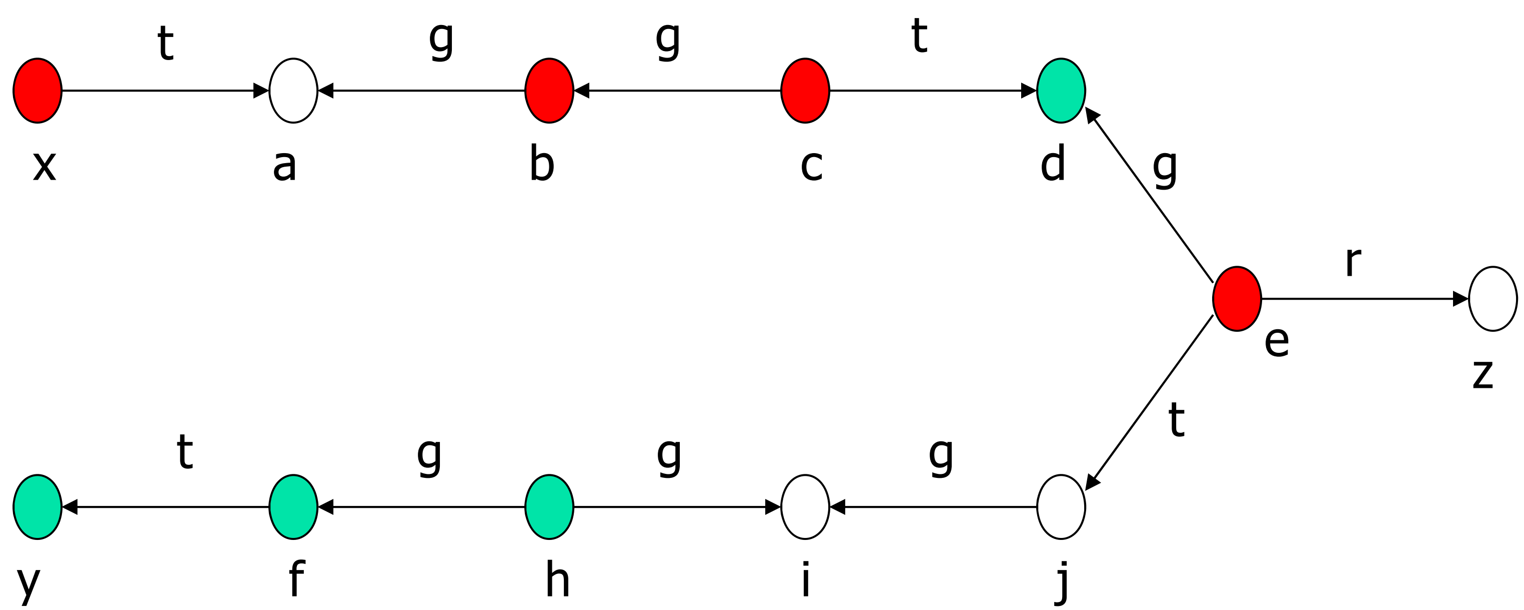
I(x) = {*h*(x)}

T(x) = {*h*(e)}

Shortest path: *h*(x), *h*(b), *h*(c), *h*(e)

Example

What is the witness??



Summary

- Take-Grant model –
  - specific system
  - Restricted
- Safety question is not undecidable
  - Linear to the size of the graph
- Theft and conspiracy issues