Conspiracy and Information Flow in the Take-Grant Protection Model

Matt Bishop

Department of Computer Science
University of California at Davis
Davis, CA 95616-8562

ABSTRACT

The Take Grant Protection Model is a theoretic model of access control that captures the notion of information flow throughout the modelled system. This paper analyzes the problem of sharing information in the context of paths along which information can flow, and presents the number of actors necessary and sufficient to share information, in this model. The results are applied to information flow in a network to reduce the size of the set of actors who could have participated in the theft.

1. Introduction

The nature of access control and information flow are critical to understanding the security of any system. The Take-Grant Protection Model is a formal model of access control which combines the transfer of rights and the transfer of information to present a cohesive picture of transfers throughout a system. It differs from other models such as the access control matrix model by specifying both the sequences of primitive operations making up the body of the commands and the set of tests upon which the execution of those sequences is conditioned.

This model represents systems as graphs to be altered by specific operations. Developed to test the limits of the results in [7], the focus of most studies of the Take-Grant Protection Model has been on characterizing conditions necessary and sufficient for the transfer of rights and information, and on the complexity of testing for those conditions in a representation of a system. For this reason it is in some sense of more "practical" use than other formal systems, in that the security question is decidable and the study of the complexity of conditions allowing compromise is emphasized.

Early work on the Take-Grant Protection Model [9][10] dealt with the transfer of rights assuming all active agents in the system would cooperate. Snyder extended these characterizations to include conditions under which rights could be stolen [14]; Bishop and Snyder introduced the notion of information flow and formulated necessary and sufficient conditions for information

sharing [4], and Bishop extended these characterizations to include conditions under which information could be stolen [2].

This paper extends those results in the direction suggested by [14] to present a notion of "conspirators" in the context of information flow. We establish precise bounds on the number of actors required for information to be transferred from one vertex to another, and contrast these results with similar results for the transfer of rights.

Applications of the Take-Grant Protection Model to various systems have been explored [3][8][13][16]. This paper also tries to place its theoretical results into an applied context by exploring how these results can be used to analyze the actors moving information around a network. Further applications are of course possible, but using the new results to analyze current models of disclosure and integrity (for example, those described in [1][5][6][11][12][15]) is itself a separate paper; it is beyond the scope of the issues addressed here.

We quickly review the basic definitions and relevant results of the Take-Grant Protection Model [2]. Following that, we present bounds on the number of actors needed for information to be shared (or stolen). We then briefly compare our results to similar ones for theft of rights. To demonstrate the usefulness of the concepts, we examine an application of this model to the Internet. Finally, we suggest areas for future research.

2. Basic Definitions and Results

The Take-Grant Protection Model represents a system by a finite, directed *protection graph*. in which labelled edges represent rights and vertices represent entities. Entities are either *subjects* (represented by ●) or *objects* (represented by ❍). Vertices which may be either subjects or objects are represented by ⊗. Changes to the protection state of the system are represented by changes to the graph. The rules governing the transfer of rights are called *de jure* rules and are as follows:

take: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** be a subject. Let there be an edge from **x** to **y** labelled γ with $t \in \gamma$, an edge from **y** to **z** labelled β, and $\alpha \subseteq \beta$. Then the *take* rule defines a new graph G_1 by adding an edge to the protection graph from **x** to **z** labelled α. Graphically,

The rule is written "**x** takes (α to **z**) from **y**."

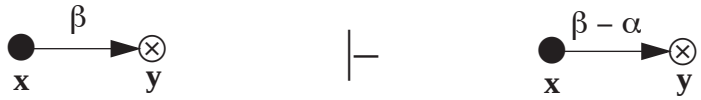
grant: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** be a subject. Let there be an edge from **x** to **y** labelled γ with $g \in \gamma$, an edge from **x** to **z** labelled β, and $\alpha \subseteq \beta$. Then the *grant* rule defines a new graph G_1 by adding an edge to the protection graph from **y** to **z** labelled α. Graphically,

The rule is written "**x** grants (α to **z**) to **y**."

create: Let **x** be any subject in a protection graph G_0 and let $\alpha \subseteq R$. Create defines a new graph G_1 by adding a new vertex **y** to the graph and an edge from **x** to **y** labelled α. Graphically,

The rule is written "**x** creates (α to new vertex) **y**."

remove: Let **x** and **y** be any distinct vertices in a protection graph G_1 such that **x** is a subject. Let there be an explicit edge from **x** to **y** labelled β, and let $\alpha \subseteq R$. Then *remove* defines a new graph G_1 by deleting the α labels from β. If β becomes empty as a result, the edge itself is deleted. Graphically,



The rule is written "**x** removes (α to) **y**."

Definition. A *tg-path* is a nonempty sequence $\mathbf{v}_0, \ldots, \mathbf{v}_n$ of distinct vertices such that for all i, $0 \le i < n$, $\mathbf{v}_i$ is connected to $\mathbf{v}_{i+1}$ by an edge (in either direction) with a label containing *t* or *g*.

Definition. Vertices are *tg-connected* if there is a tg-path between them.

Definition. An *island* is a maximal tg-connected subject-only subgraph.

With each tg-path, associate one or more words over the alphabet in the obvious way. If the $\{ \vec{t}, \overleftarrow{t}, \vec{g}, \overleftarrow{g} \}$ path has length 0, then the associated word is the null word ν. The notation *t** means zero or more occurences of the character *t*, so for example *t*g* represents the sequence *g*, *tg*, *ttg*, ….

Definition. A vertex $\mathbf{v}_0$ *initially spans* to $\mathbf{v}_n$ if $\mathbf{v}_0$ is a subject and there is a tg-path between $\mathbf{v}_0$ and $\mathbf{v}_n$ with associated word in $\{ \vec{t}*\vec{g} \} \cup \{ \nu \}$.

Definition. A vertex $\mathbf{v}_0$ *terminally spans* to $\mathbf{v}_n$ if $\mathbf{v}_0$ is a subject and there is a tg-path between $\mathbf{v}_0$ and $\mathbf{v}_n$ with associated word in $\{ \vec{t}* \}$.

Definition. A *bridge* is a tg-path with endpoints $\mathbf{v}_0$ and $\mathbf{v}_n$ both subjects and the path's associated word in $B = \{ \vec{t}* \ , \overleftarrow{t}*, \ \vec{t}*\vec{g}\overleftarrow{t}*, \ \vec{t}*\vec{g}\overleftarrow{t}* \}$.

Definition. The predicate *can•share*(α, **x**, **y**, G_0) is true for a set of rights α and two vertices **x** and **y** if and only if there exist protection graphs $G_1, \ldots, G_n$ such that $G_0 \vdash^* G_n$ using only *de jure* rules, and in G_n there is an edge from **x** to **y** labelled α.

Theorem 1. [10] The predicate *can•share*(α, **x**, **y**, G_0) is true if and only if there is an edge from **x** to **y** in G_0 labelled α, or if the following hold simultaneously:

(1.1) there is a vertex $\boldsymbol{s} \in G_0$ with an **s**-to-**y** edge labelled α;

(1.2) there exists a subject vertex **x´** such that **x´** = **x** or **x´** initially spans to **x**;

(1.3) there exists a subject vertex **s´** such that **s´** = **s** or **s´** terminally spans to **s**; and

(1.4) there exist islands $I_1, \ldots, I_n$ such that **x´** is in I_1, **s´** is in I_n, and there is a bridge from I_j to I_{j+1} ($1 \leq j < n$).

Definition. The predicate *can•steal*(α, **x**, **y**, G_0) is true if and only if there is no edge labelled α from **x** to **y** in G_0, there exist protection graphs $G_1, \ldots, G_n$ such that $G_0 \vdash^* G_n$ using only *de jure* rules, in G_n there is an edge from **x** to **y** labelled α, and if there is an edge labelled α from **s** to **q** in G_0, then no rule in a witness has the form "**s** grants (α to **q**) to **z**" for any $\mathbf{z} \in G_j$ ($1 \leq j < n$).

Theorem 2. [14] The predicate *can•steal*(α, **x**, **y**, G_0) is true if and only if the following hold simultaneously:

(2.1) there is no edge labelled α from **x** to **y** in G_0;

(2.2) there exists a subject vertex **x´** such that **x´** = **x** or **x´** initially spans to **x**;

(2.3) there is a vertex **s** with an edge from **s** to **y** labelled α in G_0;

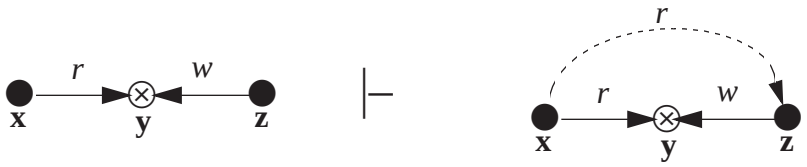
(2.4) *can•share*(t, **x´**, **s**, G_0) is true.

The *de facto* rules represent paths along which information may flow. We cannot use explicit edges for this purpose because no change in authority occurs. Hence, we use a dashed line, labelled by *r*, to represent the path of a potential *de facto* transfer (called an *implicit* edge). Implicit

edges cannot be manipulated by *de jure* rules, since the *de jure* rules only affect authorities recorded in the protection system, and implicit edges do not represent such authority.

The following set of *de facto* rules was introduced in [4] to model transfers of information:

post: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** and **z** be subjects. Let there be an edge from **x** to **y** labelled α with $r \in \alpha$ and an edge from **z** to **y** labelled β, where $w \in \beta$. Then the *post* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled *r*. Graphically,

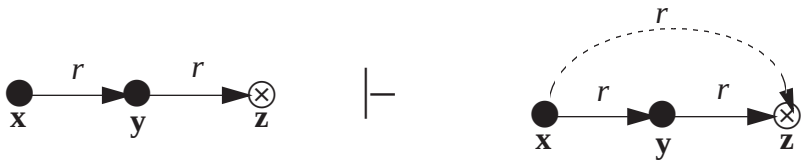


The rule is written "**z** posts to **x** through **y**."

pass: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **y** be a subject. Let there be an edge from **y** to **x** labelled α with $w \in \alpha$ and an edge from **y** to **z** labelled β, where $r \in \beta$. Then the *pass* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled *r*. Graphically,

The rule is written "**y** passes from **z** to **x**." .

spy: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** and **y** be subjects. Let there be an edge from **x** to **y** labelled α with $r \in \alpha$ and an edge from **y** to **z** labelled β, where $r \in \beta$. Then the *spy* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled *r*. Graphically,



The rule is written "**x** spies on **z** using **y**."

find: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **y** and **z** be subjects. Let there be an edge from **y** to **x** labelled α with $w \in \alpha$ and an edge from **z** to **y** labelled β,

where $w \in \beta$. Then the *find*rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled *r*. Graphically,

The rule is written "**x** finds from **z** through **y**."

Whether these rules capture all ways in which information may leak is an open question; ultimately, the answer depends on how the system being modelled controls that information flow. The above rules appear to capture the most common techniques, and have been used in tthe past, so for consistency we shall use them here.

Definition. The predicate *can•know*(**x**, **y**, G_0) is true if and only if there exists a sequence of protection graphs $G_0, \ldots, G_n$ such that G_n is derived from G_0 by rule applications, and in G_n there is an edge from **x** to **y** labelled *r* or an edge from **y** to **x** labelled *w*, and if the edge is explicit, its source is a subject.

Definition. An *rwtg-path* is a nonempty sequence $\mathbf{v}_0, \ldots, \mathbf{v}_n$ of distinct vertices such that for all *i*, $0 \le i < n$, $\mathbf{v}_i$ is connected to $\mathbf{v}_{i+1}$ by an edge (in either direction) with a label containing *t*, *g*, *r* or *w*.

With each rwtg-path, associate one or more words over the alphabet $\{ \vec{t}, \overleftarrow{t}, \vec{g}, \overleftarrow{g}, \vec{r}, \overleftarrow{r}, \vec{w}, \overleftarrow{w} \}$ in the obvious way.

Definition. A vertex $\mathbf{v}_0$ *rw-initially spans* to $\mathbf{v}_n$ if $\mathbf{v}_0$ is a subject and there is an rwtg-path between $\mathbf{v}_0$ and $\mathbf{v}_n$ with associated word in $\{ \vec{t}^*\vec{w} \} \cup \{ \nu \}$.

Definition. A vertex $\mathbf{v}_0$ *rw-terminally spans* to $\mathbf{v}_n$ if $\mathbf{v}_0$ is a subject and there is an rwtg-path between $\mathbf{v}_0$ and $\mathbf{v}_n$ with associated word in $\{ \vec{t}^* \vec{r} \}$.

Definition. A *connection* is an rwtg-path with $\mathbf{v}_0$ and $\mathbf{v}_n$ both subjects and the path's associated word in $C = \{ \vec{t}^*\vec{r}, \overleftarrow{w}\,\overleftarrow{t}^*, \vec{t}^*\vec{r}, \overleftarrow{w}\,\overleftarrow{t}^* \}$.

The next result [4] characterizes the set of graphs for which *can•know* is true:

Theorem 3. [4] The predicate *can•know*(**x**, **y**, G_0) is true if and only if there exists a sequence of subjects $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in G_0 ($n \geq 1$) such that the following conditions hold:

(3.1) $\mathbf{u}_1 = \mathbf{x}$ or $\mathbf{u}_1$ rw-initially spans to **x**;

(3.2) $\mathbf{u}_n = \mathbf{y}$ or $\mathbf{u}_n$ rw-terminally spans to **y**;

(3.3) for all i, $1 \leq i < n$, there is an rwtg-path between $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ with associated word in $B \cup C$.

Lemma 4. [2] If two subjects **x** and **y** in G_0 are connected by a bridge, then *can•know*(**x**, **y**, G_0) and *can•know*(**y**, **x**, G_0) are true.

Lemma 5. [2] Let a subject **x** be connected by a bridge to another subject **y**. If either **x** or **y** does not act, no sequence of graph transformations can add an implicit or explicit edge from **x** to **y**.

Lemma 6. [4] If two subjects **x** and **y** in G_0 are connected by a connection, then *can•know*(**x**, **y**, G_0) is true.

Definition. The predicate *can•snoop*(**x**, **y**, G_0) is true if and only if *can•steal*(r, **x**, **y**, G_0) is true or there exists a sequence of graphs and rule applications $G_0 \vdash_{\rho 1} \ldots \vdash_{\rho n} G_n$ for which all of the following conditions hold:

(a) there is no explicit edge from **x** to **y** labelled r in G_0;

(b) there is an implicit edge from **x** to **y** labelled r in G_n; and

(c) neither **y** nor any vertex directly connected to **y** in G_0 is an actor in a grant rule or a *de facto* rule resulting in an (explicit or implicit) read edge with **y** as its target.

Theorem 7. [2] For distinct vertices **x** and **y** in a protection graph G_0 with explicit edges only, the predicate *can•snoop*(**x**, **y**, G_0) is true if and only if *can•steal*(r, **x**, **y**, G_0) is true or all of the following conditions hold:

(7.1) there is no edge from **x** to **y** labelled r in G_0;

(7.2) there is a subject vertex $\mathbf{w}_1$ such that $\mathbf{w}_1 = \mathbf{x}$ or $\mathbf{w}_1$ rw-initially spans to **x** in G_0;

(7.3) there is a subject vertex $\mathbf{w}_n$ such that $\mathbf{w}_n \neq \mathbf{y}$, there is no edge labelled r from $\mathbf{w}_n$ to **y** in G_0, and $\mathbf{w}_n$ rw-terminally spans to **y** in G_0; and

(7.4) *can•know*($\mathbf{w}_1$, $\mathbf{w}_n$, G_0) is true.

3. Conspiracy in a Single-Path Graph

Given that we can determine whether knowing (that is, the sharing of information) is possible in a Take-Grant graph, how many vertices must cooperate in the sharing? The answer to this question will give us an answer to a more interesting one involving snooping, namely how many *actors* are necessary to steal information.

Before we tackle these questions in all their generality, let us restrict our attention for the remainder of this section to a specific type of graph. Let G be a graph with vertices **x**, **y**, with *can•know*(**x**, **y**, G) true, and containing only those vertices and edges needed to witness this predicate. Thus, G is composed of the vertices and edges of the path along which information is to be propagated or rights transferred. Let the set of (subject and object) vertices

$$V = \{ \mathbf{z}_i \mid \mathbf{x} = \mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m, \mathbf{y} = \mathbf{z}_{m+1} \}$$

Clearly each edge $\mathbf{z}_i\mathbf{z}_{i+1}$, where $\{ \mathbf{z}_i, \mathbf{z}_{i+1} \} \subseteq V$, is an rwtg-path of length 1; as *can•know*(**x**, **y**, G) holds, there are subject vertices $\mathbf{v}_i$, $0 \le i \le n \le m$, in this set. Consider the rwtg-paths between these subjects; by Theorem 3, the words associated with these paths are in $B \cup C$, if **y** is not a subject then there is an rw-terminal span from a subject $\mathbf{v}_n$ to **y**, and if **x** is not a subject, then there is an rw-initial span from $\mathbf{v}_0$ to **x**.

The following definitions capture the notion of the "reach" of a vertex:

Definition. A *terminal access set* T(**y**) is defined as the set containing **y** and all vertices to which **y** terminally or rw-terminally spans.

Definition. An *initial access set* I(**y**) is defined as the set containing **y** and all vertices to which **y** initially or rw-initially spans.

Here, T(**y**) is the maximal set of vertices from which **y** can obtain information, and I(**y**) is the maximal set of vertices to which **y** can pass rights or information. Note that these sets are not necessarily identical, because while a bridge between subjects allows the symmetrical transfer of rights, a connection allows only a one-way transfer of information. This adds significant complexity to the conspiracy problem.

Definition. A subject **x** is an *information gate* if any one of the following conditions holds:

(i) $\mathbf{x} = \mathbf{v}_0$, the only word associated with the edge $\mathbf{v}_0\mathbf{v}_1$ is $\overleftarrow{g}$ or $\overleftarrow{t}$ and there are no other edges incident to **x**;

(ii) $\mathbf{x} = \mathbf{v}_i$, there are exactly two edges incident upon **x**, and the word associated with the path $\mathbf{v}_{i-1}\mathbf{v}_i\mathbf{v}_{i+1}$ is in the set{ $\overrightarrow{t}\overleftarrow{t}$, $\overrightarrow{g}\overleftarrow{g}$, $\overrightarrow{t}\overleftarrow{w}$, $\overrightarrow{g}\overleftarrow{w}$, $\overrightarrow{r}\overleftarrow{t}$, $\overrightarrow{r}\overleftarrow{g}$ }; or

(iii) $\mathbf{x} = \mathbf{v}_{n+1}$, the only word associated with the edge $\mathbf{v}_n\mathbf{v}_{n+1}$ is $\overrightarrow{g}$ or $\overrightarrow{t}$, and there are no other edges incident to **x**.

For an information gate **x**, $T(\mathbf{x}) = I(\mathbf{x}) = \{ \mathbf{x} \}$. The idea is that information can be passed into an information gate, or out of an information gate, without the gate taking any action, but in order for information to be passed through a gate (that is, both in and out), the information gate must be active in a rule application. Note that the information gate need *not* apply the rule; if it does not, it must then be a subject in a *de facto* rule, because unless the subjects shown in those rules act, information cannot flow along the implicit edge. This is a subtlety not evident when dealing with conspiracies in graphs using only *de jure* rule sets, and although the information gate is analogous to a *sink* in [14], the difference in definition is substantial and reflects the difference between information and rights transfer.

Definition. An *access set cover for a protection graph* G *with foci* $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is a family of sets $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_n)$, $T(\mathbf{v}_n)$ where for $2 \le i \le n$, there exists a $j \le n$ such that$\{ \mathbf{v}_{i-1}, \mathbf{v}_i \} \subseteq I(\mathbf{v}_j) \cup T(\mathbf{v}_j)$. Clearly, this family is a covering set for *G*. If the cover minimizes *n* over all possible access set covers, it is said to be a *minimal* cover.

Notice that the set of actors needed to implement *can•know* generates a cover for *G*. In fact,

Lemma 8. A minimal set of actors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in a sequence of rule applications producing a witness to *can•know*(**x**, **y**, *G*) generates an access set cover for *G*.

Informal Proof: If this lemma is false, there is a set of actors in a witness to *can•know*(**x**, **y**, *G*) which does not produce an access set cover for *G*. Let $\mathbf{v}_k$ be one vertex not in any element of the access set cover. Then neither information nor rights is transferred through $\mathbf{v}_k$, and hence it can be deleted from the set of actors, showing that set is not minimal.

Proof: Let $\rho_1, \ldots, \rho_m$ be a set of rules required for a minimal set of actors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ to produce a witness to *can•know*(**x**, **y**, *G*). Without loss of generality we may take $\rho_1, \ldots, \rho_m$ to be the shortest sequence of rule applications for that particular set of actors. Let the access sets $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_n)$, $T(\mathbf{v}_n)$ with foci $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be defined on *G*. Suppose $\mathbf{z} \notin I(\mathbf{v}_i)$ and $\mathbf{z} \notin T(\mathbf{v}_i)$ for all *i*. By The-

orem 3 and the definition of *T*, no actor can receive information from $\mathbf{v}_i$, and by definition of *I*, **z** cannot pass on information from any other actor; hence **z** and its incident edges may be deleted without affecting rules $\rho_1, \ldots, \rho_m$. But this violates condition (3.3) of Theorem 3, as the graph is no longer connected, which in turn means that *can*•*know*(**x**, **y**, *G*) is false. This contradicts the initial assumption that *can*•*know*(**x**, **y**, *G*) is true. This proves the claim. ■

We next make formal our claim that information gates must act for information to be passed along their incident edges.

Lemma 9. If vertex $\mathbf{v}_i$ is an information gate, and in a witness to *can*•*know*(**x**, **y**, *G*) an explicit or implicit edge is constructed between some vertex $\mathbf{v}_k$, $k < i$, and another vertex $\mathbf{v}_l$, $i < l$, then the vertex $\mathbf{v}_i$ must be an actor.

Informal argument: Assume the witness is the shortest witness to*can*•*know*(**x**, **y**, *G*) . The vertex $\mathbf{v}_i$ cannot be involved in a *de jure* rule, nor in a *de facto* rule, and hence can be deleted from the witness and the set of actors. This contradicts the assumption that the witness is the shortest one.

Proof: We demonstrate this for the case of $\mathbf{v}_i$'s incident edges being $\vec{t}$ and $\overleftarrow{r}$; the proof for the other cases is similar. (The Appendix contains some useful witnesses, and proof of inability to supply other witnesses, for these proofs.)

First, by condition (3.3) of Theorem 3, $\mathbf{v}_i$ must be a subject, for if not, *can*•*know*(**x**, **y**, *G*) is false because the paths through that information gate are neither bridges nor connections. So, assume $\mathbf{v}_i$ is not an actor, and consider the effects of this on a set of rule applications $\rho_1, \ldots, \rho_m$ required for a minimal set of actors to produce a witness showing that *can*•*know*(**x**, **y**, *G*) holds. Without loss of generality we take $\rho_1, \ldots, \rho_m$ to be the shortest sequence of rule applications for that particular set of actors.

No rule is of the form "**z** takes (α to **y**) from $\mathbf{v}_i$" for any **z** in *G*, since $\mathbf{v}_i$ has no edges going from it to any other $\mathbf{v}_j \in V$, and by the nature of the *de jure* rules can never be assigned any. As the number *m* of rules applied is minimal, no rules of the form "**z** takes (*t* to $\mathbf{v}_i$) from **y**" or "$\mathbf{v}_{i-1}$ grants (*t* to $\mathbf{v}_i$) to **z**" for any vertex **z** in *G* are ever executed since the *t* right so assigned could not be used. Hence no *de jure* rule involves $\mathbf{v}_i$.

Now consider the *de facto* rules. Clearly, only information passing through $\mathbf{v}_i$ is relevant; hence, information will never be written into $\mathbf{v}_i$ and not later read (because then the rule could be deleted, contradicting the minimality of *m*), or read before any information is written into it (which makes sense only if $\mathbf{v}_i = \mathbf{v}_{n+1}$, in which case there are two incident edges to $\mathbf{v}_{n+1}$, and so it is not an

information gate, contradiction). The post, pass, and find rules could not be used as $\mathbf{v}_i$ has no incident write edges, and the spy rule could not be used because $\mathbf{v}_i$ would have to act, contradicting assumption. Hence no *de facto* rule involves $\mathbf{v}_i$.

Combining these, if $\mathbf{v}_i$ is not an actor, it and its incident edges can be deleted from G; but this contradicts the minimality of m. This proves the lemma. ■

With these two lemmata we are able to obtain a lower bound on the number of actors needed to share information.

Theorem 10. Let $2k$ be the number of access sets in a minimal cover of G, and let l be the number of information gates. Then $k+l$ actors are necessary to produce a witness to *can•know*.

Informal argument: The focus of each (initial and terminal) access set can obtain (or pass on) information or rights to those vertices in that access set. The information gates must act to pass information along. Hence the number of actors needed is the sum of the number of access set foci and the number of information gates.

Proof: Let $\rho_1, \ldots, \rho_m$ be a set of rules required for a minimal set of actors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ to produce a witness to *can•know*. Without loss of generality we take $\rho_1, \ldots, \rho_m$ to be the shortest sequence of rule applications for that particular set of actors. Let the access sets $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_n)$, $T(\mathbf{v}_n)$ with foci $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be defined on G. By Lemma 8, $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_n)$, $T(\mathbf{v}_n)$ at least cover G. Without loss of generality, take the vertices $\mathbf{v}_1, \ldots, \mathbf{v}_l$ to be the information gates. By Lemma 9, every one of these must be an actor. Then each of $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_l)$, $T(\mathbf{v}_l)$ is a singleton set, and its focus is a member of its adjacent access sets. Thus the other access sets $I(\mathbf{v}_{l+1})$, $T(\mathbf{v}_{l+1})$,…., $I(\mathbf{v}_{l+k})$, $T(\mathbf{v}_{l+k})$ (where $k + l = n$) constitute an access set cover for G, and their foci must also be actors. This proves the theorem. ■

To derive an upper bound we shall find two more results useful:

Lemma 11. Let $I(\mathbf{v}_1)$, $T(\mathbf{v}_1)$, …., $I(\mathbf{v}_n)$, $T(\mathbf{v}_n)$ be a minimal set access cover for G_0 ordered by increasing indices of $\mathbf{v}$ (that is, along the path from $\mathbf{x}$ to $\mathbf{y}$). If *can•know*($\mathbf{v}_{i+1}$, $\mathbf{y}$, G) is true, then for some index m there exists a graph G_m such that *can•know*($\mathbf{v}_i$, $\mathbf{y}$, G) is true and all rules in the derivation sequence $G_0 \vdash^* G_m$ are initiated by $\mathbf{v}_i$, $\mathbf{v}_{i+1}$, and perhaps $\mathbf{z} = T(\mathbf{v}_i) \cap I(\mathbf{v}_{i+1})$.

Proof: Recall that we are assuming throughout this section that *can•know*($\mathbf{x}$, $\mathbf{y}$, G) is true. Consider the spans to $\mathbf{z}$ from $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$. By the series of witnesses presented in the Appendix, in all cases the vertices acting in the rule applications witnessing *can•know*($\mathbf{x}$, $\mathbf{y}$, G) are $\mathbf{v}_i$, $\mathbf{v}_{i+1}$, and occasionally $\mathbf{z}$. ■

Corollary 12. For adjacent access sets, information can be transferred from $\mathbf{v}_i$ to $\mathbf{v}_{i+1}$ with no other actors unless there are consecutive edges with their only associated word in the set { $\overrightarrow{t}\overleftarrow{t}$, $\overrightarrow{g}\overleftarrow{g}$, $\overrightarrow{t}\overleftarrow{w}$, $\overrightarrow{g}\overleftarrow{w}$, $\overrightarrow{r}\overleftarrow{t}$, $\overrightarrow{r}\overleftarrow{g}$ }; in this case additional actions performed by $\mathbf{z} = T(\mathbf{v}_i) \cap I(\mathbf{v}_{i+1})$ are sufficient.

Proof: By inspection of the witnesses to the preceding lemma. ■

We can now use these two results to obtain an upper bound on the number of vertices which must act to share information:

Theorem 13. Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be foci of an access set cover of G and let G have l information gates. Then $k+l$ actors suffice to generate an (implicit or explicit) read edge from **x** to **y** in G.

Informal argument: For $\mathbf{v}_1$ to pass information to **x**, and $\mathbf{v}_k$ to obtain information from **y**, both $\mathbf{v}_1$ and $\mathbf{v}_k$ must act; **x** and **y** will need to act also if they are information gates. Each focus will need to act to pass information along, as will information gates. Summing these numbers gives the desired result.

Proof: Let $I(\mathbf{v}_1), T(\mathbf{v}_1), \ldots, I(\mathbf{v}_k), T(\mathbf{v}_k)$ be a minimal set access cover for G_0 with vertices $\mathbf{x} \in I(\mathbf{v}_1)$ and $\mathbf{y} \in T(\mathbf{v}_k)$. Consider first **y** and $\mathbf{v}_k$. Three cases arise:

- $\mathbf{v}_k = \mathbf{y}$. Then $can\bullet know(\mathbf{v}_k, \mathbf{y}, G)$ is trivially true.
- $\mathbf{v}_k$ terminally spans to **y**. By condition (3.3) of Theorem 3, this means **y** is a subject, so apply Lemma 4 to get the desired result. Note that **y** is an information gate in this case.
- $\mathbf{v}_k$ rw-terminally spans to **y**. Apply the take rule repeatedly to get an explicit edge; this gives the desired result.

In all cases where $can\bullet know(\mathbf{v}_k, \mathbf{y}, G)$ is true, the only actors are the focus of $T(\mathbf{v}_k)$ and, possibly, the vertex **y**; in addition, **y** acts only if it is an information gate. Applying Corollary 12 inductively, we have that whenever $can\bullet know(\mathbf{v}_i, \mathbf{y}, G)$ is true for $i = 1, \ldots, k$, the only actors are the foci of the relevant access sets and the information gates. So, we now consider how information is transferred from $\mathbf{v}_1$ to **x**. Again, three cases arise:

- $\mathbf{v}_1 = \mathbf{x}$. We are done.
- $\mathbf{v}_1$ initially spans to **x**. By condition (3.3) of Theorem 3, this means **x** is a subject, so apply Lemma 4 to get the desired result. Again, **x** is an information gate in this case.
- $\mathbf{v}_1$ rw-initially spans to **x**. Apply the take rule repeatedly to get an explicit write edge; then $\mathbf{v}_1$ applies the post rule to obtain the desired result.

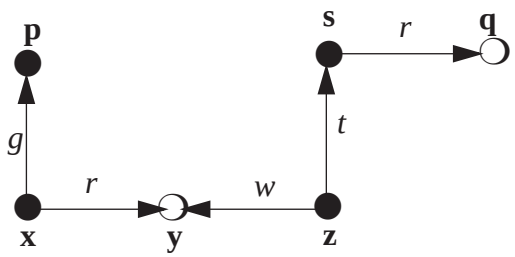


Figure 1. Sample Take-Grant protection graph demonstrating conspiracy in a single path graph.

Again, notice the only actors are the foci of the access sets and (where present) the information gates. This proves the claim. ■

As one would expect, these bounds are similar to the ones on the number of conspirators necessary and sufficient to steal rights. The difference lies in the definitions of "access set" and "information gate;" these include at least as many vertices in the *can•snoop* case as in the *can•steal* case. However, given a specific protection graph, computing the numbers k and l is of complexity comparable to the complexity of computing them in the *can•steal* case, since only a small number of new conditions in the definitions of "access set" and "information gate" must be tested.

At this point, let us take stock of what we have done by working a simple example. Consider the protection graph G in Figure 1.. Taking $\mathbf{u}_1 = \mathbf{p}$, $\mathbf{u}_2 = \mathbf{x}$, $\mathbf{u}_3 = \mathbf{z}$, and $\mathbf{u}_4 = \mathbf{s}$, we see that the predicate *can•know*(**p**, **q**, G) is true by Theorem 3. (Incidentally, so is *can•snoop*(**p**, **q**, G); in the conditions to Theorem 7, take $\mathbf{x} = \mathbf{x}' = \mathbf{p}$, $\mathbf{y}' = \mathbf{z}$, and $\mathbf{y} = \mathbf{q}$.) The graph is a single path graph of the variety we have been discussing, since information flows from **p** to **q** along the (sole) path between them. The following witness to *can•know*(**p**, **q**, G) demonstrates this:

(1) **z** takes (*r* to **q**) from **s**.

(2) **x** grants (*r* to **y**) to **p**.

(3) **p** and **z** use the post rule through **y** to add an implicit edge from **p** to **z**.

(4) **p** and **z** use the spy rule to obtain an implicit *r* edge from **p** to **q** through **z**.

In this graph, the only information gate is **p** (by condition (i) of the definition of information gate). The access sets of the four subjects are:

$I(\mathbf{p}) = \{ \mathbf{p} \}$ $T(\mathbf{p}) = \{ \mathbf{p} \}$ $I(\mathbf{z}) = \{ \mathbf{y}, \mathbf{z} \}$ $T(\mathbf{z}) = \{ \mathbf{q}, \mathbf{s}, \mathbf{z} \}$

$I(\mathbf{x}) = \{ \mathbf{p}, \mathbf{x} \}$ $T(\mathbf{x}) = \{ \mathbf{x}, \mathbf{y} \}$ $I(\mathbf{s}) = \{ \mathbf{s} \}$ $T(\mathbf{s}) = \{ \mathbf{q}, \mathbf{s} \}$

It is clear that these four access sets form a cover for G; it is equally clear that the sets $I(\mathbf{x})$, $T(\mathbf{x})$, $I(\mathbf{z})$, and $T(\mathbf{z})$ form a minimal access set cover for G. Applying Theorem 10, $k = 2$ and $l = 1$,

so a minimum of 3 actors are necessary for information to flow from **p** to **q**; similarly, by Theorem 13, 3 actors are sufficient. This agrees exactly with the witness presented above, which in fact used a minimal number of actors (**p**, **x**, and **z**).

4. Conspiracy in a General Graph

In the previous section, we restricted our attention to graphs in which *can•know* is true, and the only edges in the graph were those along which either rights or information were transferred. We shall now ease the latter restriction, and consider any valid Take-Grant protection graph in which the predicate *can•know* is true. Our goal is to derive a bound on the number of actors needed to produce a witness to *can•know*. We shall take the approach suggested by [14], again with suitable modifications.

In order to do this, we shall develop an analogue to the protection graph called an *acting graph*. Basically, this graph will consist of vertices corresponding to access sets in the original graph with edges corresponding to paths along which the focus of each access set can pass information by acting alone (that is, no other subject will have to act in a rule application to help the first transmit the information). In other words, this graph connects all actors with other subjects to which they can pass, or from which they can receive, information.

Given a protection graph G with subject vertices $\mathbf{v}_1, \ldots, \mathbf{v}_n$, we need to generate an acting graph G' with vertices $\mathbf{u}_1, \ldots, \mathbf{u}_n$. Each $\mathbf{u}_i$ has associated with it the access sets $I(\mathbf{v}_i)$ and $T(\mathbf{v}_i)$. Consider now under what circumstances information can be passed from a member of one access set to a member of another.

Let **y** be a vertex in an access set with focus **x**. There are five reasons **y** may be in that set:

- **y** = **x**;
- **x** initially spans to **y**;
- **x** terminally spans to **y**;
- **x** rw-initially spans to **y**; or
- **x** rw-terminally spans to **y**.

Define the set $\Delta(\mathbf{x}, \mathbf{x}')$ to be all vertices in $I(\mathbf{x}) \cap T(\mathbf{x}')$ *except* those vertices **y** which are information gates and the only reason **y** is in both $I(\mathbf{x})$ and $T(\mathbf{x}')$ is that the words associated with the paths **xy** and **x´y** are those that make **y** an information gate. This means the set Δ includes only those vertices to which the foci can pass information (or from which they can receive information) with the foci being the only actors.

To complete the construction of the acting graph G', we add a directed edge between $\mathbf{u}_i$ and $\mathbf{u}_j$ when $\Delta(\mathbf{v}_i, \mathbf{v}_j) \neq \varnothing$. (This corresponds to a bridge or connection existing between $\mathbf{v}_i$ and $\mathbf{v}_j$ in G.) We also define two special sets; let

$$\mathbf{I_x} = \{\ \mathbf{u}_i \mid \mathbf{v}_i = \mathbf{x} \text{ or } \mathbf{v}_i \text{ rw-initially spans to } \mathbf{x}\ \}$$

and

$$\mathbf{T_y} = \{\ \mathbf{u}_i \mid \mathbf{v}_i = \mathbf{y} \text{ or } \mathbf{v}_i \text{ rw-terminally spans to } \mathbf{y}\}$$

Since we intend to use the acting graph to derive a bound, we must first show that it accurately preserves the notion of sharing information.

Theorem 14. can•know(**x**, **y**, G) is true if and only if there is a path from some vertex $\mathbf{u}_a \in \mathbf{I_x}$ to some vertex $\mathbf{u}_b \in \mathbf{T_y}$.

Proof: ($\Rightarrow$) Let $\mathbf{v}_i$ be the vertex in G corresponding to the vertex $\mathbf{u}_i$ in G' (for $i = 1, \ldots, n$). We must consider two cases involving any vertex **z** in the definition of Δ above.

First, we restrict **z** to being an object in $T(\mathbf{v}_i) \cap I(\mathbf{v}_j)$. Note that the subjects in G correspond to vertices in G', and the edges between the vertices in G' correspond to words in $B \cup C$ in G, along which information flows from $\mathbf{v}_i$ to $\mathbf{v}_j$. So, applying Theorem 3, as *can•know*(**x**, **y**, G) is true, some $\mathbf{u}_a \in \mathbf{I_x}$ is connected to some $\mathbf{u}_b \in \mathbf{T_y}$.

Next, assume **z** is a subject in $T(\mathbf{v}_i) \cap I(\mathbf{v}_j)$. Let **z** be associated with $\mathbf{u}_a$. As **z** is a focus (since it is an information gate, and therefore the focus of an access set), it clearly has reason to be in $I(\mathbf{z})$ and $T(\mathbf{z})$; so $\{\mathbf{z}\} \subseteq \Delta(\mathbf{v}_i, \mathbf{z})$ and $\{\mathbf{z}\} \subseteq \Delta(\mathbf{z}, \mathbf{v}_j)$. Hence, by construction of G', there are paths between $\mathbf{u}_i$ and $\mathbf{u}_a$, and $\mathbf{u}_a$ and $\mathbf{u}_j$, so there is still a path between $\mathbf{u}_i$ and $\mathbf{u}_j$ (going through $\mathbf{u}_a$). Hence $\mathbf{u}_i$ and $\mathbf{u}_j$ are connected.

($\Leftarrow$) Assume there is a path from $\mathbf{u}_a$ to $\mathbf{u}_b$ with $\mathbf{u}_a = \mathbf{u'}_1, \ldots, \mathbf{u'}_n = \mathbf{u}_b$. By construction, $\mathbf{u}_{i+1}$ can pass information to $\mathbf{u}_i$, so by induction $\mathbf{u}_a$ can receive information from $\mathbf{u}_b$. Also, as $\mathbf{u}_b \in \mathbf{T_y}$, $\mathbf{u}_b$ can obtain information from **y**, and as $\mathbf{u}_a \in \mathbf{I_x}$, $\mathbf{u}_a$ can pass information to **x**. This means that *can•know*(**x**, **y**, G) is true. ■

We may now state and prove the desired result.

Theorem 15. Let n be the number of vertices on the shortest path from an element $\mathbf{u}_a \in \mathbf{I_x}$ to an element $\mathbf{u}_b \in \mathbf{T_y}$ in an acting graph G'. Then n actors are both necessary and sufficient to produce a witness to *can•know*(**x**, **y**, G).

Proof: (*Necessity*) Let $\mathbf{u}_a = \mathbf{u}'_1, \ldots, \mathbf{u}'_n = \mathbf{u}_b$ be vertices along a shortest path from $\mathbf{u}_a$ to $\mathbf{u}_b$, and let $\mathbf{v}'_i$ be the vertex in G corresponding to the vertex $\mathbf{u}'_i$ in G' (for $i = 1, \ldots, n$). If there exist only rwtg-paths in G from $\mathbf{v}'_i$ to $\mathbf{v}'_{i+1}$ ($1 \leq i < n$), the $\mathbf{v}'_i$ are foci of an access set cover for the path. By construction of G' there are no information gates and if $\mathbf{u}_a$ is not associated with $\mathbf{x}$, then the subject associated with $\mathbf{u}_a$ rw-initially or initially spans to $\mathbf{x}$. A similar argument holds for $\mathbf{u}_b$ and $\mathbf{y}$. By Theorem 10, n actors are necessary.

Now suppose there is an (induced) path in G' that is not in G. Even though redundant rule applications may occur, clearly duplicated vertices along a span affect the claim only if they reduce the number of required actors. We show this is not possible by contradiction. Suppose that actors $\mathbf{u}'_1, \ldots, \mathbf{u}'_{i-1}, \mathbf{u}'_{i+1}, \ldots, \mathbf{u}'_n$ could produce a witness. Then there is a vertex $\mathbf{z} \in T(\mathbf{v}_{i-1}) \cap I(\mathbf{v}_{i+1})$. As the $\mathbf{u}'_i$ are on the shortest path, there is no path between $\mathbf{u}'_{i-1}$ and $\mathbf{u}'_{i+1}$, so $\mathbf{z}$ is neither $\mathbf{v}'_{i-1}$ nor $\mathbf{v}'_{i+1}$, and further $\mathbf{z} \notin \Delta(\mathbf{v}_{i-1},\mathbf{v}_{i+1})$. Hence if $\mathbf{z}$ is an object, there is no word in $B \cup C$ between the vertices $\mathbf{v}_{i-1}$ and $\mathbf{v}_{i+1}$, so *can•know* is false by Theorem 3, whence $\mathbf{u}'_1, \ldots, \mathbf{u}'_{i-1}, \mathbf{u}'_{i+1}, \ldots, \mathbf{u}'_n$ cannot produce a witness. On the other hand, if $\mathbf{z}$ is a subject, it must be an information gate, in which case it must be an actor. In either case, the vertices $\mathbf{u}'_1, \ldots, \mathbf{u}'_{i-1}, \mathbf{u}'_{i+1}, \ldots, \mathbf{u}'_n$ cannot produce a witness without another vertex being added.

(*Sufficiency*) First, as $\mathbf{x}$ and $\mathbf{y}$ are distinct, and all the $\mathbf{v}'_i$ corresponding to the $\mathbf{u}'_i$ on the shortest path distinct, all spans between these vertices allow the appropriate sequence of rule applications exhibited in the Appendix to be applied, provided the foci of the access sets differ from their common elements. By inspecting the sequences, whenever a focus and a common element do coincide the rule whose application is prevented either provides a right already possessed, a right used in the subsequent rule application to acquire a right already possessed, or an implicit edge where one already exists. In these cases the rule application is unnecessary. Noting this, we need only induct on the spans corresponding to the edge of the shortest path using Lemma 11 to obtain the result. ■

In this section and the previous section, we very deliberately defined terms to capture the ability of a single node to pass information, or to prevent it from being passed; we then abstracted the instantiation of these terms to an acting graph. This is a generalization of Snyder's *conspiracy graph*, the derivation of which is similar but does not reflect information flows [14].

Let us apply these results to a simple protection graph. In Figure 2a, there are no information gates, and the access sets of the subjects are:

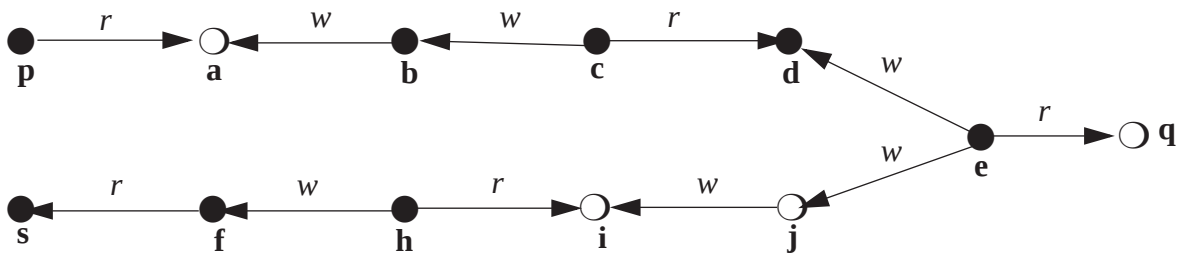


Figure 2a. A sample Take-Grant protection graph to demonstrate conspiracy in a general graph.

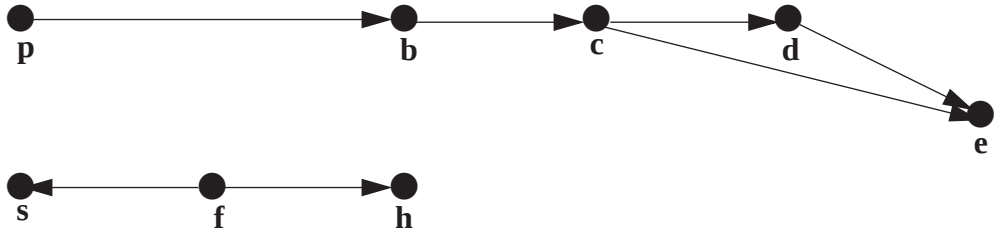


Figure 2b. The associated acting graph. For simplicity vertices are named as in the regular graph.

$I(\mathbf{p}) = \{ \mathbf{p} \}$	$T(\mathbf{p}) = \{ \mathbf{p}, \mathbf{a} \}$	$I(\mathbf{e}) = \{ \mathbf{d}, \mathbf{e}, \mathbf{j} \}$	$T(\mathbf{e}) = \{ \mathbf{e}, \mathbf{q}\}$
$I(\mathbf{b}) = \{ \mathbf{a}, \mathbf{b}\}$	$T(\mathbf{b}) = \{ \mathbf{b} \}$	$I(\mathbf{h}) = \{ \mathbf{f}, \mathbf{h} \}$	$T(\mathbf{h}) = \{ \mathbf{h}, \mathbf{i}\}$
$I(\mathbf{c}) = \{ \mathbf{b}, \mathbf{c}\}$	$T(\mathbf{c}) = \{ \mathbf{c}, \mathbf{d} \}$	$I(\mathbf{f}) = \{ \mathbf{f}\}$	$T(\mathbf{f}) = \{ \mathbf{f}, \mathbf{s}\}$
$I(\mathbf{d}) = \{ \mathbf{d}\}$	$T(\mathbf{d}) = \{ \mathbf{d} \}$	$I(\mathbf{s}) = \{ \mathbf{s}\}$	$T(\mathbf{s}) = \{ \mathbf{s}\}$

From these, we can construct $\Delta(\mathbf{x}, \mathbf{y})$ for each pair of subjects **x** and **y**; the nonempty ones are:

$\Delta(\mathbf{p}, \mathbf{b}) = \{ \mathbf{a} \}$ $\quad$ $\Delta(\mathbf{d}, \mathbf{e}) = \{ \mathbf{d} \}$

$\Delta(\mathbf{b}, \mathbf{c}) = \{ \mathbf{b} \}$ $\quad$ $\Delta(\mathbf{h}, \mathbf{f}) = \{ \mathbf{f} \}$

$\Delta(\mathbf{c}, \mathbf{d}) = \{ \mathbf{d} \}$ $\quad$ $\Delta(\mathbf{f}, \mathbf{s}) = \{ \mathbf{s} \}$

$\Delta(\mathbf{c}, \mathbf{e}) = \{ \mathbf{d} \}$

The resulting acting graph is shown in Figure 2b. By Theorem 3, *can*•*know*(**p**, **q**, *G*) is true (take $n = 5$, $\mathbf{x} = \mathbf{u}_1 = \mathbf{p}$, $\mathbf{u}_2 = \mathbf{b}$, $\mathbf{u}_3 = \mathbf{c}$, $\mathbf{u}_4 = \mathbf{d}$, and $\mathbf{u}_5 = \mathbf{e}$). Also, in G', $\mathbf{e} \in \mathbf{T_q}$ and $\mathbf{p} \in \mathbf{I_p}$, so some element of $\mathbf{I_p}$ is connected to some element of $\mathbf{T_q}$. This illustrates Theorem 14.

The following sequence of rule applications is a witness to *can*•*know*(**p**, **q**, *G*):

(1) **e** and **c** use the post rule through **d** to add an implicit read edge from **c** to **e**;

(2) **c** uses the pass rule to add an implicit read edge from **b** to **e**;

(3) **b** and **p** use the post rule through **a** to add an implicit read edge from **p** to **b**;

(4) **p** and **b** use the spy rule to add an implicit read edge from **p** to **e**;

(5) **p** and **e** use the spy rule to add an implicit read edge from **p** to **q**.

Four vertices (**b**, **c**, **e**, and **p**) act in this witness, and indeed the shortest path in G' between **p** and **e** contains four vertices. This illustrates Theorem 15.

Consider now **s** and **q**. According to Theorem 14, as they are not connected in G', *can•know*(**s**, **q**, G) should be false. As there is no rwtg-path from **h** to **e** with associated word in $B \cup C$, condition (3.3) of Theorem 3 fails, so *can•know*(**s**, **q**, G) is indeed false.

Finally, let us consider just the top part of this graph (from **p** to **q**), which is a single-path graph of the sort discussed in the previous section. There are no information gates, and the access sets with foci **p**, **b**, **c**, and **e** provide a complete cover for the subgraph. Hence by Theorem 10 and Theorem 13, four actors are necessary and sufficient to witness *can•know*(**p**, **q**, G), and our witness confirms this.

5. Comparison with Results for Theft of Rights

The similarity of the definitions of *can•steal* and *can•snoop* lead to the question of the relationship of these *de jure* and *de facto* results with the analogous *de jure* results in [14]; specifically, how different are the definitions, theorems and proofs, and how much more (or less) complex is it to determine bounds on the number of actors needed to steal information as opposed to steal rights?

The fundamental difference in the results presented here is the addition of extra conditions presenting more ways in which conspiracy can occur; for example, the *de jure* analogue to *access set* requires only that the focus initially or terminally span to every vertex in the set whereas here, we add those vertices to which the focus also rw-initially or rw-terminally spans. Most of the definitions in this work follow directly from their analogues; however, the changes add complexity to both the statements of the theorems and to the proofs. For example, the key theorem in [14] (Theorem 2 in this paper), which states necessary and sufficient conditions for rights to be stolen, requires checking for only three (simple) conditions; the analogue of that theorem for information transfer, Theorem 7, requires four (more complex) conditions to hold. The key construct in [14], the conspiracy graph, connects foci of access sets with edges showing paths along which rights can be transferred; the acting graph augments this to include a path along which information can be transferred as well.

The key difference in the conspiracy results lies in the acting graph. As rights can be transferred in either direction along a bridge, a conspiracy graph has undirected edges, because the vertices at the end of the path can share rights with one another. However, over a connection,

information can be transferred in one direction only; hence an acting graph has directed edges to represent the direction along which the information can flow. Note that if the connection between two vertices in a protection graph is a bridge, the corresponding edge in the acting graph will be bidirectional, to represent that information can be transferred in either direction over a bridge.

Consider a Take-Grant protection graph G in which the predicates $can\bullet steal(r, \mathbf{x}, \mathbf{y}, G)$ and $can\bullet snoop(\mathbf{x}, \mathbf{y}, G)$ are true. Let $A^R(\mathbf{y})$ be the set of nodes containing **y** and those nodes to which **y** initially or terminally spans, and let a *tg-sink* be a vertex with exactly two incident edges, both incoming and both labelled *t* or both labelled *g*. In [14], Snyder shows that a *conspiracy graph* can be constructed in a manner similar to the construction of an acting graph in section 4. Note that $A^R(\mathbf{y}) \subseteq I(\mathbf{y}) \cup T(\mathbf{y})$, and that a tg-sink is also an information gate. Hence, the conspiracy graph associated with G will be a (possibly improper) subgraph of the graph produced by replacing edges in the acting graph of G with undirected edges. So, in no case will stealing information require more conspirators than stealing rights; and if the acting graph contains a shorter path between the vertices associated with **x** and **y** than does the conspiracy graph, stealing information will require fewer conspirators.

6. Applications

We can apply our results to a realistic situation by considering the flow of information throughout a small local area network using the TCP/IP protocol suite. We focus on the use of the File Transfer Protocol, or *ftp*. We state the problem quite simply: a computer (subject) **p** has a file **x** containing private information. A copy of it is found on computer **y**. Our question is whether the file could have been transferred using a series of *ftp* connections, and if so, how many conspirators were necessary and sufficient?

First, we make several simplifying assumptions:

1. All *ftp* connections and accesses are anonymous. (This ability is a feature of the standard protocol.) Were this assumption not made, we would need to track user identities and authorities; while this is straightforward, it adds complexity which detracts from the issue under study, which is the abstraction of the network into a Take-Grant style model.
2. The network is not fully connected; again, this models real local area networks, on which many hosts choose not to provide *ftp* connections.

3. Only hosts directly connected to the network are involved. We will relax this assumption with the introduction of proxy servers later.

6.1. Basic Abstraction

The ftp protocol requires that objects be placed in a central area; anonymous accesses using that protocol give the remote user the ability to read (and hence download) those objects. Further, even though access may be granted, the grantor has the power to turn off all access at any time. This means all transfers of information are to be along implicit edges, which dictates the following abstract representation:

1. All hosts are represented as subject vertices, and all files as object vertices;
2. Permission for an entity on host ***x*** to retrieve files from host ***y*** via anonymous *ftp* is represented by an explicit edge labelled *r* (read) from ***x*** to ***y***.
3. Accessibility of a file ***f*** on host ***x*** to anonymous *ftp* is represented by an explicit read edge from ***x*** to ***f***.

This means that the ability to transfer file **f** from host **x** to **y** will be represented by an implicit edge from **y** to **f**. As in other *de facto* situations, this does not mean that the transfer must take place or has taken lace; it merely indicates a path along which the transfer could, or could have, taken place. Hence our interest.

6.2. Basic Examples

Consider first a situation in which there are four sites offering anonymous *ftp* for reading only (no writing): **p**, **q**, **s**, and **v**. The file **f** contains proprietary data and resides on **p**. The hosts **p**, **q**, and **s** are fully connected, but **v** can only access **s**. In the course of a police investigation into industrial espionage, a copy of file **f** is found on host **v**. The question is, which hosts could have conspired to put it there?

The abstraction of this situation is in Figure 3a. We wish to know the sets of conspirators who may have copied **f**. So, we apply the technique of the earlier section.

The access sets for the subjects involved are:

$I(\mathbf{p}) = \{ \mathbf{p} \}$ $T(\mathbf{p}) = \{ \mathbf{p}, \mathbf{q}, \mathbf{s} \}$ $I(\mathbf{s}) = \{ \mathbf{s} \}$ $T(\mathbf{s}) = \{ \mathbf{p}, \mathbf{q}, \mathbf{s} \}$

$I(\mathbf{q}) = \{ \mathbf{q} \}$ $T(\mathbf{q}) = \{ \mathbf{p}, \mathbf{q}, \mathbf{s} \}$ $I(\mathbf{v}) = \{ \mathbf{v} \}$ $T(\mathbf{v}) = \{ \mathbf{s}, \mathbf{v} \}$

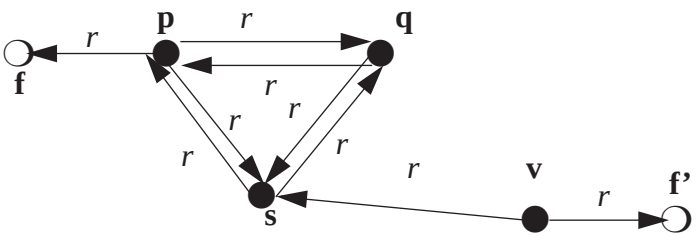


Figure 3a. The graphical representation of the network configuration. Here, **f'** is the illegitimate copy of the file **f**.

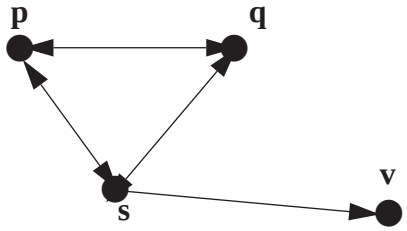


Figure 3b. The corresponding acting graph. For simplicity, vertices are named as in the regular graph.

From this, we construct the sets $\Delta(\mathbf{a}, \mathbf{b})$ for each pair of vertices **a** and **b**:

$\Delta(\mathbf{p}, \mathbf{q}) = \{ \mathbf{p} \}$	$\Delta(\mathbf{p}, \mathbf{s}) = \{ \mathbf{p} \}$	$\Delta(\mathbf{p}, \mathbf{v}) = \varnothing$
$\Delta(\mathbf{q}, \mathbf{p}) = \{ \mathbf{q} \}$	$\Delta(\mathbf{q}, \mathbf{s}) = \{ \mathbf{q} \}$	$\Delta(\mathbf{q}, \mathbf{v}) = \varnothing$
$\Delta(\mathbf{s}, \mathbf{p}) = \{ \mathbf{s} \}$	$\Delta(\mathbf{s}, \mathbf{q}) = \{ \mathbf{s} \}$	$\Delta(\mathbf{s}, \mathbf{v}) = \varnothing$
$\Delta(\mathbf{v}, \mathbf{p}) = \varnothing$	$\Delta(\mathbf{v}, \mathbf{q}) = \varnothing$	$\Delta(\mathbf{v}, \mathbf{s}) = \varnothing$

The acting graph is shown in Figure 3b. We note that $\mathbf{I_f} = \{ \mathbf{p} \}$ and $\mathbf{T_{f'}} = \{ \mathbf{v} \}$. By Theorem 15, the minimum number of actors necessary and sufficient to move the information from **p** to **v** is 3. Noting also that the acting graph captures the paths along which the information is transferred, this means that **p**, **s**, and **v** are the conspirators for the witness to this transfer.

6.3. General Example

We now present a more sophisticated example, in which many connections are one-way, and examine how many conspirators are needed to move information. Consider the situation in Figure 4a. Note that here rights for anonymous *ftp* are constrained; some uploading (*w* rights) as well as downloading (*r* rights) is allowed, and not all the vertices are directly connected. As before, **x** is the file the contents of which is secret, but during an investigation, two copies **x'** and **x''** have been found on competitors' hosts. The problem is to find a lower bound on the number of hosts involved in the transfer.

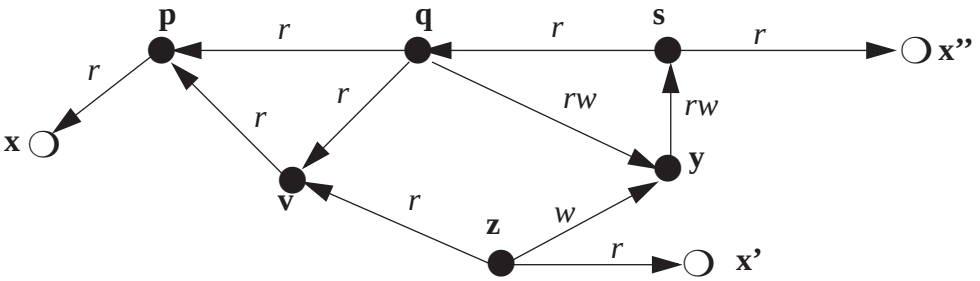


Figure 4a. The graphical representation of the network configuration. Here, **x'** and **x"** are the illegitimate copies of the file **x**.

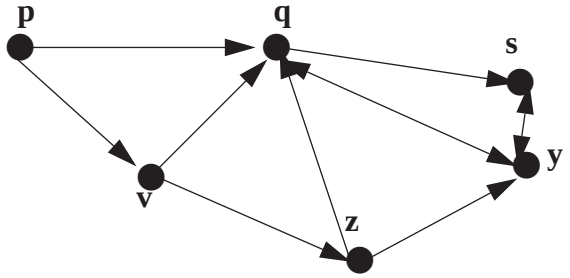


Figure 4b. The corresponding acting graph. For simplicity, vertices are named as in the regular graph.

As before, we build our access sets:

I(**p**) = { **p** }	T(**p**) = { **x** }	I(**z**) = {**z**, **y**}	T(**z**) = { **z**, **v**, **x'** }
I(**q**) = { **q**, **y** }	T(**q**) = { **p**, **q**, **v**, **y** }	I(**v**) = { **v**}	T(**v**) = { **p**, **v** }
I(**s**) = { **s**}	T(**s**) = { **q**, **s**, **x"** }	I(**y**) = { **s**, **y** }	T(**y**) = { **s**, **y** }

From this, we construct the sets Δ(**a**, **b**) for each pair of vertices **a** and **b**. The non-empty sets are:

Δ(**p**, **q**) = { **p** }	Δ(**p**, **v**) = { **p** }	Δ(**q**, **s**) = { **q** }	Δ(**q**, **y**) = { **y** }
Δ(**s**, **y**) = { **s** }	Δ(**z**, **q**) = { **y** }	Δ(**z**, **y**) = { **y** }	
Δ(**v**, **q**) = { **v** }	Δ(**v**, **z**) = { **v** }	Δ(**y**, **q**) = { **y** }	Δ(**y**, **s**) = { **s**}

The acting graph is shown in Figure 3b.

Consider the information flow from **x** to **x'**. In this case, $\mathbf{I_x}$ = { **p** } and $\mathbf{T_{x'}}$ = { **z** }. The path between **p** and **z** has three vertices (**p**, **v**, and **z**) in Figure 3b. So, by Theorem 15, the minimum number of actors necessary and sufficient to move the information from **x** to **x'** is 3 (with **p**, **v**, and **u** being the three actors)

Next, let us look at the information flow from **x** to **x"**. Here, $\mathbf{I_x}$ = { **p** } and $\mathbf{T_{x"}}$ = { **s** }. As before, the path between **p** and **s** has three vertices (**p**, **q**, and **s**) in Figure 3b. So the minimum num-

ber of actors necessary and sufficient to move information from **x** to **x''** is also 3 (with **p**, **q**, and **s** being the actors).

Note that this does not mean that the particular actors must have been involved in the transfer of information; it simply means that they could have been. Specifically, information may have been transferred along any directed path in the acting graph. In this particular example, the two enumerated paths were the shortest, but longer paths may have been used. Information can flow from **x** to **x''** along the path **pvqys**; if this had occurred, 5 actors would be involved.

6.4. Proxies

A *proxy* is a system through which all requests for *ftp* access is filtered; such programs are most often found on firewalls. They act as though the files were stored on the firewall, passing commands on to the real *ftp* server. The remote host never sees the host behind the proxy.

An example configuration is in Figure 5a. Here, vertex **c** is the proxy, and it has authority to access any file set up for retrieval in the local area network (here, composed of hosts represented by vertices **d** and **e**). As this authority depends only on the existence of the target file, and not on **d** or **e** passing the information to the proxy, the rights of **c** over **d** and **e** are represented by take edges. (An alternate situation is where **d** or **e** would need to co-operate with **c** to make the file available to hosts outside the local area network. In this case, the edges between **c** and **d** and **c** and **e** would be read edges. We use the take form to illustrate a situation involving *de jure* and *de facto* rules.)

As before, we build our access sets:

$I(\mathbf{a}) = \{ \mathbf{a} \}$	$T(\mathbf{a}) = \{ \mathbf{a}, \mathbf{c} \}$	$I(\mathbf{d}) = \{ \mathbf{d} \}$	$T(\mathbf{d}) = \{ \mathbf{d}, \mathbf{f} \}$
$I(\mathbf{b}) = \{ \mathbf{b} \}$	$T(\mathbf{b}) = \{ \mathbf{b} \}$	$I(\mathbf{e}) = \{ \mathbf{e} \}$	$T(\mathbf{e}) = \{ \mathbf{e}, \mathbf{g} \}$
	$I(\mathbf{c}) = \{ \mathbf{c}, \mathbf{b} \}$	$T(\mathbf{c}) = \{ \mathbf{c} \}$	

From this, we construct the sets $\Delta(\mathbf{a}, \mathbf{b})$ for each pair of vertices **a** and **b**. The non-empty sets are:

$$\Delta(\mathbf{c}, \mathbf{a}) = \{ \mathbf{c} \} \qquad \Delta(\mathbf{c}, \mathbf{b}) = \{ \mathbf{b} \}$$

The acting graph is shown in Figure 3b. The relationships between the objects and subjects is summarized by $\mathbf{T_f} = \{ \mathbf{d}, \mathbf{c} \}$, $\mathbf{T_g} = \{ \mathbf{e}, \mathbf{c} \}$, $\mathbf{I_h} = \{ \mathbf{a} \}$, and $\mathbf{I_{h'}} = \{ \mathbf{c} \}$. Now consider two cases. If **h'** is a copy of **f**, we note that **c** is in both $\mathbf{T_f}$ and $\mathbf{I_{h'}}$. Hence there is a single vertex on the path between an element of $\mathbf{T_f}$ and an element of $\mathbf{I_{h'}}$, and there are no information gates. So by Theorem

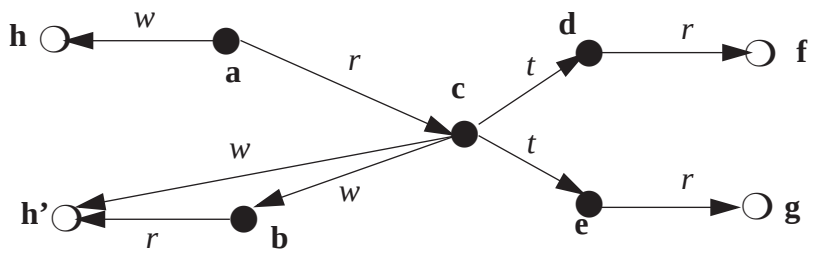


Figure 5a. An *ftp* proxy server guarding access to files on a local network. Here, **d** and **e** are on the local area network guarded by proxy **c**, and **a** and **b** are on other networks connected to the local area network. Note that **b** has writing enabled for anonymous *ftp* (the write edge from **c** to **h'**).

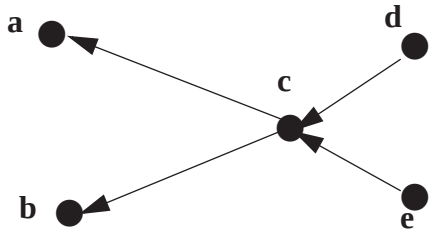


Figure 5b. The corresponding acting graph. For simplicity, vertices are named as in the regular graph.

15, the minimum number of actors necessary is 1, and the following witness to *can•know*(**h'**, **f**, *G*) substantiates this result:

(1) **c** takes (*r* to **f**) from **d**;

(2) **c** uses the pass rule to add an implicit read edge from **h'** to **f**.

Similarly, if **h** is a copy of **f**, the shortest path between an element in $\mathbf{T_f}$ and an element in $\mathbf{I_h}$ contains 2 vertices (**a** and **c**). So we need at least two actors to witness *can•know*(**h**, **f**, *G*). A witness to this is:

(1) **c** takes (*r* to **f**) from **d**;

(2) **a** and **c** use the spy rule to add an implicit read edge from **a** to **f**

(2) **a** uses the pass rule to add an implicit read edge from **h** to **f**.

Clearly, **a** and **c** must act. Note that this is clear by inspection of the graph. Since there is only an incoming write edge to **h**, only the find or post rule can add an outgoing implicit read edge. As the write edge comes from **a**, and as **a** has no incident take or grant edges, **a** must act in a find or post rule. As **a** has no incident take or grant edges, no explicit edges can be added to **a** by another vertex. Further, as **a** has no read edges to **f**, by inspection of the *de jure* and *de facto* rules, at least one other vertex must be an actor to provide an implicit read edge from **a** to **f**. Hence at least two vertices must be involved. By Theorem 15, 2 vertices are also sufficient.

7. Conclusion

This paper has explored several aspects of information transfer in the Take-Grant protection model. Building on the notion of information transfer, the information flow conspiracy results not only put a bound on the number of vertices necessary and sufficient to transfer information but also provide a better understanding about how information can flow about a system. Further, we saw that these results can be applied to a model of network services in which paths of transfer of information can be identified. From this, those subjects which could not be involved in the transfer can be identified.

This suggests a very interesting question: can those subjects which *must* be involved in a transfer of information be identified? The intuitive answer (those vertices which lie on all paths between the relevant vertices in the acting graph) does not account for edges deleted after the transfer but before the analysis. This observation is critical.

One of the problems in the application of a theoretic analysis to a practical situation is the issue of correct abstraction: does the abstract model properly capture the relevant characteristics of the system being modelled? If so, the model is a valuable tool for analyzing the real situation. If not, it may or may not be a valuable tool, but can do no more than suggest possibilities instead of provide certainties. The examples in section 7 provide an excellent illustration of this point.

If the network which is modelled in the examples in Section 6 does not have the ability to delete edges and rights, then the information transfers require that the paths in the acting graphs be used. But networks do allow systems and users to delete rights; hence, if the information has been transferred, it is possible that one or more segments of the path along which the transfer occurred was deleted in order to hide the conspirators. In that case, another path not in the acting graph might have been used. In other words, the model captures the current system state; if the transfer occurred in a prior (different) state, no conclusions can be drawn from the newer, modified state. Inferences may be made, and used as starting points for other types of analyses, however; and if the prior state can be reconstructed (perhaps because inferences can be made from the structure of the existing protection graph and, if available, knowledge of the way the system has evolved in general), one could then draw conclusions about the conspirators.

This suggests two avenues of research. The first is to determine the effect of deletions upon the set of possible witnesses to a theft. Perhaps a structure with certain (unknown) characteristics implies that some set of subjects could not have acted in witnesses to thefts, even when delete rule

can be used. Secondly, could the above results be used to design systems in which the sets of actors necessary for a theft are so large that such a conspiracy is unrealistic?

Acknowledgments: Thanks to Larry Snyder, who first interested me in the Take Grant Protection Model and whose work on it has been the basis for many of the results presented here. Thanks also to Virgil Gligor, who first suggested applying this model to networks, and to Becky Bace and Kevin Ziese, who encouraged me to look at thefts in a less abstract context. Portions of this work were supported by grant NAG2-480 from the National Aeronautics and Space Administration to Dartmouth College, a Dartmouth College Faculty Fellowship, and grant TDS 95-140 from Trident Data Systems to the University of Caifornia at Davis. Portions of this work were done while the author was with the Department of Mathematics and Computer Science at Dartmouth College.

References

[1] P. Ammann and R. Sandhu, "Safety Analysis for the Extended Schematic Protection Model," *Proc. of the 1991 IEEE Symp. on Security and Privacy* (May 1991), 87-97.

[2] M. Bishop, "Theft of Information in the Take Grant Protection Model," *Journal of Computer Security*, to appear.

[3] M. Bishop, "Hierarchical Take-Grant Protection Systems," *Proc. 8th Symp. on Operating Systems Principles* (Dec. 1981), 107-123

[4] M. Bishop and L. Snyder, "The Transfer of Information and Authority in a Protection System," *Proc. 7th Symp. on Operating Systems Principles* (Dec. 1979), 45-54.

[5] D. Brewer and M. Nash, "The Chinese Wall Security Policy," *Proc. 1989 IEEE Symp. on Security and Privacy* (May 1989) 206-214.

[6] D. Clark and D. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proc. of the 1987 IEEE Symp. on Security and Privacy* (Apr. 1987), 184-194.

[7] M. Harrison, W. Ruzzo, and J. Ullman, "Protection in Operating Systems," *CACM 19*, 8 (Aug. 1976), 461-471

[8] A. Jones, "Protection Mechanism Models: Their Usefulness," in *Foundations of Secure Computing*, Academic Press, New York City, NY (1978), 237-254.

[9] A. Jones, R. Lipton, and L. Snyder, "A Linear Time Algorithm for Deciding Security," *Proc. 17th Annual Symp. on the Foundations of Computer Science* (Oct. 1976), 33-41.

[10] R. Lipton and L. Snyder, "A Linear Time Algorithm for Deciding Subject Security," *J. ACM. 24*, 3 (Jul. 1977), 455-464.

[11] J. McLean, "The Algebra of Security," *Proc. of the 1988 IEEE Symp. on Security and Privacy* (Apr. 1988), 2-8.

[12] J. McLean, "Security Models and Information Flow," *Proc. of the 1990 IEEE Symp. on Security and Privacy* (May 1990), 180-187.

[13] L. Snyder, "On the Synthesis and Analysis of Protection Systems," *Proc. Sixth Symp. on Operating Systems Principles* (Nov. 1977), 141-150.

[14] L. Snyder, "Theft and Conspiracy in the Take-Grant Protection Model," *JCSS 23*, 3 (Dec. 1981), 333-347.

[15] J. Wittbold and D. Johnson, "Information Flow in Nondeterministic Systems," *Proc. of the 1990 IEEE Symp. on Security and Privacy* (May 1990), 144-161.

[16] M. Wu, "Hierarchical Protection Systems," *Proc. 1981 Symp. on Security and Privacy* (Apr. 1981), 113-123.

8. Appendix

The following analysis of the possible paths between two vertices **x** and **y** in a straight line graph *G* allows the derivation of the paths for information gates. Note that if three vertices **x**, **y**, and **z** are involved, $\mathbf{x} = \mathbf{v}_{i-1}$, $\mathbf{z} = \mathbf{v}_i$, and $\mathbf{y} = \mathbf{v}_{i+1}$, and all are subjects. In each case, we must show that passing information through the gate requires the active cooperation of the gate. We note that this requires us examining the case where **x** is passive, *i.e.*, effectively an object.

Case 1. $\mathbf{x} = \mathbf{v}_0$, $\mathbf{y} = \mathbf{v}_1$. We show when **x** must act to receive information.

y *to* **x** *terminal*

If **x** is an object, the word associated with the path between **x** and **y** is not an rw-initial span, and so the predicate is false. If **x** is a subject, the following is a witness:

(1) **x** creates (*rw* to new vertex) **v**.

(2) **y** takes (*r* to **v**) from **x**.

(3) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **v**.

This verifies that *can•know*(**x**, **y**, G_0) is true. Note that both vertices **x** and **y** must act, so **x** is an information gate.

y *to* **x** *rw-terminal*

If **x** is an object, the word associated with the path between **x** and **y** is not an initial span, and so the predicate is false. If **x** is a subject, the path from **y** to **x** is not in $B \cup C$, the predicate $can\bullet know(\mathbf{x}, \mathbf{y}, G_0)$ is false and hence **x** cannot be an information gate.

y *to* **x** *initial*

If **x** is an object, the word associated with the path between **x** and **y** is not an rw-initial span, and so the predicate $can\bullet know(\mathbf{x}, \mathbf{y}, G_0)$ is false. If **x** is a subject, the following is a witness:

(1) **y** creates (*rw* to new vertex) **v**.

(2) **y** grants (*r* to **v**) from **x**.

(3) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **v**.

This verifies that $can\bullet know(\mathbf{x}, \mathbf{y}, G_0)$ is true. Note that all both vertices **x** and **y** must act, so **x** is an information gate.

y *to* **x** *rw-initial*

If **x** is an object, the word associated with the path between **x** and **y** is an rw-initial span, and so the predicate $can\bullet know(\mathbf{x}, \mathbf{y}, G_0)$ is true. Hence **x** need not act, and is not an information gate.

Case 2. $\mathbf{x} = \mathbf{v}_{i-1}$, $\mathbf{z} = \mathbf{v}_i$, and $\mathbf{y} = \mathbf{v}_{i+1}$. We show when **z** must act to pass information.

x *to* **z** *terminal*, **y** *to* **z** *terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **z** creates (*rw* to new vertex) **v**.

(2) **x** takes (*r* to **v**) from **z**.

(3) **y** takes (*w* to **v**) from **z**.

(4) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that $can\bullet know(\mathbf{x}, \mathbf{y}, G_0)$ is true. Note that all three vertices **x**, **y**, and **z** must act, and so **z** is an information gate.

x *to* **z** *terminal*, **y** *to* **z** *initial*

The following is a witness whether or not **z** is a subject:

(1) **y** creates (*rw* to new vertex) **v**.

(2) **y** grants (*r* to **v**) to **z**.

(3) **x** takes (*r* to **v**) from **z**.

(4) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. In this case, only **x** and **y** need act, and so **z** is not an information gate.

x *to* **z** *terminal*, **y** *to* **z** *rw-initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **z** creates (*rw* to new vertex) **v**.

(2) **x** takes (*r* to **v**) from **z**.

(3) **x** and **z** use the post rule to obtain an implicit *r* edge from **x** to **z** through **v**.

(4) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. Here, **x**, **y** and **z** need to act, so **z** is an information gate.

x *to* **z** *terminal*, **y** *to* **z** *rw-terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **y** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *initial*, **y** *to* **z** *terminal*

The following is a witness whether or not **z** is a subject:

(1) **x** creates (*rw* to new vertex) **v**.

(2) **x** grants (*rw* to **v**) to **z**.

(3) **y** takes (*r* to **v**) from **z**.

(4) **y** takes (*w* to **v**) from **z**.

(5) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. Again, only **x** and **y** must act, and so **z** is not an information gate.

x *to* **z** *initial*, **y** *to* **z** *initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **x** creates (*rw* to new vertex) **v**.

(2) **y** creates (*rw* to new vertex) **w**.

(2) **x** grants (*w* to **v**) to **z**.

(3) **y** grants (*r* to **w**) to **z**.

(4) **x** and **z** use the post rule to obtain an implicit *r* edge from **x** to **z** through **v**.

(5) **x** and **z** use the spy rule to obtain an implicit *r* edge from **x** to **w** through **z**.

(6) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **w**.

This verifies that *can•know*(**x**, **y**, G_0) is true. As all of **x**, **y**, and **z** must act, **z** is an information gate.

x *to* **z** *initial*, **y** *to* **z** *rw-initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **x** creates (*rw* to new vertex) **v**.

(2) **x** grants (*w* to **v**) to **z**.

(3) **x** and **z** use the post rule to obtain an implicit *r* edge from **x** to **z** through **v**.

(4) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. Again, **x**, **y** and **z** need to act, so **z** is an information gate.

x *to* **z** *initial*, **y** *to* **z** *rw-terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **x** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *rw-terminal*, **y** *to* **z** *terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **z** creates (*rw* to new vertex) **v**.

(2) **y** takes (*w* to **v**) from **z**.

(3) **y** and **z** use the post rule to obtain an implicit *r* edge from **z** to **y** through **v**.

(4) **x** and **z** use the spy rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. Once more, **x**, **y** and **z** need to act, and so **z** is an information gate.

x *to* **z** *rw-terminal*, **y** *to* **z** *initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the following is a witness:

(1) **y** creates (*rw* to new vertex) **v**.

(2) **y** grants (*r* to **v**) to **z**.

(3) **y** and **z** use the post rule to obtain an implicit *r* edge from **z** to **y** through **v**.

(4) **x** and **y** use the spy rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. All of **x**, **y** and **z** need to act, and so **z** is an information gate.

x *to* **z** *rw-terminal*, **y** *to* **z** *rw-terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **y** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *rw-terminal*, **y** *to* **z** *rw-initial*

The following is a witness whether or not **z** is a subject:

(1) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **z**.

This verifies that *can•know*(**x**, **y**, G_0) is true. This time, only **x** and **y** need act , and so **z** is not an information gate.

x *to* **z** *rw-initial*, **y** *to* **z** *terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **x** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *rw-initial*, **y** *to* **z** *initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **x** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *rw-initial*, **y** *to* **z** *rw-terminal*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **x** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

x *to* **z** *rw-initial*, **y** *to* **z** *rw-initial*

If **z** is an object, the word associated with the path between **x** and **y** is not in the set $B \cup C$ and so the predicate is false. If **z** is a subject, the word associated with the path between **x** and **z** is not in the set $B \cup C$ and so the predicate is false. In either case, the predicate *can•know*(**x**, **y**, G_0) is false, and so **z** is not an information gate.

Case 3. **y**= $\mathbf{v}_{n-1}$, **x**= $\mathbf{v}_n$. We show when **x** must act to send information.

y *to* **x** *terminal*

If **x** is an object, the word associated with the path between **x** and **y** is not an rw-initial span, and so the predicate is false. If **x** is a subject, the following is a witness:

(1) **x** creates (*rw* to new vertex) **v**.

(2) **y** takes (*r* to **v**) from **x**.

(3) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **v**.

This verifies that *can•know*(**y**, **x**, G_0) is true. Note that both vertices **x** and **y** must act, so **x** is an information gate.

y *to* **x** *initial*

If **x** is an object, the word associated with the path between **x** and **y** is not an rw-terminal span, and so the predicate *can•know*(**y**, **x**, G_0) is false. If **x** is a subject, the following is a witness:

(1) **y** creates (*rw* to new vertex) **v**.

(2) **y** grants (*r* to **v**) from **x**.

(3) **x** and **y** use the post rule to obtain an implicit *r* edge from **x** to **y** through **v**.

This verifies that *can•know*(**y**, **x**, G_0) is true. Note that both vertices **x** and **y** must act, so **x** is an information gate.

y *to* **x** *rw-terminal*

If **x** is an object, the word associated with the path between **x** and **y** is an rw-terminal span, and so the predicate *can•know*(**y**, **x**, G_0) is true. If **x** is a subject, the path from **y** to **x** is in $B \cup C$, so the predicate *can•know*(**y**, **x**, G_0) is true. In both cases, only **y** need act, so **x** cannot be an information gate.

y *to* **x** *rw-initial*

If **x** is an object, the word associated with the path between **x** and **y** is not an rw-terminal span, and so the predicate *can•know*(**y**, **x**, G_0) is false. If **x** is a subject, the path from **y** to **x**

is not in $B \cup C$, so the predicate *can•know*(**y**, **x**, G_0) is false. In neither case can **x** be an information gate.