1980

Theft and Conspiracy in the Take-Grant Protection Model

Lawrence Snyder

Report Number:
80-361
THEFT AND CONSPIRACY IN THE
TAKE-GRANT PROTECTION MODEL*

LAWRENCE SNYDER

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD-TR 361

*This research was funded in part by National Science Foundation Grant
MCS77-12517 and Office of Naval Research Contract #N00014-75-C-0752 and
was performed while the author was at the Department of Computer Science,
Yale University, New Haven, CT.
1. Introduction

Models of protection in computer systems usually possess two components: a finite, labeled, directed two color graph representing the protection state of an operating system and a finite set of graph transformation rules with which the protection state may be changed. Harrison, Ruzzo and Ullman demonstrated [1] that the safety problem for a very general protection model is undecidable, i.e., no algorithm could decide, given a protection graph and a set of transformation rules, whether an edge with a particular label is ever added to the graph. The Take-Grant Model [2,3,4] has been developed in response to this negative result in order to study such questions for a particular set of transition rules. Linear-time algorithms to test safety-like problems have been found [2,3] for the Take-Grant transition rules. Although the model is simple enough to permit linear time decision procedures, it is rich enough to implement many sharing relationships [4]. Here we concentrate on the formal development supporting the motivational and interpretive treatments given in [4,5].

First, we characterize the class of graphs that can be created with the Take-Grant rules. Next, the can-steal predicate, first introduced in a limited form [4], is developed in full generality making it applicable to the common situation of "stealing files."

Another main topic is that of quantifying the amount of "cooperation" required to share or steal rights. By the amount of "cooperation" we mean the number of users (i.e., subject vertices in the model) required to initiate rules in order for a particular edge to be added to a graph. This concept has been called "conspiracy" in [2]. Exact conspiracy measurements
for arbitrary protection graphs are derived and an algorithm for discovering minimum conspiracy is presented.

2. The Take-Grant Model

The definitions for the Take-Grant model follow earlier treatments [2,3,4] differing in only inessential ways.*

Fix a finite alphabet of labels \( R = \{r_1, \ldots, r_m\} \cup \{t,g\} \) called rights containing two distinguished elements; "t" is mnemonic for "take" and "g" is mnemonic for "grant." A protection graph is a finite, directed, loop-free, two color graph with edges labeled by nonempty subsets of \( R \). (Braces around subsets are elided.) Solid vertices, •, are called subjects, empty vertices, ◦, are called objects; vertices of either type are denoted by •.

Four rewriting rules are defined to enable a protection graph to change:

**Take:** Let \( x, y, \) and \( z \) be distinct vertices in a protection graph \( G \) such that \( x \) is a subject. Let there be an edge from \( x \) to \( y \) labeled \( \gamma \) such that "t" \( \in \gamma \), an edge from \( y \) to \( z \) labeled \( \beta \) and \( a \subseteq \beta \). Then the take rule defines a new graph \( G' \) by adding an edge to the protection graph from \( x \) to \( z \) labeled \( a \). Graphically,

\[
\begin{array}{c}
\text{Take:} \\
(\text{The rule can be read: } x \text{ takes } (\alpha \text{ to } z) \text{ from } y.)
\end{array}
\]

**Grant:** Let \( x, y, \) and \( z \) be distinct vertices in a protection graph \( G \) such that \( x \) is a subject. Let there be an edge from \( x \) to \( y \) labeled \( \gamma \) such that "g" \( \in \gamma \), an edge from \( x \) to \( z \) labeled \( \beta \), and \( a \subseteq \beta \). The grant rule defines a new graph \( G' \) by adding an edge from \( y \) to \( z \) labeled \( a \). Graphically,

\[
\begin{array}{c}
\text{Grant:} \\
(\text{The rule can be read: } x \text{ grants } (\alpha \text{ to } z) \text{ to } y.)
\end{array}
\]

*Specifically, the "call" rule of [2] has been dropped, \( r \) and \( w \) labels (used in [2]), are replaced by \( t \) and \( g \), respectively, and "inert" rights [5,6] are permitted.
Create: Let \( x \) be any subject vertex in a protection graph \( G \) and let \( \alpha \) be a nonempty subset of \( R \). Create defines a new graph \( G' \) by adding a new vertex \( n \) to the graph and an edge from \( x \) to \( n \) labeled \( \alpha \). Graphically,

\[
\bullet \xrightarrow{\alpha} n
\]

The rule can be read: "\( x \) creates (\( \alpha \) to) new \{subject\} \( n \)."

Remove: Let \( x \) and \( y \) be any distinct vertices in a protection graph \( G \) such that \( x \) is a subject. Let there be an edge from \( x \) to \( y \) labeled \( \beta \), and let \( \alpha \) be any subset of rights. Then remove defines a new graph \( G' \) by deleting the \( \alpha \) labels from \( \beta \). If \( \beta \) becomes empty as a result, the edge itself is deleted. Graphically,

\[
\bullet \xrightarrow{\beta} \bullet \xrightarrow{\alpha} \bullet
\]

The rule can be read: "\( x \) removes (\( \alpha \) to) \( y \)."

In these rules, \( x \) is called the initiator.

Application of rule \( \rho \) to graph \( G \) is denoted by \( G \xrightarrow{\rho} G' \). The reflective transitive closure of this relation is denoted \( G^+ \). The notation \( x \xrightarrow{G} y \) abbreviates "there exists an edge from \( x \) to \( y \) in \( G \) labeled \( \gamma \) and \( \alpha \subseteq \gamma \)." Figure 1 illustrates the use of the rules. Although there are additional concepts to be introduced, the development thus far is adequate for proving a characterization result.

3. Take-Grant Definable Graphs

It has been argued [4] that the protection graphs actually used in an operating system will be generated by a fixed set of rule protocols, e.g., by the operating system supervisor, editors, compilers, etc.

Hence, it is important to know what class of graphs can be generated by

*Dashed lines are used in illustrations as a visual aid. Also, even though there is only one directed edge from any vertex \( a \) to any vertex \( b \), we occasionally draw two to emphasize changes in labelling.
Figure 1: Vertex a acquires g rights to b, i.e., g is added to the label on the a to b edge. The rule applications may be read:

- a creates (tg to) new object d,
- a grants (g to d) to c,
- c grants (g to b) to d,
- a takes (g to b) from d,
the Take-Grant rules. Since all of the rule applications require that
the initiator be a subject, we consider the graphs reachable from a single
subject. Notice that an "all object" graph is impossible since vertices
cannot be deleted. A complete characterization is presented in the next
theorem.

**Theorem 3.1:** Let $G_0$ be a protection graph containing exactly
one subject vertex and no edges. Then $G_0 \models G$ if and only
if $G$ is a finite, directed, loop-free, two color graph
with edges labeled from nonempty subsets of $\mathbb{R}$ such that
at least one subject has no incoming edges.

**Proof:** Let $v$ be the initial subject, and $G_0 \models G$. After reviewing the
rule definition, one sees that $G$ is obviously finite, directed, loop-free
and two colored with the indicated labelling. Since vertices cannot be
destroyed, $v$ persists in any graph derived from $G_0$. Inspection of the
rules indicates that edges cannot be directed to a vertex that has no in-
coming edges so none can be assigned to $v$. Conversely, let $G$ satisfy the
requirements. Identify $v$ with some subject $x_1$ with no incoming edges and
let $G$ have vertices $x_1, x_2, \ldots, x_n$. Follow these steps:

1. Perform "$v$ creates $(\alpha \cup \{g\}$ to) new $x_i$" for all $x_i$ ($2 \leq i \leq n$)
   where $\alpha$ is the union of all edge labels incoming to $x_i$ in $G$;
2. For all $x_i, x_j$ such that $x_i \xrightarrow{\alpha} G x_j$ perform "$v$ grants
   $(\alpha$ to $x_j$) to $x_i$";
3. If $\beta$ is the (possibly empty) set of edges from $x_1$ to $x_i$ in $G$, then execute "$v$ removes
   ($(\alpha \cup \{g\}) - \beta$) to $x_i$" for $2 \leq i \leq n$.

This sequence of operations applied to a single subject vertex yields $G$. □

In the next corollary, "component" means connected component.

**Corollary 3.2:** A $k$ component, $n$ edge protection graph can be
constructed from a single subject in $t$ rule applications,
where $2(k-1) + n < t \leq 2(k-1) + 3n$. 
Proof: We consider the lower limit first. Every component must have at least one vertex forcing at least one Create operation to be charged to each component but one, since we are given one component. Furthermore, since connectivity is maintained with Create, Take and Grant at least one Remove operation must be charged to each component but one to perform the disconnection. Together, these give 2(k-1). Since the edge that was created as an artifact of creating one of vertices of the component was then removed when separating the components, there has not been an accounting for any of the edges of G. These can be added to the graph at the rate of one per rule application, giving the lower bound.

To see the upper limit note that rules (3.1) and (3.3) are sufficient to form one vertex in each component. For each edge charge one application of (3.1) to create its target vertex, one application of (3.2) to assign the edge to the source, and, possibly, one application of (3.3) to delete the edge from v.

Clearly, the bounds are both achievable as the example in Figure 2 illustrates.

Figure 2: Illustration of the bounds of Corollary 3.2
4. Predicates and earlier results

Several properties of paths will be extremely important in our later development. A sequence of vertices \( x_0, \ldots, x_n \) is a path in \( G \) if \( x_i \rightarrow x_{i+1} \) or \( x_{i+1} \rightarrow x_i \), \( 0 \leq i < n \). Thus paths are defined independent of directionality. Vertices \( p \) and \( q \) of \( G \) are \( tg \)-connected if there is a path \( p = x_0, \ldots, x_n = q \) and the label \( a \) on the edge between \( x_i \) and \( x_{i+1} \) contains \( t \) or \( g \). An island of \( G \) is a maximal, \( tg \)-connected subject-only subgraph of \( G \).

The edge alphabet is composed of four letters \( \{t, g, t, g\} \). If \( x \rightarrow y \) (resp. \( x \leftarrow y \)) then the letter \( t \) (resp. \( g \)) is associated with the edge. Words are associated with paths in the obvious way; for example, \( t \rightarrow g \) has the words \( t^*g \) and \( t^*g^* \) associated with it. A path \( x_0, \ldots, x_n \) is an initial span if it has an associated word in \( \{t^*g\} \), it is a terminal span if \( n > 0 \) and it has an associated word in \( \{t^*\} \), and it is a bridge if (a) \( n > 1 \) and \( x_0 \) and \( x_n \) are subjects, (b) an associated word is in \( \{t^*, t^*g, t^*g^*, t^*g^*\} \), and (c) the \( x_i \) are objects \( (0 < i < n) \). Note that the initial and terminal spans have an orientation, i.e., \( x_0 \) is the source of the spans. We say \( x_0 \) initially or terminally spans to \( x_n \).

In order to share information in the protection system, an edge pointing from the recipient to the information being shared must be added to the protection graph by means of a sequence of rule transformations of the graph. Accordingly, we may define for a set of rights \( a \) and distinct vertices \( p \) and \( q \) of a protection graph \( G_0 \), the predicate

\[
\text{can\-share}(a, p, q, G_0) \iff \text{there are protection graphs } G_1, \ldots, G_n \text{ such that } G_0 \rightarrow_* G_n \text{ and } p \rightarrow_a G_n \rightarrow q.
\]

When interest is restricted to protection graphs containing only subjects, we have
Theorem 4.1 [2]: For a subject only protection graph $G_0$, 

$$\text{can\cdot share}(a,p,q,G_0)$$

is true if and only if the following two conditions hold.

\textit{Condition 1:} There exist vertices $s_1, \ldots, s_u$ such that for each $i$, $1 \leq i \leq u$, $s_i \rightarrow q$ and $a = \bigcup_{i=1}^{u} Y_i$.

\textit{Condition 2:} $p$ is tg-connected to each $s_i$, $1 \leq i \leq u$.

The conditions under which \textit{can\cdot share} holds for general protection graphs are somewhat more complicated. In particular, Condition 1 must be augmented by Condition 3.

\textit{Condition 3:} There exist subject vertices $p'$ and $s'_1, \ldots, s'_u$ such that

(a) $p = p'$ or $p'$ initially spans to $p'$
(b) $s'_i = s_i$ or $s'_i$ terminally spans to $s_i$;

and Condition 2 must be recast in terms of bridges and islands:

\textit{Condition 4:} For each $(p', s'_i)$ pair $(1 \leq i \leq u)$ there exist islands $I_1, \ldots, I_v$ such that $p' \in I_1$, $s'_i \in I_v$ and there is a bridge from $I_j$ to $I_{j+1}$ $(1 \leq j \leq v)$.

Clearly, Condition 2 is simply Condition 4 for the case $v = 1$. The counter part to Theorem 4.1 for general protection graphs is

Theorem 4.2 [3]: The predicate $\text{can\cdot share}(a,p,q,G_0)$ is true if and only if Conditions 1, 3, and 4 hold.

As corollaries, it is known that there are algorithms operating in linear time in the size $(V+E)$ of the graph to test both predicates.

5. Theft

The $\text{can\cdot share}$ predicate presumes perfect cooperation from all users (i.e., subjects). The $\text{can\cdot steal}$ predicate must capture the notion that a subject vertex acquires a new right without any cooperation from an
original owner. Formally, for two distinct vertices $p$ and $q$ in a protection graph $G_0$, and right $\alpha$, define

\[
\text{can\_steal}(\alpha, p, q, G_0) \iff \exists p \xrightarrow[\alpha]{G_0} q \text{ and there exist protection graphs } G_1, \ldots, G_n \text{ such that}
\]

\[
(5.1) \quad G_0 \xrightarrow{\rho_1} G_1 \xrightarrow{\rho_2} \cdots \xrightarrow{\rho_n} G_n;
\]

\[
(5.2) \quad p \xrightarrow[\alpha]{G_n} q, \text{ and}
\]

\[
(5.3) \text{ if } s \xrightarrow[\alpha]{G_0} q \text{ then no } \rho_j \text{ has the form}
\]

"s grants (a to q) to x_i" for any $x_i \in G_{j-1}, 1 \leq j \leq n$.

Clearly, $p$, $q$ and $s$ must be distinct since these are protection graphs.

Although a motivational discussion of can\_steal appears elsewhere [4], a few additional remarks are in order. Notice that can\_steal($\alpha, p, q, G$) does not hold when $p$ already "owns" $\alpha$ rights to $q$. This is more realistic although it is somewhat inconvenient, technically. Also an "owner" $s$ of a right $\alpha$ cannot grant the right away since to do so and to claim later that a theft occurred would strain credibility. But $s$ can participate in a theft in other ways. In particular for the protection graph of Figure 3, $s$ would be involved in any theft of $\alpha$ rights to $q$, because it must grant (t to r) to $p$. It is reasonable for the definition to allow this since $s$ could be duped into participating, but alternate definitions, e.g. where right owners are required to be completely inactive, is also reasonable.

Thus, the present definition analyzes thefts where the only limit to full cooperation of all users is that "owners" cannot grant their rights away.
Figure 3: Subject s participates in a theft.

**Theorem 5.1:** For distinct vertices p and q in a protection graph $G_0$ and right $a$, can-steal$(a, p, q, G_0)$ if and only if the conjunction of the following conditions holds:

1. $p \overset{a}{\rightarrow} G_0 q$, (5.4)
2. There is a subject $p'$ such that $p = p'$ or $p'$ initially spans to $p$, (5.5)
3. There is a vertex $s$ such that $s \overset{a}{\rightarrow} G_0 q$ and $can\cdot share(t, p, s, G_0)$. (5.6)

**Proof:** ($\Rightarrow$) Suppose can-steal$(a, p, q, G_0)$ is true. Condition (5.4) of the theorem holds by definition. Condition (5.5) is satisfied since $p \overset{a}{\rightarrow} G n q$ implies by definition that can\cdot share$(a, p, q, G_0)$ holds which by Condition 3 of Theorem 4.2 implies $p'$ exists. Also, Condition 1 of Theorem 4.2 guarantees existence of the vertex $s$ required in condition (5.6).

To see that the remainder of that condition is satisfied, let $G_0 \overset{\rho_1}{\rightarrow} G_1 \overset{\rho_2}{\rightarrow} \cdots \overset{\rho_n}{\rightarrow} G_n$ be a minimal length derivation sequence and let $i$ be the least index such that $G_{i-1} \overset{\rho_1}{\rightarrow} G_1 \overset{a}{\rightarrow} G_i$, $x \overset{a}{\rightarrow} G_i$ but $x \overset{a}{\rightarrow} G_{i-1}$.

That is, $G_i$ is the first graph where an $a$ labeled edge to $q$ is added.

Clearly, $\rho_i$ is not a Create or Remove operation. It cannot be a Grant by condition (5.3) since by our choice of $i$ all vertices with $a$ rights to $q$ in $G_i$ are also in $G_0$. 

Thus, $p_i$ must be a Take of the form

$$
\begin{array}{c}
x \\
\downarrow \quad \alpha \\
s \\
\downarrow \quad \alpha \\
q \\
\end{array} \rightarrow 
\begin{array}{c}
x \\
\downarrow \quad \alpha \\
s \\
\downarrow \quad \alpha \\
q \\
\end{array}
$$

Now, by Condition 3, there is a subject $s'$ such that $s'=s$ or $s'$ terminally spans to $s$ and by Condition 4 there exist islands $I_1, \ldots, I_v$ such that $p' \in I_1$ and $s' \in I_v$. Now, if $s \neq s'$, i.e. $s$ is an object, then either it suffices to let $x=s'$ since $s'$ is in the same island and terminally spans to $x$ or else $s'$ is in a different island and the derivation is not of minimal length.

Thus, the conditions of Theorem 4.2 are satisfied leading to the truth of $can\cdot share(t,p,s,G_0)$. If $s=s'$, i.e. $s$ is a subject, then $x$ is in island $I_v$, and the conditions for Theorem 4.2 are satisfied provided $x \in G_0$. If $x \notin G_0$ then because $s \in G_0$ and new labels on incoming edges cannot be added to extant vertices, there must be some subject $y$ in one of the islands such that $can\cdot share(t,y,s,G_0)$ is true. Thus, we have $can\cdot share(t,p,s,G_0)$ established and the theorem proved if we can show that $s$ does not have to grant away $\alpha$ to accomplish the sharing. But it is immediate that every instance of

(5.7) \hspace{1cm} s \text{ grants } (\alpha \text{ to } q) \text{ to } y

can be replaced by

(5.8) \hspace{1cm} x \text{ takes } (\alpha \text{ to } q) \text{ from } s \\
\hspace{1cm} x \text{ takes } (q \text{ to } y) \text{ from } s \\
\hspace{1cm} x \text{ grants } (\alpha \text{ to } q) \text{ to } y

with the same effect as long as $x$ and $y$ are distinct. But if they are not distinct, line (5.8) alone can replace (5.7) with the same effect. \hfill \Box
Suppose the three conditions hold. Then if $p$ is a subject, the theorem is immediately satisfied since $p$ can take $(a \rightarrow q)$ from $s$ once it gets the $t$ right to $s$. If $p$ is an object then $\text{can-share}(t,p,s,G_0)$ implies there is some subject $p'$ initially spanning to $p$ and $\text{can-share}(t,p',s,G_0,Go)$. If $p' \xrightarrow{a}{G_0} q$ then $p'$ can take the right $(a \rightarrow q)$ from $s$ and grant it to $p$. If $p' \xrightarrow{a}{G_0} q$ then the following sequence enables $p'$ to form a surrogate vertex $n$ to transmit the right $(a \rightarrow q)$ to $p$ given that $p' \xrightarrow{t}{G_0} s$ and $p' \xrightarrow{q}{G_i} p$:

- $p'$ creates $(g \rightarrow)$ a new subject $n$;
- $p'$ grants $(t \rightarrow s)$ to $n$;
- $p'$ grants $(g \rightarrow p)$ to $n$.

(These steps are legal even if $a=t$.)

Then $n$ completes the task with operations:

- $n$ takes $(a \rightarrow q)$ from $s$;
- $n$ grants $(a \rightarrow q)$ to $p$.

This is a witness for $\text{can-steal}(a,p,q,G_0)$ proving the theorem.

6. Conspiracy

In this section we are concerned with the amount of "cooperation" required to effect the sharing or stealing. This cooperation has been called "conspiracy" [2] and for a given sequence of legal rule applications $\rho_1, \ldots, \rho_n$ it is simply $|\{x | x \text{ initiates } \rho_i \}|$. Our concern in this section is determining for a given true predicate $\text{can-share}(a,p,q,G_0)$ the minimum conspiracy required to produce $G_n$ that is a witness to its truth. We will be able to find the exact value for arbitrary protection graphs.
(Conspiracy has been studied [6] and a lower bound has been established. The bound is based on edge incidence and is not tight. For example, the class of graphs of the form shown in Figure 4 require \( n+2 \) conspirators for \( p \) to acquire the \( a \) edge to \( q \), but the previous lower bound for these graphs is \( 0 \). The present formulation uses the more flexible notion of "spans" to assess protection graphs.)

Let \( G \) be a protection graph and \( y \) a subject vertex, then the *access-set with focus* \( y \)

\[ A(y) = \{ y \} \cup \{ x | y \text{ initially spans or terminally spans to } x \}. \]

Clearly, for a given focus \( y \) in \( G \), \( A(y) \) is unique. Access sets will be used to measure the size of the conspiracy.

For the remainder of the section, we restrict our attention to a protection graph \( G \) with distinct vertices \( p = x_0, \ldots, x_n = s, x_{n+1} = q \). An edge in \( G \) either forms a direct \( tg \)-connection between \( x_i \) and \( x_{i+1} \) or \( x \rightarrow a \rightarrow q \). We suppose that \( \text{can}\_\text{share}(a;p,q,G) \) holds.

Say that a vertex is a *tg-sink* if

1. the vertex is \( x_0 \) and the only letter associated with the \( x_0, x_1 \) edge is \( t \),
2. the vertex has exactly two incident edges, both are incoming and either both are labeled with \( t \) or both are labeled with \( g \).

or

1. the vertex is \( x_n \) and the only letter associated with the \( x_{n-1}, x_n \) edge is \( g \).

The motivation for this definition will become evident in the claim of Theorem 6.1.

An *access set cover* for \( G \) with focus \( y_1, \ldots, y_u \) is a family of sets \( \Lambda(y_1), \ldots, \Lambda(y_u) \) such that for each \( i \) \( (1 \leq i \leq n) \) vertices \( \{ x_{i-1}, x_i \} \subseteq \Lambda(y_j) \) for some \( j \), \( 1 \leq j \leq u \). Note that the subject requirement of access-sets...
might prevent certain tg-connected paths from having a cover. It will become clear from the subsequent theorems, however, that a tg-path has an access-set cover if and only if can·share(α,p,q,Go) is true. Finally, an access set cover is said to be minimal if it minimizes u over all access set covers.

First, we establish a lower bound.

**Theorem 6.1**: Let Go be G - {q}, i.e. G without vertex q. Let k be the number of access sets in a minimal cover of Go, and l be the number of tg-sinks. Then k + l initiators are necessary.

**Proof**: Let ρ₁,...,ρ_v be a minimal set of rules required for a minimal set of initiators y₁,...,y_u to implement can·share(α,p,q,G). Let the access sets A(y₁),...,A(y_u) with initiator foci y₁,...,y_u be defined over Go. To see that they cover Go, note that x /∈ A(y_i) for all i implies that no initiator can take from or grant to x, so x and its incident edges can be removed without affecting rules ρ₁,...,ρ_v. But this violates the connectedness Condition 4 of can·share. Thus, the access sets A(y₁),...,A(y_u) at least covers Go.

**Claim**: Every vertex x_i that is a tg-sink must be an initiator.

**Proof of Claim**: First note that each such x_i must be a subject by Condition 4. Suppose x_i fails to satisfy the claim and it is associated with x_i's incident edges. Then no rule ρ_j of the form "z takes (t to y) from x_i" is ever executed since x_i has no out edges and it cannot be assigned any. Furthermore, since v, the number of rules, is minimal, no rules of the form "z takes (t to x_i) from x_i-1" or "x_i-1 grants (t to x_i) to z" are ever executed since no use could be made of the t right thus assigned; a similar situation holds for x_i transmitting its t right to x_i. Thus x_i and its incident edges can be deleted violating the connectedness Condition 4.
As with the \( t t \) case there is no need for any \( \rho_j \) to transmit the right, so \( x_i \) can be eliminated and thus the connectedness condition is violated. The situation for the end points is analogous. The claim follows.

Let \( y_1, \ldots, y_k \) be the \( t g \)-sink initiators. Then \( A(y_1), \ldots, A(y_k) \) are singleton sets. Moreover, each of these vertices is a member of its adjacent access-sets. Thus, the other access-sets, \( A(y_{k+1}), \ldots, A(y_{k+k}) \) \((k+k = u)\) constitute a cover for \( G_0 \). The theorem follows.

Some discussion is in order. Basically, edges can be transmitted by an initiator to any vertex in its access set. Edges are passed "along the path" because access sets will overlap. If one initiator can take from the common element and the other can grant to it, then edges can move from one access set to the next. But if the common vertex is a \( t g \)-sink, then it must aid in the communication.

Next we establish a matching upper bound, but first a lemma will simplify matters.

**Lemma 6.2:** Let \( G_0 = G - \{q\} \) and let \( A(y_1), \ldots, A(y_k) \) be a minimal access-set cover sequence for \( G_0 \) ordered by increasing indices of \( x_i \). If \( y_{i+1} \xrightarrow{a} G \rightarrow q \) then there exists \( G' \) such that \( y_i \xrightarrow{a} G' \rightarrow q \) and all rules in
$G^* - G'$ are initiated by $y_i, y_{i+1}$, and perhaps, their common element, $A(y_i) \cap A(y_{i+1})$.

Proof: Let $z = A(y_i) \cap A(y_{i+1})$. Consider the spans to $z$ from $y_i$ and $y_{i+1}$. The notation "take $r$" means "perform enough takes to acquire" right $r$. Table I presents constructive means for passing the a right in each of the four cases.

Except for Ia and IVe the vertices initiating the rules are either $y_i$ or $y_{i+1}$. □
Table I:
Rules for passing α between access sets.

<table>
<thead>
<tr>
<th>span from</th>
<th>span from</th>
<th>rule sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_i ) to ( z )</td>
<td>( y_{i+1} ) to ( z )</td>
<td></td>
</tr>
</tbody>
</table>

I. \( \text{terminal}(t^*) \) \( \text{terminal}(t^*) \) \( z \) is necessarily a subject, since \( t^* t^* \) isn't a bridge.
   (a) \( z \) creates \((tg to)\) new \( n \),
   (b) \( y_{i+1} \) takes* \((g to n)\) from \( z \) via elements of the span,
   (c) \( y_{i+1} \) grants \((α to q)\) to \( n \),
   (d) \( y_i \) takes* \((α to q)\) from \( n \).

II. \( \text{terminal}(t^*) \) \( \text{initial}(gt^*) \) (a) \( y_{i+1} \) takes* \((g to z)\) from elements of the span,
     (b) \( y_{i+1} \) grants \((α to q)\) to \( z \),
     (c) \( y_i \) takes \((α to q)\) from \( z \).

III. \( \text{initial}(t^*g) \) \( \text{terminal}(t^*) \) (a) \( y_i \) creates \((tg to)\) new \( n \),
     (b) \( y_i \) takes* \((g to z)\) from elements of the span,
     (c) \( y_i \) grants \((g to n)\) to \( z \),
     (d) \( y_{i+1} \) takes* \((g to n)\) from \( z \) via elements of the span,
     (e) \( y_{i+1} \) grants \((α to q)\) to \( n \),
     (f) \( y_i \) takes \((α to q)\) from \( n \).

IV. \( \text{initial}(t^*g) \) \( \text{initial}(gt^*) \) \( z \) is necessarily a subject since \( t^*ggt^* \) isn't a bridge.
    (a) \( y_i \) creates \((tg to)\) new \( n \),
    (b) \( y_i \) takes* \((g to z)\) from elements of span,
    (c) \( y_i \) grants \((g to n)\) to \( z \),
    (d) \( y_{i+1} \) grants \((α to q)\) to \( z \) via elements of span,
    (e) \( z \) grants \((α to q)\) to \( n \),
    (f) \( y_i \) takes \((α to q)\) from \( n \).
Corollary 6.3: For adjacent access sets $A(y_i)$ and $A(y_{i+1})$, a rights to $q$ can be transferred from $y_{i+1}$ to $y_i$ with no other initiators unless there are consecutive edges with their only associated word in $\{tt, gg\}$. In this case, one additional operation initiated by $z = A(y_i) \cap A(y_{i+1})$ is sufficient.

Let $can\cdot share(a, p, q, G)$ hold via the $tg$-connected path $G_0$, ($p = x_0, \ldots, x_n = s$) and let $A(y_1), \ldots, A(y_k)$ be a minimal access-set cover for $G_0$. Let $k$ be the number of $tg$-sinks.

Theorem 6.4: For $p$ to acquire a rights to $q$, $k+1$ initiators suffice.

Proof: Clearly, $p \in A(y_1)$, $s \in A(y_k)$. If $s = y_k$ then $y_k \xrightarrow{a} q$. If $y_k$ terminally spans to $s$, then $y_k$ takes* ($a$ to $q$) from $s$ via elements of the span. If $y_k$ initially spans to $s$, then $s$ is necessarily a subject by the conditions of $can\cdot share$. Rules IIa-b of Table I (with $s = y_{i+1}$ and $y_k = z$) suffice to transfer ($a$ to $q$) to $y_k$. In all three cases $y_k \xrightarrow{a} q$, and we have a basis step. Lemma 6.2 can now be inductively applied, and $y_1 \xrightarrow{a} q$. If $y_1 = p$ we are done. If $y_1$ initially spans to $p$ then $y_1$ takes* ($g$ to $p$) from elements of the span and it grants ($a$ to $q$) to $p$. If $y_1$ terminally spans to $p$ then $p$ is necessarily a subject by conditions on $can\cdot share$ and rules (Ia-c) with $p = z$, $i=0$ suffice to transfer ($a$ to $q$) to $p$. (Note, use of Ia implies the addition of another initiator, namely $p$, but this is counted in the definition of $tg$-sink. The case is similar for use of IIa-b by above.)

7. Conspiracy in general graphs

Although the theorems of the last section give an exact measurement of the number of initiators required for sharing, they only apply to paths.
In general, extending these results to graphs cannot be done simply by looking for paths. For example, if G is the graph of Figure 5 the only path from p to s does not qualify as a legal path for can_share(a, p, q, G) to hold, even though the predicate is true. Working from the development of Section 6 we now present a finer analysis applicable to general graphs.

Recall that if v ∈ A(x), the access set with focus x, there are three possible conditions any subset of which v can satisfy: v is the focus of A(x) (i.e., v = x), x initially spans to v or x terminally spans to v.

Each of these properties is said to be a reason for v ∈ A(x).

Given a protection graph G with subject vertices \(x_1, \ldots, x_n\), we will define a new graph, the conspiracy graph, \(H\), determined by \(G\). \(H\) has vertices \(y_1, \ldots, y_n\) and each \(y_i\) has associated with it the access-set \(A(x_i)\). There is an undirected edge between \(y_i\) and \(y_j\) provided \(\delta(x_i, x_j) \neq \emptyset\) where \(\delta\) is called the deletion operation and is defined by:

\[
\delta(x, x') = \text{all elements in } A(x) \cap A(x') \text{ except those } z \text{ for which either (a) the only reason for } z \in A(x) \text{ is that } x \text{ initially spans to } z \text{ and the only reason for } z \in A(x') \text{ is that } x' \text{ initially spans to } z \text{ or (b) the only reason } z \in A(x) \text{ is } x \text{ terminally spans to } z \text{ and the only reason } z \in A(x') \text{ is } x' \text{ terminally spans to } z.
\]

The graph thus constructed is the conspiracy graph for \(G\). See the example in Figure 6.

Let \(H\) be constructed from \(G\) as just described. Define the sets...
Figure 6: A protection graph and its induced conspiracy graph.
\[ y_p = \{ y_i \mid x_i = p \text{ or } x_i \text{ initially spans to } p \}, \]

\[ y_s = \{ y_i \mid x_i = s \text{ or } x_i \text{ terminally spans to } s \}, \]

for some \( s \) such that \( s \xrightarrow{q} q \). Then we will argue that the number of vertices on a shortest path from an element \( y_u \in y_p \) to an element \( y_v \in y_s \) in \( H \) is the number of conspirators necessary and sufficient to produce a witness to \( \text{can-share}(a, p, q, G) \). Let \( |s.p.| \) denote the length of a shortest path between \( y_u \) and \( y_v \).

First we must establish that the conspiracy graph captures the notion of sharing.

**Lemma 7.1:** \( \text{can-share}(a, p, q, G) \) is true if and only if some \( y_u \in y_p \) is connected so some \( y_v \in y_s \).

**Proof:** If the vertex \( z \) mentioned in the definition of \( \delta \) is restricted to being an object element of \( A(x_i) \cap A(x_j) \) the lemma is easily proved from Theorem 4.2 by observing that the islands of \( G \) form connected components of \( y \)'s in \( H \) and the edges between these components correspond to bridges. (Deletion of object elements is obviously necessary in order to remove false bridges of the form \( t^*t^* \) and \( t^*t^*t^*t^* \).) Also, note that even with subject deletions, if \( y_u \) and \( y_v \) are connected \( \text{can-share}(a, p, q, G) \) is true. So the remaining case is when \( \text{can-share}(a, p, q, G) \) is true but removal (by \( \delta \)) of \( z \) from \( A(x_i) \cap A(x_j) \) prevents \( y_u \) and \( y_v \) from being connected. Let \( z \) be associated with \( y_z \). Note that since \( z \) is a focus it has reason to be in \( A(x_i) \cap A(z) \) and in \( A(z) \cap A(x_j) \). Thus there are edges in \( H \) between \( y_i \) and \( y_z \) and between \( y_z \) and \( y_j \). Thus, the absence of an edge between \( y_i \) and \( y_j \) cannot prevent \( y_u \) and \( y_v \) from being connected, since there is a path between \( y_i \) and \( y_j \) in any case. \( \square \)

Notice from the proof that the effect of deleting a subject via \( \delta \) is to prevent two foci, \( y_i \) and \( y_j \) from being directly connected when their only connecting spans contain a \( t^*t^* \)-sink. By deleting such vertices, we force
y_i and y_j to be connected by a path of two edges -- a means of easily
counting the tg-sinks as a conspirator.

**Theorem 7.2:** To produce a witness to \( \text{can} \cdot \text{share}(a,p,q,G) | s.p. | \)
conspirators are sufficient.

**Proof:** A simple induction on the spans corresponding to the edges
of the s.p. using Lemma 6.2 proves the result provided we observe the
following point. Since p, q, s are distinct and the y_i on the s.p. are
distinct, all rules given in Lemma 6.2 can be performed provided the foci
of the access-sets are different from their common element(s). By in-
spection of the rules of Lemma 6.2, whenever a focus and common element
coincide the rule whose application is prevented (by distinctness of
vertices for rule applications, Sec. 2) provides a right that is already
possessed (e.g., rule IIc, \( y_i = z \)) or it provides a right used in the
subsequent rule to acquire a right already possessed (e.g., rule IIa and
IIb, \( y_{i+1} = z \)). In these cases the rule whose application is prevented
is not needed. \( \square \)

**Theorem 7.3:** To produce a witness to \( \text{can} \cdot \text{share}(a,p,q,G) | s.p. | \)
conspirators are necessary.

**Proof:** Let \( y_u = z_1, ..., z_w = y_v \) be vertices along a shortest path
from \( y_u \) to \( y_v \). If there exist only tg-connected paths in G from \( z_i \) to \( z_{i+1} \)
\( (1 \leq i \leq w) \) then the \( z_i \) are foci of an access-set cover for the path. By con-
struction there are no tg-sinks and if \( y_u \) is not associated with p (resp.
\( y_v \) is not associated with s) then the subject associated with \( y_u \) (\( y_v \))
initially (terminally) spans to p (s) and so it need not conspire. By
theorem 6.1, \( w \) conspirators are necessary.

The remaining case is for an induced path in \( H \) that is not a path in
G. Although redundant rule applications may arise, it is clear the duplicated
vertices along a span are not harmful to the lemma unless they reduce the
of required conspirators. Suppose that conspirators $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_u$ can produce a witness. Then there is a $r \in \Lambda(z_{i-1}) \cap \Lambda(z_{i+1})$. But by choice of the $z_i$ vertices on a shortest path there is no edge between $z_{i-1}$ and $z_{i+1}$. Thus, $r \notin z_{i-1}, r \notin z_{i+1}$ and $r \notin \delta(z_{i-1}, z_{i+1})$. But this implies (if $r$ is an object) that there is no bridge between $z_{i-1}$ and $z_{i+1}$ (contradicting by Lemma 7.1 the assumption that the $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n$ are sufficient) or it implies (if $r$ is a subject) the presence of a $tg$-sink. By Theorem 6.1 $r$ must be counted as a conspirator.

8. Concluding Remarks

The development of the conspiracy results provides a reasonably clear picture of how sharing is accomplished in the Take-Grant Model. In particular, the notion of access-set describes that portion of a protection graph under direct "control" of the subject which is its focus. Communication outside of this region of influence requires the cooperation of other subjects. This information will doubtless be useful for designers of specific protection systems as previously explained [4].

Several problems remain open. First, there is the question of algorithmic complexity of determining the minimum number of conspirators required for a right to be shared. In Section 7 this is determined by finding a shortest path in a conspiracy graph. That question is obviously a linear time process, but the construction of a conspiracy graph (as described) requires $n^2$ operations for an $n$ subject graph just to fill in the edges. A simpler scheme that does not depend on the explicit construction of the conspiracy graph could be envisaged.

Another issue is to determine for a given graph what set of conspirators must have participated in the sharing of a right after the fact. The test is complicated by the fact that certain rights could have been removed in order
to hide the conspiracy. One might be able to infer from the structure of the graph that even though a subject has deleted the conspiratorial rights, they once existed.

Acknowledgement: Thanks are due to Mat Bishop for useful conversations concerning conspiracy to the first referee whose suggestions substantially improved the paper and to Michele Boucher for expert preparation of the original manuscript.

9. References


