1 Handles

A “handle” $h$ is a random identifier, generated by the operating system. When a process creates a handle, it specifies whether it should be included in the global sets $D_a$ and/or $D_r$. Those handles in $D_a$ are globally addable to labels. Those handles in $D_r$ are globally removable from labels.

A handle $h$ used for an integrity compartment is specified in $D_r$ by default. A handle $h$ used for a secrecy compartment is specified in $D_a$ by default. A handle used for a top-secret compartment is in neither.

2 Capabilities

A handle $h$ has associated capabilities $a(h)$, $r(h)$ and $g(h)$, which denote “addition”, “removal” and “group selection”, respectively. A process that holds $a(h)$ can freely add $h$ to labels; a process that holds $r(h)$ can freely remove $h$ from labels. A process that holds $g(h)$ also owns all capabilities in the group $G_h$.

3 Labels

A label is a set of handles. A process can add a handle $h$ to a label $L$ if either $h \in D_a$ or the process owns $a(h)$. Similarly for removal. Each process $p$ has two labels: $S_p$ and $I_p$, denoting secrecy and integrity, respectively.

4 Groups

A group is a set of capabilities. Each group is named uniquely by a single handle. Unlike labels, groups can be shared across processes. A process can add a capability $c$ to $G_h$ if it owns both $c$ and $a(h)$. That is, if it owns the capability, and the capability for adding to the group $G_h$. Similarly for removal.

A process can only make changes to a group if its secrecy label is empty — otherwise it could modulate a group as a covert channel.

Each process $p$ has a special group $O_p$ that specifies which capabilities it owns. This group is not shared with other processes. $p$ can remove from $O_p$ arbitrarily. When $p$ is granted new capabilities, they appear in $O_p$.

We say that for a capability $c$ and a group $G_h$, that $c \in G_h$ if either $c$ appears in $G_h$, or of there is a capability $g(h') \in G_h$ such that $c \in G_h'$. That is, we imagine a tree of groups, rooted at the group $G_h$, and including
a child group $G_{h'}$ if $G_h$ contains the group selection capability for $h'$. A capability $c$ is in the group $G_h$ if it appears somewhere in the tree rooted at $G_h$.

Given a group $G_h$, we can consider all of the removal and addition capabilities contained recursively in $G_h$. Thus, we have labels $R(G_h)$ and $A(G_h)$:

$$
R(G_h) = \{ h' \mid r(h') \in G_h \}
$$

$$
A(G_h) = \{ h' \mid a(h') \in G_h \}
$$

5 Changing Labels

A process $p$ has a secrecy label $S_p$, an integrity label $I_p$, and an ownership group $O_p$. Let’s say a process wants to update a label $L$, which is either its secrecy or integrity label. A process can set a label $L$ to be $L'$ if and only if:

$$
L - L' \subseteq R(O) \cup D_r \quad \text{and} \quad L' - L \subseteq A(O) \cup D_a
$$

This is stating more formally what was described above. Handles can be removed from a label if either the process owns the removal privilege for that handle, or if the handle is universally removable. Similarly for addition.

6 Interpretation of Labels

The label $S_p$ describes which secrets a process has seen. For each handle $h \in S_p$, the process is assumed to have seen all secrets associated with $h$. The larger $S_p$, the more types of secrets $p$ has seen.

The label $I_p$ describes which integrity levels $p$ belongs to. The larger $I_p$, the more integrity $p$ has.

7 Messaging

In general, a process $p$ can message a process $a$ if $S_p \subseteq S_a$ and $I_q \subseteq I_p$. That is, $p$ can message $q$ if $p$ has seen fewer secrets than $q$, and if $p$ is at a higher integrity level than $q$.

8 Per-Channel Labels

$p$ might be a server, which can talk to many different processes. In that case, $p$ might want to act as if it has many different secrecy and integrity
levels. A process $p$ can speak to $q$ with a per-channel secrecy level $S_{p→q}$ and per-channel integrity level $I_{p→q}$ so long as it is allowed to change its secrecy and integrity levels to $S_{p→q}$ and $I_{p→q}$ respectively, and it is allowed to switch back. We can apply the formulae above to see that a process can use a per-channel label $L_{p→q}$ if:

$$(L \cup L_{p→q}) - (L \cap L_{p→q}) \subseteq (A(O) \cup D_a) \cap (R(O) \cup D_r)$$

For example, say that $S_p = \{\}$, $I_p = \{\}$, $O_p = \{a(x), r(x)\}$ and $S_q = \{x\}$. Then $A$ can set a per-channel secrecy label of $S_{p→q} = \{x\}$. This allows $p$ and $q$ to communicate with the same secrecy level, and therefore, to enjoy bidirectional communication.