Homework #2

Due: May 1, 2017 **Points:** 100

Questions

1. (*10 points*) Classify each of the following as an example of a mandatory, discretionary, or originator controlled policy, or a combination thereof. Justify your answers.

   (a) The file access control mechanisms of the UNIX operating system

   (b) A system in which no memorandum can be distributed without the author's consent

   (c) A military facility in which only generals can enter a particular room

   (d) A university registrar's office, in which a faculty member can see the grades of a particular student provided that the student has given written permission for the faculty member to see them.

2. (*25 points*) Given the security levels TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED (ordered from highest to lowest), and the categories A, B, and C, specify what type of access (read, write, or both) is allowed in each of the following situations. Assume that discretionary access controls allow anyone access unless otherwise specified.

   (a) Paul, cleared for (TOP SECRET, {A, C}), wants to access a document classified (SECRET, {B, C}).

   (b) Anna, cleared for (CONFIDENTIAL, {C}), wants to access a document classified (CONFIDENTIAL, {B}).

   (c) Jesse, cleared for (SECRET, {C}), wants to access a document classified (CONFIDENTIAL, {C}).

   (d) Sammi, cleared for (TOP SECRET, {A, C}), wants to access a document classified (CONFIDENTIAL, {A}).

   (e) Robin, who has no clearances (and so works at the UNCLASSIFIED level), wants to access a document classified (CONFIDENTIAL, {B}).

3. (*25 points*) Prove that the two properties of the hierarchy function (see Section 5.2.3) allow only trees and single nodes as organizations of objects.

4. (*15 points*) In the Clark-Wilson model, must the TPs be executed serially, or can they be executed in parallel? If the former, why; if the latter, what constraints (if any) must be placed on their execution.

5. (*25 points*) A company develops a new security product using the agile programming[1] software development methodology. Programmers code, then test, then add more code, then test, and continue this iteration. Every day, they test the code base as a whole. The programmers work in pairs when writing code to ensure that at least two people review the code. The company does not adduce any additional evidence of assurance. How would you explain to the management of this company why their software is in fact not "high assurance" software?

[1] In the book, this is called "extreme programming"