

There are 42 possible points. My rough expectation is that 35 points and demonstrates A-level understanding of the material, 27 and up is a B.

Note that the solutions given represent one solution, not necessarily the only correct (or even the best) one.

1 Access Control Matrix (40 minutes, 21 points)

Sketched below is a portion of an access control matrix (as defined by Harrison, Rizzo, and Ullman) scheme supporting discretionary access control (as defined in the Bell-LaPadula model.)

The initial matrix is given below. Permissions shown are *r* (read), *w* (write), *a* (append), and *e* (execute/empty). There are also corresponding permissions *R*, *W*, *A*, and *E* that correspond to an “operation in progress”.

	<i>O</i> ₁	<i>O</i> ₂	<i>O</i> ₃	<i>O</i> ₄	...	<i>A</i>	<i>B</i>	<i>C</i>	...
<i>A</i>	<i>r</i>	<i>rw</i>	<i>ra</i>	<i>e</i>					
<i>B</i>		<i>rw</i>	<i>ra</i>	<i>aw</i>					
<i>C</i>	<i>r</i>								
...			...						

The *get_read*, *release_read*, *get_write*, and *release_write* procedures follow.

<pre> command <i>get_read</i>(<i>s</i>, <i>o</i>_{<i>i</i>}) if <i>r</i> in <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] then enter <i>R</i> into <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] </pre>	<pre> command <i>get_write</i>(<i>s</i>, <i>o</i>_{<i>i</i>}) if <i>w</i> in <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] then enter <i>W</i> into <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] </pre>
<pre> command <i>release_read</i>(<i>s</i>, <i>o</i>_{<i>i</i>}) if <i>R</i> in <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] then delete <i>R</i> from <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] </pre>	<pre> command <i>release_write</i>(<i>s</i>, <i>o</i>_{<i>i</i>}) if <i>W</i> in <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] then delete <i>W</i> from <i>a</i>[<i>s</i>, <i>o</i>_{<i>i</i>}] </pre>

1.1 Effect of an operation (2 minutes, 2 points)

First, an easy one to warm up. Show the effect of the calls:

```

get_read(B, O2)
get_write(A, O3)
    
```

(you can just write on the matrix above to show what it looks like after the two calls.)

The entry in row B column O₂ should have an R added to it (1 point), and there should be no other changes (1 point).

1.2 Decidability (4 minutes, 2 points)

Assuming equivalent definitions for other operations, do you think that safety is decidable for this model? Justify your (yes or no) answer.

Yes (1 point). The commands are mono-operational. Any system with mono-operational commands is decidable (Theorem 3.1). (1 point).

1.3 Modeling Mandatory Access Control (8 minutes, 5 points)

Your task is to update this model to support the mandatory access control facilities of Bell-LaPadula. First, describe briefly how you would model mandatory access control levels. Remember that you are modeling this with the HRU definitions/operations (e.g., you can only test for presence of a right.) For full credit, you should be able to support a lattice of security levels - but you'll only lose a point if you can only support a total ordering.

The Access control matrix only allows testing the presence of a right, not comparing (as with levels). Therefore I'll add a right corresponding to each vertex in the lattice (1 point), a negative right (nX , where X is a level) (1 point), and a corresponding `create_object_at_x` that creates an object at that security level/category (1 point). I will create a special "MAC" subject that holds the level for objects - a new object will have entries X for each level X the subject/object dominates, and nX for each level the subject/object doesn't dominate (1 point). Commands will test for the presence/absence of appropriate combinations of rights in the MAC entries for the subject and object to determine if the operation should be allowed, as well as the DAC entries as describe above (1 point).

Scoring: Basically 1 point for levels, 1 point for capturing category information, 1 for preventing write-down, 1 for noting that subject level, 1 for working in the ACM framework - possibly others as noted above.

1.4 `get_read` (8 minutes, 4 points)

Give an example `get_read` procedure for your modified scheme (it is okay if you assume the levels/categories/lattice is fixed.)

Assume three levels, $l < m < h$.

`get_read(s,o)`

if ($H \in a[MAC, s]$ or ($nH \in a[MAC, o] \wedge M \in a[MAC, s]$) or
 $nM \in a[MAC, o]$) and $r \in a[s, o]$
then enter R into $a[s, o]$

Scoring: 2 points for checking MAC in a way that matches your answer to the previous question, 1 for the DAC check, 1 for entering the proper right.

1.5 `create_object` (8 minutes, 4 points)

Define a command for creating an object. Note that the object must have a security level when it is created - it is up to you to decide how this is done. For example, you could have security level as a parameter, or you could have a different command to create objects at different security levels. You'll get partial credit if you just define one for the discretionary mechanism (possibly even full credit if you left question 1.3 blank.)

Assume three levels, $l < m < h$. Note that I'm also creating an "owner" right to go with the DAC.

`create_object_m(s,o)`

if $nH \in a[MAC, s]$ then
 `create_object O`
 enter o into $a[s, o]$

enter L into $a[s, o]$
enter M into $a[s, o]$
enter nH into $a[s, o]$

Scoring: 1 for creating the object, 2 for proper MAC entries, 1 for catching that you shouldn't be able to create lower-level objects (would open a covert channel).

1.6 Safety / decidability (again) (6 minutes, 4 points)

What would be a meaningful safety analysis in this system? Remember that a system is safe with respect to a right s if s can't be leaked. What rights would you want to check, and how would you define the initial matrix, so that the safety question would tell you something interesting about mandatory access control? Does your previous decidability answer still hold?

Fill the matrix with right $R / W / A / E$ everywhere that it is allowed (1 point). The safety check is then to see if one of these rights is leaked (1 point). I'm not sure if this is decidable (1 point), it is neither mono-operational or mono-conditional, so we don't have any theorems that would say it is decidable (1 point).

2 Cryptography and Integrity (10 minutes, 6 points)

Consider the following three approaches to managing integrity. For each, give an example of a type of attack where the method does provide integrity, and a type of attack where the integrity is suspect.

1. A one-way hash of the data item is computed and stored with the item. A reader can compare the hash of the item read with the stored hash value to verify integrity.

Protects against any bits being changed (1 point). However, an adversary could replace the complete item AND replace the hash to violate integrity (1 point).

2. The item is encrypted with the private key of the writer. The reader uses the writer's public key to decrypt the item.

As long as the public key has integrity (is known to belong to the original writer), this provides both value and origin integrity (1 point). An attacker could alter a single bit and cause loss of the object, but this is really availability (1 point). An integrity attack would require convincing the reader that a public key generated by the attacker was that of the correct author (1 point).

3. The item is encoded using an error-correcting code (e.g., Hamming code); when read the error correction can be used to fix any loss of integrity.

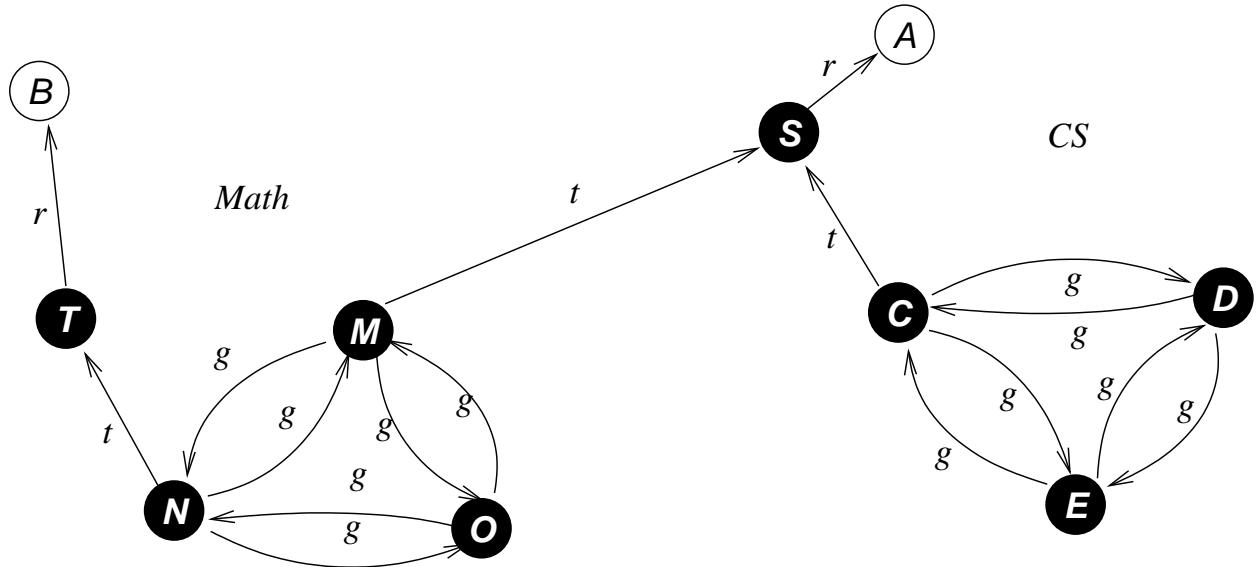
This allows correcting a few damaged bits, providing both an integrity check for errors of a limited number of bits (1 point) as well as improving availability. However, an attacker can violate integrity by changing more bits than the error correction is capable of correcting, even if the entire object isn't changed (1 point).

Scoring: Reasoning that shows a decent knowledge of the material, even if leading to an incorrect conclusion, may be worth a point.

3 Analyzing Allowed Access (20 minutes, 15 points)

To aid in advising students, I have (hypothetically) developed a system designed to give faculty access to the work of students in their class. Faculty are also allowed to consult with other faculty *in their department* on the work done for their class.

To evaluate the effectiveness of my system in meeting the access control requirements, I have modeled it using the take-grant model. A sample is shown below. There are two departments shown, *Math* and *CS*. In particular, there is a student *S* taking courses both from *M* (a professor in Math) and *C* (a professor in CS). The student has done assignment *A* for the CS course taught by *C*. There is also a student *T* taking only a math course from Professor *N*.



3.1 Model analysis (10 minutes, 8 points)

Does the system meet the informally stated goals? In particular, answer the following (give a brief justification of your answer, e.g., reference a theorem and describe why it applies or a show how the operation described would be performed using the take/grant/create rules of the model.)

1. Can *C* see the assignment *A* of *S*?
Yes (1 point). *Use the take rule to get r on A from S* (1 point.)
2. Can *C* allow other *CS* faculty to see the assignment *A*?
Yes (1 point). *After taking, use the grant rule to grant r on A to E* (1 point.)
3. Are faculty in other departments prevented from getting access to *A*?
No (1 point). *M can use the take rule to get r on A from S* (1 point.)
4. Is it guaranteed that *CS* faculty can never see the assignment *B* of student *T*?
No (1 point). *S creates an object O , and has g and t on O . M takes g on O from S . M grants r on B to O . S takes r on B from O . C takes r on B from S .* (1 point.)

3.2 Extended Schematic Protection Model (10 minutes, 7 points)

Because of the problems with the first approach, I've decided that I need more checks to control access. I'm going to model this using the Extended Schematic Protection model as follows:

Types: $TS = \{ \text{Professor, Student} \}$, $TO = \{ \text{Assignment} \}$

Rights: $RC = \{ \text{in course, grant} \}$, $RI = \{ \text{read, write} \}$

Link predicates: $\text{link}(X,Y) = Y/g \in \text{dom}(X) \vee X/i \in \text{dom}(Y)$

Filter Rule: $f(P,P) = \{ A/r \}$

Create Rule: $cc(P,S) = A$

$cr_P = A/r:c$

$cr_S = A/rw$

Each Professor will hold a grant ticket for every other Professor in their department (and only those in their department.) The in course ticket will be held by every Professor for students in (and only students in) their courses.

3.2.1 Does this help? (4 minutes, 4 points)

For which of the items in question 3.1 does this ESPM model give the correct result (a “yes” answer)? Just list the numbers, you don't need to justify your answers.

1,2,3,4 (1 point each).

3.2.2 Justify your answer (4 minutes, 3 points)

Explain your answer to 3.2.1 for **one** of the items that the ESPM model handles correctly (e.g., that you listed in your answer to 3.2.1 - if none, choose any one.) For full credit, if you came up with different answers for 3.1 and 3.2.1, then choose one that the ESPM model handled correctly and the take-grant model didn't (i.e., something you listed in question 3.2.1 for which you said no in question 3.1.)

3: Push all rights out to determine the maximal rights (1 point). Since the only transfer allowed is between linked faculty (by the filter function), the only way faculty can get access is if they are linked to other faculty (1 point). Since there is no link between faculty who are not in the same department, there can be no transfer of rights between faculty who are not in the same department (1 point). Such rights can only come through a (joint) create, and then only on the created object (1 point).