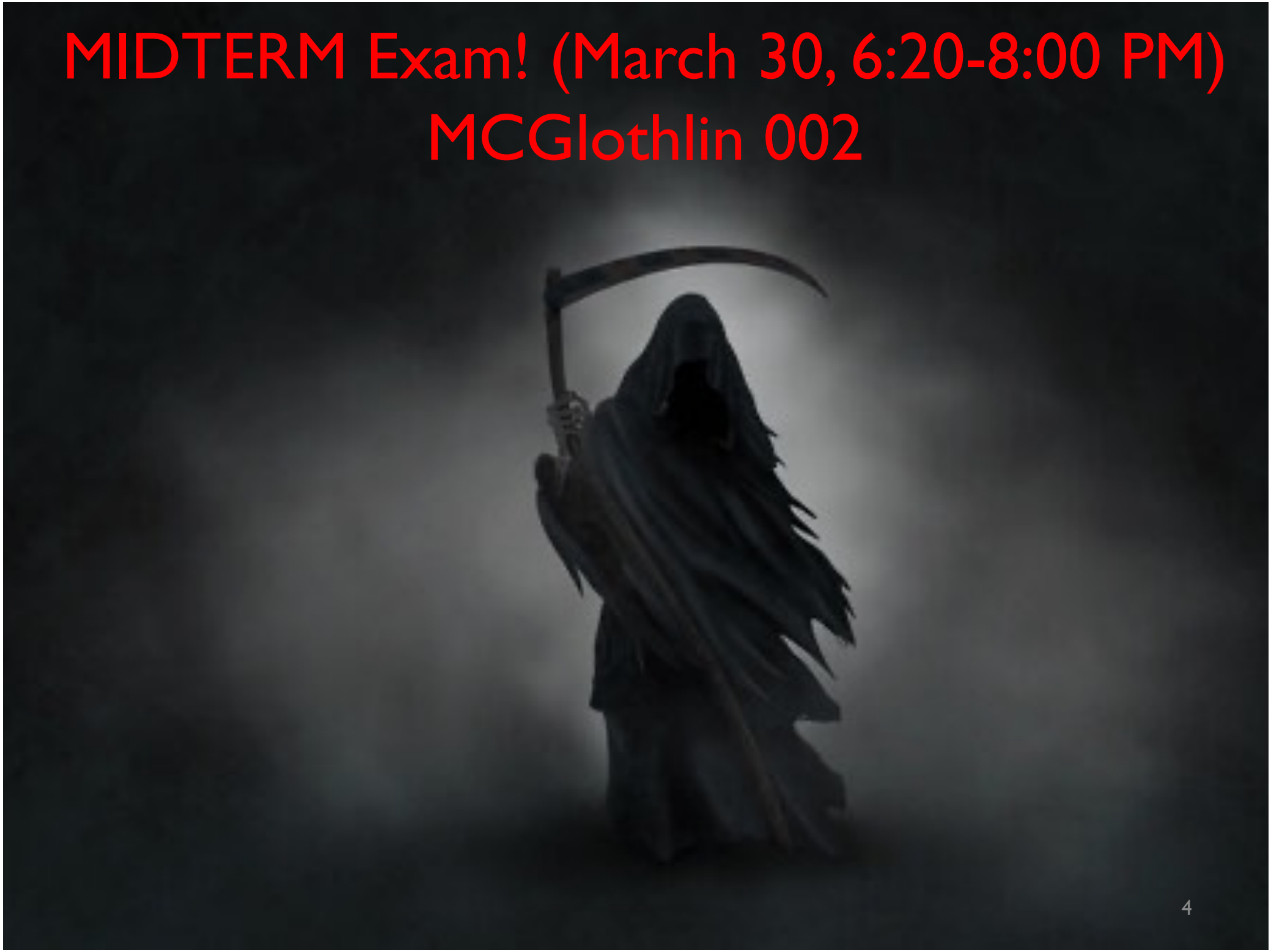
CSCI 667: Concepts of Computer Security

Lecture 12

Prof. Adwait Nadkarni

Derived from slides by William Enck, Patrick McDaniel and Trent Jaeger

MIDTERM Exam! (March 30, 6:20-8:00 PM)
MCGlothlin 002



Midterm

- **Crib sheet:** *1 page, both sides, handwritten*
- **Graphing calculator**

Syllabus for the midterm

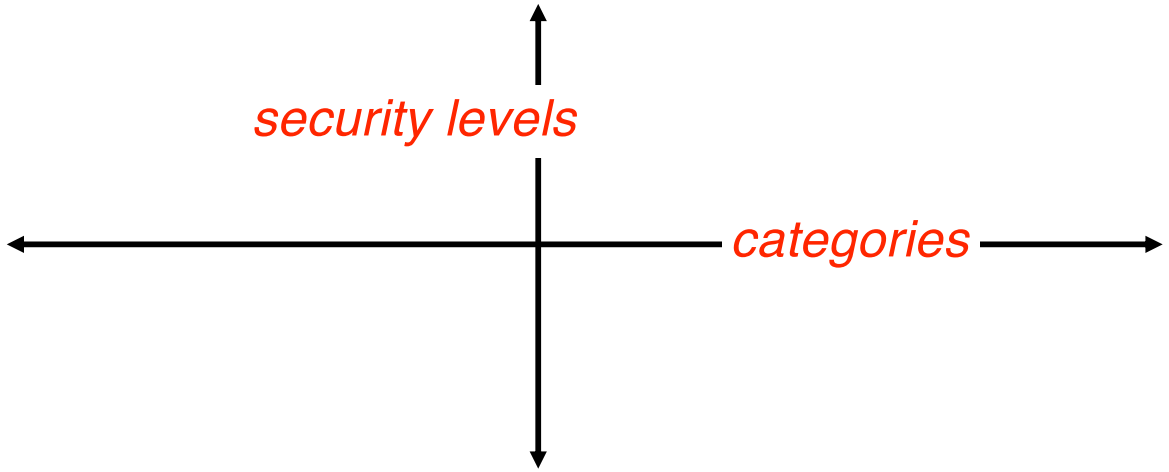
- Everything before Spring break.
- Exam will test three kinds of things:
  - *knowledge* (do you know terminology/approaches)
  - *synthesis* (can you extrapolate or compare concepts)
  - *application* (can you apply what you learned)

Sample Questions

- Short answer question: *Why are active attacks easier to detect than passive attacks?*
- Long answer question: *Explain what the safety problem is and how mandatory access control systems deal with it?*
- Problem question: *Acme archival storage systems is a company that promises to securely store customer data. They provide a online system that the customer submits documents for storage which Acme encrypts using AES and a key specific to each request. Acme only accepts requests from 8am to 5pm, Monday through Friday, and they are open on all holidays not falling on a weekend. For the purposes of this exercise, you can assume that Acme has been in operation for exactly 700 days. A customer document di is encrypted as E(di , kr), where the key kr is computed the kr = h(ti) and ti is the timestamp (with millisecond granularity) of the request submission. What is the entropy of the key?*

Multilevel Security

- A multi-level security system tags all object and subject with security tags classifying them in terms of sensitivity/access level.
  - We formulate an access control policy based on these levels
  - We can also add other dimensions, called categories which horizontally partition the rights space (in a way similar to that as was done by roles)



US DoD Policy

- Used by the US military (and many others), the Lattice model uses MLS to define policy
- Levels:

UNCLASSIFIED < CONFIDENTIAL < SECRET < TOP SECRET

- Categories (actually unbounded set)

NUC(lear), INTEL(igence), CRYPTO(graphy)

- Note that these levels are used for physical documents in the governments as well.

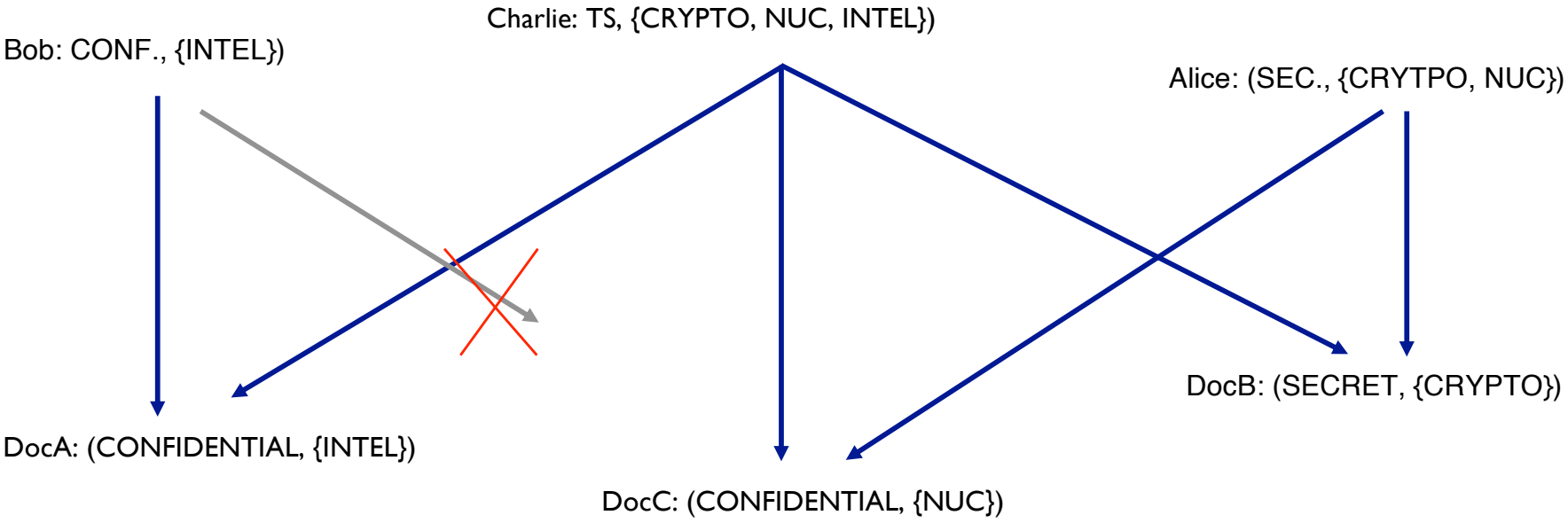
Assigning Security Levels

- All subjects are assigned clearance levels and compartments
  - Alice: (SECRET, {CRYTPO, NUC})
  - Bob: (CONFIDENTIAL, {INTEL})
  - Charlie: (TOP SECRET, {CRYPTO, NUC, INTEL})

- All objects are assigned an access class
  - DocA: (CONFIDENTIAL, {INTEL})
  - DocB: (SECRET, {CRYPTO})
  - DocC: (CONFIDENTIAL, {NUC})

Evaluating Policy

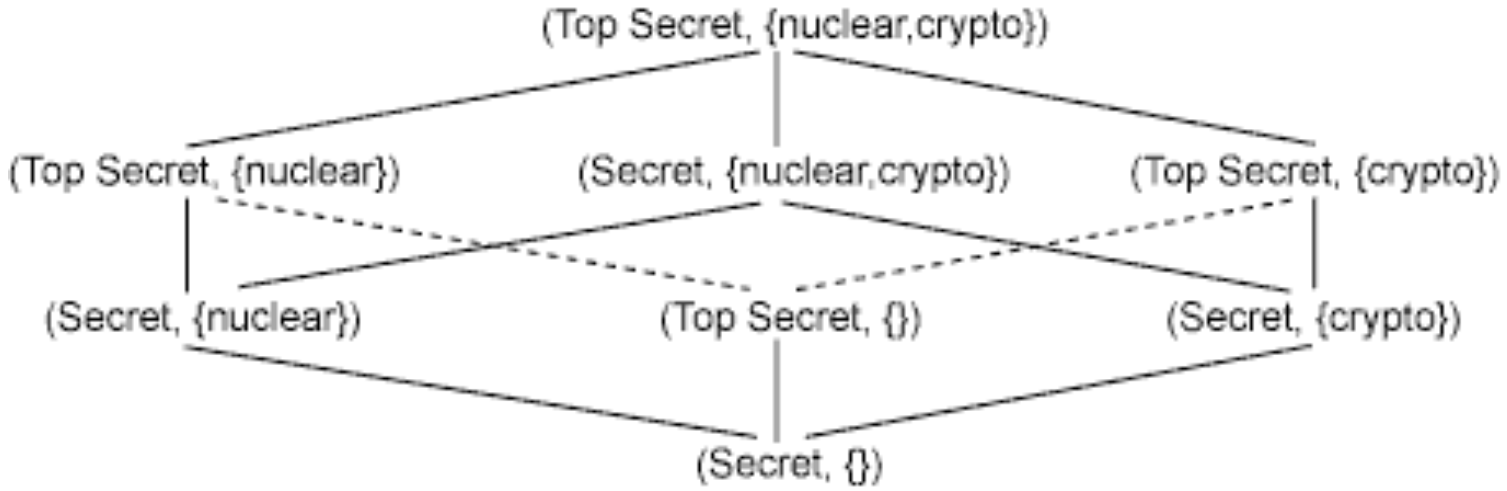
- Access is allowed if
- subject clearance level >= object sensitivity level *and* subject categories ⊇ object categories (*read down*)



Q: What would *write-up* be?

Bell-LaPadula (BLP) Model

- A Confidentiality MLS policy that enforces:
  - *Simple Security Property*: a subject at specific classification level cannot read data with a higher classification level. This is short hand for "*no read up (i.e., read down)*".
  - * *(star) Property*: also known as the confinement property, states that subject at a specific classification cannot write data to a lower classification level. This is short for "*no write down (i.e., write up)*".



How about integrity?

- MLS as presented before talks about who can "read" a document (confidentiality)
- Integrity considers who can "write" to a document
  - Thus, who can effect the integrity (content) of a document
  - Example: You may not care who can read DNS records, but you better care who writes to them!
- Biba defined a *dual* of secrecy for integrity
  - Lattice policy with, "no read down, no write up"
    - Users can only *create* content at or *below* their own integrity level (a monk may write a prayer book that can be read by commoners, but not one to be read by a high priest).
    - Users can only *view* content at or *above* their own integrity level (a monk may read a book written by the high priest, but may not read a pamphlet written by a lowly commoner).

Biba (example)

- Which users can read what documents?
- Which users can write what documents?
  - Remember "*no read down, no write up*"

Charlie: (TS, {CRYPTO, NUC, INTEL})

Bob: (CONF., {INTEL})

Alice: (SEC., {CRYTPO, NUC})

?????

DocB: (SECRET, {CRYPTO})

DocA: (CONFIDENTIAL, {INTEL})

DocC: (CONFIDENTIAL, {NUC})

Integrity, Sewage, and Wine

- Mix a gallon of sewage and one drop of wine gives you?
- Mix a gallon of wine and one drop of sewage gives you?

Integrity is really a contaminant problem: you want to make sure your data is not contaminated with data of lower integrity.

LOMAC

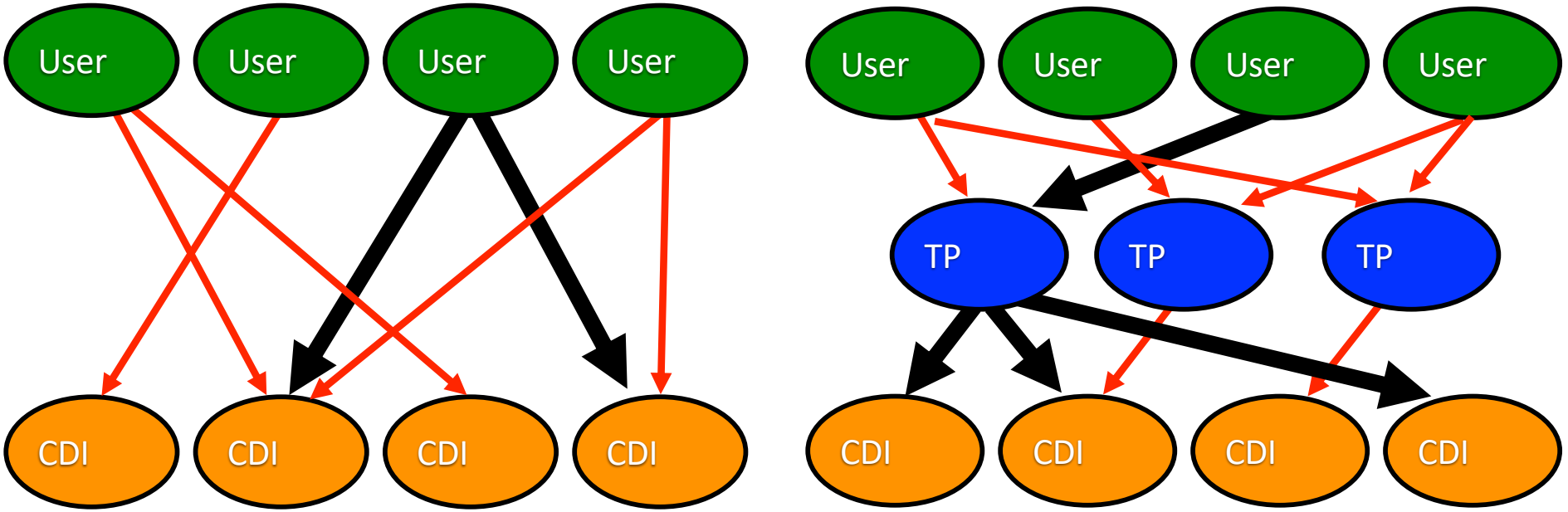
- Low-Water Mark integrity
  - Change integrity level based on actual dependencies
- Subject is initially at the highest integrity
  - But integrity level can change based on objects accessed
- Ultimately, subject has integrity of lowest object read
  - Example of "*self revocation*"

Clark-Wilson Integrity

- Map Integrity in Business (e.g., accounting) to Computing
- High Integrity Data (objects)
  - "Constrained Data Items" (CDIs)
- High Integrity Processes (programs)
  - "Transformation Procedures" (TPs)
- Check Integrity of Data Initially (verification)
  - "Integrity Verification Procedures" (IVPs)
- Premise
  - If the IVPs verify initial integrity
  - and high integrity data is only modified by TPs
  - Then, the integrity of computation is preserved

CW Permissions

- A user can access an CDI using TP iff
  1. The user has been granted CDI access
  2. The TP has been granted CDI access
  3. The user has been granted access to the TP



Clark-Wilson Issues

- Assure Function
  - Certify IVPs, TPs to be 'valid' (i.e., correct) (C1,C2)
  - Is there a general way of defining correctness?
- Handle Low Integrity Data
  - A TP must upgrade or discard any UDI (low integrity data) it receives (C5)

Reality: nice model, but too heavyweight in general for most applications. CW-lite (Jaeger) is an alternative that is tractable to implement.

The End