DSCI 519: Foundations and Policy for Information Security

Bell-LaPadula System Interpretation, U. S. Classified Information Policy

Tatyana Ryutov

Outline

- Review
- Bell-LaPadula (BLP) model
  - Preliminary concepts: inductive reasoning, MAC access classes, lattice structures
  - Formal BLP definition
  - Refinements to BLP (tranquility, communicating down)
- Bell-LaPadula system interpretation
- U. S. Classified Information Policy

Presentation 2

Multimodal and Multi-pass Authentication (MMA) Mechanisms
for Electric Vehicle Charging Networks

Eddie Garcia & Lily Guilfoil

USC Viterbi
School of Engineering

University of Southern California

L4.Q6

- Think about all material covered in class today. Identify:
  1. Points that are "crystal clear"
  2. The "muddy points"

HW 1 Discussion

- Due on September 20th
- Intentionally open-ended, no single correct solution
  - Make sure to state your relevant reasonable assumptions
- Problem 1: threats and countermeasures
- Problem 2: DAC policy as an access control matrix, interpretation of RM
- Problem 3: MAC policy (levels and categories)

In-Class Presentation Topics

- Research some **current effort** directed to build secure (high assurance) systems
  - ~~Zero Trust architecture, operating systems, hypervisors, container security, microkernels, secure mobile devices, IoT, TPM, secure HW extensions (e.g., Arm TrustZone), cyber-physical systems~~
  - Application whitelisting
  - Cloud security
  - GPS security
  - Satellite system security
  - Graphics subsystem security
  - High performance computing security
  - 5G security
    - What is the underlying idea/framework/model?
    - Compare this methodology to the Reference Monitor approach
    - What are the advantages and limitations?
    - The more technical details, the better!

Your Questions

- What coding languages should we know?
- Are there alternatives or valid arguments against the RM?
- I would like to learn more details about Tainted Flow Analysis
  - A. Sabelfeld and A. C. Myers, "Language-based information-flow security", *IEEE Journal on Selected Areas in Communications*, 2003
- When discussing the basic security theorem, what is the definition of "secure" state? You mentioned that the definition of secure varies, but I was curious what secure is typically described as.

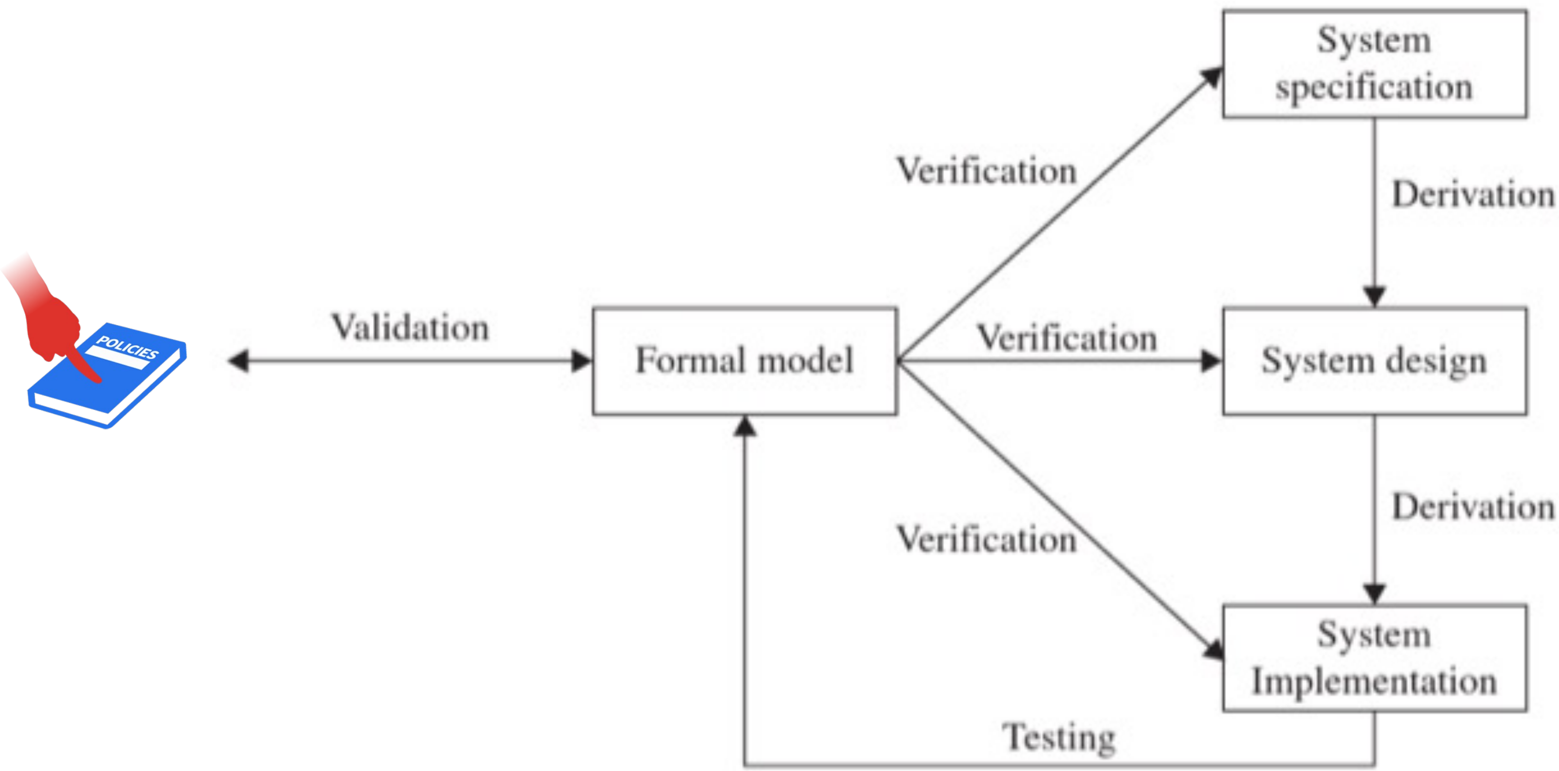
USC Viterbi School of Engineering

Your Questions

- Can you redefine the differences between the UNIX setuid formal model and BLP?
- Can we please review setuid and how it compares to BLP model.
- How would you make the Unix setuid reflective of a proper security model? Would adding which states satisfy security requirements be sufficient?
- Can we go over the FSPM Abstraction for MAC
- A little more elaboration on Lattices please!
- What is a Lattice? What is the difference between LUB and GLB?
- Can you clarify the difference between elements in the set and the nodes in the lattice chart? additionally, the greatest and lowest bounds?
- Why are the LUB and GLB important to the lattice structure.
- Can I get another example regarding lattice?
- …. more questions about lattices

FSPM

- FSPM enables leap from policy to software and HW (trusted computing base - TCB)
  - Provides an inspectable intermediate step
  - Formal description of the functions that the TCB will perform

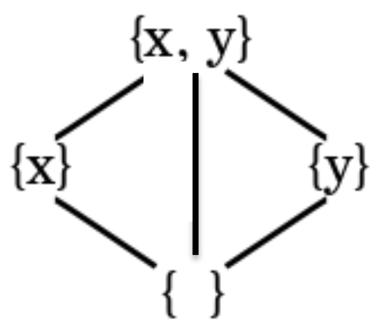


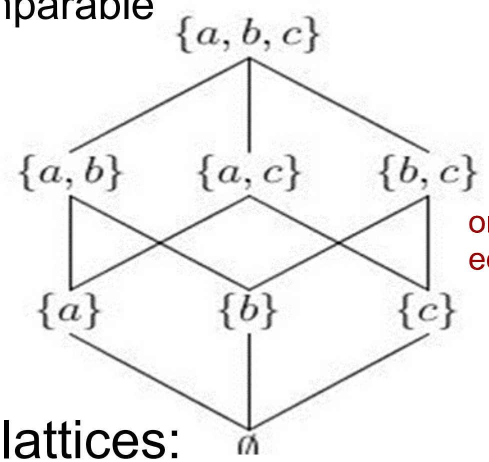
Lattice Structure

- **Total order** is linear (like a chain): a ≥ b, b ≥ c
- **Partial order** of a set occurs when a relation orders some, but not all elements of a set
  - Binary relation that is:
    1. Reflexive: a ≥ a
    2. Anti-symmetric: a ≥ b and b ≥ a then a = b
    3. Transitive: a ≥ b and b ≥ c then a ≥ c
- **Lattice**: a partially ordered finite set in which every two elements have a least upper bound (LUB) **and** a greatest lower bound (GLB)

Lattice Examples

- Lattice representation (Hasse diagram):
  - *dom* relationship is shown by undirected edges (assuming edges oriented downwards)
  - absence of a horizontal connection means incomparable

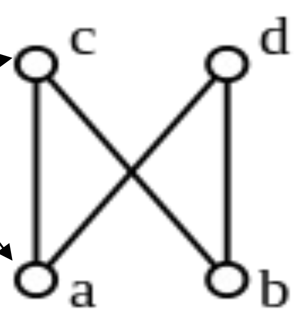


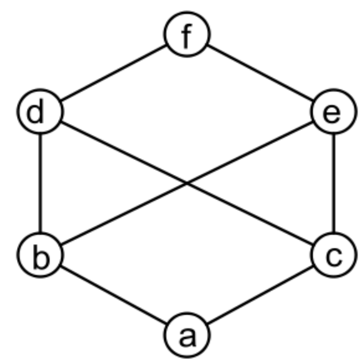


omit some vertical edges for clarity

- Most partial ordered sets (posets) are **not** lattices:

names of the nodes, not elements in a set

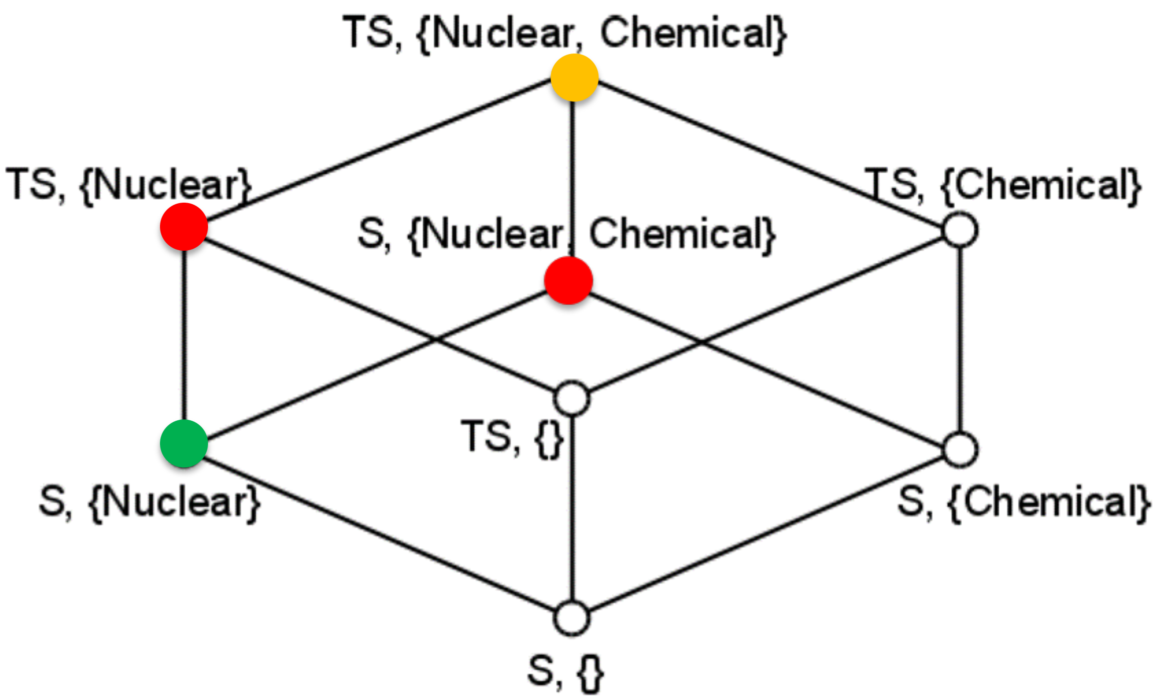




c and d: no LUB; a and b: no GLB

b and c: upper bounds d, e, and f, but none of them is LUB

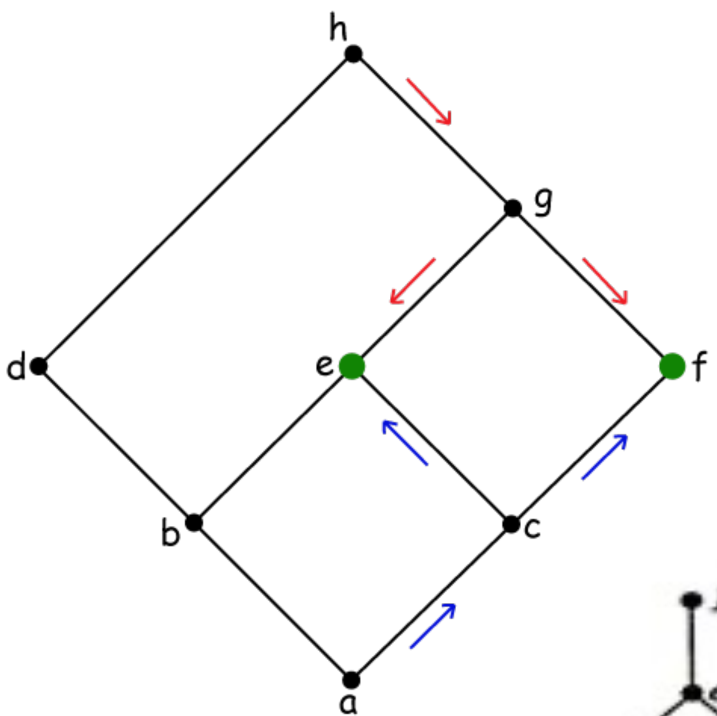
BLP Lattice Example



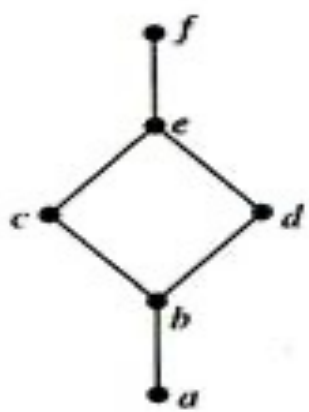
lub? $lub((TS, \{Nuclear\}), (S, \{Nuclear, Chemical\})) = (TS, \{Nuclear, Chemical\})$

glb? $glb((TS, \{Nuclear\}), (S, \{Nuclear, Chemical\})) = (S, \{Nuclear\})$

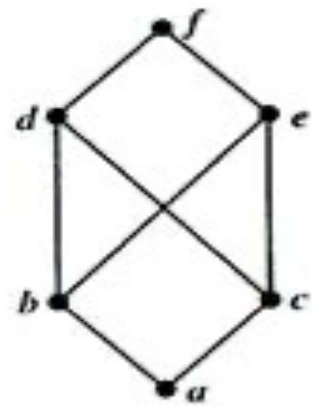
How can we determine GLB and LUB?



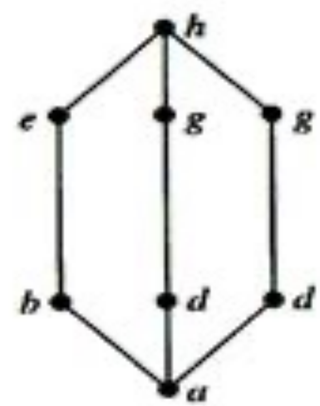
Which of these are lattices?



a.



b

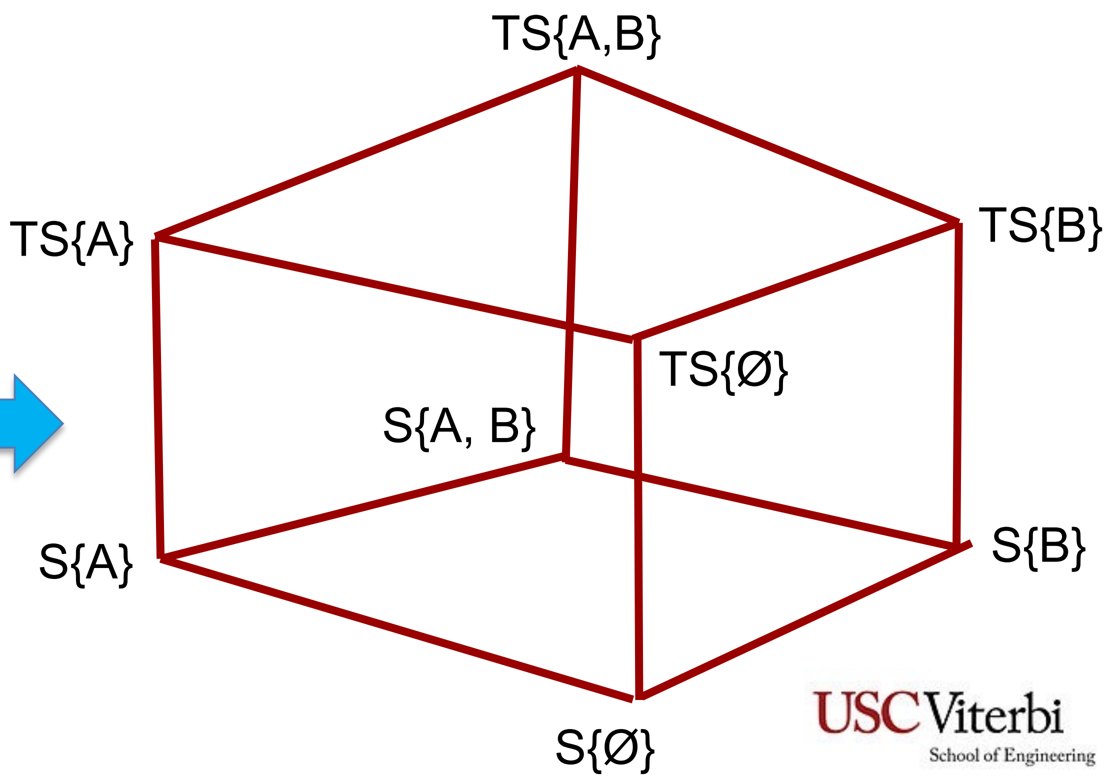


c

Example: Lattice Product

- Product of 2 lattices is a lattice
- Hierarchical classes with compartments (categories):
  - Levels: TS (top secret), S (secret), TS > S
  - Categories: A and B
  - Sets of categories: Ø, {A}, {B}, {A, B}

```
TS          {A,B}
|          /     \
S       {A}       {B}
           \     /
              Ø
```



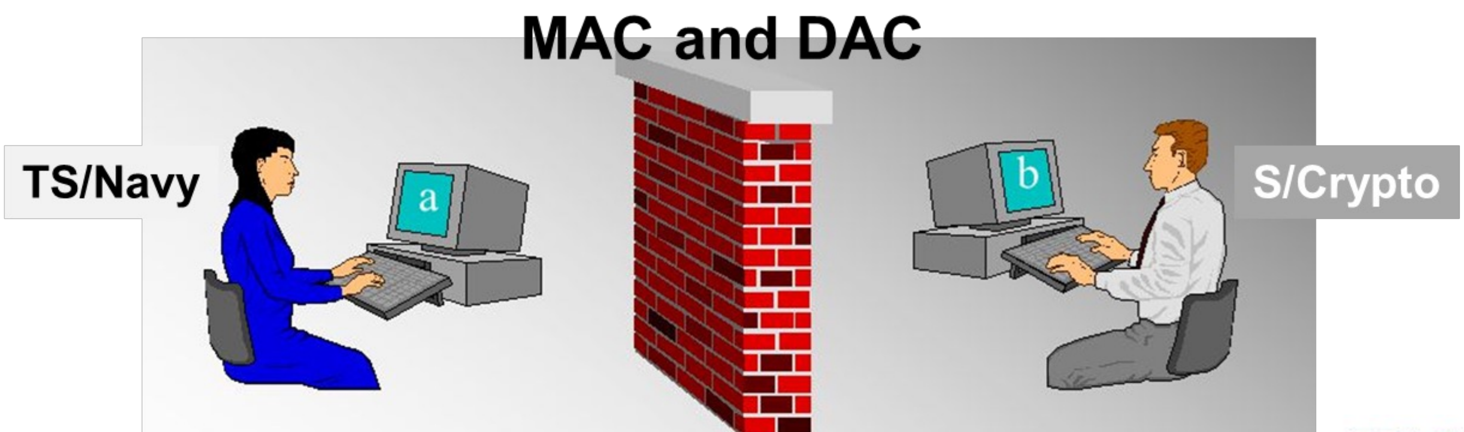
Outline

- Review
- Bell-LaPadula (BLP) model
  - Preliminary concepts: inductive reasoning, MAC access classes, lattice structures
  - Formal BLP definition
  - Refinements to BLP (tranquility, communicating down)
- Bell-LaPadula system interpretation
- U. S. Classified Information Policy

USC Viterbi School of Engineering

Bell-LaPadula Model (BLP): Background

- Time sharing environment
  - Sharing computer systems across security levels
- The need for **sound** notion of security
  - Limitations of penetrate and patch
- The first mathematical model of a multilevel security policy
- The best known and highly influential model
  - Published in 1973
- BLP used to build trusted OS, e.g., Multics and GEMSOS



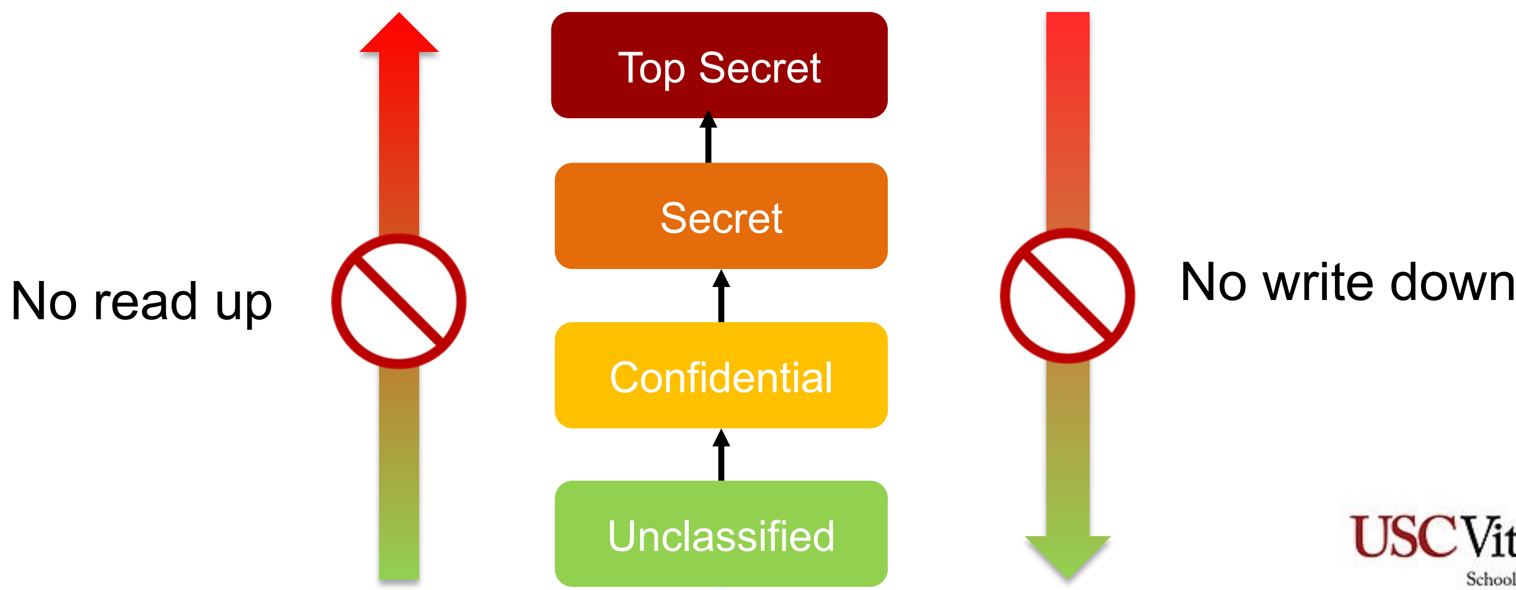
USC Viterbi School of Engineering

Bell-LaPadula Model

- BLP is the most widely used example of a FSPM
- Formalizes mandatory policy for confidentiality
- It corresponds to military-style environments
  - Subjects and objects are often partitioned into different security levels
  - Subject can only access objects at certain levels determined by its security level
  - Examples:
    - “Unclassified personnel cannot read data at confidential levels”
    - Top-Secret data cannot be written into the files at unclassified levels”
- **Goal**: prevent information flow to lower or incomparable security classes
- **Idea**: augment DAC with MAC to enforce information flow policies
- Two-step approach
  1. Operations authorized by MAC policy, over which users have no control
  2. Discretionary access matrix

Bell-LaPadula Model (informally)

- A state machine model that enforces the confidentiality aspects of access control, but not integrity or availability
- Considers *semantics* of the information
- Summarized in two axioms:
  1. No user may read information classified above his/her clearance level ("No read up")
  2. No user may lower the classification of information ("No write down")

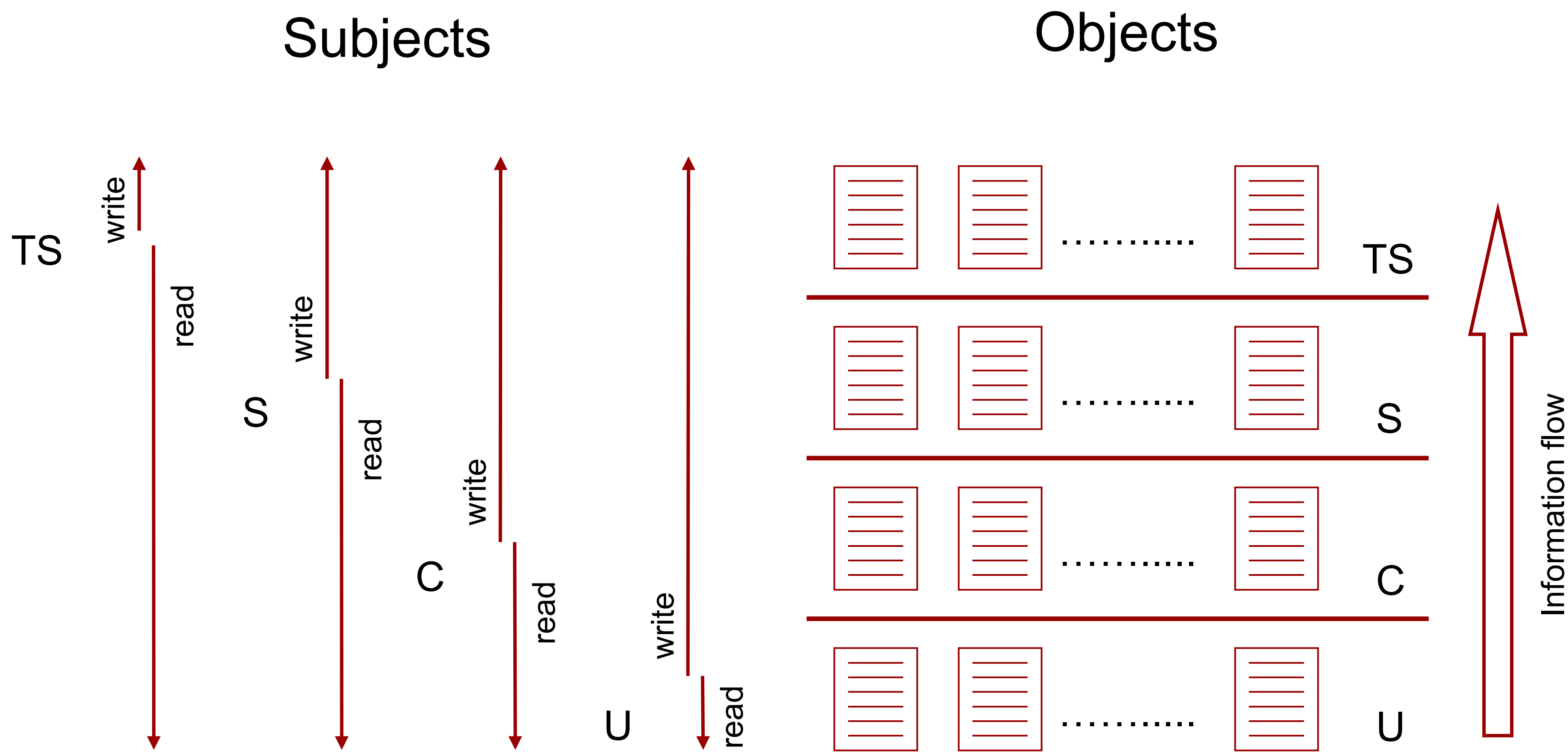


USC Viterbi School of Engineering

Bell-LaPadula Model (Formally)

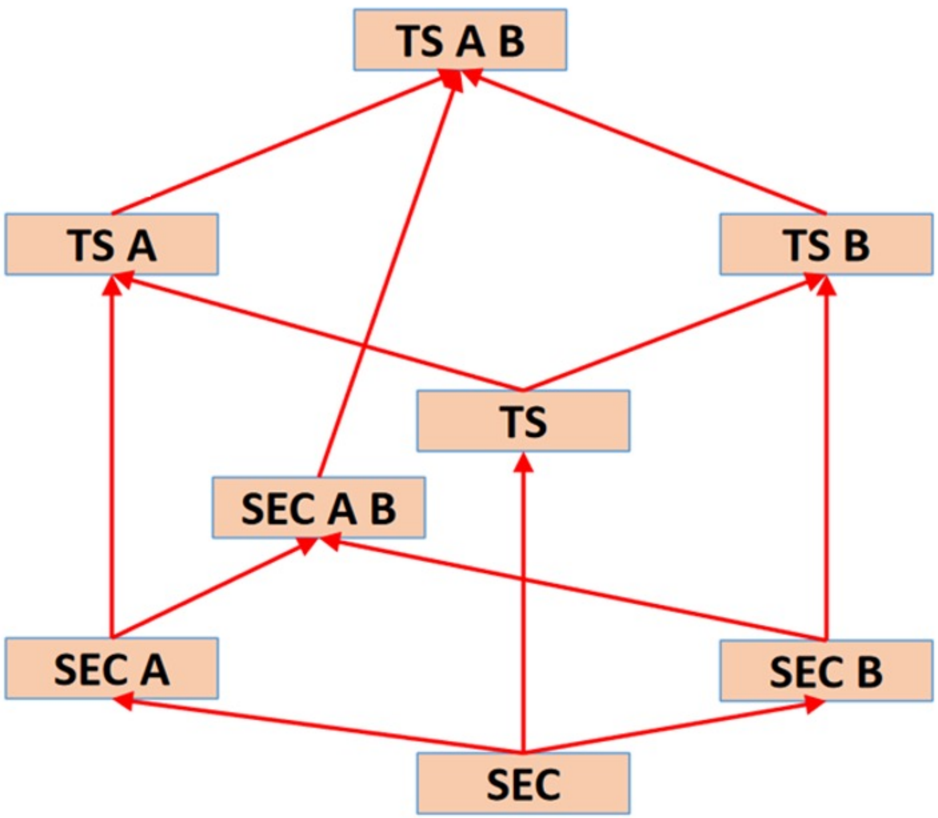
- Simple Security Condition:
  - subject with access class S can **read** object with access class O IFF S *dom* O ("No read up")
- *-Property (or Confinement property):
  - subject with access class S can **write** object with access class O IFF O *dom* S ("No write down")
- The Discretionary Security Property:
  - access must be permitted by the access control matrix
  - allows users to grant access to other users at **the same** clearance level

BLP Information Flow



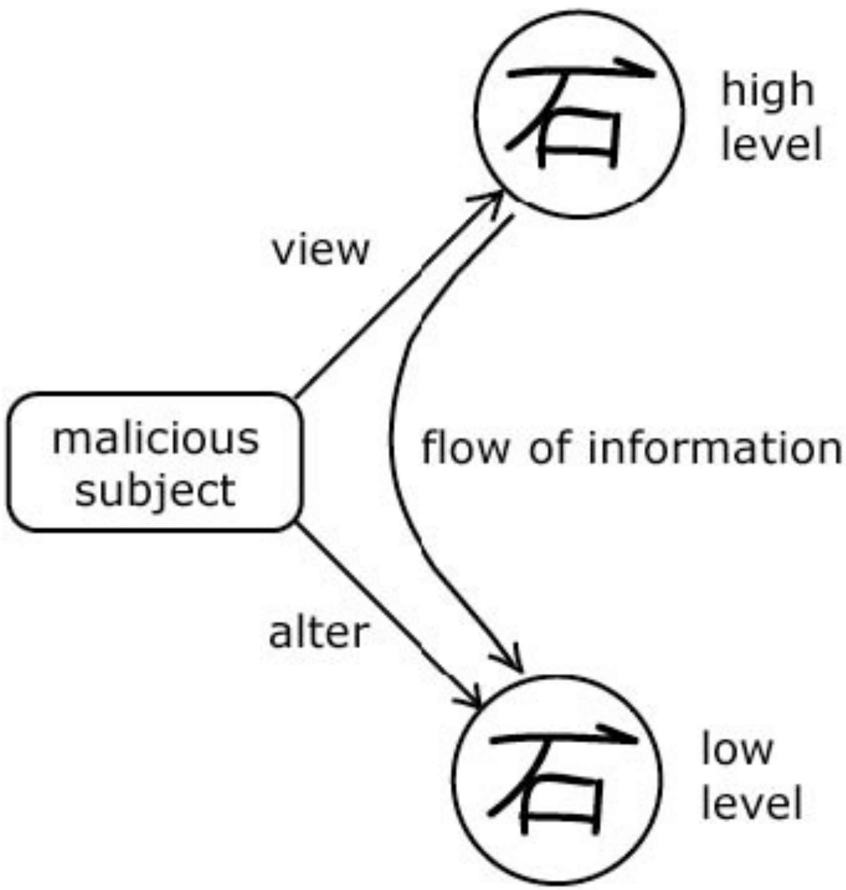
Example: Information Flow in BLP

- Alice is cleared to (TS, {A})
- A document may be classified as (TS, {B})
- Will Alice be able to access the document?



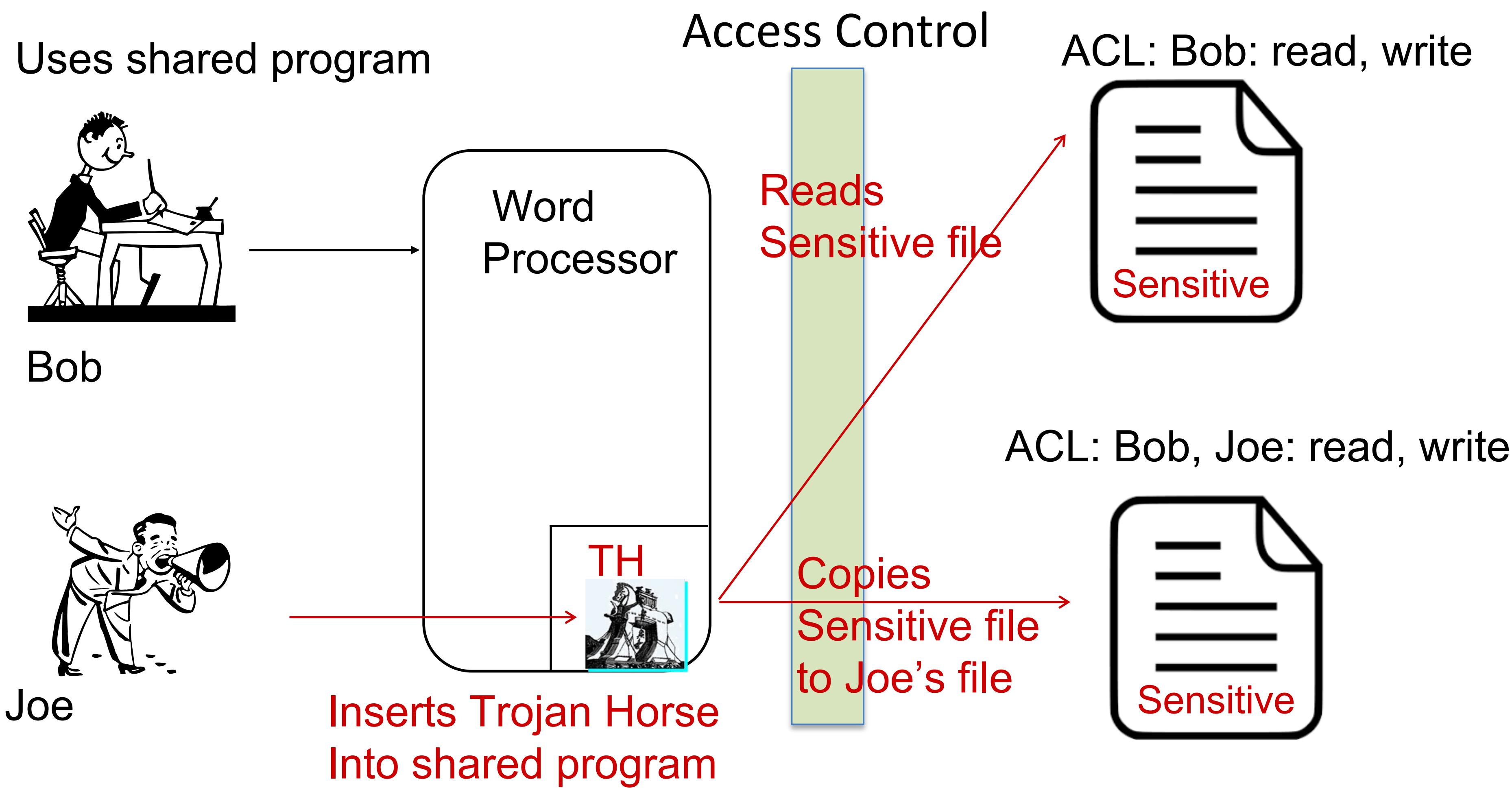
*-Property vs. Trojan Horses

- Want to prevent Trojan Horse from violating security policy
- Trojan horse blocked by *-property because can't write

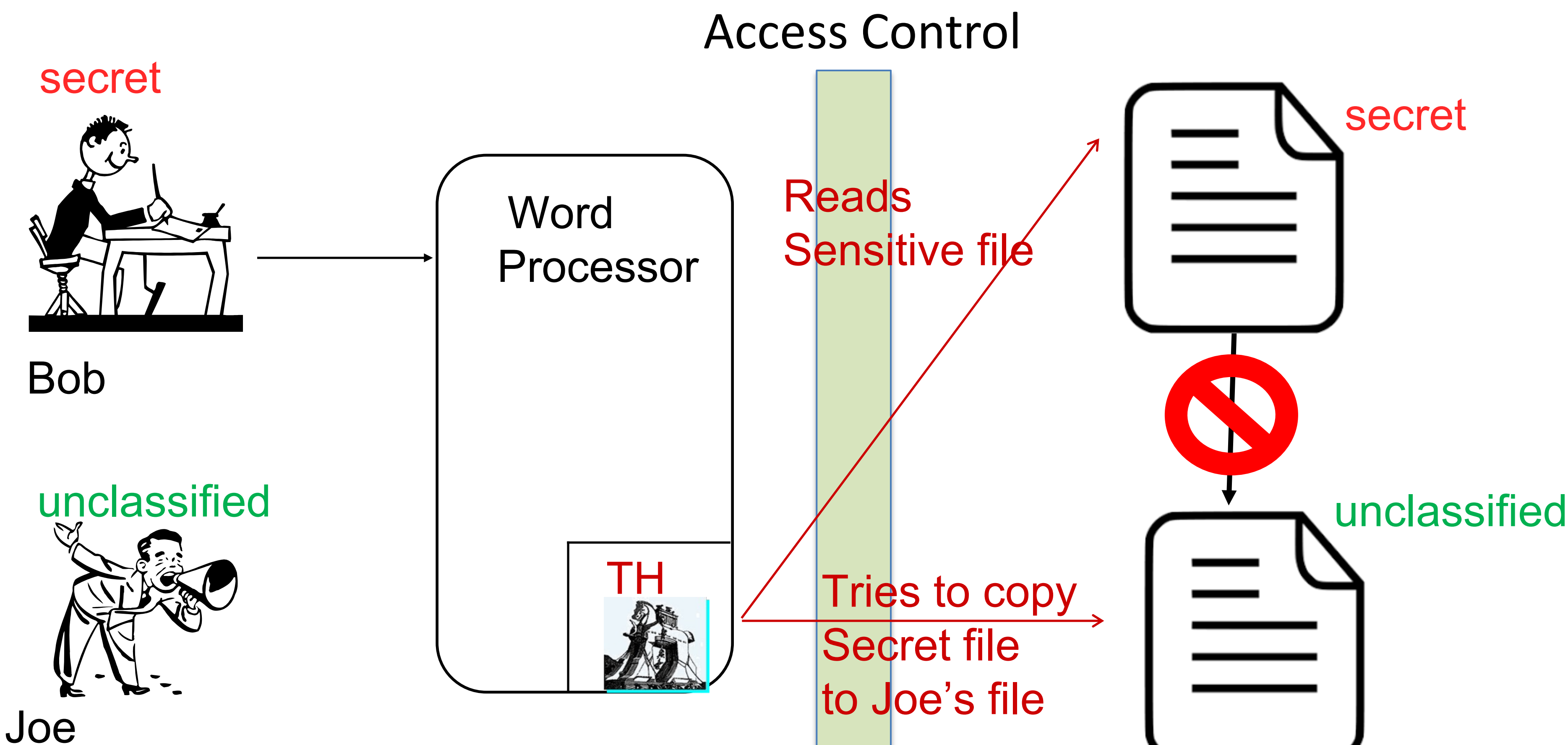


USC Viterbi School of Engineering

Example: Trojan Horse and DAC



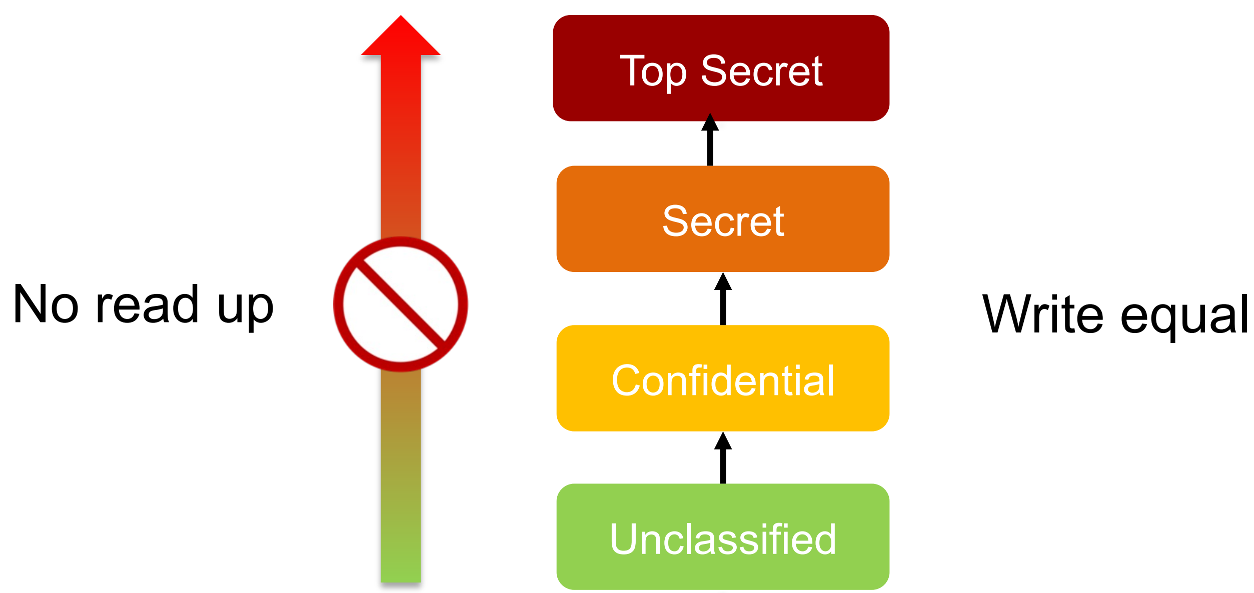
Example: Trojan Horse and MAC



USC Viterbi School of Engineering

BLP: Strong Star Property

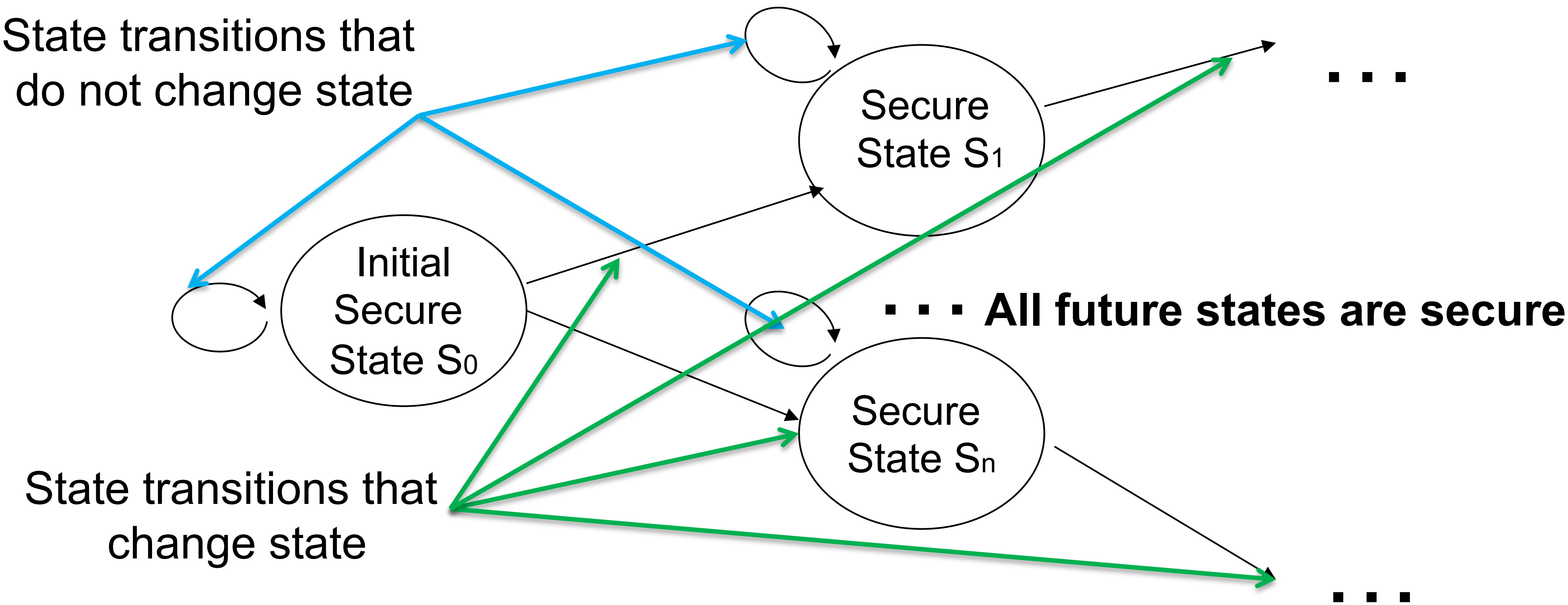
- The Strong Star Property makes the limitations even more stringent, as it changes “no read up, no write down” to “no read up, write only to same”



BLP System Security

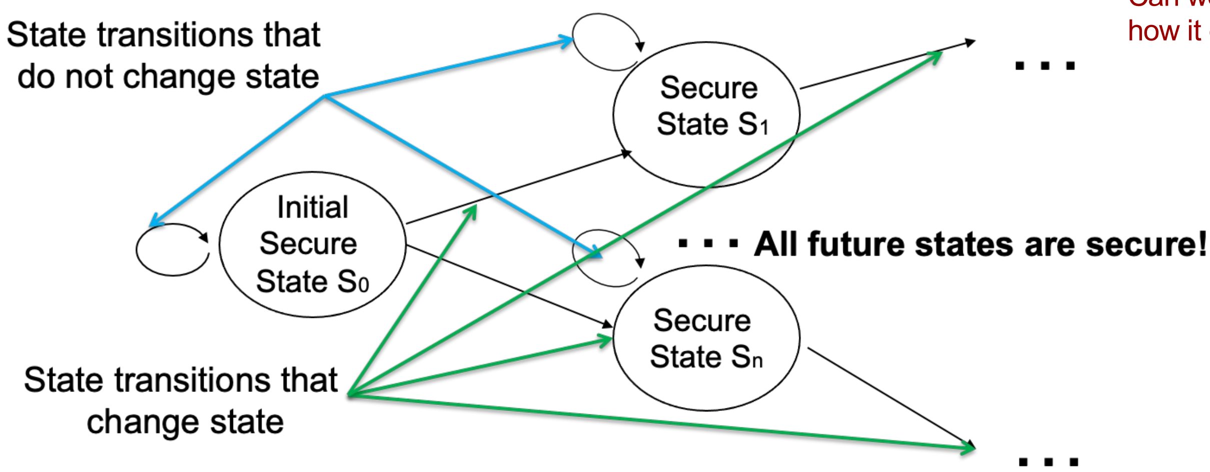
- System modeled as a finite state machine (set of states and transition functions)
- A state contains information about the current security labels and authorizations
- A transition transforms a state into another state
  - Adding or removing authorizations
- A state is **secure** if the current authorizations satisfy the simple, *, and DAC security properties
- A system is secure if (starting from a secure initial state) every reachable state is secure

BLP as a State Machine

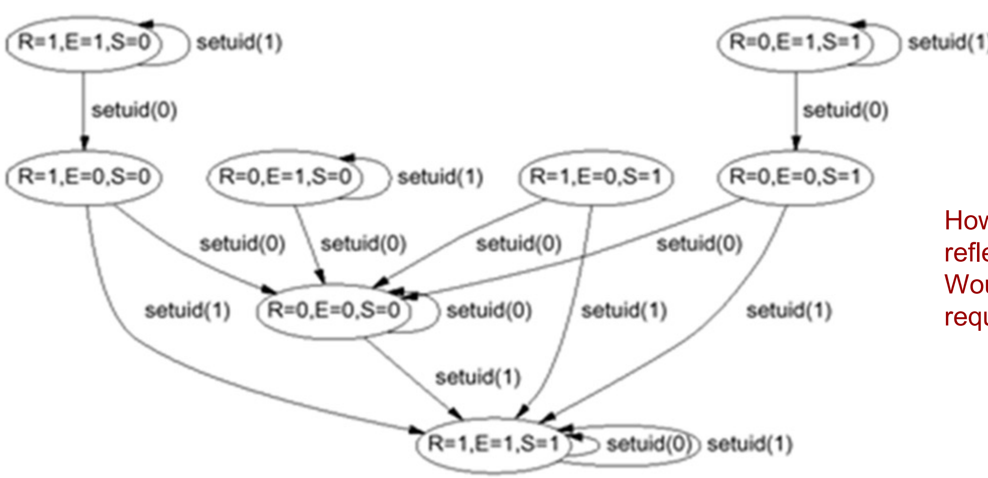


The basic security theorem has nothing to do with the BLP security policies, only with the state machine modeling

BLP vs. `setuid` Formal Model



Can we please review setuid and how it compares to BLP model?



How would you make the Unix setuid reflective of a proper security model? Would adding which states satisfy security requirements be sufficient?

Outline

- Review
- Bell-LaPadula (BLP) model
  - Preliminary concepts: inductive reasoning, MAC access classes, lattice structures
  - Formal BLP definition
  - Refinements to BLP (tranquility, communicating down)
- Bell-LaPadula system interpretation
- U. S. Classified Information Policy

Communicating Down: Problem

- *-property protects against TH but causes difficulties in practice
- How to communicate from a higher security level to a lower one?
- Example:
  - Colonel has (Secret, {NUC, EUR}) clearance
  - Major has (Secret, {EUR}) clearance
  - Major can talk to colonel (write up)
  - Colonel cannot talk to major (no write down)

Communicating Down: Solution

- A subject has a **maximum** security level and a **current** security level
- Maximum security level must dominate current security level
- Temporarily downgrade high level subject
  - Colonel's maximum security level is (Secret, {NUC, EUR})
  - He changes the current security level to (Secret, {EUR})
  - Now he can create document at Major's clearance level (Secret, {EUR})

Tranquility Principle

- Tranquility principle:
  – Subjects and objects may not change their security levels once they have been instantiated
- **Strong** tranquility: security labels never change during system operation
  – Advantage: system state always satisfies security requirements
  – Disadvantage: not flexible
- **Weak** tranquility: labels never change in such a way as to violate a defined security policy (for BLP simple-security and *)
  – e.g., dynamic upgrade of labels (least privilege)

USC Viterbi School of Engineering

Characteristics of Access Classes

- MAC access class of information is *global*
  - Has same sensitivity regardless of where it is
- MAC access class of information is *persistent*
  - Same sensitivity at all times, i.e., labels are tranquil
  - Declassification?

USC Viterbi School of Engineering

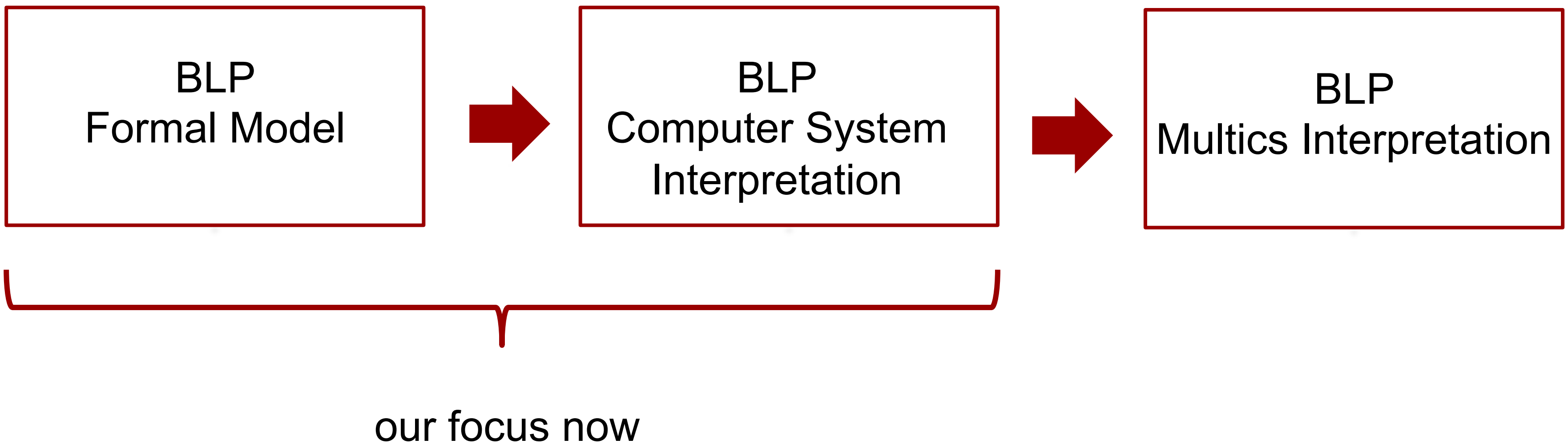
BLP Model Limitations

- Restricted to confidentiality of content
  - How about origin confidentiality?
- No policies for changing access rights
  - Does not work well for commercial systems
    - Users given access to data as needed:
      - would require large number of categories and classifications
    - Centralized handling of "security clearances" intended for systems with static security levels
- Does not protect against *covert channels*
  - A low subject can detect the existence of high objects when it is denied access

Outline

- Review
- Bell-LaPadula (BLP) model
  - Preliminary concepts: inductive reasoning, MAC access classes, lattice structures
  - Formal BLP definition
  - Refinements to BLP (tranquility, communicating down)
- Bell-LaPadula system interpretation
- U. S. Classified Information Policy

BLP Interpretation



USC Viterbi
School of Engineering

BLP Abstraction for System State

- The **state** of the system is (b, M, f, H) where:
  - **b** indicates which subjects can access which objects
    - The set of rights that may <u>actually</u> be exercised in the current state
  - **M** is the access control matrix for the current state
    - DAC
    - Note that MAC can make some rights unusable
  - **f** is tuple indicating subject and object access classes
    - MAC
    - For subjects: max and current clearances
  - **H** is the *hierarchy* of objects (for <u>naming</u> objects)
    - Can't do access control unless can uniquely name objects
- Recall that we have defined a "secure" state
  - MAC (Simple security, *-property) and DAC

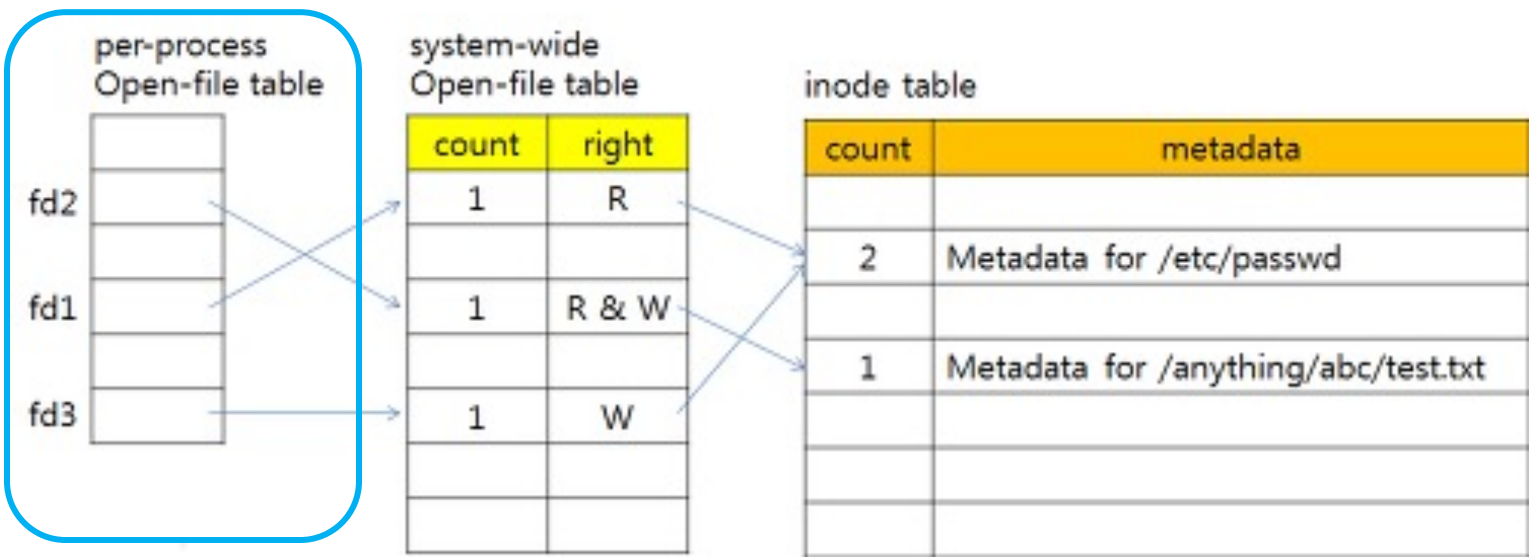
System State: Current Access b

- Modes of access are called **access attributes**
  - Execute, e (neither observation nor alteration)
  - Read, r (observation with no alteration)
  - Append, a (alteration with no observation)
  - Write, w (both observation and alteration)
- Current access by subject to an object is a triple:

  <subject, object, access-attribute>
- Set b is the set of triples for all current accesses

Current Access b: Unix Example

- Per-process open file table
  - A file descriptor is an index into this table
  - Flags to indicate read/write access, etc.
  - Each entry contains a pointer to the kernel system-wide file table
- Open file table - one entry for each file opened by any process
  - Open count – number of processes that have file open
  - inode describes a file, stores file's attributes and disk block locations



System State: Access Permission M

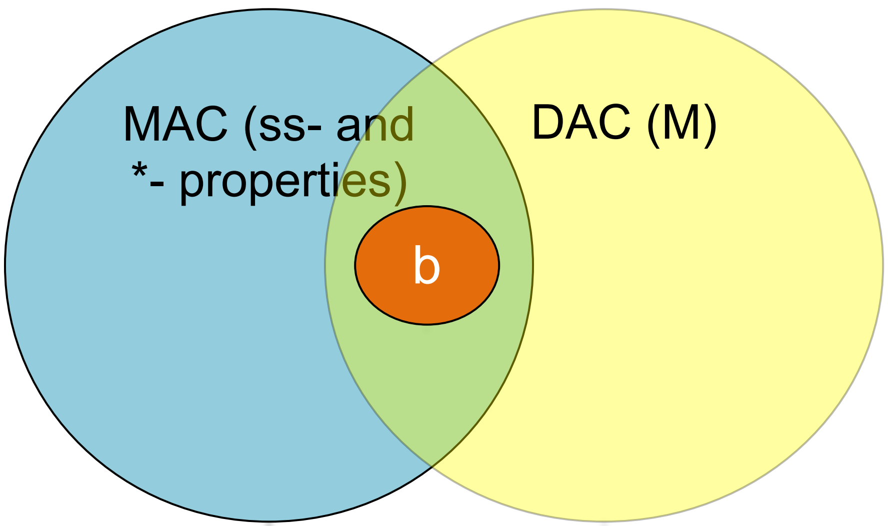
- DAC

Subjects \ Objects	Object 1	Object 2	Object 3	Object 4
Bob Process	read	read, write		write
Flo Process	read, write	write		
Alice Process	read	read	read	read, write, **own**
Dan Process	read		read, write	read

- Entries are only "permissions"
  - May not be current access in b
  - MAC (access class) may make some modes unusable
    - Example: Alice has Secret clearance; Object 4 is Confidential, Alice can read but not write Object 4

Relationship between b, M, and MAC

- b is a subset of all accesses that a subject can have in the system based on DAC (M) and MAC (ss- and *-properties)
- M can contain some access permissions that can be disallowed by MAC and, therefore, will never be reflected in b
- MAC permissions can be further restricted by DAC (enforcing “need to know”)



Example: MAC and DAC

- Further restriction: Alice removes Bob's W access to File4 in M
- MAC overrides DAC: Alice grants Bob R access to File4, but Bob cannot read it because of f

M

DAC

Objects / Subjects	File1	File2	File3	File4	File5
Alice	R	R	RW	RW Own	RW Own
Bob	RW	W	W	RW	
Carol	RW	W	RW	RW	R

MAC

f_s

Clearance / Subjects	Max	Current
Alice	Top Secret, Navy	Secret
Bob	Confidential	Confidential
Carol	Top Secret	Secret

f_o

Objects	Classification
File1	Confidential
File2	Secret
File3	Top Secret
File4	Top Secret
File5	Secret, Navy

When will granting DAC right succeed in this example?

Access control decision = MAC & DAC

USC Viterbi School of Engineering

System State: Level Function <u>f</u>

- Embodiment of access classes in model
- The level function encompasses three functions:
  1. Function f_o assigns the access class of an object
  2. Function f_s assigns **maximum** access class of subject
  3. Function f_c assigns **current** access class of a subject
- For any subject S, required that $f_s(S)$ *dom* $f_c(S)$
- Triple (f_s, f_o, f_c) is called level function, denoted <u>f</u>

Relationship between M, f, and b

Access control decision = MAC & DAC

M

Subjects \ Objects	File1	File2	File3	File4
Alice	R	R	RW	RW Own
Bob	RW	W	W	~~RW~~

DAC

f_s

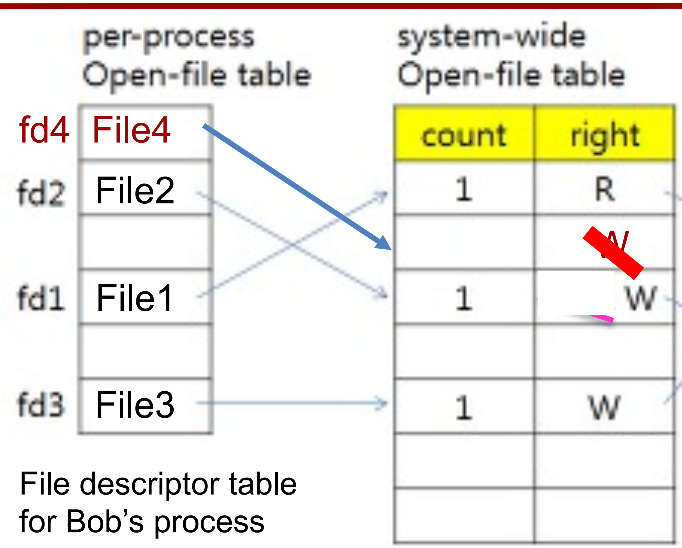
Subjects \ Clearance	Max	Current
Alice	Top Secret	Secret
Bob	Confidential	Confidential

f_o

Objects	Classification
File1	Confidential
File2	Secret
File3	Top Secret
File4	Top Secret

MAC

b



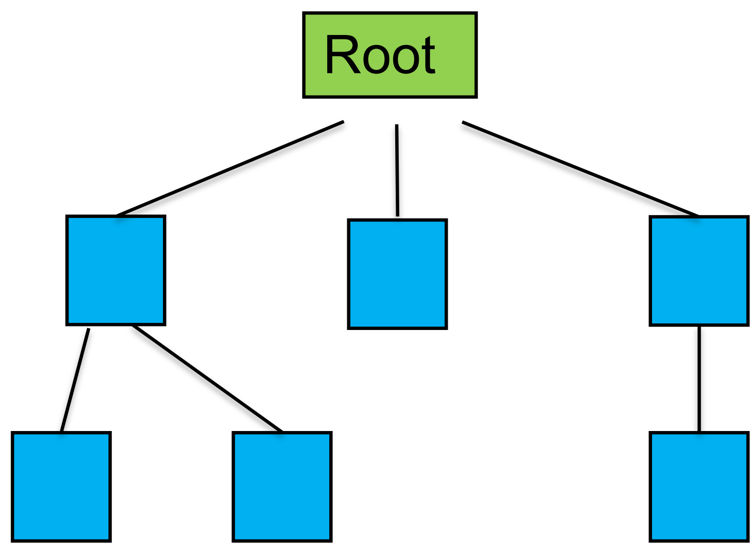
File descriptor table for Bob's process

"Cache"

USC Viterbi School of Engineering

System State: Hierarchy H

- Unique object ID number is impractical and insecure
  - may become very large over life of system
  - either duplicate in each domain or covert channel
- “H” Reflects structure imposed on objects
  - child/parent allows only directed, rooted trees
- Control of parent permits control of access to child object



System Representation

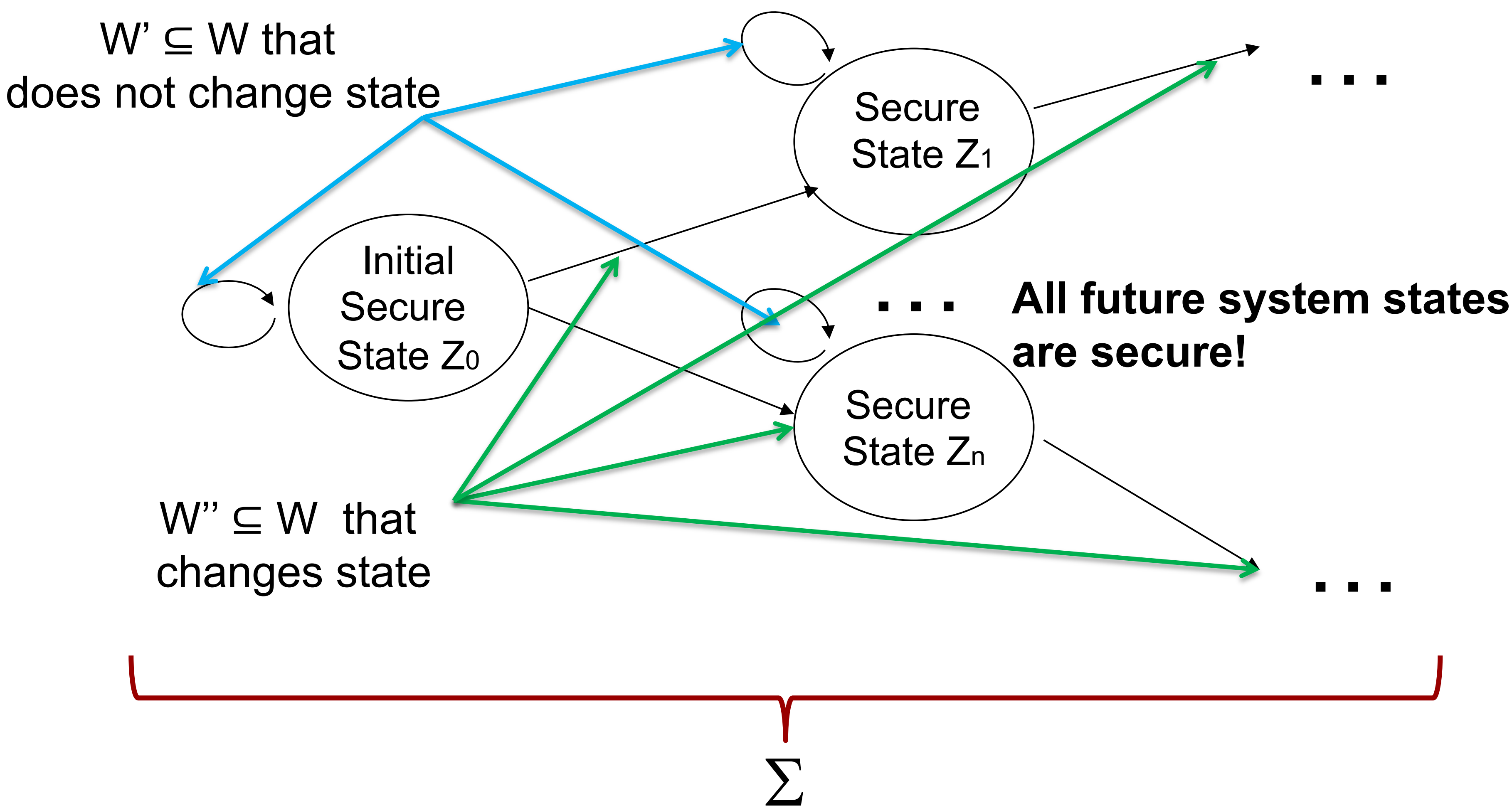
- System is represented by $\Sigma(R, D, W, z_0)$ where:
  - R denotes the set of requests for access (inputs)
  - D denotes the set of outcomes (outputs)
    - (y)es, (n)o, (i)llegal, (o) error
  - W is the set of actions of the system
    - subset of (R x D x V x V), where V is set of states
    - system moves from a state in V to another (possibly different) state in V
    - example of W – kernel calls
  - z_0 is the initial state of the system
- System is all sequences of <request, decision, state> triples, with initial state z_0

USC Viterbi School of Engineering

Basic Security Theorem

- Basic Security Theorem – proof that Σ is secure
- System is secure if z_0 is a secure state and W:
  - satisfies the simple security condition
  - satisfies the *-property
  - satisfies the discretionary security property
- A state is secure, if all current access tuples are permitted by the 3 BLP properties
- A state transition is secure if it goes from a secure state to a secure state
- If the initial state of a system is secure and if all state transitions are secure, then the system will always be secure

BLP as a State Machine: System View



- Defined **all possible** transitions (W), each transition satisfies the 3 properties

USC Viterbi School of Engineering

Optional Reading

- [BLP-96] in Resources

Secure Computer Systems:
Mathematical Foundations

November, 1996

An electronic reconstruction
by
Len LaPadula
of
the original

MITRE Technical Report 2547, Volume I
titled "Secure Computer Systems: Mathematical Foundations
by D. Elliott Bell and Leonard J. LaPadula
dated 1 March 1973

ABSTRACT

This paper reports the first results of an investigation into solutions to problems of security in computer systems; it establishes the basis for rigorous investigation by providing a general descriptive model of a computer system.

Borrowing basic concepts and constructs from general systems theory, we present a basic result concerning security in computer systems, using precise notions of "security" and "compromise". We also demonstrate how a change in requirements can be reflected in the resulting mathematical model.

A lengthy introductory section is included in order to bridge the gap between general systems theory and practical problem solving.

Proof of the Basic Security Theorem

Basic Security Theorem:
Let $W \subseteq R \times D \times V \times V$ be any relation such that $(R_i, D_j, (b^*, M^*, f^*), (b, M, f)) \in W$ implies

(i) $f = f^*$ and
(ii) every $(S,O) \in b^* - b$ satisfies SC rel f^*.

$\Sigma(R,D,W,z)$ is a secure system for any secure state z.

Proof: Let $z_0 = (b,M,f)$ be secure. Pick $(x,y,z) \in \Sigma(R,D,W,z)$ and write $z_t = (b^{(t)}, M^{(t)}, f^{(t)})$ for each $t \in T$.

z_1 is a secure state. $(x_1,y_1,z_1,z) \in W$. Thus by (i), $f^{(1)} = f$. By (ii), every (S,O) in $b^{(1)} - b$ satisfies SC rel $f^{(1)}$. Since z is secure, every $(S,O) \in b$ satisfies SC rel f. Since $f = f^{(1)}$, every $(S,O) \in b^{(1)}$ satisfies SC rel $f^{(1)}$. That is z_1 is secure.

If z_{t-1} is secure, z_t is secure. $(x_t,y_t,z_t,z_{t-1}) \in W$. Thus by (i), $f^{(t)} = f^{(t-1)}$. By (ii), every (S,O) in $b^{(t)} - b^{(t-1)}$ satisfies SC rel $f^{(t)}$. Since z_{t-1} is secure, every $(S,O) \in b^{(t-1)}$ satisfies SC rel $f^{(t-1)}$. Since $f^{(t)} = f^{(t-1)}$, every $(S,O) \in b^{(t)}$ satisfies SC rel $f^{(t)}$. That is, z_t is secure. By induction, z is secure so that (x,y,z) is a secure appearance. (x,y,z) being arbitrary, $\Sigma(R,D,W,z_0)$ is secure.

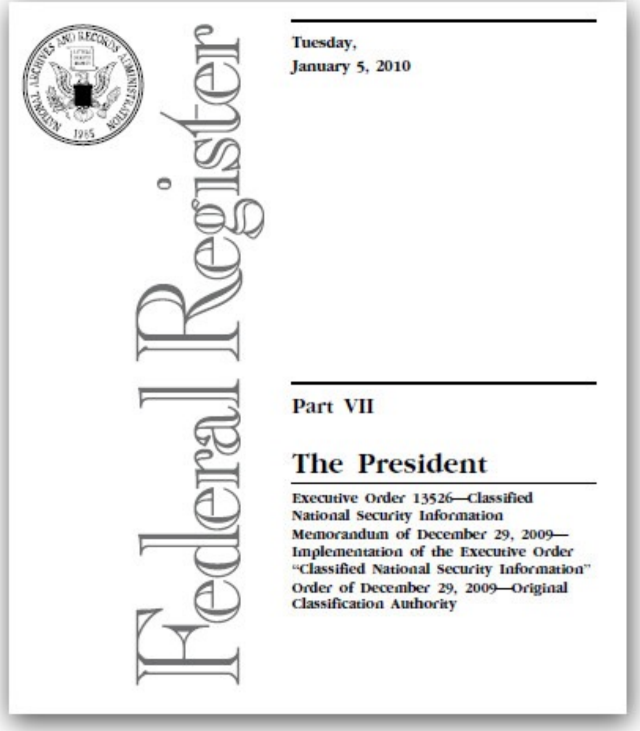
Search docur

Outline

- Review
- Bell-LaPadula (BLP) model
  - Preliminary concepts: inductive reasoning, MAC access classes, lattice structures
  - Formal BLP definition
  - Refinements to BLP (tranquility, communicating down)
- Bell-LaPadula system interpretation
- U. S. Classified Information Policy

U. S. Classified Information Executive Order 13526

- Policy for classified national security information (CNSI)
- Presidential Executive Order 13526 delivers a unified method for designation classification, protecting and declassifying national security information
- Only Original Classification Authority (OCA) can classify
- Three factors are considered before implementation:
  1. Level of damage to national security
  2. Existing or anticipated threat to disclosure
  3. Long and short term costs



Original Classification Authorities

CNSI Levels

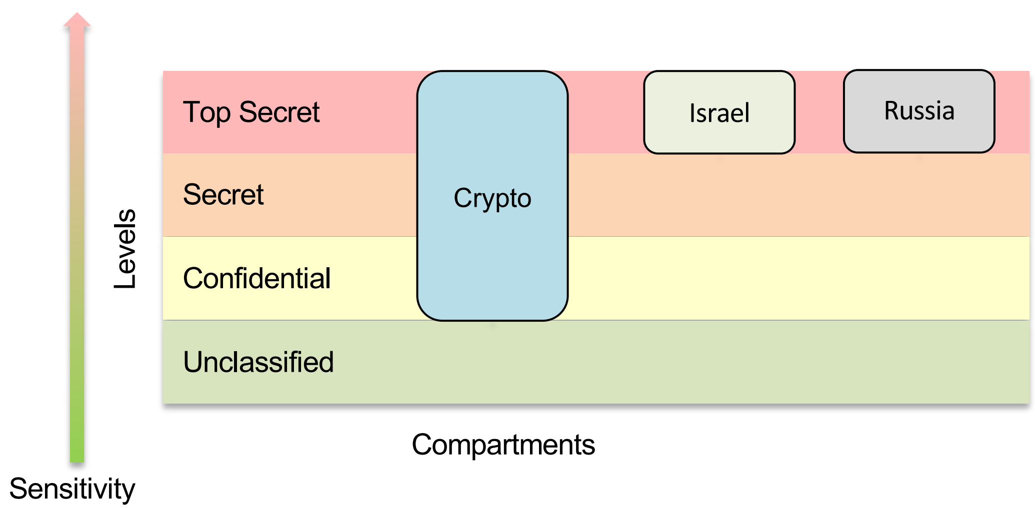
- Clear reflection of non-discretionary nature
- There are only three levels of CNSI:
  - TOP SECRET
    - Exceptionally Grave Damage to the National Security
  - SECRET
    - Serious Damage to the National Security
  - CONFIDENTIAL
    - Damage to the National Security

Reasons for Classification

1. Military plans, weapons systems, or operations
2. Foreign government information
3. Intelligence activities, sources or methods, or cryptology
4. Foreign relations or foreign activities of the US
5. Scientific/technological/economic matters relating to national security
6. US Gov. programs for safeguarding nuclear materials/facilities
7. Vulnerabilities/capabilities of systems, infrastructures, projects, or services relating to the national security
8. Development/production/use of weapons of mass destruction

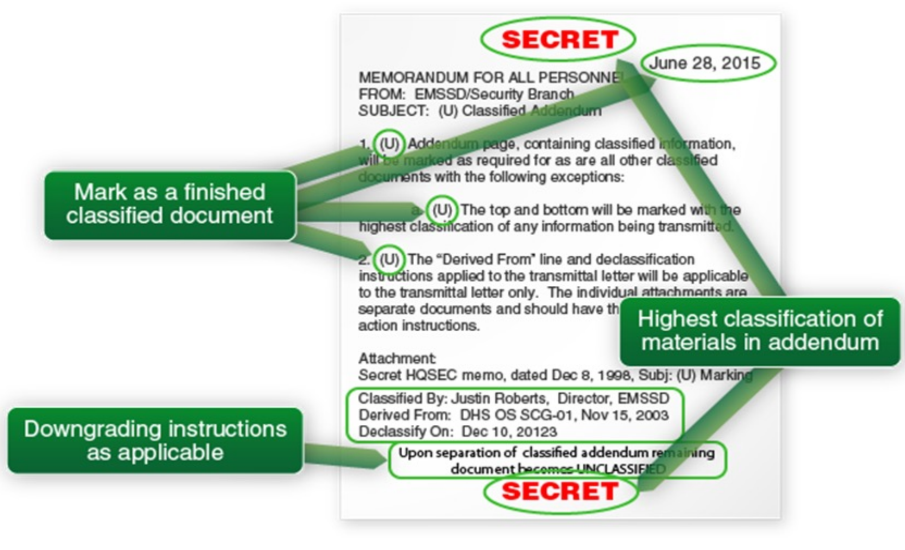
Special Access Programs (Categories)

- For "enhanced protection"
- Protect information classified at same level
  - Normal criteria for access not deemed sufficient
  - Number of persons having access reasonably small
- Only senior managers may create special access program, but with constraints
  - Keep number of programs at absolute minimum
  - Requires exceptional vulnerability or threat
  - E.g., cryptographic keys
- Sometimes called "formalized need-to-know"



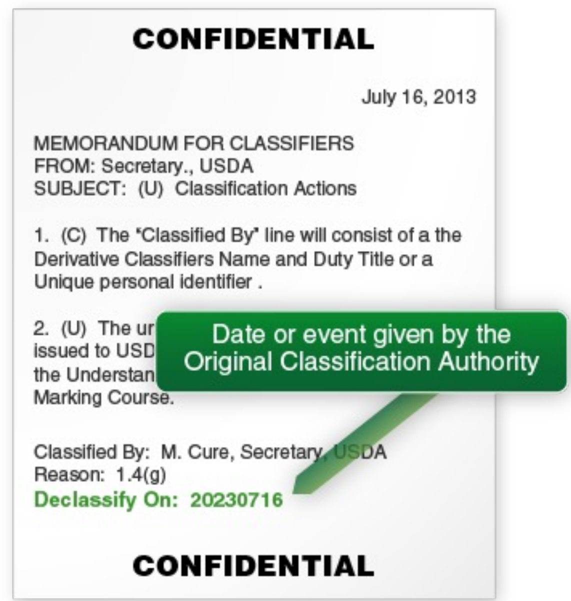
Policy for Label Integrity

- Clear identification and markings requirements
- Indicated in a manner immediately apparent
  - One of the three classification levels
- Whenever practicable, use classified addendum
  - Maximize available info and simplify doc handling
- For public release, declassified records marked



Policy for “Persistence” Access Class

- Explicit duration of assigned object classification
  - Mandatory policy is global and <u>persistent</u>
- Specific date or event for declassification
  - Upon date or event, be automatically declassified
- Default is declassification 10 years from decision
  - May have a downgrade “schedule”
    - E.g., downgrade from TS to S after 10 years, etc.
  - Exemptions
    - Human source
    - Key design concepts of weapons of mass destruction
- No information may remain classified indefinitely
- Facilitate the public release of declassified info


CONFIDENTIAL

July 16, 2013

MEMORANDUM FOR CLASSIFIERS
FROM: Secretary., USDA
SUBJECT: (U) Classification Actions

1. (C) The "Classified By" line will consist of a the Derivative Classifiers Name and Duty Title or a Unique personal identifier .

2. (U) The ur
issued to USD
the Understan
Marking Course.

Classified By: M. Cure, Secretary, USDA
Reason: 1.4(g)
Declassify On: 20230716

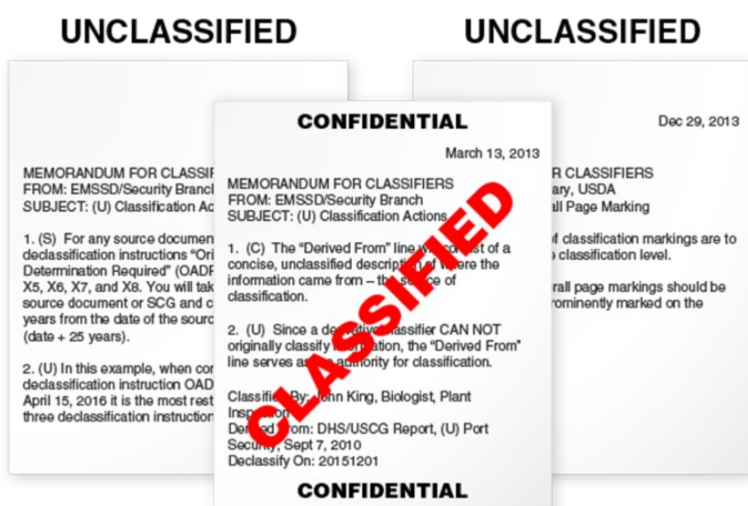
CONFIDENTIAL

Complexities

- Aggregation
  - The whole is more sensitive than the parts
- Derivation
  - Reproduction, extraction or summarization
- Labeling is not always straightforward
- Procedures for people and Automatic Data Processing (ADP) differ

Aggregation

- Aggregations (compilations) of items may result in a classification higher than the constituent items!
  - If reveals an additional association or relationship
- E.g., “phone book problem”
  - Can know one person’s number
  - Entire phone book may reveal
    - number of employees in the organization
    - who works in the same departments or locations
    - other associations



USC Viterbi School of Engineering

Aggregation Example

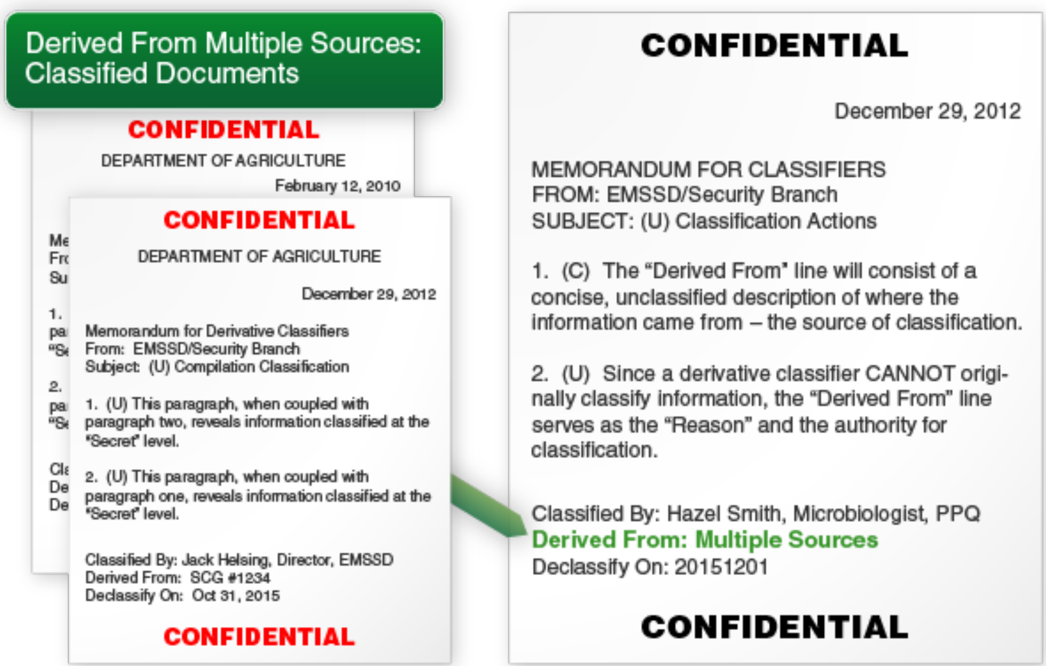
- Consider a thread of emails about a classified meeting in a meeting room that is outside a security area (not

Derivation

- Data Derivation refers to the process of creating a data value from one or more contributing data values through a data derivation algorithm
- Want global, persistent label
  - But derived data is most common
    - Data may be processed; not in original form
    - Copied, extracted, or summarized
    - E.g., redaction (filtering), census statistics
- New classification derived from
  - Original classification
  - Guidelines
  - Judgment
- Human is responsible for decision
  - Provide training in derivative classification
  - Release data with noise

Policy for "Global" Access Class on Derived Data

- Observe and respect original classification
- List source materials (unless *exists* is sensitive)
- Use a classified addendum
- Provide training in derivative classification
  - Can't generally or reliably mechanize
  - Human is responsible for decision



Access Control Policy for Individual

- Define controls on access to information
- Person who has met the standards for access
  - Has signed an approved nondisclosure agreement
  - Has a need-to-know the information
- Every person must receive up-to-date training
  - Proper safeguarding of classified information
    - E.g., how to send securely
  - Policy should define “proper” safeguarding
- Criminal, civil, and administrative sanctions if fail

Automated Information Systems

- Uniform procedures systems and networks
  - Collect, create, communicate classified information
  - Compute, disseminate, process, or store
  - Ensure the integrity of the information
- Common standards, protocols, and interfaces
  - Standardized formats
    - How do you implement labels? Which bits?
- Establish controls to ensure adequate protection
  - Information is used, processed, stored, reproduced,
  - Conditions under which transmitted and destroyed
  - Prevent access by unauthorized persons

0	1	1	0	0	1	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Big Endian
= 0x64 = 100

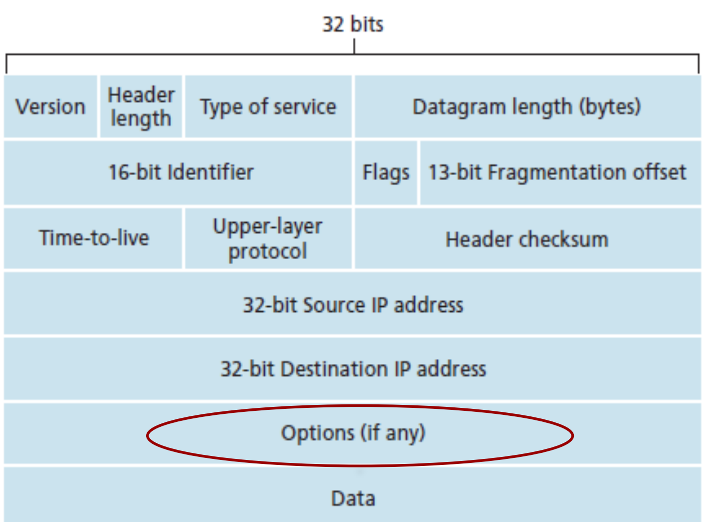
0	1	1	0	0	1	0	0
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7

Little Endian
= 0x26 = 38

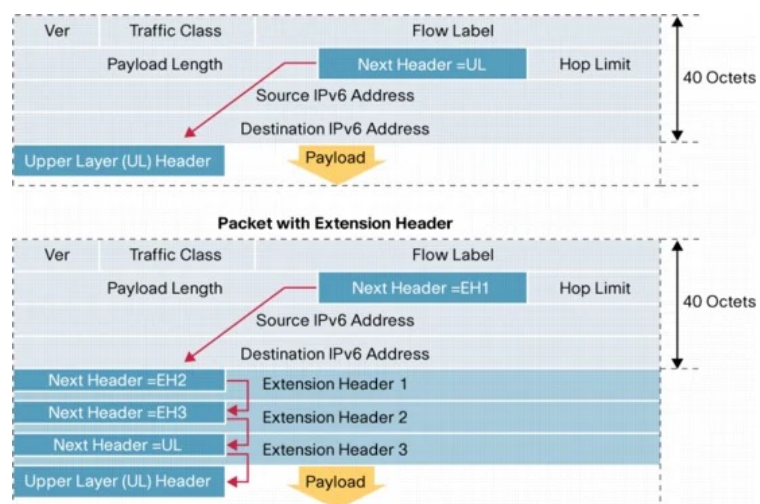
MAC Network Labeling

- Standards developed and implemented to share data between systems that have implemented MAC (Solaris, Linux, etc.)
  - Application of labels at an IP packet level
1. Commercial IP Security Option (CIPSO)
   - https://tools.ietf.org/html/draft-ietf-cipso-ipsecurity-01
   - Never officially adopted
   - IPv4 labels (Options 130, 133, 134)
2. Common Architecture Label IPv6 Security Option (CALIPSO)
   - https://tools.ietf.org/html/rfc5570
   - IPv6 labels
- CIPSO and CALIPSO are only useful on internally controlled networks or VPN solutions due to risk of label alteration

IPv4 header



IPv6 header



Next Steps Towards Assurance

- Interpret organizational policy for computers
  - Help your organization by making policy clear, easy to understand, and unambiguous
- Refine with Formal Security Policy Model
  - Mathematically express access control policy
    - Define “secure state”
- Apply Reference Monitor (RM) abstraction
- Interpret policy in terms of **specific** RM functions
  - E.g., with US classified information policy