DSCI 519: Foundations and Policy for Information Security

Cryptography policy, Course review

Tatyana Ryutov

Outline

- Student presentations
- Deployed network cascade problem
- Cryptography policy
- Course review

Final Exam Details

- Final exam on December 7th 11am-1pm
  - 2hrs + 15minues
  - Covers all material
  - Closed notes/books
  - One handwritten cheat sheet (both sides)
  - Exam will be accessible via D2L in Quiz section
  - I'll be available to answer your exam-related questions via Piazza

Reminder: Learning Experience Evaluation

- Please complete Learning Experience Evaluation
- USC Learning Experience Evaluation guidelines:
  - This will be your opportunity to provide feedback about your learning experience in the class
  - This feedback helps the instructor determine whether students are having the intended learning experiences for the class
  - It is important to remember that the learning process is collaborative and requires significant effort from the instructor, individual students, and the class as a whole
  - Students should provide a thoughtful assessment of their experience, as well as of their own effort, with comments focused on specific aspects of instruction or the course
  - Comments on personal characteristics of the instructor are not appropriate and will not be considered
  - For this feedback to be as comprehensive as possible, all students should complete the evaluation

DSCI-525 Topics

- Goal: **balance old and new**
  - Security kernel (SK)
    - HW & SW that implements Reference Monitor (RM)
    - SK design, analysis, development
    - GEMSOS A1 security kernel
  - Course significantly about software engineering
    - How to build a verifiable security kernel and how to develop secure software in general
  - Current topics in HW & SW security
  - Case studies

- Secure SW development topics
  - Development best practices
    - Principles, methods, and technologies to make SW more secure
    - Typical threats and vulnerabilities in SW, and how to avoid them
  - Low-level, memory-based attacks and defenses
  - Principles of secure and robust coding
  - Use of automated tools to analyze and test existing code and reduce vulnerabilities
    - Automated code review with static analysis and symbolic execution
  - Penetration testing: an overview of goals, techniques, and tools
- Secure HW topics
  - Trusted Platform Module (TPM)
  - A Physical Unclonable Function (PUF)
  - HW Security
    - Vulnerabilities and attacks
    - Techniques for building trusted HW

CSCI-531 Topics

- Information Security Objectives
- Brief history of encryption and cryptanalysis
- Mathematical background
  - Discrete probability
  - Introduction to Number Theory
- Definitions of security:
  - Perfect secrecy (OTP)
  - Information Theoretic Security (Shannon)
  - Semantic security (stream ciphers)
- Cryptographic functions
  - Encryption
  - Message authentication and data integrity techniques
  - Identification/entity authentication techniques
  - Digital signatures

- Cryptographic building blocks
  - Stream ciphers
  - Block ciphers
  - Public-key encryption
  - One-way hash functions
  - Message authentication codes
  - Signature schemes
- Infrastructure techniques and applications
  - Identification and authentication
  - Key establishment protocols and key management
  - PKI
- Case studies:
  - Authentication service, Kerberos
  - Secure communications, SSL/TLS, IPsec, wireless LAN security
  - Electronic mail security, PGP
  - Crypto currency, Bitcoin
  - Anonymity, TOR (if time permits)
- Hardware–based security
  - Side channel attacks
  - Physically Unclonable Function
  - Trusted Platform Module
- Special topics
  - Quantum computing and cryptography

Presentation 22

5G SECURITY

Curtis Norris
Kaylin Martin

DSCI 519

Presentation 23

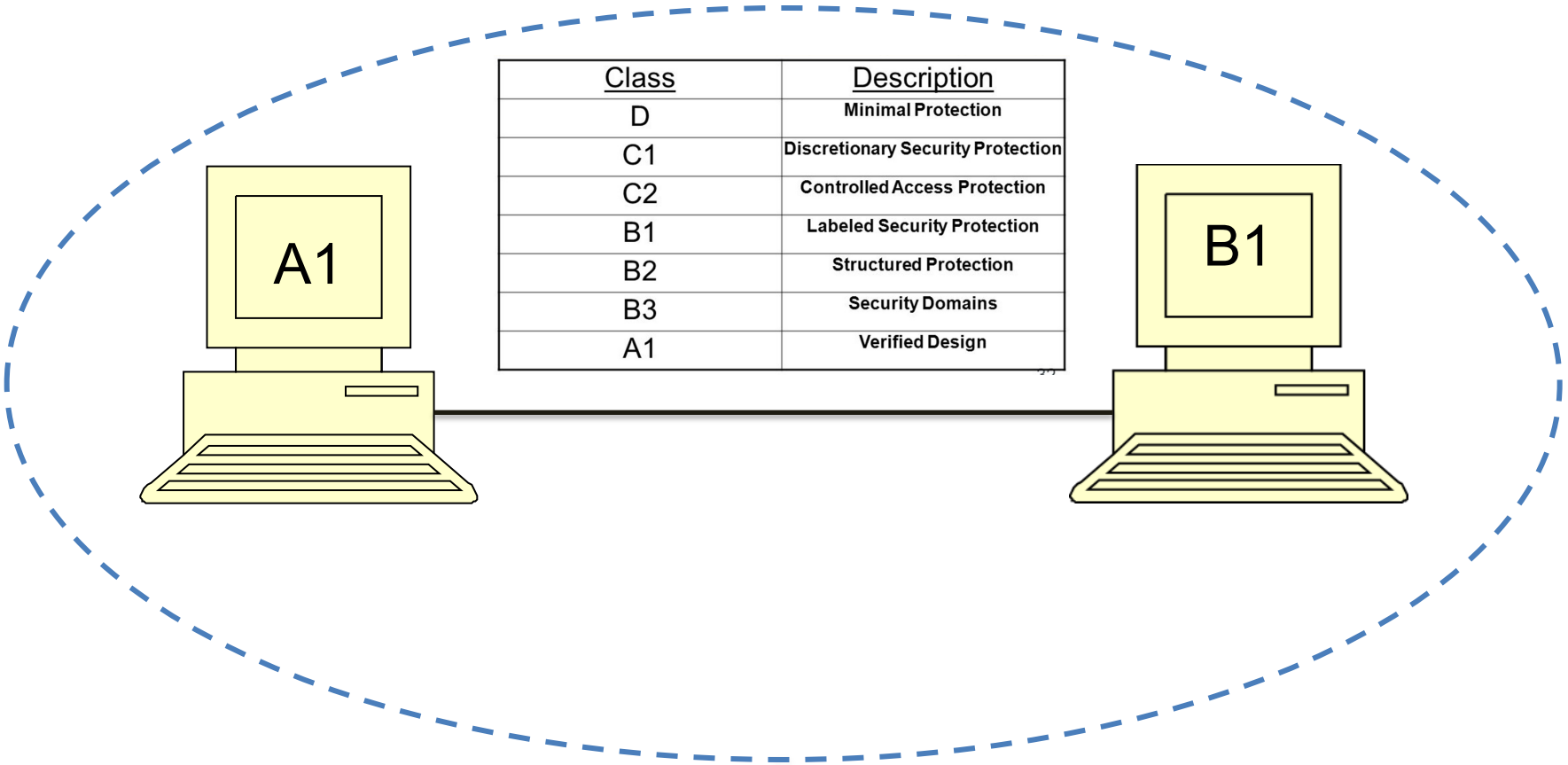
Presentation 24

Outline

- Student presentations
- Deployed network cascade problem
- Cryptography policy
- Course review

Simple Network Example

- What is the TCSEC assurance level of the overall system?



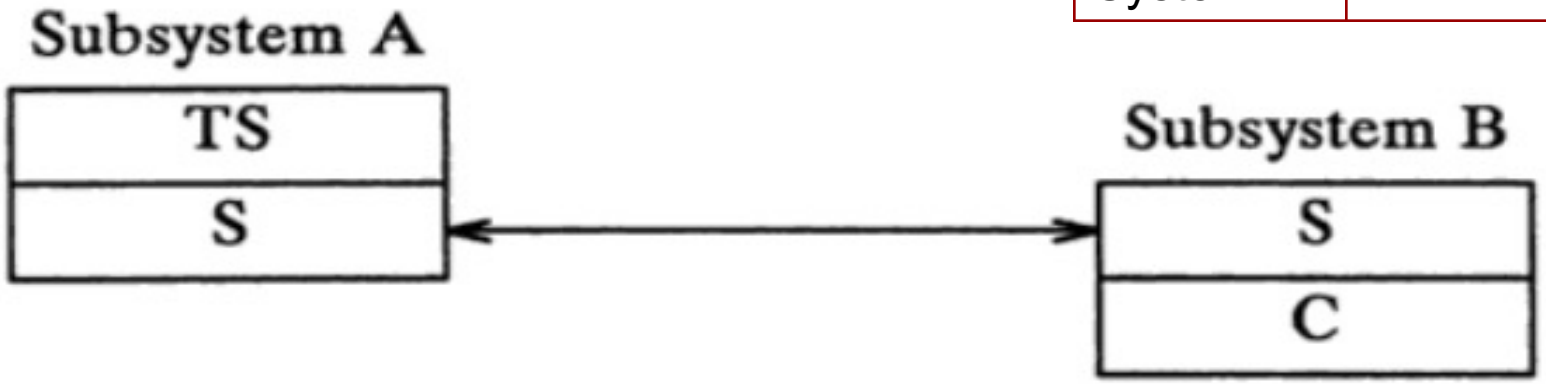
Class	Description
D	Minimal Protection
C1	Discretionary Security Protection
C2	Controlled Access Protection
B1	Labeled Security Protection
B2	Structured Protection
B3	Security Domains
A1	Verified Design

Deployed Network Cascade Problem

- Cascade problem exploits network connections
  - Enable penetrator to compromise information
    - Across range of levels greater than accredited range
  - Even when each component meets deployment policy
  - Defined in TNI Appendix C

Example:

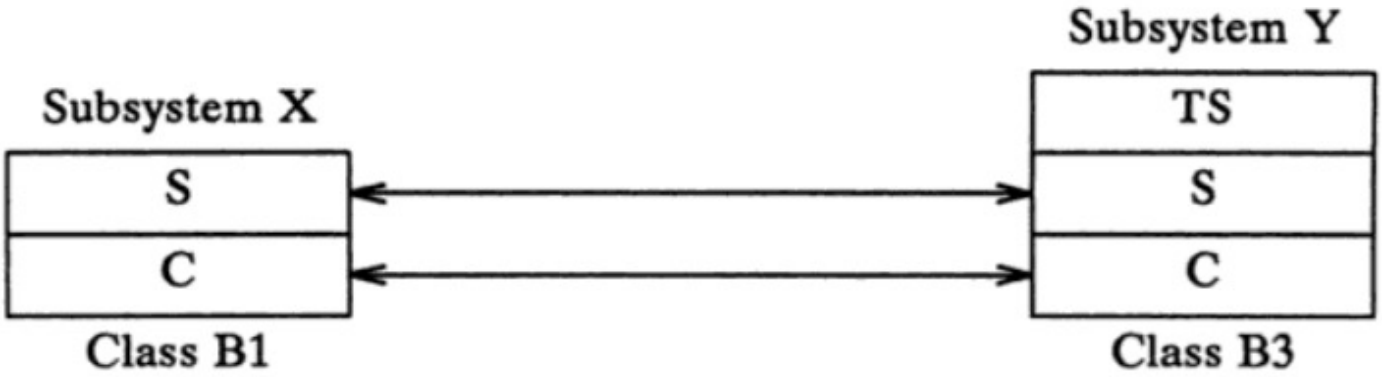
System	Level
System A	B2
System B	B1



The network connection created a TCB with users cleared to at least the C-level with data on it at the TS-level

Identification of Cascade Problem

- Simple, conservative test
  - Works only in totally ordered labels
  - Conservative: there are cases where the nesting condition fails, but there is actually no cascading problem
- If network meets nesting condition
  - Ensures network does NOT have cascading problem
- Every two components meet one of the following
  1. Ranges are disjoint – no level in common
  2. Ranges are nested – one included within the other

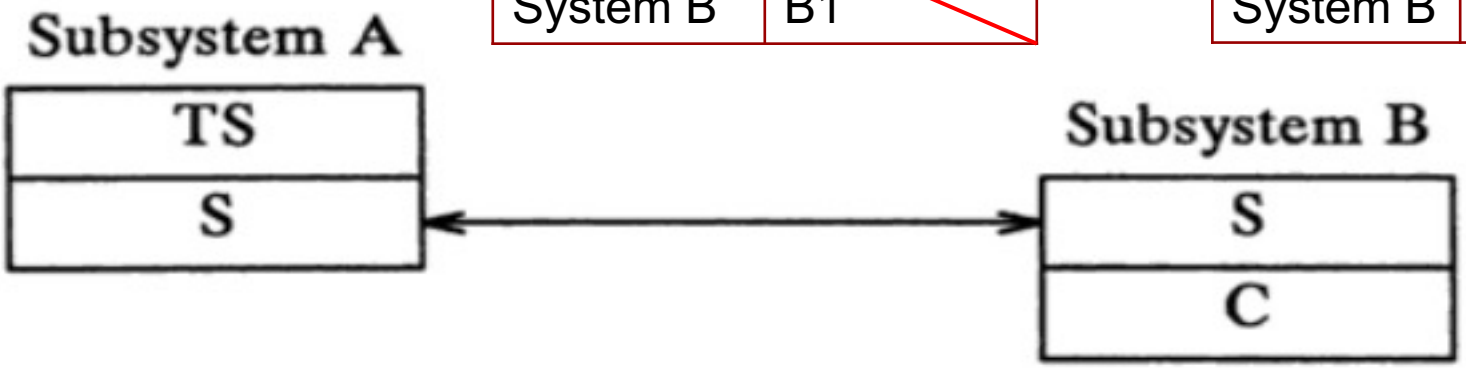


Addressing Cascade Problem

- Solutions to the cascading problem
  - Use a more trusted system at appropriate nodes
  - Physically eliminate certain connections
  - End-to-end encryption logically eliminates connection
- Using evaluated components aids analysis

System	Level
System A	B2
System B	B1

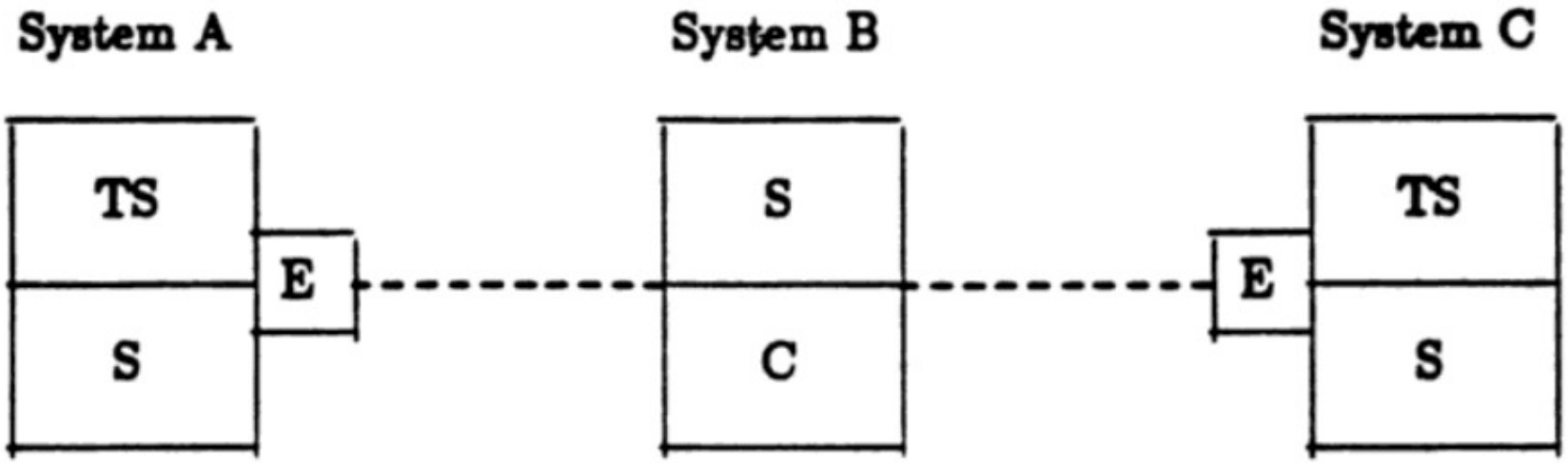
System	Level
System A	B3
System B	B3



Will encryption help here?

Cascade Problem Solution: End-to-End Encryption

- System A needs only to communicate with System C, and B is just an intermediate node



Deployment Policy Conclusions

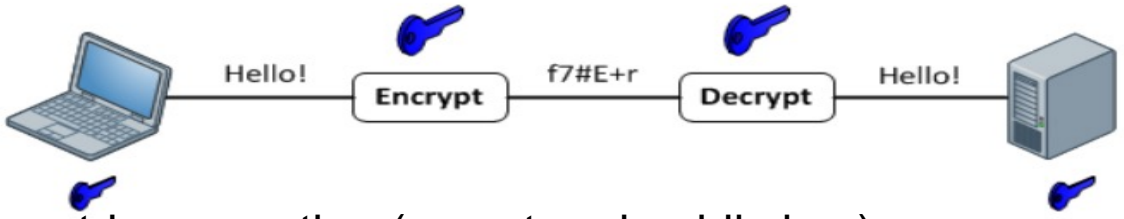
- One of the most important security policies
  - Difficult to assess risk without such a policy
  - Essential to meet other information security policies
  - Was governing DoD policy for several years
- Evaluation against sound criteria is a key factor
  - Foundation for technical certification
  - Basis for informed accreditation
- Networks require added attention to cascading
  - Systematic analysis key to avoiding cascade problem
  - Can highlight flaws in some “defense in depth”

Outline

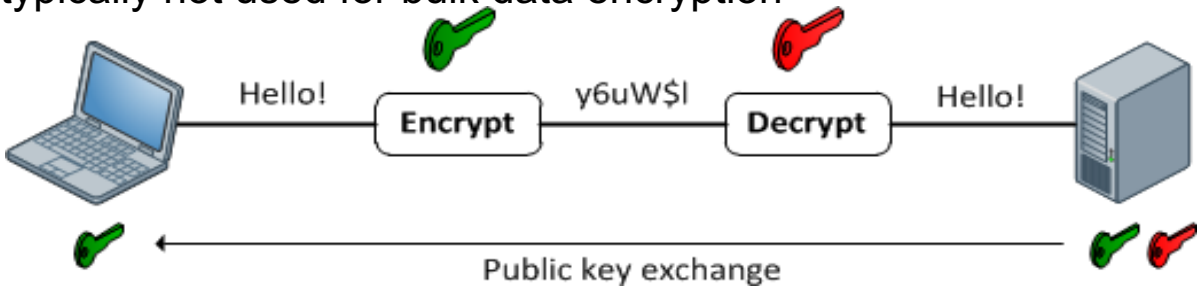
- Student presentations
- Deployed network cascade problem
- Cryptography policy
- Course review

Cryptography as a Security Tool

- Goal: keep information from those who aren't supposed to see it
  - Do this by "scrambling" the data
- Use a well-known algorithm to scramble data
  - Algorithm has two inputs: data & key
  - Key is known only to authorized users
- Based on secrets (**keys**)
  - Symmetric (shared or secret key) encryption
  - Same key is used for encryption/decryption



- Asymmetric encryption (secret and public key)
  - based on mathematical functions
  - much more compute intensive
  - typically not used for bulk data encryption



Security Goals Of Cryptography

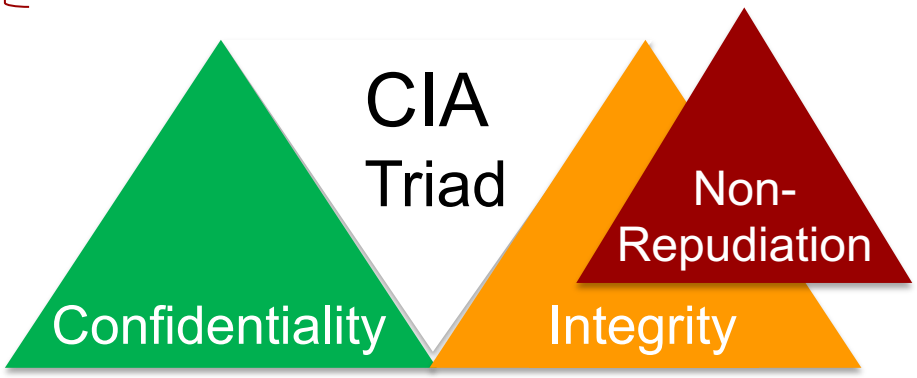
- The most basic problem: ensure security of communication over insecure medium
- Security goals:

encryption

1. Confidentiality (secrecy, privacy): only the intended recipient can see the communication

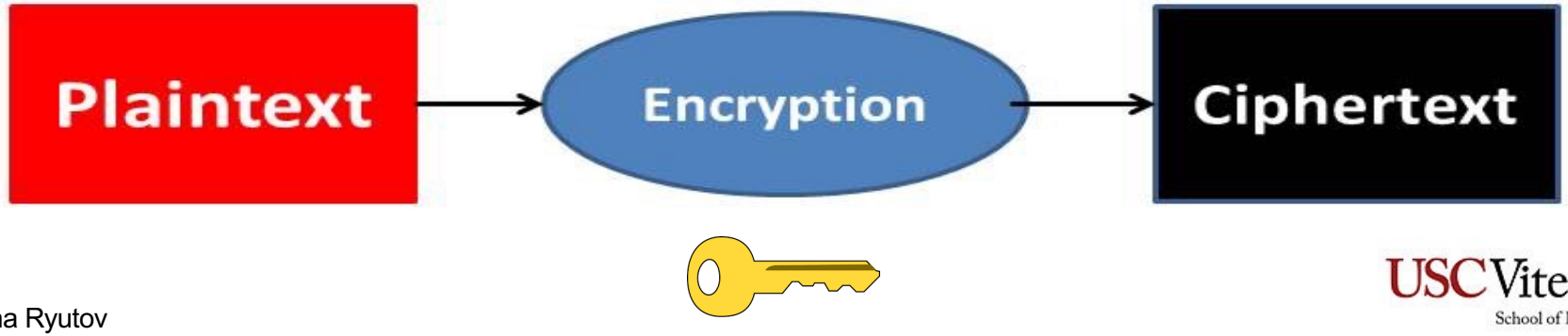
message authentication, digital signatures

2. Authenticity: the communication is generated by the intended sender information is not altered or destroyed
3. Integrity: the communication is not altered or destroyed
4. Non-repudiation: party in a dispute cannot repudiate, or refute the validity of a statement



Red/Black/Key Separation Policy

- Red is classified – “plaintext”
  - Information protected from external access
- Black is unclassified – “ciphertext”
  - Can be accessed by anyone
- What separates red from black?
  - An encryption module that does downgrade
  - Must be a device that is “trusted” to ensure separation
- Classical MAC policy for levels/categories
- Key is yet another separate level/category
  - Even more sensitive than “red” information
  - Special clearances to handle key material



Aspects of a Cryptographic Policy

- Security services to be provided
  - Encrypt, authenticate, sign, manage keys
- An access control policy for information objects
  - Type of access required for red, black, and keys
- Identification & authentication policy for subjects
  - Roles: user, crypto officer, maintenance, etc.
- Physical security policy
  - Security mechanisms – seals, "zeroization" switches
  - Actions – periodic inspection of tamper seals, etc.

Erasing all data encryption keys and other sensitive encryption information in an encryption engine

FIPS 140 Security Policy

- FIPS (Federal Information Processing Standard), NIST and the Communications Security Establishment (CSE)
  - A mandatory standard for the protection of valuable and sensitive but unclassified information throughout the government and DoD
  - Secure design and implementation of a cryptographic module
- Distinct FIPS versions evolve over time
  - FIPS 140-1 (1994)
  - FIPS 140-2 (2001) continues on through 2026
  - FIPS 120-3 (2019) **current standard**
    - FIPS 140-3 is an incremental advancement of FIPS 140-2, which now standardizes on the ISO 19790:2012 and ISO 24759:2017 specifications
- 140-3 Cryptographic Module Security Policy
  - Specification of the security rules of operation
  - Rules derived from this standard and the vendor
  - Framework for validating claims of vendors' products
  - Provides vocabulary and framework for crypto policy

Example 1: Block Cipher Algorithms

Algorithm	Status
Two-key TDEA Encryption	Disallowed
Two-key TDEA Decryption	Legacy use
Three-key TDEA Encryption	Deprecated through 2023 Disallowed after 2023
Three-key TDEA Decryption	Legacy use
SKIPJACK Encryption	Disallowed
SKIPJACK Decryption	Legacy use
AES-128 Encryption and Decryption	Acceptable
AES-192 Encryption and Decryption	Acceptable
AES-256 Encryption and Decryption	Acceptable

Table: Approval Status of Symmetric Algorithms Used for Encryption and Decryption

Example 2: Digital Signatures

Digital Signature Process	Domain Parameters	Status
Digital Signature Generation	<112 bits of security strength: DSA: (L, N) ≠ (2048, 224), (2048,256) or (3072, 256) ECDSA: len(n) < 224 RSA: len(n) < 2048	Disallowed
	≥ 112 bits of security strength: DSA: (L, N) = (2048, 224), (2048,256) or (3072, 256) ECDSA or EdDSA: len(n) ≥ 224 RSA: len(n) ≥ 2048	Acceptable
Digital Signature Verification	< 112 bits of security strength: DSA32: ((512 ≤ L < 2048) or (160 ≤ N < 224)) ECDSA: 160 ≤ len(n) < 224 RSA: 1024 ≤ len(n) < 2048	Legacy use
	≥ 112 bits of security strength: DSA: (L, N) = (2048, 224), (2048,256) or (3072, 256) ECDSA and EdDSA: len(n) ≥ 224 RSA: len(n) ≥ 2048	Acceptable

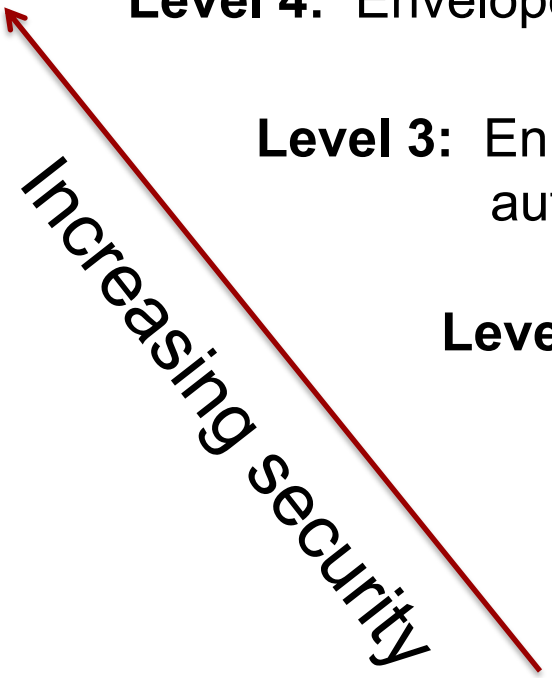
Example 3: Hash Functions

Hash Function	Use	Status
SHA-1	Digital signature generation	Disallowed, except where specifically allowed by NIST protocol-specific guidance
	Digital signature verification	Legacy use
	Non-digital signature applications	Acceptable
SHA-2 family (SHA224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256)	Acceptable for all hash function applications	
SHA-3 family(SHA3-224, SHA3-256, SHA3-384, and SHA3-512)	Acceptable for all hash function applications	
TupleHash and ParallelHash	Acceptable for the purposes specified in SP 800-185	

Table: Approval Status of Hash Functions

FIPS Security Levels

- Levels specifies “implementation robustness”



Level 4: Envelope and environmental protection

Level 3: Enhanced physical security, user based authentication

Level 2: Physical Tamper evidence, role based authentication

Level 1: Basic security requirements

FIPS 140-(2 and 3) Security Level 1

- Software/firmware on unevaluated OS
- Basic security requirements
  - Very limited requirements
  - Approved cryptographic algorithms and approved modes of operation
  - Specification of all interfaces and all input/output data paths
  - Production grade equipment
    - Allows crypto functions to be done on a PC
  - No physical security requirements
  - Example: PC encryption board

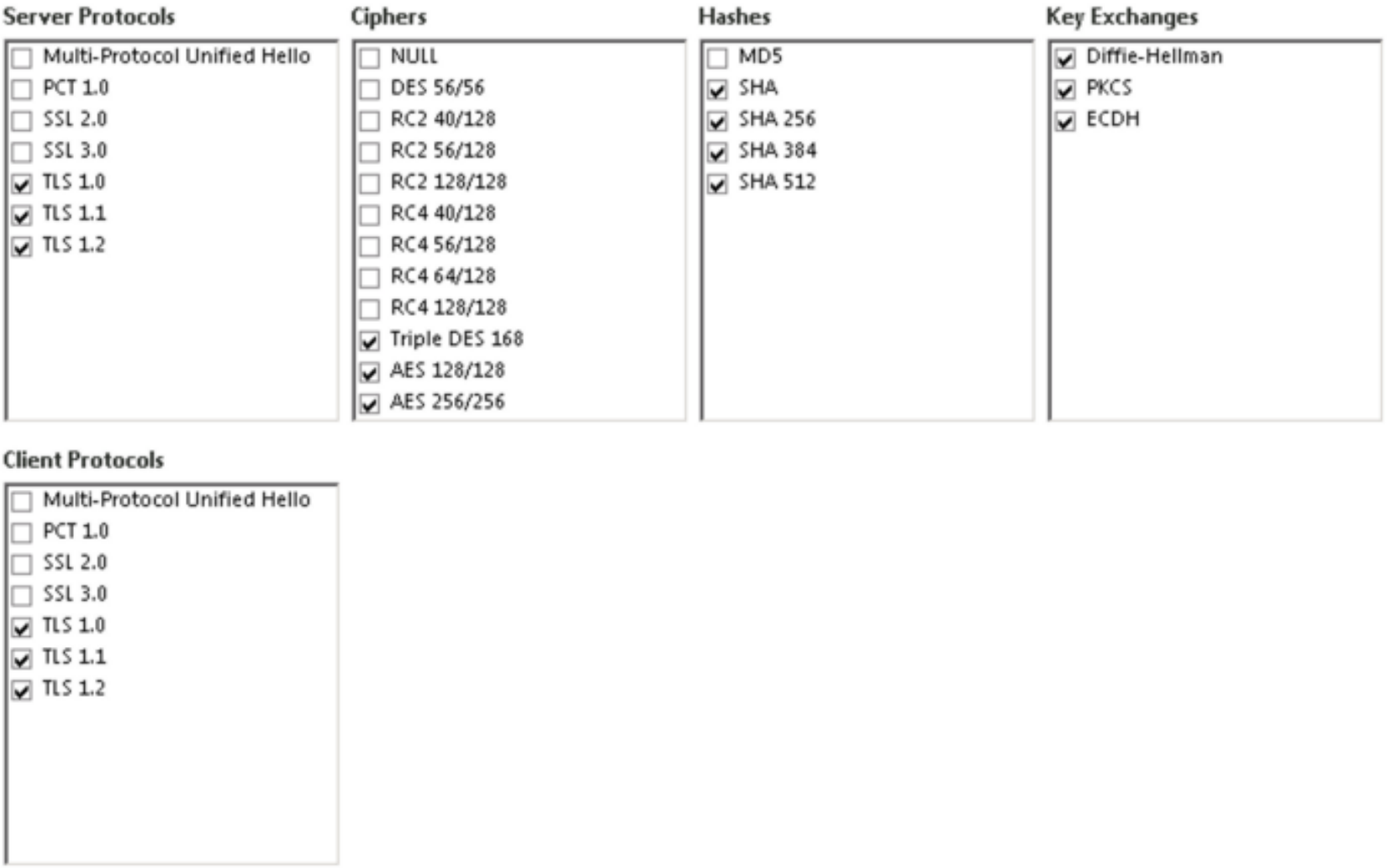
FIPS 140-(2 and 3) Security Level 2

- Physical tamper evidence, role based authentication
- Industry standard
  - Tamper evident seals or locks
  - Role-based or identity-based authentication
  - Stringent cryptography algorithms
  - Allows cryptography in multi-user timeshared systems
  - Examples: network appliances, secure data storage devices

Example: Mozilla Firefox

- Firefox “Network Security Services” (NSS)
  - Set of libraries that support security-enabled client/server services
    - E.g., SSL, TLS, X.509v3
- Has been evaluated at FIPS140-2 Level 1 and 2 (2009)
  - Level 1: Windows XP SP2, Mac OSX 10.5
  - Level 2: Red Hat Enterprise Linux 5, Solaris 10
- In FIPS Mode
  - strong master password
  - only use TLS (not SSL 2 or SSL 3.0)
  - only use FIPS encryption algorithms such as AES or triple-DES (not RC4, etc.)
- Must configure Firefox for “FIPS” mode

Example Settings for a Web Server



FIPS 140- (2 and 3) Security Level 3

- Enhanced physical security, user based authentication
- Security policy modeling
- Less than 7% of all certificates
  - Attempts to prevent intruders from gaining access to critical security parameters
  - Identity-based authentication
  - Additional functional requirement of a Trusted Path
  - Requires a physical or logical separation between the interfaces by which critical security parameters enter and leave the module
  - Benchmark for finance and high-risk areas
- Examples:
  - IBM Cloud Hardware Security Module 7.0
  - D200 Mlc Memory Device

FIPS 140- (2 and 3) Security Level 4

- Envelope and environmental protection
- Less than 1% of all certificates
  - Must detect & respond to all unauthorized attempts at physical access
  - Requires circuitry that zeroizes all plain text critical security parameters when the removable covers/door are opened
  - Environmental protection (temperature/voltage)
  - Used in defense-related areas
- Example:
  - IBM4767 Cryptographic Coprocessor

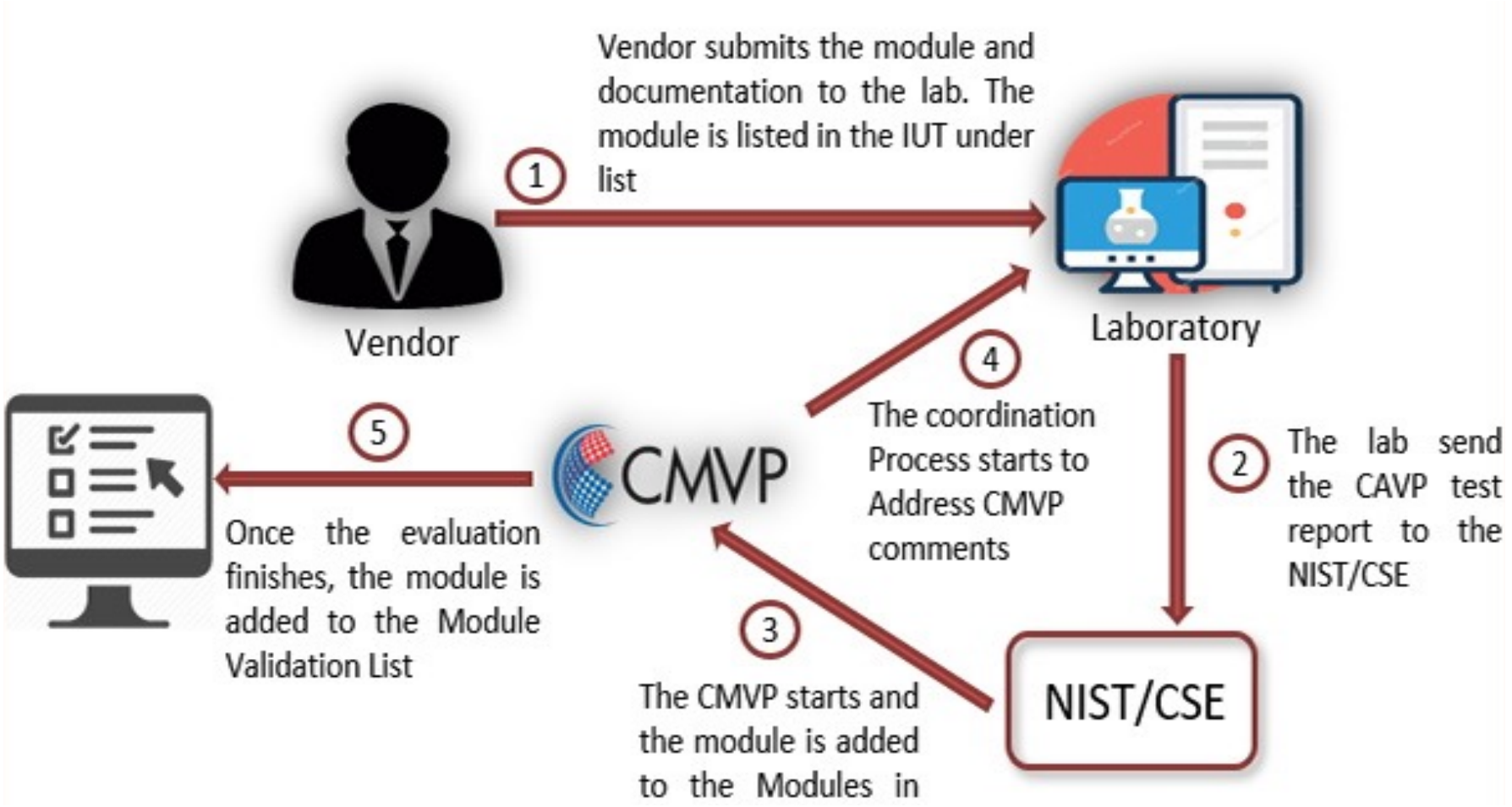
FIPS-140-2 Levels Summary

	Security Level 1	*Security Level 2*	*Security Level 3*	*Security Level 4*
Cryptographic Module Specification	Specification of cryptographic module, cryptographic boundary, Approved algorithms, and Approved modes of operation. Description of cryptographic module, including all hardware, software, and firmware components. Statement of module security policy.			
Cryptographic Module Ports and Interfaces	Required and optional interfaces. Specification of all interfaces and of all input and output data paths.		Data ports for unprotected critical security parameters logically or physically separated from other data ports.	
Roles, Services, and Authentication	Logical separation of required and optional roles and services.	Role-based or identity-based operator authentication.	Identity-based operator authentication.	
Finite State Model	Specification of finite state model. Required states and optional states. State transition diagram and specification of state transitions.			
Physical Security	Production grade equipment.	Locks or tamper evidence.	Tamper detection and response for covers and doors.	Tamper detection and response envelope. EFP or EFT.
Operational Environment	Single operator. Executable code. Approved integrity technique.	Referenced PPs evaluated at EAL2 with specified discretionary access control mechanisms and auditing.	Referenced PPs plus trusted path evaluated at EAL3 plus security policy modeling.	Referenced PPs plus trusted path evaluated at EAL4.
Cryptographic Key Management	Key management mechanisms: random number and key generation, key establishment, key distribution, key entry/output, key storage, and key zeroization.			
	Secret and private keys established using manual methods may be entered or output in plaintext form.		Secret and private keys established using manual methods shall be entered or output encrypted or with split knowledge procedures.	
EMI/EMC	47 CFR FCC Part 15. Subpart B, Class A (Business use). Applicable FCC requirements (for radio).		47 CFR FCC Part 15. Subpart B, Class B (Home use).	
Self-Tests	Power-up tests: cryptographic algorithm tests, software/firmware integrity tests, critical functions tests. Conditional tests.			
Design Assurance	Configuration management (CM). Secure installation and generation. Design and policy correspondence. Guidance documents.	CM system. Secure distribution. Functional specification.	High-level language implementation.	Formal model. Detailed explanations (informal proofs). Preconditions and postconditions.
Mitigation of Other Attacks	Specification of mitigation of attacks for which no testable requirements are currently available.			

FIPS 140-2 vs. FIPS 140-3

- **Stricter integrity test requirements**:
  - Level 2 modules must provide software/firmware integrity testing using digital signatures or HMAC (hash-based message authentication code)
  - Level 3 and Level 4 modules must provide integrity using digital signatures only
- **New required service** -- to output the module name/identifier and version that can be mapped to validation records/certificates
- **Key zeroization is required** -- for ALL unprotected "Sensitive Security Parameters" (SSP) at all levels, including public keys:
  - Level 2+ require a status indicator when the zeroization process is completed
- **Roles, services and authentication** – must be met by a cryptographic module's implementation
- **Non-invasive security** – is required for hardware and firmware components of a module, optional for software modules operating in a modifiable operating environment, and the module must protect against a list of non-invasive attacks
- **Lifecycle assurance** -- vendors need to perform their own testing on a module, in additional to the validation lab testing
- **Operational environment** -- software modules no longer need to operate in a Common Criteria (CC) evaluated OS or 'trusted operating system' in order to meet Level 2 requirements, however, these Level 2 modifiable operational environments require an audit mechanism

FIPS 140-3 Validation Process



CMVP - Cryptographic Module Validation Program
IUT - Implementation Under Test

Outline

- Student presentations
- Deployed network cascade problem
- Cryptography policy
- Course review

Review for Final Exam

- Emphasis (50%) on things covered since the mid-term
- Topics:
  - Challenge of security policy breaches
  - Characteristics of policy
  - Reference monitor and security policy models
  - U.S. classified information policy
  - Bell-LaPadula model and Multics interpretation
  - Theoretical limits on system security (no proofs)
  - Biba integrity problem and policy interpretation
  - Other integrity policies (Lipner, Clark-Wilson)
  - Hybrid Policies (Chinese Wall, RBAC, ORCON)
  - Policy composition (TCB subsets and TCB partitions)
  - MAID Components
  - System evaluation
  - Deployment policy
  - Cryptographic policy

Your Questions

- Is it possible to take an operating system like windows and use Intel protection rings to separate the applications, file systems and hardware or does the OS need to be built from scratch with the protection rings included in the design?
- What do we need to understand in the general undecidability of security?
- Can you please explain again why integrity policies are harder to implement than confidentiality policies?
- Is it possible to use Biba to implement a practical system that supports Clark-Wilson policy?
- How do I participate in research to extend or re-develop/engineer missing or lost security concepts?

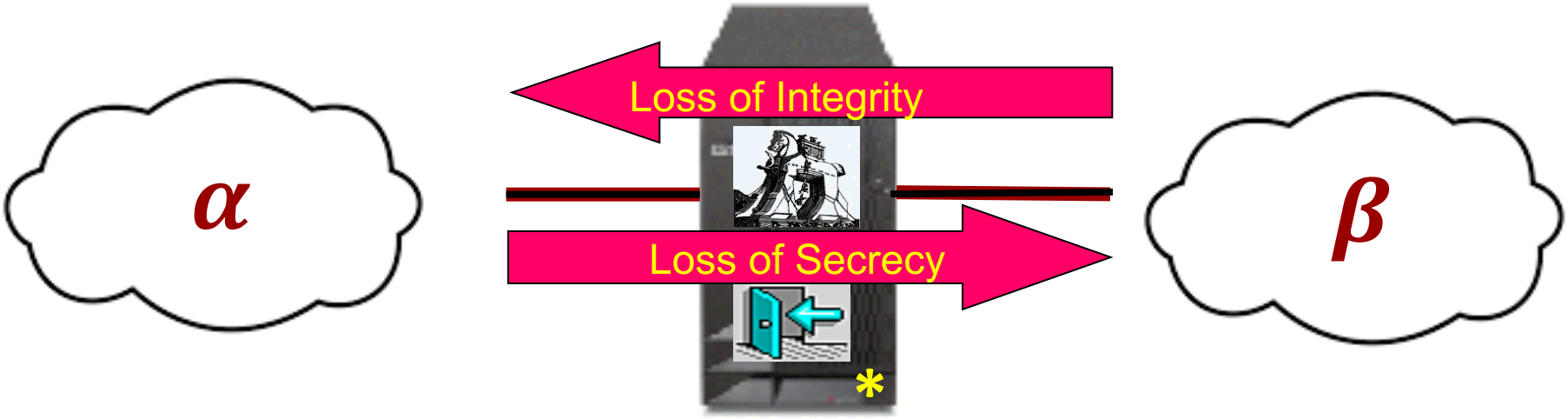
"The most effective approach to evaluating the security of complex systems is to deliberately construct the systems using security patterns specifically designed to make them evaluable"
Using Proven Reference Monitor Patterns for Security Evaluation, Mark R. Heckman, and Roger R. Schell, 2016.

Your Review Topics

- TCB subsets and partitions
- Can you talk more about similarities and differences between TCSEC and CC?
- Difference between PP and ST
- Review RBAC/ABAC and Chinese wall policy
- BLP and Multics

CDS Subversion Vulnerability

- Connection of disparate domains is multilevel (requires MLS)



Corporate or Government
High Networks Domain

Must be MLS!

Low Networks or
Internet Domain

We need secure (trusted) MLS components!

Student's Question

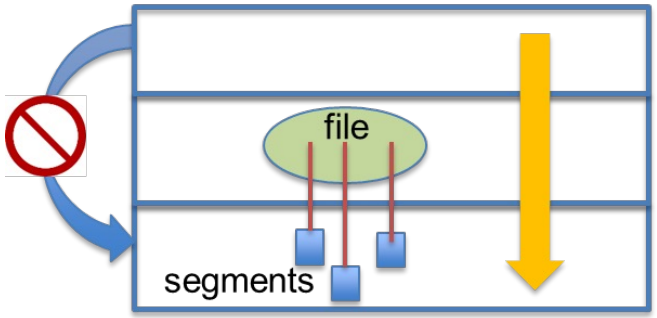
- What access (read/write/both) will the subject have in the following cases (DAC has granted all access):
  - Subject's access label (Secret, {A})
  - Object's access label (Top Secret, {B})
  - Subject: (Secret, {A})
  - Object: (Top Secret, {A})
  - Subject: (Secret, {A})
  - Object: (Top Secret, {A,B})
  - Subject: (Secret, {A,B})
  - Object: (Secret, {A})

Student's Question

- Can we please review the first two questions from quiz 4.
  1. Explain the "composition problem". What are the two main approaches that Shockley discusses (hint: one for single computer; one for loosely coupled network)?
  2. Why do we require confinement of subjects and objects to a single network component in one of the approached your listed in question 1?

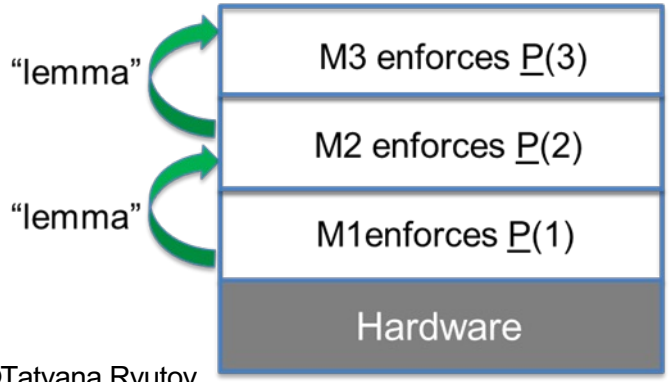
Policy Composition with TCB Subsets

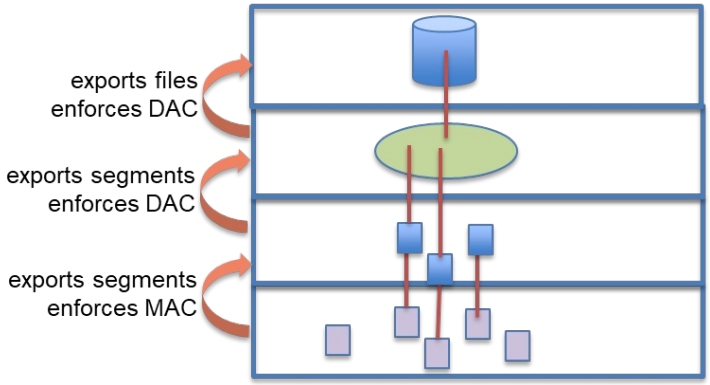
- Only two validated composition methods
  - Known as (1) Partitioned TCB and (2) TCB Subset
- The TCB subsets concept divides the overall policy into security **policy subsets**, which are then each allocated to a TCB subset for enforcement
- These TCB subsets are hierarchal in nature and follow the concept of incremental evaluation
- We defined TCB subsets – for a single computer
  - Leverage hierarchical domains ordered by privilege
  - Each TCB subset resides in an individual domain
  - Every access request submitted to every subset
  - Most privileged domain (e.g., ring) is security kernel
  - Decomposing a policy into subsets is “art” not science
  - System policy is the union of all TCB Subset policies



Question (from HW3)

- How does the concept of ordered domains help in the "incremental evaluation" of a system composed of TCB subsets?
  - TCB subsets are arranged in a hierarchal fashion such that one domain can only use services of the domains below it and provide services to ones above it
  - For any TCB subset to be evaluated, all the subsets in domains more privileged than it, i.e. arranged below it in the order, must have been successfully and positively evaluated
  - This is called **incremental evaluation** and the concept of ordered domains is crucial to its proper working



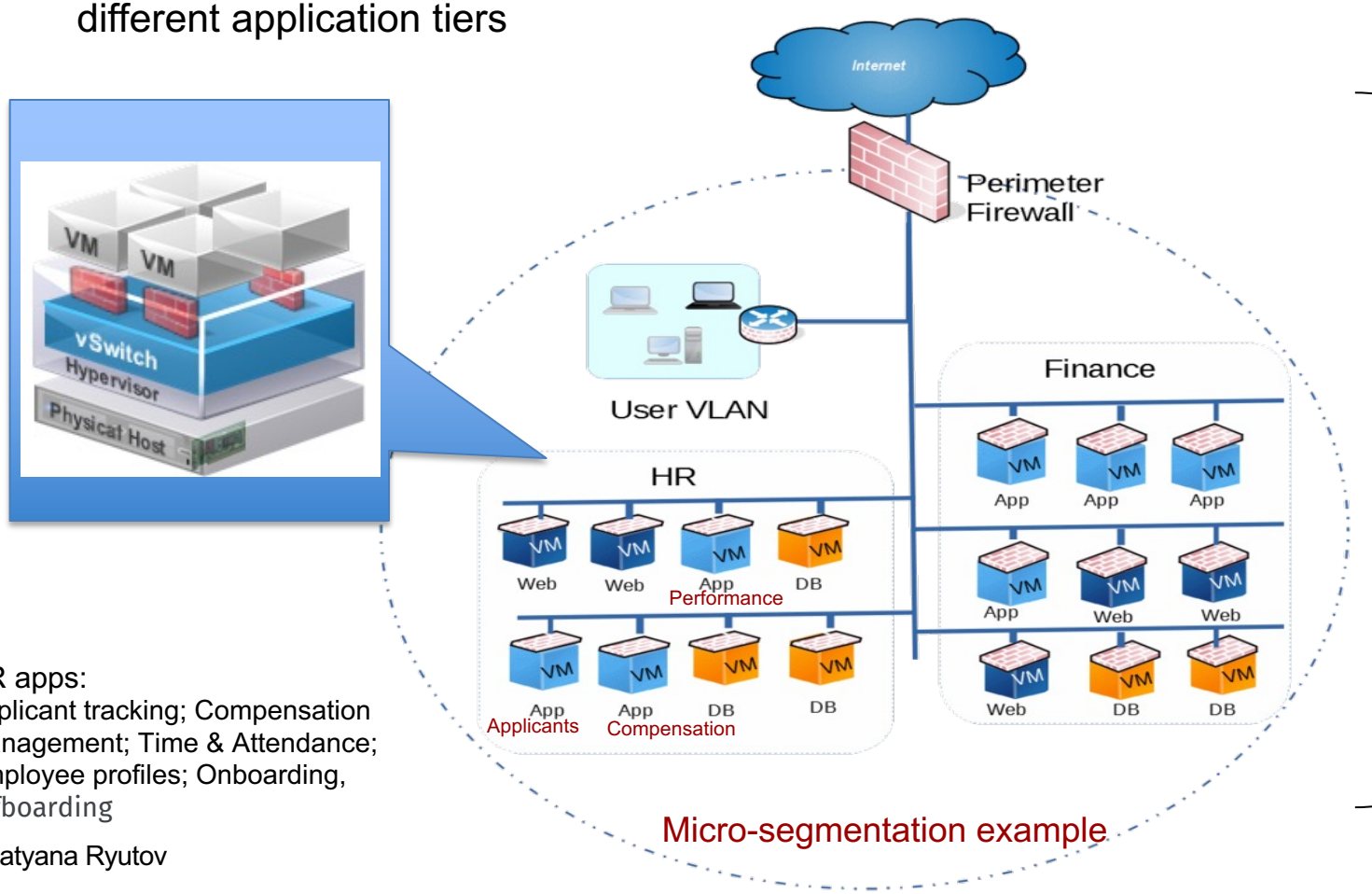


Question (from HW3)

- How can the “Reference Monitor” be applied as part of the incremental evaluation?
  - The TCB subsets have one domain i.e., ring 0 that do not depend on any other domain to be evaluated
  - It should be small enough to be evaluated **independently,** should be tamperproof and completely mediated
  - This subset is distinguished as **the only** subset to directly access the hardware platform, and is therefore the reference monitor and is the most privileged TCB subset

Example: TCB Subsets a Zero Trust Architecture

- "Microsegmentation" is a method of creating secure zones within the network at the applications and services layer
- Typically a host-based security segmentation
- Network segment is divided into "micro-segments" that enforce security policies for different application tiers

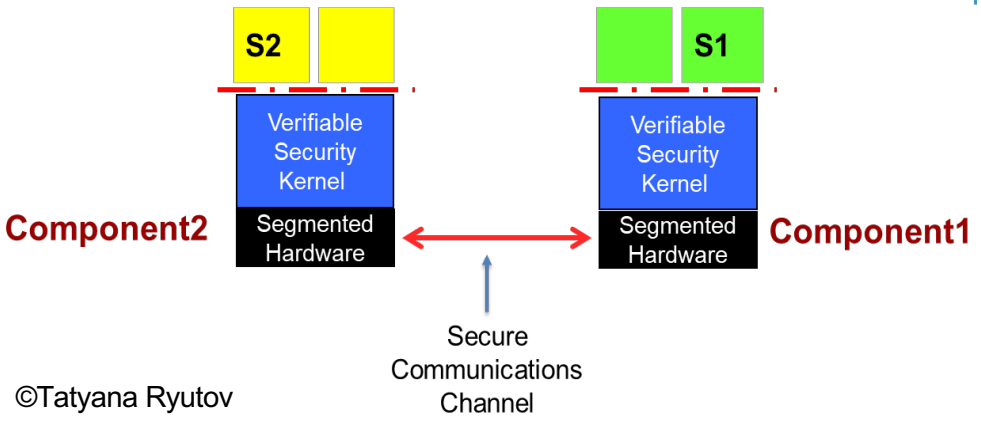


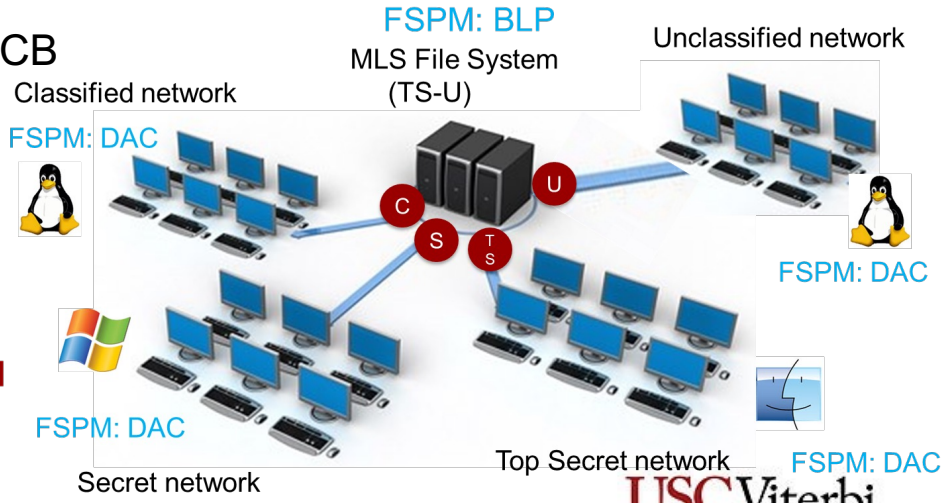
HR apps:
Applicant tracking; Compensation Management; Time & Attendance; Employee profiles; Onboarding, Offboarding

USC Viterbi School of Engineering

Partitioned TCB for Policy Composition

- The partitioned TCB concept divides the system policy into **disjoint** security policy partitions allocated to **distinct** components of a network
- NTCB is cooperating, loosely-coupled partitions
  - Ideal communications channels connect devices
- All functions of the NTCB must be allocated
  - In some coherent way to the various components
- Network components & channels are exhaustive
  - Parts are disjoint (none shared between components)
- Access mediated by component security kernel
  - Totality of security kernels mediates all accesses
- Partitioning may be applied recursively
  - A single TCB partition may itself be a NTCB





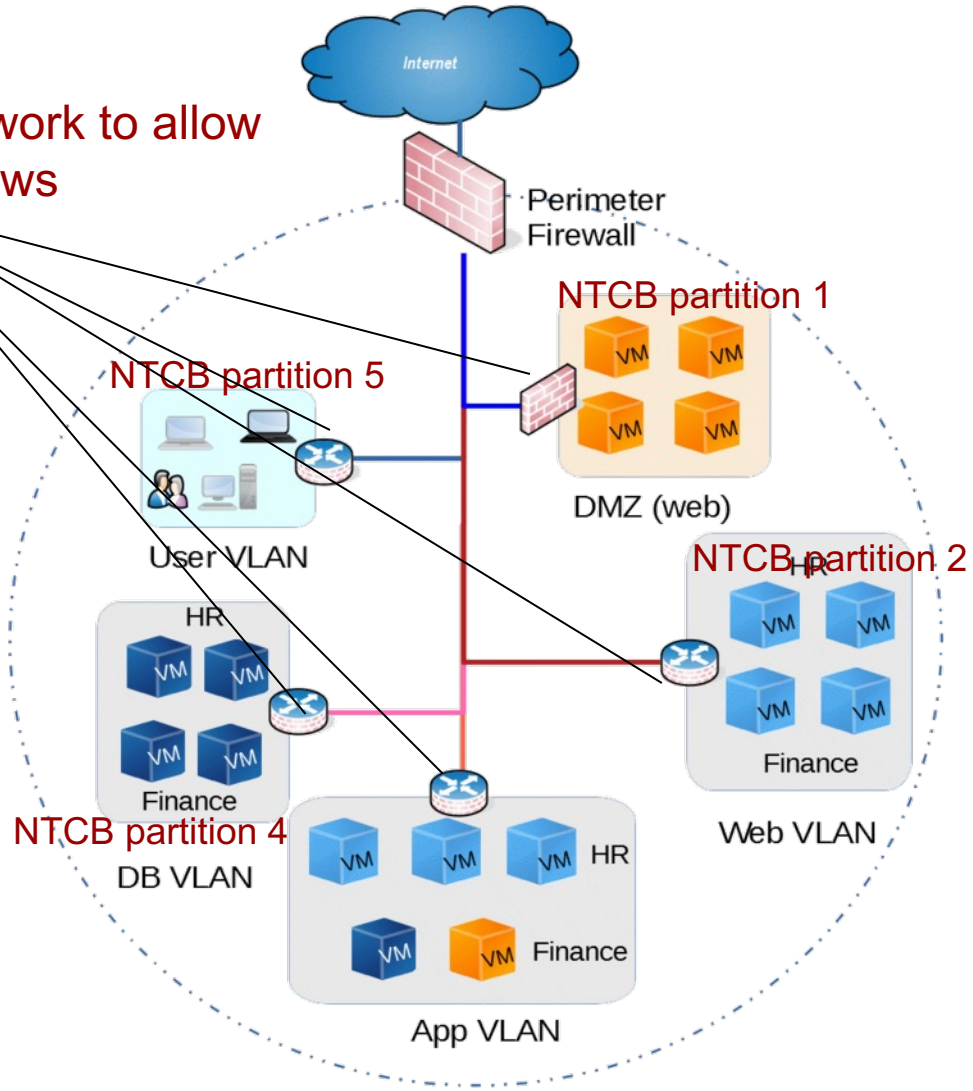
USC Viterbi School of Engineering

Student's Question

- How else are TCB subsets and TCB domains different other than that subsets are for single computers while domains are for loosely coupled systems?

Example: TCB Partitions for Zero Trust Architecture

Architect your network to allow only authorized flows

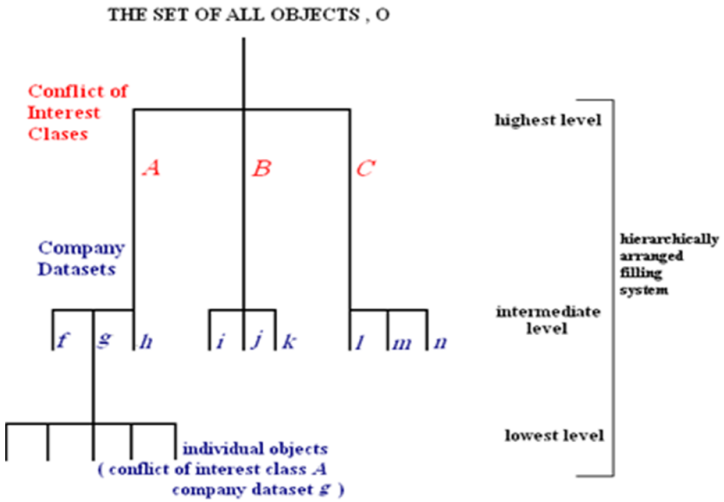


NTCB policy

USC Viterbi School of Engineering

Chinese Wall Policy

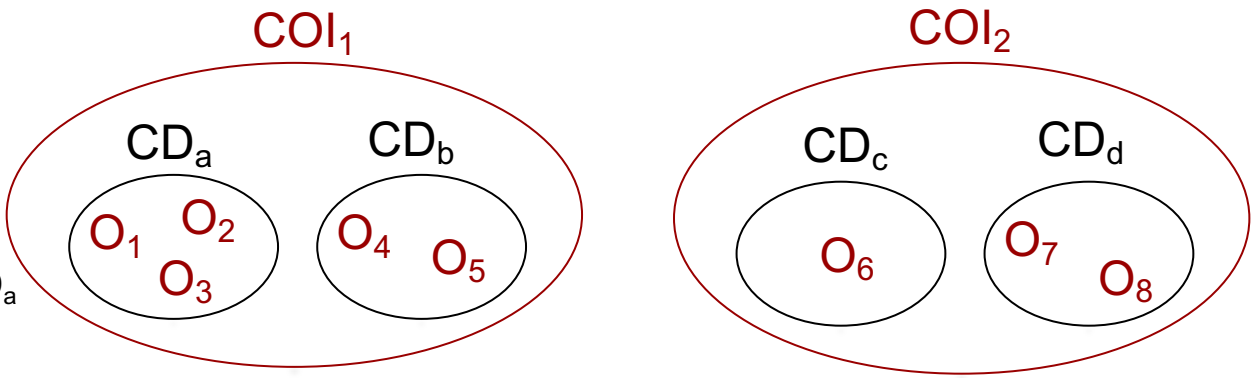
- Chinese Wall policy focuses on conflict of interest
  - Information is stored in hierarchically arranged levels
  - Subjects are only allowed access to information which is not in conflict with any other information that they already possess
  - CW is based on access history
- ChW Model rules:
  - Simple Security Rule: subject *s* can **read** object *o* only if:
    1. object *o* is in the same company datasets as all objects already accessed by *s*, **OR**
    2. object *o* belongs to an entirely different conflict of interest class
  - *-property: subject *s* can **write** object *o* only if:
    1. subject *s* can read *o* by simple security rule, **AND**
    2. no object *o'* can be read which is in a **different** company dataset than the one for which write access is requested



Student's Question

- In Chinese wall model, if there are 2 COIs; one COI has CDa and CDb, and the other COI has CDc, and CDd.
- Alice has read and write access to CDa, so will she have read access to CDc/CDd since it's in another COI and it satisfies simple security property?

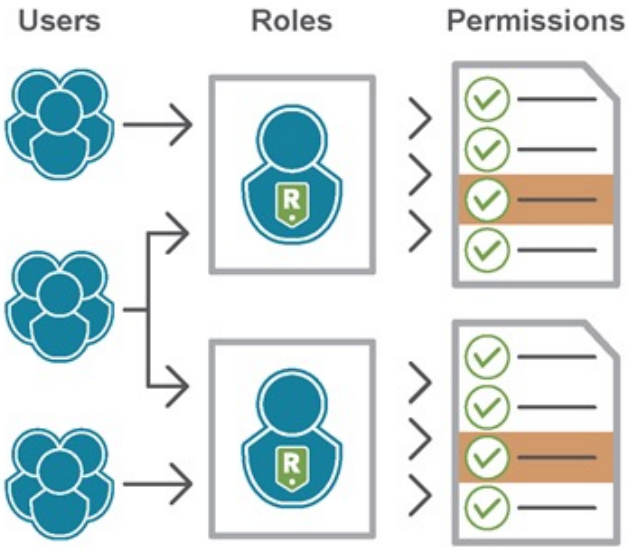
Alice:
read/write CD_a
read CD_c



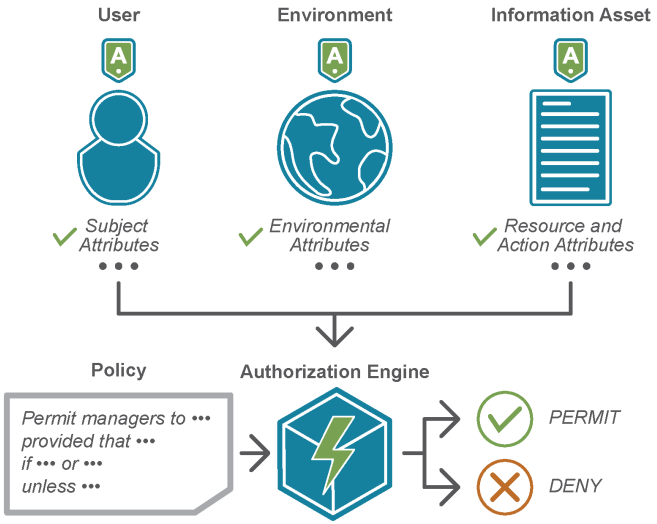
- Simple Security Rule: subject *s* can **read** object *o* only if:
  1. object *o* is in the same company datasets as all objects already accessed by *s* **OR**
  2. object *o* belongs to an entirely different conflict of interest class
- *-property: subject *s* can **write** object *o* only if:
  1. subject *s* can read *o* by simple security rule, **AND**
  2. no object *o'* can be read which is in a **different** company dataset than the one for which write access is requested

RBAC and ABAC

- RBAC
  - Uses roles to simplify administration of access control
  - Family of models (add role hierarchy, inheritance, role activation constraints)
- ABAC
  - Uses attributes as building blocks in a structured language that defines access control rules and describes access requests
  - Rules can be extremely fine-grained and contextual



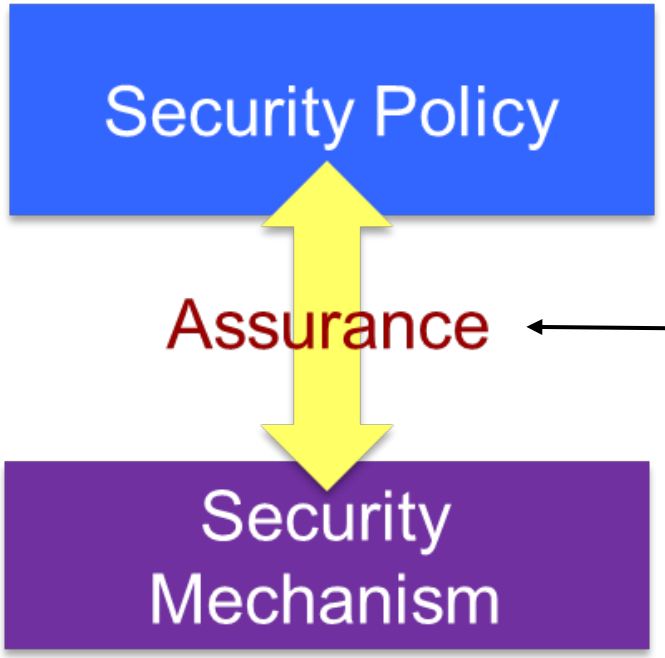
RBAC



ABAC

Three “Legs” of Security

- Policy – definition of security for the system
- Mechanism – technical, administrative, and physical controls
- Assurance – evidence that mechanisms enforce policy

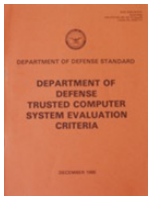


Access control policies (MD)
Supporting policies (IA)

System evaluation process

Interpretation of reference monitor; must be tamperproof, non-bypassable, verifiable

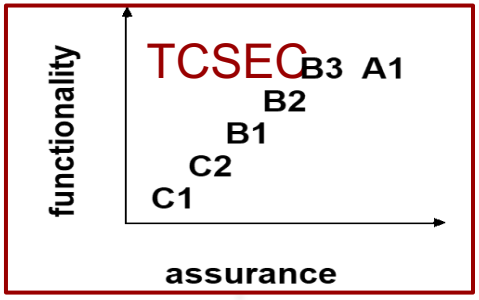
TCSEC



- Known as Orange Book, DoD 5200.28-STD
- First widely used evaluation criteria, withdrawn in 1999
- Repeatable analysis of evidence
- Seven distinct “evaluation classes”
  - Progressive increments in confidence
- Four trust rating divisions (classes)
  - D: Minimal protection
  - C (C1,C2): Discretionary protection
  - B (B1, B2, B3): Mandatory protection
  - A (A1): Highly-secure

SECURITY POLICY	C1	C2	B1	B2	B3	A1
Discretionary Access Control						
Object Reuse						
Labels						
Label Integrity						
Exportation of labelled information						
Exportation to Multi-Level Devices						
Exportation to Single-Level Devices						
Labelling Human-Readable Output						
Mandatory Access Control						
Subject Sensitivity Labels						
Device Labels						

ASSURANCE	C1	C2	B1	B2	B3	A1
System Architecture						
System Integrity						
Security Testing						
Design Specification and Verification						
Covert Channel Analysis						
Trusted Facility Management						
Configuration Management						
Trusted Recovery						
Trusted Distribution						



USC Viterbi School of Engineering

Trusted System Analysis Factors

- TCSEC/TNI security kernel evaluation factors
- Issue of completeness

1. Object Reuse [4.1.1.2]
2. Labels [4.1.1.3]
3. Label Integrity [l 4.1.1.3.1]
4. Exportation of Labeled Information [4.1.1.3.2]
5. Exportation to Multilevel Devices [4.1.1.3.2.1]
6. Exportation to Single-Level Devices [4.1.1.3.2.2]
7. Labeling Human-Readable Output [4.1.1.3.2.3]
8. Subject Sensitivity Labels [4.1.1.3.3]
9. Device Labels [4.1.1.3.4]
10. Mandatory Access Control [4.1.1.4]
11. Trusted Path [4.1.2.1.1]
12. System Architecture [4.1.3.1.1]

7. System Integrity [4.1.3.1.2]
8. Covert Channel Analysis [4.1.3.1.3]
15. Trusted Facility Management [4.1.3.1.4]
16. Trusted Recovery [4.1.3.1.5]
17. Security Testing [4.1.3.2.1]
18. Design Specification and Verification [4.1.3.2.2]
19. Configuration Management [4.1.3.2.3]
20. Trusted Distribution [4.1.3.2.4]
21. Security Features User's Guide [4.1.4.1]
22. Trusted Facility Manual [4.1.4.2]
23. Test Documentation [4.1.4.3]
24. Design Documentation [4.1.4.4]
25. Ratings Maintenance Phase (RAMP)

Common Criteria (CC)

- An international standard (ISO/IEC 15408)
- Framework (not actually a criteria) in which users can specify their security functional and assurance requirements
- Defines security targets and protection profiles
  - Protection profile (PP) is an implementation-independent set of security requirements for a category of products
  - Security Target is implementation-specific – a basis against which evaluation is performed; may match a protection profile or a set of PPs
- Separate functionality and assurance classes
  - Predefines "packages" of features & assurance
- Seven Evaluation Assurance Levels (EAL):
  - EAL1 Functionally tested
  - EAL2 Structurally tested
  - EAL3 Methodically tested and checked
  - EAL4 Methodically designed, tested & reviewed
  - EAL5 Semi-formally designed and tested
  - EAL6 Semi-formally verified design and tested
  - EAL7 Formally verified design and tested

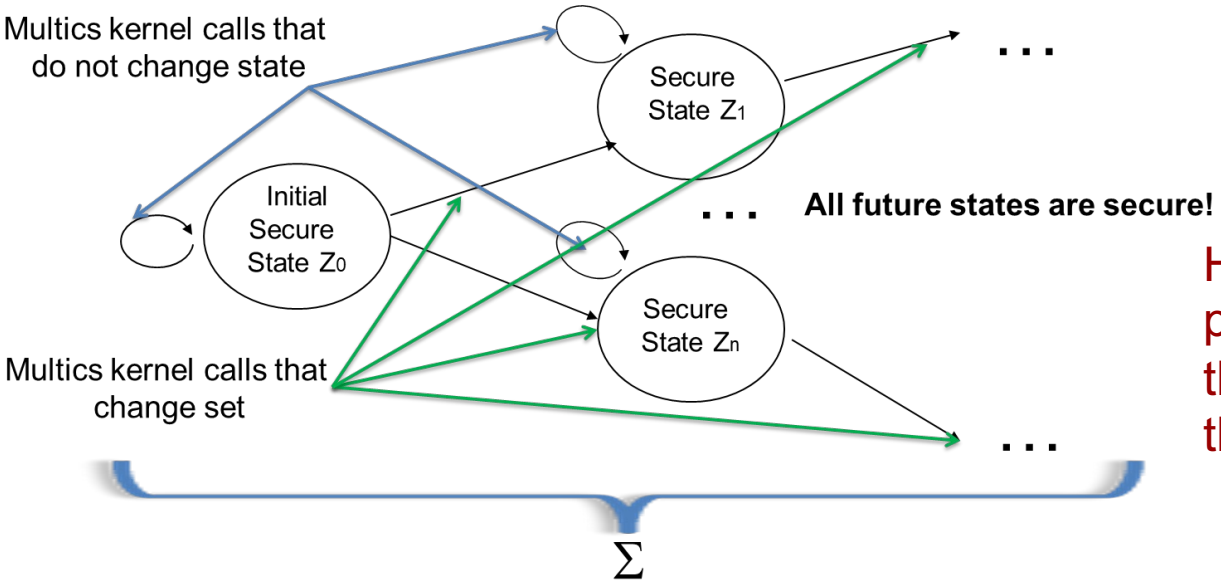
How is TCSEC fundamentally different from CC?

Bell-LaPadula Interpretation for RM

- BLP goal is to enforce confidentiality policy
- Define *Simple Security Condition (ssc)*:
  - S can read O IFF S *dom* O
- Define **-Property*:
  - S can write O IFF O *dom* S
  - Particularization for computer system, not for people
- A state of the system is (b, M, f, H) where:
  - b shows which subjects have access to which objects
  - M is the access control matrix for the current state
  - f is tuple indicating subject and object access classes
  - H is the hierarchy of objects (for naming an object)

Multics Interpretation of BLP System Σ

- In formal terms, system $\Sigma(R, D, W, z_0)$ where:
  - R denotes the set of requests for access
  - D denotes the set of outcomes
  - W is the set of actions of the system
  - z_0 is the initial state of the system
- Rules for transition from one state to another
  - Functions to change each element of state (b, M, f, H)
- Basic Security Theorem – Multics (Σ) is secure
  - If initial state secure, all states from rules are secure



How the 3 RM properties relate to the basic security theorem?

RM Concept as a Framework

- Using the RM concept as a framework could be very valuable for designing a system and identifying necessary security controls
- Steps:
  1. Create a suitable security policy
  2. Identify the identification and authentication controls
     - Important to be able to enforce the policy and accountability
  3. Identify the authorization controls (reference validation function of RM)
  4. Identify controls to enforce the completeness and isolation properties of the reference monitor
     - **Verifiability?**
  5. Design the audit log mechanism
- The RM concept encapsulates what it means to say a system is “secure”

Lessons Learned

- The computer security legacy needs to be appreciated, shared and used
  - Otherwise we risk losing (little) hard science and engineering wisdom we have
- Multilevel secure components are unavoidable
  - Only MAC policy can enforce secure information flow
  - Multilevel links between security levels must be as strong as we can make them, we need trusted (high assurance) components
- It is impossible to build “secure” products without a policy and reference monitor
  - You could have totally bug-free code but still not have a secure (trusted) system
  - Tight configuration and add-on gadgets are not a substitute for good security
  - Policy is the definition of security for a system, it holds everything together
- Security kernel is the only known verifiable protection technology