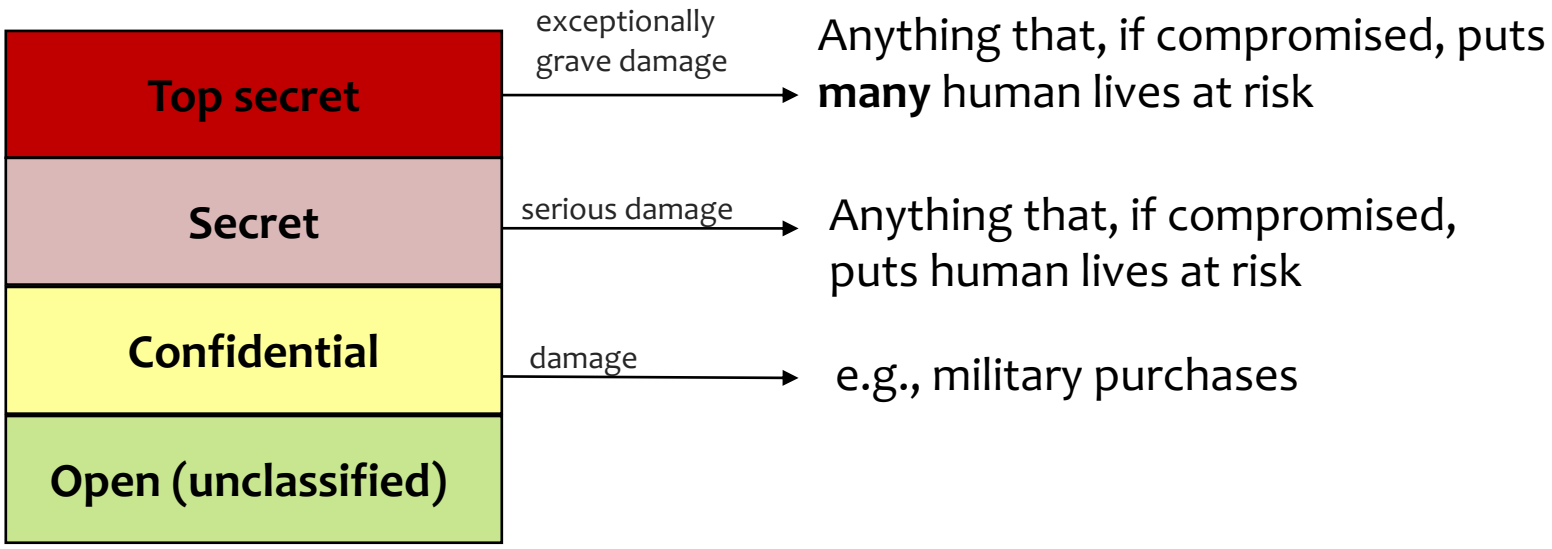
# Introduction to Information Security
# 14-741/18-631 Fall 2021
# Unit 3: Lecture 2:
# Multilevel and Multilateral Security

**Hanan Hibshi** hhibshi@cmu.edu

# This Lecture's Agenda

- **Outline**
  - Multilevel security (MLS)
  - Multilateral security
  - Inference control
- **Objective**
  - Expose you to information security access control problems outside of operating systems security
  - Make you aware that there is much more to access control than ACLs and capabilities
  - Give you examples of security policies

# Multilevel Security (MLS): Basic Problem



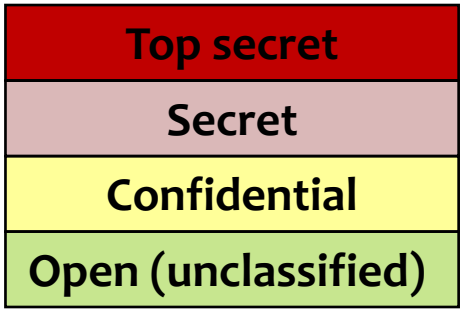

- **Clearance: Level of permission required to view classified information**
- **How do you ensure proper security at each level?**
  - What is a meaningful policy?
  - What are meaningful mechanisms?

# Importance of MLS

- **Used in military security**
- **Used in modern operating systems**
  - SELinux
    - Mandatory access control to shield processes
    - Labels are in the format user:role:type:level (level is optional)
  - Vista
    - Internet explorer runs at “Low” so as not to tamper integrity of “High” files (e.g. system files)

# The Bell-LaPadula Model

- **Consists of two simple properties**
  - Thought extremely useful formalism at the time
  - Enthusiasm has faded a bit since then (multilevel security is very hard) but still very useful model
- **The Simple security property**
  - No process may read data at a higher level
  - **No read up** (NRU)
- **The *-property (**strong star**)**
  - No process may write data at a lower level
  - **No write down** (NWD)



# Simple Security Property (No read-up)

Top secret clearance


Can read

Cannot read

No clearance

- Common sense: someone with no clearance shouldn't be able to read classified info!
- On the other hand, nothing prevents someone with a top-secret clearance to read secret or confidential materials

# *-property (No write-down)

Top secret clearance

Cannot write to publicly viewable files about classified info

Open

Can write about anything Homer knows

No clearance

- **Common sense again: prevent information leaks**
- **Much harder to implement in practice than it sounds**
  - E.g., when an attacker puts in a guessed password, the server always tells the attacker whether the password is correct or not (1 bit of information)

# Improvements on BLP

- **How do you deal with changes?**
  - Declassification: when data is no longer classified
    - Once declassified to public domain, information cannot be erased
  - Change in clearance level: when an entity gains or loses clearance

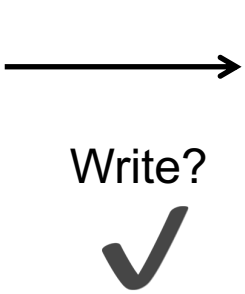
# Security Properties of BLP

- **Noninterference: information at a higher level is invisible at a lower level**
  - E.g., Homer doesn't know any upcoming military operations
- **(Probabilistic) Nondeducibility: someone at a low level cannot deduct with 100% probability what happens at a high level**
  - E.g., Homer cannot deduce any information about military operations
    - by activities of a base
    - a military family in the neighborhood
  - Used in anonymous systems
  - What about 99% probability? 90%?

# The Biba Model

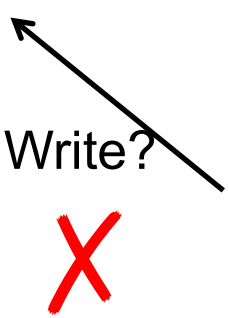
- BLP was good for confidentiality, avoided integrity
- Biba model deals with integrity only

Trusted



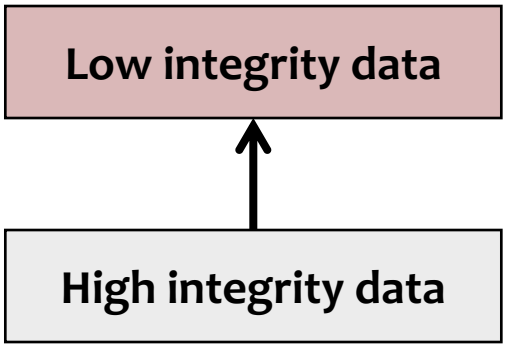

Highly sensitive data

Which targets to attack




Not trustworthy

# The Biba Model



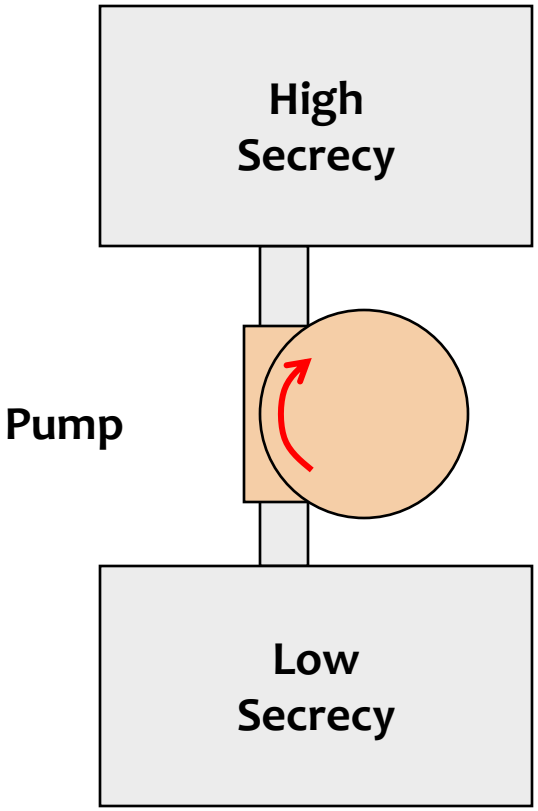

- **"BLP upside down"**
  - Integrity is the dual of confidentiality
  - Only read up
    - E.g., a user process can read calibration data
  - Only write down
    - Calibration process can write user data, but not the other way around
  - No read-down or write-up to prevent contamination of "High" objects with "Low" objects
    - High = trusted/high integrity
    - Low = untrusted/low integrity
- **Used in Linux for protecting against malicious code**
  - LOMAC extensions
  - System files are "High", Network is "Low"
  - As soon as a prog receives network traffic, prog downgraded to Low
  - This is precisely what Windows Vista implements

# Biba Examples

- **Windows Vista**
  - By default, everything is Medium
  - Admin files are High
  - Internet Explorer + downloaded files are Low
  - System files are System
  - Biba: Only read-up, only write down
    - Vista is *read-down*, write-down
    - Possible issue?
- **SE Linux**
- **Embedded Systems**
  - Meters are observed but are not influenced by billing systems
  - Systems that can observe everything but not alter anything
  - Dispatching systems can read-up

# The Naval Research Lab (NRL) Pump

- **BLP or Biba model require one-way communications**
- **Impractical in many applications**
  - E.g., server/password example
- **Instead, rate limit traffic that can go from high to low (if implementing a BLP policy) or low to high (if implementing Biba)**
  - Timing randomization of ACKs
- **With the cost of hardware going down, data diodes are more feasible**



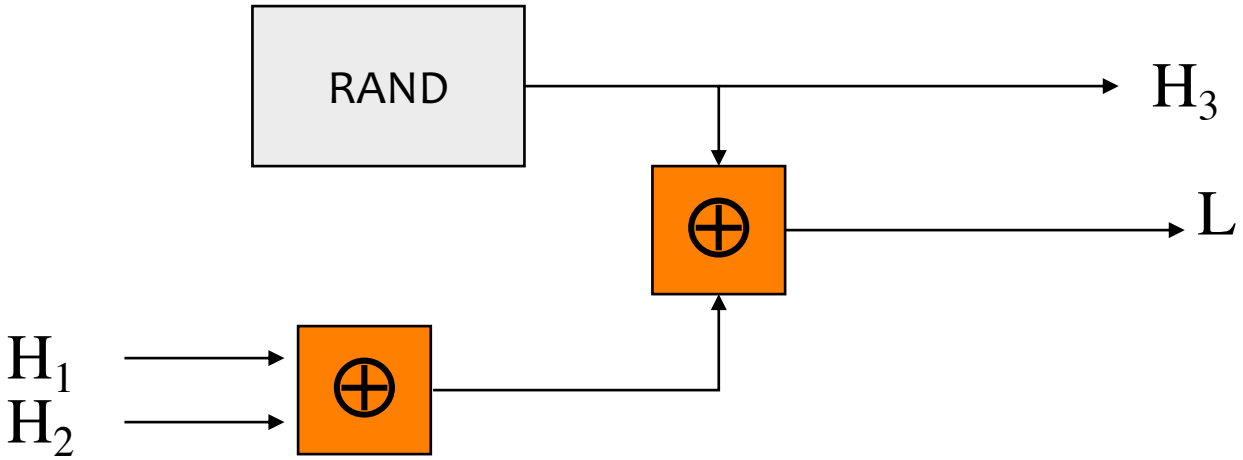

(from Anderson)

# Theoretical Difficulties in MLS

- Composability
- The Cascade Problem
- Covert Channels
- The Threat from Malware
- Polyinstantiation

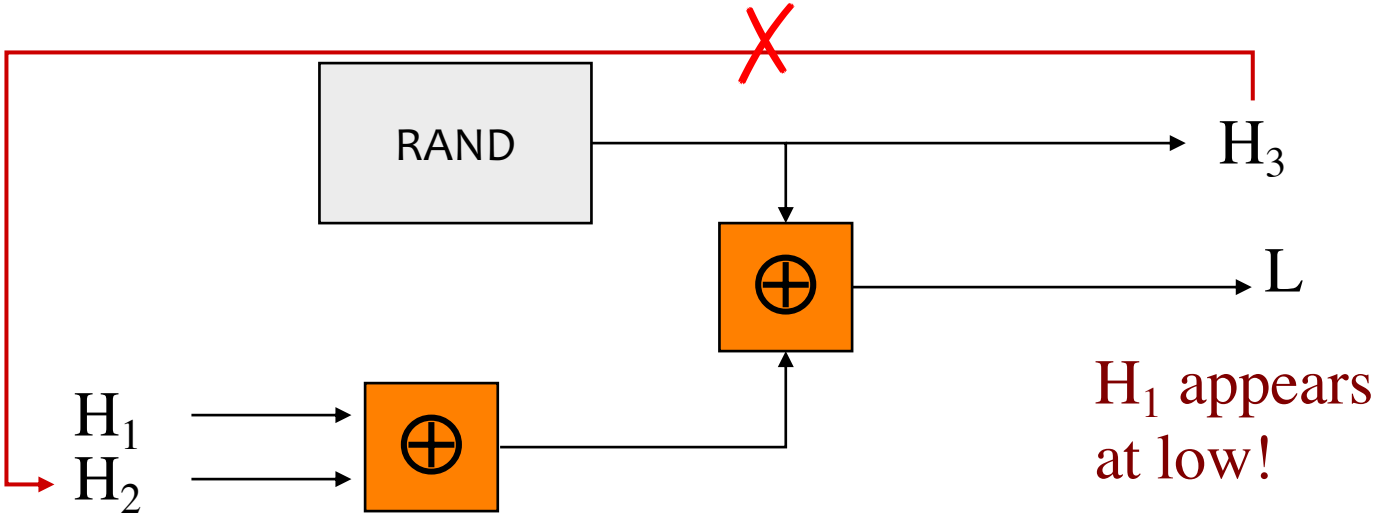
# Difficulties in MLS: Composability

- How do we compose two or more secure components into a secure system?
- Most problems arise when feedback is introduced



# Difficulties in MLS: Composability

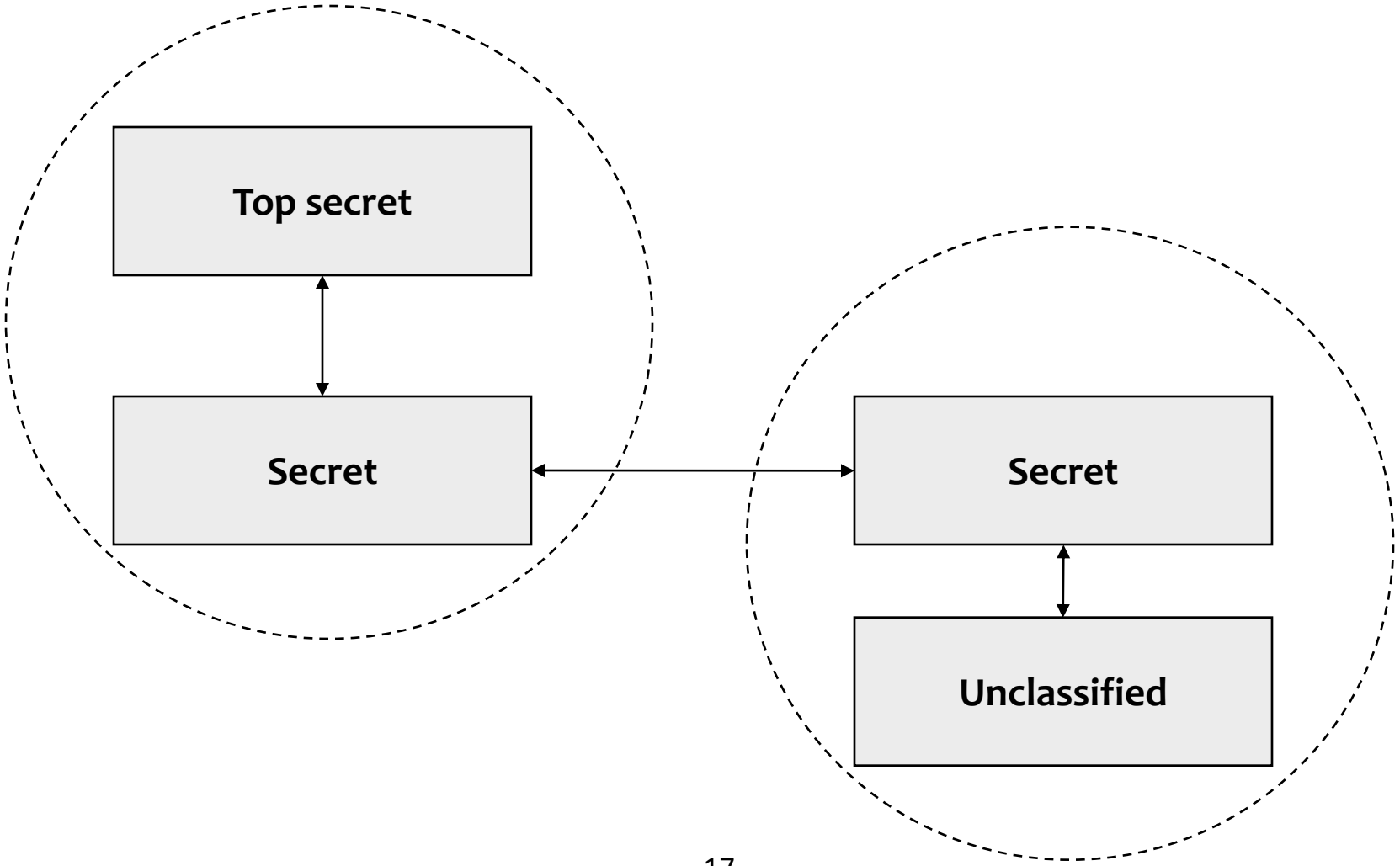
- How do we compose two or more secure components into a secure system?
- Most problems arise when feedback is introduced



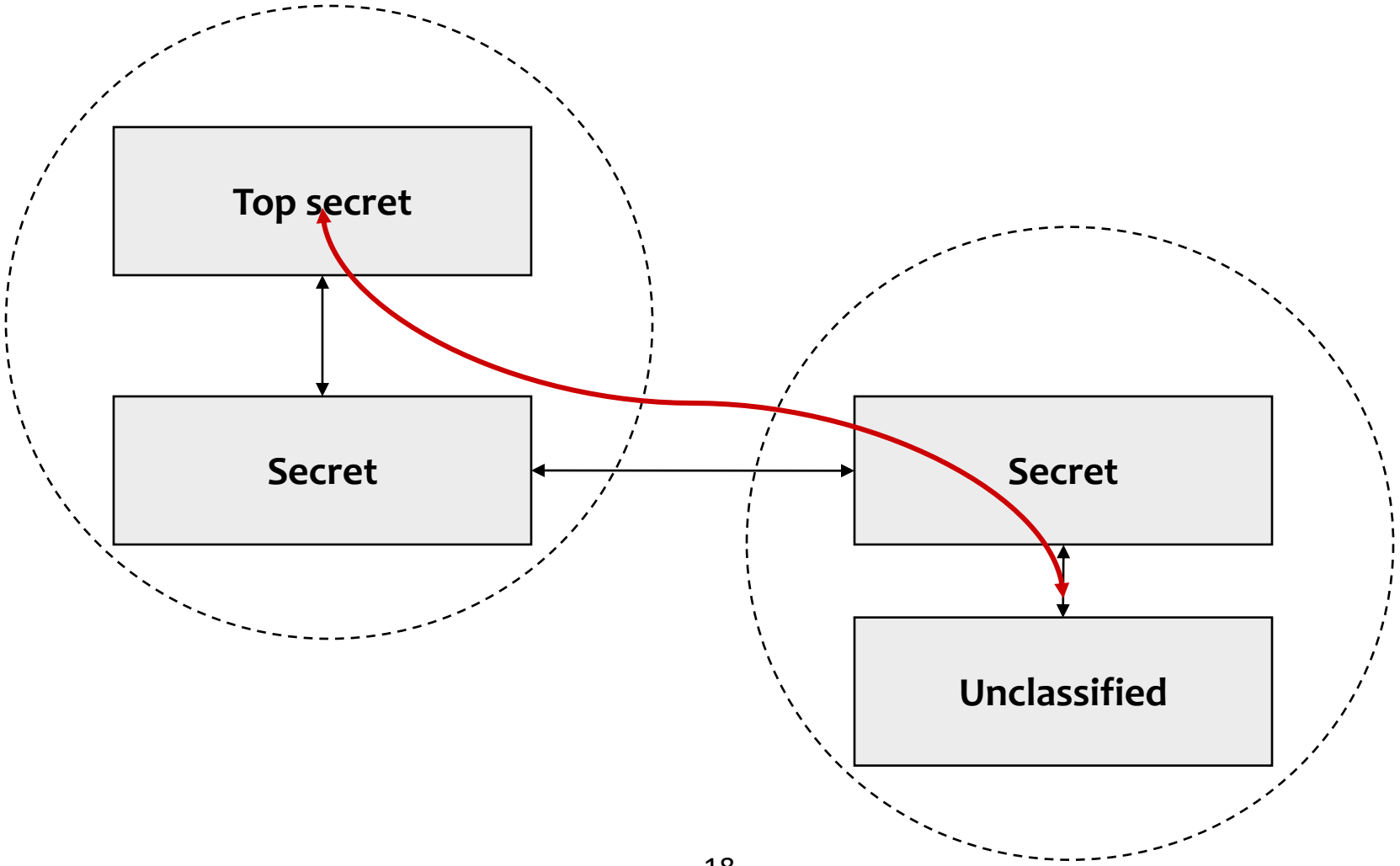
# Difficulties in MLS: Cascade Effects

- **Assume you have a system that can only manage securely interaction between two neighboring levels**
  - Don't want "Unclassified" and "Top secret" handled by the same machine
- **Interconnecting two such systems breaks the policy!**

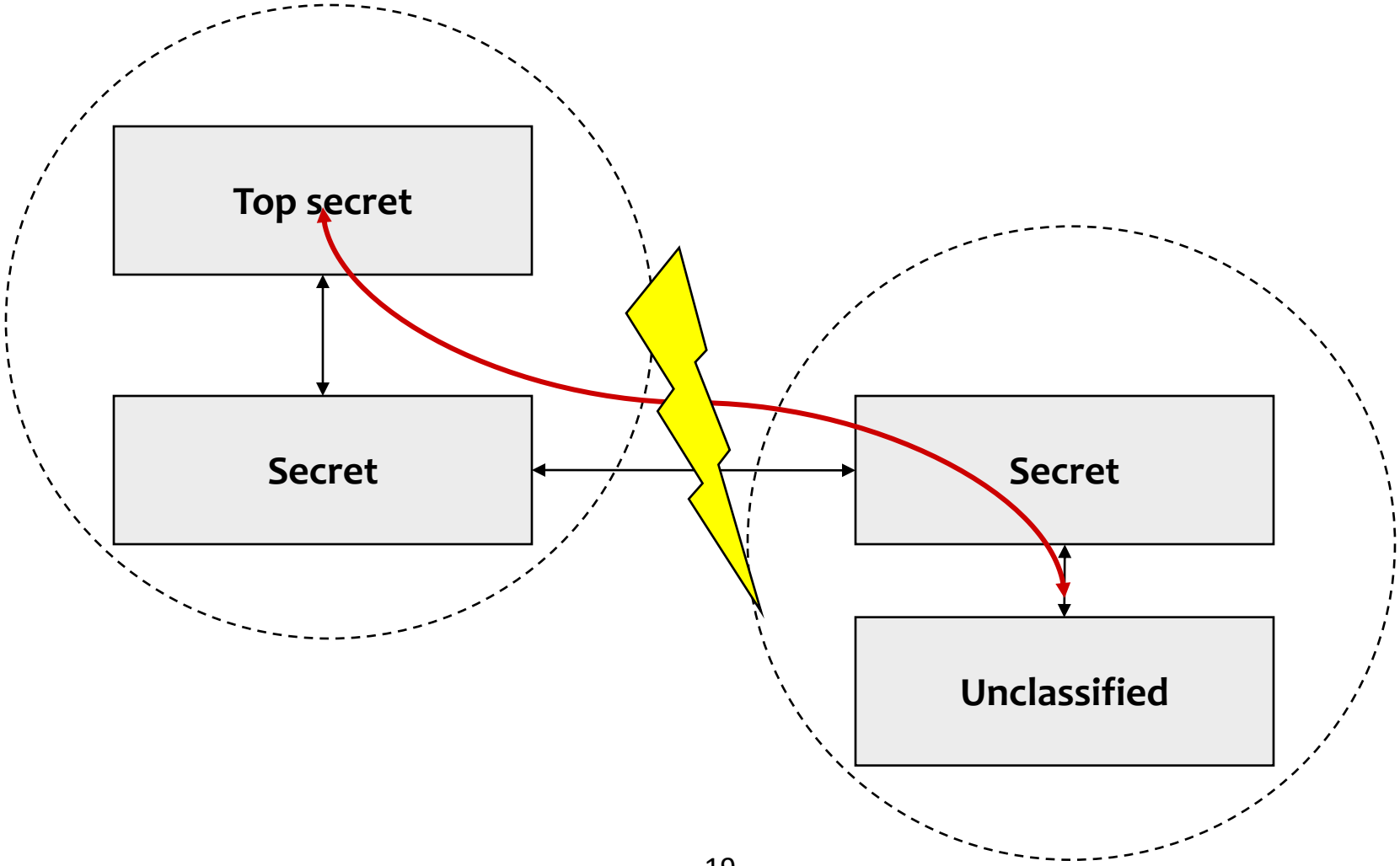
# Difficulties in MLS: Cascade Effects



# Difficulties in MLS: Cascade Effects



# Difficulties in MLS: Cascade Effects



# Difficulties in MLS: Covert Channels

- **MSL Secrecy Goal: prevent unauthorized communication among processes**
- **Protection models we discussed so far are not sufficient, processes can leak information through covert channels**
- **Examples:**
  - System status checks : `ps`, `time`, `netstat`, ...
  - File system
    - Write (temporary) files, directories
    - Test presence of files
  - Network
    - Sockets, DNS, HTTP GET, ...
  - Cache, pipelining (Spectre, Meltdown)

(Adapted from Adrian Perrig)

# Dealing with Covert Channels

- **Confined program shall be memory-less**
  - no persistent storage across invocations
- **Total isolation: a confined program shall make no calls to another program**
- **Transitivity: if a confined program calls an untrusted program, untrusted program must also be confined**
- **Masking: caller can determine all inputs into legitimate and covert channels**
- **Enforcement: ensure that input into covert channel conforms to caller's specification**

(Adapted from Adrian Perrig)

# Even More Problematic Covert Channels

- **Magnetic emanations**
  - E.g., coming from monitors (CRT),
- **Acoustic emanations**
- **…**

# Covert Channels in Practice

- **Extremely hard to deal with**
  - See previous slides
- **Most of the decisions involve a trade-off between convenience/ease and amount of leakage**
- **Covert channels also apply outside of computer systems**
  - Example in Anderson

# Difficulties in MLS: Threat from Malware

- **First virus in 1983 by Fred Cohen to penetrate MLS**
  - Wasn't a trojan, took 8 hours to write!
- **Malware can break access controls**
  - By corrupting reference monitor or TCB
    - TPM was suggested
  - Malware to copy itself from low to high (BLP won't prevent)
    - Use covert channel for leakage

# Difficulties in MLS: Polyinstantiation

- **High user creates *agent* file, system blocks low users from creating the same file**
  - Information is now leaked!
- **Naming conventions helps (e.g. user directories)**
  - What about systems with databases?
- **Other solutions:**

| Level | Cargo | Destination |
|---|---|---|
| Secret | Missiles | Iran |
| Restricted | – | – |
| Unclassified | Engine spares | Cyprus |

US solution (Anderson)

| Level | Cargo | Destination |
|---|---|---|
| Secret | Missiles | Iran |
| Restricted | Classified | Classified |
| Unclassified | — | — |

UK solution (Anderson)

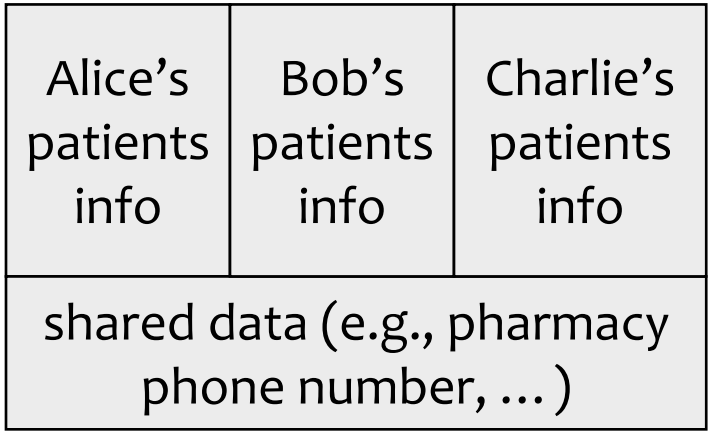
# Practical Difficulties in MLS

- **Built in small volumes; require high standards of robustness**
- **Require extensive training on tools and procedures**
- **Might require applications to be re-written**
- ***Over classification* can lead to a bigger TCB**
- **Classification can get complex**
  - Conflict; downgrade information when needed; nonmonotonic; volume of information; composability
- **Might prevent desired actions**
- **Can impair operations**
  - Lead to insecure trade-offs for usability

# Multilateral security

# Multilateral Security

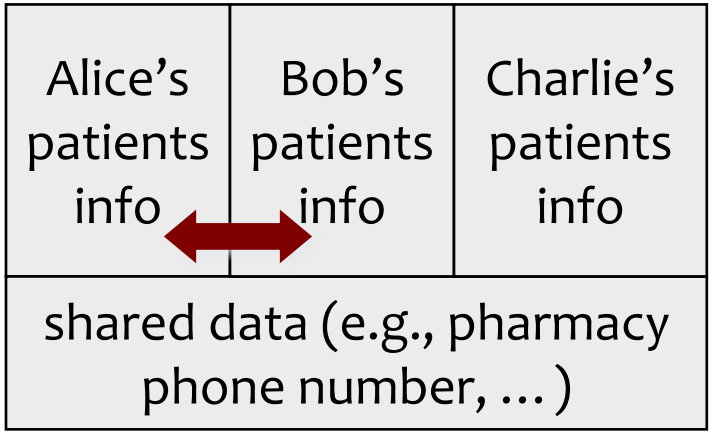
- **Also known as compartmentation in the US**
- **The idea is that you may want to prevent information from flowing across domains**
- **E.g., hospital**
  - Alice, Bob, Charlie are doctors



# Multilateral Security

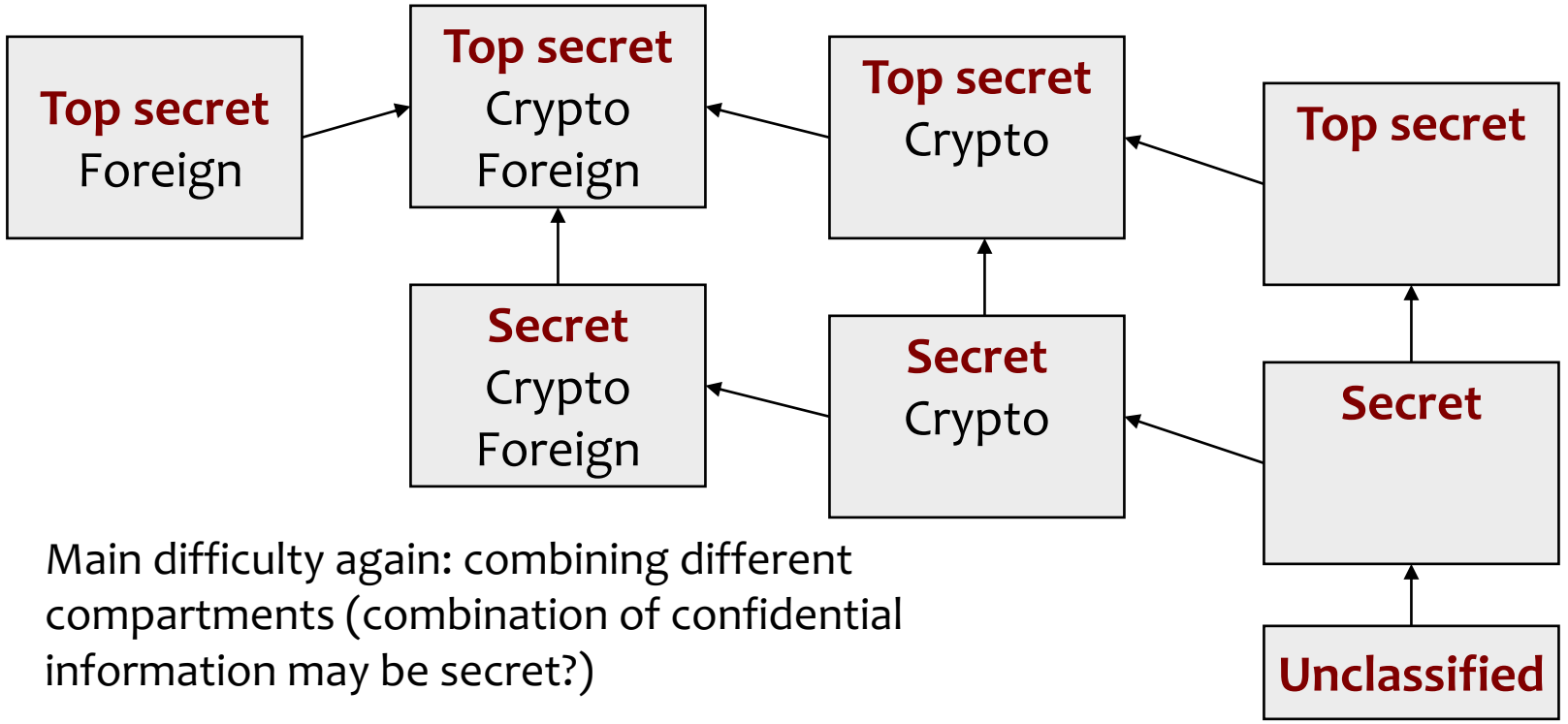
- **Also known as compartmentation in the US**
- **The idea is that you may want to prevent information from flowing across domains**
- **E.g., hospital**
  - Alice, Bob, Charlie are doctors

**Violation of doctor-patient privilege!**



# The Lattice Model

- **Generalizes the BLP model by adding codewords to security levels**
  - arrow represents allowed direction of information flow




Main difficulty again: combining different compartments (combination of confidential information may be secret?)

# Medical Records

- **BMA is British Medical Association**
- **Implemented access control policy in 1995**
- **Common issue: consistent classification**
  - People with a lethal disease "Secret"
  - Prescriptions at "Restricted"
  - Possibly easy to infer the disease with a "Restricted" clearance
  - Should everything be "Secret"?
  - Tough access control problem

# BMA: Principles (1/3)

- **Access control**
  - Every record needs a complete ACL. Default is deny access
- **Record opening**
  - Doctors can open records where they, the patient, and possibly the referring doctor are on the ACL
- **Control**
  - One of the physicians on the ACL is responsible ('owner') - only one who can modify the ACL

# BMA: Principles (2/3)

- **Consent and notification**
  - Physician must inform patient of initial ACL and of subsequent modifications
  - Consent is needed (unless emergency)
- **Persistence**
  - Can't delete any medical information before a (pre-determined) time period has expired
- **Attribution**
  - All accesses shall be marked and timestamped
  - Audit trail for deletions

# BMA: Principles (3/3)

- **Information flow**
  - Info derived from record A may be appended to record B only if B's ACL is a subset of A's
- **Aggregation control**
  - Prevent aggregation; e.g., special notification is required when an element that has access to a large number of records is added to the ACL
- **Trusted Computing Base (TCB)**
  - Computer systems that implement the policy should be reviewed by independent experts

# Issues with BMA

- Emergency care
- Resilience
- Secondary Uses
- Confidentiality
- Ethics
- Social care and Education

# The Chinese Wall

- **Initially used in banking to prevent conflicts of interests**
  - Two or more of the bank's customers may be competing with each other
  - Isolation is necessary
- **A mix of free choice and Mandatory Access Control**
  - A partner can choose the sector, but after that constraints apply

# The Chinese Wall

- **S is a subject (e.g., one of the bankers)**
- **C is a customer**
- **X(C) is C's competitors**
- **Y(C) is C's own company**

- **Simple security property – read policy**
  - S can read C if and only if for any C' that S can read, either Y(C') = Y(C) or Y(C') is not in X(C)
- ***-property – write policy**
  - S can write to C only if S can read C, and
  - Only if S cannot read any C' for which X(C') is not empty and Y(C)≠Y(C')

# The Chinese Wall: Example

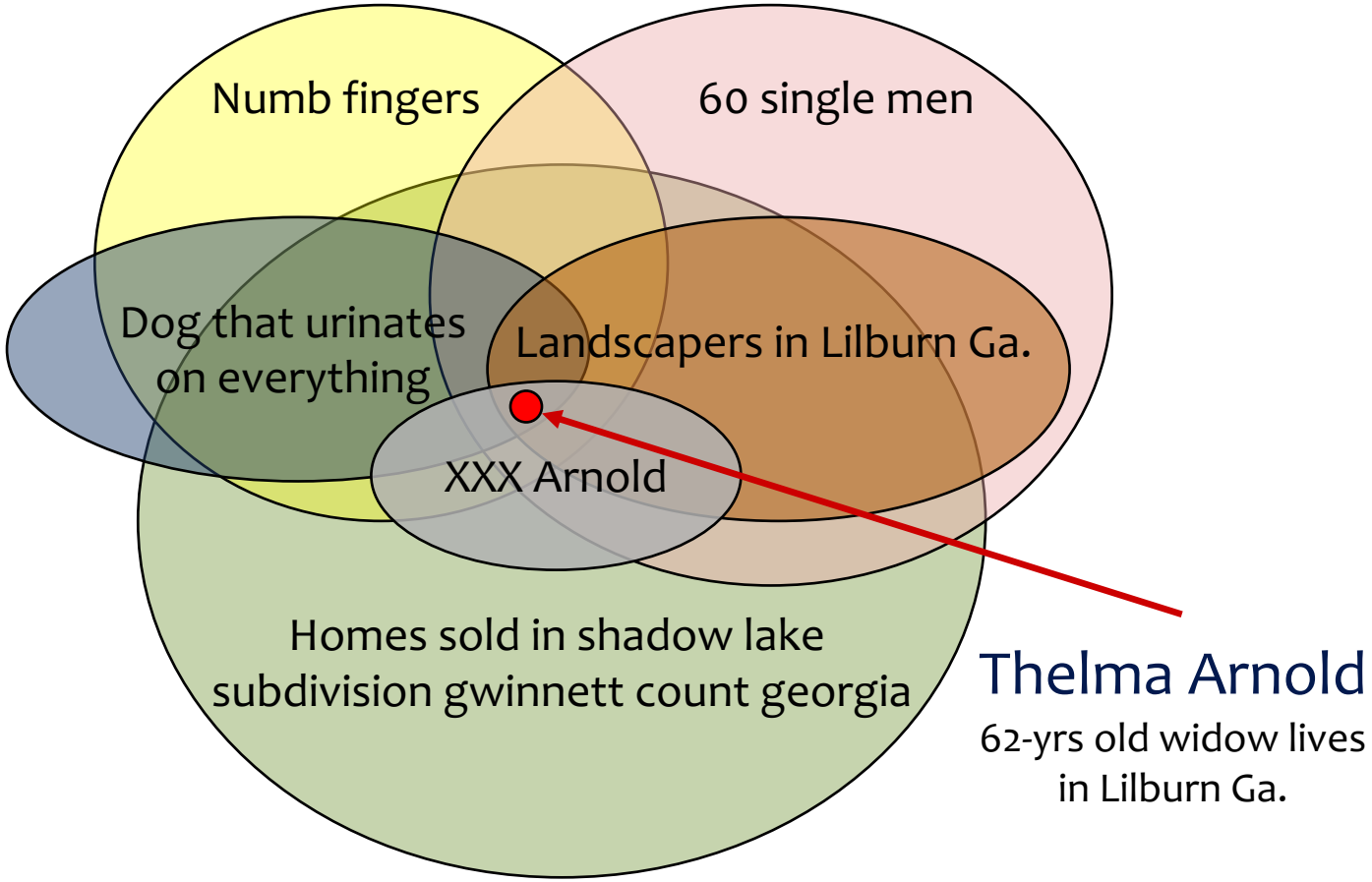
- **S = Scrooge is a banker**
- **Simple security property**
  - C = Donald Duck
  - Y(C) = Disney Ducks (C's own company)
    - Daisy Duck in Y(C)
  - X(C) = All other Disney characters (C's competitors)
  - The two customers that Scrooge can read are Daisy Ducks and Hello Kitty
  - Can scrooge read Donald Duck?
    - Scrooge can read Donald Duck's account if and only if for any C' that S can read either C' is part of Disney Ducks (e.g., C' = Daisy Duck… ) or Y(C') is not in X(C), e.g. C' = Hello Kitty
- ***-property**
  - Can Scrooge write to Hello Kitty? (Y(Hello Kitty) = Sanrio)
    - Scrooge can write to Hello Kitty account only if Scrooge can read it, and
    - Only if Scrooge cannot read any account C' which has competitors and who is not part of Sanrio
  - Star property prevents Scrooge from passing information to Rockerduck about Donald Duck through the Hello Kitty account, if Rockerduck is in charge of Mickey Mouse's account

# Comparison

- **Lattice model**
  - Describes compartmentation, but doesn't say how to manage information flows
- **Chinese Wall and BMA**
  - Say how to manage information flows
  - Chinese Wall = centralized
  - BMA = decentralized

# Inference Control: Problem

- **Even anonymized data can reveal too much**
- **AOL searcher No. 4417749 (among 20 million search queries)**



# Inference Control: Solutions

- **Restrict size of query set**
  - Reject queries that return less than *N* returns (where *N* is small, e.g., 6)
- **Reject extreme values in statistical data**
  - If you see that the average height in a particular village is 1m95,
  - If you see that, for many smaller groups, the average falls to 1m72
  - You can easily figure that Yao Ming or Shaquille O'Neal probably lives in the village
- **Cell suppression**
  - Conceal records that allow individual data to be reverse-engineered
  - May result in obfuscating large parts of a database
- **Only use random samples in aggregate reports**
- **…**

# Solution: Differential Privacy Solution

- **Add noise to prevent disclosure of sensitive information**
- **Limit the probability of of disclosure**
  - Even if adversary has unlimited computational power
- **Can be used in**
  - Statistical database security
  - Anonymization

# Take Away Slide

- **Multilevel security is a very hard problem**
  - Preventing flows of information can be done in theory
  - Harder to do in practice
    - Most communications are two-way
  - Derived from the military but extremely useful for networked systems, for instance
- **Multilateral security is an equally hard problem**
  - Setting up interfaces (ACL) is the easiest part
  - Statistical problems are much harder to deal with (inference control)
  - The interplay between inference control and proper ACL is even harder to figure out
  - Privacy concerns
- **Access control is an issue that far exceeds computer systems**
- **Problem is complicated by the notion of information flows**

# Multilevel and Multilateral Summary

| Multilevel Security(MLS) | Multilateral Security |
| --- | --- |
| Security Levels are used for decisions | Define Security policies according to some rules |
| BLP: Confidentiality<br>Biba: Integrity | Security between different actors (agents). Actors can have the same clearance level |
| Enforces access control up and down | Enforces access control by compartments. Compartments can be at the same level |