Access Control

Access Control

- Two parts to access control…
- ***Authentication:*** Are you who you say you are?
    - Determine whether access is allowed or not
    - Authenticate human to machine
    - Or, possibly, machine to machine
- ***Authorization:*** Are you allowed to do that?
    - Once you have access, what can you do?
    - Enforces limits on actions
- Note: "access control" often used as synonym for authorization

Chapter 7: Authentication

Guard: Halt! Who goes there?
Arthur: It is I, Arthur, son of Uther Pendragon,
from the castle of Camelot. King of the Britons,
defeater of the Saxons, sovereign of all England!
Monty Python and the Holy Grail

Then said they unto him, Say now Shibboleth:
and he said Sibboleth: for he could not frame to pronounce it right.
Then they took him, and slew him at the passages of Jordan:
and there fell at that time of the Ephraimites forty and two thousand.
Judges 12:6

Are You Who You Say You Are?

- Authenticate a human to a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a smartcard
  - Something you **are**
    - For example, your fingerprint

Something You Know

- Passwords
- Lots of things act as passwords!
  - PIN
  - Social security number
  - Mother's maiden name
  - Date of birth
  - Name of your pet, etc.

Trouble with Passwords

- "Passwords are one of the biggest practical problems facing security engineers today."
- "Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed.)"

Why Passwords?

- Why is "something you know" more popular than "something you have" and "something you are"?
- **Cost**: passwords are free
- **Convenience**: easier for sysadmin to reset pwd than to issue a new thumb

Keys vs Passwords

- **Crypto keys**
- Spse key is 64 bits
- Then 2^{64} keys
- Choose key at random…
- …then attacker must try about 2^{63} keys

- **Passwords**
- Spse passwords are 8 characters, and 256 different characters
- Then $256^8 = 2^{64}$ pwds
- Users do not select passwords at random
- Attacker has far less than 2^{63} pwds to try (**dictionary attack**)

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - Password
  - incorrect
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jfIej,43j-EmmL+y
  - 09864376537263
  - P0kem0N
  - FSa7Yago
  - 0nceuP0nAt1m8
  - PokeGCTall150

Password Experiment

- Three groups of users each group advised to select passwords as follows
  - **Group A:** At least 6 chars, 1 non-letter
  - winner **Group B:** Password based on passphrase
  - **Group C:** 8 random characters
- Results
  - **Group A:** About 30% of pwds easy to crack
  - **Group B:** About 10% cracked
    - Passwords easy to remember
  - **Group C:** About 10% cracked
    - Passwords hard to remember

Password Experiment

- User compliance hard to achieve
- In each case, 1/3rd did not comply
  - And about 1/3rd of those easy to crack!
- Assigned passwords sometimes best
- If passwords not assigned, best advice is…
  - Choose passwords based on passphrase
  - Use pwd cracking tool to test for weak pwds
- Require periodic password changes?

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
- Common attack path
  - Outsider → normal user → administrator
  - May only require ***one*** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock?
  - 5 seconds
  - 5 minutes
  - Until SA restores service
- What are +'s and -'s of each?

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Solution? **Hash** passwords
  - Store y = h(password)
  - Can verify entered password by hashing
  - If Trudy obtains the password file, she does not (directly) obtain passwords
- But Trudy can try a forward search
  - Guess x and check whether y = h(x)

Dictionary Attack

- Trudy pre-computes h(x) for all x in a **dictionary** of common passwords
- Suppose Trudy gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work of computing hashes in dictionary, actual attack is trivial
- Can we prevent this forward search attack? Or at least make it more difficult?

Salt

- Hash password with **salt**
- Choose random salt s and compute

$$y = h(\text{password}, s)$$

and store (s,y) in the password file
- Note that the salt s is not secret
  - Analogous to IV
- Still easy to verify salted password
- But lots more work for Trudy
  - Why?

Password Cracking: Do the Math

- Assumptions:
- Pwds are 8 chars, 128 choices per character
  - Then $128^8 = 2^{56}$ possible passwords
- There is a **password file** with 2^{10} pwds
- Attacker has **dictionary** of 2^{20} common pwds
- **Probability** 1/4 that password is in dictionary
- **Work** is measured by number of hashes

Password Cracking: Case I

- Attack 1 specific password **without** using a dictionary
  - E.g., administrator's password
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Like exhaustive key search
- Does **salt** help in this case?

Password Cracking: Case II

- Attack 1 specific password **with** dictionary
- With ***salt***
  - Expected work: 1/4 (2^{19}) + 3/4 (2^{55}) ≈ $2^{54.6}$
  - In practice, try all pwds in dictionary…
  - …then work is at most 2^{20} and probability of success is 1/4
- What if ***no salt*** is used?
  - One-time work to compute dictionary: 2^{20}
  - Expected work is of same order as above
  - But with precomputed dictionary hashes, the "in practice" attack is essentially free…

Password Cracking: Case III

- Any of 1024 pwds in file, **without** dictionary
  - Assume all 2^{10} passwords are distinct
  - Need 2^{55} **comparisons** before expect to find pwd
- If **no salt** is used
  - Each computed hash yields 2^{10} comparisons
  - So expected work (hashes) is $2^{55}/2^{10} = 2^{45}$
- If **salt** is used
  - Expected work is 2^{55}
  - Each comparison requires a hash computation

Password Cracking: Case IV

- Any of 1024 pwds in file, **with** dictionary
  - Prob. one or more pwd in dict.: $1 - (3/4)^{1024} \approx 1$
  - So, we ignore case where no pwd is in dictionary
- If **salt** is used, expected work less than 2^{22}
  - See book, or slide notes for details
  - Work $\approx$ size of dictionary / P(pwd in dictionary)
- What if **no salt** is used?
  - If dictionary hashes not precomputed, work is about $2^{19}/2^{10} = 2^9$

Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Why is this a problem?
- Who suffers from bad password?
  - Login password vs ATM PIN
- Failure to change default passwords
- Social engineering
- Error logs may contain "almost" passwords
- Bugs, keystroke logging, spyware, etc.

Passwords

- The bottom line…
- Password attacks are too easy
  - Often, one weak password will break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Trudy has (almost) all of the advantages
- All of the math favors bad guys
- Passwords are a **BIG** security problem
  - And will continue to be a problem

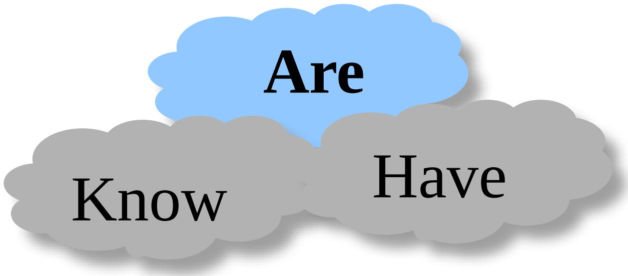
Password Cracking Tools

- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admins should use these tools to test for weak passwords since attackers will
- Good articles on password cracking
  - Passwords - Conerstone of Computer Security
  - Passwords revealed by sweet deal

Biometrics

Something You Are

- Biometric
  - "You are your key" Schneier
- Examples
  - Fingerprint
  - Handwritten signature
  - Facial recognition
  - Speech recognition
  - Gait (walking) recognition
  - "Digital doggie" (odor recognition)
  - Many more!



Why Biometrics?

- May be better than passwords
- But, cheap and reliable biometrics needed
  - Today, an active area of research
- Biometrics **are** used in security today
  - Thumbprint mouse
  - Palm print for secure entry
  - Fingerprint to unlock car door, etc.
- But biometrics not really that popular
  - Has not lived up to its promise/hype (yet?)

Ideal Biometric

- *Universal* — applies to (almost) everyone
  - In reality, no biometric applies to everyone
- *Distinguishing* — distinguish with certainty
  - In reality, cannot hope for 100% certainty
- *Permanent* — physical characteristic being measured never changes
  - In reality, OK if it to remains valid for long time
- *Collectable* — easy to collect required data
  - Depends on whether subjects are cooperative
- Also, safe, user-friendly, and ???

Identification vs Authentication

- ***Identification*** — Who goes there?
  - Compare **one-to-many**
  - Example: FBI fingerprint database
- ***Authentication*** — Are you who you say you are?
  - Compare **one-to-one**
  - Example: Thumbprint mouse
- Identification problem is more difficult
  - More "random" matches since more comparisons
- We are (mostly) interested in authentication

Enrollment vs Recognition

- Enrollment phase
  - Subject's biometric info put into database
  - Must carefully measure the required info
  - OK if slow and repeated measurement needed
  - Must be very precise
  - May be a weak point in real-world use
- Recognition phase
  - Biometric detection, when used in practice
  - Must be quick and simple
  - But must be reasonably accurate

Cooperative Subjects?

- Authentication   cooperative subjects
- Identification   uncooperative subjects
- For example, facial recognition
  - Used in Las Vegas casinos to detect known cheaters (also, terrorists in airports, etc.)
  - Often, less than ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier
  - We are focused on authentication
  - So, we can assume subjects are cooperative

Biometric Errors

- **Fraud rate** versus **insult rate**
  - Fraud Trudy mis-authenticated as Alice
  - Insult Alice not authenticated as Alice
- For any biometric, can decrease fraud or insult, but other one will increase
- For example
  - 99% voiceprint match low fraud, high insult
  - 30% voiceprint match high fraud, low insult
- **Equal error rate:** rate where fraud == insult
  - A way to **compare** different biometrics

Fingerprint History

- 1823 Professor Johannes Evangelist Purkinje discussed 9 fingerprint patterns
- 1856 Sir William Hershel used fingerprint (in India) on contracts
- 1880 Dr. Henry Faulds article in Nature about fingerprints for ID
- 1883 Mark Twain's Life on the Mississippi (murderer ID'ed by fingerprint)

Fingerprint History

- 1888   Sir Francis Galton developed classification system
  - His system of "minutia" can be used today
  - Also verified that fingerprints do not change
- Some countries require fixed number of "points" (minutia) to match in criminal cases
  - In Britain, at least 15 points
  - In US, no fixed number of points

Fingerprint Comparison

- Examples of **loops**, **whorls**, and **arches**
- Minutia extracted from these features

Loop (double)

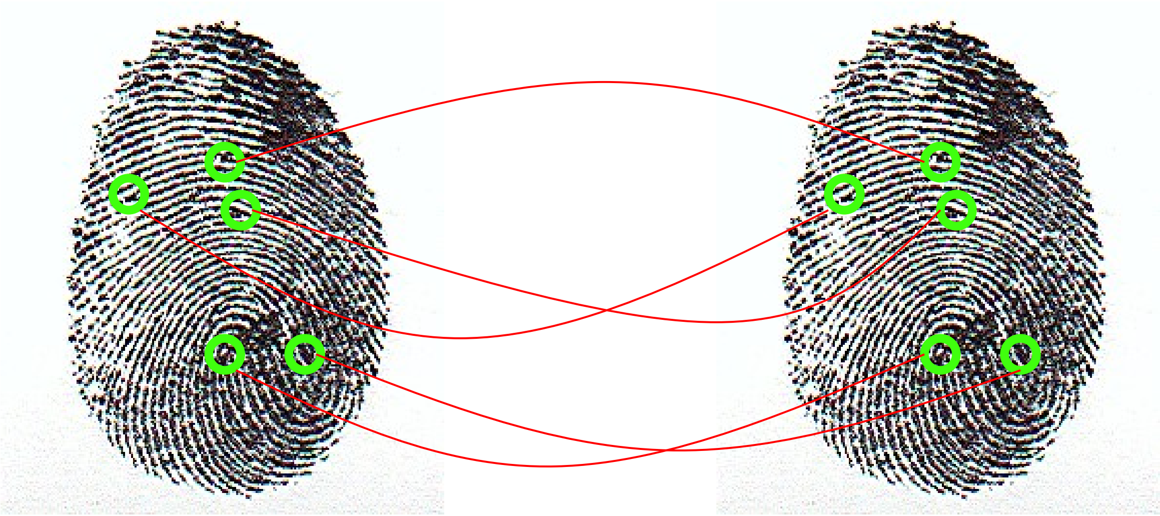
Whorl

Arch

Fingerprint: Enrollment

- Capture image of fingerprint
- Enhance image
- Identify "points"

Fingerprint: Recognition



- Extracted points are compared with information stored in a database
- Is it a statistical match?
- Aside: [Do identical twins' fingerprints differ](#)?

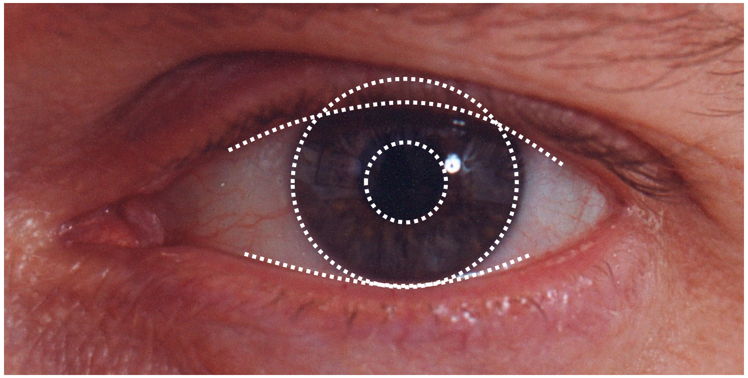
Hand Geometry

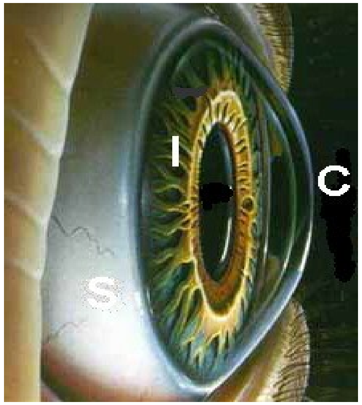
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not so unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

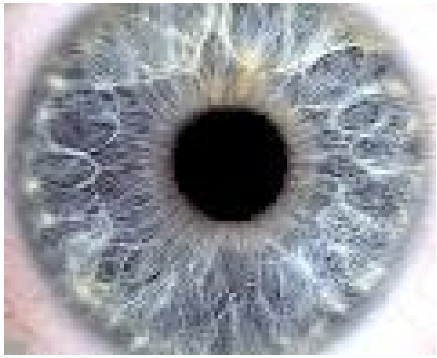
Hand Geometry

- Advantages
  - Quick   1 minute for enrollment,   5 seconds for recognition
  - Hands are symmetric   so what?
- Disadvantages
  - Cannot use on very young or very old
  - Relatively high equal error rate

Iris Patterns







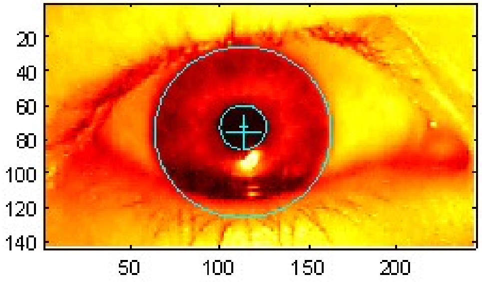
- Iris pattern development is "chaotic"
- Little or no genetic influence
- Even for identical twins, uncorrelated
- Pattern is stable through lifetime

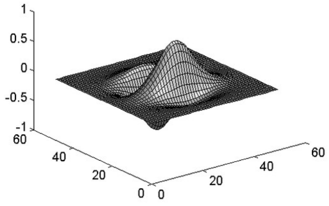
Iris Recognition: History

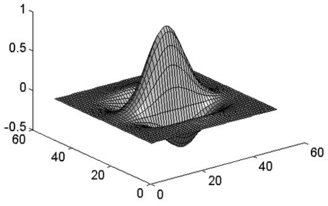
- 1936 suggested by ophthalmologist
- 1980s James Bond film(s)
- 1986 first patent appeared
- 1994 John Daugman patents new-and-improved technique
  - Patents owned by Iridian Technologies

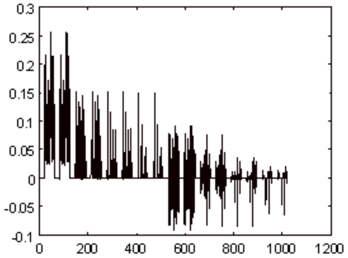
Iris Scan

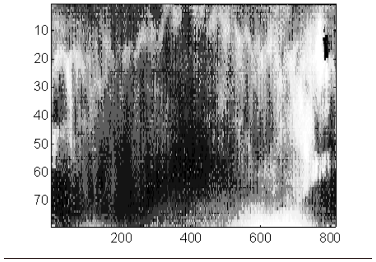
- *Scanner locates iris*
- *Take b/w photo*
- *Use polar coordinates…*
- *2-D wavelet transform*
- *Get 256 byte iris code*











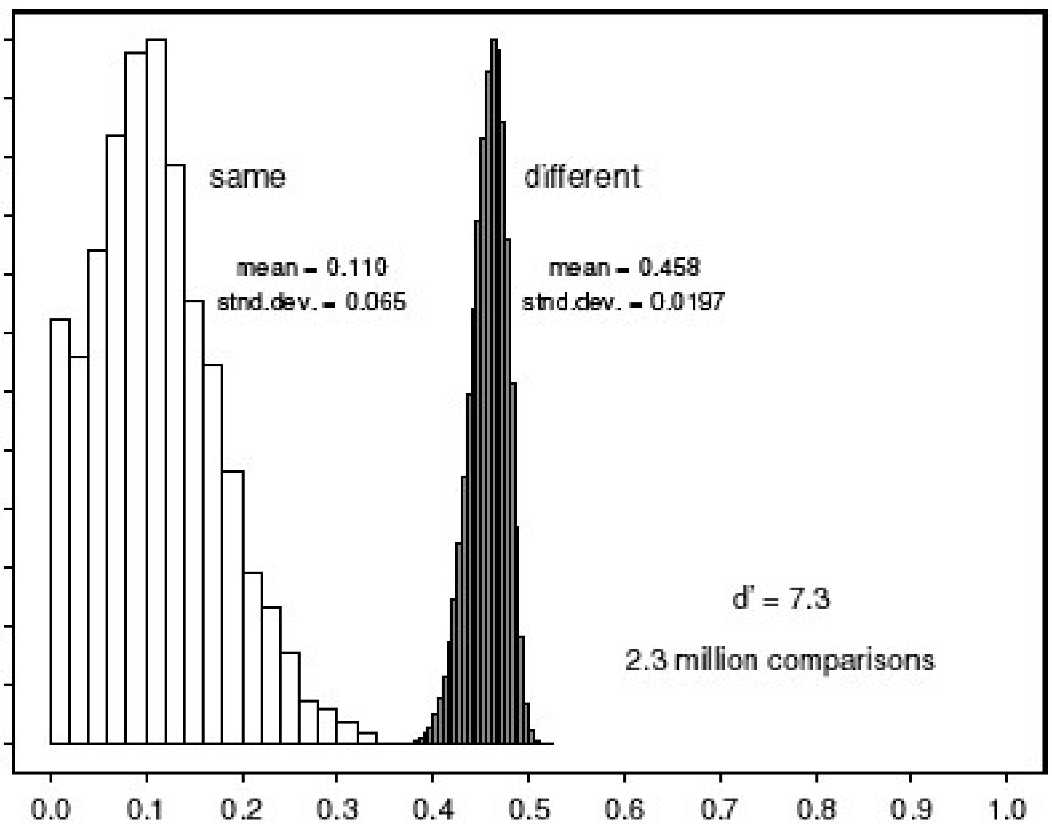
Measuring Iris Similarity

- Based on Hamming distance
- Define d(x,y) to be
  - # of non-match bits / # of bits compared
  - d(0010,0101) = 3/4 and d(101111,101001) = 1/3
- Compute d(x,y) on 2048-bit iris code
  - Perfect match is d(x,y) = 0
  - For same iris, expected distance is 0.08
  - At random, expect distance of 0.50
  - Accept iris scan as match if distance < 0.32

Iris Scan Error Rate

distance	Fraud rate
0.29	1 in 1.3×10^{10}
0.30	1 in 1.5×10^{9}
0.31	1 in 1.8×10^{8}
0.32	1 in 2.6×10^{7}
0.33	1 in 4.0×10^{6}
0.34	1 in 6.9×10^{5}
0.35	1 in 1.3×10^{5}

== *equal error rate*



distance

Attack on Iris Scan

- Good **photo** of eye can be scanned
  - Attacker could use photo of eye
- Afghan woman was authenticated by iris scan of old photo
  - Story can be found here
- To prevent attack, scanner could use light to be sure it is a "live" iris

Equal Error Rate Comparison

- Equal error rate (EER): fraud == insult rate
- **Fingerprint** biometrics used in practice have EER ranging from about 10^{-3} to as high as 5%
- **Hand geometry** has EER of about 10^{-3}
- In theory, **iris scan** has EER of about 10^{-6}
  - Enrollment phase may be critical to accuracy
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
  - …but for identification, not so impressive today

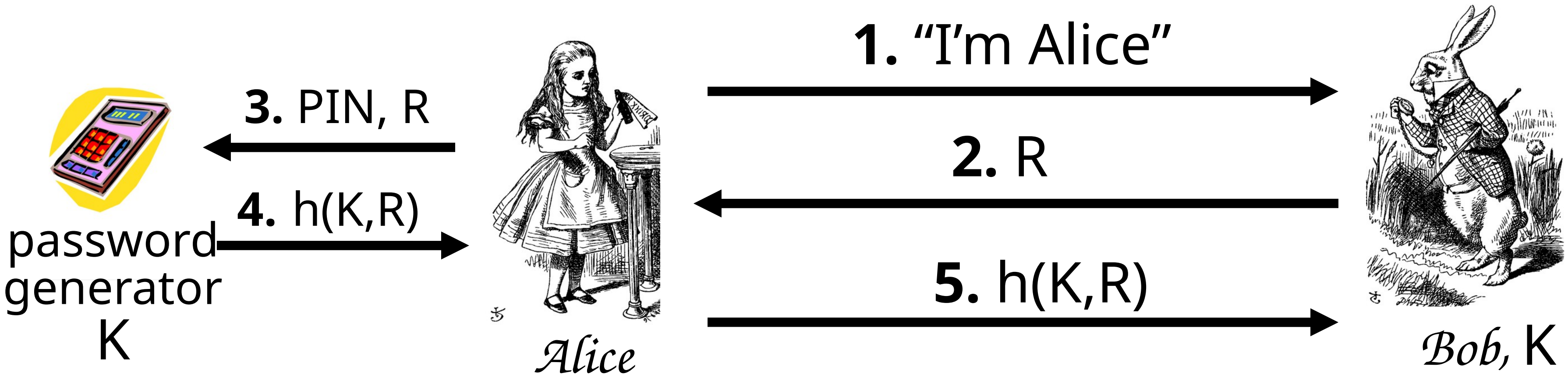
Biometrics: The Bottom Line

- Biometrics are hard to forge
- But attacker could
  - Steal Alice's thumb
  - Photocopy Bob's fingerprint, eye, etc.
  - Subvert software, database, "trusted path" …
- And how to revoke a "broken" biometric?
- **Biometrics are not foolproof**
- Biometric use is relatively limited today
- That should change in the (near?) future

Something You Have

- Something in your possession
- Examples include following…
  - Car key
  - Laptop computer (or MAC address)
  - Password generator (next)
  - ATM card, smartcard, etc.

Password Generator



- *Alice receives random "challenge"* R *from Bob*
- *Alice enters* PIN *and* R *in password generator*
- *Password generator hashes symmetric key* K *with* R
- *Alice sends "response"* h(K,R) *back to Bob*
- *Bob verifies response*
- *Note: Alice* ***has*** *pwd generator and* ***knows*** PIN

2-factor Authentication

- Requires any 2 out of 3 of
  - Something you **know**
  - Something you **have**
  - Something you **are**
- Examples
  - ATM: Card and PIN
  - Credit card: Card and signature
  - Password generator: Device and PIN
  - Smartcard with password/PIN

Single Sign-on

- A hassle to enter password(s) repeatedly
  - Alice would like to authenticate only once
  - "Credentials" stay with Alice wherever she goes
  - Subsequent authentications transparent to Alice
- Kerberos a single sign-on protocol
- Single sign-on for the Internet?
  - Microsoft: **Passport**
  - Everybody else: **Liberty Alliance**
  - Security Assertion Markup Language (**SAML**)

Web Cookies

- Cookie is provided by a Website and stored on user's machine
- Cookie indexes a database at Website
- Cookies **maintain state** across sessions
  - Web uses a stateless protocol: HTTP
  - Cookies also maintain state within a session
- Sorta like a single sign-on for a website
  - But, very, very weak form of authentication
- Cookies also create privacy concerns

Authorization

Chapter 8: Authorization

It is easier to exclude harmful passions than to rule them,
and to deny them admittance
than to control them after they have been admitted.
Seneca

You can always trust the information given to you
by people who are crazy;
they have an access to truth not available through regular channels.
Sheila Ballantyne

Authentication vs Authorization

- Authentication Are you who you say you are?
    - Restrictions on who (or what) can access system
- **Authorization** Are you allowed to do that?
    - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- But first, we look at system certification…

System Certification

- Government attempt to certify "security level" of products
- Of historical interest
  - Sorta like a history of authorization
- Still important today if you want to sell a product to the government
  - Tempting to argue it's a failure since government is so insecure, but…

Orange Book

- Trusted Computing System Evaluation Criteria (TCSEC), 1983
  - Universally known as the "orange book"
  - Name is due to color of it's cover
  - About 115 pages
  - Developed by U.S. DoD (NSA)
  - Part of the "rainbow series"
- Orange book generated a pseudo-religious fervor among some people
  - Less and less intensity as time goes by

Orange Book Outline

- Goals
  - Provide way to assess security products
  - Provide general guidance/philosophy on how to build more secure products
- Four **divisions** labeled D thru A
  - D is lowest, A is highest
- Divisions split into numbered **classes**

D and C Divisions

- D minimal protection
  - Losers that can't get into higher division
- C discretionary protection, i.e., don't enforce security, just have means to detect breaches (audit)
  - C1 discretionary security protection
  - C2 controlled access protection
  - C2 slightly stronger than C1 (both vague)

B Division

- B mandatory protection
- B is a **huge** step up from C
  - C: break security, you might get caught
  - B: "mandatory", so you can't break it
- B1 labeled security protection
  - All data labeled, which restricts what can be done with it
  - This access control cannot be violated

B and A Divisions

- B2 structured protection
  - Adds covert channel protection onto B1
- B3 security domains
  - On top of B2 protection, adds that code must be **tamperproof** and "small"
- A verified protection
  - Like B3, but **proved** using formal methods
  - Such methods still (mostly) impractical

Orange Book: Last Word

- Also a 2nd part, discusses rationale
- Not very practical or sensible, IMHO
- But some people insist we'd be better off if we'd followed it
- Others think it was a dead end
  - And resulted in lots of wasted effort
  - Aside… people who made the orange book, now set security education standards

Common Criteria

- Successor to the orange book (ca. 1998)
  - Due to inflation, more than 1000 pages
- An international government standard
  - And it reads like it…
  - Won't ever stir same passions as orange book
- CC is relevant in practice, but usually only if you want to sell to the government
- Evaluation Assurance Levels (EALs)
  - 1 thru 7, from lowest to highest security

EAL

- Note: product with high EAL may not be more secure than one with lower EAL
  - Why?
- Similarly, product with an EAL may not be any more secure than one without
  - Why?

EAL 1 thru 7

- EAL1 functionally tested
- EAL2 structurally tested
- EAL3 methodically tested, checked
- EAL4 **designed**, tested, reviewed
- EAL5 semiformally designed, tested
- EAL6 verified, designed, tested
- EAL7 formally … (blah blah blah)

Common Criteria

- EAL4 is most commonly sought
  - Minimum needed to sell to government
- EAL7 requires formal proofs
  - Author could only find 2 EAL7 products…
- Who performs evaluations?
  - Government accredited labs, of course (for a hefty fee, like 6 figures)

Authentication vs Authorization

- Authentication $\rightarrow$ Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** $\rightarrow$ Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- Classic view of authorization…
  - Access Control Lists (ACLs)
  - Capabilities (C-lists)

Lampson's Access Control Matrix

- **Subjects** (users) index the rows
- **Objects** (resources) index the columns

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r		
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Are You Allowed to Do That?

- **Access control matrix** has **all** relevant info
- Could be 100's of users, 10,000's of resources
  - Then matrix has 1,000,000's of entries
- How to manage such a large matrix?
- Note: We need to check this matrix before access to any resource by any user
- How to make this more efficient/practical?

Access Control Lists (ACLs)

- ACL: store access control matrix by **column**
- Example: ACL for **insurance data** is in **blue**

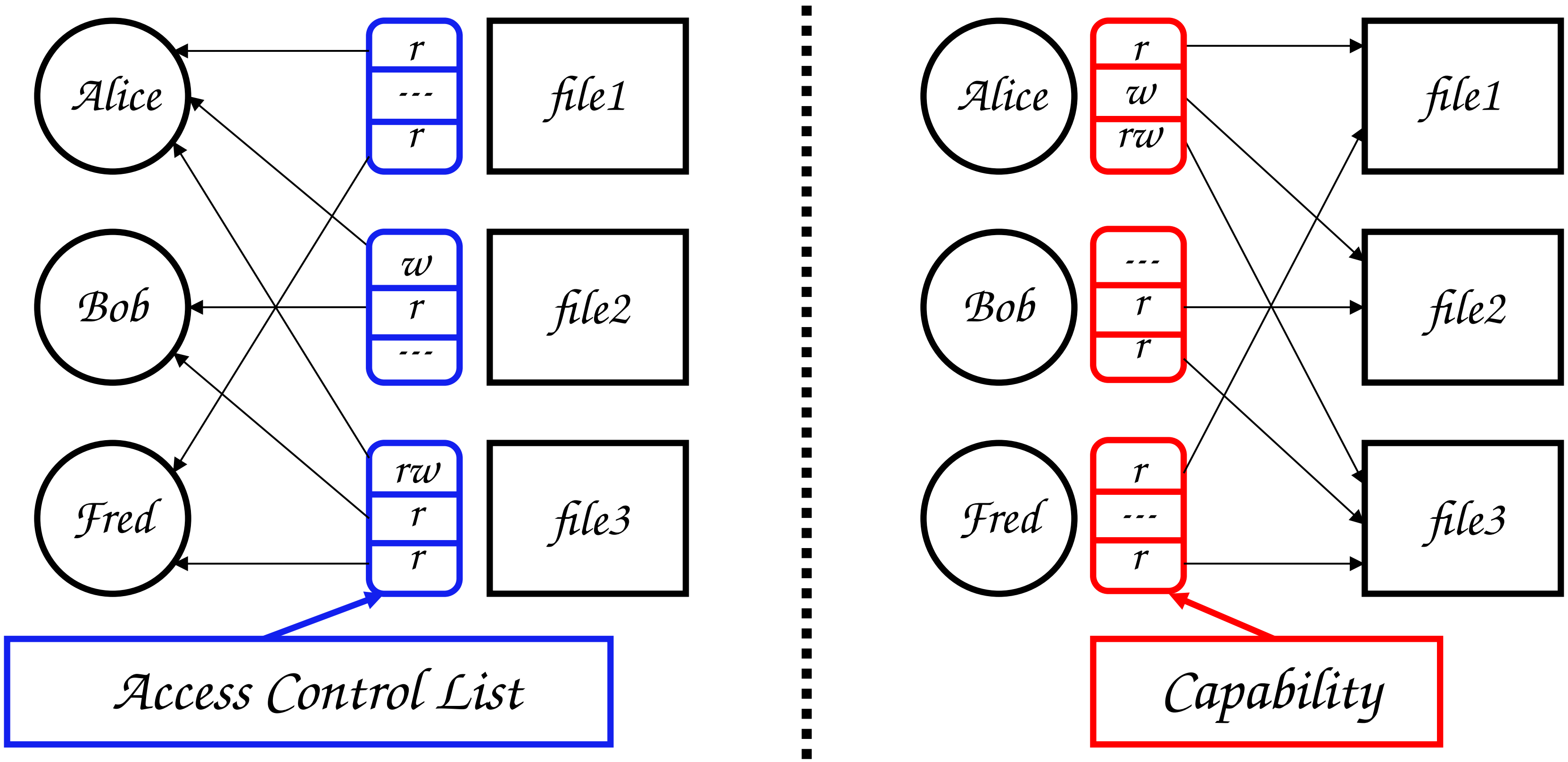
	OS	Accounting program	Accounting data	**Insurance data**	Payroll data
Bob	rx	rx	r		
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Capabilities (or C-Lists)

- Store access control matrix by **row**
- Example: Capability for **Alice** is in **red**

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r		
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

ACLs vs Capabilities



- Note that arrows point in opposite directions…
- With ACLs, still need to associate users to files

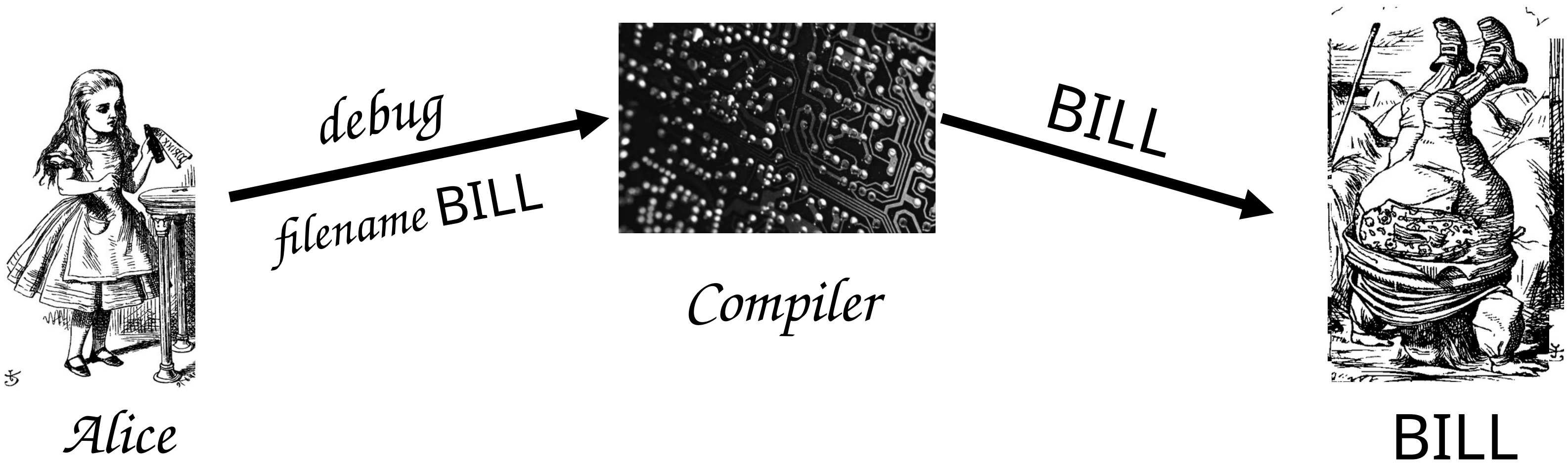
Confused Deputy

- Two resources
  - Compiler and BILL file (billing info)
- Compiler can write file BILL
- Alice can invoke compiler with a debug filename
- Alice not allowed to write to BILL

- Access control matrix

	Compiler	BILL
Alice	x	
Compiler	rx	rw

ACL's and Confused Deputy



- Compiler is ***deputy*** acting on behalf of Alice
- Compiler is ***confused***
  - Alice is not allowed to write BILL
- Compiler has confused its rights with Alice's

Confused Deputy

- Compiler acting for Alice is confused
- There has been a separation of **authority** from the **purpose** for which it is used
- With ACLs, more difficult to prevent this
- With Capabilities, easier to prevent problem
  - Must maintain association between authority and intended purpose
- Capabilities easy to **delegate** authority

ACLs vs Capabilities

- ACLs
  - Good when users manage their own files
  - Protection is data-oriented
  - Easy to change rights to a resource
- Capabilities
  - Easy to delegate avoid the confused deputy
  - Easy to add/delete users
  - More difficult to implement
  - The "Zen of information security"
- Capabilities loved by academics
  - Capability Myths Demolished

Multilevel Security (MLS) Models

Classifications and Clearances

- ***Classifications*** apply to ***objects***
- ***Clearances*** apply to ***subjects***
- US Department of Defense (DoD) uses 4 levels:

  TOP SECRET

  SECRET

  CONFIDENTIAL

  UNCLASSIFIED

Clearances and Classification

- To obtain a **SECRET** clearance requires a routine background check
- A **TOP SECRET** clearance requires extensive background check
- Practical classification problems
  - Proper classification not always clear
  - Level of granularity to apply classifications
  - Aggregation flipside of granularity

Subjects and Objects

- Let O be an ***object***, S a ***subject***
  - O has a classification
  - S has a clearance
  - Security ***level*** denoted L(O) and L(S)
- For DoD levels, we have

  TOP SECRET > **SECRET** >

  CONFIDENTIAL > **UNCLASSIFIED**

Multilevel Security (MLS)

- MLS needed when subjects/objects at different levels access **same system**
- MLS is a form of **Access Control**
- Military and government interest in MLS for many decades
  - Lots of research into MLS
  - Strengths and weaknesses of MLS well understood (almost entirely theoretical)
  - Many possible uses of MLS outside military

MLS Applications

- Classified government/military systems
- **Business example:** info restricted to
  - Senior management only, all management, everyone in company, or general public
- Network firewall
- Confidential medical info, databases, etc.
- Usually, MLS not really a technical system
  - More like part of a legal structure

MLS Security Models

- MLS models explain **what** needs to be done
- Models **do not** tell you **how** to implement
- Models are descriptive, not prescriptive
  - That is, high-level description, not an algorithm
- There are many MLS models
- We'll discuss simplest MLS model
  - Other models are more realistic
  - Other models also more complex, more difficult to enforce, harder to verify, etc.

Bell-LaPadula

- BLP security model designed to express essential requirements for MLS
- BLP deals with **confidentiality**
  - To prevent unauthorized reading
- Recall that O is an object, S a subject
  - Object O has a classification
  - Subject S has a clearance
  - Security level denoted L(O) and L(S)

Bell-LaPadula

- *BLP consists of*
  - ***Simple Security Condition**: S can read O if and only if $L(O) \leq L(S)$*
  - ****-Property** (**Star Property**): S can write O if and only if $L(S) \leq L(O)$*
- ***No read up, no write down***

McLean's Criticisms of BLP

- McLean: BLP is "so trivial that it is hard to imagine a realistic security model for which it does not hold"
- McLean's "system Z" allowed administrator to reclassify object, then "write down"
- Is this fair?
- Violates spirit of BLP, but **not** expressly forbidden in statement of BLP
- Raises fundamental questions about the nature of (and limits of) modeling

B and LP's Response

- BLP enhanced with ***tranquility property***
  - *Strong tranquility*: security labels never change
  - *Weak tranquility*: security label can only change if it does not violate "established security policy"
- Strong tranquility impractical in real world
  - Often want to enforce "least privilege"
  - Give users lowest privilege for current work
  - Then upgrade as needed (and allowed by policy)
  - This is known as the ***high water mark*** principle
- Weak tranquility allows for **least privilege** (high water mark), but the property is vague

BLP: The Bottom Line

- BLP is simple, probably too simple
- BLP is one of the few security models that can be used to prove things about systems
- BLP has inspired other security models
  - Most other models try to be more realistic
  - Other security models are more complex
  - Models difficult to analyze, apply in practice

Biba's Model

- BLP for confidentiality, Biba for ***integrity***
  - Biba is to prevent unauthorized writing
- Biba is (in a sense) the dual of BLP
- Integrity model
  - Spse you trust the integrity of **O** but not **O**
  - If object **O** includes **O** and **O** then you cannot trust the integrity of **O**
- Integrity level of O is minimum of the integrity of any object in O
- ***Low water mark*** principle for integrity

Biba

- *Let* I(O) *denote the integrity of object* O *and* I(S) *denote the integrity of subject* S
- *Biba can be stated as*

  Write Access Rule: S *can write* O *if and only if* I(O) ✹I(S)

  (if S *writes* O*, the integrity of* O ✹*that of* S*)*

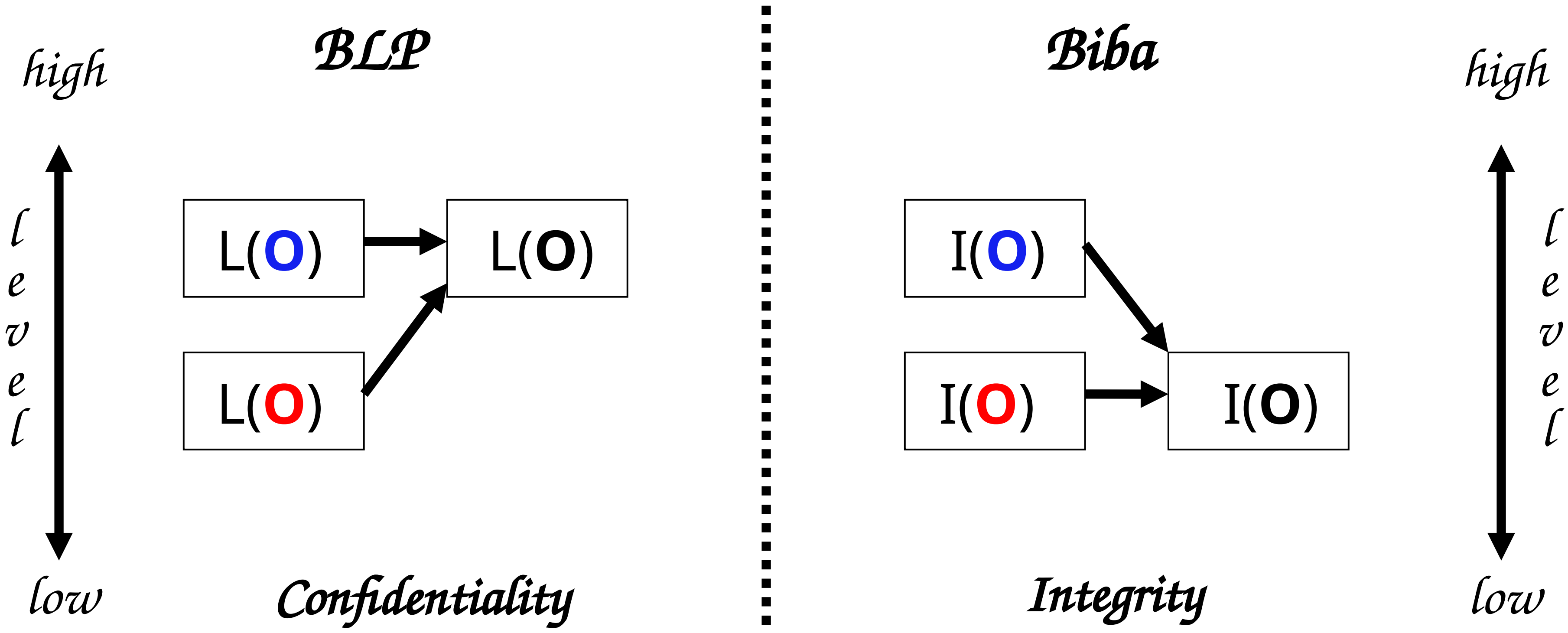
  Biba's Model: S *can read* O *if and only if* I(S) ✹ I(O)

  (if S *reads* O*, the integrity of* S ✹*that of* O*)*

- *Often, replace Biba's Model with*

  Low Water Mark Policy: *If* S *reads* O*, then* I(S) = min(I(S), I(O))

BLP vs Biba



BLP
Biba
high
high
level
level
L(O)
L(O)
L(O)
I(O)
I(O)
I(O)
low
low
Confidentiality
Integrity

Compartments

Compartments

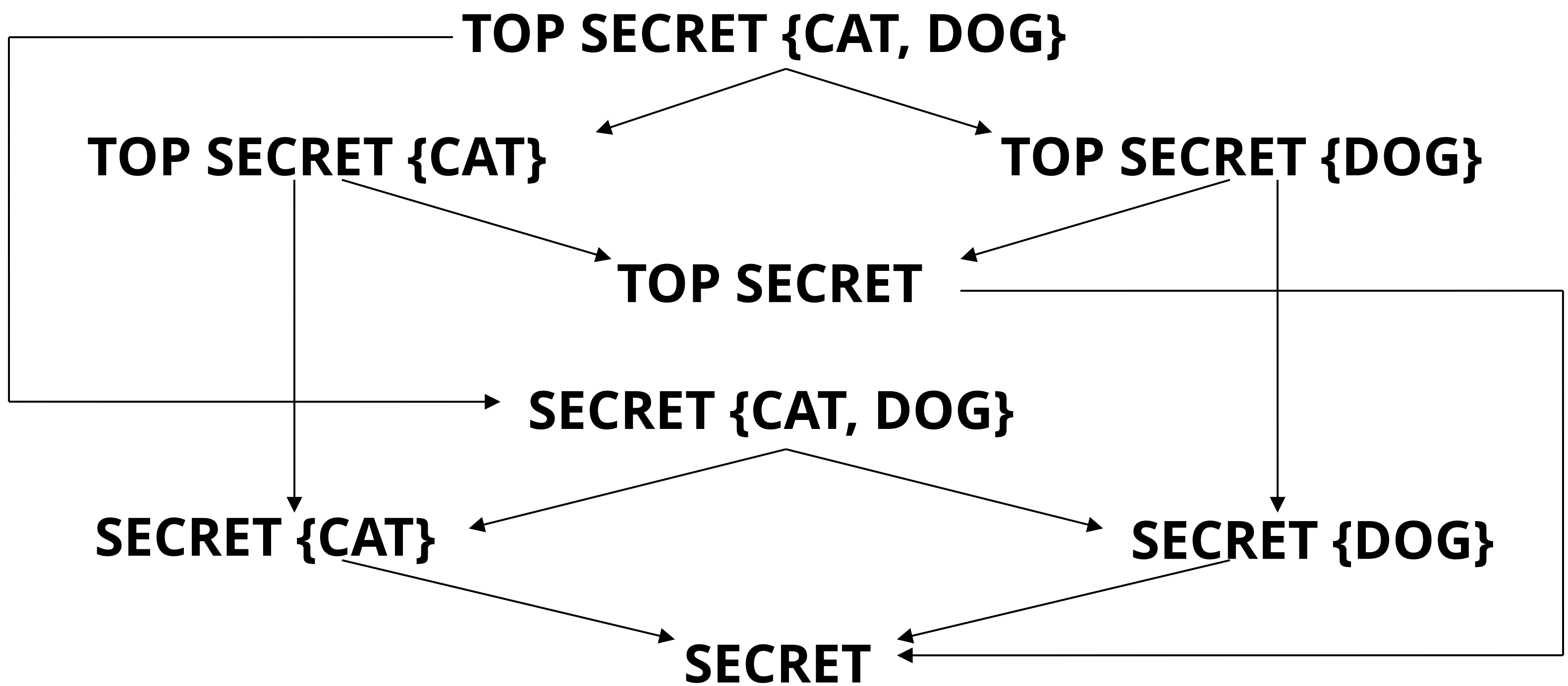
- *Multilevel Security (MLS) enforces access control* ***up and down***
- *Simple hierarchy of security labels is generally not flexible enough*
- *Compartments enforces restrictions* ***across***
- *Suppose* **TOP SECRET** *divided into* **TOP SECRET {CAT}** *and* **TOP SECRET {DOG}**
- *Both are* **TOP SECRET** *but information flow restricted across the* **TOP SECRET** *level*

Compartments

- *Why compartments?*
  - *Why not create a new classification level?*
- *May **not** want either of*
  - **TOP SECRET {CAT}   TOP SECRET {DOG}**
  - **TOP SECRET {DOG}   TOP SECRET {CAT}**
- *Compartments designed to enforce the* ***need to know*** *principle*
  - *Regardless of clearance, you only have access to info that you need to know to do your job*

Compartments

- *Arrows indicate “ ” relationship*



- *Not all classifications are comparable, e.g.,*

TOP SECRET {CAT} *vs* **SECRET {CAT, DOG}**

MLS vs Compartments

- MLS can be used without compartments
  - And vice-versa
- But, MLS almost always uses compartments
- Example
  - MLS mandated for protecting medical records of British Medical Association (BMA)
  - AIDS was **TOP SECRET**, prescriptions **SECRET**
  - What is the classification of an AIDS drug?
  - Everything tends toward **TOP SECRET**
  - Defeats the purpose of the system!
  - Compartments-only approach used instead