Chapter 14: Identity

- What is identity
- Multiple names for one thing
- Different contexts, environments
- Pseudonymity and anonymity

Overview

- Files and objects
- Users, groups, and roles
- Certificates and names
- Hosts and domains
- State and cookies
- Anonymity

Identity

- *Principal*: a unique entity
- *Identity*: specifies a principal
- *Authentication*: binding of a principal to a representation of identity internal to the system
  - All access, resource allocation decisions assume binding is correct

Files and Objects

- Identity depends on system containing object
- Different names for one object
  - Human use, *eg*. file name
  - Process use, *eg*. file descriptor or handle
  - Kernel use, *eg*. file allocation table entry, inode

More Names

- Different names for one context
  - Human: aliases, relative *vs*. absolute path names
  - Kernel: deleting a file identified by name can mean two things:
    - Delete the object that the name identifies
    - Delete the name given, and do not delete actual object until *all* names have been deleted
- Semantics of names may differ

Example: Names and Descriptors

- Interpretation of UNIX file name
  - Kernel maps name into an inode using iterative procedure
  - Same name can refer to different objects at different times without being deallocated
    - Causes race conditions
- Interpretation of UNIX file descriptor
  - Refers to a specific inode
  - Refers to same inode from creation to deallocation

Example: Different Systems

- Object name must encode location or pointer to location
  - *rsh*, *ssh* style: *host*:*object*
  - URLs: *protocol*://*host*/*object*
- Need not name actual object
  - *rsh*, *ssh* style may name pointer (link) to actual object
  - URL may forward to another host

Users

- Exact representation tied to system
- Example: UNIX systems
  - Login name: used to log in to system
    - Logging usually uses this name
  - User identification number (UID): unique integer assigned to user
    - Kernel uses UID to identify users
    - One UID per login name, but multiple login names may have a common UID

Multiple Identities

- UNIX systems again
  - Real UID: user identity at login, but changeable
  - Effective UID: user identity used for access control
    - Setuid changes effective UID
  - Saved UID: UID before last change of UID
    - Used to implement least privilege
    - Work with privileges, drop them, reclaim them later
  - Audit/Login UID: user identity used to track original UID
    - Cannot be altered; used to tie actions to login identity

Groups

- Used to share access privileges
- First model: alias for set of principals
  - Processes assigned to groups
  - Processes stay in those groups for their lifetime
- Second model: principals can change groups
  - Rights due to old group discarded; rights due to new group added

Roles

- Group with membership tied to function
  - Rights given are consistent with rights needed to perform function
- Uses second model of groups
- Example: DG/UX
  - User *root* does not have administration functionality
  - System administrator privileges are in *sysadmin* role
  - Network administration privileges are in *netadmin* role
  - Users can assume either role as needed

Naming and Certificates

- Certificates issued to a principal
  - Principal uniquely identified to avoid confusion
- Problem: names may be ambiguous
  - Does the name "Matt Bishop" refer to:
    - The author of this book?
    - A programmer in Australia?
    - A stock car driver in Muncie, Indiana?
    - Someone else who was named "Matt Bishop"

Disambiguating Identity

- Include ancillary information in names
  - Enough to identify principal uniquely
  - X.509v3 Distinguished Names do this
- Example: X.509v3 Distinguished Names
  - /O=University of California/OU=Davis campus/OU=Department of Computer Science/CN=Matt Bishop/

    refers to the Matt Bishop (CN is *common name*) in the Department of Computer Science (OU is *organizational unit*) on the Davis Campus of the University of California (O is *organization*)

CAs and Policies

- Matt Bishop wants a certificate from Certs-from-Us
  - How does Certs-from-Us know this is "Matt Bishop"?
    - CA's *authentication policy* says what type and strength of authentication is needed to identify Matt Bishop to satisfy the CA that this is, in fact, Matt Bishop
  - Will Certs-from-Us issue this "Matt Bishop" a certificate once he is suitably authenticated?
    - CA's *issuance policy* says to which principals the CA will issue certificates

Example: Verisign CAs

- Class 1 CA issued certificates to individuals
  - Authenticated principal by email address
    - Idea: certificate used for sending, receiving email with various security services at that address
- Class 2 CA issued certificates to individuals
  - Authenticated by verifying user-supplied real name and address through an online database
    - Idea: certificate used for online purchasing

Example: Verisign CAs

- Class 3 CA issued certificates to individuals
  - Authenticated by background check from investigative service
    - Idea: higher level of assurance of identity than Class 1 and Class 2 CAs
- Fourth CA issued certificates to web servers
  - Same authentication policy as Class 3 CA
    - Idea: consumers using these sites had high degree of assurance the web site was not spoofed

Internet Certification Hierarchy

- Tree structured arrangement of CAs
  - Root is *Internet Policy Registration Authority*, or IPRA
    - Sets policies all subordinate CAs must follow
    - Certifies subordinate CAs (called *policy certification authorities*, or PCAs), each of which has own authentication, issuance policies
    - Does not issue certificates to individuals or organizations other than subordinate CAs
  - PCAs issue certificates to ordinary CAs
    - Does not issue certificates to individuals or organizations other than subordinate CAs
  - CAs issue certificates to organizations or individuals

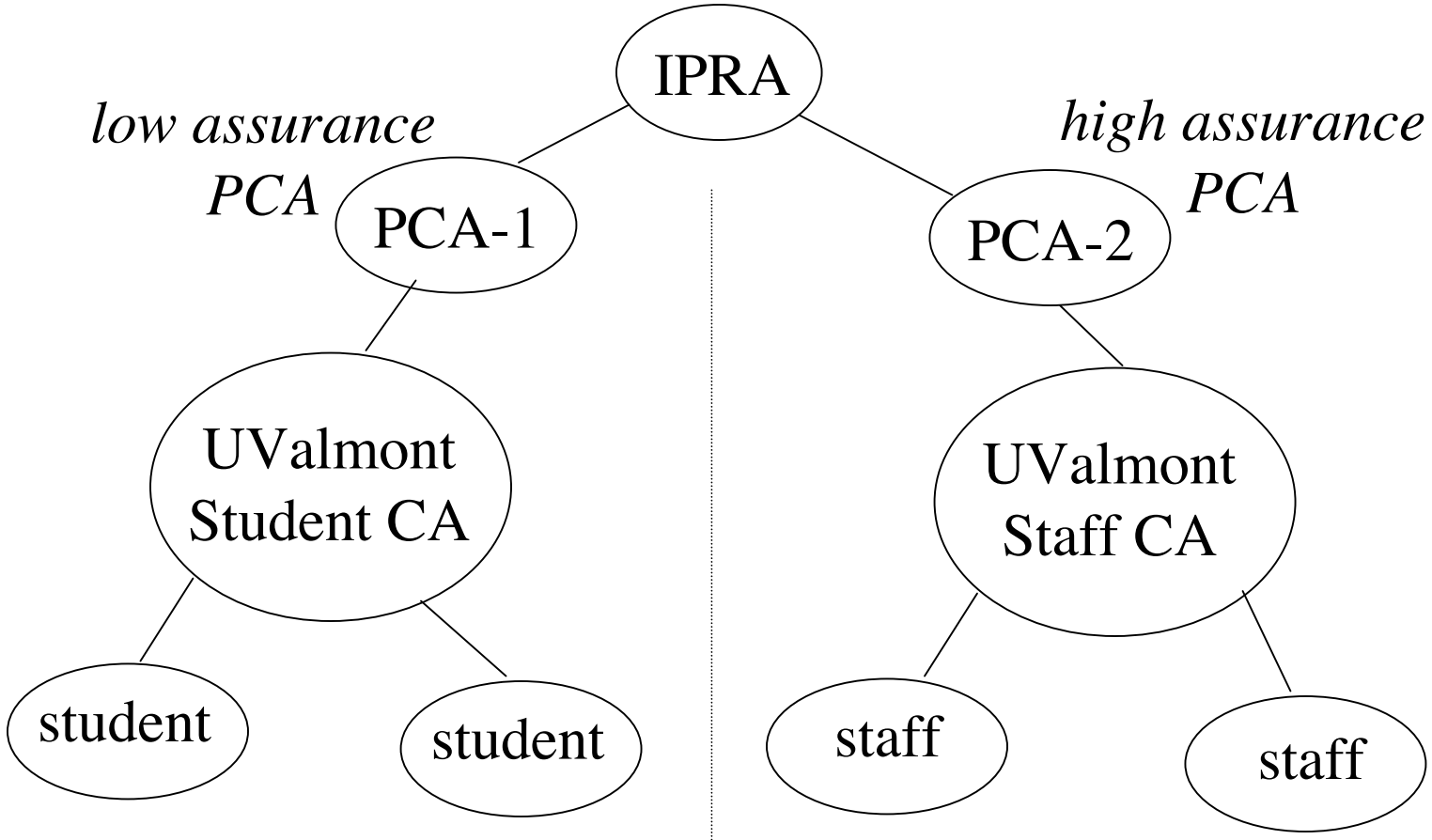
Example

- University of Valmont issues certificates to students, staff
  - Students must present valid reg cards (considered low assurance)
  - Staff must present proof of employment and fingerprints, which are compared to those taken when staff member hired (considered high assurance)

UValmont and PCAs

- First PCA: requires subordinate CAs to make good-faith effort to verify identities of principals to whom it issues certificates
  - Student authentication requirements meet this
- Second PCA: requires use of biometrics to verify identity
  - Student authentication requirements do not meet this
  - Staff authentication requirements do meet this
- UValmont establishes to CAs, one under each PCA above

UValmont and Certification Hierarchy



Certificate Differences

- Student, staff certificates signed using different private keys (for different CAs)
  - Student's signed by key corresponding to low assurance certificate signed by first PCA
  - Staff's signed by key corresponding to high assurance certificate signed by second PCA
- To see what policy used to authenticate:
  - Determine CA signing certificate, check its policy
  - Also go to PCA that signed CA's certificate
    - CAs are restricted by PCA's policy, but CA can restrict itself further

Types of Certificates

- Organizational certificate
  - Issued based on principal's affiliation with organization
  - Example Distinguished Name

    /O=University of Valmont/OU=Computer Science Department/CN=Marsha Merteuille/
- Residential certificate
  - Issued based on where principal lives
  - No affiliation with organization implied
  - Example Distinguished Name

    /C=US/SP=Louisiana/L=Valmont/PA=1 Express Way/CN=Marsha Merteuille/

Certificates for Roles

- Certificate tied to a role
- Example
  - UValmont wants comptroller to have a certificate
    - This way, she can sign contracts and documents digitally
  - Distinguished Name

    /O=University of Valmont/OU=Office of the Big Bucks/RN=Comptroller

    where "RN" is *role name*; note the individual using the certificate is not named, so no CN

Meaning of Identity

- Authentication validates identity
  - CA specifies type of authentication
  - If incorrect, CA may misidentify entity unintentionally
- Certificate binds *external* identity to crypto key and Distinguished Name
  - Need confidentiality, integrity, anonymity
    - Recipient knows same entity sent all messages, but *not* who that entity is

Persona Certificate

- Certificate with meaningless Distinguished Name
  - If DN is

    /C=US/O=Microsoft Corp./CN=Bill Gates/

    the real subject may not (or may) be Mr. Gates
  - Issued by CAs with persona policies under a PCA with policy that supports this
- PGP certificates can use any name, so provide this implicitly

Example

- Government requires all citizens with gene X to register
  - Anecdotal evidence people with this gene become criminals with probability 0.5.
  - Law to be made quietly, as no scientific evidence supports this, and government wants no civil rights fuss
- Government employee wants to alert media
  - Government will deny plan, change approach
  - Government employee will be fired, prosecuted
- Must notify media anonymously

Example

- Employee gets persona certificate, sends copy of plan to media
  - Media knows message unchanged during transit, but not who sent it
  - Government denies plan, changes it
- Employee sends copy of new plan signed using same certificate
  - Media can tell it's from original whistleblower
  - Media cannot track back whom that whistleblower is

Trust

- Goal of certificate: bind correct identity to DN
- Question: what is degree of assurance?
- X.509v3, certificate hierarchy
  - Depends on policy of CA issuing certificate
  - Depends on how well CA follows that policy
  - Depends on how easy the required authentication can be spoofed
- Really, estimate based on the above factors

Example: Passport Required

- DN has name on passport, number and issuer of passport
- What are points of trust?
  - Passport not forged and name on it not altered
  - Passport issued to person named in passport
  - Person presenting passport is person to whom it was issued
  - CA has checked passport and individual using passport

PGP Certificates

- Level of trust in signature field
- Four levels
  - Generic (no trust assertions made)
  - Persona (no verification)
  - Casual (some verification)
  - Positive (substantial verification)
- What do these mean?
  - Meaning not given by OpenPGP standard
  - Signer determines what level to use
  - Casual to one signer may be positive to another

Identity on the Web

- Host identity
  - Static identifiers: do not change over time
  - Dynamic identifiers: changes as a result of an event or the passing of time
- State and Cookies
- Anonymity
  - Anonymous email
  - Anonymity: good or bad?

Host Identity

- Bound up to networking
  - Not connected: pick any name
  - Connected: one or more names depending on interfaces, network structure, context
- *Name* identifies principal
- *Address* identifies location of principal
  - May be virtual location (network segment) as opposed to physical location (room 222)

Example

- Layered network
  - MAC layer
    - Ethernet address: 00:05:02:6B:A8:21
    - AppleTalk address: network 51, node 235
  - Network layer
    - IP address: 192.168.35.89
  - Transport layer
    - Host name: cherry.orchard.chekhov.ru

Danger!

- Attacker spoofs identity of another host
  - Protocols at, above the identity being spoofed will fail
  - They rely on spoofed, and hence faulty, information
- Example: spoof IP address, mapping between host names and IP addresses

Domain Name Server

- Maps transport identifiers (host names) to network identifiers (host addresses)
  - Forward records: host names → IP addresses
  - Reverse records: IP addresses → host names
- Weak authentication
  - Not cryptographically based
  - Various techniques used, such as reverse domain name lookup

Reverse Domain Name Lookup

- Validate identity of peer (host) name
  - Get IP address of peer
  - Get associated host name via DNS
  - Get IP addresses associated with host name from DNS
  - If first IP address in this set, accept name as correct; otherwise, reject as spoofed
- If DNS corrupted, this won't work

Dynamic Identifiers

- Assigned to principals for a limited time
  - Server maintains pool of identifiers
  - Client contacts server using *local identifier*
    - Only client, server need to know this identifier
  - Server sends client *global identifier*
    - Client uses global identifier in other contexts, for example to talk to other hosts
    - Server notifies intermediate hosts of new client, global identifier association

Example: DHCP

- DHCP server has pool of IP addresses
- Laptop sends DHCP server its MAC address, requests IP address
  - MAC address is local identifier
  - IP address is global identifier
- DHCP server sends unused IP address
  - Also notifies infrastructure systems of the association between laptop and IP address
- Laptop accepts IP address, uses that to communicate with hosts other than server

Example: Gateways

- Laptop wants to access host on another network
  - Laptop's address is 10.1.3.241
- Gateway assigns legitimate address to internal address
  - Say IP address is 101.43.21.241
  - Gateway rewrites all outgoing, incoming packets appropriately
  - Invisible to both laptop, remote peer
- Internet protocol NAT works this way

Weak Authentication

- Static: host/name binding fixed over time
- Dynamic: host/name binding varies over time
  - Must update reverse records in DNS
    - Otherwise, the reverse lookup technique fails
  - Cannot rely on binding remaining fixed unless you know the period of time over which the binding persists

DNS Security Issues

- Trust is that name/IP address binding is correct
- Goal of attacker: associate incorrectly an IP address with a host name
  - Assume attacker controls name server, or can intercept queries and send responses

Attacks

- Change records on server
- Add extra record to response, giving incorrect name/IP address association
  - Called “cache poisoning”
- Attacker sends victim request that must be resolved by asking attacker
  - Attacker responds with answer plus two records for address spoofing (1 forward, 1 reverse)
  - Called “ask me”

Cookies

- Token containing information about state of transaction on network
  - Usual use: refers to state of interaction between web browser, client
  - Idea is to minimize storage requirements of servers, and put information on clients
- Client sends cookies to server

Some Fields in Cookies

- *name*, *value*: name has given value
- *expires*: how long cookie valid
  - Expired cookies discarded, not sent to server
  - If omitted, cookie deleted at end of session
- *domain*: domain for which cookie intended
  - Consists of last *n* fields of domain name of server
  - *Must* have at least one "." in it
- *secure*: send only over secured (SSL, HTTPS) connection

Example

- Caroline puts 2 books in shopping cartcart at books.com
  - Cookie: *name* bought, *value* BK=234&BK=8753, *domain* .books.com
- Caroline looks at other books, but decides to buy only those
  - She goes to the purchase page to order them
- Server requests cookie, gets above
  - From cookie, determines books in shopping cart

Who Can Get the Cookies?

- Web browser can send *any* cookie to a web server
  - Even if the cookie's domain does not match that of the web server
  - Usually controlled by browser settings
- Web server can *only* request cookies for its domain
  - Cookies need not have been sent by that browser

Where Did the Visitor Go?

- Server books.com sends Caroline 2 cookies
  - First described earlier
  - Second has *name* "id", *value* "books.com", *domain* "adv.com"
- Advertisements at books.com include some from site adv.com
  - When drawing page, Caroline's browser requests content for ads from server "adv.com"
  - Server requests cookies from Caroline's browser
  - By looking at *value*, server can tell Caroline visited "books.com"

Anonymity on the Web

- Recipients can determine origin of incoming packet
  - Sometimes not desirable
- Anonymizer: a site that hides origins of connections
  - Usually a proxy server
    - User connects to anonymizer, tells it destination
    - Anonymizer makes connection, sends traffic in both directions
  - Destination host sees only anonymizer

Example: *anon.penet.fi*

- Offered anonymous email service
  - Sender sends letter to it, naming another destination
  - Anonymizer strips headers, forwards message
    - Assigns an ID (say, 1234) to sender, records real sender and ID in database
    - Letter delivered as if from anon1234@anon.penet.fi
  - Recipient replies to that address
    - Anonymizer strips headers, forwards message as indicated by database entry

Problem

- Anonymizer knows who sender, recipient *really* are
- Called *pseudo-anonymous remailer* or *pseudonymous remailer*
  - Keeps mappings of anonymous identities and associated identities
- If you can get the mappings, you can figure out who sent what

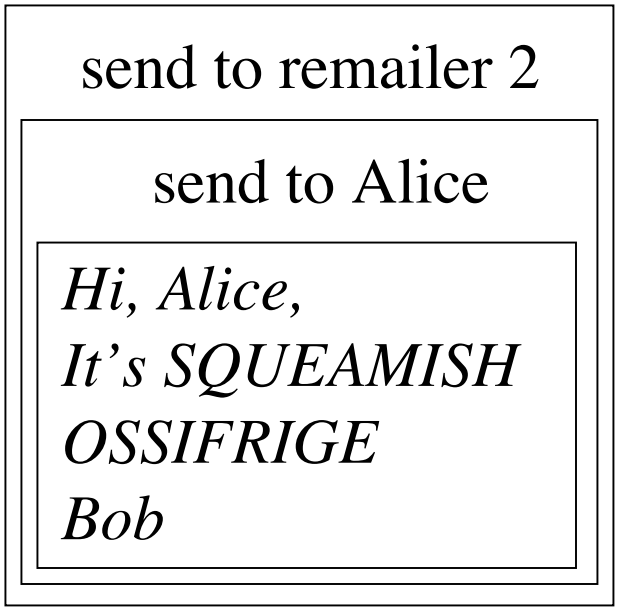
More *anon.penet.fi*

- Material claimed to be copyrighted sent through site
- Finnish court directed owner to reveal mapping so plaintiffs could determine sender
- Owner appealed, subsequently shut down site

Cypherpunk Remailer

- Remailer that deletes header of incoming message, forwards body to destination
- Also called *Type I Remailer*
- No record kept of association between sender address, remailer's user name
  - Prevents tracing, as happened with *anon.penet.fi*
- Usually used in a chain, to obfuscate trail
  - For privacy, body of message may be enciphered

Cypherpunk Remailer Message



- Encipher message
- Add destination header
- Add header for remailer *n*

  …
- Add header for remailer 2

Weaknesses

- Attacker monitoring entire network
  - Observes in, out flows of remailers
  - Goal is to associate incoming, outgoing messages
- If messages are cleartext, trivial
  - So assume all messages enciphered
- So use traffic analysis!
  - Used to determine information based simply on movement of messages (traffic) around the network

Attacks

- If remailer forwards message before next message arrives, attacker can match them up
  - Hold messages for some period of time, greater than the message interarrival time
  - Randomize order of sending messages, waiting until at least *n* messages are ready to be forwarded
    - Note: attacker can force this by sending *n*–1 messages into queue

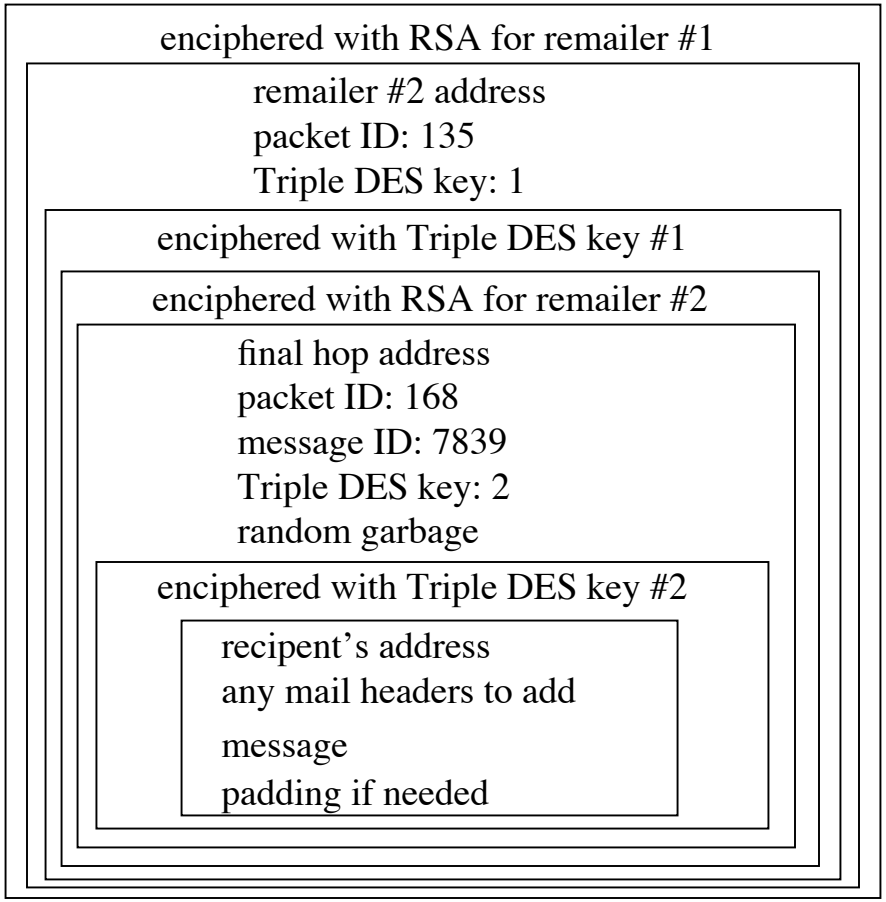
Attacks

- As messages forwarded, headers stripped so message size decreases
  - Pad message with garbage at each step, instructing next remailer to discard it
- Replay message, watch for spikes in outgoing traffic
  - Remailer can't forward same message more than once

Mixmaster Remailer

- Cypherpunk remailer that handles only enciphered mail and pads (or fragments) messages to fixed size before sending them
  - Also called Type II Remailer
  - Designed to hinder attacks on Cypherpunk remailers
    - Messages uniquely numbered
    - Fragments reassembled *only* at last remailer for sending to recipient

Cypherpunk Remailer Message



Anonymity Itself

- Some purposes for anonymity
  - Removes personalities from debate
  - With appropriate choice of pseudonym, shapes course of debate by implication
  - Prevents retaliation
- Are these benefits or drawbacks?
  - Depends on society, and who is involved

Privacy

- Anonymity protects privacy by obstructing amalgamation of individual records
- Important, because amalgamation poses 3 risks:
  - Incorrect conclusions from misinterpreted data
  - Harm from erroneous information
  - Not being let alone
- Also hinders monitoring to deter or prevent crime
- Conclusion: anonymity can be used for good or ill
  - Right to remain anonymous entails responsibility to use that right wisely

Key Points

- Identity specifies a principal (unique entity)
  - Same principal may have many different identities
    - Function (role)
    - Associated principals (group)
    - Individual (user/host)
  - These may vary with view of principal
    - Different names at each network layer, for example
  - Anonymity possible; may or may not be desirable
    - Power to remain anonymous includes responsibility to use that power wisely