Single Sign-On, Two Factor & more: Advanced Authentication & Authorization at the University of Pennsylvania

Shumon Huque & Deke Kassabian
University of Pennsylvania

Internet2 Fall Member Meeting
September 21st 2005, Philadelphia, PA

Historical

- PennNet Authentication System
  - Home grown
  - Not standards based, relied on custom network protocols
  - Reusable passwords transmitted in the clear
  - Not highly available
  - No Single Sign-On capability
- We needed something a lot better

New Requirements

- Standards based
- Cryptographic authentication
- Mutual authentication
- Single Sign-On
- High Availability
- Wide application support

Cryptographic Authentication

- No password or long term key is transferred over the network
- Users prove their identity to a service by performing a cryptographic operation,usually on a quantity (nonce) supplied by the server
- Crypto operation based on user's secret key or password

Single Sign-On (SSO)

- An authentication system
- Login or sign-on once (per time period)
- User is automatically authenticated to subsequent network services, without being prompted for his authentication credentials again (eg. password)

- SSO != password synchronization + caching

Why Single Sign-On?

- Convenience and security?
  - *Huh? You cannot be serious!*
- Convenience:
  - Users have a single password and only need to use it once (per day)
- Security (on true SSO systems):
  - Passwords on central authn server(s) only
    - Easier to defend a smaller set of computers
    - Centralized password quality enforcement
  - Users will (probably) be less cavalier about password security

Candidate Authentication Systems

- Kerberos
- Public Key Infrastructure (PKI)

Kerberos

- Standards based strong authentication system
- Authentication mediated by trusted 3rd party
  - Key Distribution Center (KDC)
- Uses secret key cryptography
- Provides mutual authentication
- Provides single sign-on capability
- Can optionally support:
  - Hardware tokens, smartcards, pubkey crypto
- Inter-domain authentication mechanisms exist

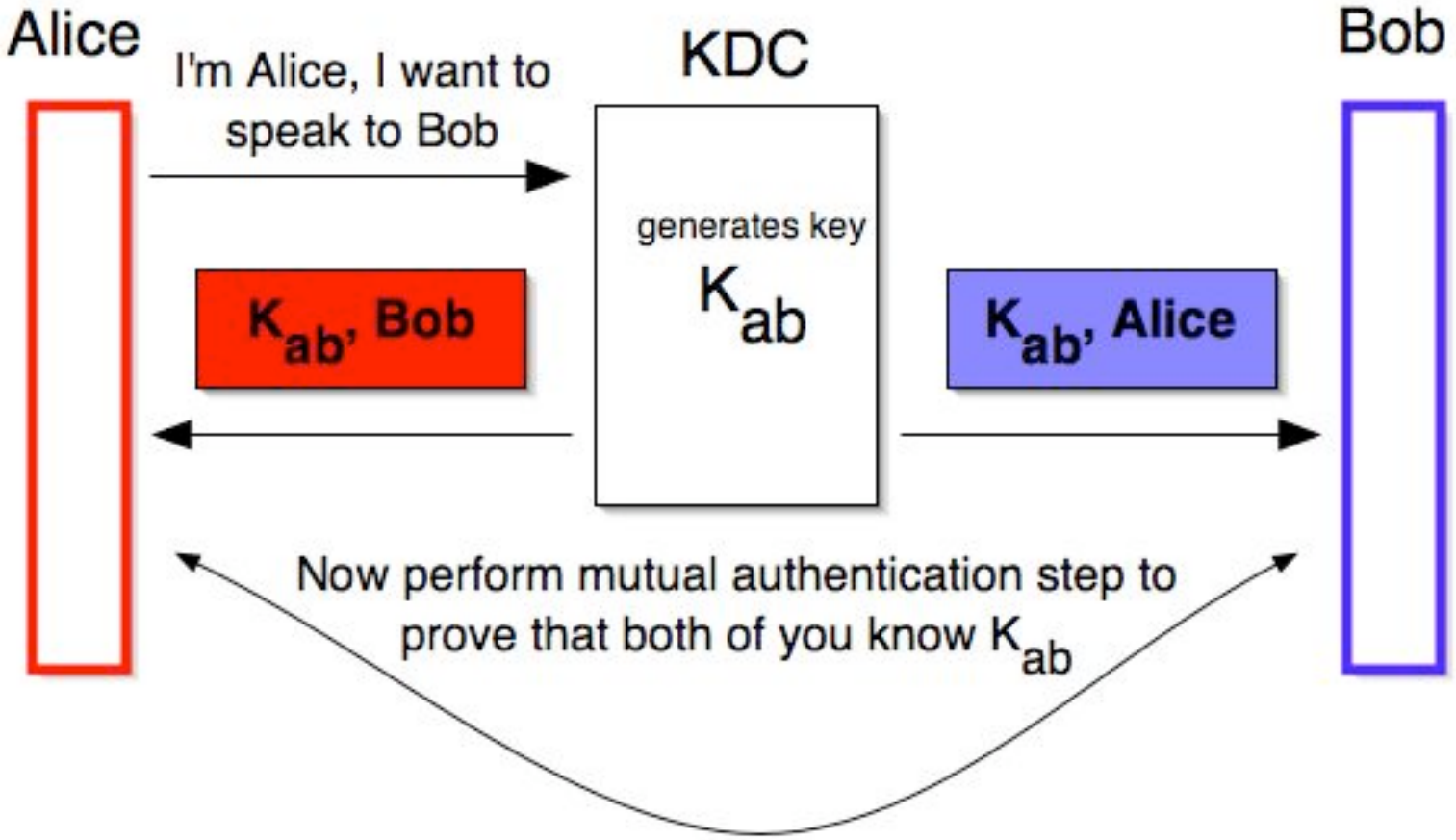
Kerberos (cont)

- Employs passwords
  - but they are never transmitted over the network
  - Other cryptographic credentials (tickets and authenticators) are sent over the network instead

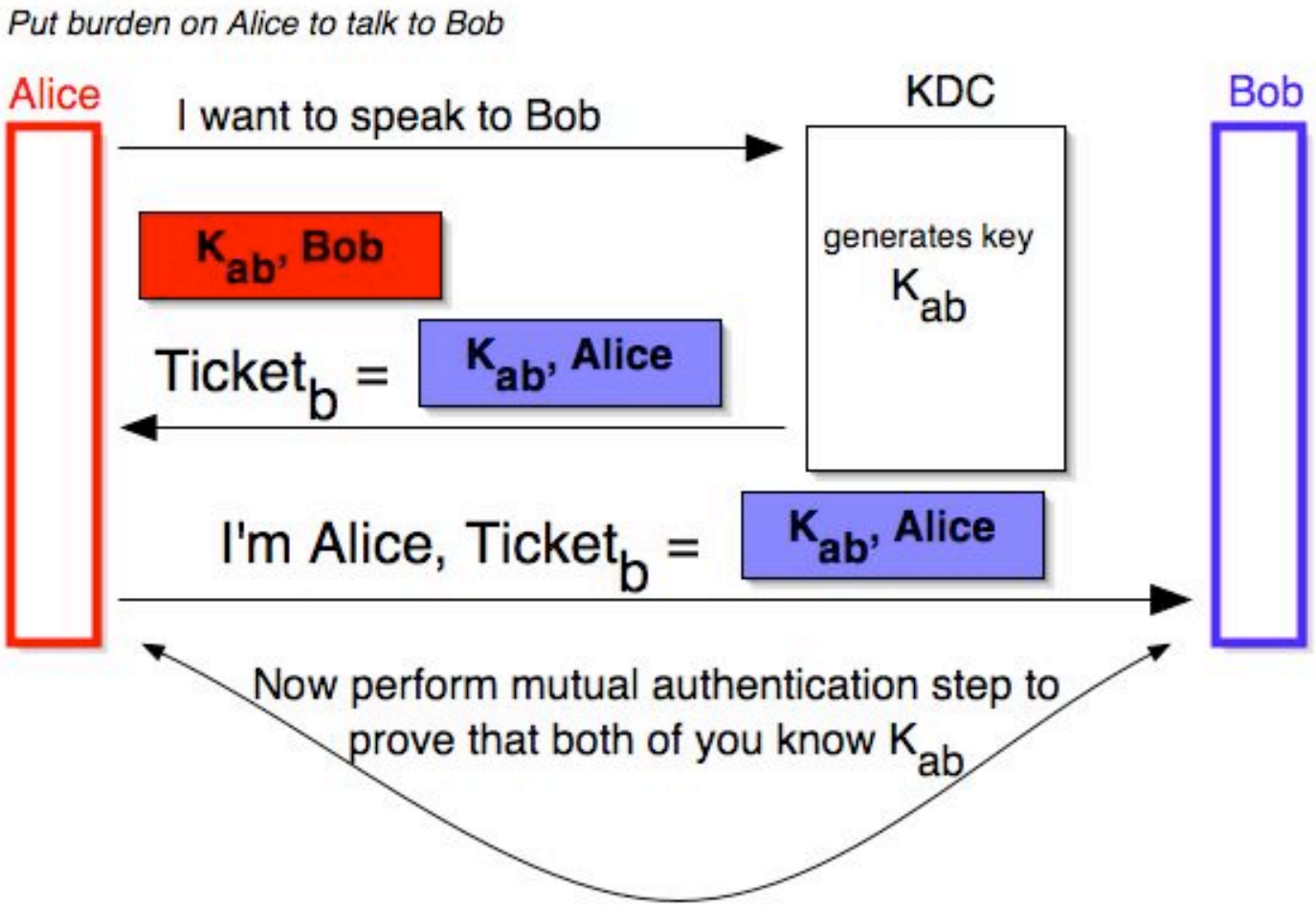
Mediated Authentication

- A trusted third party mediates the authentication process
  - Called the *Key Distribution Center* (KDC)
- Each user and service shares a secret key with the KDC
- KDC generates a session key, and securely distributes it to communicating parties
- Communicating parties prove to each other that they know the session key

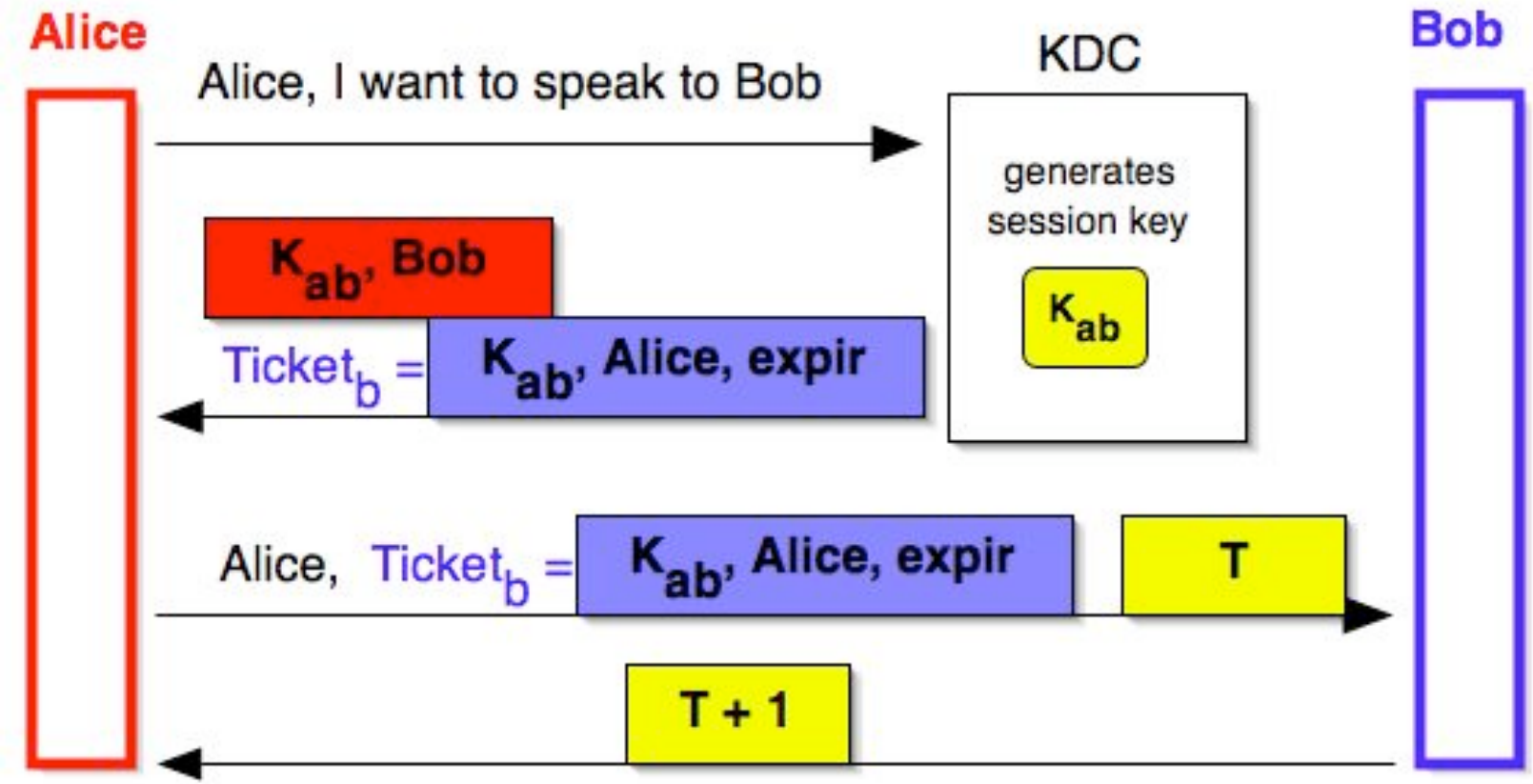
Mediated Authentication



Mediated Authentication

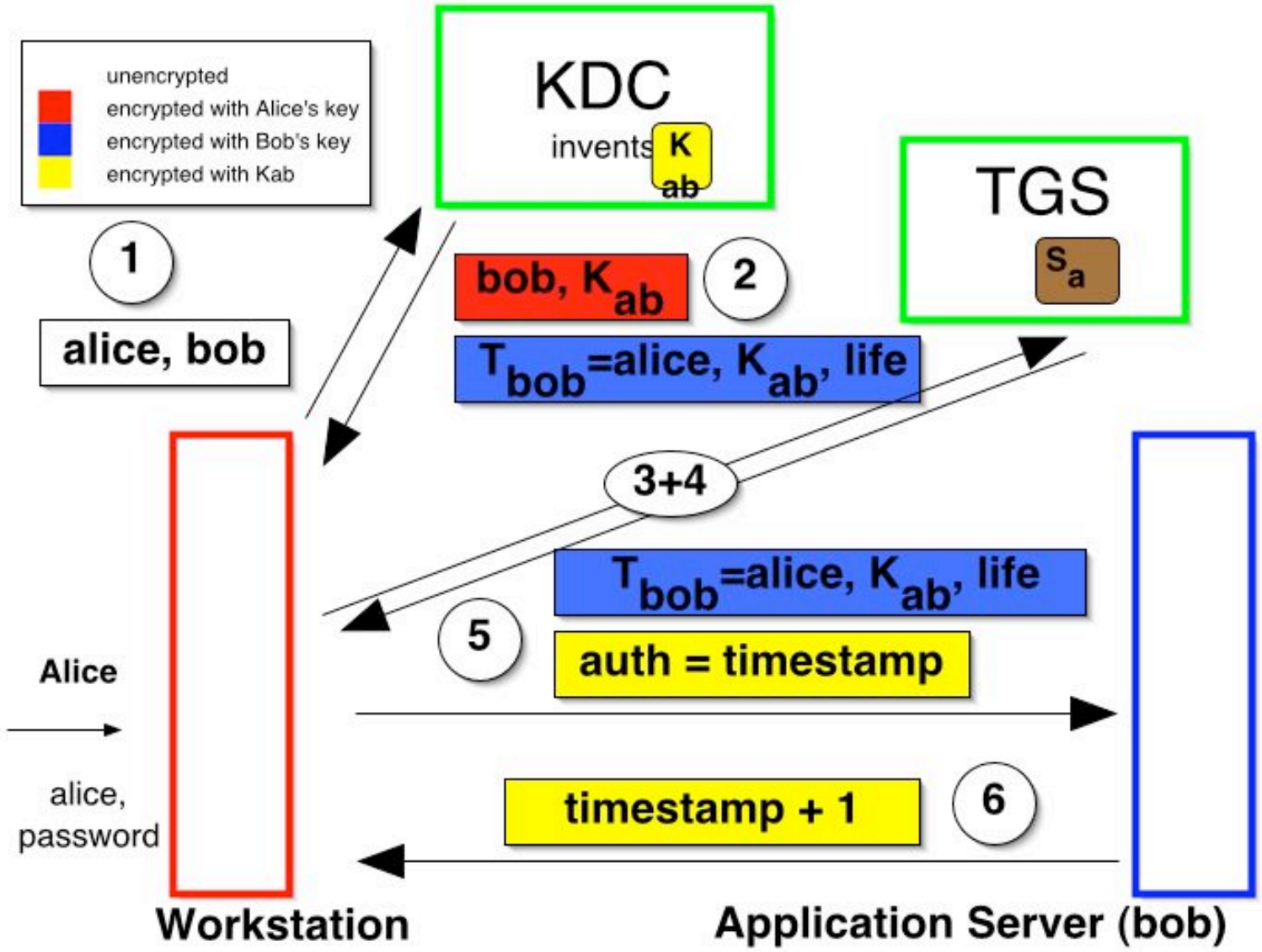


Kerberos (roughly)



Kerberos with Single Sign-On

- Ticket Granting Service (TGS):
  - A special Kerberos authenticated service, that allows user to obtain tickets for other services
  - Kerberos client software automatically obtains these tickets as needed
  - Co-located at the KDC
- Ticket Granting Ticket (TGT):
  - Ticket used to access the TGS and obtain service tickets
- Limited-lifetime session key: TGS sessionkey
  - Shared by user and the TGS


unencrypted
encrypted with Alice's key
encrypted with Bob's key
encrypted with Kab
KDC
invents K ab
TGS
S a
1
alice, bob
bob, K ab
2
T bob=alice, K ab, life
3+4
T bob=alice, K ab, life
5
auth = timestamp
Alice
alice,
password
timestamp + 1
6
Workstation
Application Server (bob)

Kerberos enabled applications

- Windows domain authentication
- E-mail (SMTP/POP/IMAP)
- File transfer (FTP, SCP)
- File sharing (NFS, DFS, Samba)
- Remote Login (TELNET, rlogin, SSH*)
- Directory (LDAP)

- Authen frameworks: SASL, GSS-API, TLS

Kerberos OS support

- Microsoft Windows
- Apple MacOS X
- Solaris, HP-UX, IBM AIX
- Linux, *BSD

Specific Applications

- Some applications where Penn has helped implement Kerberos support
  - Qualcomm's Eudora (POP/IMAP/SMTP)
  - Newswatcher (NNTP)
  - Mozilla/Thunderbird (POP/IMAP/SMTP - LDAP soon)

Kerberos devoid applications

- Some notable applications that don't yet have Kerberos support:
  - WWW (HTTP)
    - Workarounds exist; webiso systems like pubcookie, websec
    - KX.509 protocol from UMich
      - Combines Kerberos with short term PK credentials that are then used in SSL/TLS authentication

Kerberos devoid applications

- Attempts to support native Kerberos authentication in HTTP
  - Microsoft's HTTP/SPNEGO/GSS-API
    - Not standards based
    - No channel protection - easily victimized by session hijacking
    - SPNEGO protocol is being repaired
  - IETF efforts in progress
    - Kerberos/GSS-API ciphers in TLS
    - SASL in HTTP

Kerberos devoid applications

- EAP (used by IEEE 802.1x & PANA)
  - Awaiting EAP-GSS, EAP-Kerberos5 (SECMECH)
- IPSEC
  - IETF KINK protocol under development
- SNMP (Simple Network Management Protocol)
  - IETF ISMS working group

Kerberos devoid applications

- Interim measure:
- Deploy Kerberos password verification service
  - Receives Kerberos principal and password over some secure channel, then authenticates against KDC
  - We do this centrally with a RADIUS server infrastructure

Other issues

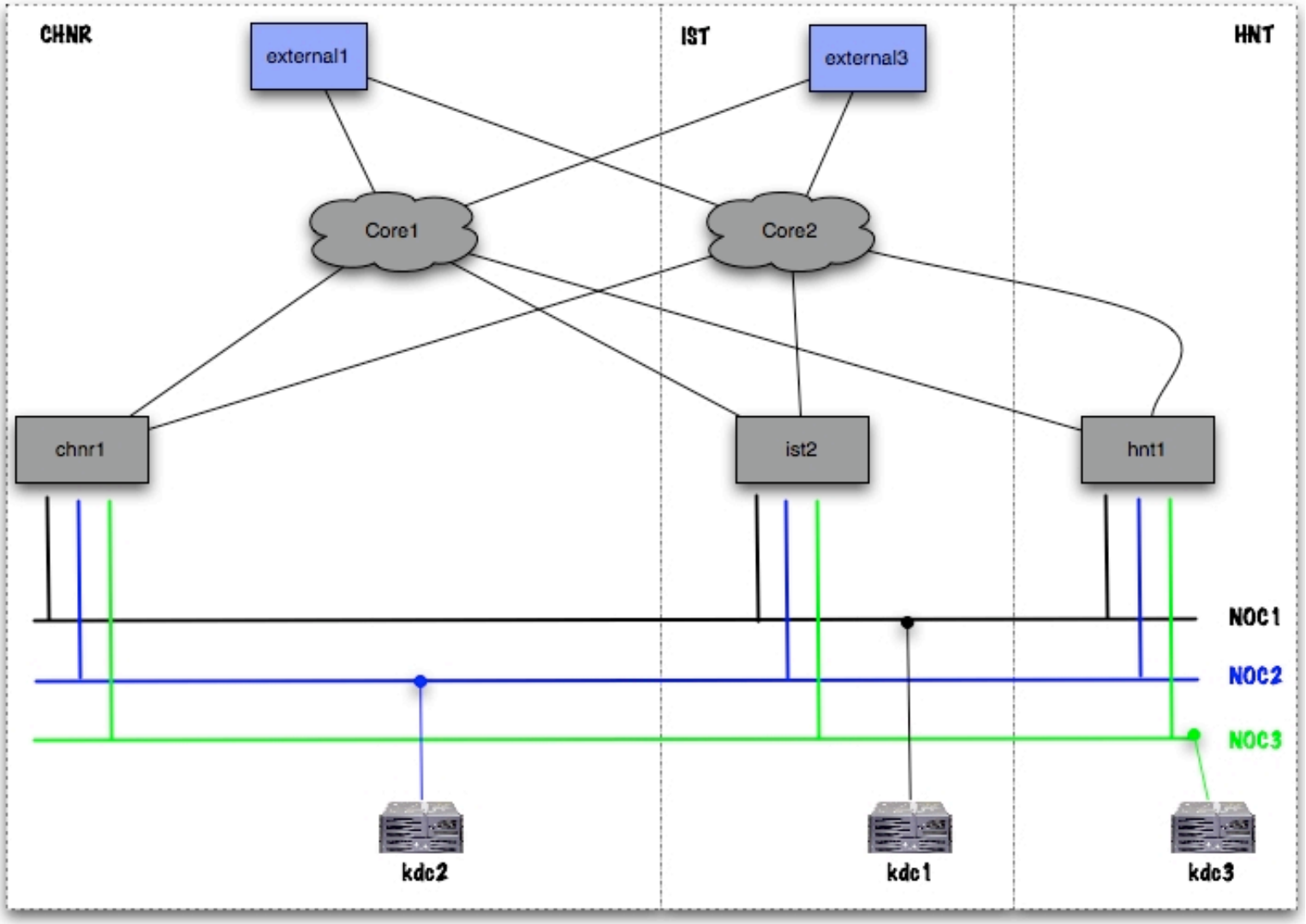
- Inter-institutional authentication
  - Federations
- Roaming scholar problem
  - See FWNA work

Kerberos caveats

- KDC is a single point of failure
  - Can have replicated KDC's
- KDC could be a performance bottleneck
  - Everyone needs to communicate with it frequently
  - Having multiple KDC's alleviates the problem
- If local workstation is compromised, user's password could be stolen by a trojan horse
  - Only use a desktop machine or laptop that you trust
  - Use hardware token pre-authentication
- AS exchange vulnerable to offline dictionary attack
  - Solution: Strong password rules, 2-factor authentication

Designing for High Availability

- Multiple Kerberos servers (3)
- Employ “failover” model
- Each KDC in a distinct machine room in a distinct geographic location
- Each on a distinct (logically & physically isolated) IP subnet
- Each IP subnet multihomed to 3 campus core routers



Central Infrastructure Networks design

Kerberos future?

- Make initial exchange invulnerable to dictionary attack:
  - EKE, SRP, SPEKE, SRP, PDM etc
  - *Problems: IPR issues*
  - PKINIT exists --> but needs PK credentials and possibly PKI
- Identity privacy
- Identifier remapping

Kerberos & Two-factor auth

- In addition to a secret password, user is required to present a physical item:
  - A small electronic device: h/w authentication token
  - Generates non-reusable numeric responses
  - Could employ challenge response
- Called 2-factor authentication, because it requires 2 things:
  - Something the user knows (password)
  - Something the user has (hardware token)


RSA SecurID®

Two-factor deployment

- Token technology selection
- Infrastructure setup
  - Authentication server infrastructure
    - Redundancy, high availability important
  - Manage, distribute, initialize tokens
  - Problem resolution: diagnosis & repair of faulty tokens
  - Money :-)

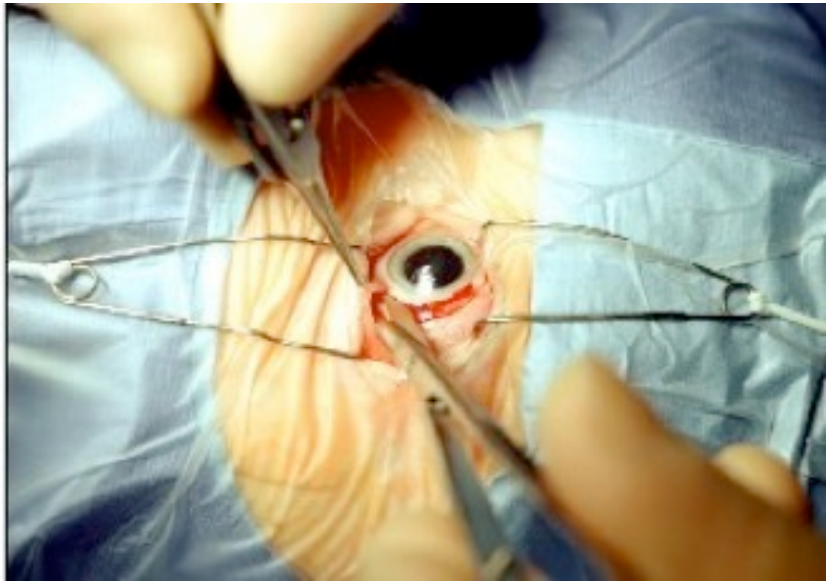
Are Biometrics the answer?

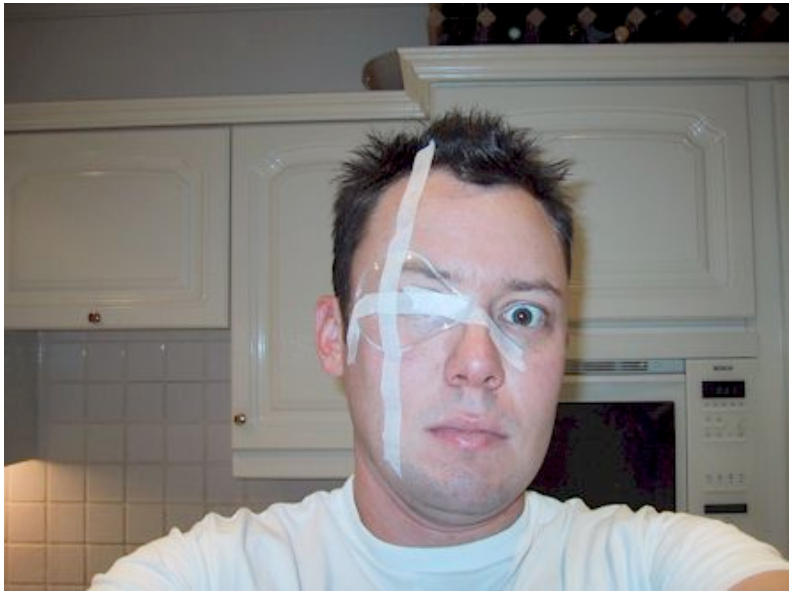
- Fingerprint, retina print, iris print etc
- Could be useful as an additional authentication factor, but ..

Are Biometrics the answer?

- Users may be reluctant to have biometric data stored in central databases
  - Privacy objections, linkage with health
- Reliability?
  - Biometric measurements noisy by nature
- Low level of secrecy
  - People leave fingerprints everywhere
  - Iris images may be captured by photography
- Irrevocable nature
  - How do you change a compromised biometric?

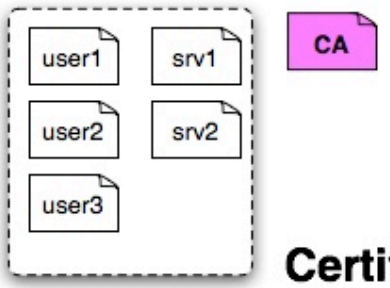
Revocation protocol :-)





Public Key Infrastructure (PKI)

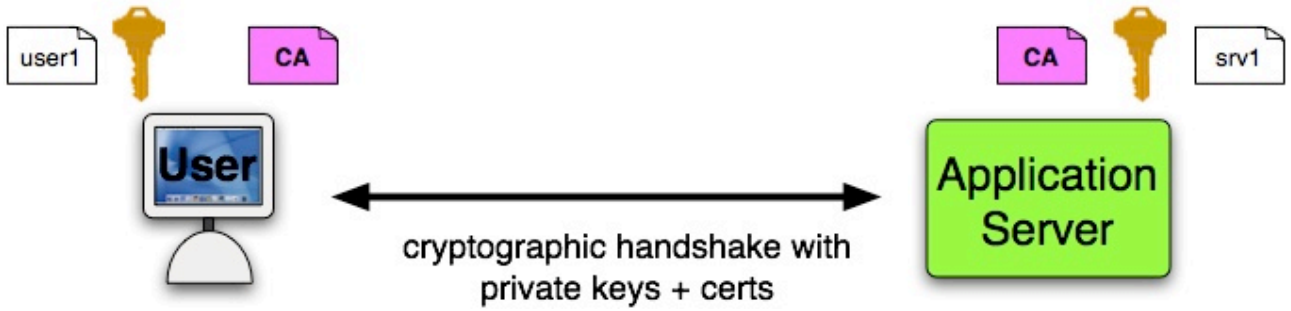
- A system for managing public keys & certificates
  - A Certification Authority (CA) or hierarchy of Certication Authorities
  - Protocols for key acquisition, validation, distribution and revocation.
  - The CA maintains directories of digitally signed associations of public keys and their owners.
  - IETF standards in progress:
    - PKIX (an X.509 derivative) and SPKI.



CA

Certification Authority

PKI based Authentication



PKI

- ***If used properly***, one of the most secure systems around (for now)
- Great scalability characteristics (some gotchas ..)

PKI outstanding issues

- Authentication of the CA or CA chain
- Protection of user's private keys
- Certificate Revocation
- Credential Mobility
- Single Sign-on issues
- Challenging user education problems

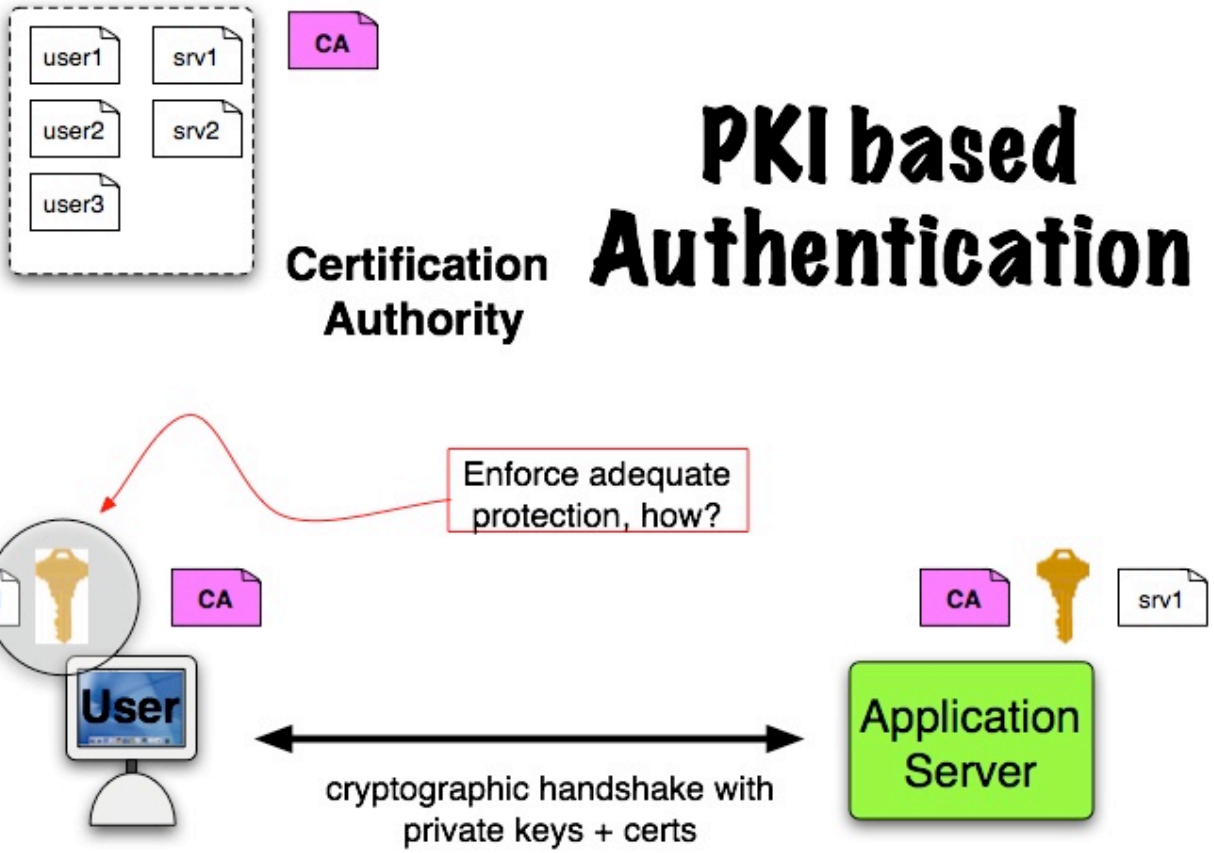
PKI issue 1

- How can the CA or CA chain be properly authenticated?
- Most protocols that employ PKI explicitly ignore this problem
- Software often comes initialized with root CA public keys
  - And hope that no-one ever encounters trojans or malware

PKI Issue 2

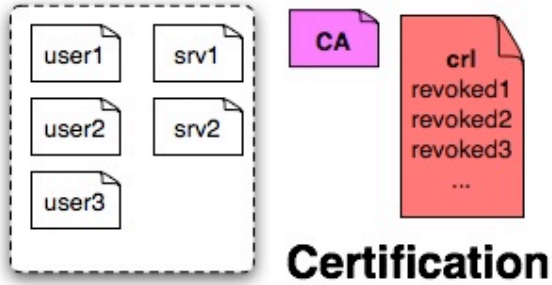
- How do we enforce adequate protection of a user's private key?

PKI based Authentication



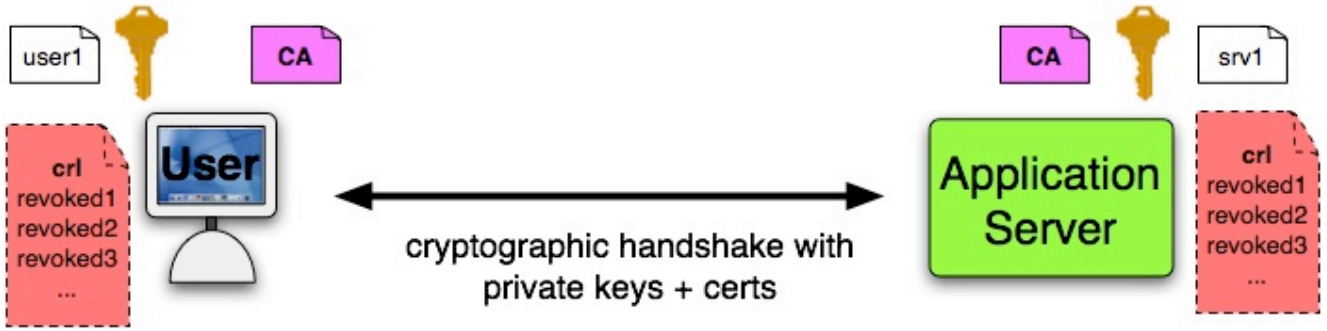
PKI issue 3

- Certificate Revocation
  - How do users and servers get up-to-date CRLs?
  - How does the system enforce that it's users are using up-to-date CRLs?



Certification Authority

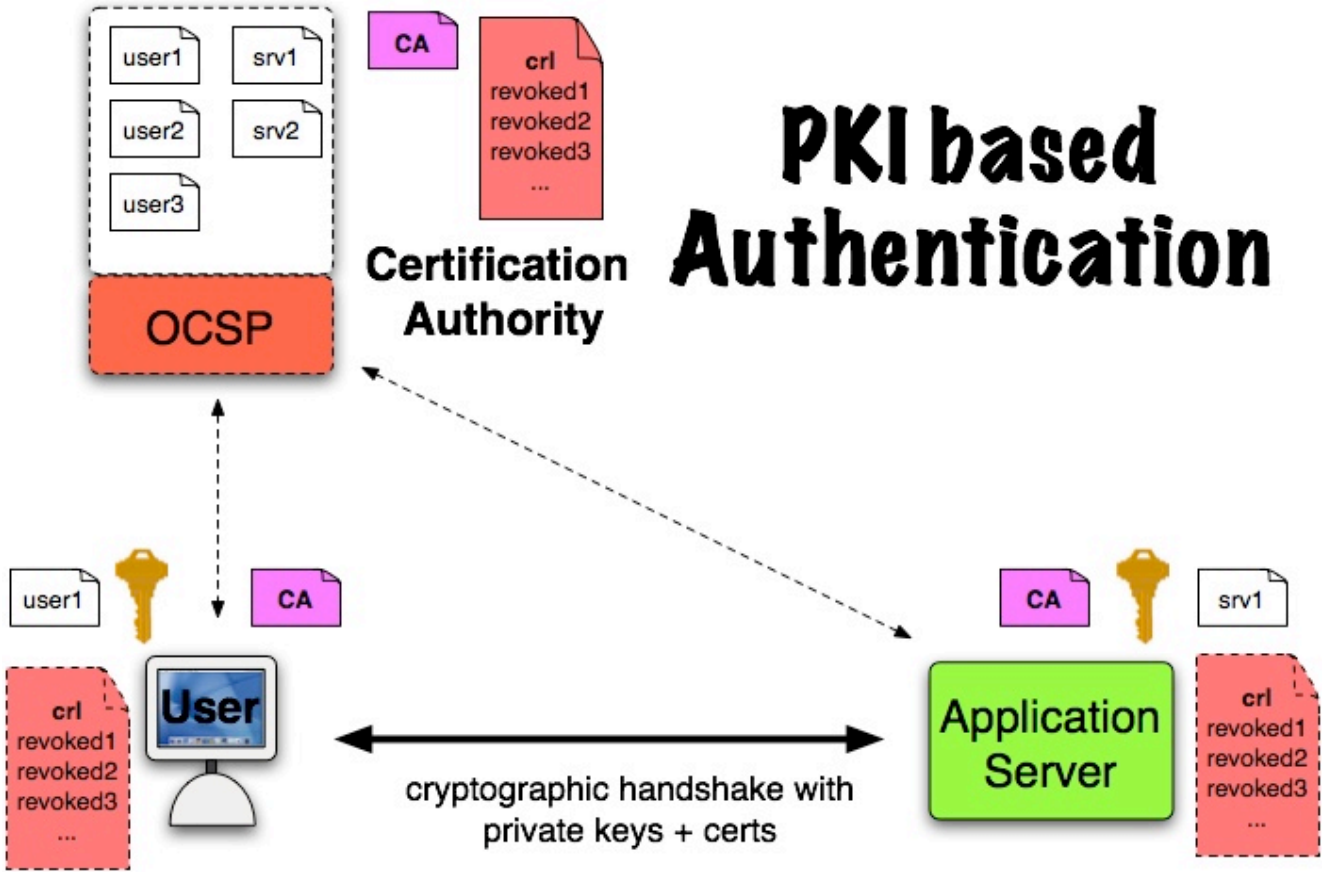
PKI based Authentication



PKI issue 3 (cont)

- Certificate revocation (cont)

- Online Certificate Status Protocol (OCSP)
  - Which deployments use OCSP?
  - Do they introduce a performance and reliability problem?


PKI based Authentication
user1
srv1
user2
srv2
user3
OCSP
CA
crl
revoked1
revoked2
revoked3
...
Certification Authority
user1
CA
crl
revoked1
revoked2
revoked3
...
User
CA
srv1
Application Server
crl
revoked1
revoked2
revoked3
...
cryptographic handshake with private keys + certs

PKI issue 4

- Credential Mobility
  - IETF SACRED working group
    - Various password based protocols for key retrieval
  - Smartcards and tokens for transport
    - Need token readers everywhere!

PKI issue 5

- No good Single Sign-On solution today
  - Long term exposure of private key is not good

PKI issue 6

- User Education
  - Public Key crypto places undue burden on users to rigorously validate keys and certificates
  - And provides no way to secure user's compliance in these tasks
  - Teaching unsavvy users about key management and key hygiene is difficult (probably impossible)

- Is PKI a good consumer technology?

Useful applications for PKI

- Server to server authentication
- Inter-institutional authentication
  - eg. Federated authentication systems like Shibboleth

- What about PKI only for managing server certificates?
  - More manageable than user AuthN but still has issues (see previous slides)

Do you *trust* your CA?

- “A CA can protect you from anyone they are not taking money from.”
  - - Matt Blaze

- January 2001: Verisign issued two Class-3 certificates to an unknown individual with the common name “Microsoft Corporation”

Speaker change ..

Unified Namespace

- Decided in 1995 to unify disjoint user namespaces at Penn
- Developed a basic name registry service (PennNames) and tools for applications
- Coordinated with application owners from throughout Penn
- Group effort to resolve name conflicts over the course of 6 or 7 years (fairly painful)

Why do we care about Unified Namespace?

- Reduces confusion and misdirected communications
- Provides a simpler handle for a broad range of campus IT services
- Simplified design of campus-wide authentication system
- Probably simplifies future work on centralized authorization

Authentication & Authorization

- The act of verifying someone's identity
- The process by which users prove their identity to a service
  - (and vice versa "Mutual authentication")

- Doesn't specify what a user is allowed or not allowed to do (Authorization)

What do we have so far?

- We “know” that the user is who they claim to be *(authentication)*
- We don’t know anything about them *(roles, affiliations)*
- We don’t know what they can do *(privileges)*

Simple Scenario

- “Hi! I’m Mark!” *(Identity)*
- “… And here is my PennKey and password to prove it.” *(Authentication)*
- “I want to connect to the IMAP server to read my mail.” *(Authorization)*
- “And now I want to shut down the DNS server.” *(Authorization)*

Authorization Decisions

- Is the user on a list of approved users?
- Is the user a member of an approved group?

The Not-So-Good Old Days

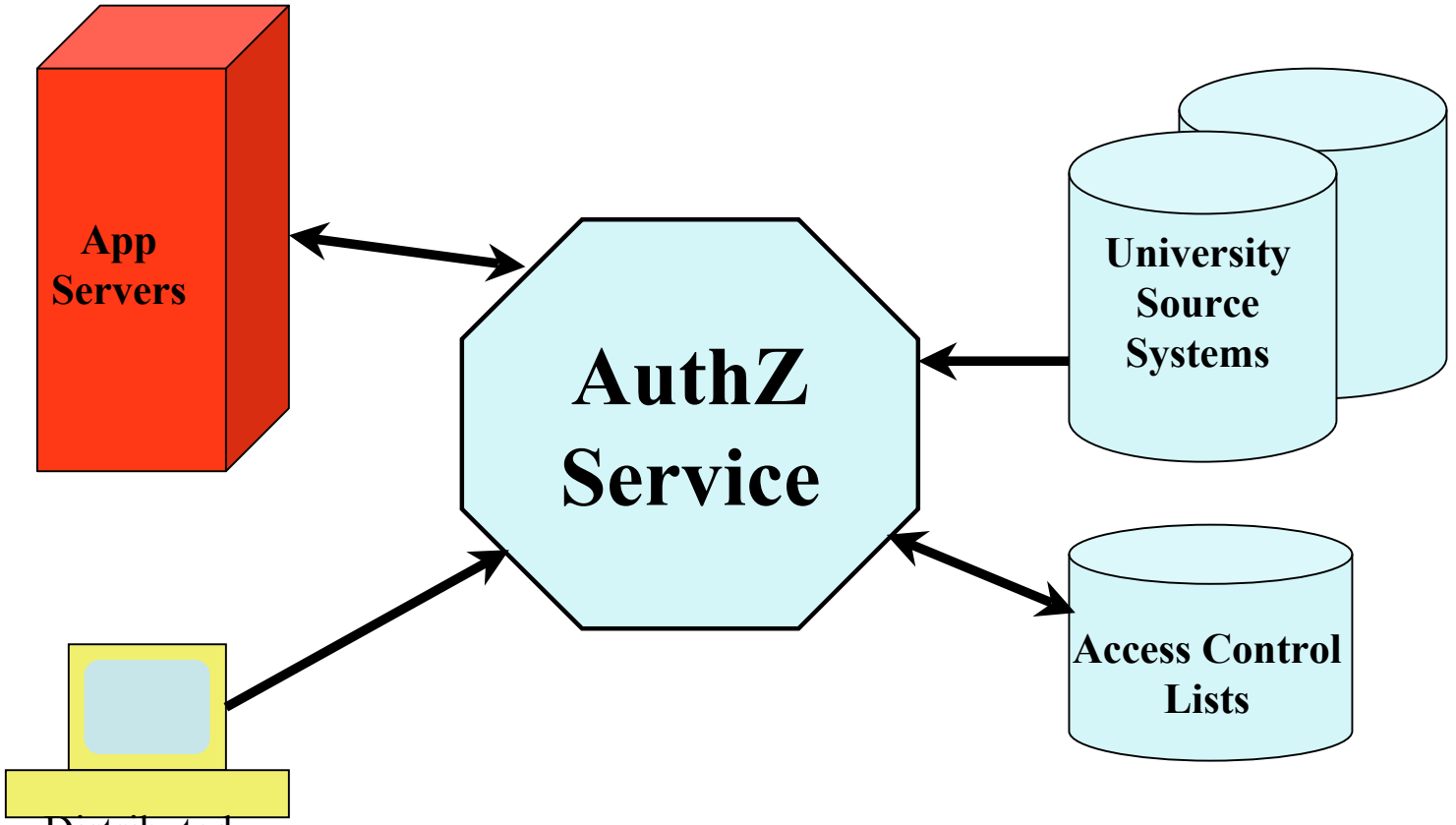
- Every application on its own to make authorization decisions
- In practice, many assumed that authentication was good enough ("if you can log in, you're in")
- Every application must maintain its own access control lists or eligibility/ privilege rules

A Better Way

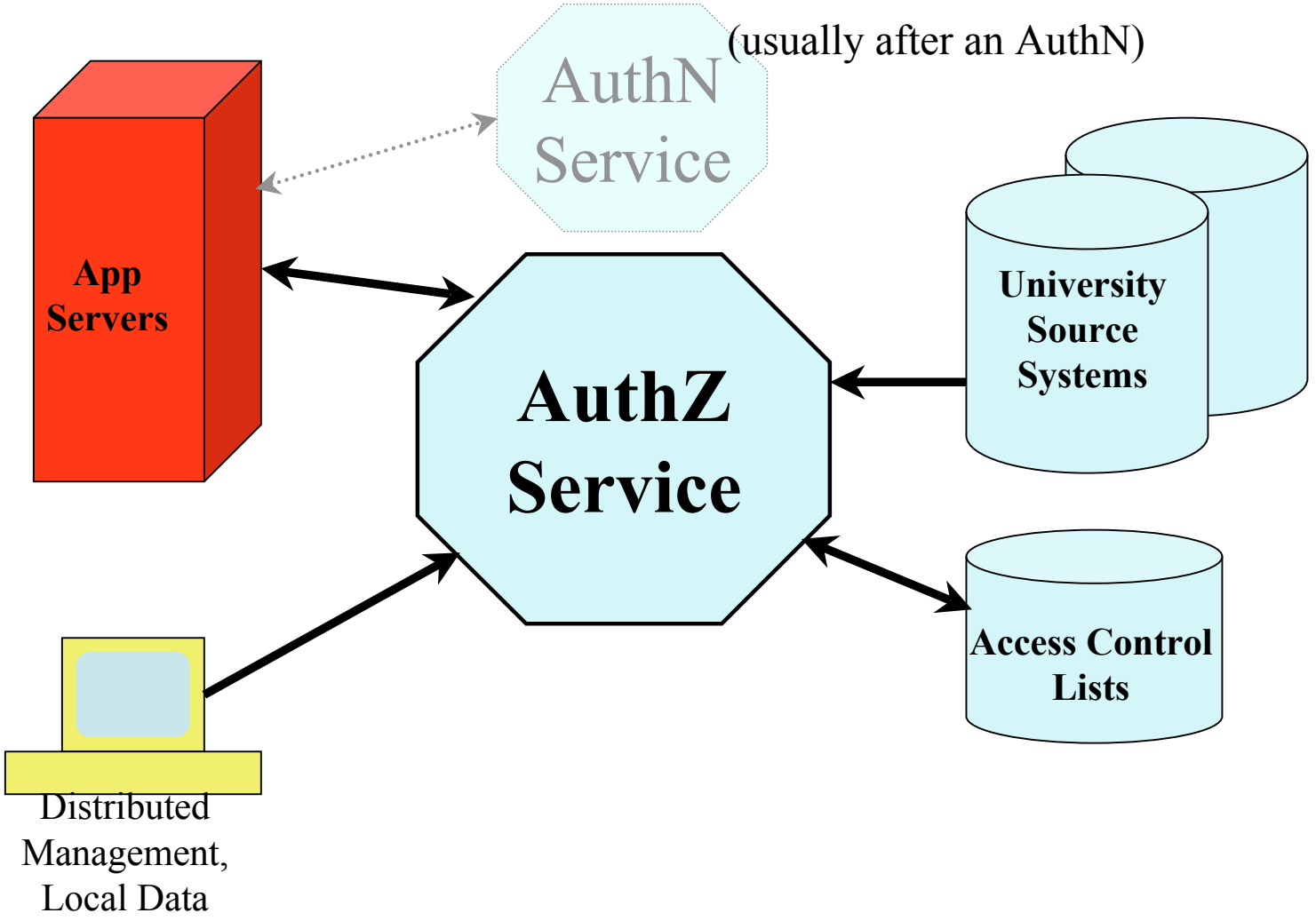
- Make authorization decisions according to *local* eligibility policy using *central* role and privilege definitions

- “All Senior Law Faculty”
- “Any staff in my department, except the birthday boy”

High Level AuthZ Design



High Level AuthZ Design



Likely Components

- Grouper and Signet as elements of the AuthZ service
- Web UI that allows distributed management of central store of local data
- Application access to the AuthZ service by widely available mechanisms/protocols like LDAP

Benefits of Centralization

- Consistent application of authority rules
- (Many) privileges for an individual can be viewed in one place
- Allows for a historical view of privileges over time
- Allows for automatic revocation based on status or affiliation changes
- Facilitates hierarchical control of authority

Making the case for centralization

- Stay in compliance with a growing list of policy mandates
  - Consistent rules
  - Easy auditing
- Save both dollars and time
  - Automated privilege changes
  - Less specific knowledge needed for every application

Challenges of centralization

- Sufficient motivation for change
- Users and application providers may need related education
- Resources, control
  - Centralized authentication forces units to relinquish control
  - Perhaps some software engineering required to separate authentication from authorization

Challenges of centralization

- Units must understand current authorization/privilege policies
- This will likely trigger a thorough review of those policies (probably not a bad thing, but takes time)
- Units must translate those policies into new format

Summing up

- Unified user name space (PennNames)
- Addressing several password issues (many passwords, varying rules, poor password handling practices) with central AuthN
- Driving towards secure and practical single signon through the native use of Kerberos
- Working on two-factor AuthN possibilities
- Pulling together relevant directory, AuthN, AuthZ technology pieces, plus policies, and physical identification, towards early stage Identity Management

References

- Kerberos: An Authentication Service for Computer Networks
  - Neuman and Ts'o, IEEE Communications, Sep 1994
- RFC 4120: The Kerberos Network Authentication Service (V5)
  - Neuman, Yu, Hartman, Raeburn, 2005
  - Updates RFC 1510: Kohl & Neuman
- RFC 3280: Internet X.509 PKI: Certificate & Certificate Revocation List Profile
  - Housley, Polk, Ford, Solo

References (cont)

- Compliance Defects in Public-Key Cryptography
  - Don Davis, 6th USENIX Security Symposium, 1996
  - http://www.usenix.org/publications/library/proceedings/sec96/davis.html
- PennNet Central Authentication Infrastructure
  - http://www.huque.com/~shuque/doc/2004-03-p21-authn.html

References (cont)

- Internet2 Middleware Initiative
  - http://middleware.internet2.edu/
- Signet
  - http://middleware.internet2.edu/signet/
- Grouper
  - http://middleware.internet2.edu/signet/
- Shibboleth
  - http://shibboleth.internet2.edu/

Questions or comments?

- Shumon Huque
  - shuque -AT- isc.upenn.edu
- Deke Kassabian
  - deke -AT- isc.upenn.edu