Dependable Security Token Services

Jim Basney

jbasney@ncsa.uiuc.edu

Motivation

- Online security token services are important for grid usability and manageability
  - Ease of enrollment/access/modification
  - Management of authorizations in a Privilege Management Infrastructure
- Availability Requirement: Access to credentials is in the critical path for secure grid computing
- Security Requirement: Protect against impersonation and privilege escalation

Security Token Services

- Issuers of Identity Credentials
  - Kerberos KDC
  - Online CA (KCA, COCA)
  - Web Single Sign-On (Liberty Alliance, Passport)
- Issuers of Authorization Credentials
  - Attribute Authority, SAML, Shibboleth, XACML
- Credential Repositories
  - MyProxy, SACRED, Active Directory, CODEX, TruePass, Web PassPort, KeySafe, Virtual Smart Card
- WS-Trust defines Web Service interface

Faults

- Key or password compromise
- Token expiration
- Token service unavailable
- Access control policy error / compromise
- Impersonation / Misidentification
- Resource consumption (DoS)

Fault Tolerance

- Diversity
- Obfuscation
- Proactive refresh of keys, software, and system configuration
- Least privilege rights delegation
- Protocol-level DoS prevention
- Quorum / Voting protocols
- Replication / Caching (next slide)

Replication / Caching

- Avoid replicating keys
  - Distinct keys/certificates/tokens enable auditing and selective revocation
  - Use delegation or trust hierarchy to empower replicas
  - Use threshold cryptography to reduce key vulnerability
- Replicate revocation information (CRLs, OCSP)
- Set lifetime restrictions and cache timeouts
  - Certificate/token/CRL lifetimes
- Use local token caches for network partition
  - Ex. Smartcard with authorization token cache

Sensors

- Intrusion detection (network & host-based)
- Polling for service availability
- Load monitoring
- Policy checking

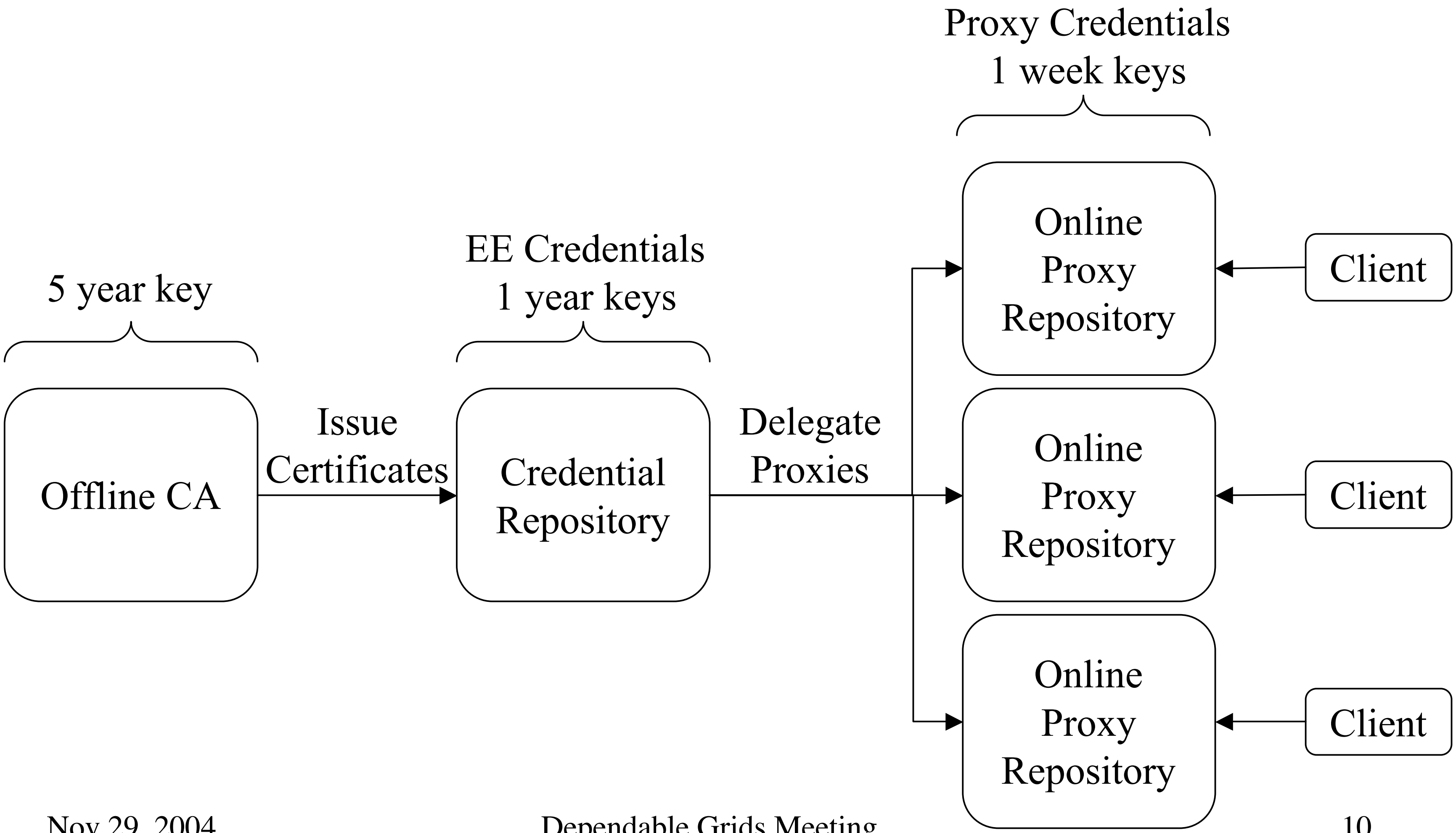
Actuators

- Revoke and re-key
- Refresh state (software, configuration, keys)
- Increase/Decrease replication/caching
- Posturing: remove vulnerable systems from network when under attack
- Increase security requirements at relying parties in an emergency

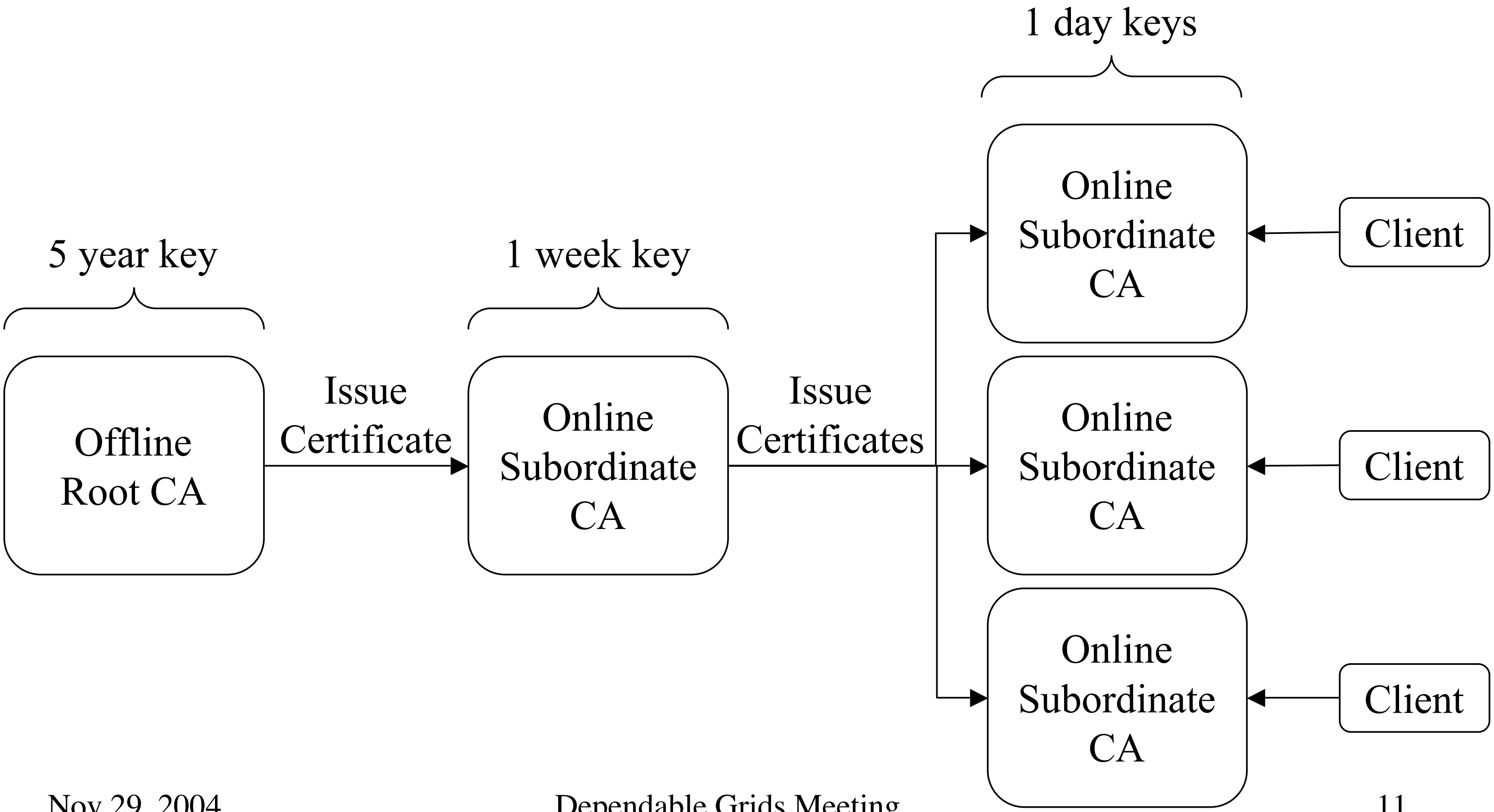
Examples
(in the following slides)

- Tiered token service architecture
  - Delegation to MyProxy servers
  - Hierarchy of online CAs
- Intrusion-Tolerant Password-Enabled PKI
- Cornell Online Certification Authority

Tiered Architecture (MyProxy)



Tiered Architecture (Online CA)



Intrusion-Tolerant Password-Enabled PKI

- X. Wang. Intrusion Tolerant Password-Enabled PKI. 2nd PKI Research Workshop, Apr 2003.
- Password-Authenticated Key Exchange (PAKE)
  - Password Verification Data (PVD) shared among n servers using (t, n)-Shamir secret sharing
  - User authenticates to subset of servers (≥t)
  - Proactively update PVD shares
- Password-adapted threshold RSA
  - RSA private key is split with one share derived from password and remaining shares distributed to servers

Intrusion-Tolerant Password-Enabled PKI

- Virtual Soft Token protocol
  - Client runs PAKE protocol with subset of servers to establish authenticated connections
  - Servers send password-encrypted private key shares
  - Client reconstructs password-encrypted key from server shares and decrypts key with user password
- Virtual Smart Card protocol
  - Client runs PAKE protocol as above and sends message hash to subset of servers for partial signature
  - Client constructs signature from password and server-computed partial signatures

COCA

- L. Zhou, F. Schneider, R. van Renesse. COCA: A Secure Distributed On-line Certification Authority. ACM TOCS 20, 4 (Nov 2002), p. 329-368.
- Threat model
  - n servers
  - at most t of n servers are ever compromised during each window of vulnerability
  - 3t+1 <= n
- Byzantine quorum for agreement
- Threshold cryptography to protect private key

COCA

- All COCA servers know each other's public keys
  - Periodically re-keyed by administrator
- One service private/public key pair
  - Everyone knows service public key
  - Service private key is split using threshold crypto
  - Key shares periodically refreshed via proactive secret sharing protocol

COCA

- Proactive recovery
  - Reload the code, eliminating Trojan horses
  - Reconstitute server state
  - Obsolete confidential info attacker may have obtained
  - Generate new service private key shares
- DoS attack protection
  - Process only authorized requests
  - Group requests into classes and multiplex resources
  - Cache results of expensive crypt ops