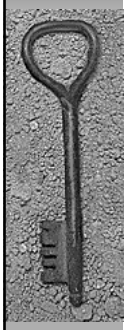
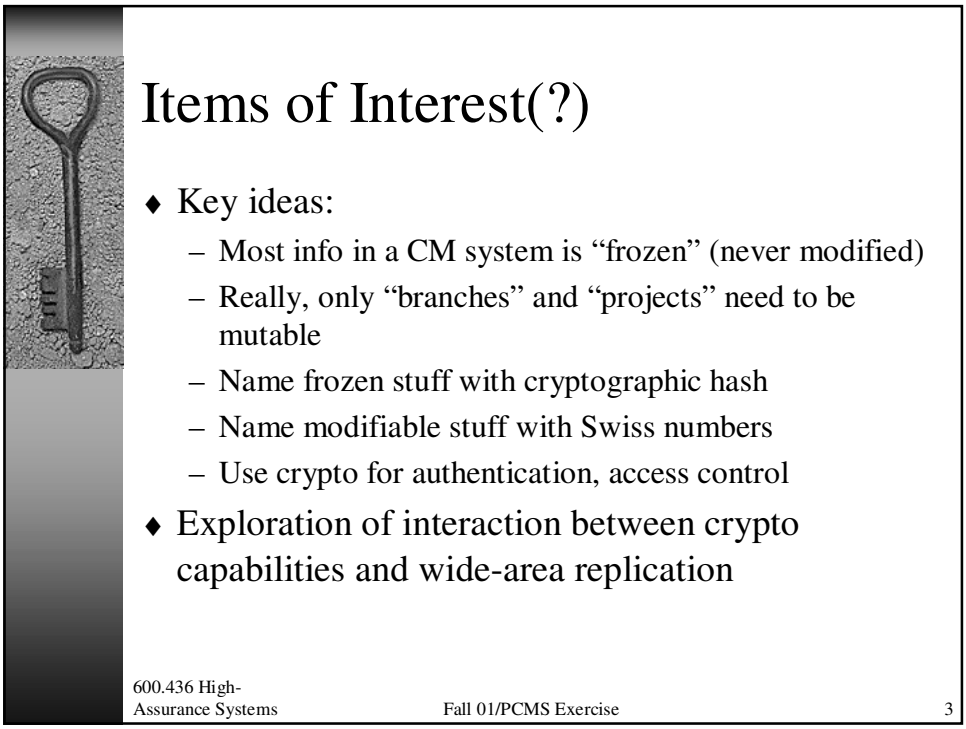
PCMS Exercise

- Do a simple threat analysis
- Ask if the results make sense



What is PCMS?

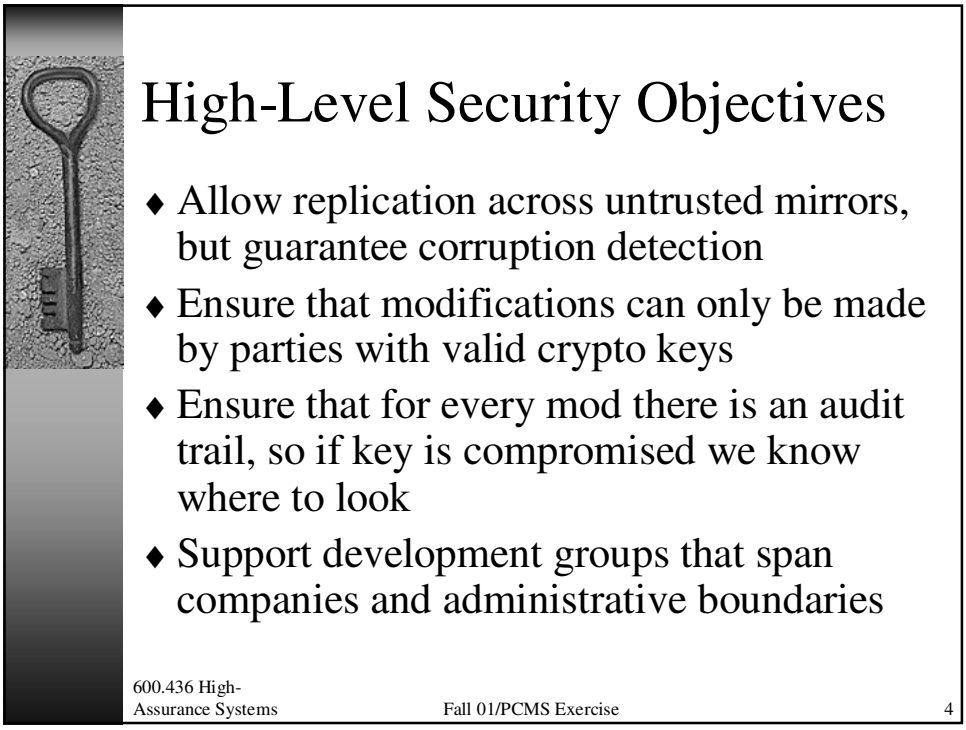
- PCMS is a high-integrity configuration management system
- Most objects are written once and never modified (we call these “entities”)
  - Archival store, not a database
- What security policies apply to such objects?
- What are the threats?
- What does this mean in implementation terms?



Items of Interest(?)

- Key ideas:
  - Most info in a CM system is “frozen” (never modified)
  - Really, only “branches” and “projects” need to be mutable
  - Name frozen stuff with cryptographic hash
  - Name modifiable stuff with Swiss numbers
  - Use crypto for authentication, access control
- Exploration of interaction between crypto capabilities and wide-area replication

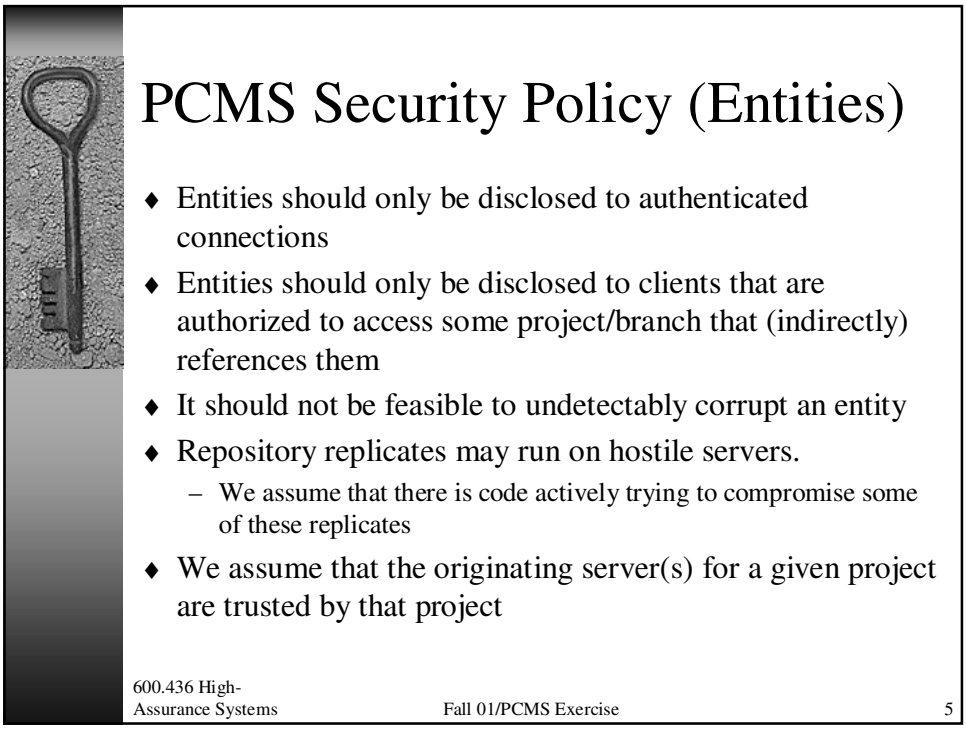
600.436 High-Assurance Systems Fall 01/PCMS Exercise 3

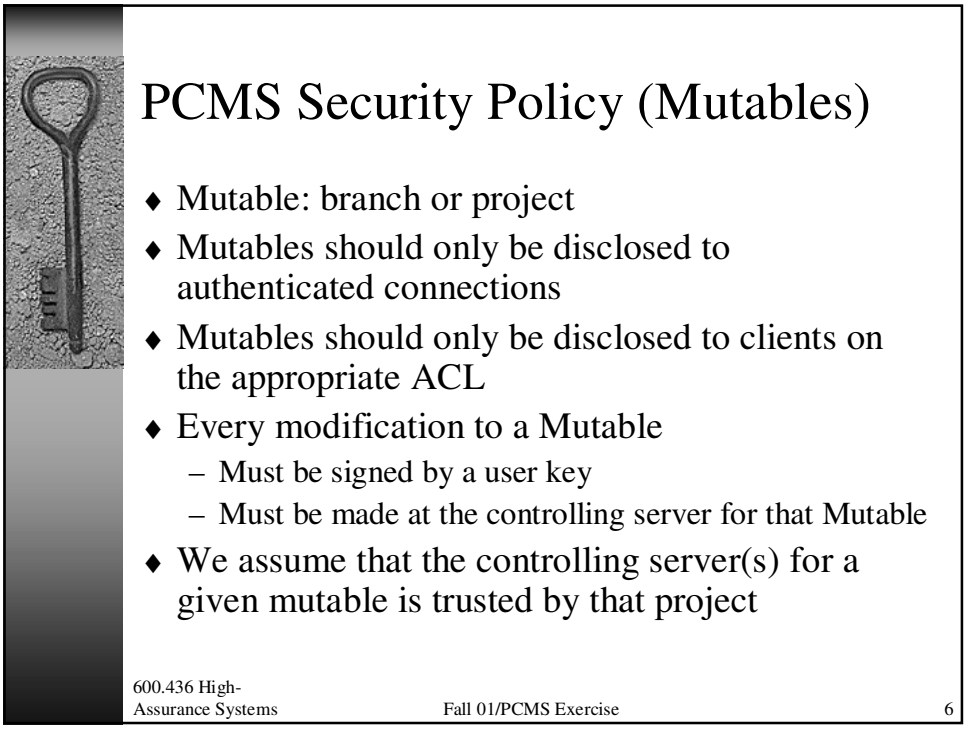


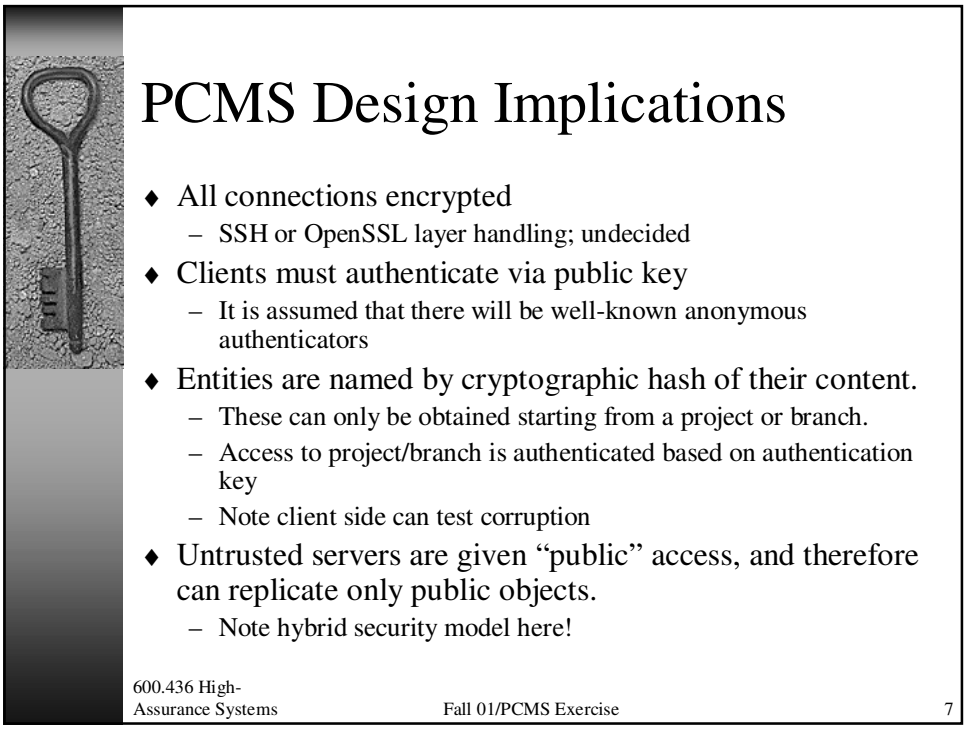
High-Level Security Objectives

- Allow replication across untrusted mirrors, but guarantee corruption detection
- Ensure that modifications can only be made by parties with valid crypto keys
- Ensure that for every mod there is an audit trail, so if key is compromised we know where to look
- Support development groups that span companies and administrative boundaries

600.436 High-Assurance Systems Fall 01/PCMS Exercise 4


PCMS Security Policy (Entities)
♦ Entities should only be disclosed to authenticated connections
♦ Entities should only be disclosed to clients that are authorized to access some project/branch that (indirectly) references them
♦ It should not be feasible to undetectably corrupt an entity
♦ Repository replicates may run on hostile servers.
– We assume that there is code actively trying to compromise some of these replicates
♦ We assume that the originating server(s) for a given project are trusted by that project
600.436 High-Assurance Systems
Fall 01/PCMS Exercise
5

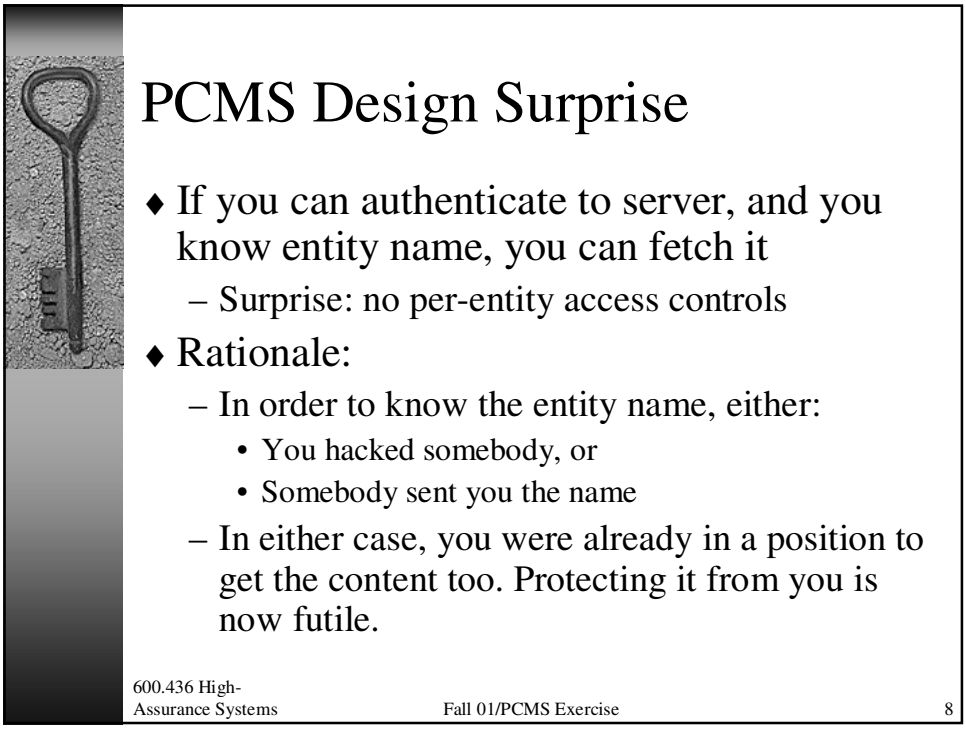

PCMS Security Policy (Mutables)
♦ Mutable: branch or project
♦ Mutables should only be disclosed to authenticated connections
♦ Mutables should only be disclosed to clients on the appropriate ACL
♦ Every modification to a Mutable
– Must be signed by a user key
– Must be made at the controlling server for that Mutable
♦ We assume that the controlling server(s) for a given mutable is trusted by that project
600.436 High-Assurance Systems
Fall 01/PCMS Exercise
6



PCMS Design Implications

- All connections encrypted
  - SSH or OpenSSL layer handling; undecided
- Clients must authenticate via public key
  - It is assumed that there will be well-known anonymous authenticators
- Entities are named by cryptographic hash of their content.
  - These can only be obtained starting from a project or branch.
  - Access to project/branch is authenticated based on authentication key
  - Note client side can test corruption
- Untrusted servers are given "public" access, and therefore can replicate only public objects.
  - Note hybrid security model here!

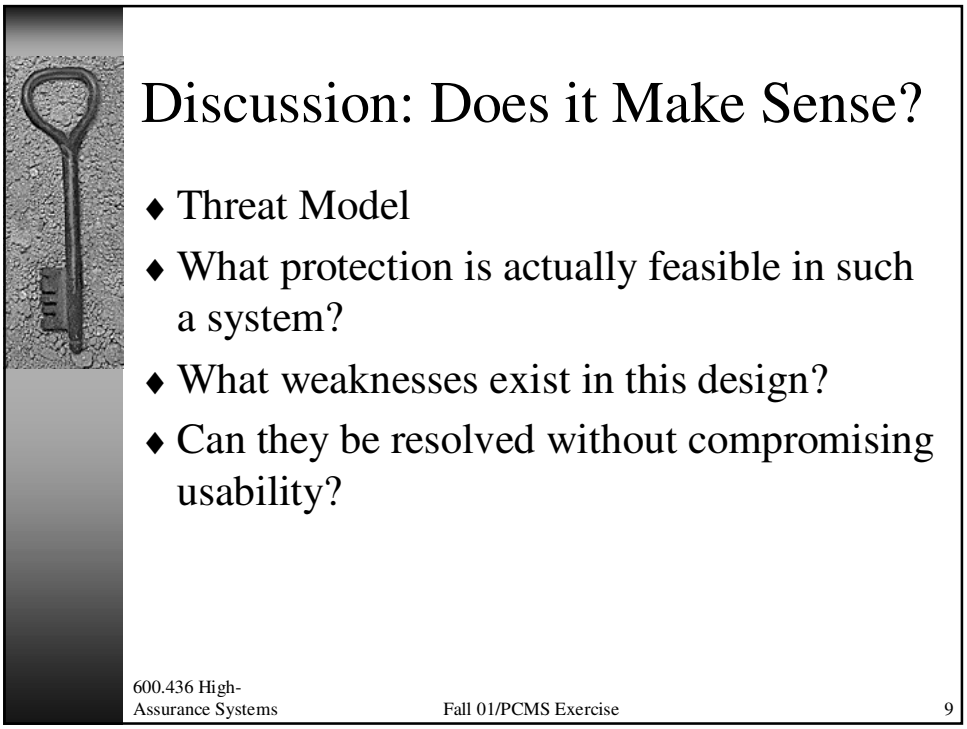
600.436 High-Assurance Systems Fall 01/PCMS Exercise 7



PCMS Design Surprise

- If you can authenticate to server, and you know entity name, you can fetch it
  - Surprise: no per-entity access controls
- Rationale:
  - In order to know the entity name, either:
    - You hacked somebody, or
    - Somebody sent you the name
  - In either case, you were already in a position to get the content too. Protecting it from you is now futile.

600.436 High-Assurance Systems Fall 01/PCMS Exercise 8


Discussion: Does it Make Sense?
♦ Threat Model
♦ What protection is actually feasible in such a system?
♦ What weaknesses exist in this design?
♦ Can they be resolved without compromising usability?
600.436 High-Assurance Systems
Fall 01/PCMS Exercise
9