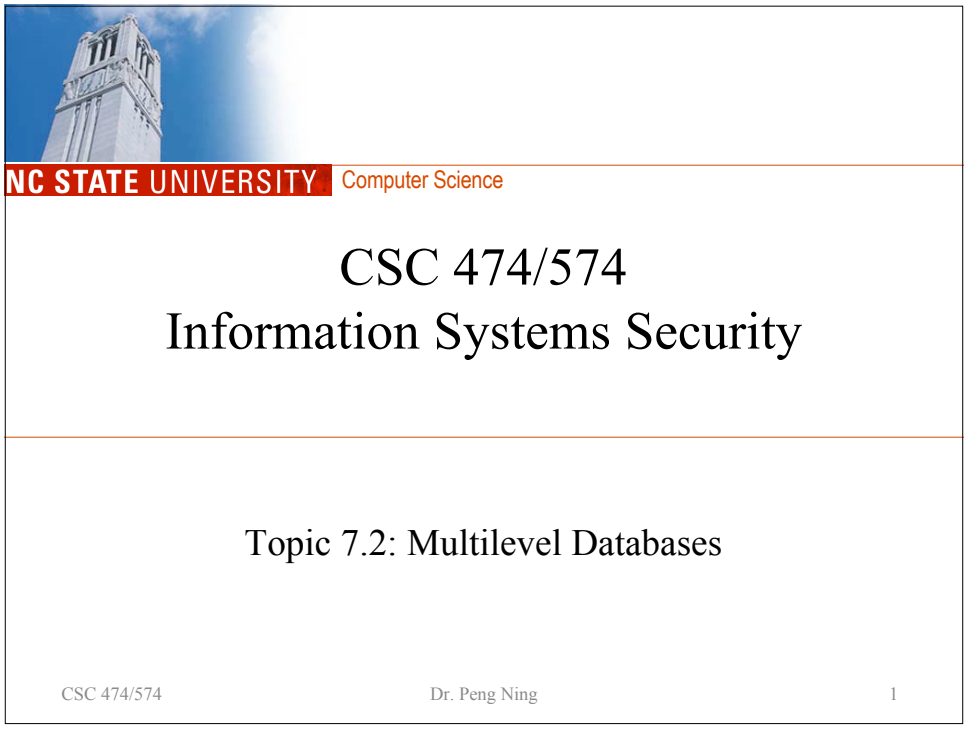


CSC 474/574
Information Systems Security

Topic 7.2: Multilevel Databases

Approaches to Multi-level Databases

- Partitioning
- Encryption
- Integrity lock
- Trusted Front-End
- Distributed Databases

Partitioning

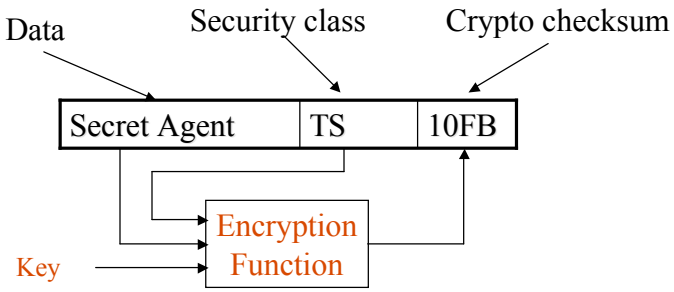
- Separate data in different levels into different partitions.
  - Redundancy
    - Example: the primary key of a logical relation must be duplicated in all partitions in which the relation are stored.
  - Usability
    - Example: a high-level user needs to combine both high-level and low-level data.

Encryption

- Encrypt the sensitive data at each level with a key unique to that level.
  - Known plaintext attack
    - Example:
      - Party attribute is encrypted.
      - Alice knows party="Democrat" for Bob; she can compare the ciphertext of Bob's party attribute with other tuples
    - Reason: Limited set of plaintexts.
  - Authentication
    - Example:
      - Replace one ciphertext with another
  - Above problems can be partially avoided with multiple keys.
  - Unable to use DBMS functionalities for encrypted data.
    - Query optimization, indexes, etc.

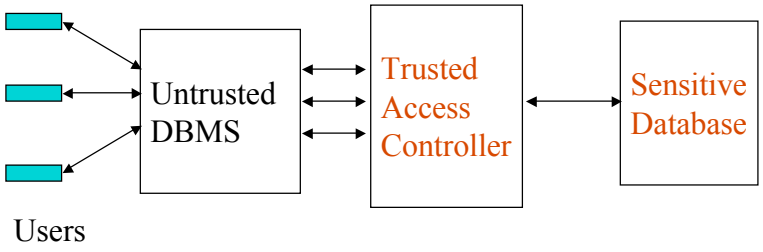
Integrity Lock

- Provide integrity and limited access for a database.



- Any unauthorized changes to data items can be detected.
- Access to data items is based on the security labels.

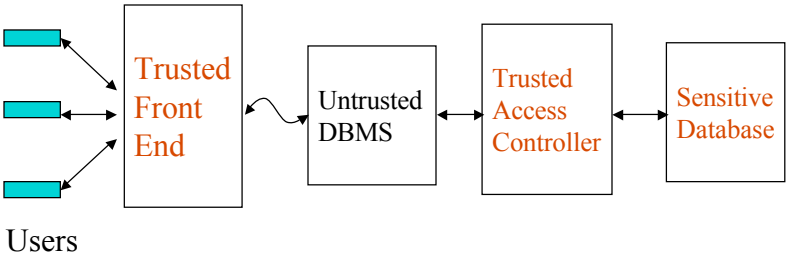
Integrity Lock DBMS



- Problems
  - Efficiency
    - Data expansion
    - Processing time required for generating, modifying, and verifying integrity locks
  - Security
    - Untrusted DBMS sees all data passing through it.

Trusted Front End

- Trusted Front End
  - User authentication
  - Access control
  - Verification
  - Essentially a reference monitor



Trusted Front End (Cont'd)

- Commutative Filters
  - Processes that interfaces to both the user and the DBMS.
  - Reformat the query by putting in more conditions to filter out unnecessary records.
  - Example:
    - Retrieve NAME where ((Occup= Physicist) ^ (City =WashDC))
      From all records R
    - After reformatting
    - Retrieve NAME where ((Occup= Physicist) ^ (City =WashDC))
      From all records R where
      (Name-level (R) <= User-level) ^
      (Occup-level (R) <= User-level) ^
      (City-level (R) <= User-level)

Distributed Databases

- Store data items at different level in different physical databases
- Trusted front-end translates each query into single-level queries and send to different databases
- Trusted front-end combines results and returns to the user.