Chapter 3: Design Principles

# DATABASE SECURITY

# Mandatory Access-Control Policies

- Bell and LaPadula policy
  - subjects
    - are assigned clearance levels
    - they can operate a level up to and including their clearance levels.
  - Objects are assigned sensitivity levels.
  - The clearance levels as well as the sensitivity levels are called security levels.
  - The set of security levels:
    - Unclassified < Confidential < Secret < TopSecret

# Mandatory Access-Control Policies

- Bell and LaPadula policy
  - The following are the two rules of the policy:
    1. Simple Security Property: A subject has read access to an object if its security level dominates the level of the object.
    2. *-Property : A subject has write access to an object if the subject's security level is dominated by that of an object.
  - For database systems
    - *-property: A subject has write access to an object if the subject's level is that of the object

# Mandatory Access-Control Policies

- Polyinstantiation
  - the same object can have different interpretation and values at different levels
    - Example
      - at the Unclassified level an employee's salary may be 30,000
      - at the Secret level the salary may be 70,000

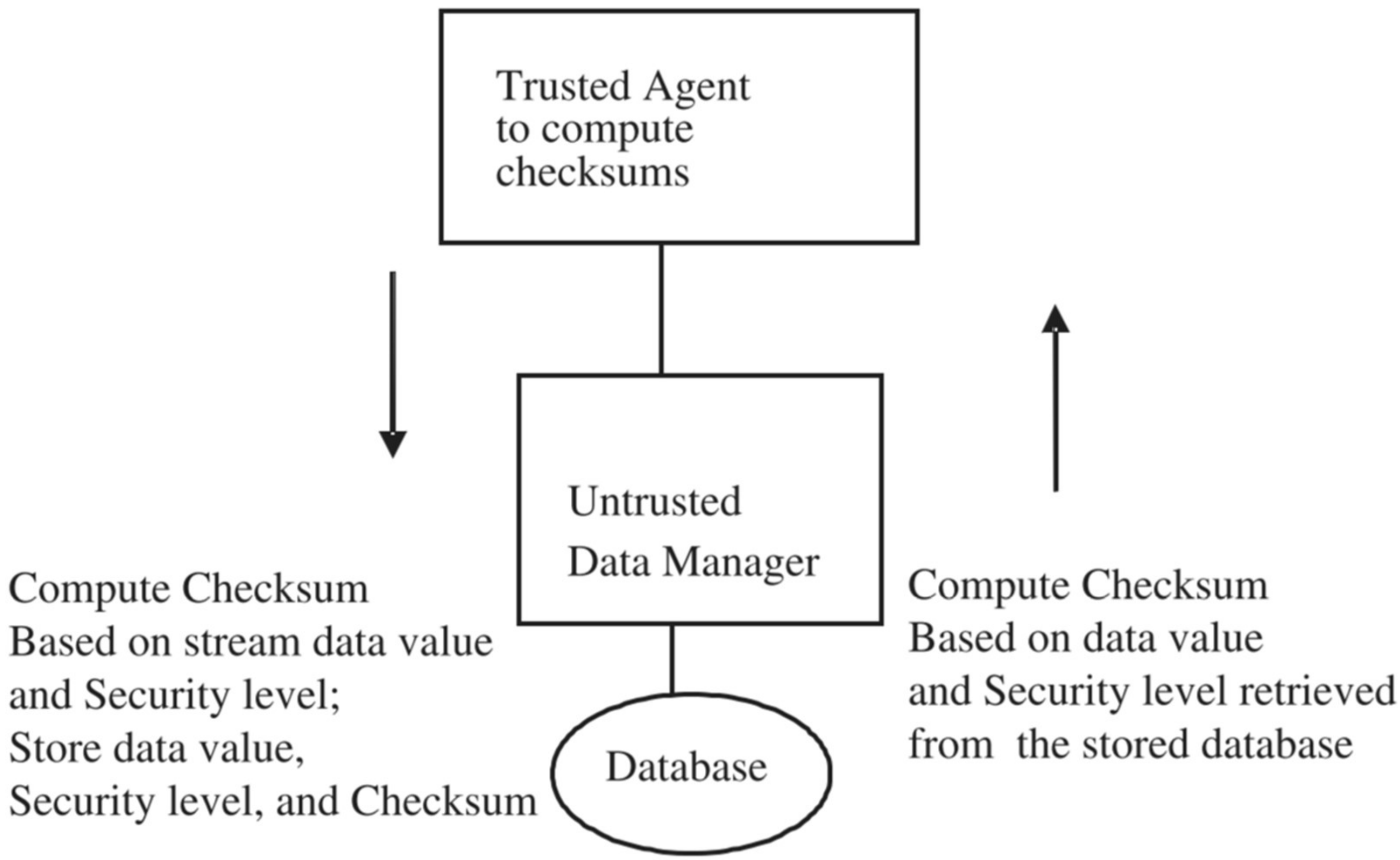
# Security Architectures

Taxonomy/Security
Architectures for MLS/DBMSs:

Integrity Lock
Trusted Subject
Operating System Providing
Mandatory Access Control
Distributed: Partitioned and Replicated
Kernel Extensions

# Security Architectures

- Integrity Lock



# Security Architectures

- Integrity Lock
  - Multiple instantiations of the front end
    - one instantiation for each user level
  - every tuple is associated with
    - a security label : encrypted
    - a cryptographic checksum
  - data is not encrypted.
  - The checksums are computed by the trusted filter on insertion and recomputed during retrieval.

# Security Architectures

- Integrity Lock
  - For insertions
    - the trusted filter computes the checksum
    - the untrusted back-end DBMS stores data and associated label and checksum in the database
  - On retrieval
    - the back end retrieves the data tuples and passes them to the trusted filter
    - trusted filter recomputes the checksum based on the tuple and label
      - If data has not been tampered with, it passes the data to the user

# Security Architectures

- Integrity Lock
  - Advantage:
    - small amount of additional trusted code
    - performance is independent of the number of security levels involved
  - Disadvantage:
    - subject to a threat
      - untrusted back end is able to
        - view classified data
        - encode it as a series of unclassified data tuples
        - pass the encoded data tuples to the trusted front end
      - Because the data tuples are unclassified
        - the trusted filter will not be able to detect

# Security Architectures

- Operating System Providing Access Control
  - also known as the Hinke–Schaefer
  - utilizes the underlying trusted operating system for access-control
  - No access-control is performed by the DBMS.
  - The DBMS objects (e.g., tuples) are aligned with the underlying operating system objects (e.g., files).
    - Secret tuples are stored in Secret files
    - Top Secret tuples are stored in Top Secret files
  - There is no single DBMS
    - an instantiation of the DBMS for each security level

# Security Architectures

- Operating System Providing Access Control
  - Also called the single kernel approach
  - Advantage
    - it is simple and secure
  - Disadvantage
    - performance will decrease with the number of security levels

# Security Architectures

- Kernel Extensions Architecture
  - is an extension of the single kernel approach
  - The underlying operating system is utilized to provide the basic MAC and DAC
  - DBMS will supplement this access mediation
    - For example
      - DBMS might provide context-dependent DAC on views.
  - has the same performance problems associated with the single kernel approach.
  - But it provides more sophisticated access-control mechanisms
    - it could address some real-world access-control needs

# Security Architectures

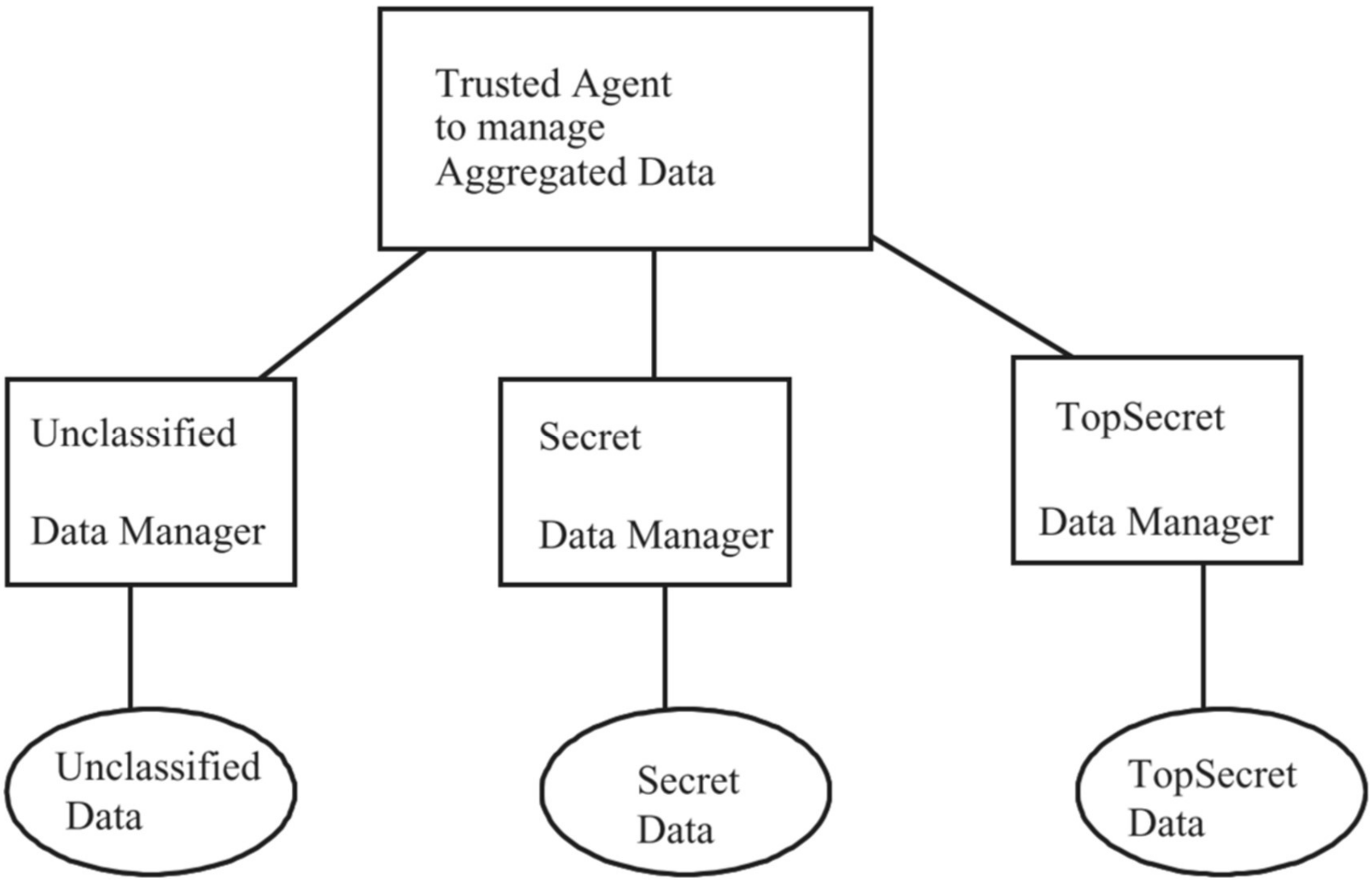
- Trusted Subject Architecture
  - sometimes called dual kernel-based architecture
  - does not rely on the underlying operating system to perform access-control
  - DBMS performs its own access mediation
  - Advantage
    - it can provide good security
    - its performance is independent of the number of security levels
  - Disadvantage is that the DBMS code must be trusted
    - large amount of trusted code may be needed for this approach

# Security Architectures

- Distributed Architecture
  - there are
    - multiple untrusted back-end DBMSs
    - single trusted front-end DBMS
  - Communication between the back-end DBMSs occurs through the front-end DBMS
  - two main approaches
    - Partitioned
    - Replicated

# Security Architectures

- Partitioned distributed architecture

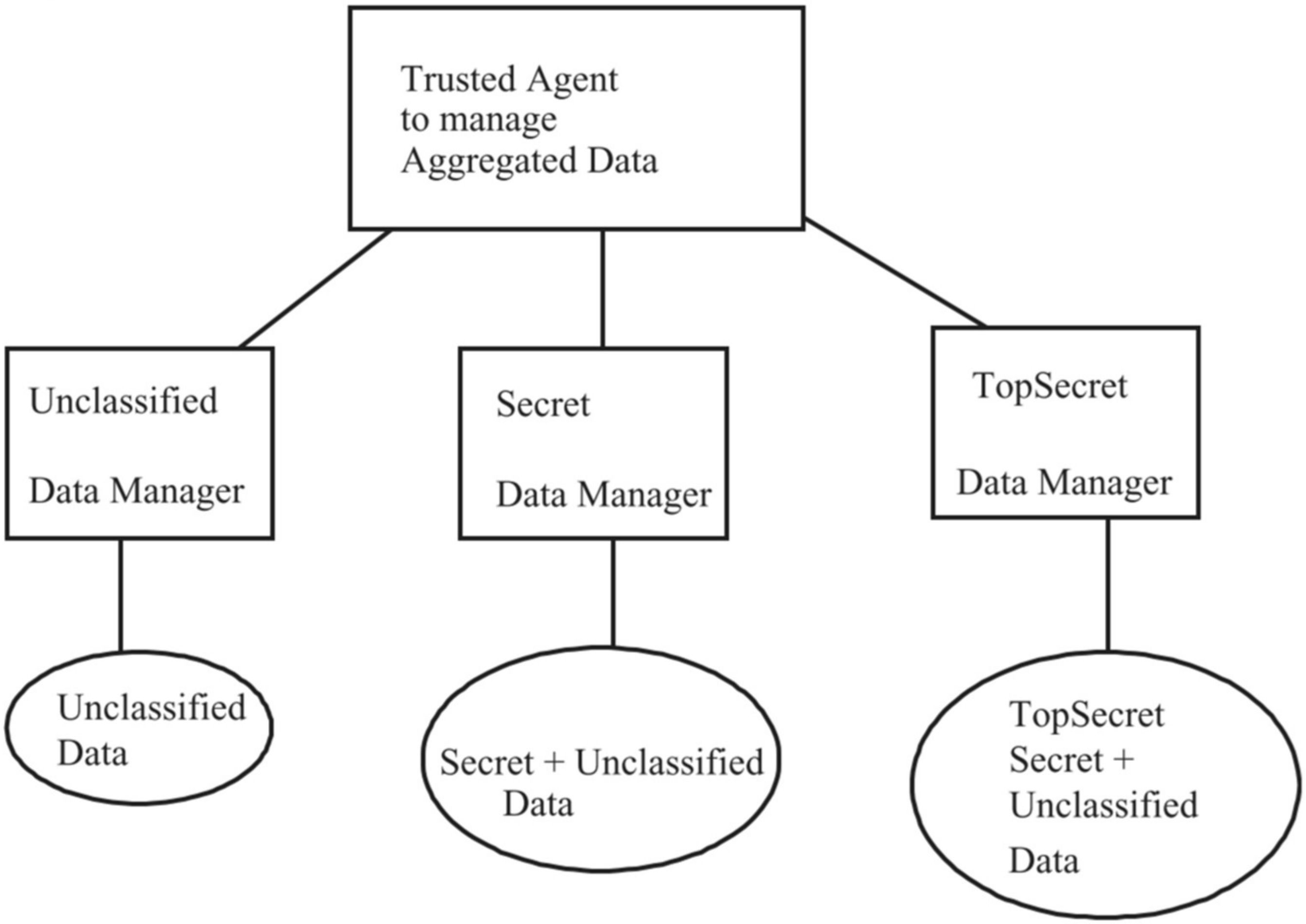


# Security Architectures

- Partitioned distributed architecture
  - the trusted front end is responsible for
    - ensuring that the query is directed to the correct back-end DBMS
    - performing joins on the data sent from the back-end DBMSs.
  - query itself could contain information classified higher than the backend DBMSs
    - queries should not be sent to the DBMSs that are operating at levels lower than the user.

# Security Architectures

- Replicated distributed architecture



# Security Architectures

- Replicated distributed architecture
  - trusted front end ensures that the query is directed to a single DBMS
  - only the DBMSs operating at the same level as the user are queried
  - this approach does not require front-end DBMSs to perform the join operations.
  - front end must ensure consistency of the data maintained by the different DBMSs