Security in Computing & Information Technology

Lecture 7

Database Security

Lecture Schedule

Foundations

1. Introduction
2. Vulnerabilities, Threats, Attacks

Basic mechanisms

3. Security mechanisms, Elementary cryptography
4. Authentication
5. Access control

Major computing security areas

6. Operating systems
7. Databases
8. Networks
9. Web
10. Mobile computing

Applications

11. Privacy
12. Internet banking

Lecture Topics

- Databases & components
- Reliability, integrity & security of databases
- Attacks on databases
  - Inference
  - Injection
- Backup

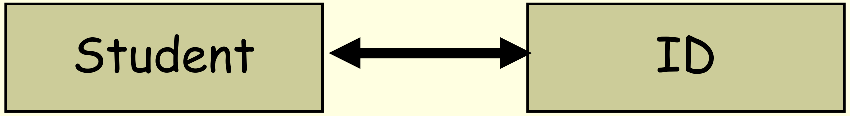
Databases

- Organised collections of data for storing, managing and retrieving information
- Basic terms
  - Entity: a single object about which data is stored
  - Record (tuple)

    Structured data item, consists of fields
    - Field, attribute, element: single unit of data (part of a record)
  - Schema
    - Logical structure of the database
    - Describes the entities and their relations
  - Basic operations
    - Query: retrieving data values
    - Update: modifying data values

Relation Types

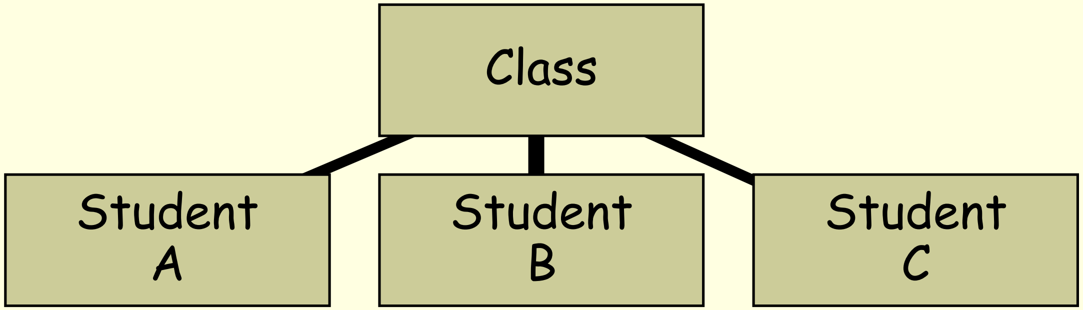
One-to-one



Student

ID

One-to-many



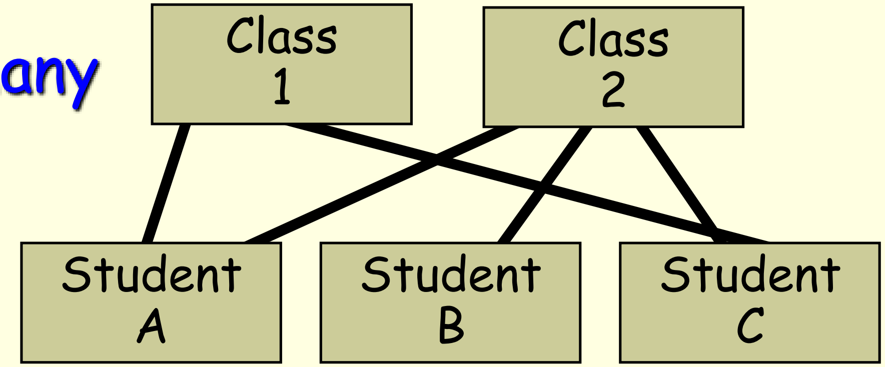
Class

Student A

Student B

Student C

Many-to-many



Class 1

Class 2

Student A

Student B

Student C

Database Management System (DBMS)

- Software to create and maintain data
- Independent of specific computer programs
- Advantages of DBMS
  - Efficiency
    - Shared access by many users
    - Minimal redundancy by having one copy of data
  - Security
    - Controlled access for authorised users only
    - Data consistency
      - Internal: data obeys certain rules (e.g. stock level ≥ 0)
      - External: the database entries are correct
    - Data integrity

DBMS Components

- Data definition language
  - Defines data structures
    Example: XML schemas
- Data manipulation language
  - Used to insert, delete and update data in a database
  - Querying data (read-only) may or may not be part of it
  - Most popular: Structured Query Language (SQL)
- Data dictionary
  - Central repository of information about data
  - Formal definitions of all variables in database
    - Meaning, relationships to other data, origin, format …

SQL Basics

- Queries
  - **`SELECT (FROM, WHERE, GROUP BY, …)`**

    Returns a set of records described in the query
- Data manipulation
  - **`INSERT, UPDATE, DELETE, MERGE, …`**

    Enter, remove, modify records in the database
- Transaction control

  (Transaction: a sequence of coherent operations)
  - **`COMMIT, ROLLBACK`**

    Save or discard the result of a transaction
- Data definition
  - **`CREATE, DROP, ALTER`**

    Manipulate the schema
- Data control
  - **`GRANT, REVOKE`**

    Modify access rights

DBMS Security Requirements

- Integrity
  - Physical (hardware) integrity
  - Logical (schema) integrity

    Protects against database corruption
  - Element (data) integrity

    Ensures data accuracy and correctness
- Auditability
  - Ascertain the validity and reliability of data
- Access control
  - Authentication

    Verifies user's eligibility to use the system
  - Confidentiality

    Only authorised users can access the system
- Availability

Techniques for Reliability and Integrity

- Reliable data updating techniques
  - Two-phase update
    - Phase 1 (intent): collect information for changes
    - Phase 2 (commit): make permanent changes
- Internal consistency
  - Error detection code

    Checksum/hash stored together with the data
  - Shadow copy

    Duplication of data
- Monitors

  Assure the availability and correct operation of the database, and enforce
  - value constraints
  - state constraints
  - transition constraints

Physical and Logical Security

- Physical protection
  - Disk, USB memory, tape
  - Need protection from
    - harm (fire, flood, etc)
    - unauthorised access (encrypted data)
- Logical protection
  - Data as interpreted by the application (facts)
  - Protection needed for
    - data dictionary (schema integrity)
    - data (accuracy and integrity)

Sensitive Data

- Inherently sensitive data

  E.g. location of missiles
- Data from a sensitive source

  E.g. police informant
- Data declared to be sensitive

  E.g. anonymous donor
- Part of a sensitive record
- Sensitive in relation to previously disclosed information

  E.g. longitude + latitude

Handling Sensitive Data

- Access decisions
  - Data availability

    Scenarios when data cannot be accessed

    E.g. access is blocked while data is being updated
  - Acceptability of access

    Access to certain fields or to a combination of certain fields may not be allowed

    E.g. access to student number and result at the same time is blocked
  - Other constraints
    - Time of access

      E.g. data is accessible during working hours only
    - Location of access

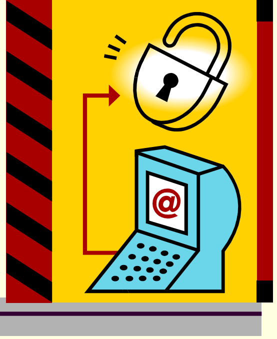
      E.g. data can be accessed from within the organisation only
    - History of user queries

      Current query, combined with precious ones, can reveal sensitive information

Types of Disclosures

- Exact data
  - Results in immediate breach of security/privacy
- Bounds of data values
  - Can lead to informed guess about data values, e.g. by iteratively reducing range
- Probable value
  - Sometimes almost as good as an accurate value
- Existence
  - E.g. being on a patient list provides medical information
- Negative query result
  - E.g. a person does not have a particular disease

Data Protection



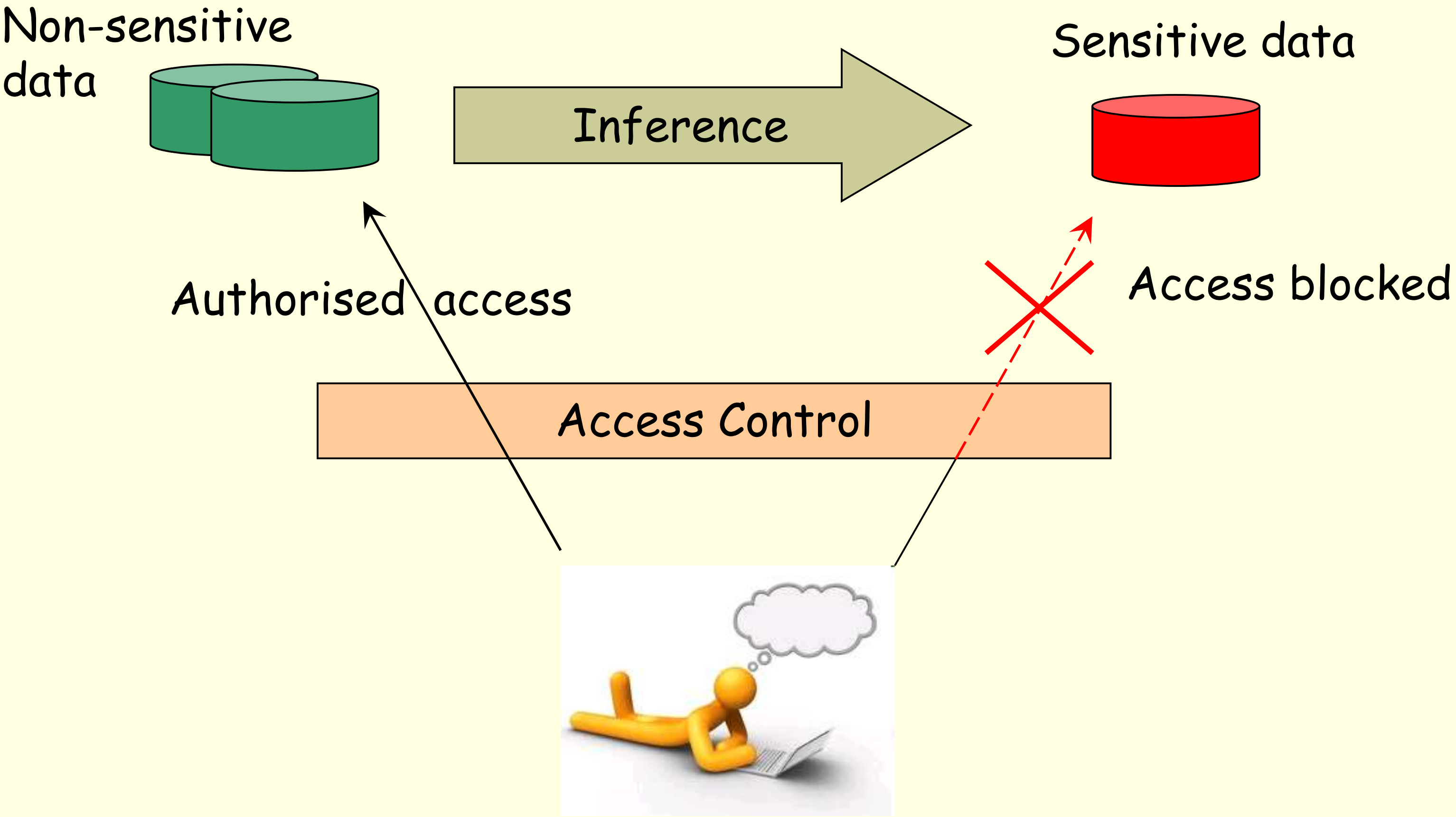
- Data suppression
  - Data access explicitly denied
  - Combine multiple answers (to hide actual data)
- Data concealing
  - Data returned is not exact, but still close enough
    - rounding
    - range of result given
    - obfuscation (data masking): data is replaced with realistic, but not real data

Database Security

- Operational issues
  - Major threats
    - Query (read): information leak
    - Update (write): integrity compromise
  - Other security services (authentication, access control) can eliminate unauthorised modes of access
  - Authorised mode of access can still have problems
    - Inference: indirect attack

      Inferring sensitive data from non-sensitive data
      - Logical inference
      - Statistical inference

Indirect Access via Inference



Non-sensitive data

Inference

Sensitive data

Authorised access

Access blocked

Access Control

Logical Inference Example

- Constraints
  - There are four categories (A, B, C, D)
  - The system does not answer, if the answer would imply the secret
- Queries
  - Q1: Is element X of type A? – A: - (No answer)
  - Q2: Is element X of type B? – A: No
  - Q3: Is element X of type C? – A: No
  - Q4: Is element X of type D? – A: - (No answer)
  - Conclusion: Element X is of type A

Statistical Inference

- Statistics
  - Macro-statistics: collection of related data
  - Micro-statistics: individual records without identifying information
- Legitimate aim: aggregate information about groups of entries (sum, count, mean, etc)
- Risk: leaking specific information about individual entries
- Inference attack: extract sensitive data from statistics
- Compromise
  - Exact: find an exact value of an individual entry
  - Partial: find an estimate of an attribute of an individual entry (e.g. the GPA of s1234567 is between 3.5 and 3.7)

Statistical Inference Attack

- Sum

```
Select SUM(salary);
```

- Count

```
List (employees);
```

- Mean

  Mean = Sum / Count

- Median

  Slightly more complex process, may determine individual values

A combination of the above can narrow down the answer to generally looking queries

E.g.

```
Select SUM(salary)-
Select SUM(salary) where lastname != 'Smith';
```

Direct Inference Attack

- Privacy constraint: Direct access to certain individual records is not allowed
- Small query set attack: a query that yields a few records
  - Trivial attack: the answer is a single record
    - Example

```
List NAME where
     (sex=m and drugs = y) or
     (sex≠m and sex≠f) or
     (home=nowhere)
```

These two lines select no records, only make the query less obvious

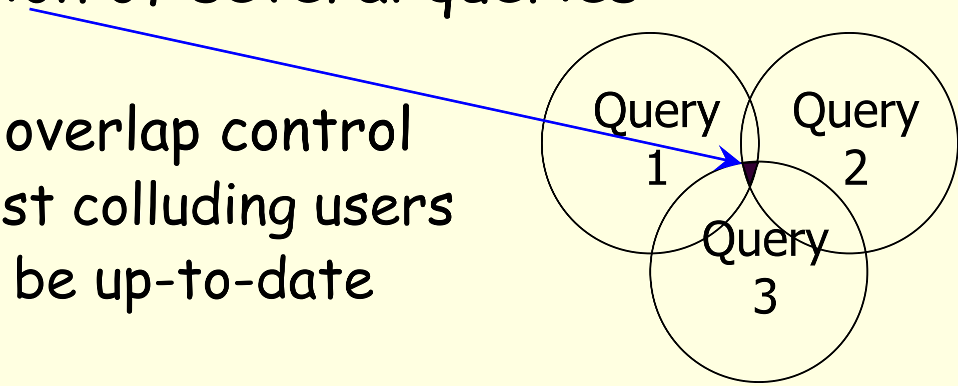
- Prevention
  - Query size restriction
    - AKA limited response suppression
    - The user may not access any query set with less than *k* records
    - *K-anonymity*: at least *k* records have the same attribute

Indirect Inference Attacks

- Tracker attack: issue two queries
  - The difference of two queries identifies a single record
  - Neither of the two queries violates the query size restriction

    Example

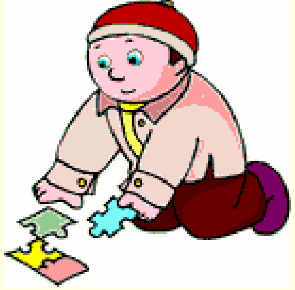
    (mean salary of employees and president) minus (mean salary of employees) = ?
- Overlap: The intersection of several queries identifies a record
  - Prevention: query set overlap control
    - Not effective against colluding users
    - User history has to be up-to-date



Data Mining

- (The activity of) analysing data from different perspectives
- Aims
  - Establish certain patterns
  - Extract useful information
- Data warehouses
  - Integration of various databases
  - Designed to facilitate data analysis
  - Types
    - On-line transactional processing (OLTP) warehouses
      - Store up-to-date information
      - Support day-to-day operations
    - On-line analytical processing (OLAP) warehouses
      - Store historical information
      - Support decisions, long-term information needs

Data Aggregation



- Combining different data
- Can easily lead to the identification of a single item/person

  Examples

  - Combining longitude and latitude to pinpoint a geographical location
  - Combining date of birth and home address to identify a person
- Data aggregators
  - Organisations collecting people's data from different databases and selling the information to others

SQL Injection Attack

- Entering user input that can be interpreted as an SQL command
- Improperly filtering programs can execute the (probably malicious) code

Embedded SQL

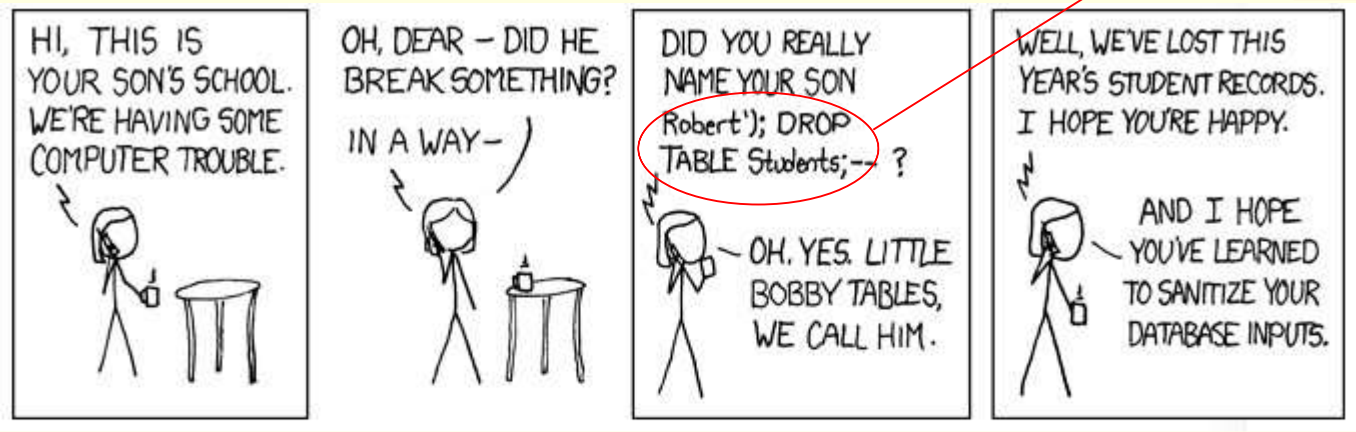

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Image source unknown

SQL Injection Example

- Code

```
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE
                owner = "'"
                + userName + "' AND itemname = '"
                + ItemName.Text + "'";
```

- Intention

```
SELECT * FROM items
WHERE owner =
AND itemname = ;
```

Restrict query to items whose owner matches the currently authenticated user

- Possible input

```
SELECT * FROM items
WHERE owner = 'john'
AND itemname = 'name' OR 'a'='a';
```

Returns all entries, because 'a' = 'a' means the second condition is always true

- Result

```
SELECT * FROM items
```

Backups

- Making additional copy/copies, in case the original gets damaged or lost
  - Backup of data only, or the whole system (including or excluding data)
- Protection against
  - hardware errors
    - Original data and backup should be on different devices
  - user or program errors
    - Production of a backup is built into the process
      - E.g. Recycle bin (Windows) or separate directories (`.ckpt` in Unix) against accidental deletions
  - malicious actions
    - Backup on well-protected, possibly remote systems/sites

Backup Methods

- Data backup methods
  - Off-line

    CD, DVD, flash drive (e.g. USB memory)
    - Also used for archivation (permanent records)
  - Hot-swap

    External hard disk, flash drive
- System backup
  - Size problem: too large for a DVD or even for flash drives
  - Hot-swap
  - External drives, disconnected after backing up

Backup Strategies

- Incremental
  - Saves changes since the last backup
  - Faster to do a backup
    - Less information/data to save
  - Takes longer to restore
- Mirroring
  - Saves all data or the whole system
  - May take a long time to do a backup
  - Restore is easy and straightforward
- Frequency
  - Should be regular/periodic
  - Should mix incremental and mirroring
    - E.g. mirroring once a week, incremental all other days

Backup Separation

- Place
  - Backup should be stored offsite
    - In case of natural disaster, data will still be safe
- Method
  - Different methods should be used together

    If the backup hardware/software (tape, DVD …) fails, you still have another method to rely on (on-line backup, offsite …)
- Timing
  - Backup should be made when data is not in use (files not locked, …)
  - File system snapshots (instantaneous image of a file system) can be made while the system is active (files are open/locked)

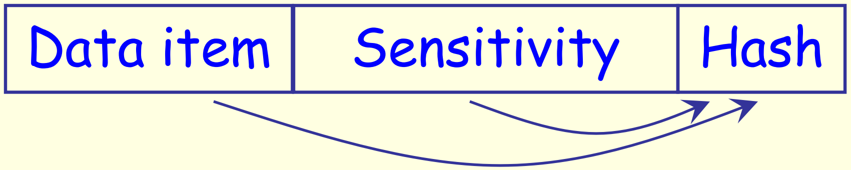
Multilevel Database Security Issues

- Data in a database can have different sensitivity levels
  - A single element's sensitivity level can differ from that of other elements in the same record (e.g. name, salary)
  - More than two levels of sensitivity are possible (e.g. top secret, secret, confidential, free access)
  - The security of an aggregate may be different from that of the individual elements (e.g. student number + result)
    - Protection granularity

      The size of protected object
- Solutions
  - Encryption: using different keys for different sensitivity levels
  - Integrity lock
  - Sensitivity lock

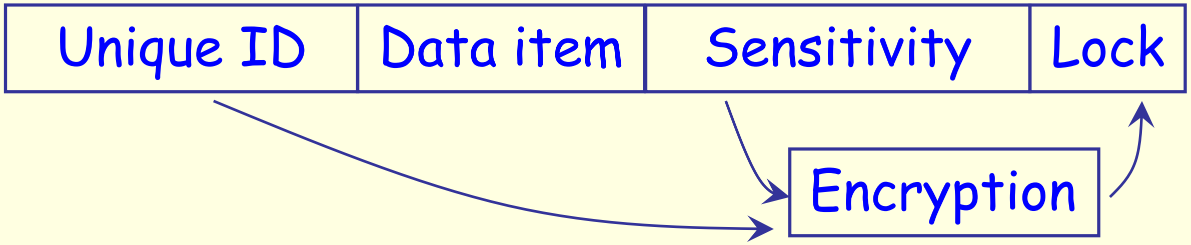
Multilevel Database Security Solutions

- Encryption
  - Encrypting data with different keys for different sensitivity levels
- Integrity lock
  - Sensitivity level is stored with data and both are protected by a hash



Data item | Sensitivity | Hash

- Sensitivity lock
  - Combination of a unique ID and the sensitivity level in encrypted form
  - The lock's content is not accessible in ordinary view



Unique ID | Data item | Sensitivity | Lock

Encryption

Distributed Databases

- Data is stored on different hosts connected by a computer network
- Issues
  - Partitioning
    - The network is split into domains that cannot communicate with each other
    - Data may become unavailable
    - Modifications may not be propagated
  - Replication
    - Multiple copies of the data exist on different computers
    - Consistency between copies need to be maintained

Summary

- Databases can reveal data directly, or allow users to infer from statistics or from other data
- Data aggregation can lead to identification attack
- Injection attacks can corrupt the database
- Regular backups are needed for reliable operation