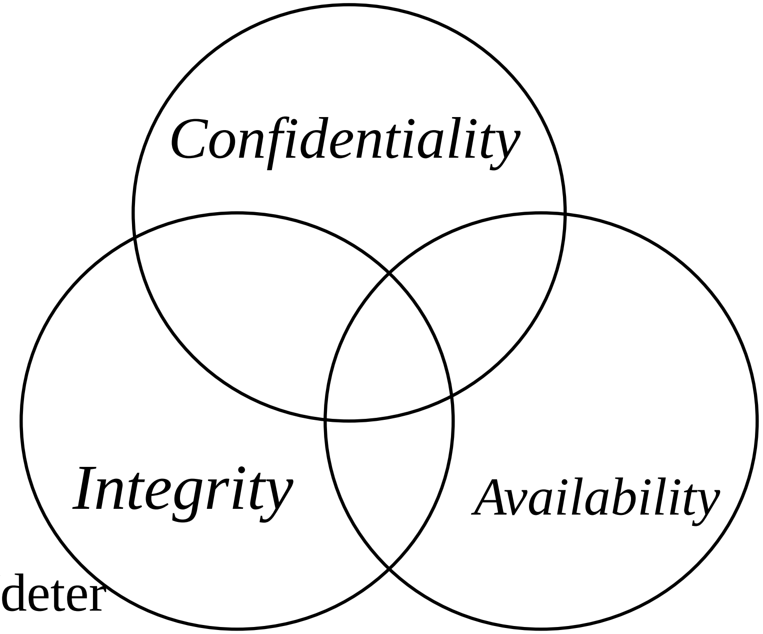
EECS565 Intro to Computer and Information Security

Intro to Database Security

Bo Luo
bluo@ku.edu

Database Security

Prevent/detect/deter improper
Disclosure of information



Prevent/detect/deter
Improper **modification**
of information

Prevent/detect/deter improper
Denial of access to services

DBMS

- Collection of
  - interrelated data and
  - set of programs to access the data
- Convenient and efficient processing of data
- Database Application Software

Database Security

- Protect Sensitive Data from
  - Unauthorized disclosure
  - Unauthorized modification
  - Denial of service attacks
- Security Controls
  - Security Policy
  - Access control models
  - Integrity protection
  - Privacy problems
  - Fault tolerance and recovery
  - Auditing and intrusion detection

Data Confidentiality

- Access control
  - which data users can access
- Information flow control
  - what users can do with the accessed data
- Inference Attacks
- Data Mining

Access Control

- Ensures that all direct accesses to object are authorized
- Protects against accidental and malicious threats by regulating the read, write and execution of data and programs
- Requires:
  - Proper user identification
  - Information specifying the access rights is protected form modification

Access Control

- Access control components:
  - Access control policy
    - specifies the authorized accesses of a system
  - Access control mechanism
    - implements and enforces the policy

Access Control Models

- How to describe the policies?
  - “Who can access what?”
  - Subject: active entity that requests access to an object
    - - e.g., user or program
  - Object: passive entity accessed by a subject
    - - e.g., record, relation, file
  - Access right (privileges): how a subject is allowed to access an object
    - - e.g., subject s can read object o

Access Control Models

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)

Mandatory Access Control- MAC

- Motivated by government in late 1980’s/early 1990’s
- Utilize security classifications
  - TS: Top Secret, S: Secret, C: Classified, U: Unclassified
  - TS > S > C > U
- Each subject and object are classified into one of the security classifications (TS, S, etc.)
- Bell-LaPadulla properties (restrictions on data access)
  - simple property: No READ UP
  - star (*) property: No WRITE DOWN (write at own level)

MLS

- Multilevel relational (MLS) schema
  - classification attribute C
  - tuple classification TC
  - R(A_1, C_1, A_2, C_2, ...A_n, C_n, TC) *Jajodia-Sandhu*

MLS Relation Example

Vessel	Objective	Destination
Micra	Shipping	Moon
Vision	Spying	Saturn
Avenger	Spying	Mars
Logos	Shipping	Venus

MLS Relation Example

Vessel	Objective	Destination
Micra	Shipping	Moon
Vision	Spying	Saturn
Avenger	Spying	Mars
Logos	Shipping	Venus

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S

MLS

- Read from MLS tables
  - Access to each record is determined by TC
  - Level U sees first 2 tuples
  - Level C sees first 3 tuples
  - Level S sees all tuples

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S

MLS Insert

- What if a U user wants to insert a tuple with vessel = Avenger?
- If reject the insert – what will happen?
  - Covert channel
- If insert another Avenger, what about the primary key? Will have 2 Avengers
  - PK + Classification

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S

MLS Insert

- What if a U user wants to insert a tuple with vessel = Avenger?
- If reject the insert – what will happen?
  - Covert channel
- If insert another Avenger, what about the primary key? Will have 2 Avengers
  - New primary key: PK + Classification

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S

MLS Insert

- What if a U user wants to insert a tuple with vessel = Avenger?
- If reject the insert – what will happen?
  - Covert channel
- If insert another Avenger, what about the primary key? Will have 2 Avengers
  - New primary key: PK + Classification

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S
Avenger	U	Shipping	U	Mars	U	U

MLS Update

- What if the S level wants to update one of the tuples at the U level?
  - U cannot see the update
  - Replicate the tuple

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S
Avenger	U	Shipping	U	Mars	U	U

MLS Update

- What if the S level wants to update one of the tuples at the U level?
  - U cannot see the update
  - Replicate the tuple
    - Replicate (Vision, ………., U)
    - Update it to (Vision, ……….., S)

Vessel	C-V	Objective	C-O	Destination	C-D	TC
Micra	U	Shipping	U	Moon	U	U
Vision	U	Spying	U	Saturn	U	U
Avenger	C	Spying	C	Mars	C	C
Logos	S	Shipping	S	Venus	S	S
Avenger	U	Shipping	U	Mars	U	U
Vision	U	Spying	U	Venus	S	S

MLS

- Wrap up
  - Simple
  - Easy to manage: just assign a security classification to each user
  - The security classifications are used in government and military
  - MLS is widely adopted in government and military.

Discretionary Access Control (DAC)

- All commercial DB systems adopt DAC
- Current discretionary authorization models for relational DBMS are based on the System R authorization model
    - P. P. Griffiths and B. W. Wade, "An Authorization Mechanism for a Relational Database System," *ACM Transactions on Database Systems (TODS)* Volume 1 Issue 3, Sept. 1976.
- It is based on ownership administration with administration delegation

Discretionary Access Control (DAC)

- For each subject access right to the objects are defined
  - (subject, object, +/- access mode)
  - E.g. (Black, Employee-relation, read)
- Based on granting and revoking privileges
- Assign privileges
  - to account level (subject)
    - independent of the relations
    - create schema, create table, create view
  - on relation level (object)
    - on a particular base relation or view

Authorization ID's

- A user is referred to by *authorization ID*, typically their login name.
- There is an authorization ID PUBLIC.
  - Granting a privilege to PUBLIC makes it available to any authorization ID.

- The objects on which privileges exist include stored tables and views.
- Other privileges are the right to create objects of a type, e.g., triggers.

Privileges

- A file system identifies certain privileges on the objects (files) it manages.
  - Typically read, write, execute.
- SQL identifies a more detailed set of privileges on objects (relations) than the typical file system.
- Nine privileges in all, some of which can be restricted to one column of one relation.

Privileges

- Some important privileges on a relation:
  1. SELECT = right to query the relation.
     - May apply to only one attribute.
  2. INSERT = right to insert tuples.
  3. DELETE = right to delete tuples.
  4. UPDATE = right to update tuples.
     - May apply to only one attribute.

Granting Privileges

- You have all possible privileges on the objects, such as relations, that you create.
- You may grant privileges to other users (authorization ID’s), including PUBLIC.
- You may also grant privileges WITH GRANT OPTION, which lets the grantee also grant this privilege.

The GRANT Statement

- To grant privileges, say:

  GRANT <list of privileges>

  ON <relation or other object>

  TO <list of authorization ID’s>;

- If you want the recipient(s) to be able to pass the privilege(s) to others add:

  WITH GRANT OPTION

Example: GRANT

- Suppose you are the owner of Sells. You may say:

```
GRANT SELECT, UPDATE(price)
ON Sells
TO sally;
```

- Now Sally has the right to issue any query on Sells and can update the price component only.

Example: Grant Option

- Suppose we also grant:

```
GRANT UPDATE ON Sells TO sally
WITH GRANT OPTION;
```

- Now, Sally not only can update any attribute of Sells, but can grant to others the privilege UPDATE ON Sells.
  - Also, she can grant more specific privileges like `UPDATE(price)ON Sells`.

Revoking Privileges

```
REVOKE <list of privileges>
ON <relation or other object>
FROM <list of authorization ID’s>;
```

- Your grant of these privileges can no longer be used by these users to justify their use of the privilege.
  - But they may still have the privilege because they obtained it independently from elsewhere.

REVOKE Options

- We must append to the REVOKE statement either:
  1. CASCADE. Now, any grants made by a revokee are also not in force, no matter how far the privilege was passed.
  2. RESTRICT. If the privilege has been passed to others, the REVOKE fails as a warning that something else must be done to “chase the privilege down.”

RBAC

- Role-based access control (RBAC)
- Sandhu, R., Coyne, Feinstein, Youman: “Role-Based Access Control Models,” *IEEE Computer*, 29 (2): 38–47.
  - Semantic construct
  - System administrator creates roles according to job functions

RBAC

- Role
  - Specific task competency
  - duty assignments
  - Embody authority and responsibility
- Grant permissions to users in these roles
  - Roles & permissions
  - Users & roles

Motivation

- Roles define individuals and extent of resource access
- Combination of users and permissions can change
  - E.g. user membership in roles
- Permissions associated with roles stable
- Administration of roles rather than permissions
- Role permission predefined
  - Easier to add/remove users membership than create new roles/permissions
- Roles part of SQL3
- Supported by many software products
  - Roles used in Windows NT, XP (system admin)

RBAC basics

- Access control in RBAC exists in:
  - Role-permission (stable)
  - User-role (dynamic)
  - Role-role relationships (stable)
- RBAC supports principles:
  - Least privilege
  - Separation of duties- mutually exclusive roles
  - Data abstraction- abstract permissions (not just R/W)
- Limitations
  - RBAC cannot enforce way principles applied – system admin could configure to violate

Constraints

- Mutually exclusive roles
  - User at most 1 role in ME set
  - Combinations of roles and permissions can be prohibited
- Cardinality
  - Maximum number of members in a role
  - Minimum cardinality difficult to implement
- Prerequisite role
  - User assigned to role B, only if assigned to A
  - Permission p assigned to role only if role has permission q

DAC, MAC vs. RBAC

- DAC vs. MAC emerged from defense security research
- RBAC independent of access control
- RBAC can be used to implement DAC, MAC

Database Encryption

- Application level encryption
- Database encryption

- Protect keys

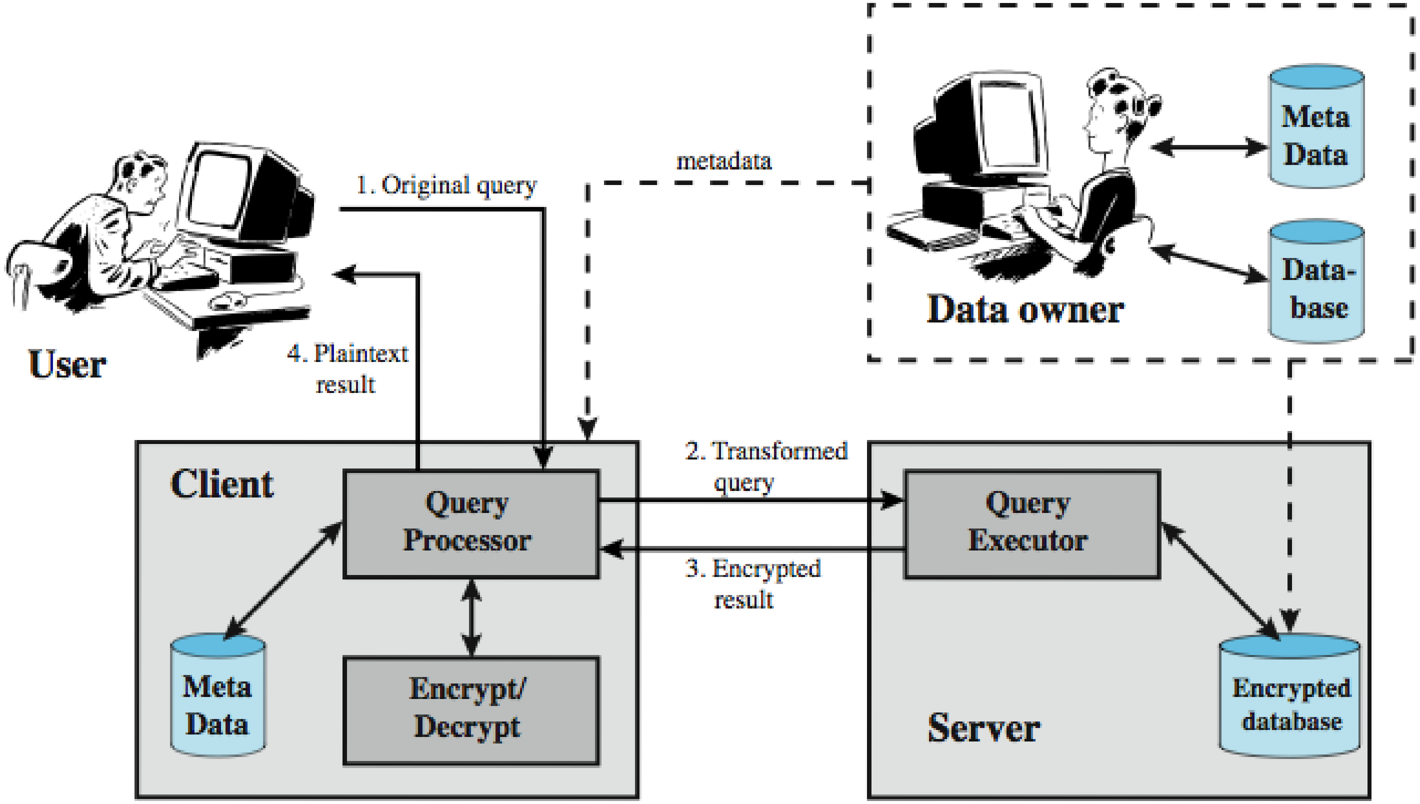
Application Level Encryption

- Data protected in database & storage
- Data protected in use and transit
- Programming needed in the applications
- Power of database limited -indexing, searching, stored procedures
- All access must go through application
- Key management

Database Encryption

- Protects data as it is written to and read from a database
- Secures data in the file-system used by database
- Enables field level encryption
- Transparent to applications
- Watch data in transit

Database Encryption



Limit the amount of encrypt/decrypts

- Three Important Techniques
  - Column level encryption
  - Search without decrypting data
  - Conduct data operations without decrypting data

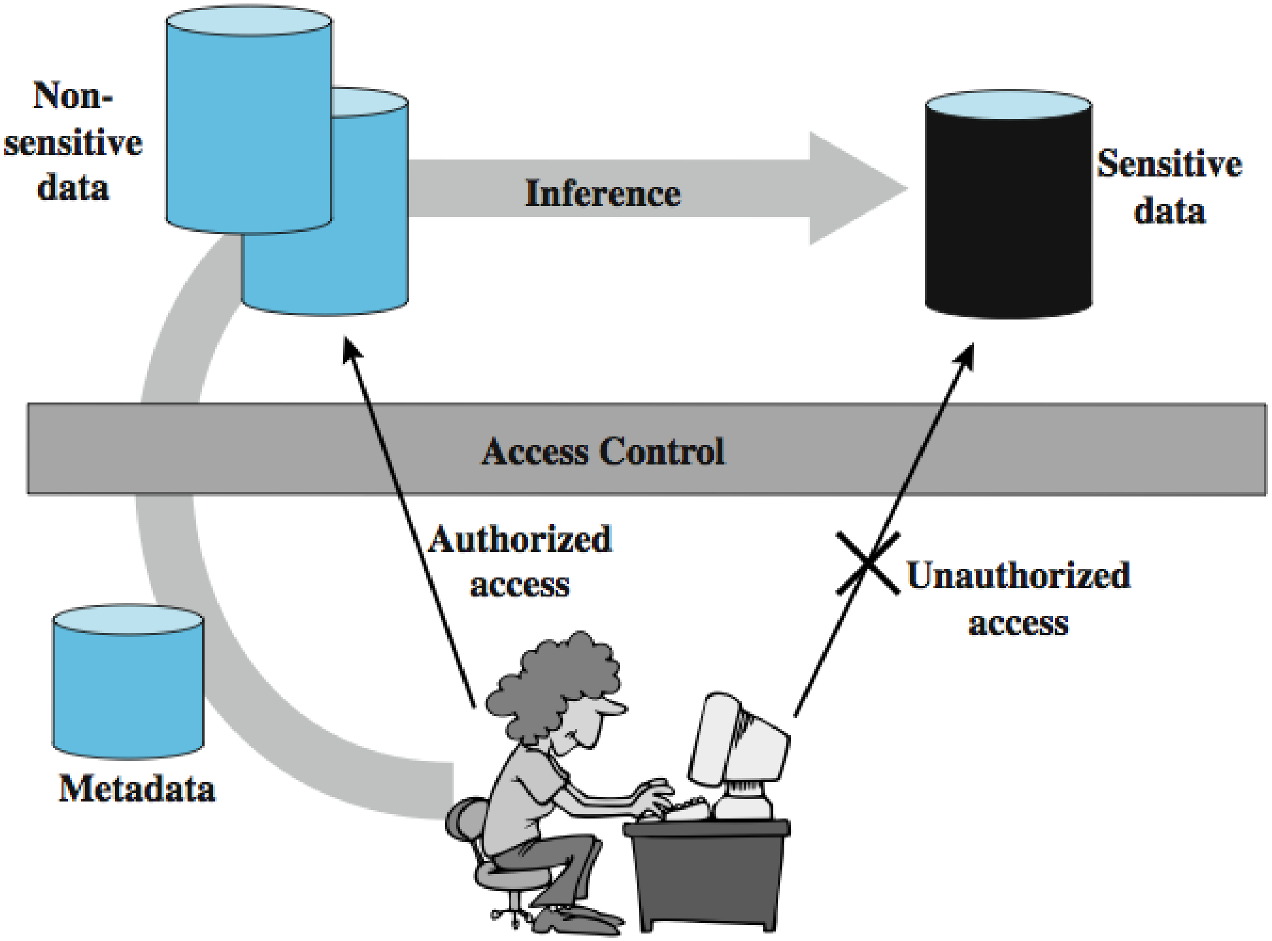
Sharing Non-Sensitive Data

- A database may contain both sensitive (e.g. salary) and non-sensitive (e.g. name) data.
- Security Goal: To disclose only non-sensitive data.
  - In some cases: aggregated sensitive data (e.g. average salary)
- Precision Goal: To protect all sensitive data while disclosing as much non-sensitive data as possible.
- Ideal combination is to maintain perfect security with maximum precision.

Sensitive Data Disclosure

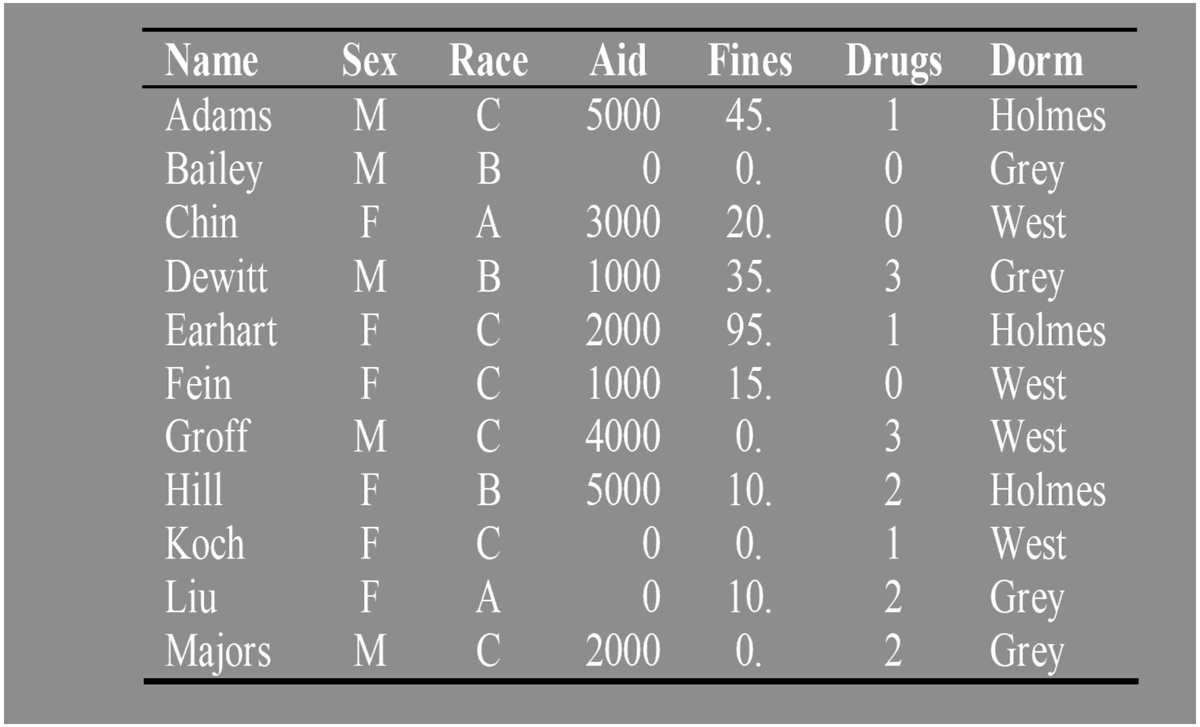
- Exact data: exact value of a sensitive data item
- Bounds: knowing the lower and upper bounds values of a sensitive data item
- Negative Result: queries may be made to determine a negative result from which sensitive data may be disclosed (e.g. Alice’s salary is not lower than 150K/year).
- Existence: the existence of data is itself a sensitive piece of data
- Probable value: it may be possible to determine the probability that a certain element has a certain value.

The Inference Problem



The Inference Problem

- Inference problem is to infer or derive sensitive data from non-sensitive data.
- Sample table with 3 sensitive fields: Aid, Fines and Drugs when disclosed for a given individual



Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

Direct Attack

```
SELECT Name
FROM Sample
WHERE Sex=‘M’ AND Drugs=1;

SELECT Name
FROM Sample
WHERE (Sex=‘M’ AND Drugs=1)
    OR (Sex<>’M’ AND Sex<>’F’)
    OR Dorm=‘Ayres’;
```

To confuse the DB

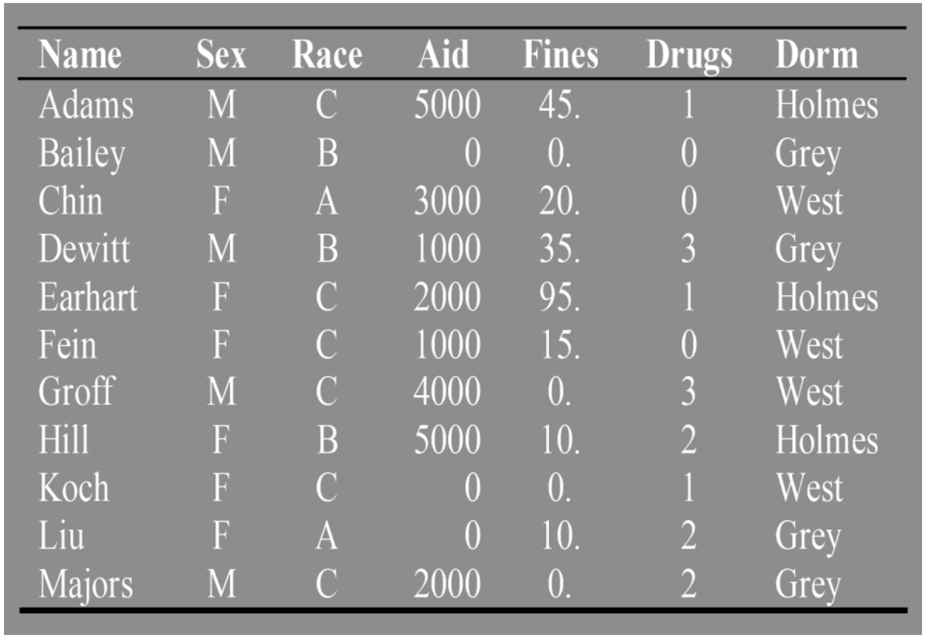
Indirect Attack

- Statistical databases (queries)
- provides data of a statistical nature
  - e.g. counts, averages
- two types:
  - pure statistical database
  - ordinary database with statistical access
    - some users have normal access, others statistical
- access control objective to allow statistical use without revealing individual entries
- security problem is one of inference
- **Indirect attack - infer a result based on one or more statistical results.**

Indirect Attack

- **Sum**: a reported sum may be used to infer a value.

SELECT Dorm, Sex, SUM(Aid) As Sum_Aid
FROM Sample
Group BY Dorm, Sex;



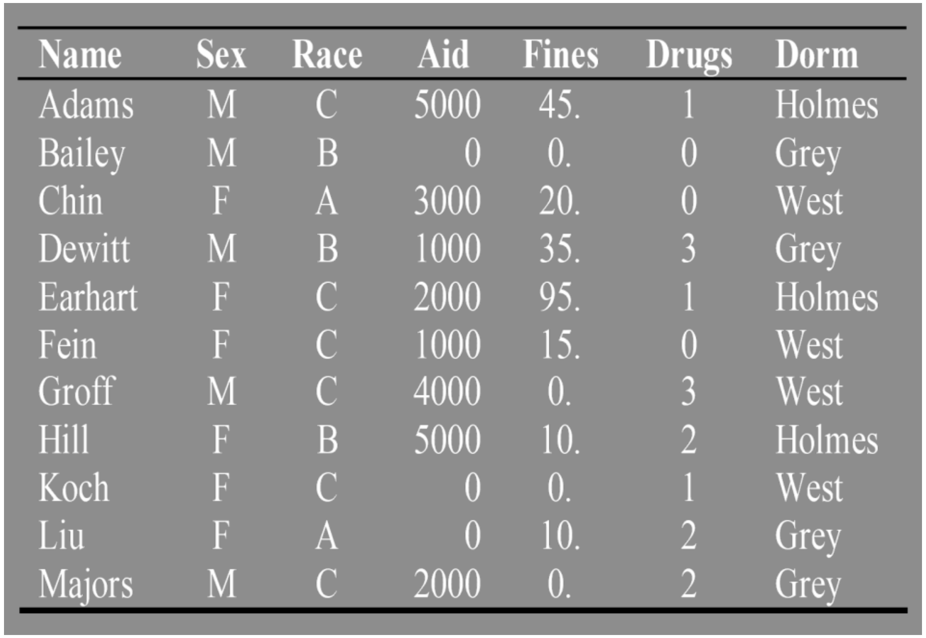
Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

Dorm	Sex	Sum_Aid
Holmes	M	5000
Holmes	F	7000
Grey	M	3000
Grey	F	0
West	M	4000
West	F	4000

Indirect Attack

- **Sum**: a reported sum may be used to infer a value.

```
SELECT  Dorm, Sex, COUNT(*) As    Tot_Students
FROM  Sample
Group BY Dorm, Sex;
```

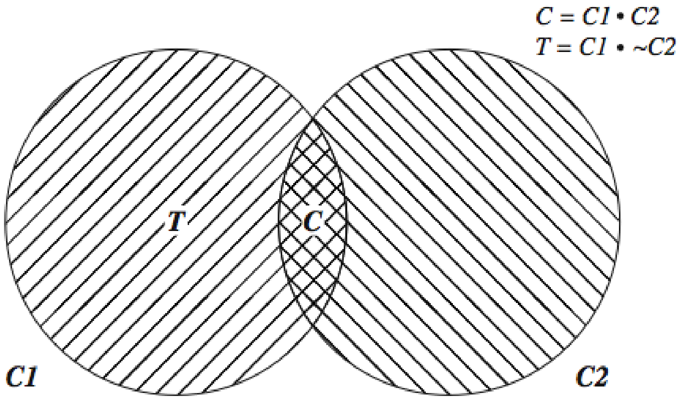


Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

Dorm	Sex	Tot_Students
Holmes	M	1
Holmes	F	2
Grey	M	3
Grey	F	1
West	M	1
West	F	3

Tracker Attacks

- divide queries into parts
  - C = C1 AND C2
  - count(C) = count(C1) - count (C1 AND ~*C2)*
- combination is called a tracker
- each part acceptable query size
- overlap is desired result



Tracker Attacks

- Tracker attacks: Generate the desired data by using additional queries that generate small results.

  SELECT COUNT(*)
  FROM Sample
  WHERE Sex = ‘F’ AND Race = ‘C’ AND Dorm = ‘Holmes’;

- Use rules of logic and algebra to rewrite query:

  count (a AND b AND c)
  = count(a) - count(a AND NOT (b AND c))
  = count(a) - count(a AND (NOT b OR NOT c))

Tracker Attacks

```
SELECT COUNT(*)
FROM Sample
WHERE Sex = ‘F’;

SELECT COUNT(*)
FROM Sample
WHERE Sex = ‘F’ AND ((Race  <>‘C’) OR
   (Dorm <> ‘Holmes’));
```

Controls for Inference Attacks

- Query controls
  - Limit overlap between new and previous queries
- Partitioning
  - Cluster records into exclusive groups and only allow queries on entire groups
- Item controls
  - Suppression: query is rejected without sensitive data provided.
  - Concealing: the answer provided is close to but not exactly the actual answer.

Controls for Inference Attacks

- No perfect solution
- Three paths to follow:
  - Suppress obvious sensitive information (easily).
  - Track what the user knows (costly).
    - They are used to limit queries accepted and data provided.
  - Disguise the data (problem with precision).
    - It’s applied only to the released data.