CS5322 Database Security

Limitations with DAC

- Global policy:
  - DAC lets users to decide the access control policies on their data, regardless of whether those policies are consistent with the global policies.
- Information flow:
  - Information can be copied from one object to another.
  - The owner of the original may not have control over the copy.
  - This is a major concern in certain domains, e.g., military.
- Mandatory access control (MAC) could mitigate these limitations

Mandatory Access Control (MAC)

- In DAC, users have the discretion to specify policy themselves
- In MAC, a system-wide policy decides who is allowed to have access; individual users cannot alter the policy
- There are several different models for MAC
  - We will consider multi-level security (MLS)

Multi-Level Security (MLS)

- Three key concepts related to objects: security level, compartment, and label
  - Each object has a security level that indicates its sensitivity
    - E.g., unclassified, confidential, secret, top secret
  - Each object may belong to some compartments (i.e., categories)
    - E.g., finance, manufacture, agriculture
  - The security level and compartment of an object form the label of the object
- Example: a label (confidential, {finance, Europe}) means that
  - The object concerns finance in Europe
  - Its security level is confidential

Multi-Level Security (MLS)

- Each subject also has a label
- Example: If a user has a label (secret, {finance, Asia}), then
  - She has clearance to access secret documents concerning finance and/or Asia
- Multi-level Security:
  - A subject S may access an object O, if S’s has clearance to access information in O’s compartments and at O’s security level
- Example: A user with a label (secret, {finance, Asia})
  - Can read a document with a label (confidential, {Asia})
  - Cannot read a document with a label (confidential, {finance, Europe})

Multi-Level Security (MLS)

- More formally, a subject with a label can read an object with a label , if and only
  - i.e., *dominates*
- means:
  - The security level of is lower than or equal to that of
  - The set of compartments in is a subset of or equal to the set of compartments in
- Example: (confidential, {Asia}) (secret, {Asia, finance}), because
  - confidential <= secret
  - {Asia} {Asia, finance}
- Example 2: (confidential, {}) (secret, {finance})
- Example 3: (confidential, {Asia, finance}) (secret, {finance})

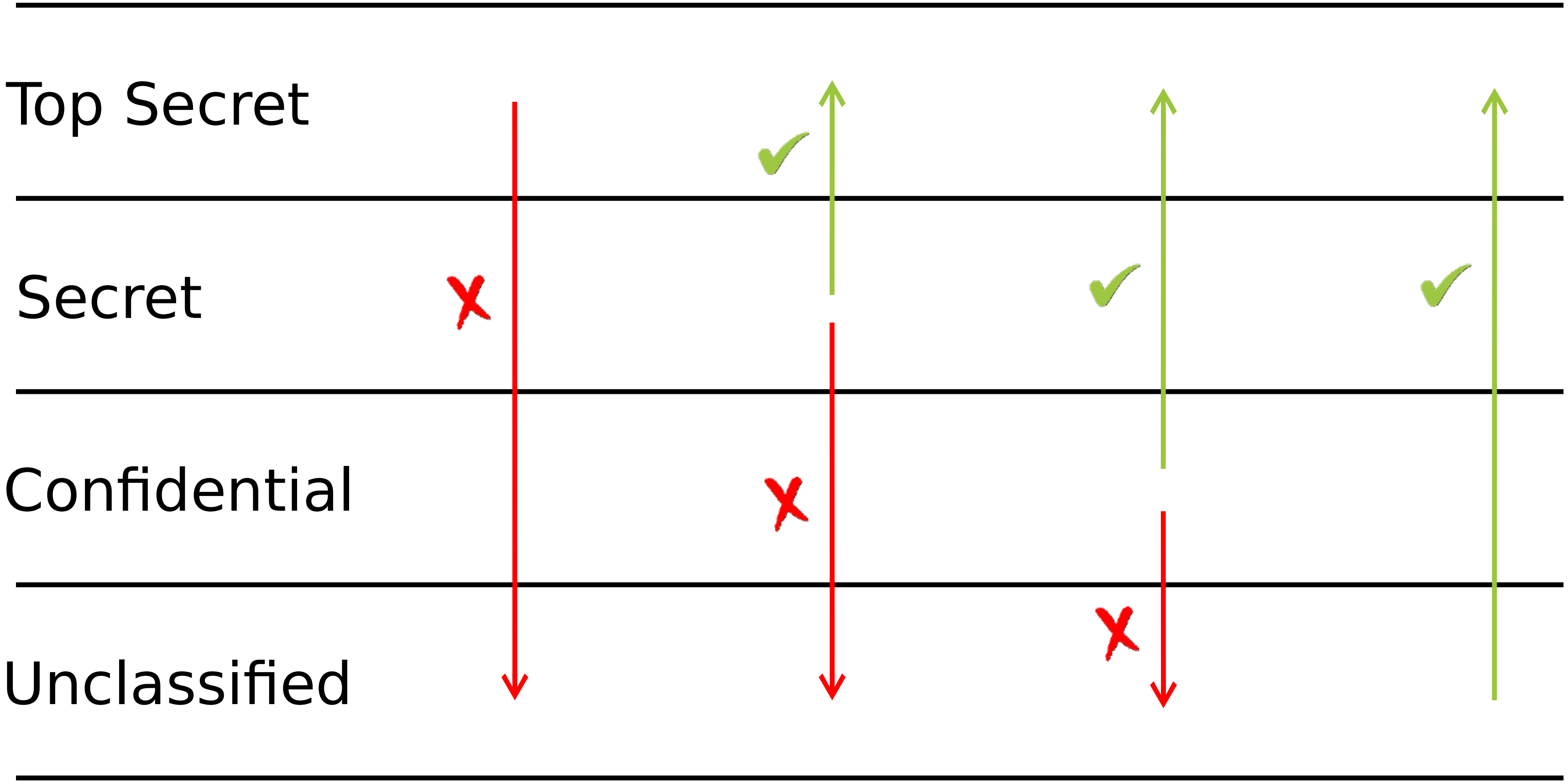
BLP Model

- [Bell and LaPadula 1973] proposes a formal mathematical model of MLS
- Prove that information cannot leak to subjects not cleared for it, if the following two properties are ensured:
    - "No read up": *S* can read *O* iff
        - This is intuitive
    - "No write down": *S* can write *O* iff
        - This may seem counter-intuitive
- Why no write down?
- To prevent illegal information flow

BLP Model

- Suppose that a user account, President, has clearance to access top secret documents
- If we allow President to write unclassified documents, then the following information flow is possible:
  - President reads top secret document T
  - President writes information from T to an unclassified document U (intentionally or unintentionally)
  - A user without top-secret clearance reads U
- The “no write down” rule prevents this

Information Flow under BLP



Top Secret

Secret

Confidential

Unclassified

Applying MAC to Databases

- Idea:
  - Attach a label to each database object and subject
  - Conduct access controls based on the labels
- Possible granularities of access control:
  - One label for each table
  - One label for each tuple
    - This is a common choice in commercial databases
  - One label for each value in a tuple
    - We will consider this case

Multilevel (ML) Relations

- A relation $R(A_1, A_2, \ldots, A_d)$ is extended to an ML relation $R'(A_1, C_1, A_2, C_2, \ldots, A_d, C_d, TC)$, where
  - C_i is an attribute that represents the security levels associated with A_i
  - TC is an attribute that represents the security levels associated with the tuples
  - For a tuple, the value of TC should be the highest security level among all of its attributes
- Example:
  - A relation Grades(Name, Gender, Grade) is extended to Grades'(Name, C_1, Gender, C_2, Grade, C_3, TC)
  - A tuple (Alice, female, 90) can be extended to (Alice, unclassified, female, unclassified, 90, confidential, confidential)
  - A tuple (Bob, male, 40) can be extended to (Bob, unclassified, male, unclassified, 40, secret, secret)

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	C	Female	C	70	C	C
Dave	S	Male	S	60	S	S

- We use the following notations:
  - U: unclassified
  - C: confidential
  - S: secret
  - TS: top secret
- U < C < S < TS

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	C	Female	C	70	C	C
Dave	S	Male	S	60	S	S

- U < C < S < TS
- Level U users can see only the first two tuples
- Level C users can see only the first three tuples
- Level S users can see only the first four tuples

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	U	Male	C	60	S	S

- U < C < S < TS
- Now what can level U users see?
- Intuitively, we should let them see Cath and Dave’s names, but not the other information
- Let’s use NULL for this purpose

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	U	Male	C	60	S	S

- What level U users see:

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U
Dave	U	NULL	U	NULL	U	U

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	U	Male	C	60	S	S

- Level C users see…

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	U	Male	C	60	S	S

- Level C users see…

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	U	Male	C	NULL	U	C

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- Now level U users see...

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- Now level U users see…

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U

Multilevel (ML) Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- Why don’t level U users see the following?
- Because the primary key Name can never have NULL

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U
NULL	U	NULL	U	NULL	U	U

Tricky Issue in ML

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- Suppose that a level U user sees the table below, and tries to insert (Dave, U, Male, U, 100, U, U)
- What should we do?

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U

Tricky Issue in ML

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- A level U user inserting (Dave, U, Male, U, 100, U, U)
- Option 1: Deny the insertion
- Problem: the user learns that a higher-level tuple for Dave exists

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U

Tricky Issue in ML

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- A level U user inserting (Dave, U, Male, U, 100, U, U)
- Option 2: Modify the C-level tuple for Dave
- Problem: Dave’s grade is changed to 100!

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U

Tricky Issue in ML

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S

- A level U user inserting (Dave, U, Male, U, 100, U, U)
- Option 3: Insert the tuple while keeping the C-level tuple
- Problem: Two tuples have the same primary key
- Solution: Polyinstantiation

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	NULL	U	NULL	U	U

Polyinstantiation: Intuition

- Intuitive idea: Use the original primary key + TC as the new primary key
  - Note: later we will see that even this is not enough
- Example below: (Name, TC) as the new primary key
- As such, we may have different instances of the same tuple for different security levels
- This works, but will make things a lot more “interesting”

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Suppose that we want to update the second tuple as (Bob, U, Male, U, 100, C, C)
- Option 1: Directly modify the second tuple
- Problem: Level U users would learn that Bob’s grade now has a higher security level
- Solution: Keep the second tuple while inserting (Bob, U, Male, U, 100, C, C)

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Suppose that we want to insert a new tuple (Alice, U, Female, U, 100, S, S)
- Should it be denied due to the first tuple?
- No; recall that the primary key is Name+TC
- The tuple will be inserted

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Suppose that a level S user executes DELETE FROM Grades WHERE Name = ‘Alice’
- What will happen?
- Nothing will happen, since Alice’s tuple needs to remain for level U and level C users

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Now suppose that we want to update the 4-th tuple to (Dave, C, Female, S, 60, S, S)
- Option 1: Change the 4-th tuple accordingly
- Problem: Level C users would learn that Dave’s gender now has a higher security level

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Now suppose that we want to update the 4-th tuple to (Dave, C, Female, S, 60, S, S)
- Option 2: Deny the update
- Problem: Legitimate updates should not be denied

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Issues in Polyinstantiation

- Now suppose that we want to update the 4-th tuple to (Dave, C, Female, S, 60, S, S)
- Option 3: Insert (Dave, C, Female, S, 60, S, S) into the table
- Problem: This violates the primary key constraint since there is already a tuple with primary key (Dave, S)
- Solution: Do not use (Name, TC) as the primary key
  - Instead, use (Name, C1, C2, C3, TC) as the primary key

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
Bob	U	Male	U	80	U	U
Cath	U	Female	C	70	C	C
Dave	C	Male	C	60	S	S
Dave	U	Male	U	100	U	U

Formalization of Polyinstantiation

- Towards a Multilevel Secure Relational Data Model. Proceedings of the ACM International Conference on Management of Data (SIGMOD), pages 50-59, 1991.
- A more sophisticated version of polyinstantiation that uses the original primary + all security levels as the new primary key
- Allows instances like the following

Name	C1	Gender	C2	Grade	C3	TC
Sam	C	Female	U	60	U	C
Sam	C	Male	C	60	U	C
Sam	C	Female	U	90	C	C
Sam	C	Male	C	90	C	C

Formalization of Polyinstantiation

- Which version is used more frequently in practice?
- Neither version is used much
  - Probably because of their complexities

Name	C1	Gender	C2	Grade	C3	TC
Sam	C	Female	U	60	U	C
Sam	C	Male	C	60	U	C
Sam	C	Female	U	90	C	C
Sam	C	Male	C	90	C	C

Multilevel Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
NULL	U	Male	U	80	U	U
Cath	S	Female	C	70	C	C
Alice	U	NULL	U	NULL	U	U

- Can you identify all problems in the above multilevel relation?

Multilevel Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
NULL	U	Male	U	80	U	U
Cath	S	Female	C	70	C	C
Alice	U	NULL	U	NULL	U	U

- Problem 1:
  - The second tuple has NULL for Name, which is the primary key
- Problem 2:
  - The TC value of the third tuple is C, and yet, the Name value of the tuple has level S

Multilevel Relations

Name	C1	Gender	C2	Grade	C3	TC
Alice	U	Female	U	90	U	U
NULL	U	Male	U	80	U	U
Cath	S	Female	C	70	C	C
Alice	U	NULL	U	NULL	U	U

- Problem 3:
  - Level C users see the third tuple as (NULL, C, Female C, 70, C, C), which is incorrect since the primary key cannot be NULL
- Problem 4:
  - The first and fourth tuples have the same (Name, C1, C2, C3, TC) combination