Database Security – Term Project (Phase 2)

Secure Student Records Management System

Part A: Core Project (15 Marks)

1. Project Overview

The **Secure Student Records Management System (SRMS)** project is designed to help students apply advanced database security concepts in a realistic academic setting. The system manages highly sensitive academic data such as student profiles, grades, attendance records, and staff information.

Students must design and implement a secure system that incorporates:

- **Access Control (RBAC)**
- **Inference Control**
- **Flow Control**
- **Multilevel Security (MLS)**
- **Encryption (At Rest)**

- Students must also develop a functional GUI application using any technology from the following (Windows Forms, WPF, JavaFX, Python Tkinter, or any GUI technology) connected securely to SQL Server.

- Different users (Admin, Instructor, TA, Student, Guest) must only access operations allowed by their **role** and **clearance**.

2. Project Objectives

1. Design a secure SQL Server database schema.
2. Develop a GUI desktop application using ADO.NET/C# or any GUI technology.
3. Enforce Access Control via SQL roles and application-level RBAC.
4. Implement Inference Control to prevent deduction of sensitive data.
5. Apply Flow Control to prevent illegal movement of classified data.
6. Implement Multilevel Security (MLS) using clearance levels and Bell–LaPadula.
7. Secure all sensitive data using encryption.

3. Required Database Schema (Detailed)

The following schema is the minimum requirement and must be enforced with appropriate security classifications and constraints.

3.1 STUDENT (Confidential)

Column	Type	Description	Security
Student ID (PK)	INT	Unique student ID	AES Encryption
Full Name	NVARCHAR(100)	Student name	Confidential
Email	NVARCHAR(100)	Contact	Confidential
Phone	NVARCHAR(20)	Phone number	Encrypt At Rest
DOB	DATE	Birthdate	Confidential
Department	NVARCHAR(50)	Academic department	Unclassified
Clearance Level	INT	MLS level	Required for MLS

3.2 INSTRUCTOR (Confidential)

Column	Type	Security
Instructor ID (PK)	INT	Confidential
Full Name	NVARCHAR(100)	Confidential
Email	NVARCHAR(100)	Confidential
Clearance Level	INT	MLS level

3.3 COURSE (Unclassified)

Column	Type	Security
Course ID (PK)	INT	Public
Course Name	NVARCHAR(100)	Unclassified
Description	NVARCHAR(MAX)	Unclassified
Public Info	NVARCHAR(MAX)	Visible to Guest

3.4 GRADES (Secret)

Column	Type	Security
Grade ID (PK)	INT	
Student ID (FK)	INT	AES encrypted
Course ID (FK)	INT	Secret
Grade Value	DECIMAL	Encrypt At Rest
Date Entered	DATETIME	Instructor-only

3.5 ATTENDANCE (Secret)

Column	Type	Security
Attendance ID (PK)	INT	
Student ID (FK)	INT	
Course ID (FK)	INT	
Status	BIT	
Date Recorded	DATETIME	Secret

3.6 USERS (Authentication Table)

Column	Type	Notes
Username (PK)	NVARCHAR(50)	Encrypted
Password	VARBINARY	Encrypted
Role	NVARCHAR(20)	Admin/Instructor/TA/Student/Guest
Clearance Level	INT	Enforces MLS

4. Security Requirements (Five Major Models) Each model is worth **2 marks**

4.1 Access Control (RBAC)

Mandatory Requirements

- Create SQL roles: Admin, Instructor, TA, Student, Guest
- Use `GRANT`, `REVOKE`, `DENY`
- Enforce role-based visibility in the GUI
- Every operation must call a stored procedure that verifies the user's role

Security Matrix

Function / Data	Admin	Instructor	TA	Student	Guest
View own profile	✓	✓	✓	✓	✕
Edit own profile	✓	✓	✓	✕	✕
View grades	✓	✓	✕	✕	✕
Edit grades	✓	✓	✕	✕	✕
View attendance	✓	✓	✓	✓ (own)	✕
Edit attendance	✓	✓	✓	✕	✕
Manage users	✓	✕	✕	✕	✕
View public course info.	✓	✓	✓	✓	✓

4.2 Inference Control

Mandatory Requirements

- Implement *Query Set Size Control* (minimum group size = 3)
- Create restricted views for TA/Student
- Block aggregate results that reveal identity

4.3 Flow Control

Flow Control prevents data from moving **downward** from higher to lower classifications, avoiding leaks.

Mandatory Requirements

1. Prevent Down flow

- Top Secret → Secret/Confidential/Unclassified (blocked)
- Secret → Confidential/Unclassified (blocked)

2. GUI Flow Restrictions

- Block export of Secret/Top Secret data (**Bonus** **+1**)

  "Prevent users from downloading, exporting, saving, printing, or copying highly classified data out of the system."

- Disable copy/paste for high-classification panels (**Bonus** **+1**)

  When a user views Secret or Top Secret information (e.g., grades, disciplinary records), the system must block Copying classified data (BLOCK)

4.4 Multilevel Security (MLS)

MLS must follow **Bell–LaPadula** principles:

Mandatory Requirements

1. **No Read Up (NRU):** Cannot read higher classification
2. **No Write Down (NWD):** Cannot write to lower classification(**Bonus +1**)

> *"A user with a higher security clearance is NOT allowed to write data into a location (table, column, file) that has a LOWER security classification."*

Implementation Requirements

- Assign clearance levels to all users and records
- Use views based on classification
- Stored procedures must enforce classification checks

4.5 Encryption

Mandatory Requirements

Encryption at Rest

Use AES functions:

- `EncryptByKey()`
- `DecryptByKey()`

Sensitive data to encrypt:

- Grades
- Student ID / Phone
- Passwords

5. GUI Requirements Students must build a fully functional role-based GUI.

5.1 Login Form

- Username / password authentication
- Detection of role + clearance
- Secure login using hashing

5.2 Admin GUI

- Manage users
- Assign/edit roles

5.3 Instructor GUI

- Enter/view grades
- View attendance
- Access Secret-level data only

5.4 TA GUI

- Manage attendance
- View only student data registered in courses assigned of TAs (Confidential)
- No access to grades

5.5 Student GUI

- View own profile
- View own attendance
- View published grades

5.6 Guest GUI

- View only public course information

6. Deliverables

Students must submit:

1. **Complete VS Solution (.sln)**
2. **SQL Server backup (.bak)**
3. **Documentation Report (PDF/Word)** including:
   - Screenshots
   - Test cases for each security model
   - Schema description
4. **5-minute demo video**

7. Grading Rubric (15 Marks + 3 Bonus)

Component	Marks	Bonus	Notes
Access Control	2		**Full marks for each security module will be awarded not only for correct implementation, but also for clearly**
Inference Control	2		
Flow Control	2	+2	

Component	Marks	Bonus	Notes
MLS	2	+1	**presenting, explaining, and discussing the module during the project demonstration**.
Encryption	2		
GUI Application	4		
Documentation + Video	1		

Total = 15 Marks + Up to 3 Bonus Marks

8. Final Student Responsibilities

Every student group must:

- Design and secure the full database schema
- Implement all five security mechanisms
- Build a functional GUI application
- Enforce RBAC, MLS, Flow/Inference Control
- Use AES encryption properly
- Document and present their work clearly

Part B: 10 Marks

- The Role Request Workflow is an additional security feature that simulates how real organizations handle privilege escalation.

- It allows users (especially students) to request higher privileges, while ensuring that the Admin controls all role changes.

1. A Student Can Submit a Role Upgrade Request [5 Marks]

➢ A **Student** user (or any lower-privileged user) may want to request an upgrade such as:

- Student → TA
- TA → Instructor

➢ The GUI should include a page/form: **“Request Role Upgrade”**

The student selects:

- The role they want
- A reason/justification
- Optional comments

- The request is saved into a **Role Requests** table

- The request is assigned a status:

    - Pending
    - Approved
    - Denied

- A timestamp and user ID are recorded. No changes to actual roles occur automatically

- This ensures the student cannot upgrade themselves

2. Admin Dashboard Shows Pending Requests [5 Marks]

A dedicated interface in the Admin GUI: **“Pending Role Requests”**

This table/list must show:

- Username
- Current role
- Requested role
- Reason
- Date submitted
- Status (Pending)

Admin action options:

For each request, Admin can click:

- **Approve**
- **Deny**

- If approved:
    - The user’s role in the Users table is updated
    - Update Users. Role to the new role
    - Mark the request as “Approved”
- If denied:
    - Only the request status changes
    - Mark the request as “Denied”
    - Keep the user’s role unchanged

Hint: Part B serves as an alternative to the final practical laboratory examination, and no separate practical exam will be conducted.