Chapter 3````x

Database Integrity, Security and Recovery

Outline

- Database integrity concepts and subsystem
- Integrity constraints
- Types of constraints
- Database security
- Database threats
- Threats Identification and Authentication
- Database Access control
- Categories of access control
- Implementation of security system
- Data Encryption

Database integrity

- Data integrity refers to the overall completeness, accuracy and consistency of data in database.
- A good database will enforce data integrity whenever possible.
- Data integrity used to restrict unnecessary data entire to database.
- For example, a user could accidentally try to enter a phone number into a date field.
- If the system enforces data integrity, it will prevent the user from making these mistakes. ***For example: in customer database we can enforce an integrity that it must accept the customer only from Ethiopia or other only specific input.***
- data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle.
- integrity aims to prevent the unintentional changes to information.

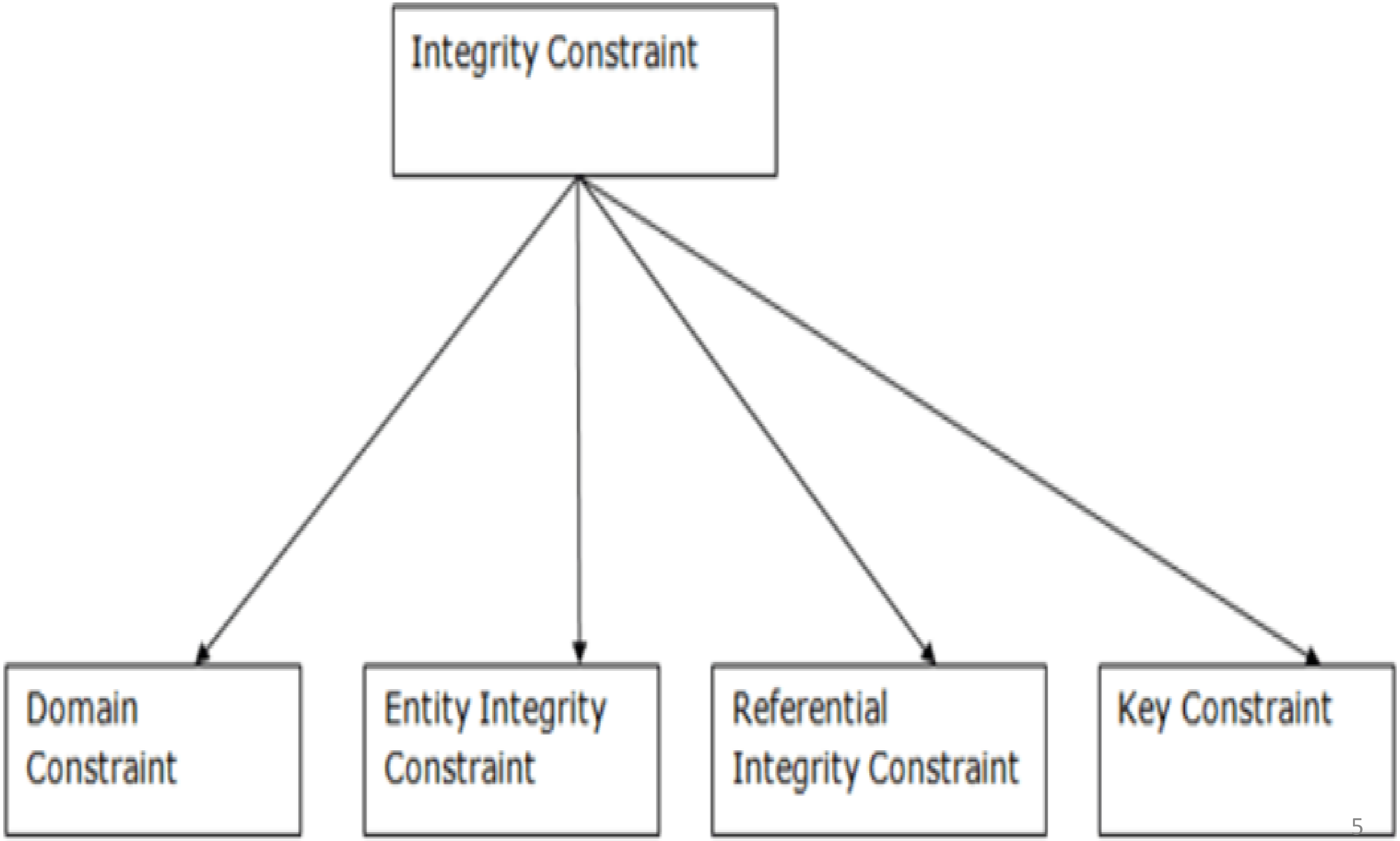
Integrity Constraints

- Integrity constraints are **rules** in a database that ensure the **accuracy**, **correctness**, and **consistency** of data.
- They prevent invalid data from being inserted, deleted, or updated.
- Constraints are a very important feature in any data model.
- *Constraints* are the rules that force DBMSs to check that data satisfies the semantics.
- Integrity constraints are a set of rules used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not

affected.

- Integrity constraints guard against accidental damage to the database, by ensuring that authorized changes to the database do not result in a loss of data consistency.

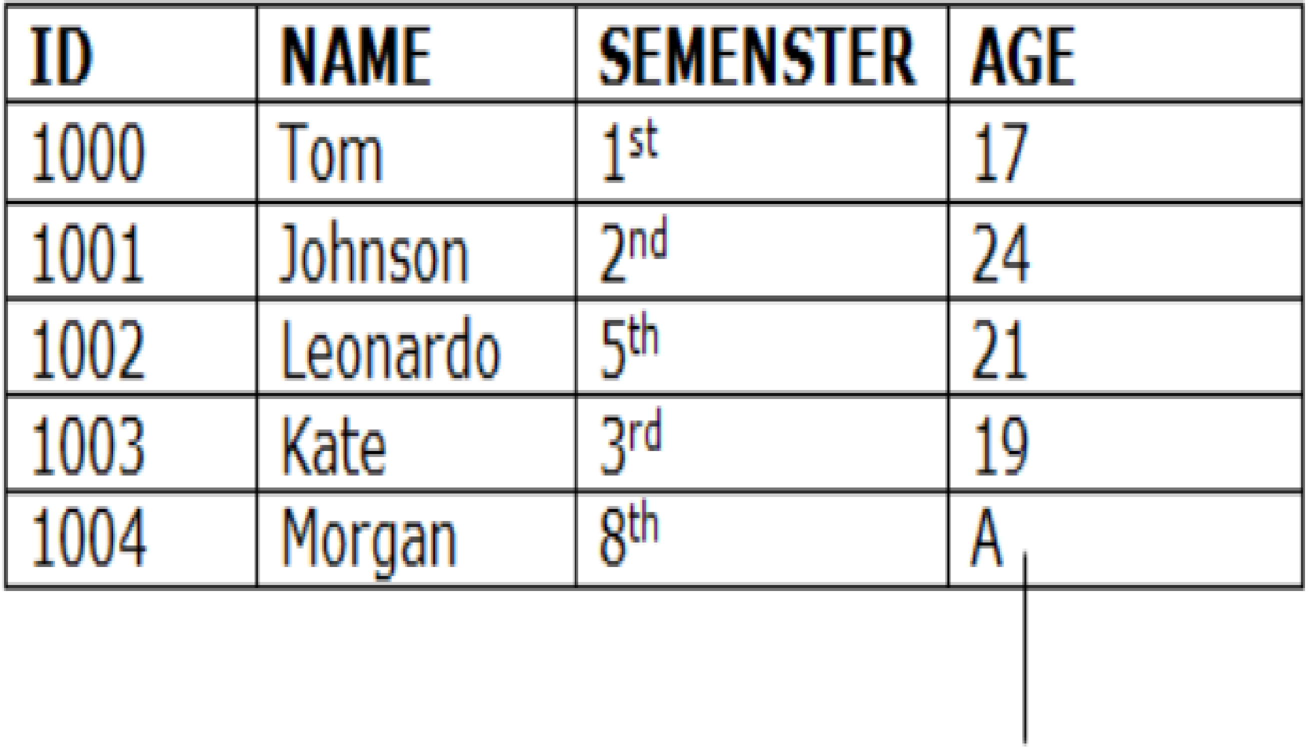
Types of constraints



1. Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
- Domain restricts the values of attributes in the relation and is a constraint of the relational model.
- The data type of domain includes string, character, integer, time, date, currency, etc.
- The value of the attribute must be available in the corresponding domain.
- *For example, the Employee ID (EID) must be unique or the employee Birthdate is in the range [Sep 1, 2024, Jan 1, 2025].*

Example:



ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

- The entity integrity constraint states that primary key value can't be null.
- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.
- To ensure *entity integrity*, it is required that every table have a primary key.
- Neither the PK nor any part of it can contain null values. This is because null values for the primary key mean we cannot identify some rows.
- For example, in the EMPLOYEE table, Phone cannot be a primary key since some people may not have a telephone.

➢ The NOT NULL constraint enforces a column to NOT accept NULL values.

➢ This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

✓ Every table must have a primary key.

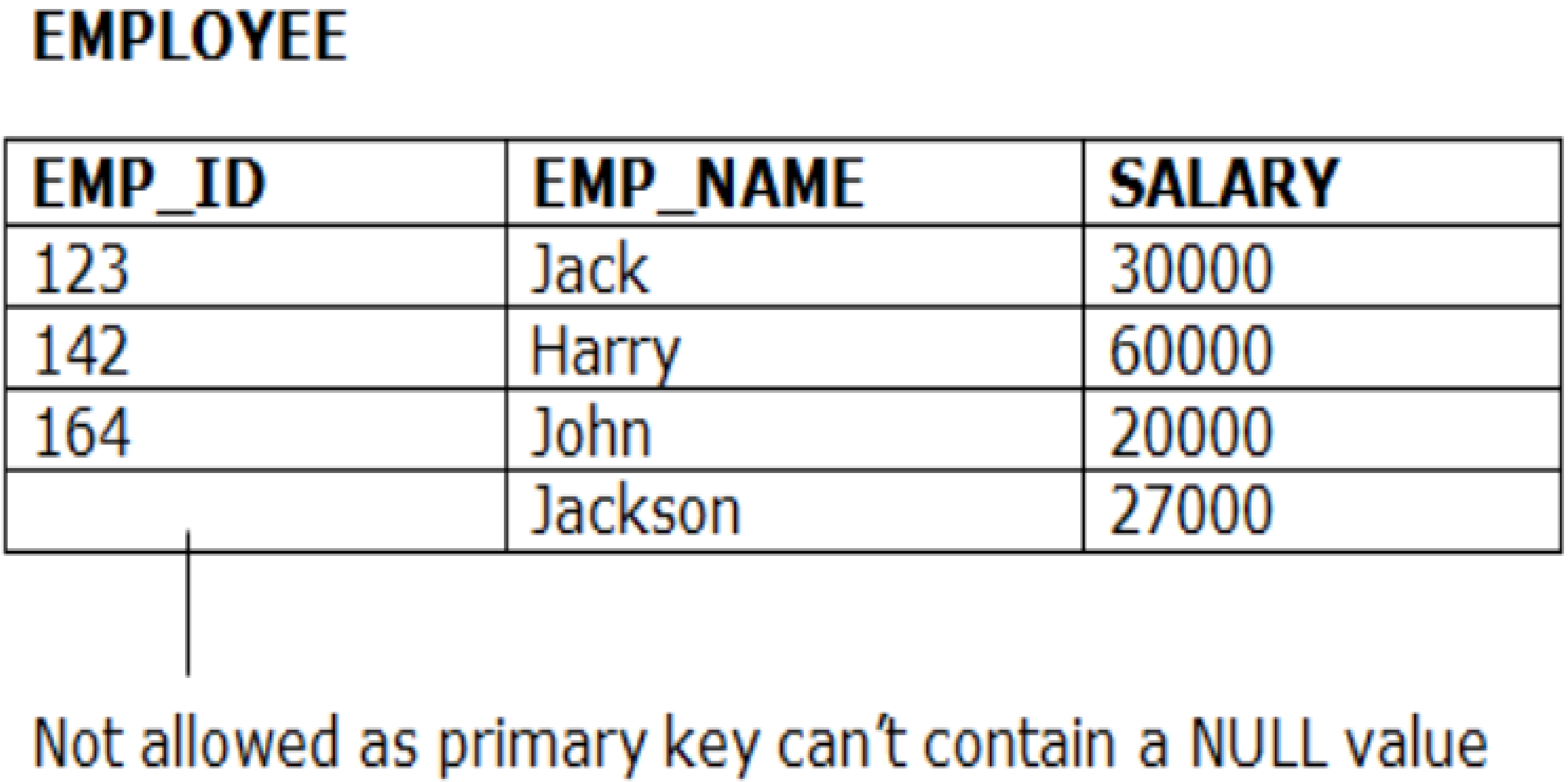
✓ The primary key must uniquely identify each record.

✓ Primary key values cannot be NULL.

✓ No two rows can have the same primary key value.

Generally Entity integrity makes sure that **every record can be uniquely identified** by its primary key, and that the primary key is always **unique and not empty**.

Example:



EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

3. Referential Integrity Constraints

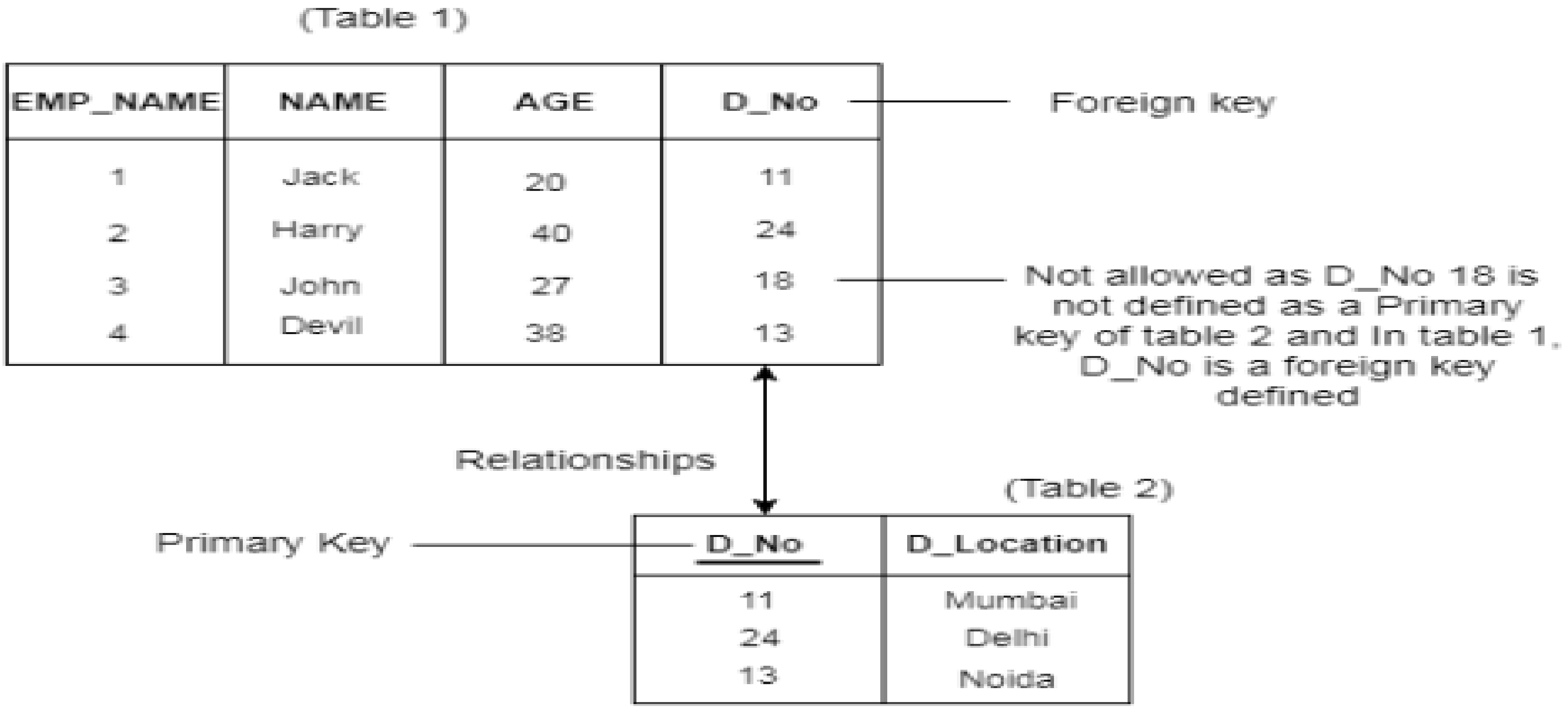
- *Referential integrity* requires that a foreign key must have a matching primary key or it must be null.
- This constraint is specified between two tables (parent and child); it maintains the correspondence between rows in these tables.
- It means the reference from a row in one table to another table must be valid.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2

Examples of referential integrity constraint in the Customer/Order database of the Company:

Customer(**CustID**, CustName)

Order(**OrderID**, CustID, OrderDate)

Example:



(Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

Foreign key

Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined

Relationships

(Table 2)

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

Primary Key

4. Key constraints/Unique Constraint/

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- A primary key can contain a unique and null value in the relational table.
- The UNIQUE constraint ensures that all values in a column are different.
- Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.
- A PRIMARY KEY constraint automatically has a UNIQUE constraint.
- However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

Key Constraint / Unique Constraint ensures that the values in a column must not repeat.

- Examples: UNIQUE Constraint on CREATE TABLE

```
CREATE TABLE Persons
 (
ID int NOT NULL UNIQUE,
 LastName varchar(255) NOT NULL UNIQUE,
FirstName varchar(255),
Age int
);
```

Database Security

- Database security means protection of a database against unauthorized access, either intentional or unintentional
- Database security requires the mechanisms, that protect a database against the intentional or accidental threats
- *More generally speaking, database security is concerned with ensuring the secrecy, integrity, and availability of data stored in a database.*

- Threat may be any situation or event, whether intentional or accidental, that may adversely affect a system and consequently the organization

Cont..

- Threats to databases : It may results in degradation of some/all security goals like;
  - ✓ Loss of Integrity
    - *Only authorized users should be allowed to modify data.*
    - *For example, students may be allowed to see their grades, but not allowed to modify them.*
  - ✓ Loss of Availability-if DB is not available for those users/ to which they have a legal right to uses the data
    - Authorized users should not be denied access.
    - For example, an instructor who wishes to change a grade should be allowed to do so.
  - ✓ Loss of Confidentiality
    - Information should not be disclosed to unauthorized users.
    - For example, a student should not be allowed to examine other students' grades.

Security attackers

In database security attackers are divided into three segments that are –

Administrator

- An admin is an authorized person who has permission to control the system but misuses his/her privileges against the security policies to get the important information.

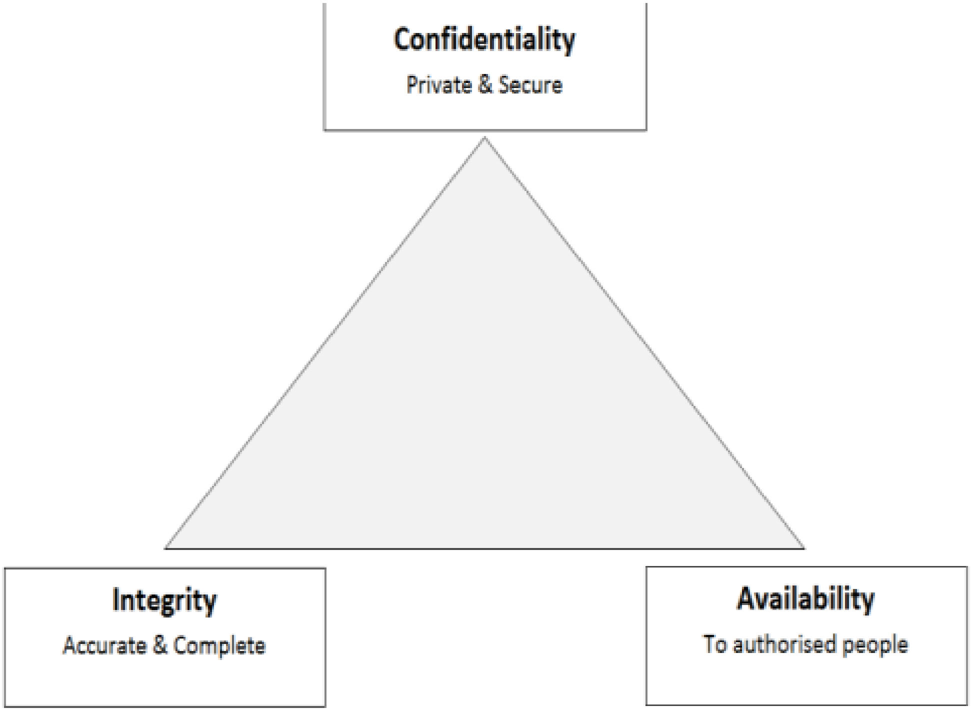
Insider

- An insider is also a member of trusted committee in an organization but did misuse of his/her authority and want to get some sensitive or any other important information

Intruder

- An intruder is not a part of an organization.
- Actually he/she is unauthorized person who access the personal data of an organization and want to get the sensitive information.

Database Security Properties



Database security protects against:

- Theft and fraud,
- Loss of confidentiality
- Loss of privacy
- Loss of integrity
- Loss of availability

- The security of data basically requires three things- Confidentiality, Integrity and Availability. Where **Confidentiality** means the data must be used by an authorized person, **Integrity** means the data must be controlled by an authorized person in an authorized manner and **Availability** means the data must be available to an authorized user at appropriate time .

.

- For databases, requirements on the security can be classified into the following categories:

Identification, Authentication

- Usually before getting access to a database each user has to identify himself to the computer system.
- Authentication is the way to verify the identity of a user at log-on time.
- Most common authentication methods are passwords but more advanced techniques like badge readers, **biometric recognition** techniques, or signature analysis devices are also available.

Authorization, Access Controls

- Authorization is the specification of a set of rules that specify who has which type of access to what information.
- Authorization policies therefore govern the disclosure and modification of information.
- Access controls are procedures that are designed to control authorizations.
- They are responsible to limit access to stored data to authorized users only.

Authentication

- All users of the database will have different access levels and permission for different data objects, and authentication is the process of checking whether the user is the one with the privilege for the access level.
- Thus the system will check whether the user with a specific username and password is trying to use the resource

Authorization/Privilege

- Authorization refers to the process that determines the mode in which a particular (previously authenticated) client is allowed to access a specific resource controlled by a server.
- Any database access request will have the following three major components

1. Requested Operation: what kind of operation is requested by a specific query?
2. Requested Object: on which resource or data of the database is the operation sought to be applied?
3. Requesting User: who is the user requesting the operation on the specified object?

Forms of user authorization

There are different forms of user authorization on the resource of the database. These includes :

1. Read Authorization: the user with this privilege is allowed only to read the content of the data object.
2. Insert Authorization: the user with this privilege is allowed only to insert new records or items to the data object.
3. Update Authorization: users with this privilege are allowed to modify content of attributes but are not authorized to delete the records.
4. Delete Authorization: users with this privilege are only allowed to delete a record and not anything else.

- Note: Different users, depending on the power of the user, can have one or the combination of the above forms of authorization on different data objects.

Database Security and the DBA

- The database administrator (**DBA**) is the central authority for managing a database system.
  - The DBA's responsibilities include
    - Account creation
    - granting privileges to users who need to use the system
    - Privilege revocation
    - classifying users and data in accordance with the policy of the organization

Access Protection, User Accounts, and Databases Audits

- Whenever a person or group of persons need to access a database system, the individual or group must first apply for a user account.
- The DBA will then create a new **account id** and **password** for the user if he/she believes there is a legitimate need to access the database

- A **database audit** is the systematic process of tracking and documenting database activities to ensure compliance, security, and operational integrity.
- It involves the **collection, recording, and analysis** of database events and actions.
- This is typically done using **audit trails** (logs) that capture:
- **Who** accessed the data (user, application, IP address)
- **What** action was performed (SELECT, UPDATE, DELETE, etc.)
- **When** it happened (timestamp)
- **What** specific data was affected (e.g., which rows were changed)
- **Where** the request came from (which machine or application)

- To protect databases against the possible threats two kinds of countermeasures can be implemented: ***Access control ,and Encryption***

2 . Access Control (AC)

2. 1. Discretionary Access Control (DAC)

- The typical method of enforcing discretionary access control in a database system is based on the granting and revoking privileges.
- These are defined by user identification during authentication
- Example: username, password.
- Its main aim is to grant and revoke privileges to users.
- Access is controlled by the owner of the data (table, row, file).
- The owner can grant or revoke permissions to other users.
- More flexible, but less secure.
- Example: A user who owns a table uses GRANT to allow others to read or write.

Syntax :-GRANT permission_list ON object_name TO user_name;

Example 1: Give SELECT permission

GRANT SELECT ON Employee TO user A;

Example 2: Give SELECT and UPDATE

GRANT SELECT, UPDATE ON Employee TO user B;

Example 3: Give all permissions

GRANT ALL PRIVILEGES ON Employee TO userC;

- Revoking Privileges
  - SQL, a **REVOKE** command is included for the purpose of **canceling privileges**.
  - Used to **remove permissions** that were given using GRANT.

  Syntax :- REVOKE permission_list ON object_name FROM user_name;

  Example 1: Remove SELECT permission

  REVOKE SELECT ON Employee FROM userA;

Example 1

- Suppose that the DBA creates four accounts:A1, A2, A3, A4 and wants only A1 to be able to create relations. Then the DBA must issue the following GRANT command in SQL

```
GRANT CREATE TABlE TO A1;
```

Example 2

- Suppose that A1 **creates** the two base relations **EMPLOYEE** and **DEPARTMENT**
  - A1 is then **owner** of these two relations and hence A1 has all the relation privileges on each of them.
- Suppose that A1 wants to grant A2 the privilege to insert and delete rows in both of these relations, but A1 does not want A2 to be able to propagate these privileges to additional accounts:

```
GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
```

Example 3

- Suppose that A1 wants to allow A3 to retrieve information from either of the table (Department or Employee) and also to be able to propagate the SELECT privilege to other accounts.
- A1 can issue the command:

```
GRANT SELECT ON EMPLOYEE, DEPARTMENT
      TO A3 WITH GRANT OPTION;
```

- A3 can grant the **SELECT** privilege on the **EMPLOYEE** relation to A4 by issuing:

```
GRANT SELECT ON EMPLOYEE TO A4;
```

  - Notice that A4 can’t propagate the SELECT privilege because GRANT OPTION was not given to A4

Example 4

- Suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 can issue:

```
REVOKE SELECT ON EMPLOYEE FROM A3;
```

- The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4, too, because A3 granted that privilege to A4 and A3 does not have the privilege any more.

2.2 Mandatory Access Control

- Access is controlled by the system or security policy, not by the owner.
- Users and data objects are assigned security levels (e.g., Top Secret, Secret).
- Very strict and highly secure.
- Common in military, government, and high-security systems.
- In many applications, **additional security policy** is needed that classifies data and users based on security classes.
- Typical **security classes** are top secret (TS), secret (S), confidential (C), and unclassified (U), where TS is the highest level and U the lowest: $TS \geq S \geq C \geq U$

2.2 Mandatory Access Control

Example:

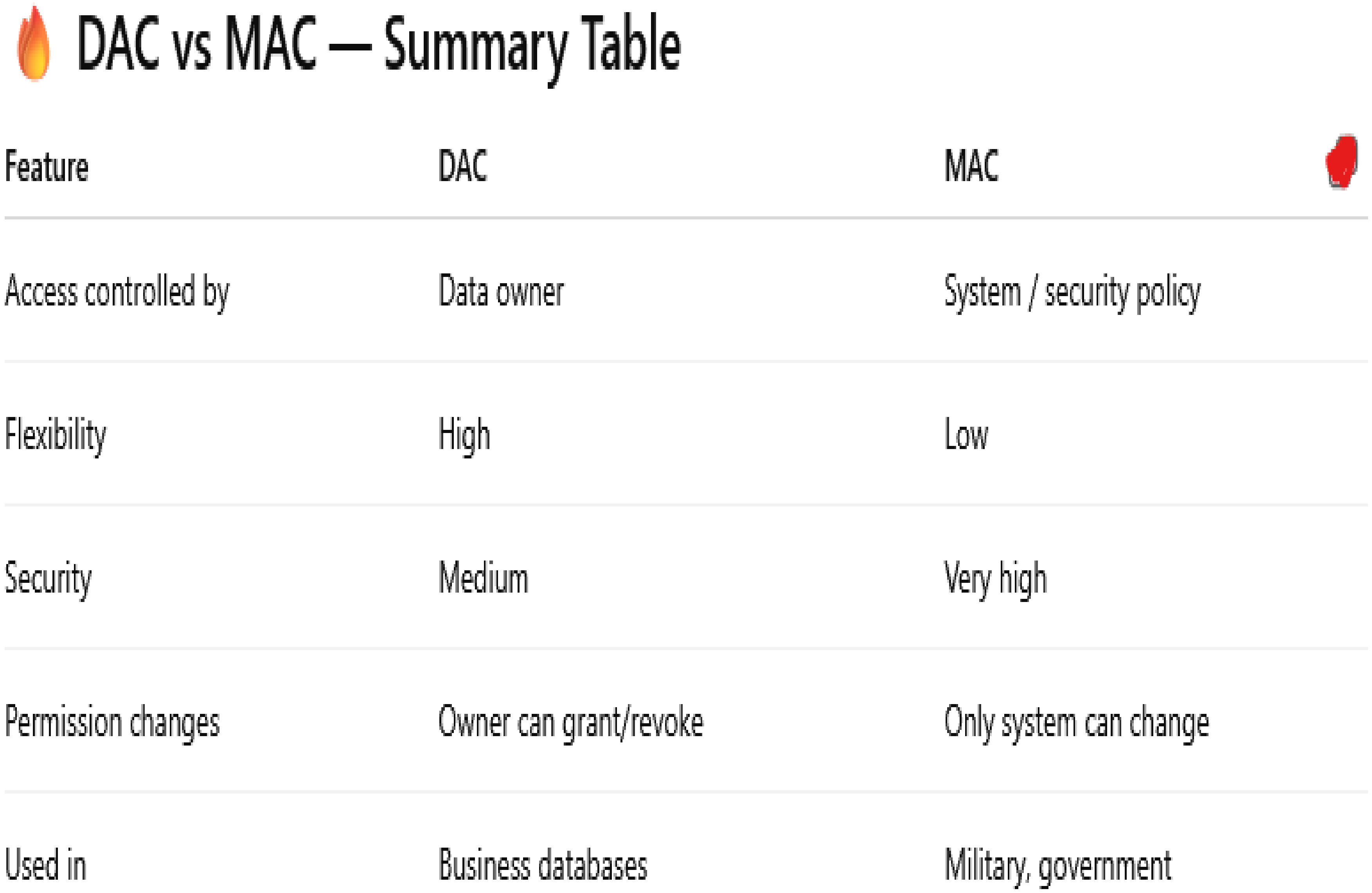
1. Hospital Medical Records (High-security systems)

- ➢ Doctors: can see medical details
- ➢ Nurses: can only see basic care instructions
- ➢ Admin staff: can only see billing info
- ➢ Access is controlled by **hospital policy**, not by the doctor who created the data.

2. Military Security System

- ❖ Documents labeled: **Top Secret**, **Secret**, **Confidential**, **Unclassified**
- ❖ Users (soldiers/officers) have clearance levels.
- ❖ If a user has **Secret clearance**, they **cannot** read **Top Secret** documents. The **system** decides access, not the user.

❑ Comparing DAC and MAC



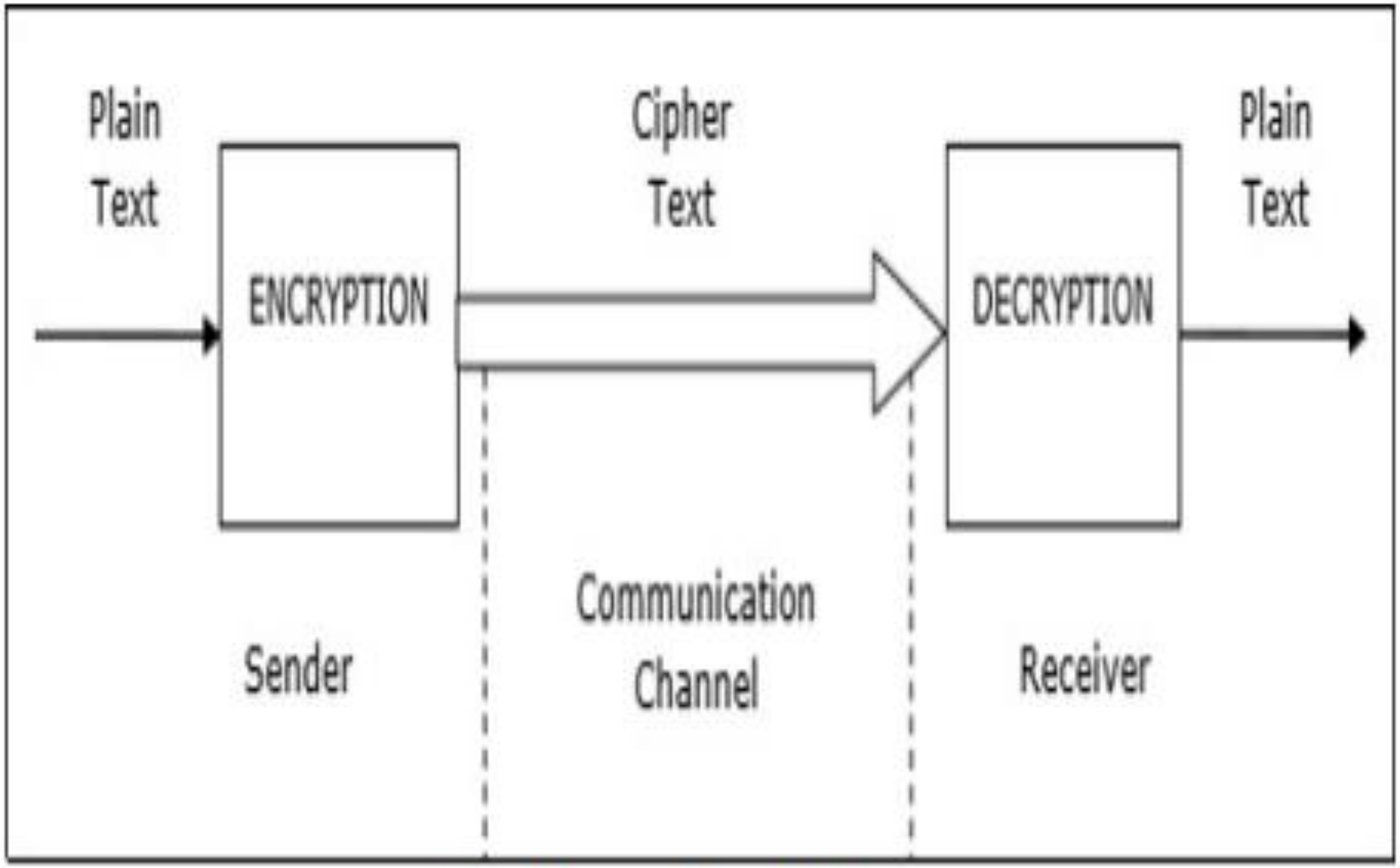
🔥 DAC vs MAC — Summary Table

Feature	DAC	MAC
Access controlled by	Data owner	System / security policy
Flexibility	High	Low
Security	Medium	Very high
Permission changes	Owner can grant/revoke	Only system can change
Used in	Business databases	Military, government

Data Encryption

- **Encryption** refers to the coding of information in order to keep it secret.
- Encryption of data means encoding of data by a special algorithm, that senders the data unreadable by any program without the decryption key
- This security issue used to protect sensitive data (such as credit card numbers) that is being transmitted via some type communication network.
- The data is encoded using some encoding algorithm.
- An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting algorithms (or keys) to decipher data.

Encryption Process



Types of encryption

- There are two types of encryption in widespread use today: **symmetric** and **asymmetric** encryption.
- The name derives from whether or not the same key is used for encryption and decryption.

Symmetric encryption

- In symmetric encryption the same key is used for encryption and decryption.
- It is therefore critical that a secure method is considered to transfer the key between sender and recipient.

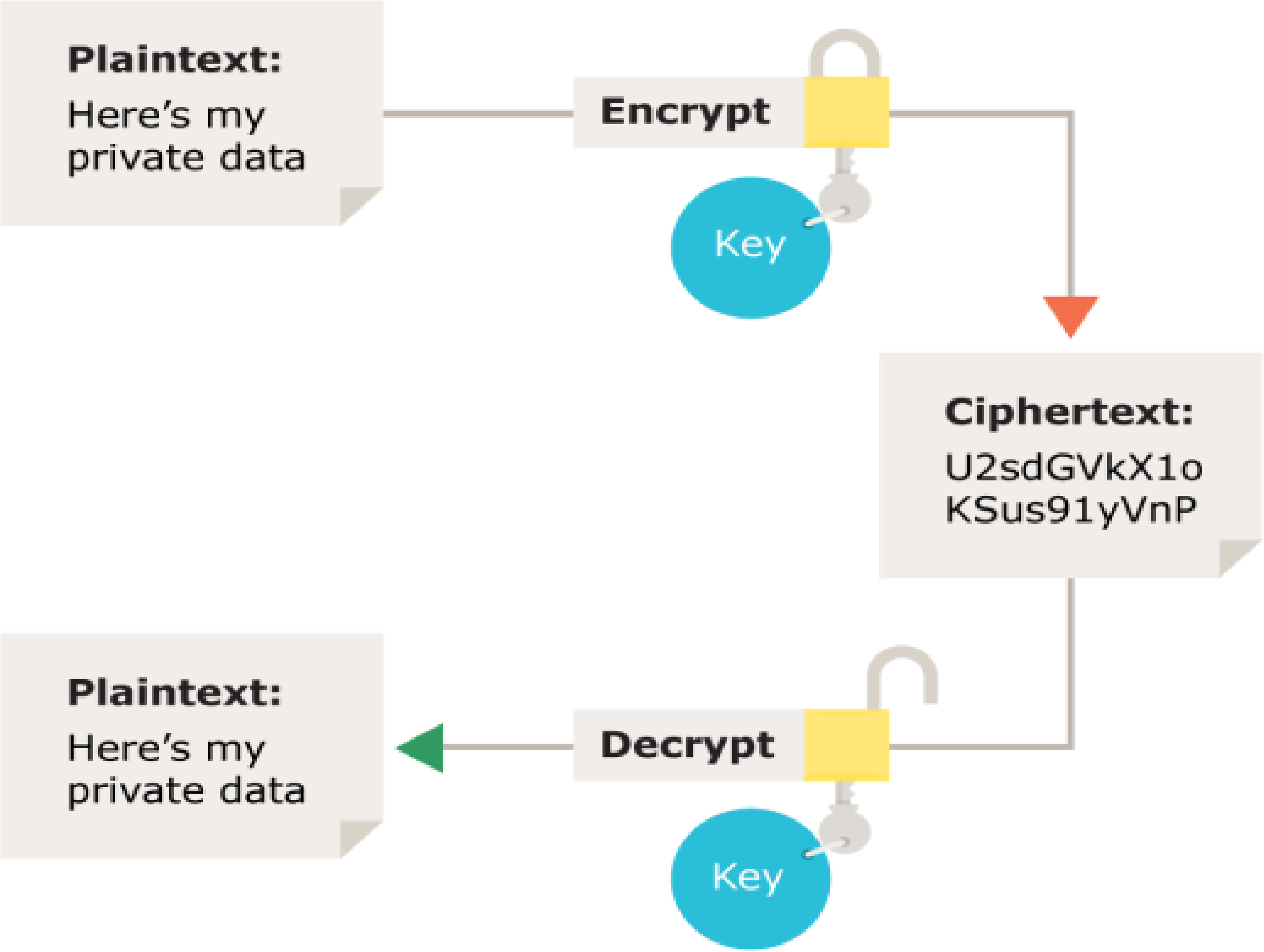


Figure 1: Symmetric encryption – Using the same key for encryption and decryption

Asymmetric encryption

- Asymmetric encryption uses the notion of a key pair: a different key is used for the encryption and decryption process.
- One of the keys is typically known as the private key and the other is known as the public key.
- The private key is kept secret by the owner and the public key is either shared amongst authorized recipients or made available to the public at large.
- Data encrypted with the recipient’s public key can only be decrypted with the corresponding private key.
- Data can therefore be transferred without the risk of unauthorized or unlawful access to the data.

Implementing Encryption

- Choosing the right algorithm
- Choosing the right key size
- Choosing the right software
- Keeping the key secure

Database Security and the DBA

- The database administrator (DBA) is the central authority for managing a database system.
- The DBA's responsibilities include
- granting privileges to users who need to use the system
- classifying users and data in accordance with the policy of the organization

□ The DBA is responsible for the overall security of the database system.

..

- The DBA has a DBA account in the DBMS Sometimes these are called a system or super user account
- These accounts provide powerful capabilities such as:

1. Account creation
2. Privilege granting
3. Privilege revocation
4. Security level assignment

Action 1 is access control, whereas 2 and 3 are discretionary and 4 is used to control mandatory authorization

?