**RBAC(Role-based access control )**

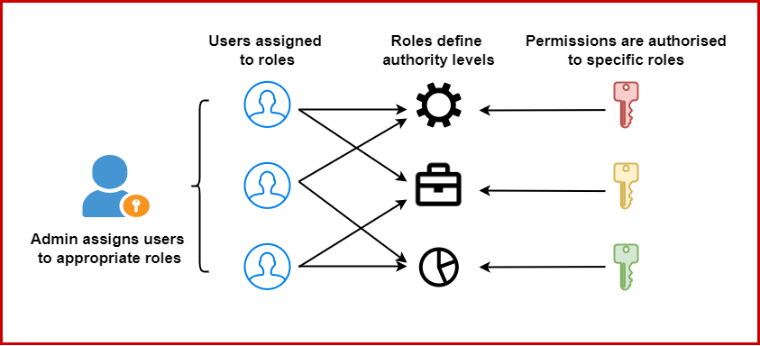
Role-Based Access Control (RBAC) is a pivotal mechanism in Database Management Systems (DBMS) for managing and enforcing security policies. By assigning roles to users and granting permissions based on these roles, RBAC simplifies the administration of access rights and enhances security.

**Principles of RBAC**

RBAC is built on the idea of assigning permissions to roles rather than individuals. Here’s how it works:

* **Roles:** Defined based on job functions within an organization (e.g., admin, manager, employee).
* **Permissions:** Specific access rights assigned to roles (e.g., read, write, delete).
* **Users:** Individuals assigned to one or more roles based on their responsibilities.

By separating the assignment of permissions from individual users, RBAC provides a more manageable and scalable approach to access control.



**Core Components**

* **User Assignment:** Users are assigned to roles based on their job responsibilities.
* **Role Permissions:** Roles are granted specific permissions that define what actions can be performed.
* **Role Hierarchies:** Roles can be organized in a hierarchy, allowing for inheritance of permissions (e.g., a senior manager role might inherit permissions from a manager role).

**Advantages of RBAC**

* **Simplified Administration:** Admins can manage access rights more efficiently by modifying roles rather than individual user permissions.
* **Enhanced Security:** By enforcing the principle of least privilege, users only have the permissions necessary for their role, reducing the risk of unauthorized access.
* **Scalability:** As organizations grow, adding new users and modifying roles becomes more straightforward without compromising security.
* **Compliance:** Helps in meeting regulatory requirements by ensuring that access controls are systematically enforced.

**Disadvantages of RBAC**

* **Complexity:** Implementing RBAC makes the system more complex. Setting up roles and permissions for each user can take time, and access levels may require regular maintenance to ensure they stay appropriate.
* **Inflexible:** Roles and permissions designated on a set of pre-designed rules are difficult to edit to accommodate exception cases where access is required to resources by users not within their access definitions.
* **Delegation:** Delegation of roles and respective accesses is difficult, especially in large organizations. This can cause delays in gaining access to resources, affecting productivity and corporate operations.
* **Granularity:** RBAC limits the ability to make granular access decisions based on specific user attributes, resulting in users not having specific accesses they require or having access to more resources than they need, which can lead to data breaches.

**Real-World Application of RBAC**

Let’s imagine a scenario at a university to better understand how RBAC can be implemented:

* **Roles:** Student, Faculty, Registrar, IT Support.
* **Permissions:**
  + Students can register for courses and view their own academic records. They cannot access other students' or faculty records.
  + Faculty can access and update the academic records of students enrolled in their courses. They cannot access personal information of students or other faculty members.
  + Registrars have comprehensive access to all student and faculty records for administrative purposes.
  + IT Support manages the system infrastructure without accessing academic or personal data.

**Encryption**

Data encryption is the process of converting readable information (plaintext) into an unreadable format (ciphertext) to protect it from unauthorized access.

**Key Objective of Encryption Data**

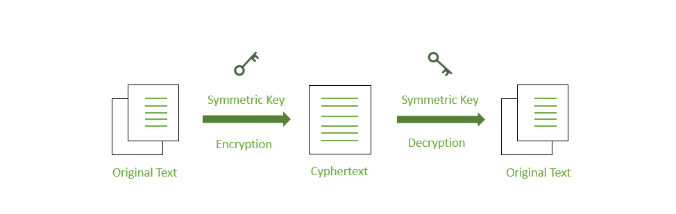
* **Confidentiality:**Encryption ensures that only authorized parties can get access to data and recognize the information.
* **Data Integrity:**Encryption can also provide data integrity by making sure that the encrypted data remains unchanged during transmission. Any unauthorized changes to the encrypted information will render it undecipherable or will fail integrity checks.
* **Authentication:** Encryption may be used as part of authentication mechanisms to verify the identification of the communication party.
* **Non-Repudiation:** Through encryption, events can make sure that they cannot deny their involvement in growing or sending a selected piece of data.

**Types of Data Encryption**

There are multiple encryption techniques, each of which have been developed with various security requirements in mind. [Symmetric and Asymmetric encryption](https://www.geeksforgeeks.org/difference-between-symmetric-and-asymmetric-key-encryption/) are the two types of data encryption.

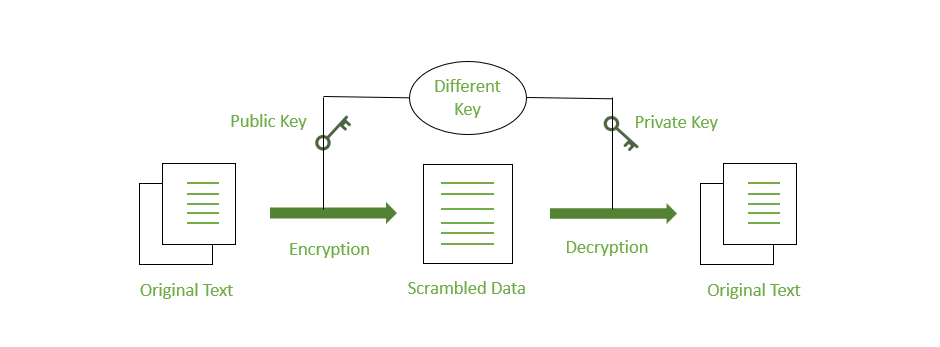
**1. Symmetric Key Encryption**

There are a few strategies used in cryptography algorithms. For encryption and decryption processes, some algorithms employ a unique key.



**2. Asymmetric Key Encryption**

Some cryptography methods employ one key for data encryption and another key for data decryption.



**Data masking techniques**

**What is Data Masking?**

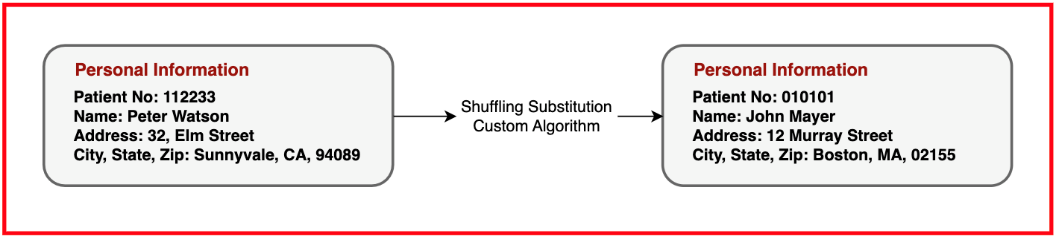
**Data masking**, also known as **data obfuscation**, hides original data by replacing it with modified content. This ensures that sensitive information is not accessible to unauthorized users. Masked data keeps the same format as the original, making it suitable for testing and development without risking privacy.

**Techniques of Data Masking**

* **1. Static Data Masking (SDM)**

Static Data Masking modifies data at rest within the database. It is useful for creating copies of databases for testing, development, or training purposes.

**Example:** A healthcare organization masks patient names, Social Security numbers, and medical records when creating a copy of its database for testing.



* **2. Dynamic Data Masking (DDM)**

Dynamic Data Masking hides data as it is accessed by unauthorized users, without changing the original data. It is ideal for real-time data access control.

**Example:** In an online banking system, customer account numbers are masked for service representatives, showing only the last four digits.

* **3. Deterministic Data Masking**

Deterministic Data Masking ensures that a specific data value is replaced with the same masked value every time, providing consistency across databases.

**Example:** A retail company masks customer emails so "john.doe@example.com" is always replaced with "masked.email1@example.com".

* **4. Non-Deterministic Data Masking**

Non-Deterministic Data Masking replaces data with different values each time, adding more protection through variability.

**Example:** Employee phone numbers in an HR database are masked differently each time, making it harder to trace back to the original number.

* **5. Format-Preserving Masking**

This technique maintains the original format and structure of data even after masking, making it ideal for applications that require specific data formats.

**Example:** A credit card company masks card numbers but retains their 16-digit structure.

* **6. Shuffling**

Shuffling reorders existing data within a column, keeping data realistic while breaking its relationship with original records.

**Example:** A university masks student grades by shuffling them, keeping them realistic but not linked to specific students.

* **7. Redaction**

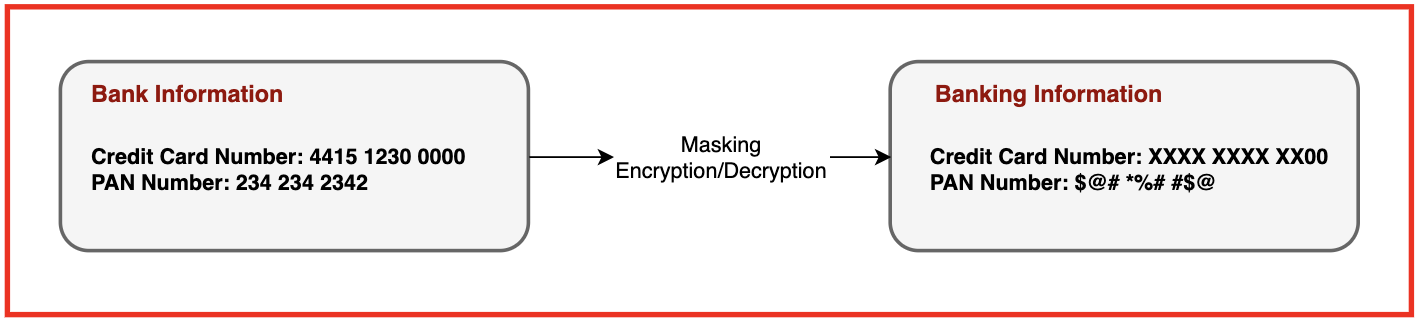
Redaction completely removes sensitive information, often used for data that should never be exposed.

**Example:** Legal documents have client details redacted before being shared externally.

* **8. Nulling Out**

Nulling Out replaces sensitive data with NULL values, used when the data is not needed for the current task.

**Example:** In market research, names and addresses are nulled out before sharing data with third-party analysts.



**Applications of Data Masking**

* **Healthcare Industry:** Hospitals use data masking for patient privacy when developing new systems. This ensures compliance with regulations like HIPAA.
* **Financial Sector:** Banks mask customer data during app development to keep account numbers and transaction details secure.
* **E-commerce:** Online stores mask payment data in analytics to prevent unauthorized access and maintain customer trust.
* **Education:** Universities protect student records during research by masking personal data, ensuring compliance with FERPA.