**Database Security**

Database security is the range of methods used to protect information stored within a database. While hacking attempts are the most commonly thought of hazard to database information, there are many other dangers. Physical damage to the computer, improper coding or corruption, and data overload are all potential threats to a database. This means there usually are many security measures in place — from firewalls to auditing and backup disks — to keep any potential damage to a minimum and prevent the loss of an entire database. Most businesses have their own database security protocols to guard against specific attacks and potential damage.

Installing a database firewall, a protective barrier that keeps all unknown connections at bay, is the most basic form of database security. Firewalls are installed on most computers and are made so hackers have a difficult time connecting to a victim’s computer. Firewalls work by filtering through connections in the network and only allowing trusted computers or users to access the database. While skilled hackers can get around this, a firewall does provide a high level of security.

**Database Security** is the mechanisms that protect the database against intentional or accidental threats.

We consider database security in relation to the following situations:

- Theft and Fraud

- Loss of confidentiality

**Threat** is any intentional or accidental event that may adversely affect the system.

Examples of threats:

- Using another person’s log-in name to access data

- Unauthorized copying data

- Program/Data alteration

- Illegal entry by hacker

- Viruses

**Countermeasures & Computer-Based Controls:**

- Authorization

- Views

- Backup and Recovery

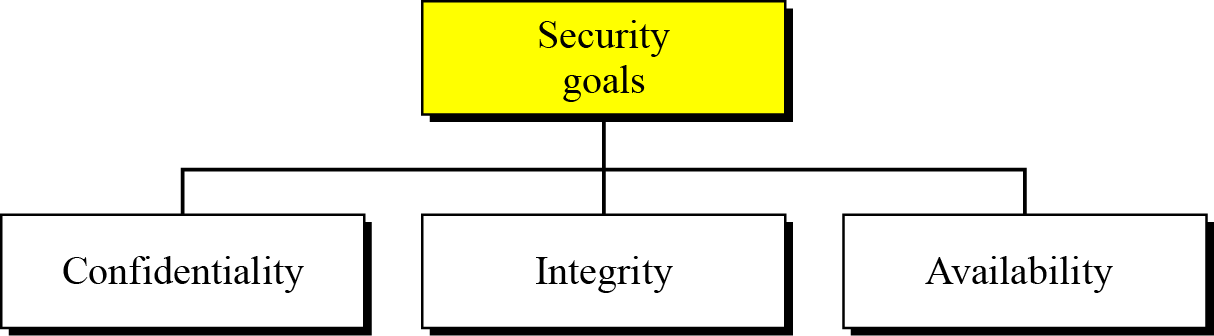
- Integrity

- Encryption

- RAID Technology

**List the importance of security**   
- To prevent unauthorized data observation.   
- To prevent unauthorized data modification.   
- To ensure the data confidential.  
- To make sure the data integrity is preserved.  
- To make sure only the authorized users have access to the data.

Security Goals:



Confidentiality:

Confidentiality, keeping information secret from unauthorized access, is probably the most common aspect of information security: we need to protect confidential information. An organization needs to guard against those malicious actions that endanger the confidentiality of its information.

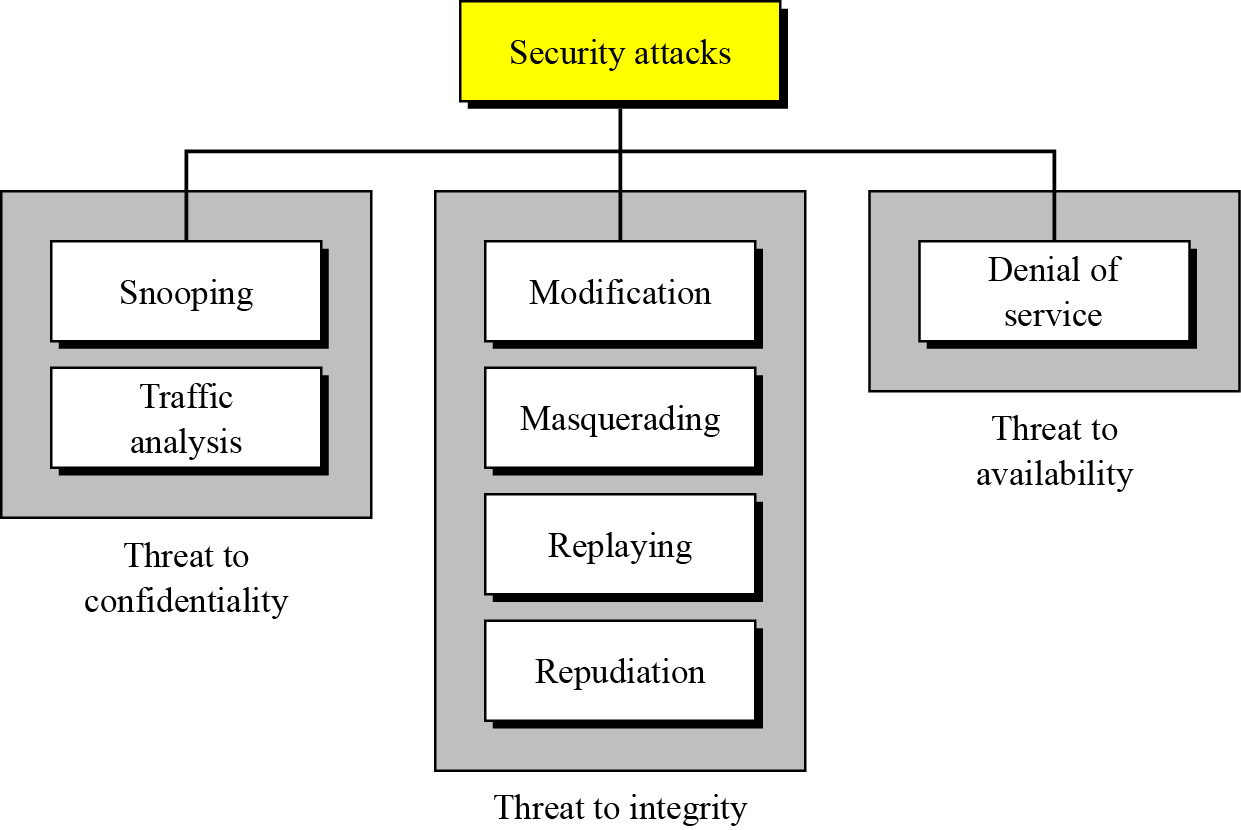
Integrity:

Information needs to be changed constantly. In a bank, when a customer deposits or withdraws money, the balance of their account needs to be changed. Integrity means that changes should be done only by authorized users and through authorized mechanisms.

Availability:

The third component of information security is availability. The information created and stored by an organization needs to be available to authorized users and applications. Information is useless if it is not available. Information needs to be changed constantly, which means that it must be accessible to those authorized to access it. Unavailability of information is just as harmful to an organization as a lack of confidentiality or integrity. Imagine what would happen to a bank if the customers could not access their accounts for transactions.

**Attacks:** The three goals of security—confidentiality, integrity and availability—can be threatened by security attacks



To protect the database we must take security measures at different levels:

* **Database System:** Some database system users may be authorized to access only a limited portion of the database. Other users may be allowed to issue queries, but may be forbidden to modify the data. It is the responsibility of the database system to ensure that these authorization restrictions are not violated.
* **Operation System:** No matter how secure the database system is, weakness in operating system security may serve as a means of unauthorized access to the database.
* **Network:** Since almost all database systems allow remote access through terminals or networks, software level security within the network software is as important as physical security, both on the Internet and in private networks.
* **Physical:** Sites with computer system must be physically secured against armed or secret entry by intruders.
* **Human:** Users must be authorized carefully to reduce the chance of any user giving access to an intruder in exchange for a bribe or other favors.

**Access Control**

The primary method used to protect data is limiting access to the data. This can be done through Authentication, authorization, and access control. These three mechanisms are distinctly different but usually used in combination with a focus on access control for granularity in assigning rights to specific objects and users. For instance, most database systems use some form of authentication, such as username and password, to restrict access to the system. Further, most users are authorized or assigned defined privileges to specific resources. Access control further refines the process by assigning rights and privileges to specific data objects and data sets. Within a database, these objects usually include tables, views, rows, and columns. For instance, StudentA may be given login rights to the University database with authorization privileges of a student user which include read-only privileges for the Course\_ Listing data table. Through this granular level of access control, students may be given the ability to browse course offerings but not to peruse grades assigned to their classmates. Many students, today, inherently understand the need for granularity in granting access when framed in terms of granting ‘friends’ access to their Facebook site.

### Discretionary Access Control vs Mandatory Access Control

In discretionary access control (DAC), the owner of the object specifies which subjects can access the object. This model is called discretionary because the control of access is based on the discretion of the owner

In mandatory access control (MAC), the system (and not the users) specifies which subjects can access specific data objects

The MAC model is based on security labels. Subjects are given a security clearance (secret, top secret, confidential, etc.), and data objects are given a security classification (secret, top secret, confidential, etc.). The clearance and classification data are stored in the security labels, which are bound to the specific subjects and objects.

When the system is making an access control decision, it tries to match the clearance of the subject with the classification of the object. For example, if a user has a security clearance of secret, and he requests a data object with a security classification of top secret, then the user will be denied access because his clearance is lower than the classification of the object.  
  
The MAC model is usually used in environments where confidentiality is of utmost importance, such as a military institution.

In the most common implementation of Discretionary Access Control (DAC) users “own” their directories, files and programs they contain. They can grant and deny access and execution privileges for these to other users. Users can also be parts of group. They may be able to grant or restrict access to all members of a group. In Mandatory Access Control (MAC), operating system controls the ability of a subject or initiator to access or generally to perform some sort of operations on an object or target. The object or target can be like a process, a file, a directory, a program or a memory segment. Subject and Object each have a set of security attributes. Whenever a subject attempts to access an object, an authorizations rule enforced by the operating system kernel compares these security attributes to the policy and decides whether the access or operation will be allowed. With Mandatory Access Control (MAC), this security policy is centrally controlled by the security policy administrator; users do not have the ability to override the policy, and for example; grant access to files that would otherwise be restricted. "ow

**Authentication**

Authentication verifies **who you are**. For example, you can login into your Unix server using the ssh client, or access your email server using the POP3 and SMTP client. Usually, PAM (Pluggable Authentication Modules) are used as low-level authentication schemes into a high-level application programming interface (API), which allows programs that rely on authentication to be written independently of the underlying authentication scheme.

**Authorization**

Authorization verifies **what you are authorized to do**. For example, you are allowed to login into your Unix server via ssh client, but you are not authorized to browser /data2 or any other file system. Authorization occurs after successful authentication.

Usually, the connection attempt must be both authenticated and authorized by the system. You can easily find out why connection attempts are either accepted or denied with the help of these two factors.

Authentication means verifying the user is a valid user with username and password.

Authorization is the permissions of a valid user. So once the user is authenticated, his permissions are assigned using authorization so the access over resources is controlled.

There is indeed a fundamental difference. Authentication is the mechanism whereby systems may securely identify their users. Authentication systems seek to provide answers to the questions:

* Who is the user?
* Is the user really who he/she represents himself to be?

Authorization, by contrast, is the mechanism by which a system determines what level of access a particular (authenticated) user should have to resources controlled by the system. For an example that may or may not be related to a web-based scenario, a database management system might be designed so as to provide certain specified individuals with the ability to retrieve information from a database but not the ability to change data stored in the database, while giving other individuals the ability to change data. Authorization systems provide answers to the questions:

* Is user X authorized to access resource R?
* Is user X authorized to perform operation P?
* Is user X authorized to perform operation P on resource R?

**Encryption** is the conversion of data into a form, called a [ciphertext](http://searchcio-midmarket.techtarget.com/definition/ciphertext), that cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood.

The use of encryption/decryption is as old as the art of communication. In wartime, a [cipher](http://searchsecurity.techtarget.com/definition/cipher), often incorrectly called a code, can be employed to keep the enemy from obtaining the contents of transmissions. (Technically, a code is a means of representing a signal without the intent of keeping it secret; examples are [Morse code](http://searchnetworking.techtarget.com/definition/Morse-code) and [ASCII](http://searchcio-midmarket.techtarget.com/definition/ASCII).) Simple ciphers include the substitution of letters for numbers, the rotation of letters in the alphabet, and the "scrambling" of voice signals by inverting the [sideband](http://whatis.techtarget.com/definition/sideband) frequencies. More complex ciphers work according to sophisticated computer [algorithm](http://whatis.techtarget.com/definition/algorithm)s that rearrange the data bits in digital signals.

In order to easily recover the contents of an encrypted signal, the correct decryption [key](http://searchsecurity.techtarget.com/definition/key) is required. The key is an algorithm that undoes the work of the encryption algorithm. Alternatively, a computer can be used in an attempt to break the cipher. The more complex the encryption algorithm, the more difficult it becomes to eavesdrop on the communications without access to the key.

Encryption/decryption is especially important in [wireless](http://searchmobilecomputing.techtarget.com/definition/wireless) communications. This is because wireless circuits are easier to tap than their hard-wired counterparts. Nevertheless, encryption/decryption is a good idea when carrying out any kind of sensitive transaction, such as a credit-card purchase online, or the discussion of a company secret between different departments in the organization. The stronger the cipher -- that is, the harder it is for unauthorized people to break it -- the better, in general. However, as the strength of encryption/decryption increases, so does the cost.

In recent years, a controversy has arisen over so-called strong encryption. This refers to ciphers that are essentially unbreakable without the decryption keys. While most companies and their customers view it as a means of keeping secrets and minimizing fraud, some governments view strong encryption as a potential vehicle by which terrorists might evade authorities. These governments, including that of the United States, want to set up a key-escrow arrangement. This means everyone who uses a cipher would be required to provide the government with a copy of the key. Decryption keys would be stored in a supposedly secure place, used only by authorities, and used only if backed up by a court order. Opponents of this scheme argue that criminals could hack into the key-escrow database and illegally obtain, steal, or alter the keys. Supporters claim that while this is a possibility, implementing the key escrow scheme would be better than doing nothing to prevent criminals from freely using encryption/decryption

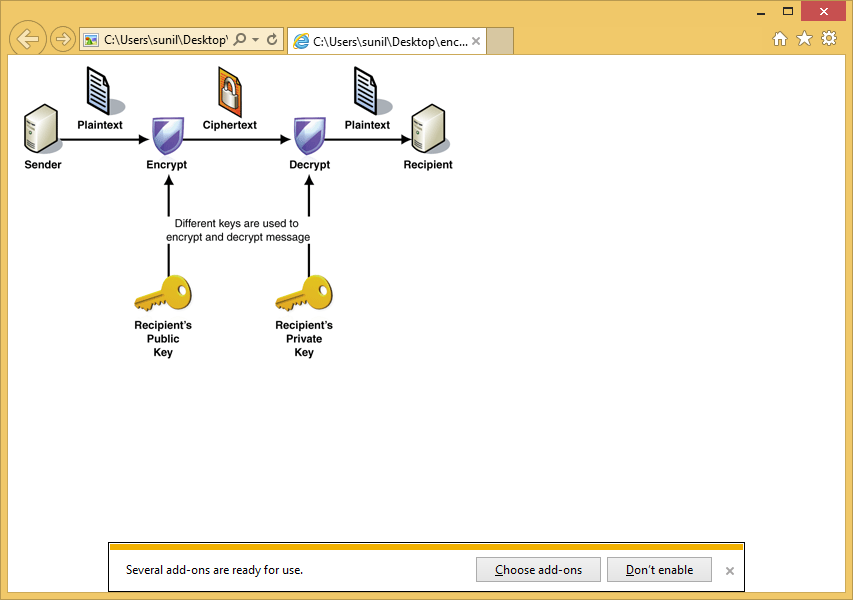
# Data Encryption and Decryption

Encryption is the process of translating plain text data ([plaintext](http://msdn.microsoft.com/en-us/library/windows/desktop/ms721603(v=vs.85).aspx#_security_plaintext_gly)) into something that appears to be random and meaningless ([ciphertext](http://msdn.microsoft.com/en-us/library/windows/desktop/ms721572(v=vs.85).aspx#_security_ciphertext_gly)). Decryption is the process of converting ciphertext back to plaintext.

To encrypt more than a small amount of data, [symmetric encryption](http://msdn.microsoft.com/en-us/library/windows/desktop/ms721625(v=vs.85).aspx#_security_symmetric_encryption_gly) is used. A [symmetric key](http://msdn.microsoft.com/en-us/library/windows/desktop/ms721625(v=vs.85).aspx#_security_symmetric_key_gly) is used during both the encryption and decryption processes. To decrypt a particular piece of ciphertext, the key that was used to encrypt the data must be used.

The goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated ciphertext without using the key. If a really good encryption algorithm is used, there is no technique significantly better than methodically trying every possible key. For such an algorithm, the longer the key, the more difficult it is to decrypt a piece of ciphertext without possessing the key.

It is difficult to determine the quality of an encryption algorithm. Algorithms that look promising sometimes turn out to be very easy to break, given the proper attack. When selecting an encryption algorithm, it is a good idea to choose one that has been in use for several years and has successfully resisted all attacks.

In reference to digital security, nonrepudiation means to ensure that a transferred message has been sent and received by the parties claiming to have sent and received the message. Nonrepudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.