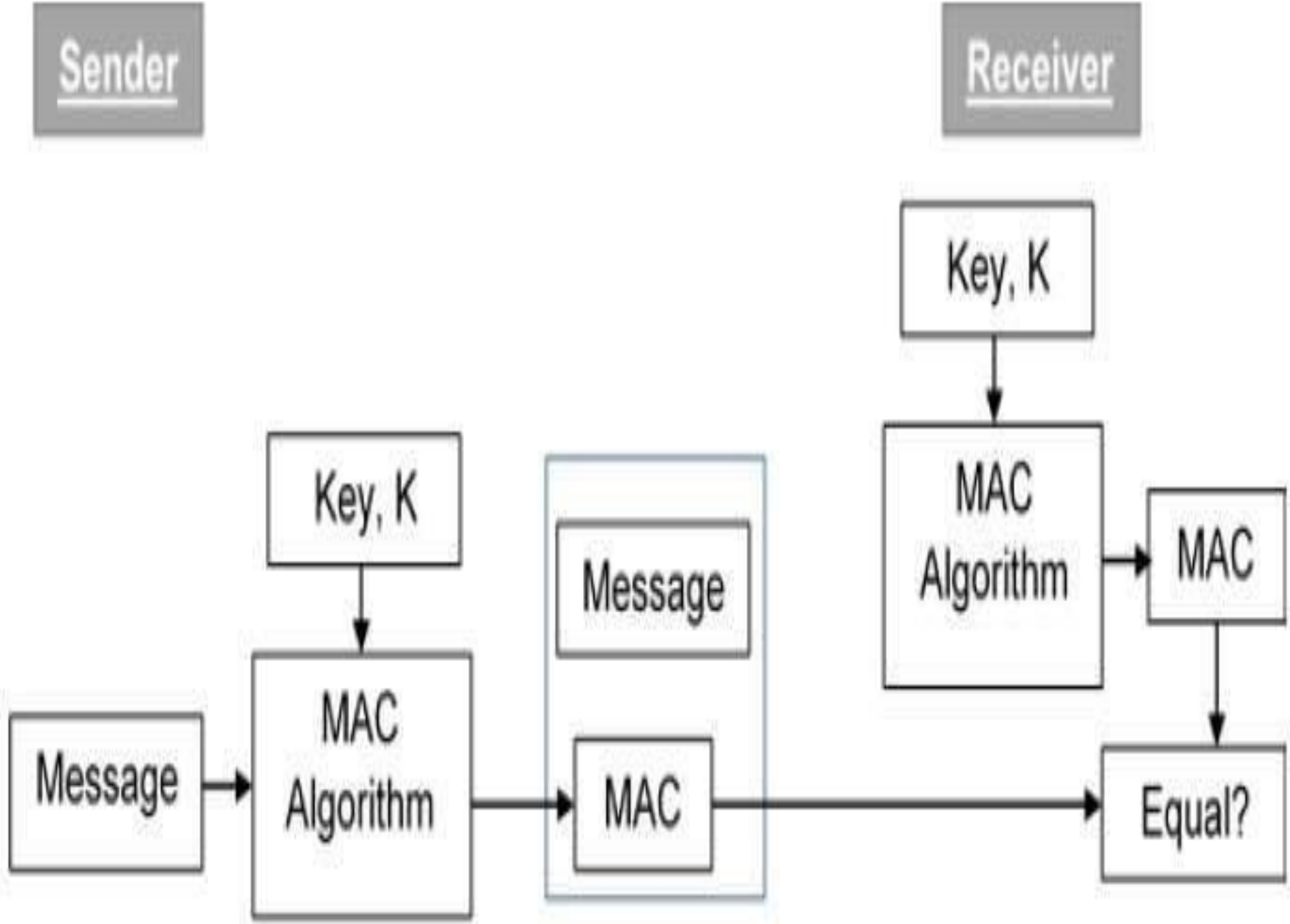
Cryptographic Hash Function

Message Authentication Code (MAC)

- MAC algorithm is a symmetric key cryptographic technique to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key K.
- Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication.
- The process of using MAC for authentication is depicted in the following illustration –
-



Sender

Receiver

Key, K

Message

MAC Algorithm

Message

MAC

Key, K

MAC Algorithm

MAC

Equal?

- Let us now try to understand the entire process in detail −
- The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.
- Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses secret key during the compression.
- The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned of providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.

- On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
- The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.
- If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not the genuine.

Key Management in Cryptography

- In cryptography, it is a very tedious task to distribute the public and private keys between sender and receiver. If the key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

Distribution of Public Key:

- The public key can be distributed in four ways:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates.

1. **Public Announcement:** Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.

2. **Publicly Available Directory:** In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

- **Public Key Authority:** It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

- **Public Certification:** This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority’s public key.

- First sender and receiver both request CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communication.

Kerberos

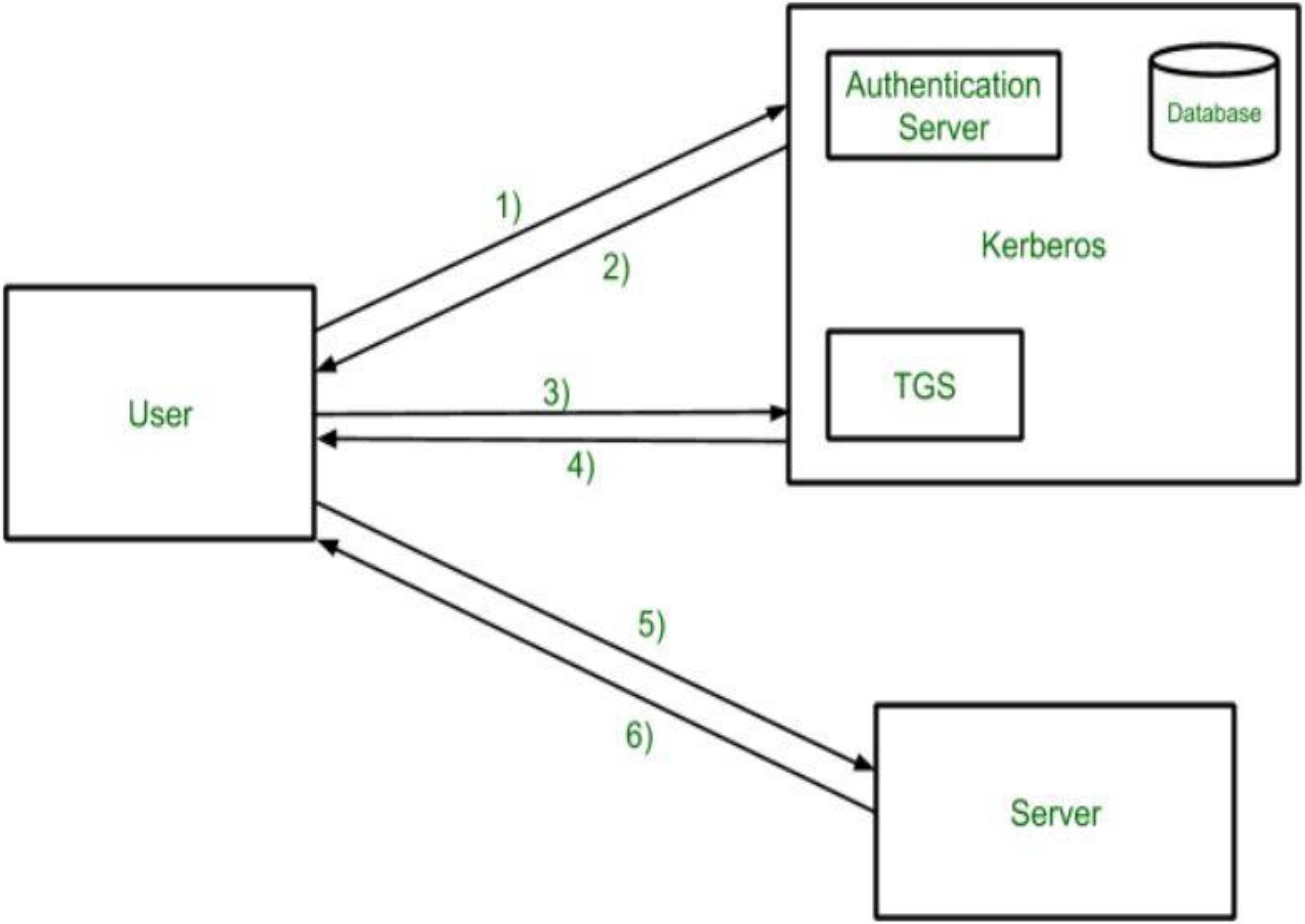
- **Kerberos** provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.

- The main components of Kerberos are:

- **Authentication Server (AS):**
  The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.

- **Database:**
  The Authentication Server verifies the access rights of users in the database.

- **Ticket Granting Server (TGS):**
  The Ticket Granting Server issues the ticket for the Server


Authentication Server
Database
Kerberos
TGS
User
1)
2)
3)
4)
5)
6)
Server

- **Step-1:**
  User login and request services on the host. Thus user requests for ticket-granting service.

- **Step-2:**
  Authentication Server verifies user’s access right using database and then gives ticket-granting-ticket and session key. Results are encrypted using the Password of the user.

- **Step-3:**
  The decryption of the message is done using the password then send the ticket to Ticket Granting Server. The Ticket contains authenticators like user names and network addresses.

- **Step-4:**
  Ticket Granting Server decrypts the ticket sent by User and authenticator verifies the request then creates the ticket for requesting services from the Server.

- **Step-5:**
  The user sends the Ticket and Authenticator to the Server.

- **Step-6:**
  The server verifies the Ticket and authenticators then generate access to the service. After this User can access the services.

X.509 Authentication Service

- X.509 is a digital certificate that is built on top of a widely trusted standard known as ITU or International Telecommunication Union X.509 standard, in which the format of PKI certificates is defined. X.509 digital certificate is a certificate-based authentication security framework that can be used for providing secure transaction processing and private information. These are primarily used for handling the security and identity in computer networking and internet-based communications.

- **Version number:** It defines the X.509 version that concerns the certificate.
- **Serial number:** It is the unique number that the certified authority issues.
- **Signature Algorithm Identifier:** This is the algorithm that is used for signing the certificate.
- **Issuer name:** Tells about the X.500 name of the certified authority which signed and created the certificate.

- **Period of Validity:** It defines the period for which the certificate is valid.
- **Subject Name:** Tells about the name of the user to whom this certificate has been issued.
- **Subject’s public key information:** It defines the subject’s public key along with an identifier of the algorithm for which this key is supposed to be used.

- **Extension block:** This field contains additional standard information.
- **Signature:** This field contains the hash code of all other fields which is encrypted by the certified authority private key.

Public Key Infrastructure (PKI)

- PKI provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.

Entity

Certification Authority.

Registration Authority.

Certificate Repository

- Registration Authority (RA)
- CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity. The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

- Certifying Authority (CA)
- As discussed above, the CA issues certificate to a client and assist other users to verify the certificate. The CA takes responsibility for identifying correctly the identity of the client asking for a certificate to be issued, and ensures that the information contained within the certificate is correct and digitally signs it.

Entity: Entity is nothing but it is the user of

PKI. Who is using this it is called entity. It

Can be single person, organization, router it can be anything the group of people whoever is trying to access the PKI that entity is called entity.

- Certificate Repository: In general, repository we store something. This is used to store all the certificates and information related to the certificates store in the certificate Repository.