Unit-V
Authentication

Authentication

Authentication is the process of verifying the identity of user or information. It uses information provided to the authenticator to determine whether someone (or something) is, in fact, who (or what) it is declared to be.

Authentication process can be described in two distinct phases: identification and actual authentication (verification phase).

1. ***Identification Phase:*** Identification phase provides a user identity to the security system. This identity is provided in the form of a user ID. The security system will search all the abstract objects that it knows and find the specific one of which the actual user is currently applying. Once this is done, the user has been identified.
2. ***Verification Phase:*** The fact that the user claims does not necessarily mean that this is true. An actual user can be mapped to other abstract user object in the system, and therefore be granted rights and permissions to the user and user must give evidence to prove his identity to the system. The process of determining claimed user identity by checking user-provided evidence is called authentication and the evidence which is provided by the user during process of authentication is called a credential.

There are four general means of authenticating a user's identity, which can be used alone or in combination:

1. ***Something the individual knows***
   - Password, a personal identification number (PIN)
2. ***Something the individual possesses***
   - Cryptographic keys, electronic keycards, smart cards, and physical keys.
3. ***Something the individual is (static biometrics)***
   - Recognition by fingerprint, retina, and face.
4. ***Something the individual does (dynamic biometrics)***
   - Recognition by voice pattern, handwriting characteristics, and typing rhythm.

Types of Authentication

There are two basic types of authentication: non-repudiable and repudiable. Other types of authentication include user, client and session authentication.

- ***Non-repudiable Authentication:*** It involves characteristics whose proof of origin cannot be denied. Such characteristics include biometrics like iris pattern, retinal images, and hand geometry and they positively verify the identity of the individual.
- ***Repudiable Authentication:*** It involves factors, "what you know" and "what you have", that can present problems to the authenticator because the information presented can be unreliable because problems including the fact that possessions can be lost, forged, or easily duplicated.

Authentication System

Authentication system consists of five components that are required for overall authentication process and they are as follows:

- ***Authentication Information (A)***→ information that proves identity.
- ***Complementary Information(C)***→ information stored on computer and used to validate authentication information.
- ***Complementation function (F)***→ function that generates the complementary information from the authentication information.
- ***Authentication Function (L)***→ function that proves identity.
- ***Selection function (S)***→ function enabling entity to create or alter information *A* or *C*.

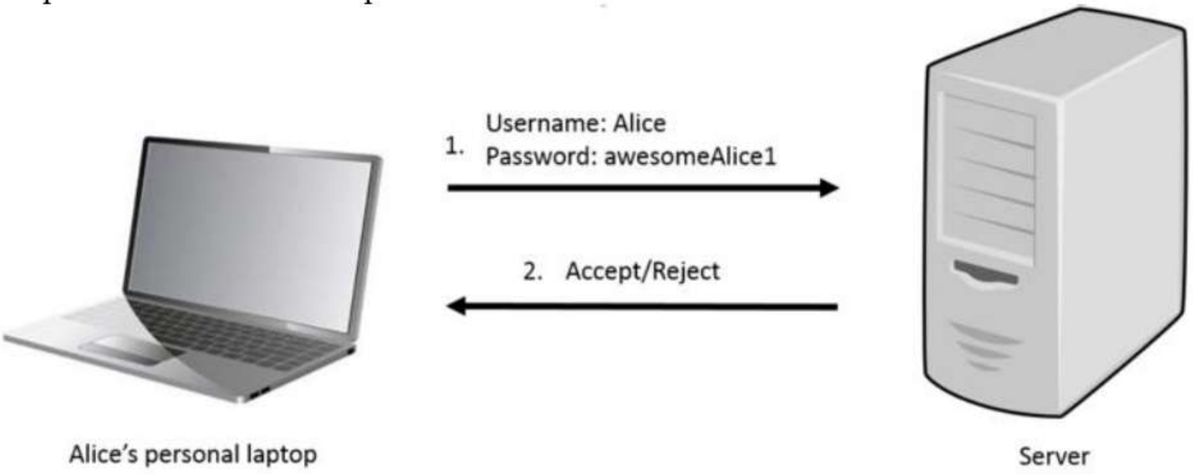
Password Based Authentication

A **password** is an information associated with an entity that confirms the entity's identity. It is a string of alphabets, numbers and special characters, which is supposed to be known only to the entity (usually person) that is being authenticated.

Password based authentication involves authenticating a client (user) by using name and password. The user requests a resource controlled by the server. The server requires client authentication before permitting access to the requested resource.

- In response to an authentication request from the server, the client displays a dialog box requesting the user's name and password for that server. The user supplies a name and password separately for each new server the user uses during a work session.
- The client sends the name and password across the network, either in the clear or over an encrypted SSL connection.
- The server looks up the name and password in its local password database and, if they match, accepts them as evidence authenticating the user's identity.
- The server determines whether the identified user is permitted to access the requested resource, and if so, allows the client to access it.

With this arrangement, the user supplies a new password for each server, and the administrator keeps track of the name and password for each user.



Password Aging

Password aging is the requirement that a password be changed after some period of time has passed or after some event has occurred.

- If change time is too short, users have difficulty recalling passwords.
- Cannot allow users to change password to current one.
- Also prevents users from changing passwords too soon.
- Give notice of impending password change requirement.

Dictionary Attack

Dictionary attack is the guessing of password by repeated trial and error. A dictionary attack attempts to defeat an authentication mechanism by systematically entering each word in a dictionary as a password or trying to determine the decryption key of an encrypted message or document.

A dictionary attack can be performed both ***online*** and ***offline***.

1. ***Online Dictionary Attack***

In online dictionary attack, the attacker tries to guess the correct password by interacting with the login server. The attacker repeatedly tries to login or gain access like any other user. This type of attack works better if the hacker has a list of likely passwords. If the attack takes too long, it might get noticed by a system administrator or the original user.

Online attack can be very slow because the speed of attack depends on the speed of the internet connection and the speed of the target server.

Defense: maximizing the time to guess the password, exponential backoff, disconnection, disabling, and jailing.

2. ***Offline Dictionary Attack***

In offline dictionary attack, the attacker first collects message between the users and servers or finds a copy of the password file. Then, the attacker tries to guess correct password by matching the passwords in his dictionary with the collected information without requiring any feedback from the login server.

Attackers need to get their hands on the password storage file from the system they want to access, so it’s more complicated than an online attack. But once they have the correct password, they will be able to log in without anyone noticing.

Defense: append the password with a random string (salt) and hash the result. E.g.

User ID: Alice
Salt value: 12598
Password hash: Hash (12598, password-Alice)

Q. How Jailing and Backoff can be used to demotivate online dictionary attack in authentication system?

Sol^n:

Dictionary attack is the guessing of password by repeated trial and error. Jailing and Backoff helps to slow down these attacks.

Jailing:

Jailing means giving the attacker limited access to the system without the ability to make actual change but making the attacker believe that he has complete access to the system when wrong credentials has been repeatedly given for some set numbers of times. This technique is used to determine what the attacker wants or simply to waste the attacker's time.

Backoff:

Backoff means the amount of time to wait before next attempt keep increasing with each wrong input. The most common form is the exponential backoff.

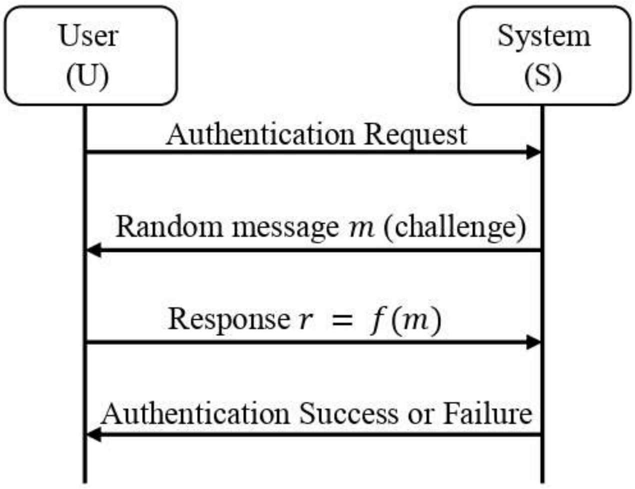
Let x be the parameter selected by system administrator; system waits for $x^0 = 1$ sec before reprompting the user. If the system fails, again waits for $x^1 = x$ sec. After n failures, waits for x^{n-1} sec.

Challenge Response System

A fundamental problem with passwords is that passwords are used repeatedly. When a password is intercepted, the authentication system cannot determine if the real user is entering the password or if an imposter is supplying the password. One strategy to counter this situation is to allow a password to be used only once. There are several different types of challenge-response systems.

Def^n: Let user U desire to authenticate himself to system S. Let U and S have an agreed-on secret function f. A challenge-response authentication system is one in which S sends a random message m, the challenge, to U. U replies with the transformation $r = f(m)$, the response. S validates r by computing r separately.

Here, the password is not transmitted during the authentication.


User
(U)
System
(S)
Authentication Request
Random message m (challenge)
Response $r = f(m)$
Authentication Success or Failure

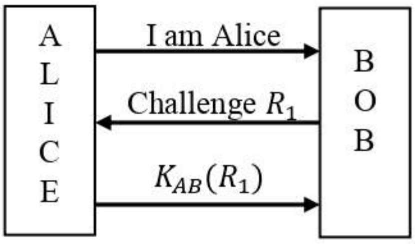
In a challenge-response authentication system in which the function f is a secret, the function f is called a pass algorithm.

One-Time Passwords

Once the password is used it is no more valid for the further usage, this type of password is called one-time password.
In case of challenge response mechanism, challenge is the number associated with the authentication attempt; response is the one-time password for that authentication prompt.

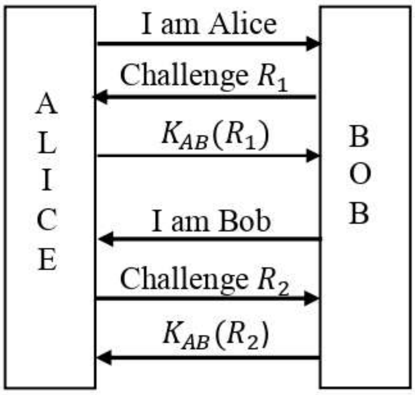
One Way Authentication and Mutual Authentication

- In ***one-way authentication***, only one party verifies the identity of the other party.



Assume that Alice and Bob share a secret key K_{AB}.

- In ***mutual authentication***, both communicating parties verify each other's identity.



Biometric Based Authentication

Biometric authentication is a user identity verification process that involves biological input, or the scanning or analysis of some part of the body. Whenever the user accesses the system, the biometric authentication mechanism verifies the identity.

Process:

- The user database contains a sample of user's biometric characteristics.
- During authentication process, the user is required to provide a new sample of the user's biometric characteristic.
- This is matched with the one in the database, and if the two samples are same, the user is considered to be a valid one.
- The samples produced during every authentication process can vary slightly. (e.g. cuts on the finger)
- An approximate match can be accepted.

Types of Biometrics

Biometric characteristics can be divided in two main classes, as represented in the following figure:

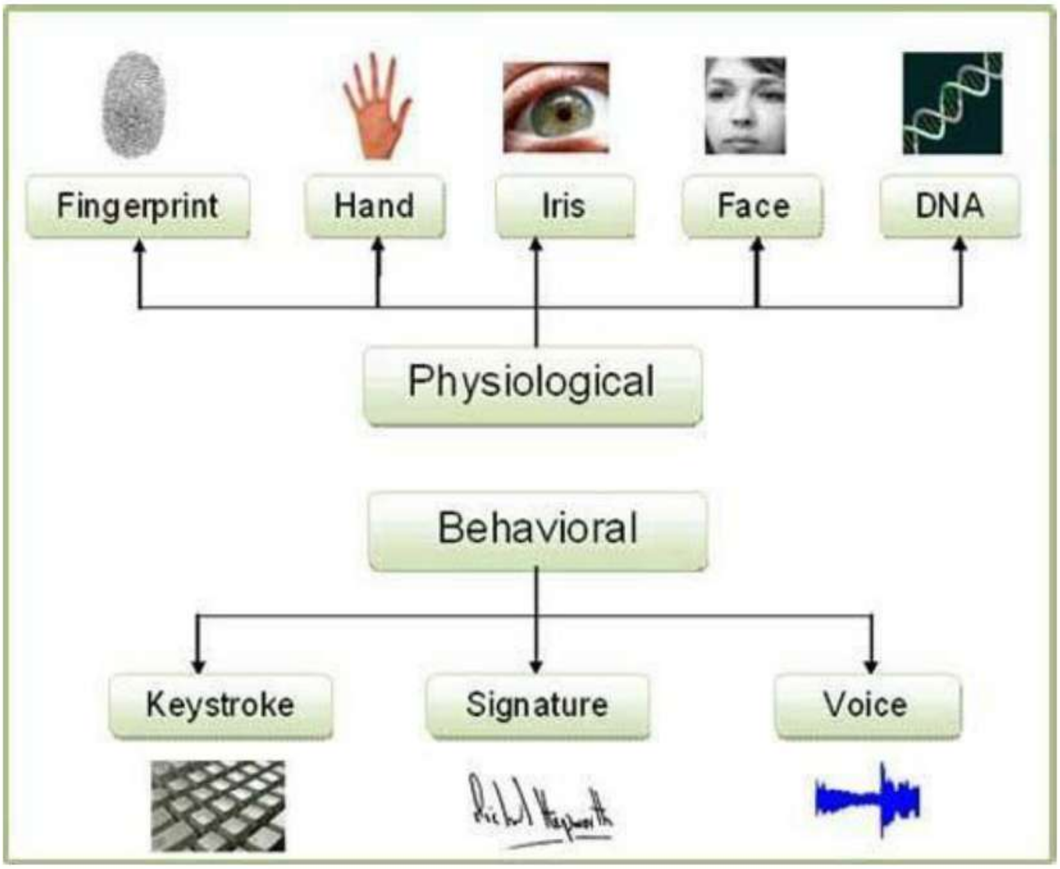


Fig: Types of Biometrics

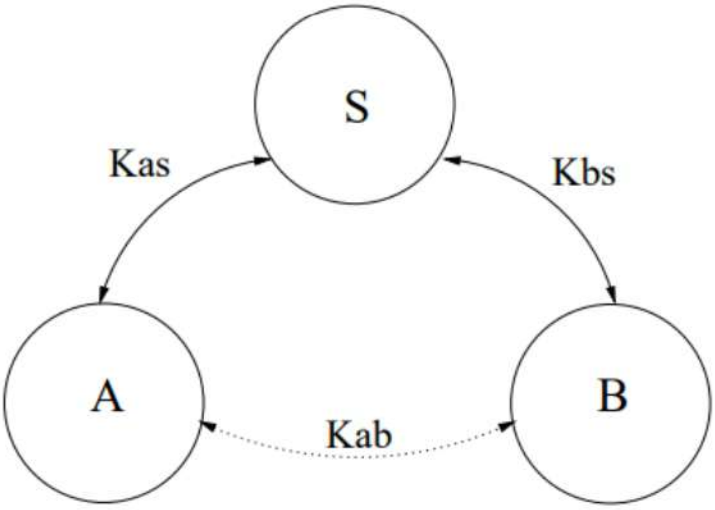
1. ***Physiological Characteristics*** are related to the shape of the body. Examples include, but are not limited to fingerprint, face recognition, hand and palm geometry and iris recognition.
   - ***Finger Print Recognition:*** The use of the ridges and valleys (minutiae) found on the surface tips of a human finger to identify an individual.
   - ***Hand Geometry Recognition:*** The use of the geometric features of the hand such as the lengths of fingers and the width of the hand to identify an individual.
   - ***Iris Recognition:*** The use of the features found in the iris to identify an individual.
   - ***Retina Recognition:*** The use of patterns of veins in the back of the eye to accomplish recognition.
   - ***Face Recognition:*** The analysis of facial features or patterns for the authentication or recognition of an individual's identity. Most face recognition systems either use eigenfaces or local feature analysis.
   - ***DNA Matching:*** The identification of an individual using the analysis of segments from DNA.

2. ***Behavioral Characteristics*** are related to the behavior of a person. Characteristic implemented by using biometrics are signature verification, keystroke dynamics, and voice.
   - ***Signature Recognition:*** Biometric signature recognition systems will measure and analyze the physical activity of signing, such as the stroke order, the pressure applied and the speed. Some systems may also compare visual images of signatures, but the core of a signature biometric system is behavioral, i.e. how it is signed rather than visual, i.e. the image of the signature.
   - ***Voice Recognition:*** Voice or speech recognition is the ability of a machine or program to receive and interpret dictation, or to understand and carry out spoken commands. Strictly speaking, voice is also a physiological trait because every person has a different pitch, but voice recognition is mainly based on the study of the way a person speaks, commonly classified as behavioral.
   - ***Key Stroke Recognition:*** Keystroke recognition is a behavioral biometric which authenticates an individual not on the basis of what is typed but the nature of how it is typed (such as typing pattern, rhythm and speed).

Needham-Schroeder(N-S) Authentication Protocol

Needham-Schroeder is a shared key authentication protocol designed to generate and propagate a session key i.e. a shared key for subsequent symmetrically encrypted communication.

There are three principals: A and B, two principals desiring mutual communication, and S, a trusted key server.



It is assumed that A and B already have secure symmetric communication with S using keys K_{as} and K_{bs} , respectively. K_{ab} is a symmetric, generated key, which will be the session key of the session between A and B.

The protocol is based on the generation and transmission of the ticket by the authentication server. A ticket is an encrypted message containing a secret key for use in communication between A and B.

N-S uses nonces (short for "numbers used once"), randomly generated values included in messages.

The Needham-Schroeder protocol can be specified as follows in security protocol notation:

1.	$A \rightarrow S:\ A, B, N_a$	A requests S to supply a key for communication with B.
2.	$S \rightarrow A$: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$	S returns a message encrypted in A's secret key, containing a newly generated key K_{ab}, and a 'ticket' encrypted in B's secret key. The nonce N_a demonstrates that the message was sent in response to the preceding one. A believes that S sent the message because only S knows A's secret key.
3.	$A \rightarrow B$: $\{K_{ab}, A\}_{K_{bs}}$	A sends the 'ticket' to B.
4.	$B \rightarrow A$: $\{N_b\}_{K_{ab}}$	B decrypts the ticket and uses the new key K_{ab} to encrypt another nonce N_b.
5.	$A \rightarrow B$: $\{N_b - 1\}_{K_{ab}}$	A demonstrates to B that it was the sender of the previous message by returning an agreed transformation of N_b.

Here N_a and N_b are nonces.

Weakness:

If a session key between A and B is compromised, and the ticket to B is recorded, an intruder can impersonate A by carrying out last 3 steps.

The weakness can be remedied by adding a timestamp to message 3. T is a timestamp that assures A and B that the session key has only just been generated. If intruder even with knowledge of an old session key, cannot succeed because a reply of step 3 will be detected by B as untimely.

Kerberos Protocol

Kerberos is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed it primarily at a client-server model and it provides mutual authentication- both the user and the server verify each other's identity.

Kerberos was designed to authenticate user requests for network resources. Kerberos is based on the concept of a trusted third party that performs secure verification of users and services. This trusted third party is called the key distribution center (KDC), sometimes also called the authentication server. The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues 'tickets' to users.

*A **ticket** is something a clients present to an application server to demonstrate the authenticity of its identity.*

Main entities involved in Kerberos flow are:

- ***Client:*** Initiates the communication for a service request. Clients are applications acting on behalf of user who need access to a resource or service.

- ***Server:*** The server with the service the user wants to access.
- ***Authentication Sever (AS):*** Performs client authentication. If the client is authenticated successfully the AS issues a ticket called TGT (Ticket Granting Ticket). TGT proves to other servers that client has been authenticated.
- ***Key Distribution Center (KDC):*** In Kerberos environment authentication sever is logically separated into three parts: Database (DB), Authentication Server (AS) and Ticket Granting Server (TGS). Physically these 3 parts are existing in a single server and it is called as Key Distribution Center.
- ***Ticket Granting Server (TGS):*** An application server which provides the issuing of service tickets as a service.

The following figure shows the sequence of events required for a client to gain access to a service using Kerberos authentication:

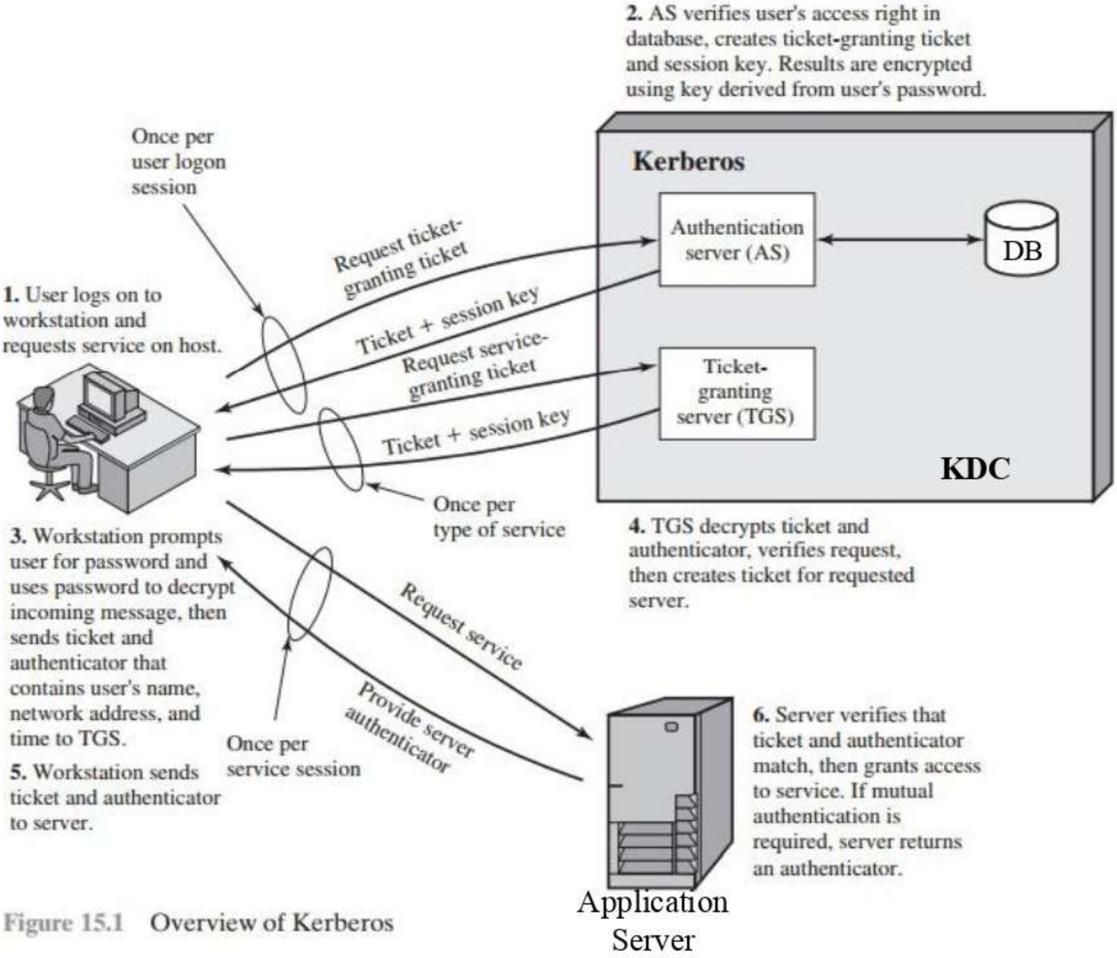

2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
Once per user logon session
Kerberos
Authentication server (AS)
DB
Request ticket-granting ticket
Ticket + session key
1. User logs on to workstation and requests service on host.
Request service-granting ticket
Ticket-granting server (TGS)
Ticket + session key
KDC
Once per type of service
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
Request service
Provide server authenticator
Once per service session
5. Workstation sends ticket and authenticator to server.
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.
Application Server

Figure 15.1 Overview of Kerberos

<u>Characteristics of Kerberos</u>

- It is secure: it never sends a password unless it is encrypted.
- Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.
- The concept depends on a trusted third party – a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.
- It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client.

Kerberos Version 4 vs Version 5

Basis of Comparison	Kerberos Version 4	Kerberos Version 5
Year of Release	Kerberos Version 4 was released in 1980's way before version 5 was released.	The Kerberos version 5 was published in 1993, 13 year after the release of Kerberos Version 4.
Principal Name	Kerberos version 4 uses the principle name partially.	Kerberos Version 5 uses the entire principal name.
Encryption Techniques	Kerberos version 4 uses DES encryption techniques.	In Kerberos version 5, the cipher text is tagged with an encryption type identifier and therefore any type of encryption can be used.
Encoding	Kerberos version 4 uses the "receiver-makes-right" encoding system.	The Kerberos version 5 uses the ASN.1 coding system.
Ticket Lifetime	In Kerberos version 4, the ticket lifetime has to be specified in units of 5 minutes.	In Kerberos version 5, ticket one lifetime can specify an explicit start and finish times allowing arbitrary lifetimes.
Ticket Support	Ticket support is satisfactory in this version.	Ticket support is well extended. Facilitates forwarding, renewing and postdating tickets.
IP Addresses	Only a few IP addresses and other addresses for other sorts of network protocols are included.	Multiple IP addresses and other addresses for various network protocols are included.

Please let me know if I missed anything or anything is incorrect.

poudeljayanta99@gmail.com