CHAPTER 6 : Authentication and Authentication Protocols

OBJECTIVES

- Access Control mechanisms
- Key Distribution Center KDC
- Kerberos : details of operation

INTRODUCTION

- Much of security is based on good access control
- Access control only works if you have good authentication
- What is authentication

Authentication

- Determining the identity of some entity
  - Process
  - Machine
  - Human user
- Requires notion of identity
- And some degree of proof of identity

Proving Identity in the Physical World

- Most frequently done by physical recognition
  - I recognize your face, your voice, your body
- What about identifying those we do not already know?

Other Physical World Methods of Identification

- Identification by recommendation
  - You introduce me to someone
- Identification by credentials
  - You show me your driver’s license
- Identification by knowledge
  - You tell me something only you know
- Identification by location
  - You are behind the counter at the DMV
- These all have cyber analogs

Differences in Cyber Identification

- Usually the identifying entity is not human
- Usually the identified entity is not human
- Often no physical presence required
- Often no later rechecks of identity

Identifying With a Computer

- Not as smart as a human
  - Steps to prove identity must be well defined
- Can’t do certain things as well
  - E.g. face recognition
- But extremely fast on computations and less prone to simple errors
  - Mathematical methods are acceptable

Identifying Computers and Programs

- No physical characteristics
  - Faces, fingerprints, etc.
- Generally easy to duplicate programs
- Not smart enough to be flexible
  - Must use methods they will understand
- Again, good at computations

Physical Presence Optional

- Often must be identified over a network or cable
- Even if the party to be identified is human
- So authentication mechanisms must work in face of network characteristics

Identity Might Not be Rechecked

- Human beings make identification mistakes
- But they often recover from them
  - Often quite easily
- Based on observing behavior that suggests identification was wrong
- Computers and programs rarely have that capability
  - If the identify something, they believe it

Authentication Mechanisms

- Something you know
  - E.g. passwords
- Something you have
  - E.g. smart cards or tokens
- Something you are
  - Biometrics
- Somewhere you are
  - Usually identifying a role

Passwords

- One of the oldest and most commonly used security mechanisms
- Authenticate the user by requiring him to produce a secret
  - Known only to him and to the authenticator
  - Or, if one-way encryption used, known only to him
- Problems :
  - Guessing
  - Susceptible to password sniffers in a network
  - Susceptible to Brute Force attacks

Passwords and Single Sign-On

- Many systems ask for password once
  - Resulting authentication lasts for an entire “session”
- Unless other mechanisms in place, complete mediation definitely not achieved
- Trading security for convenience

Handling Passwords

- The OS must be able to check passwords when users log in
- So must the OS store passwords?
- Not really
  - It can store an encrypted version
- Encrypt the offered password
  - Using a one-way function
- And compare it to the stored version

Is Encrypting the Password File Enough

- What if an attacker gets a copy of your password file?
- No problem, the passwords are encrypted
  - Right?
- Yes , but.....
  - Dictionary attacks
  - All Linux machines use the same one-way function to encrypt passwords
  - If someone runs the entire dictionary through that function will they have a a complete list of all encrypted dictionary passwords?

The Real Problem

- Not that two users may choose the same password
- But that anyone who chose that password got the same encrypted result
- So the attacker need only encrypt every possible password once
- And then he has a complete dictionary usable against anyone

Salted Passwords

- Combine the plaintext password with a random number
  - Then run it through the one-way function
- The random number need not be secret
- It just has to be different for different users

Protecting the Password File

- Is it OK to leave the encrypted version of the password file around?
- No it is not
- Why make it easy for attackers?
- Dictionary attacks against single accounts still work
- Generally, do not give access to the encrypted file, either

Password Authentication Protocol PAP

- Offers no true security
- One of the simplest forms of authentication
- Username and password values are both sent to the server as clear text and checked for match
- If they match, the user is granted access, if not the user is denied access

Challenge/Response Authentication

- Authentication by what questions you can answer correctly
  - Again, by what you know
- The system asks the user to provide some information
- If it is provided correctly, the user is authenticated

Differences from passwords

- Challenge/response systems ask for different information every time
- Or at least the questions come from a large set
- Best security achieved by requiring what amounts to encryption of the challenge
  - Special H/W required : smart card
- Problems : spoofing or questions too hard to answer
- Still in use :
  - Authentication by asking for mother’s maiden name

Identification Devices

- Authentication by what you have
- A smart card or other hardware device that is readable by the computer
- Authenticate by providing the device to the computer

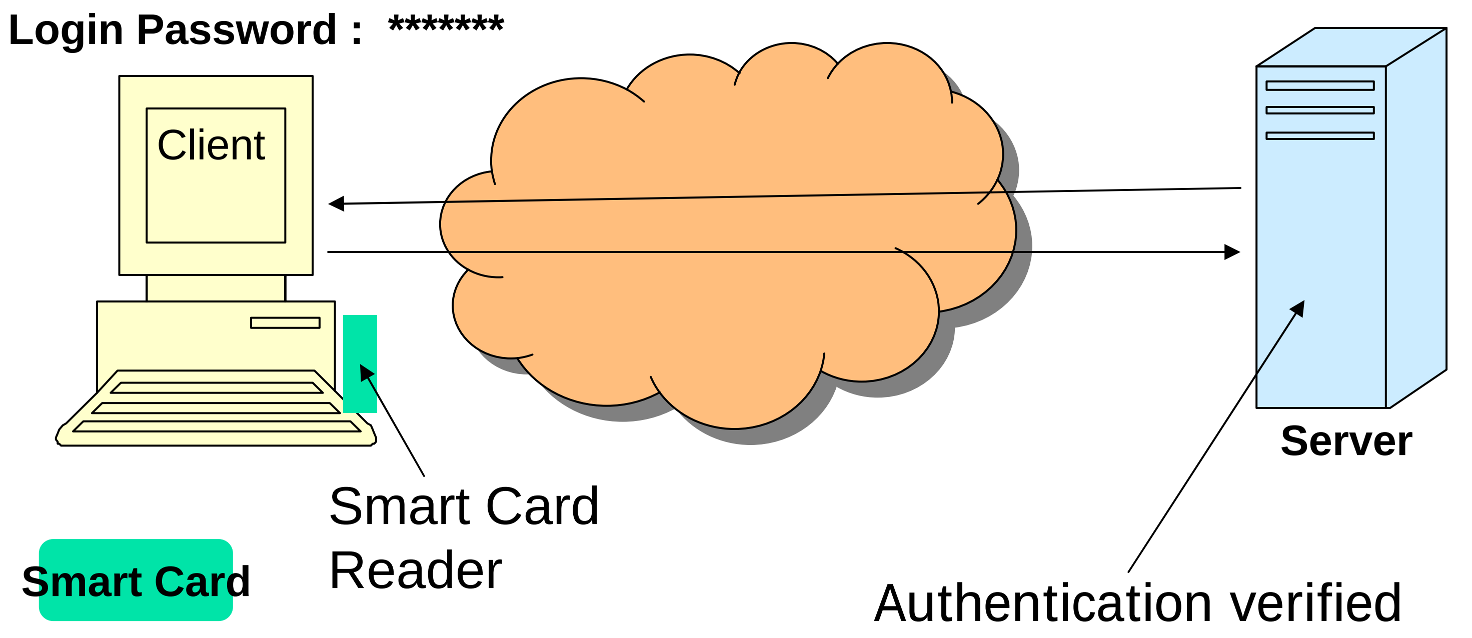
Simple Use of Authentication Tokens

- If you have the token, you are identified
- Generally requires connecting the authentication device to the computer (or wireless)
- They contain the rights and access privileges of the token bearer.
- Most common token is a plastic card with a magnetic strip
- New type : Proximity card
- Weak : because it is subject to theft and spoofing

Smart Cards

- It is a type of badge or card that gives access to resources, including buildings, parking lots, computers.
- It contains information about identity and access privileges
- Cryptography performed on smart card
  - So compromised client machine can't steal key
- Likely to use PK cryptography
- Often user must enter password to activate card...this provides some security in case card is stolen

Authentication with Smart Cards and Passwords



Two-factor Authentication

Authentication Through Biometrics

- Authentication based on who you are
- Problems :
  - Requires special hardware
  - Are not as perfect as one thinks
  - Many physical characteristics change by time
  - Generally not helpful for authenticating programs

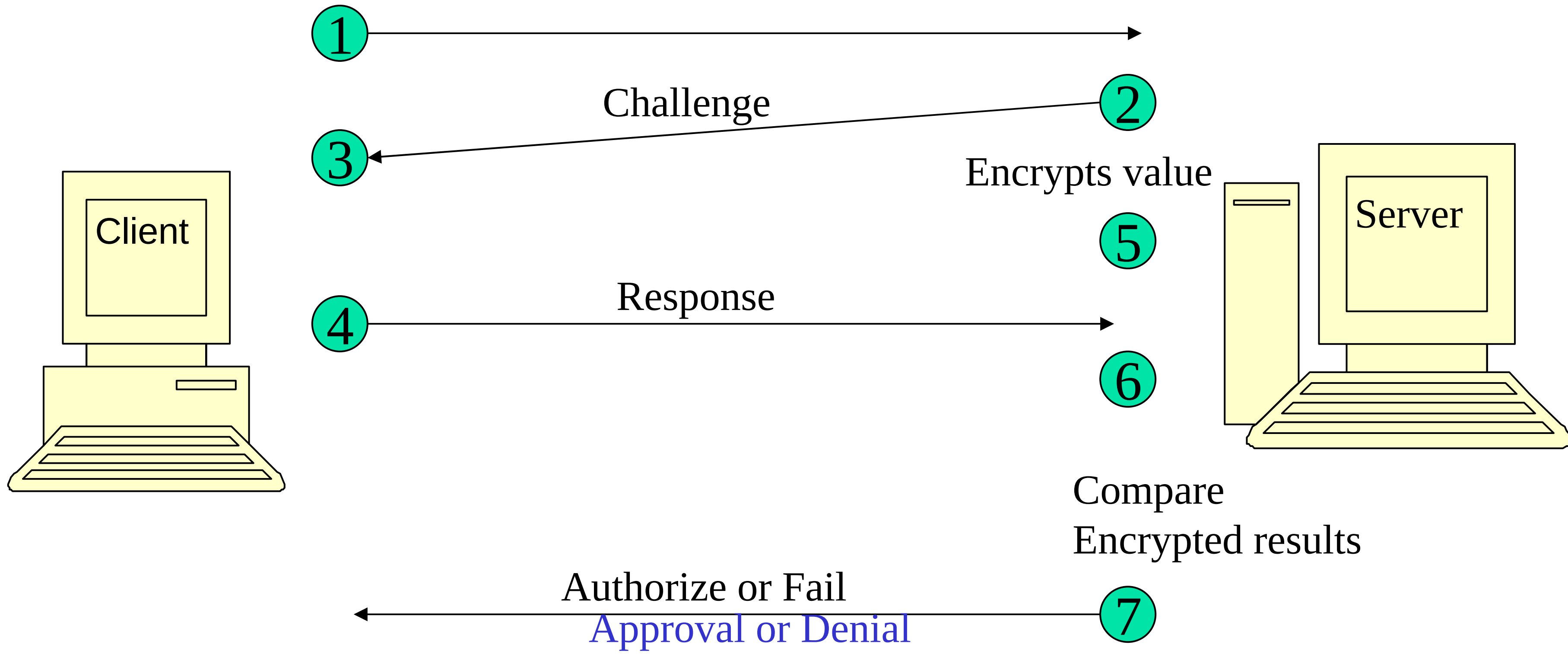
Challenge Handshake Authentication Protocol CHAP

- The Point-to-point protocol (PPP) that is the widely used protocol for establishing dial-in connections over serial lines or ISDN services , uses several authentication protocols of which CHAP is one as well as PAP.
- CHAP does not use a user ID/password mechanism.

CHAP

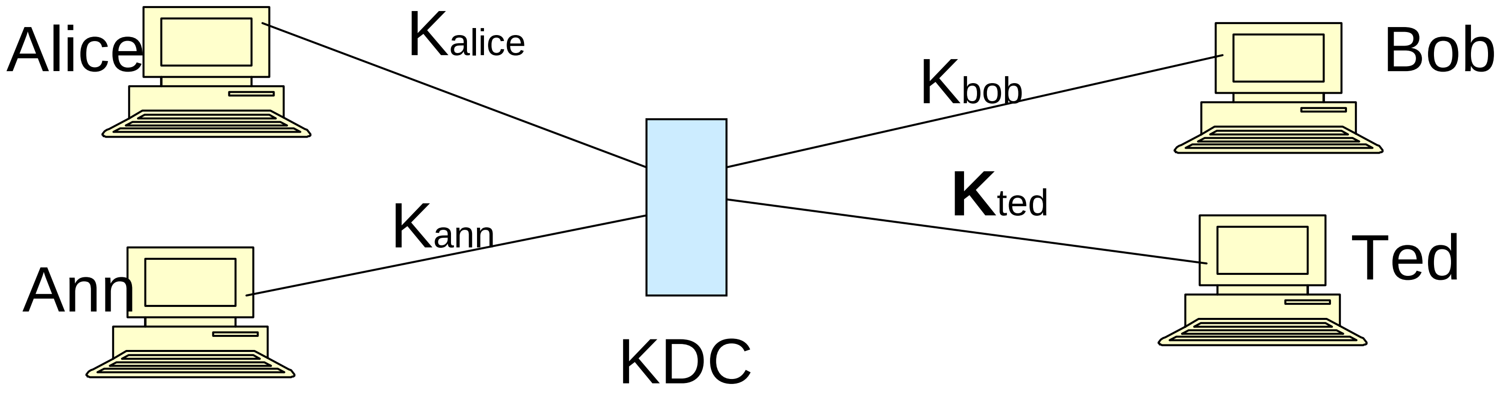
- In this authentication method, the initiator sends a logon request from the client to the server.
- The server sends a **challenge** back to the client.
- The challenge is a time-varying value (**nonce**) such as a random number or a timestamp.
- The challenge is encrypted and then sent back to the server
- The server compares the value from the client and if the information matches, grants authorization.
- If the response fails, the session fails, and the request phase starts over.
- Note: The use of nonce prevents replay attacks.
- Either symmetric or asymmetric keys may be used in encryption

CHAP For Authentication



Key Distribution Center KDC

- A practical solution for the problem of maintaining and distributing secret keys
- To reduce the number of keys, the KDC establishes for each person a shared secret key with the key distribution center (KDC)



KDC in Use

- For Alice to send a confidential message to Bob :
  - Alice sends a request to KDC, stating that she needs a session key between herself and Bob
  - KDC informs Bob of Alice’s request
  - If Bob agrees, a session key is created between the two
  - The secret key between Alice and Bob that is established with KDC is used to authenticate Alice and Bob to the KDC and prevent Eve from impersonating either of them.

KDC and Session Keys

- The member’s secret key with the KDC can only be used between the member and the KDC, not between members.
- The keys of Alice and Bob are used to authenticate Alice and Bob to the center and to each other before the session key is established .
- After communication is terminated, the session key is no longer valid.

Creating Session Keys in KDC

Step 1

- Alice sends a plaintext message to the KDC to obtain a symmetric session key between Bob and herself
- The message contains her registered identity

  And the identity of Bob
- The message is **not** encrypted, it is public.

Creating Session Keys in KDC (cont.)

Step 2

- KDC receives the message and creates what is called a **Ticket**
- The ticket is encrypted using Bob's key (K_b)
- The ticket contains the identities of Alice and Bob and the session key K_{AB}
- The ticket with a copy of the session key is sent to Alice
- Alice receives the message, decrypts it, and extracts the session key
- She cannot decrypt Bob's ticket; the ticket is for Bob not for Alice.

Creating Session Keys in KDC (cont.)

Step 3

- Alice sends the ticket to Bob.
- Bob opens the ticket and knows that Alice needs to send messages to him using KAB as the session key.
- **Note**: in this message, Bob is authenticated to the KDC because only Bob can open the ticket. Since Bob is authenticated to the KDC, he is also authenticated to Alice who trusts the KDC. In the same way, Alice is also authenticated to Bob, because Bob trusts the KDC and the KDC has sent the ticket which includes the identity of Alice.

KERBEROS

- Is a commonly used authentication method on the Internet.
- It is available in several commercial products.
- Developed by MIT’s project Athena
- Kerberos is a ticket-based network authentication protocol utilizing symmetric cryptography
- Kerberos employs a client/server architecture and provides user-to-server authentication rather than host-to-host authentication.

KERBEROS (cont.)

- It is a single sign-on technology.
- It is IP-based service.
- It uses secret-key cryptography to provide strong authentication for client/server applications.
- Every host on the network has its own secret key.
- A secure trusted host on the network, known as the Key Distribution Center KDC , knows the keys for all of the hosts (within a *realm*)

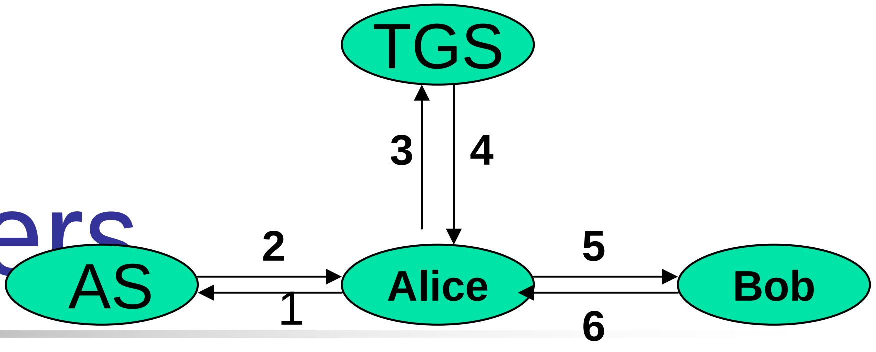
KERBEROS (cont.)

- It is an authentication protocol and at the same time a KDC that has become very popular
- Several systems including Windows 2000 use Kerberos
- Three servers are involved in the Kerberos protocol

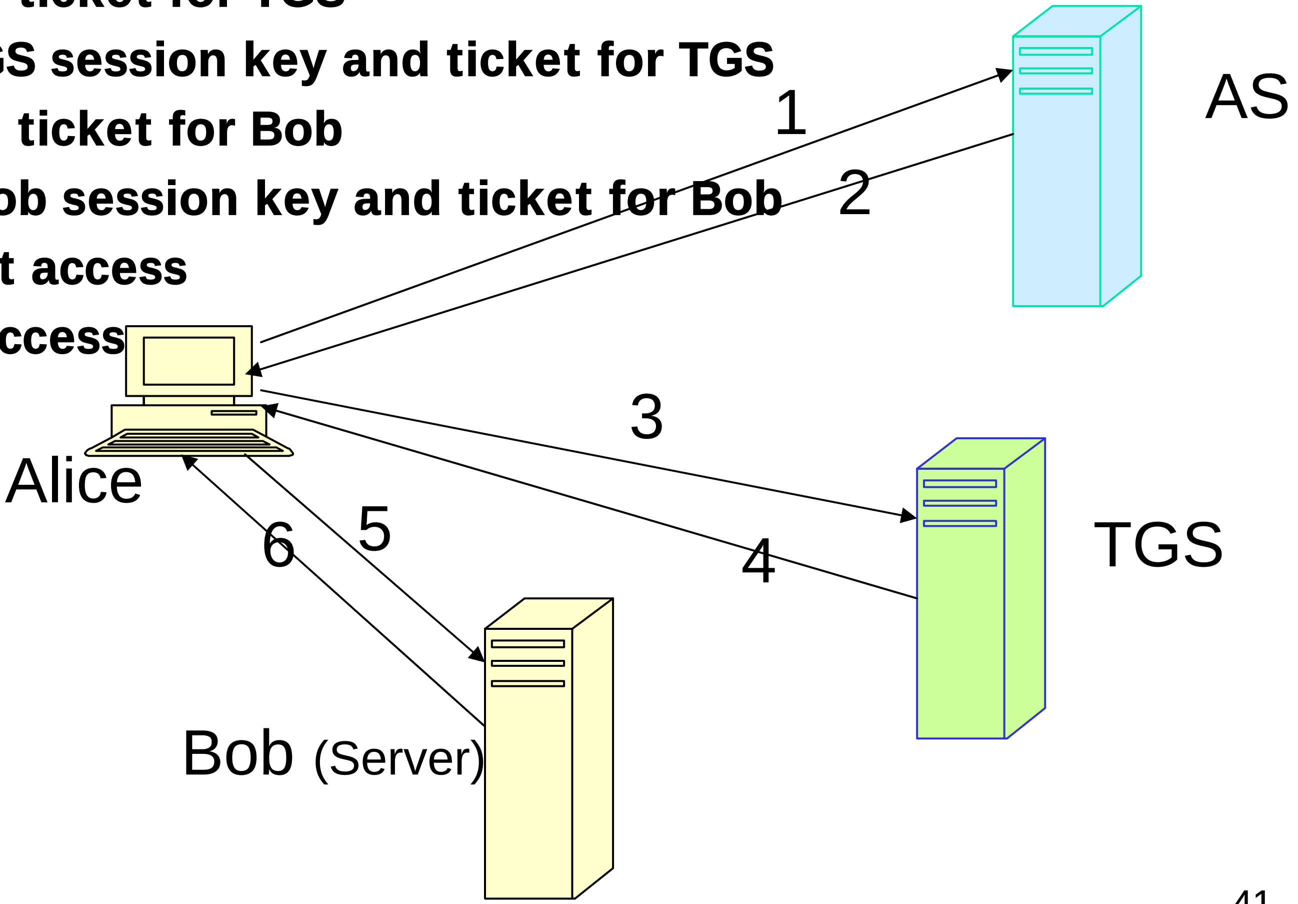
KERBEROS and Single Sign On

- Most users access many systems and networks during the day.
- A single sign-on technology allows a user to sign on once at the beginning of the day and remain authorized throughout the day on the entire network.

Kerberos Servers



- 1-Request ticket for TGS
- 2-Alice-TGS session key and ticket for TGS
- 3-Request ticket for Bob
- 4- Alice-Bob session key and ticket for Bob
- 5- Request access
- 6- Grant access



The Kerberos Servers

- Authentication Server (AS)
- Is the KDC in Kerberos protocol
- Each user registers with AS and is granted a user identity and a password.
- AS has a database with these identities and the corresponding passwords
- AS verifies the user, issues a session key to be used between Alice and TGS, and sends a ticket for TGS.

Authentication Server or KDC

- Holds secret keys for "*principals*"
  - Principal : Any object such as user, application ,service, or resource which utilizes Kerberos authentication is referred to as principal.
- A KDC is responsible for one or more "realms" of principals
- Provides authentication
  - Any principal must trust the KDC
- Creates and distributes session keys
- KDC uses symmetric cryptography

Realm

- The group or set of principals which are grouped together logically by a network administrator is called a realm
- A KDC is responsible for one or more realms

The Kerberos Servers (cont.)

- Ticket Granting Server (TGS)
- Issues a ticket for the real server (Bob).
- It also provides the session key K_{AB} between Alice and Bob
- Kerberos has separated the user verification from the ticket issuing
- *In this way, Alice verifies her ID just once with AS, she can contact TGS multiple times to obtain tickets for different real servers.*

The Kerberos Servers (cont.)

- Real Server (Bob)
- Provides services for the user (Alice)
- Kerberos is designed for a client/server program in which a user uses the client process to access the server process
- Kerberos is not used for person-to-person authentication

Kerberos Operation

A client process (Alice) can access a process running on the real server (Bob) in six steps:

- Step 1 – Alice sends her request to AS using her registered identity.

Kerberos Operation (cont.)

- **Step 2**
- AS sends a message encrypted with Alice's symmetric key K_A . The message contains :
  - a session key K_S that is used by Alice to contact TGS
  - And a ticket for TGS that is encrypted with the TGS symmetric key KTG
- Alice does not know KA, but when the message arrives, she types her symmetric password.
- The password and the appropriate algorithm together create KA if the password is correct
- The password is then immediately destroyed, it is not sent to the network . It is only used for a moment to create KA
- The process now uses KA to decrypt the message sent
- Both KS and the ticket are extracted.

Kerberos Operation (cont.)

- Step 3 :
- Alice now sends three items to TGS :
  - The ticket received from AS
  - The name of the real server (Bob)
  - A timestamp which is encrypted by K_S
- The timestamp prevents a replay by Eve

Kerberos Operation (cont.)

- Step 4
- TGS sends two tickets, each containing the session key between Alice and Bob KAB
- The ticket for Alice is encrypted with KS
- The ticket for Bob is encrypted with KB
- Eve cannot extract KAB because she does not know Ks or KB
- She cannot replay step 3 because she cannot replace the timestamp with a new one (she does not know Ks)
- Even if she is very quick and sends the step 3 message before the timestamp has expired, she still receives the same two tickets that she cannot decipher.

Kerberos Operation (cont.)

- Step 5
- Alice sends Bob's ticket with the timestamp encrypted by K_{AB}

- Step 6
- Bob confirms the receipt by adding 1 to the timestamp.
- The message is encrypted with K_{AB} and sent to Alice
- At this point, the client can initiate the intended service requests (FTP, HTTP, or e-commerce transaction session establishment)

Kerberos Operation (cont.)

- Using Different Servers
- If Alice needs to receive services from different servers, she needs to repeat steps 3 to 6
- The first two steps have verified Alice's identity and need not be repeated
- Alice can ask TGS to issue tickets for multiple servers by repeating steps 3 to 6.

Some Features and Problems of Kerberos

- KDC can be a single point of failure
- Secret keys stored on workstations
- Vulnerable to dictionary attacks
- If encryption is not enabled, the network is not protected
- All principals must have Kerberos software installed
- The KDC must be readily available at all times and must be able to support the number of requests that it receives from all the principals in the realm.

Public–Key Distribution

- Public announcement
- Trusted center
- Controlled trusted center
- Certification authority CA
- When we want to use public key universally, we cannot have one KDC , a hierarchical structure PKI is needed