Chapter 15

Key Management

#Symmetric key Distribution :-

Symmetric-key cryptography is more efficient then asymmetric-key cryptography for enciphering large message.

~~How~~

If Alice needs to communicate with N people She need : N keys.

But If N people wants to communicate with each other then they need N(N-1)/2 keys.

~~Ends~~

In simple term required Number of keys for N people is Approximatly N^2.

The Number of key Not a only problem, the distribution of keys is another problem.

If Alice & Bob want to communicate, they need a way to exchange a secret key.

If Alice ward to communicate with one million people, then ~~they~~ How can she exchange one million keys with one million people ??

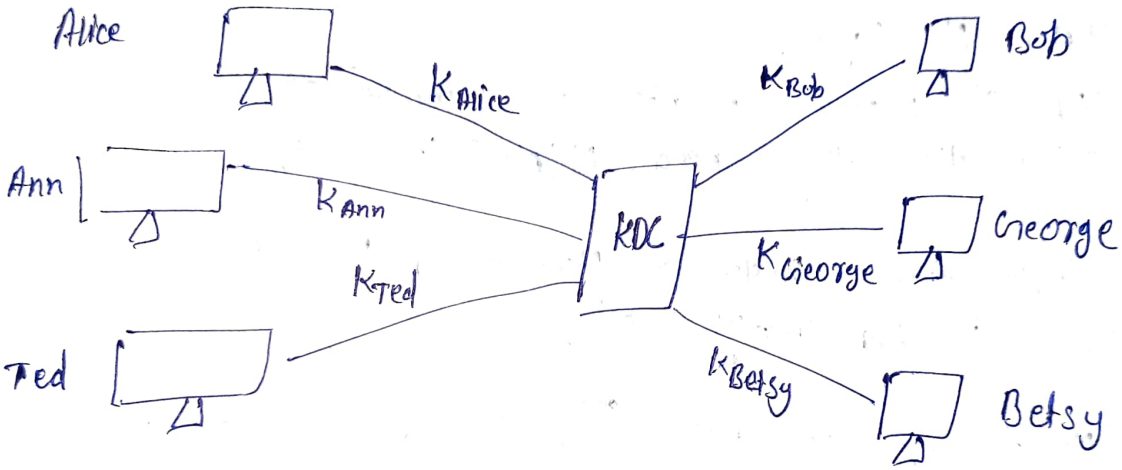
Using the internet definitely not secure method.

It is obvious that we need an efficient way to maintain and distribute secret keys.

Key - Distribution center : KDC

A practical solution is that the use of a third party, referred to as a "key-distribution center (KDC)".

To reduce the no. of keys, each person establishes a shared secret key with the KDC.



A secret key is established b/w KDC & each member.

Alice has a secret key with the KDC, which we refer to as K_{Alice}.

Bob has a secret key with the KDC, which we refer to as K_{Bob} and so on.

How Alice send confidencial message to Bob?—

(1) Alice sends a request to the KDC stating that she need a session secret key b/w herself & Bob.

2. The KDC informs Bob about Alice's request.
3. If Bob agrees, a session key is created b/w the two.
4. The secret key b/w Alice & Bob that is established with the KDC is used to authenticate Alice and Bob to the KDC and to prevent Eve from impersonating either of them.

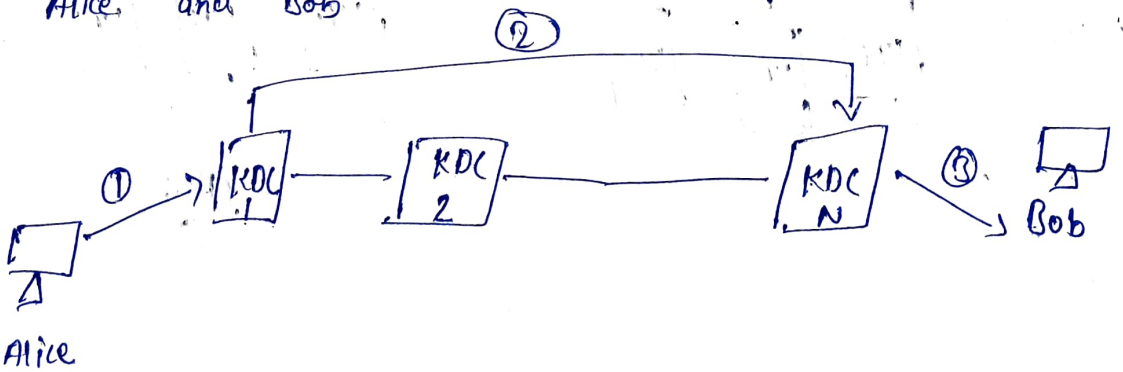
Types of KDC :-

(1) Flat Multiple KDC :-

If the Number of people using a that KDC is increases then the system become Unmanageble.

To Solve this problem, we need to have multiple KDCs. we can divide the world into Domains.
Each domain can have one or more KDCs.
Now if Alice wants to send a confidential message to Bob, who belongs to another domain, Alice contacts her KDC which in turn contacts the KDC in Bob's Domain.

The two KDCs can create a secret key between Alice and Bob.

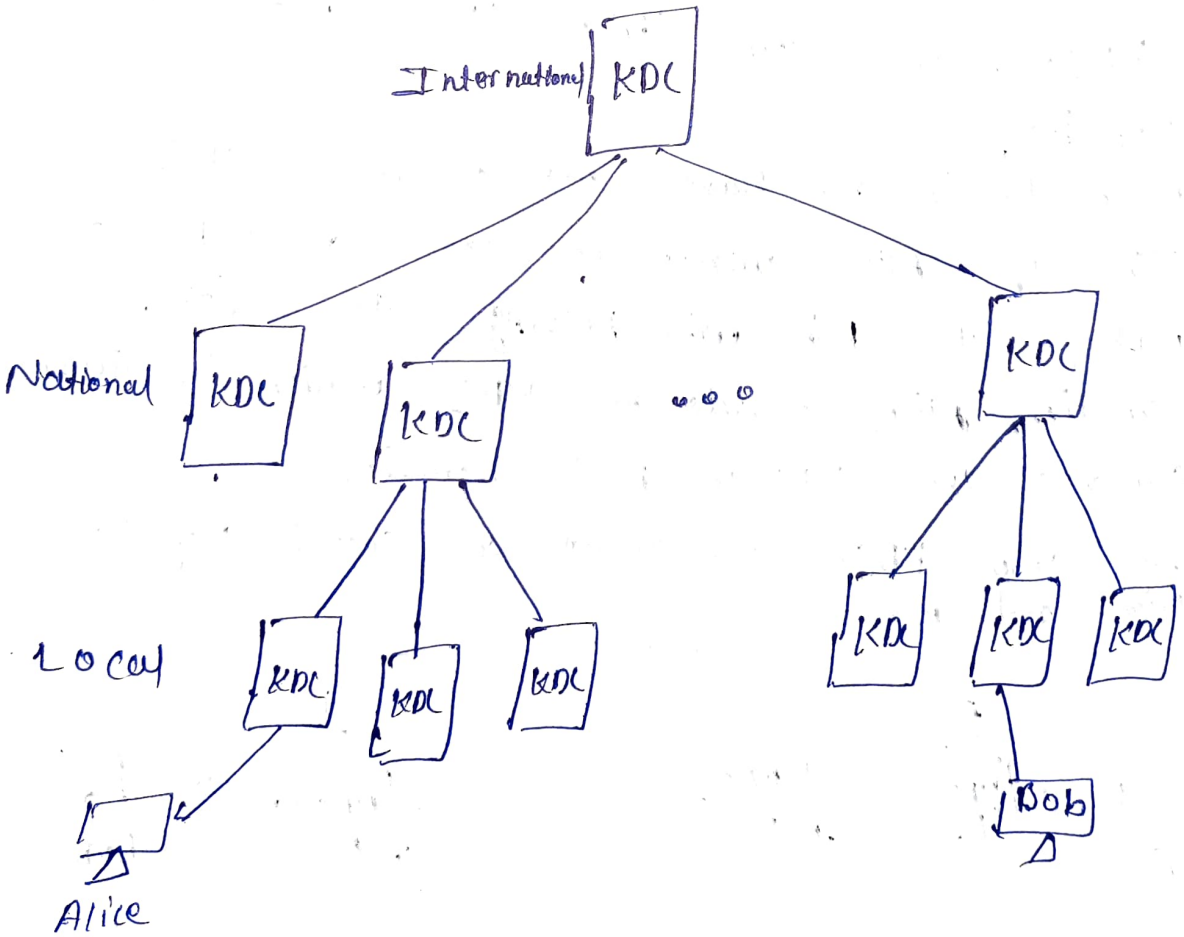


② Hierarchical multiple KDC:-

The concept of flat multiple KDC can be extended to a hierarchical system of KDCs, for Example:-

There are local KDC, ~~nat~~ national KDC, international KDCs, when Alice needs to communicate with Bob, who lives in another country, she sends her request to a local KDCs, the local KDC relays the request to the national KDC, the national KDC relays the request to an international KDC.

The request is then relayed all the way down to the local KDC where Bob lives.



Session key :-

A KDC creates a secret key for each member This's secret key can be used only blw the member and the KDC, not blw two members. If Alice needs to communicate secretly with Bob, She need secret key blw herself & Bob.
A KDC can creates a session key blw Alice & Bob using their keys with the center. The keys of Alice & Bob are used to authenticate Alice & Bob to the center and to each other before session key is established. After communication is terminated, the session key is no longer useful.

Approches for creating session key :-

1. Needham - Schroeder Protocol :-

It is a foundation of other Protocols.
Needham & Schroeder uses two nonce R_A & R_B.

five steps used in this protocol :-

(1) Alice send a message to the KDC that includes :-
her nonce R_A, her identity & Bob's identity.

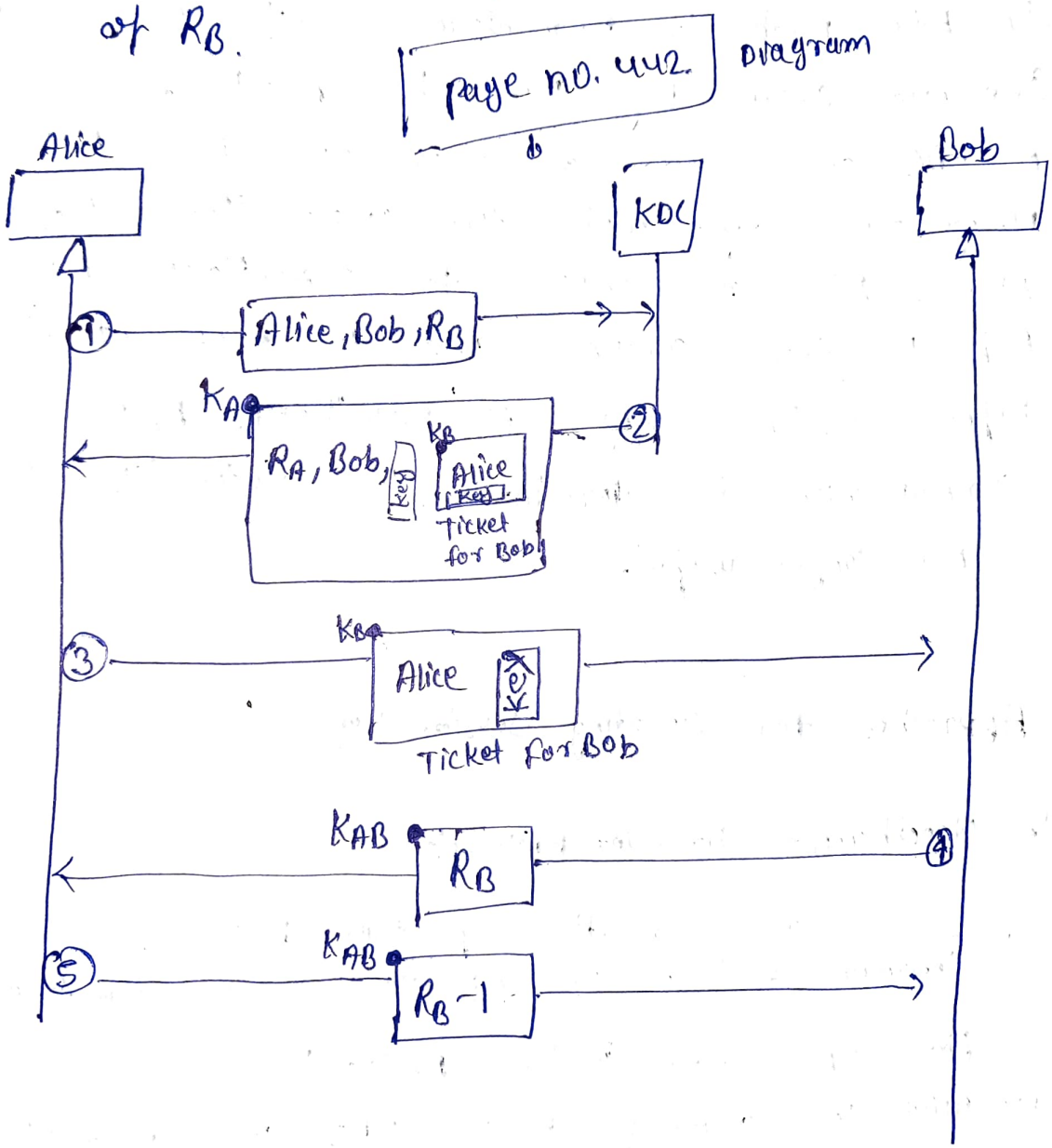
(2) The KDC sends encrypted message to Alice that includes Alice's nonce Bob's identity, the session key and an encrypted ticket for Bob. the whole message is encrypted with Alice's key

③ The Alice sends Bob's ticket to him.

④ Bob sends his challenge to Alice (R_B) encrypted with the session key.

⑤ Alice responds to Bob's challenge.
Note that the response carries $R_B - 1$ instead of R_B.

Page no. 442.



KERBEROS

Kerberos is an authentication protocol, and at the same time a KDC, that has become very popular. Several system, including windows 2000, use kerberos. It is named after the three-headed dog in Greek mythology that guards the gates of Hades. Originally designed at MIT, it has a several version But we discuss only version 4.

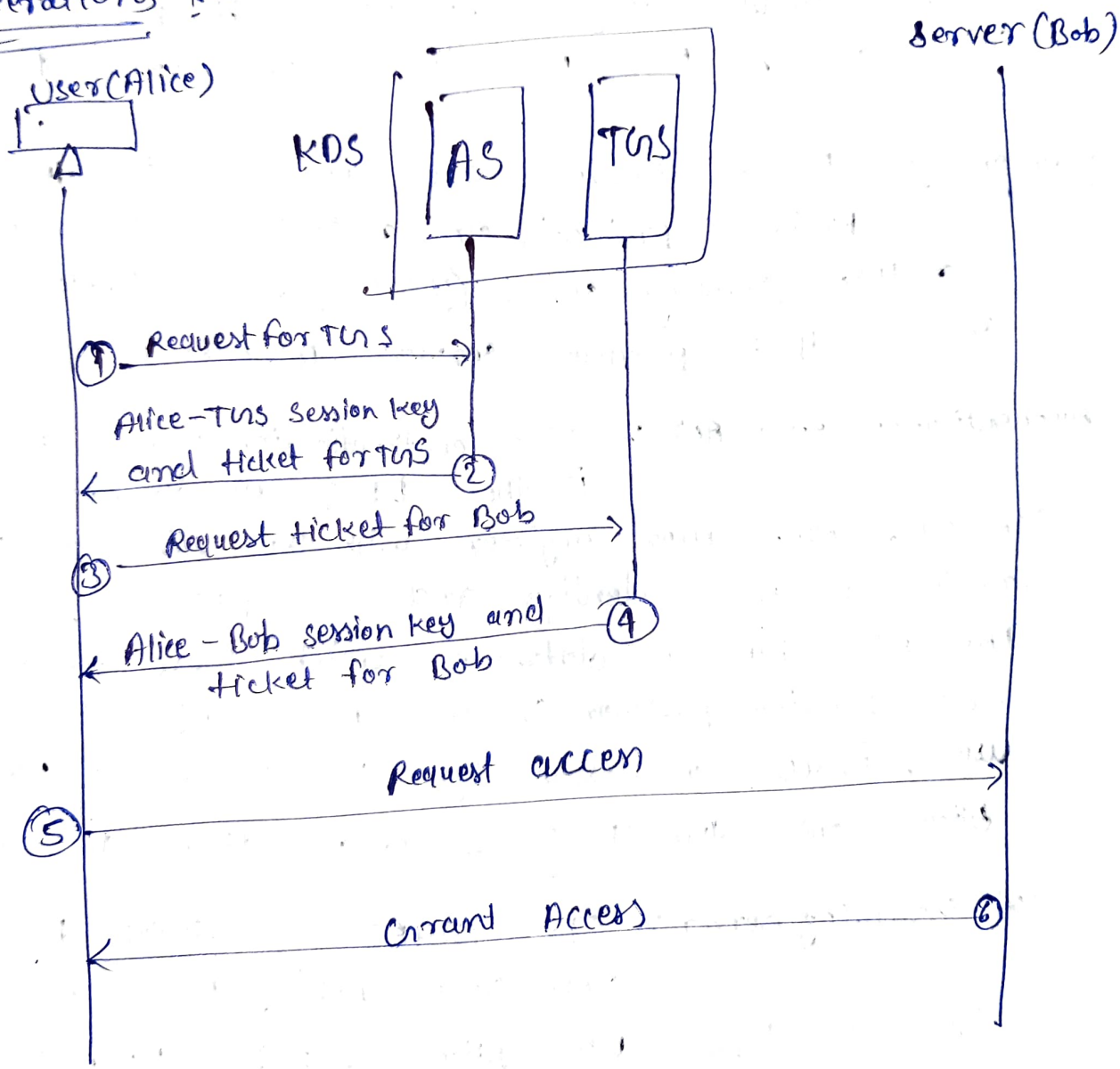
Servers :- Three servers are involved in the kerberos protocol, an authentication server (AS) a ticket-granting server (TGS), & Real (data) Server that provides services to others.

Authentication Server (AS) :- The authentication server (AS) is the KDC in kerberos Protocol. each user register with the AS and is granted a user identity and a password. The AS has a database with these identity & corresponding passwords. The AS verifies the user, issues a session key to be used b/w Alice and the TGS & sends a ticket for the TGS.

Ticket-Granting Server :- It issues a ticket for the real server (Bob). It also provide Session key (K_{AB}) b/w Alice & Bob. Keberos has separated user verification from the issuing of tickets. In this way, though Alice verifies her ID just once with the AS, she can contact TGS multiple times to obtain tickets for different real servers.

Real Server :- The real server (Bob) provides services for the user (Alice). Kerberos designed for a client-server program, such as a FTP, in which the user uses the client process to access the server process. Kerberos is not used for person to person authentication.

Operations :-



Diffie-Hellman key Agreement :- (without KDC)

In Diffie-Hellman protocol two parties create a Symmetric session key without the need of KDC. Before establishing a symmetric key, the two parties need to choose two numbers p & g.
p is large prime number of order 300 digits (1024 bits)
The second number g, is a generator of order p-1 in the group $\langle Z_p^*, \times \rangle$, these two group & generator no need to confinential. They can be sent through the internet. they can public.

The steps are as follows :-

(1) Alice chooses a large random number x such that $0 \leq x \leq p-1$ then calculate

$$R_1 = g^x \bmod p$$

(2) Bob choose another large number y such that $0 \leq y \leq p-1$ and calculates

$$R_2 = g^y \bmod p$$

(3) Alice send R_1 to Bob, NOTE the Alice does not send value x, sends only R_1

(4) Bob sends R_2 to Alice, Again Bob does not send value y.

(5) Alice calculates

$$K = (R_2)^x \bmod P$$

(6) Bob also calculates

$$K = (R_1)^y \bmod P$$

K is symmetric key of the session

Example :- $g = 7, \quad P = 23$

(1) Alice choose $x = 3$ and calculates

$$R_1 = 7^3 \bmod 23 = 21$$

(2) Bob chooses $y = 6$ and calculates

$$R_2 = 7^6 \bmod 23 = 4$$

(3) Alice sends the number 21 to Bob

(4) Bob sends the number 4 to Alice

(5) Alice calculates a symmetric key

$$K = 4^3 \bmod 23 = 18$$

(6) Bob calculates the symmetric key

$$K = 21^6 \bmod 23 = 18$$

It is vulnareble to

→ Discrete Logarithm Attacks

→ Man in the middle Attacks.