UNIT - IV

Part-A

1. Symmetric key distribution using symmetric encryption
2. Symmetric key distribution using Asymmetric Encryption
3. Distribution of public keys.
4. X.509 certificate.

Part-B

1. Remote user - Authentication principles
2. Remote user - Authentication using symmetric encryption
3. Kerberos
4. Remote user-Authentication using Asymmetric encryption.

PART-A

1) Symmetric key distribution using symmetric Encryption:
For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others.
The strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties who wish to exchange data without allowing others to see the key. For two parties A and B, key distribution can be achieved in a no. of ways, as follows:
1. A can select a key and physically deliver it to B
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used o

key, one party can transmit the new key to the other, encrypted using the old key.

4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

communication between end system is encrypted using a temporary key, often referred to as a session key.

session keys are transmitted in encrypted form, using a master key that is shared by the key distribution center and an end system or user.

A key distribution scenario:-

The scenario assumes that each user shares a unique master key with the key distribution center (KDC).

Let us assume that user A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection. A has a master key, K_a, known only to itself and the KDC; similarly, B shares the master key K_b with the KDC.

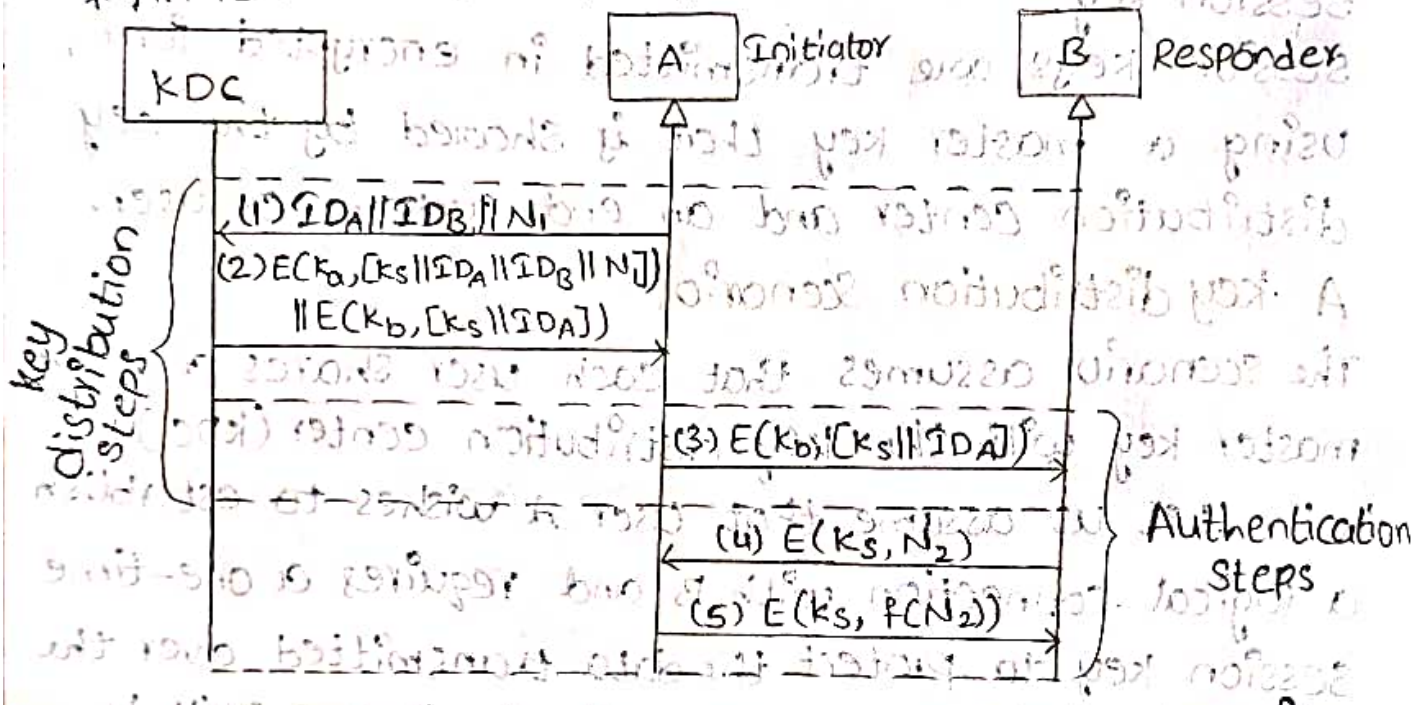
The following steps occur,

1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier, N_1, for this transaction, which we refer to as a nonce. The nonce may be a timestamp, a counter, or a random number.
2. The KDC responds with a message encrypted using K_a. Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC. The message includes two items intended for A:

* The one-time session key, K_s, to be used for the session.
* The original request message, including the nonce, to enable A to match this response with the appropri-ate request

In addition, the message includes two items intended for B:

* The one-time session key, K_s, to be used for the session
* An identifier of A, ID_A



3, A stores the session key for use in the upcoming session and forwards to B the information that originated at the KDC for B, namely, $E(K_b, [K_s || ID_A])$

4, Using the newly minted session key for encryption; B sends a nonce, N_2 to A.

5, Also using K_s, A responds with $f(N_2)$, where f is a function that performs some transformation on N_2

2, Symmetric key distribution using asymmetric encryption:

One of the most important uses of a public-key cryptosystem is to encrypt secret key for distribution.

Secret key Distribution with confidentiality & Authentication:

It provides protection against both active & passive attacks. We begin at a point when it is assumed that

A and B have exchanged public keys by one of the schemes described subsequently in this chapter. Then the following steps occur.

1, A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.

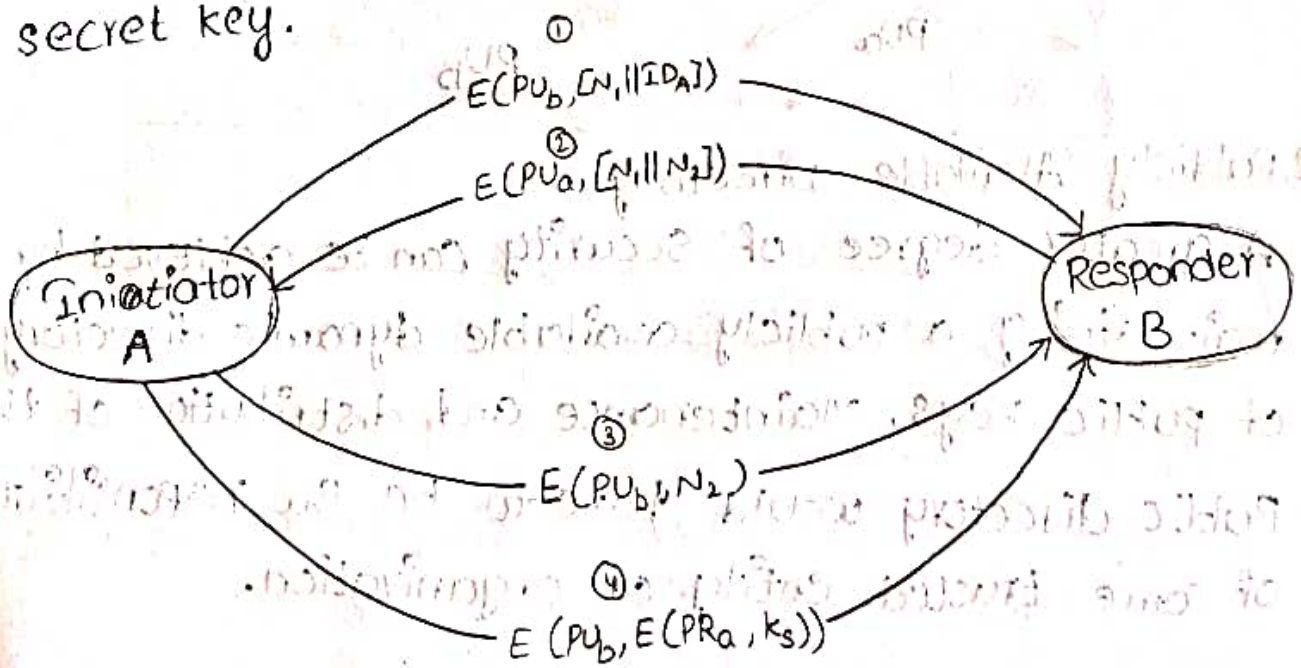
2, B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (1), the presence of N_1 in message (2) assures A that the correspondent is B.

3, A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

4, A selects a secret key, Ks and sends $M = E(PU_b, E(PR_a, Ks))$ to B. Encryption of this message with B's public key ensures that only B can read it. Encryption with A's private key ensures that only A could have sent it.

5, B computes $D(PU_a, D(PR_b, M))$ to recover the secret key

The result is that this scheme ensures both confidentiality and authentication in the exchange of a secret key.



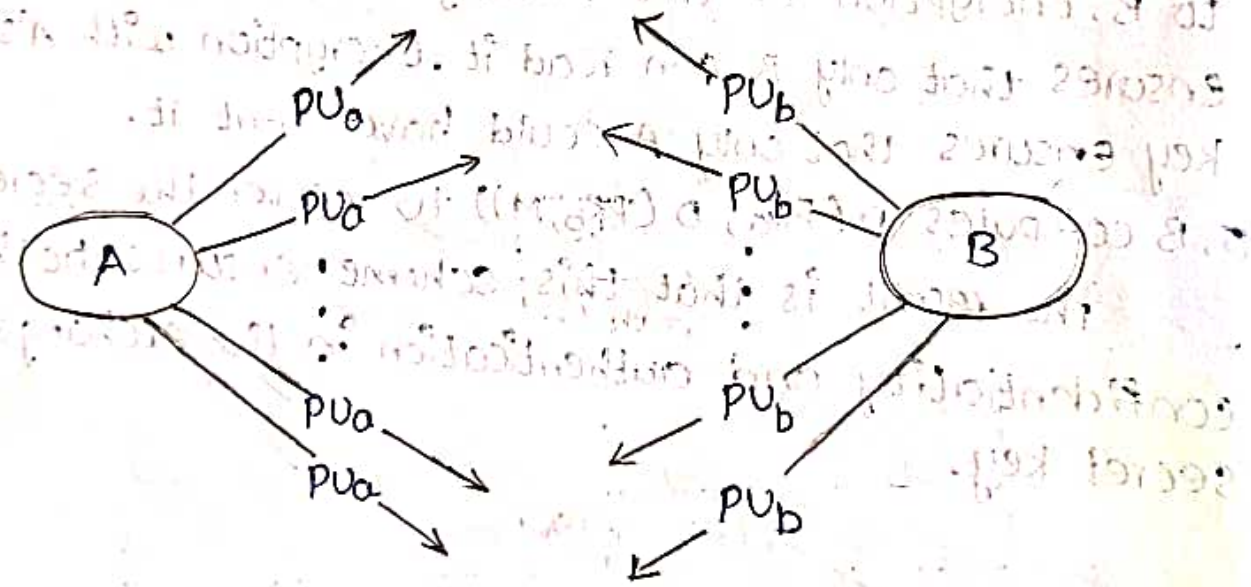
3, Distribution of Public keys:-

Several techniques have been proposed for the distribution of Public keys.

They are 1, Public ~~key~~ announcement
2, Publicly available directory
3, Public-key authority
4, Public-key certificates.

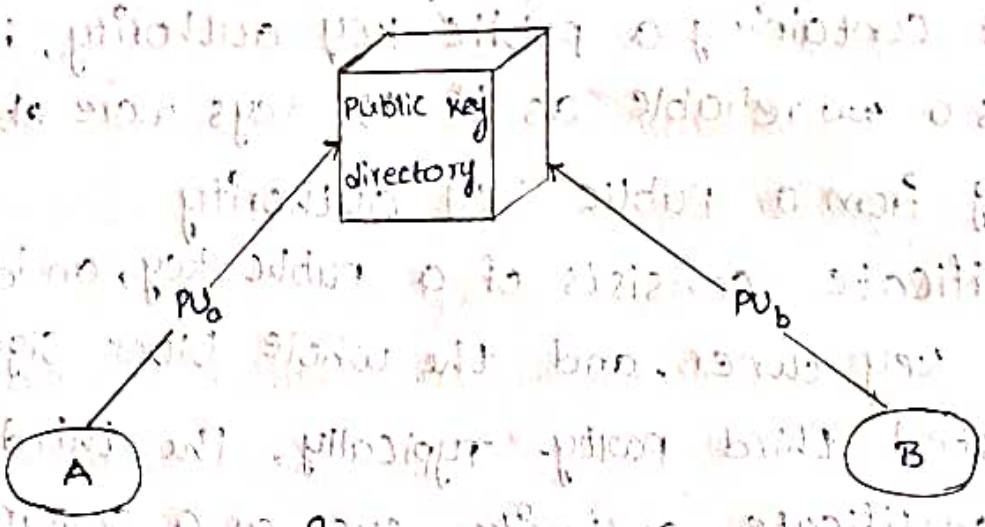
1, Public announcement of Public keys :-

The point of public key encryption is that the public key is public. Thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the key to the community at large.



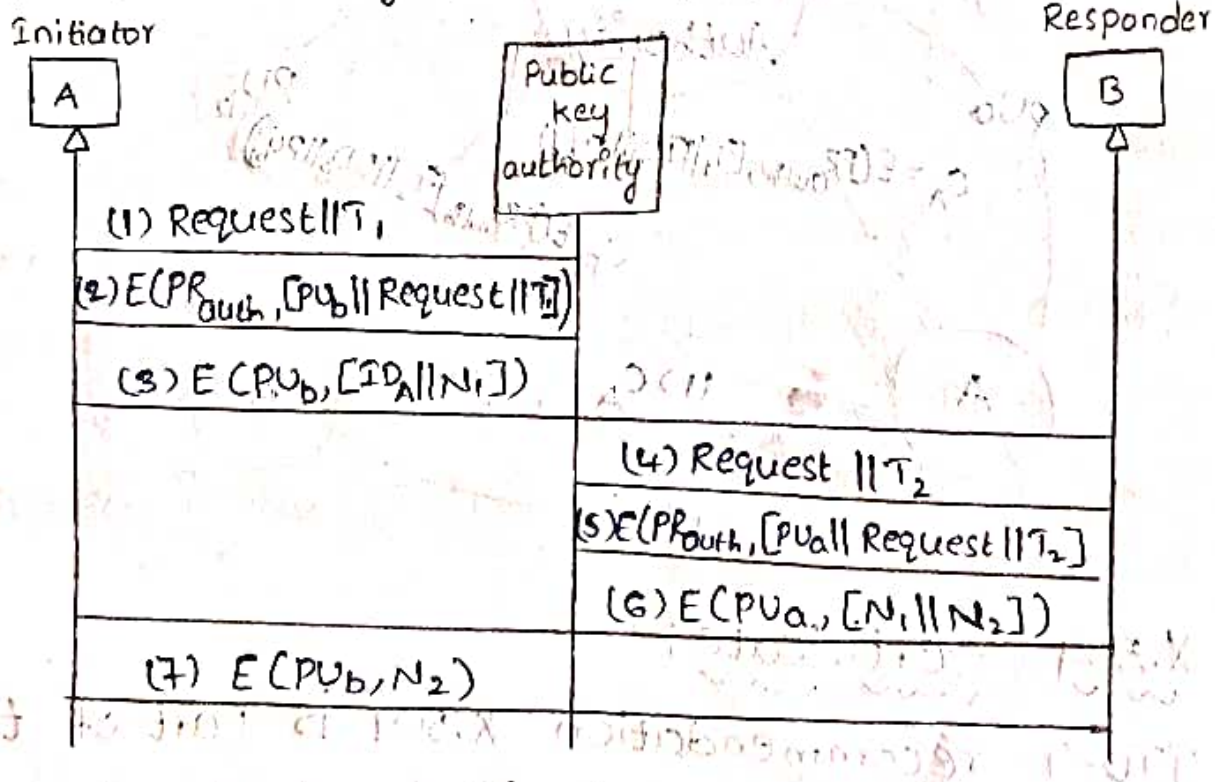
2, Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.



3. public-key Authority:-

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. The scenario assumes that a central authority maintains a dynamic directory of public keys of all participants. In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.



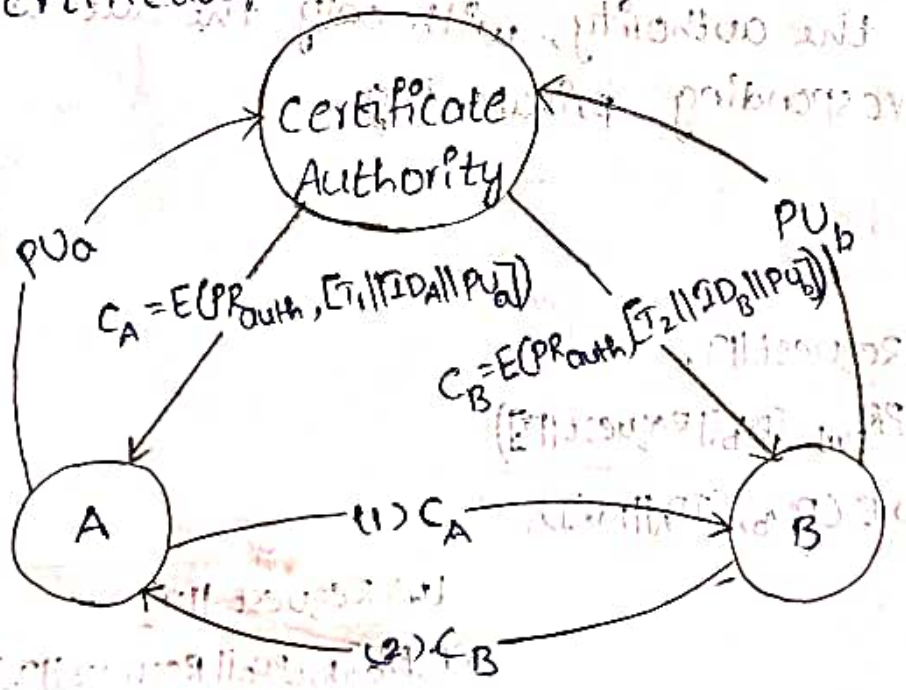
4. public-key Certificates:-

Use certificates that can be used by participants to exchange keys

Without containing a public-key authority, in a way that is a ~~as~~ reliable as if the keys were obtained directly from a public key authority

A certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party. Typically, the third party is a certificate authority, such as a government agency or a financial institution, that is trusted by the user community.

A user can present his or her public key to the authority in a secure manner and obtain a certificate. The user can then publish the certificate.



4. X.509 certificate :-

ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a directory service.

The directory is, in effect, a server or distributed set of servers that maintains a database of information about users. The information includes a mapping from user name to network address, as well as other attributes and information about the users.

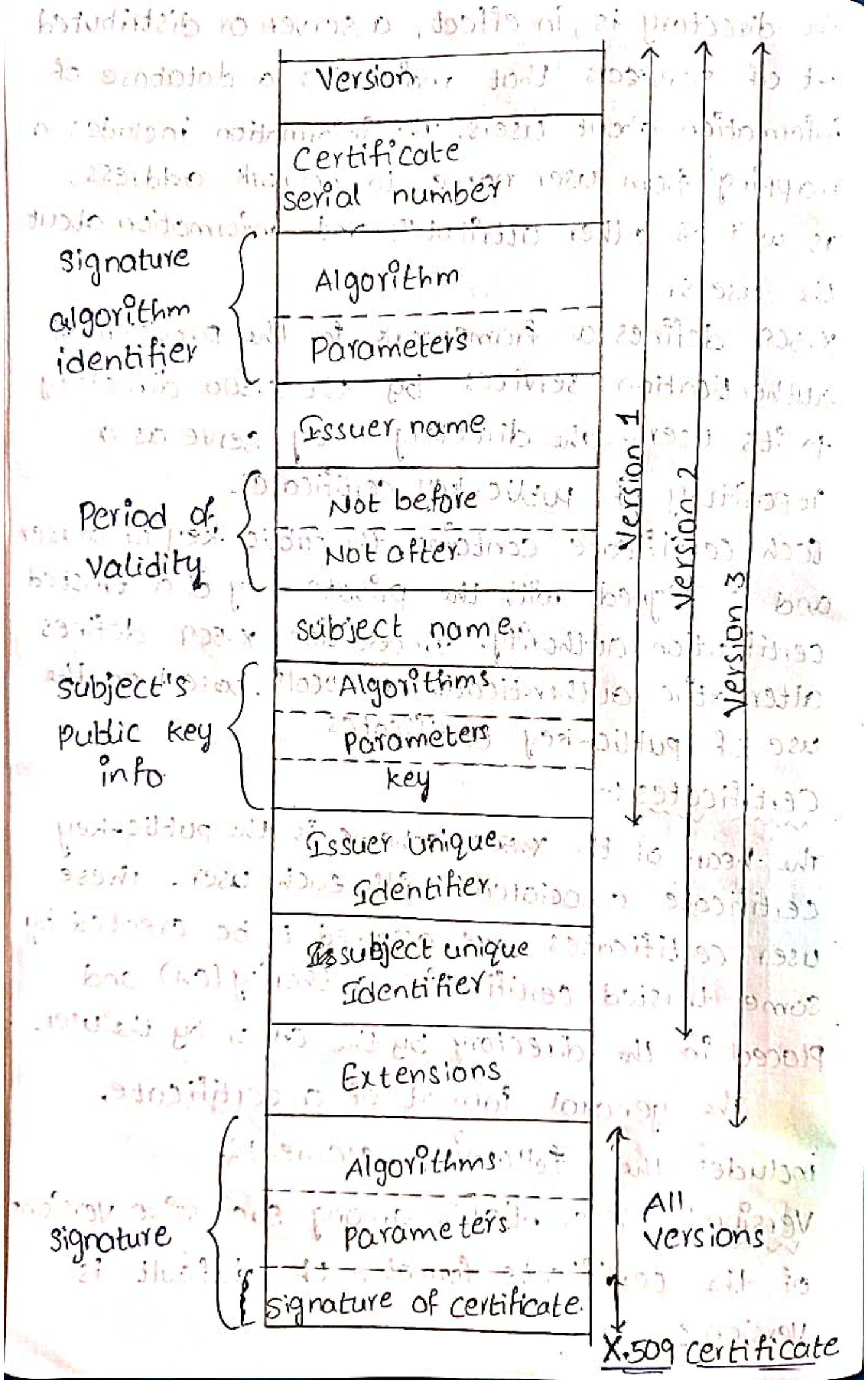
X.509 defines a framework for the provision of authentication services by the X.500 directory to its users. The directory may serve as a repository of public-key certificates. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority. In addition, X.509 defines alternative authentication protocols based on the use of public-key certificates.

Certificates :-

The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

The general format of a certificate, includes the following elements.

Version :- Differentiates among successive versions of the certificate format, the default is version 1.


Version
Certificate serial number
Signature algorithm identifier
Algorithm
Parameters
Issuer name
Period of validity
Not before
Not after
Subject name
Subject's public key info
Algorithms
Parameters
key
Issuer Unique Identifier
Subject unique Identifier
Extensions
Signature
Algorithms
parameters
signature of certificate
Version 1
Version 2
Version 3
All versions
X.509 certificate

serial number :- An integer value unique within the issuing CA that is unambiguously associated with this certificate.

Signature and algorithm identifier :- The algorithm used to sign the certificate together with any associated parameters.

Issuer name :- X.500 name of the CA that created and signed this certificate.

Period of Validity :- Consists of two dates; the first and last on which the certificate is valid.

subject name :- The name of the user to whom this certificate refers.

subject's public key information :- The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.

Issuer unique identifier :- An optional-bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities

subject unique identifier :- An optional-bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.

Extensions :- A set of one or more extension fields.

Signature :- Cover all of the other fields of the certificate.

Part - B

1) Remote user - Authentication principles :-

This process consists of two steps:

Identification step: presenting an identifier to the security system.

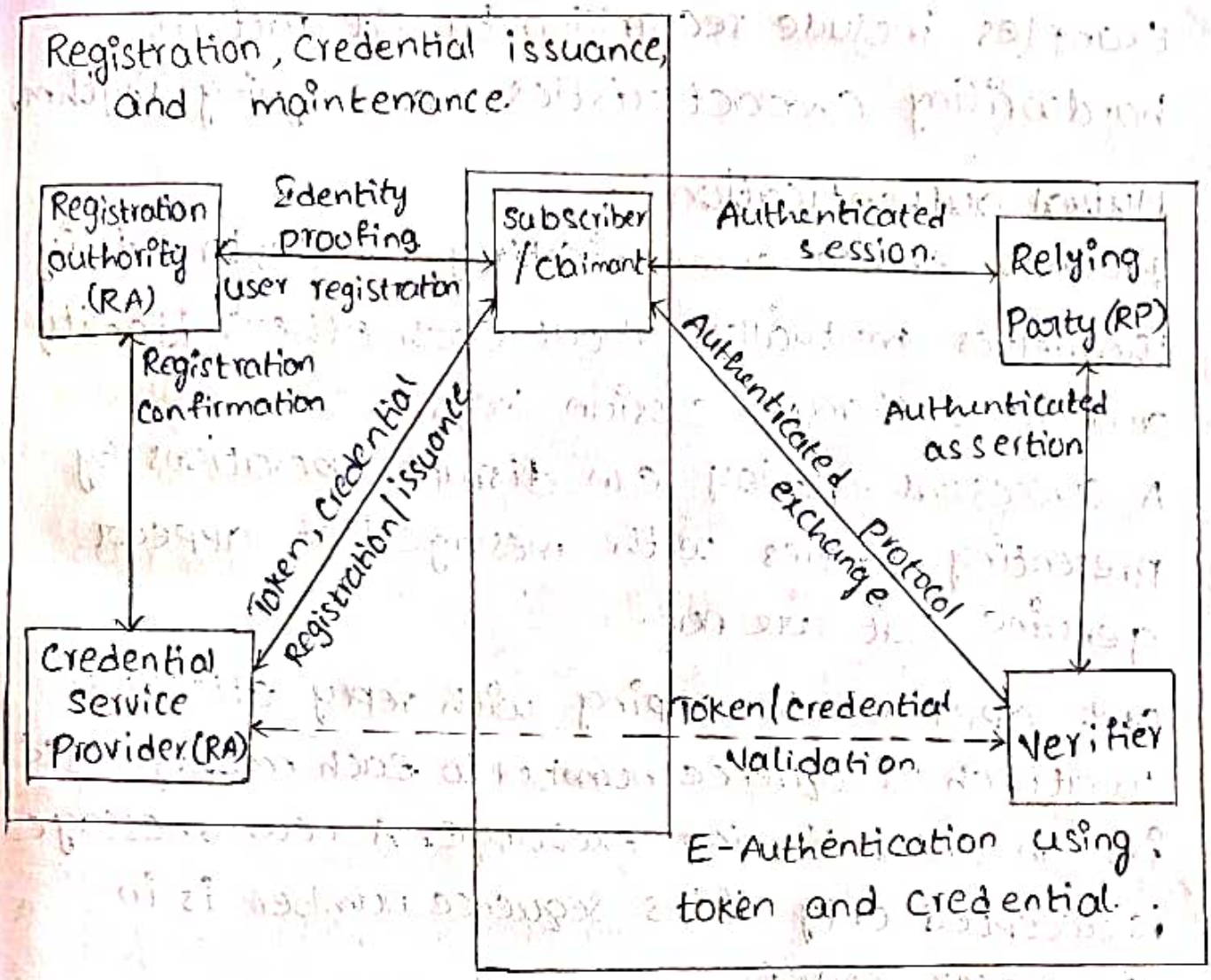
Verification step :- presenting or generating authentication information that corroborates the binding between the entity and the identifier.

The initial requirements for performing user authentication is that the user must be registered with the system. The following is a typical sequence for registration. An applicant applies to a registration authority (RA) to become a subscriber of a credential service provider (CSP). The token could be an encryption key or an encrypted password that identifies the subscriber.

The party to be authenticated is called a claimant and the party verifying that identity is called a verifier.

The verifier passes on an assertion about the identity of the subscriber to the relying party (RP).

Means of authentication:-



There are four general means of authenticating a user's identity:

Something the individual knows: Examples include a password, a personal identification number (PIN).

Something the individual possesses:- Examples include cryptographic keys, electronic keyboards, smart cards, and physical keys. This type of authenticator is referred to as a token.

Something the individual is (static biometrics):- Examples include recognition by fingerprint, retina, and face.

something the individual does (dynamic biometrics). Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

Mutual authentication :-
protocols enable communicating parties to satisfy themselves mutually about each others identity and to exchange session keys.
A successful replay can disrupt operations by presenting parties with message that appear genuine but are not.
one approach to coping with reply attacks is to attach a sequence number to each message used in an authentication exchange. A new message is accepted only if its sequence number is in the proper order.
sequence numbers are generally not used for authentication and key exchange. Instead, one of the following two general approaches is used.
Timestamps :- party A accepts a message as fresh only if the message contains a timestamp that, in A's judgement, is close enough to A's knowledge of current time.
This approach requires that clocks among the various participants be synchronized.
challenge/response: party A, expecting a fresh message from B, first sends B a nonce

(challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

2) Remote user-Authentication using symmetric encryption

Mutual authentication :-

A two level hierarchy of symmetric encryption keys can be used to provide confidentiality for communication in a distributed environment.

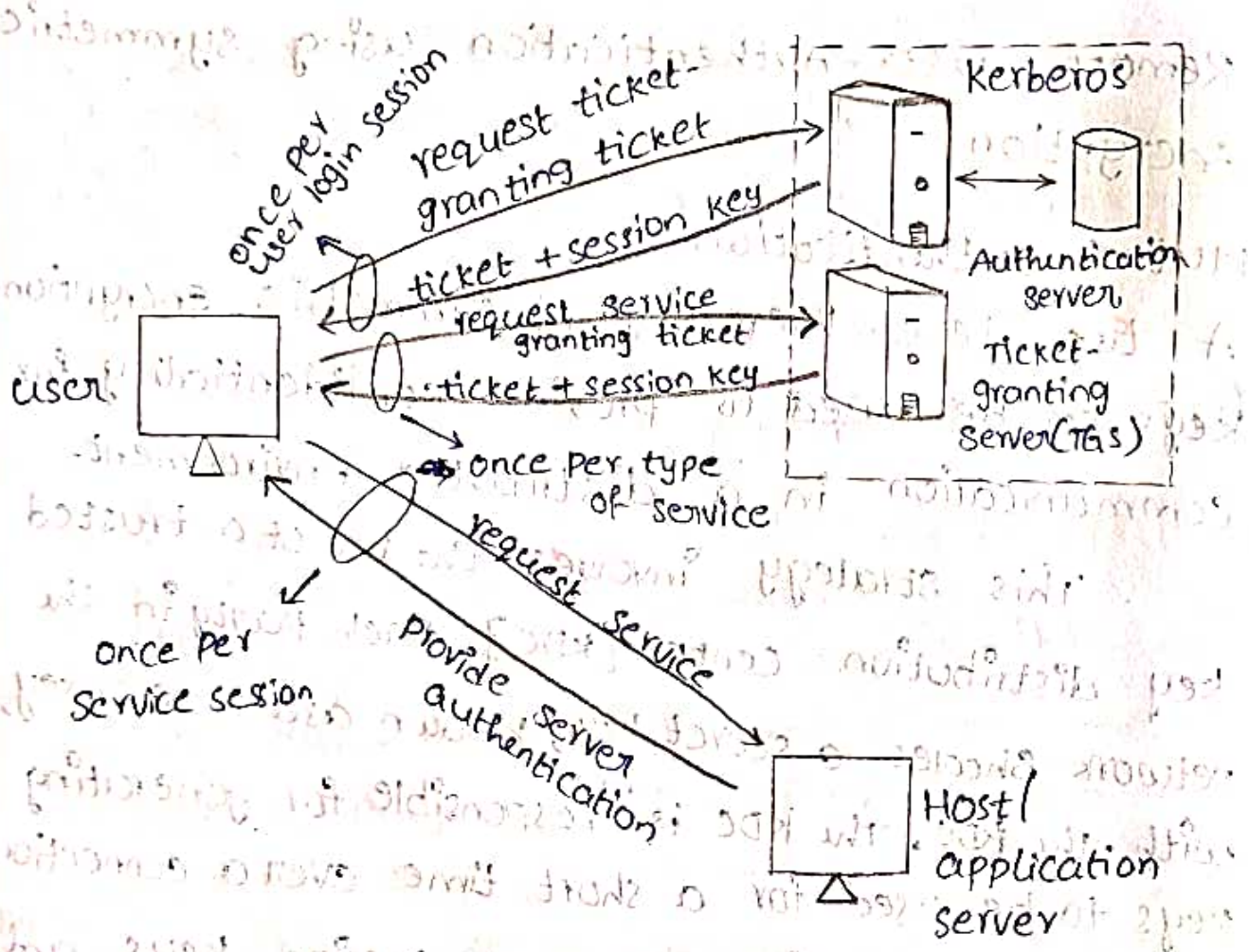
This strategy involves the use of a trusted key distribution center [KDC]. Each party in the network shares a secret key, known as a master key, with the KDC. The KDC is responsible for generating keys to be used for a short time over a connection between two parties, known as session keys, and for distributing those keys using the master keys to protect the distribution.

The protocol can be summarized as follows.

1. $A \rightarrow KDC$: $ID_A || ID_B || N_1$
2. $KDC \rightarrow A$: $E(K_a, [K_s || ID_B || N_1 || E(K_b, [K_s || ID_A])])$
3. $A \rightarrow B$: $E(K_b, [K_s || ID_A])$
4. $B \rightarrow A$: $E(K_s, N_2)$
5. $A \rightarrow B$: $E(K_s, f(N_2))$ where $f()$ is a generic function that modifies the value of the nonce.

3. Kerberos Version 4 :-

Version 4 of Kerberos makes use of DES.



1, User logs on to workstation and requests service on host

2, AS verifies user's access right in database and creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

3, workstation prompts user for password to decrypt incoming message, and then send ticket and authenticator that contains user's name, network address, and time to TGS.

4, TGS decrypts ticket and authentication, verifies

request, and then creates ticket for requested application server.

5. workstation sends ticket and authenticator to host.

6. Host verifies that ticket and authenticator match, and then grants access to service. If mutual authentication is required, server returns an authenticator.

C = client

AS = authentication server

V = server

ID_C = identifier of user on C

ID_V = identifier of V.

P_C = password of user on C

AD_C = network address of C

K_V = secret encryption key shared by AS and V.

once per user login session:-

(1) $C \rightarrow AS$: $ID_C || ID_{tgs}$

(2) $AS \rightarrow C$: $E(K_C, Ticket_{tgs})$

once per type of service:

(3) $C \rightarrow TGS$: $ID_C || ID_V || Ticket_{tgs}$

(4) $TGS \rightarrow C$: $Ticket_V$

once per service session:

(5) $C \rightarrow V$: $ID_C || Ticket_V$

$Ticket_{tgs} = E(K_{tgs}, [ID_C || AD_C || ID_{tgs} || TS_1 || Lifetime_1])$

$Ticket_V = E(K_V, [ID_C || AD_C || ID_V || TS_2 || Lifetime_2])$

4. Remote user-authentication using Asymmetric encryption.

1. $A \rightarrow KDC$: $ID_A || ID_B$
2. $KDC \rightarrow A$: $E(PR_{auth}, [ID_B || PU_b])$
3. $A \rightarrow B$: $E(PU_b, [N_a || ID_A])$
4. $B \rightarrow KDC$: $ID_A || ID_B || E(PR_{auth}, N_a)$
5. $KDC \rightarrow B$: $E(PR_{auth}, [ID_A || PU_a]) || E(PU_b, E(PR_{auth}, [N_a || k_s || ID_A || ID_B]))$
6. $B \rightarrow A$: $E(PU_a, [N_b || E(PR_{auth}, [N_a || k_s || ID_A || ID_B])])$
7. $A \rightarrow B$: $E(k_s, N_b)$

[elaborate the points]