3. Digital Signature & Authentication Application

Digital Signature

why we use digital signature
- suppose in banking transaction we sign on the check to redraw your account.
- signature prove that it is coming from correct person.
- If signature not done on check means anyone can withdraw the money from account balance. so that reason we need to do unique sign.

- It is very important role in e-commerce & online transaction etc.
- It is based on asymmetric key cryptography. for encryption we use private key & for decryption we use public key.
- Digital signature is used for message authentication & non repudiation of msg integrity
- It is not used for confidentiality (Not change at trans)

A → B

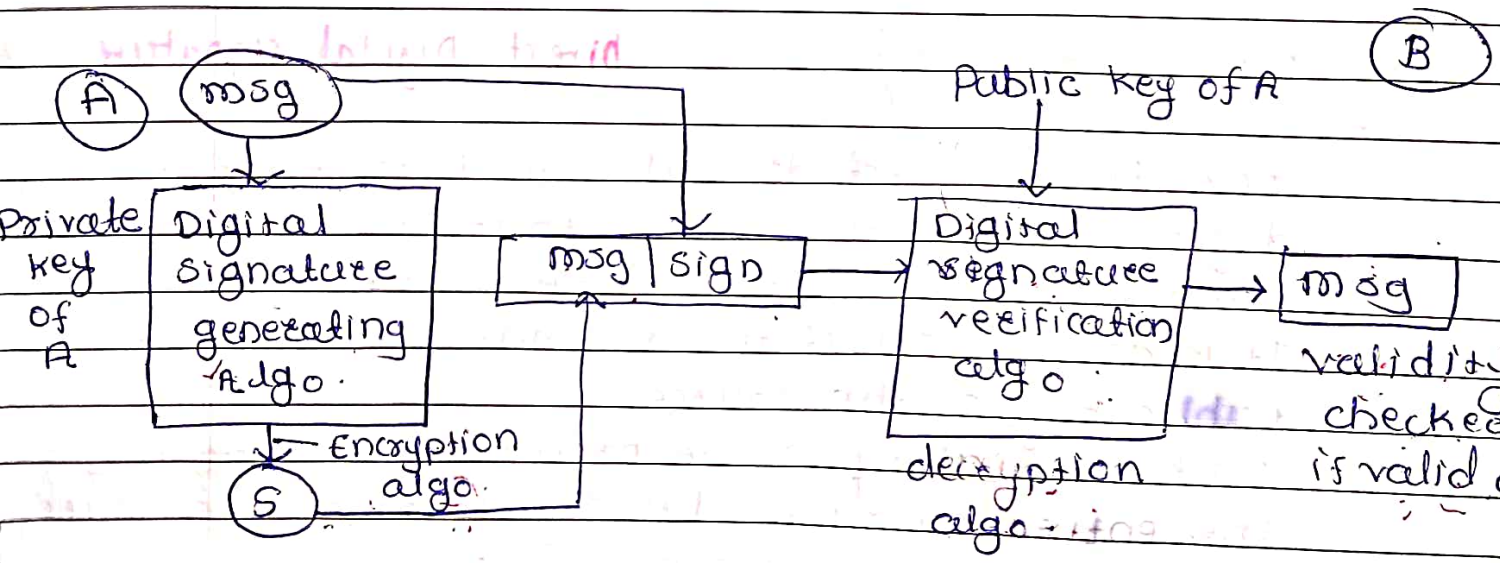


Fig - Simple Digital signature

- In this fig. A send message by using private key of A & generating the digital signature algorithm & encrypt that message by using encryption algo. & adding sign into that msg. then that msg + sign send to the receiver side.
- Receiver can decrypt by using public key of A & verify the mes signature by using algorithm if match then this is correct message from A.

- Rules →

1) The signature must be a bit patteen that depends on the message being signed.
2) The signature must use some information unique to the sender to prevent both forgery & denial.
3) It must be relatively easy to produce the digital signature.
4) It is easy to recognize & verify the digital signature.
5) It must be practical to retain a copy of the digital signature in storage.

Direct Digital signature. Direct Digital signature

- the term direct digital signature refers to a digital signature scheme that involves only the communicating parties. like source & destinatio
- It is assumed that the destination knows the public key of the source.
- confidentiality can be provided by encrypting the entire message plus signature with a share secret key (symmetric encryption)

Digital signature standard

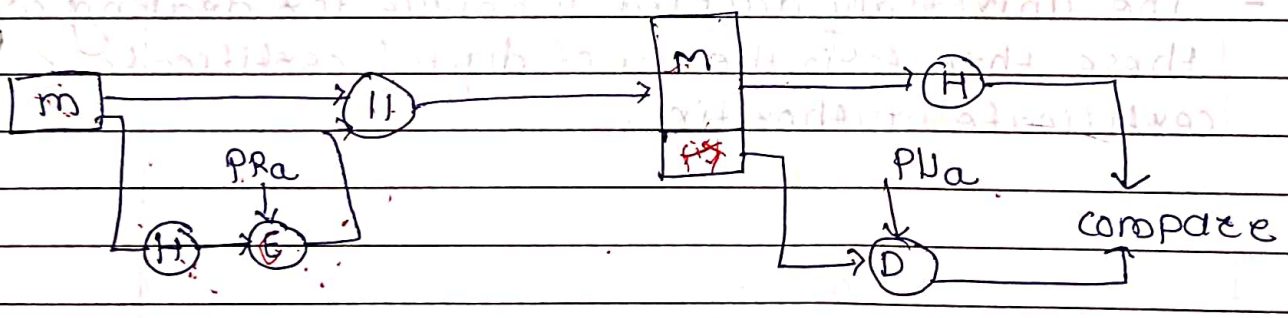
Digital signature standard

- The National Institute of standard & Technology (NIST) has published Federal information processing standard FIPS 186 known as the Digital signature standard (DSS)
- The DSS makes use of the SHA & presents a new digital signature signature technique, the Digital signature Algorithm (DSA)
- The latest version also incorporates digital signature algorithms based on RSA & an elliptic curve cryptography.

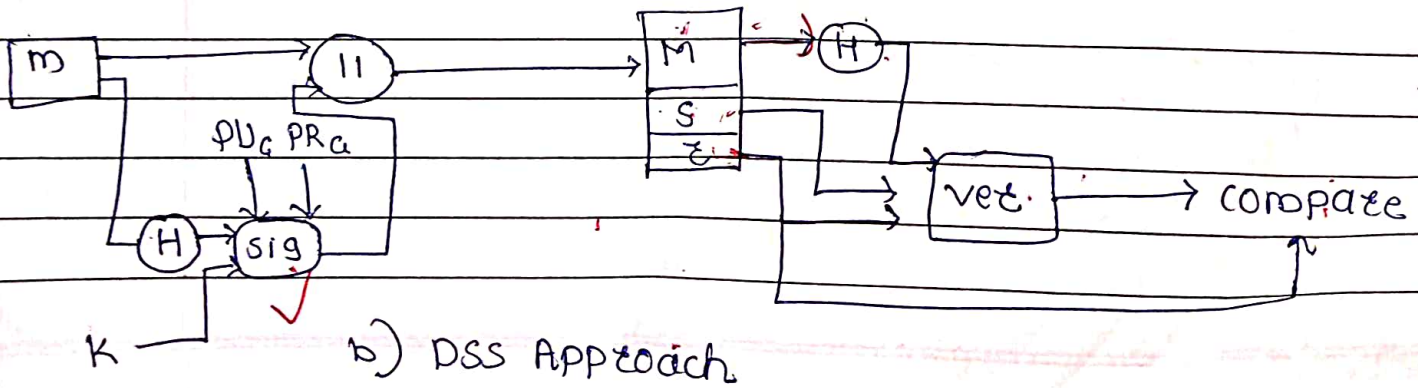
The DSS Approach

The DSS Approach

- The DSS uses an algorithm that is designed to provide only the digital signature fun.
- Unlike RSA, it can't be used for encryption or key exchange, Nevertheless, it is public key technique.



a) RSA Approach.



b) DSS Approach

- It is important to perform the signature func. first & then an outer confidentiality fun.
- In case of dispute, some 3rd party must view the message & it's signature.
- If the signature is calculated on an encrypted msg., then the 3rd party also needs access to the decryption key to read the original msg.
- However if the signature is the inner operation then the recipient can store the plaintext msg & it's signature for later use in dispute resolution
- The validity of the scheme just described depends on the security of the sender's private key.
- If a sender, later wishes to deny sending a particular msg, the sender can claim that the private key was lost or stolen & that someone else forged his or her signature.
- Another threat is that some private key might actually be stolen from X at time T
- The opponent can then send a msg. with X's signature & stamped with a time before or equal to T.
- The universally accepted technique for dealing with these threats is the use of digital certificate & certificate authorities.

signature is valid.
- The signature fun. is such that only the sender, with knowledge of the private key, could have produced the valid signature.

The Digital signature Algorithm

The DSA is based on the difficulty of computing discrete logarithms & based on schemes originally presented by ElGamal & Schnorr.

① Global Public key component

p Prime no. where $2^{L-1} < p < 2^L$
for $512 < L < 1024$ & L multiple of 64;
i.e bit length of betn 512 & 1024 bits
in increments of 64 bits.

q Prime divisor of (P-1), where $2^{159} < q < 2^{160}$
i.e bit length of 160 bits.

$g = h^{(P-1)/q} \bmod p$.
where h is any integer with $1 < h < (P-1)$.
such that $h^{(P-1)/q} \bmod P > 1$.

② User's Private key
x random or pseudorandom integer with $0 < x < q$

③ User's Public key.
$y = g^x \bmod p$.

RSA Approach

- In the RSA approach, the msg. to be signed input to a hash function that produces a secure ha code to fixed length.
- This hash code is then encrypted using the sende private key to form the signature.
- Both the msg & the signature then transmitted.
- The recipient takes the msg & produces hash code.
- The recipients also decrypts the signature using the sender's public key.
- If the calculated hash code matches the decrypted signature, the signature is accepted as valid.

DSS Approach

- The DSS approach also makes use of hash function.
- The hash code is provided as input to a signature fun- along with a random no. k, generate for this particular signature.
- The signature function also depends on the sender's private key (PR_a) & a set of parameters known to a group of communicating principals.
- We can consider this set to constitute a global public key (PU_G)
- The result is signature consisting of two compone labeled s & r.
- At the receiving end, the hash code of the incoming msg. is generated.
- This plus the signature is i/p to a verification fun.
- The verification fun. also depends on the global publi key as well as the sender's public key (PU_a) which is paired with the senders private key.
- The output of the verification fun. is a value tha is equal to the signature component r, if the

User's private key

- With these no. in hand, each user selects a private key & generates a public key.
- The private key x must be a no. from 1 to (q-1) & should be chosen randomly.

Users public key

- The public key is calculated from the private key as shown in figure.
- It should be computationally infeasible to determine x from y.

User's per message secret No.

- Random no. K is integer with $0 < K < q$

Creating signature

To create a signature a user calculates two quantities r & s that are fun. of the public key components (p, q, g) the users private key x, the hash code of msg H(m) & an additional integer K that should be generated randomly & be unique for each signing.

Verification

- At the receiving end, verification is performed using the formulas shown in fig.
- The receiver generates a quantity that is a function of the public key components, the sender's public key & the hash code of the incoming msg.
- If this quantity matches the component of the signature, then signature is validated.
- figure below depicts fun. of signing & verifying.

④ User's Per-Message Secret No.

K = random or pseudorandom integer with $0 < K < q$

⑤ signing

$r = (g^k \bmod p) \bmod q$

$s = [K^{-1}(H(M) + xr)] \bmod q$

signature = (r, s)

⑥ verifying

$w = (s')^{-1} \bmod q$

$u_1 = [H(m')w] \bmod q$

$u_2 = (r')w \bmod q$

$v = [g^{u_1} y^{u_2} \bmod p] \bmod q$

TEST : $v = r'$

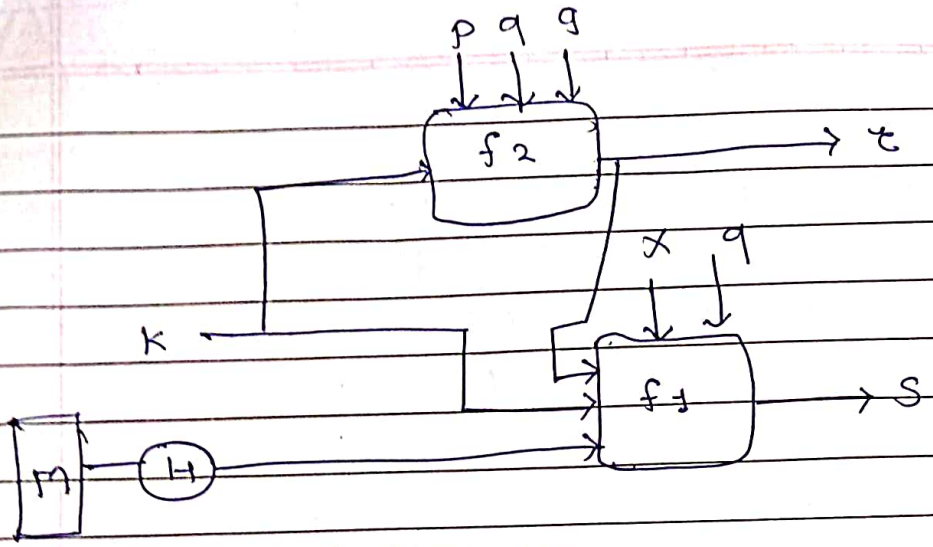
M = msg. to be signed

H(M) = hash of M using SHA-1

m', r', s' = received versions of M, r, s

④ Global public key components:

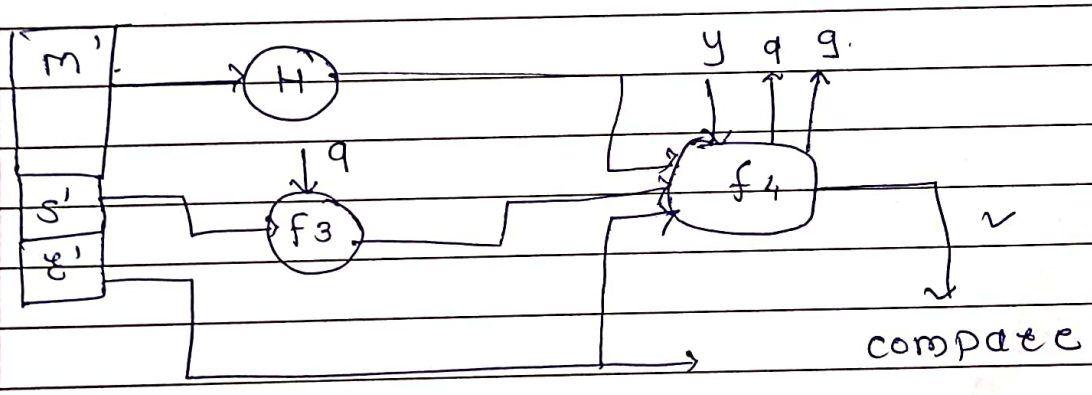
- There are three parameters that are public & can be common to a group of users.
- A 160 bit prime no. q is chosen
- Next, prime no. P is selected with a length bet^n 512 & 1024 bits such that q divides (P-1)
- Finally g is chosen to be of the form $h^{(P-1)/q} \bmod p$ where h is an integer bet^n 1 & (P-1) with the restriction that g must be greater than 1



a) signing

$s = f_1(H(m), k, x, r, q) = (k^{-1}(H(m) + xr)) \bmod q$

$r = f_2(K, P, q, g) = (g^k \bmod P) \bmod q.$



b) verifying

$w = f_3(s', q) = (s)^{-1} \bmod q$

$v = f_4(y, q, g, H(m'), w, r')$

$=$

Page No.:
Date: 25/11/24

Kerberos

i) Kerberos is computer network authentication protocol that works on the basis of tickets to allow nodes communicating over nonsecure network to prove their identity to one another in secure manner

ii) In kerberos authentication server & database is used for client authentication.

iii) Kerberos runs as a third party trusted server known as the key distribution center (KDC)

1) Authentication server (AS)
A server that issues tickets for desired service which are in turn given to users for access to the service

2) client → An entity on the network that can receive a ticket from kerberos.

3) Credential cache or ticket file →
A file which contains the keys for encrypting communications between a user & various network services.

4) credential
A temporary set of electronic credentials that verify the identity of client for particular service it is ca. also called a ticket.

5) crypt hash - one way hash used to authenticate user

6) key → Data used when encrypting or decrypting data

7) Key distribution center (KDC)
The service that issue kerberos tickets & which usually run on the same host as the ticket granting server (TGS)

8) Realm ⇒ A network that uses kerberos composed of one or more servers called KDC's & potentially large no. of clients.

Date: 26

9) Ticket granting server (TGS)
A server that issues tickets for a desired ser which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.

10) Ticket granting ticket (TGT)
A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

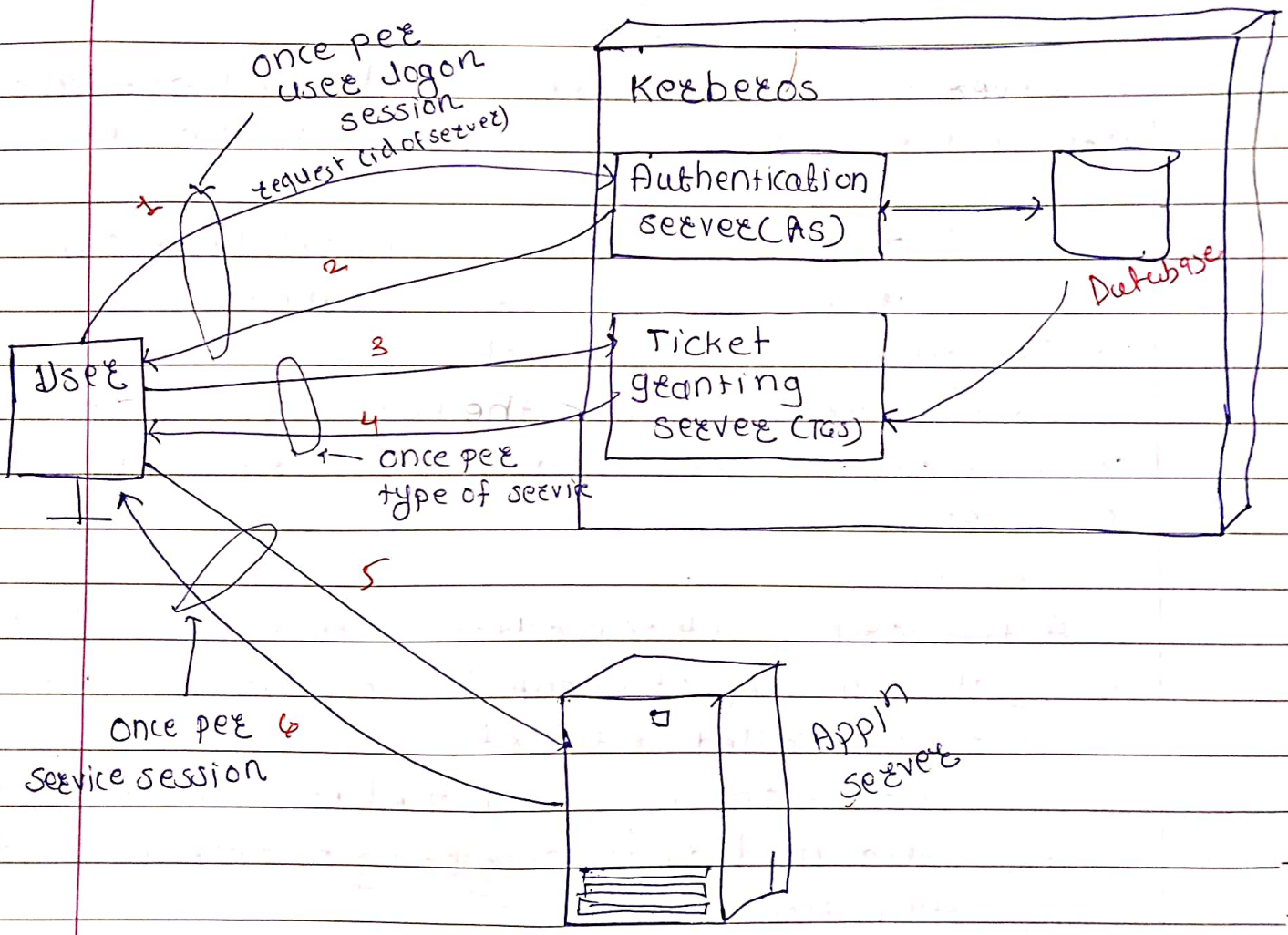


fig → Overview of Kerberos

1 Req → Id, Passw
2 Resp – Ticket & session
3 TGS-Req – perfect time, timestamp + Appln server details attach
Req ←

Page No.:
Date: 26/11/21
YOUVA

1) User logs on to workstation & requests service on hosts. (user id - passwd)
2) As verifies user's access right on database creates tickets granting ticket & session key. Results are encrypted using key derived from users password.
3) Workstation prompts user for password & uses password to decrypt incoming message, then sends ticket & authenticator that contain users name, network address & time to TGS.
4) TGS decrypts ticket & authenticator, verifies requ then create ticket for requested server.
5) Workstation sends ticket & authenticator to serv
6) server verifies that ticket & authenticator matc then grants access to service. If mutual authent tion is required, server return an authenticator

* Kerberos authentication Dialogue
i) The following msg exchanges take place for authenti cation through kerberos, authentication service excha to obtain ticket granting ticket (TGT)
ii) This exchange takes place only once per user logon session
iii) User obtain a ticket granting ticket from the Authentication server. This ticket is sent to the ticket Granting server to obtain service tickets.

C → ~~AS~~ : IDc || IDtgs || TS1

~~AS~~ → C : E(Kc, [Kc,tgs || IDtgs || TS2 || Lifetime2 || Ticket tgs])

Ticket tgs = E(Ktgs, [Kc,tgs || IDC || ADC || IDtgs || TS2 || Lifetime2])

Ticket granting service exchange to obtain service granting Ticket

iv) This exchange takes place for each type of servic
v) Here, the user presents the TGT to the ticket granting server, the TGS return a service granting ticket to the user after proper authentication.
vi) An authentication is added in the msg - which is encrypted using the key shared by the user & TGS

C → TGS : IDv || Ticket tgs || Authenticator c

TGS → C : E(Kc, tgs, [Kc,v || IDV || TS4 || Lifetime4 || Ticket v])

Ticket tv = E(Kv, [Kc, v || IDC || ADC || IDV || TS4 || Lifetime4])

Authenticator c = E(Kc, tgs, [IDC || ADC || TS3])

Client server authentication exchange to obtain service.

vii) The user sends service granting Ticket to the appln server.

viii) The msg also contains authenticator which proves the senders identity to the server. Moreover the server replies with the timestamp present in the authenticator. This authenticates the server to the user

C → V : Ticket v || Authenticator C

V → C : E(Kc,v, TSS+1).

Authenticator C = E(Kc, v [IDC || ADC || TS5])

Kerberos Realm

i) A Kerberos Realm is a set of managed nodes that share the same kerberos database.
ii) The kerberos database resides on the kerberos master computer system, which should be kept in a physically secure room.
iii) A read only copy of the kerberos database might also reside on other kerberos computer systems.
iv) However, all changes to the DB must be made on the master computer system using kerberos master passwo
v) A kerberos principal is a service or user that is known to the kerberos system. Each kerberos principal is identified by its principal name.
vi) Networks of clients & servers under different administrative organizations constitute different realms.
vii) For inter realm communication, the kerberos severs in the two realm must be authenticated & registered to each other.

A user wishing service on a server in another realm obtains a ticket for that server as given below

1) C → AS : IDc || IDtgs || TS1
2) AS → C : E[Kc, [Kc, tgs || IDtgs || TS2 || Lifetime2 || Tickettgs]
3) C → TGS : IDtgsrem || Tickettgs || Authenticator c
4) TGS → C : E(Kc, tgs, [KC, tgsrem || IDtgsrem || TS4 || Tickettgsrem])
5) C → TGSrem : IDvrem || Tickettgsrem || Authenticator c
6) TGSrem → C : E(Kc, tgsrem, [KC, vrem || IDvrem || TS6 || Ticketvrem
7) C → Vvrem : Ticketvrem || Authenticator c

where IDtgsrem → is the identity of remote TGS
Tickettgsrem → is the TGT for remote TGS,
IDvrem → is the identity of remote server
Ticketvrem → is the service granting ticket for remote server

	Kerberos version 4	Kerberos version 5
Encryption system dependence	It requires the use of DES	It includes ciphertext tag an encryption type identifier so that any encryption tech may be used
Internet protocol dependence	It requires use of IP address	It allows any network address type to be used
message byte ordering	Sender of msg employs a byte ordering of it's own choosing	All msg structures are defined using abstract syntax Notation one & basic encoding Rules which provide an unambiguous byte ordering.
Ticket lifetime	Lifetime value in version 4 are encoded in 8-bit, each unit of 5 minutes, thus max. lifetime that can be expressed is $28 \times 5 = 1280$ min or 21 hours.	It include explicit start time & end time allowing tickets with arbitrary lifetimes
Authentication forwarding	It does not allow credentials issued to one client to be forwarded to some other host & used by some other client. e.g client issues a request to a print server that then can't access the clients file from file server using clients credentials for access	Version 5 provides this capability.

Inter realm authentication

Interoperability among realm requires many Kerberos to Kerberos relationships but

It supports method that requires fewer relationships.

Requirements of Kerberos.

1) **Secure**

Kerberos should be strong enough that potential opponent does not find it to be the weak link.

2) **Reliable**

For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of supported services, with one system able to back up another.

3) **Transparent**

Ideally the user should not be aware that authentication is taking place beyond the requirement to enter password.

4) **Scalable.**

The system should be capable of supporting large no. of clients & servers.

X.509 Authentication service

i) X.509 uses the public key cryptography & digital signature
ii) It does not dictate the use of specific algo. but recommends RSA.
iii) The DS scheme assumed to require the use of hash function.

Certificate.

i) The key of X.509 scheme is the public key certificate associated with each user.
ii) These user certificates are assumed to be created by some trusted certification authority (CA) & placed in directory by CA or by user.
iii) The directory server itself is not responsible for creation of public keys or for the certification function.

1) version :-

i) Differentiates among successive versions of the certificate format, the default is version 1.
ii) If the Issuer unique Identifier or subject unique identifier are present the value must be version 2.
iii) If one or more extensions are present, the versi must be version 3

2) Serial Number.

An integer value, unique within the issuing CA, that is unambiguously associated with this certificat

3) Signature algorithm identifier
The algorithm used to sign the certificate, together with any associated parameters. because this information is repeated in the signature field at the end of the certificate, this field has little if any utility.

4) Issuer Name.
- X.509 name of the CA that created & signed this certificate.

5) Period of validity
Consists of two dates, the 1st & last on which the certificate is valid.

6) Subject Name
The name of the user to whom this certificate refers. i.e this certificate certifies the public key of the corresponding private key.

7) subjects 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.

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

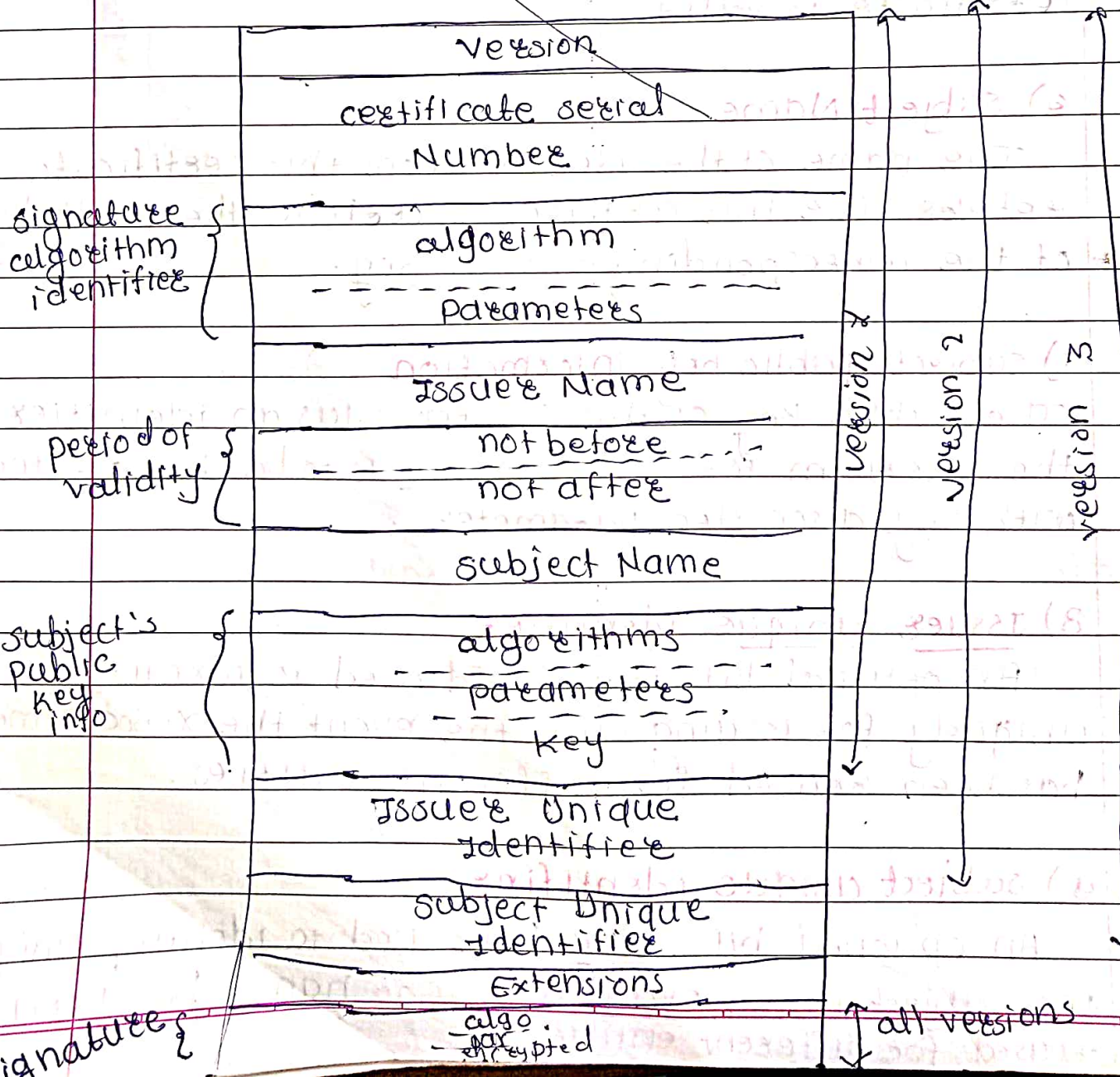
9) subject unique identifier.
An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.

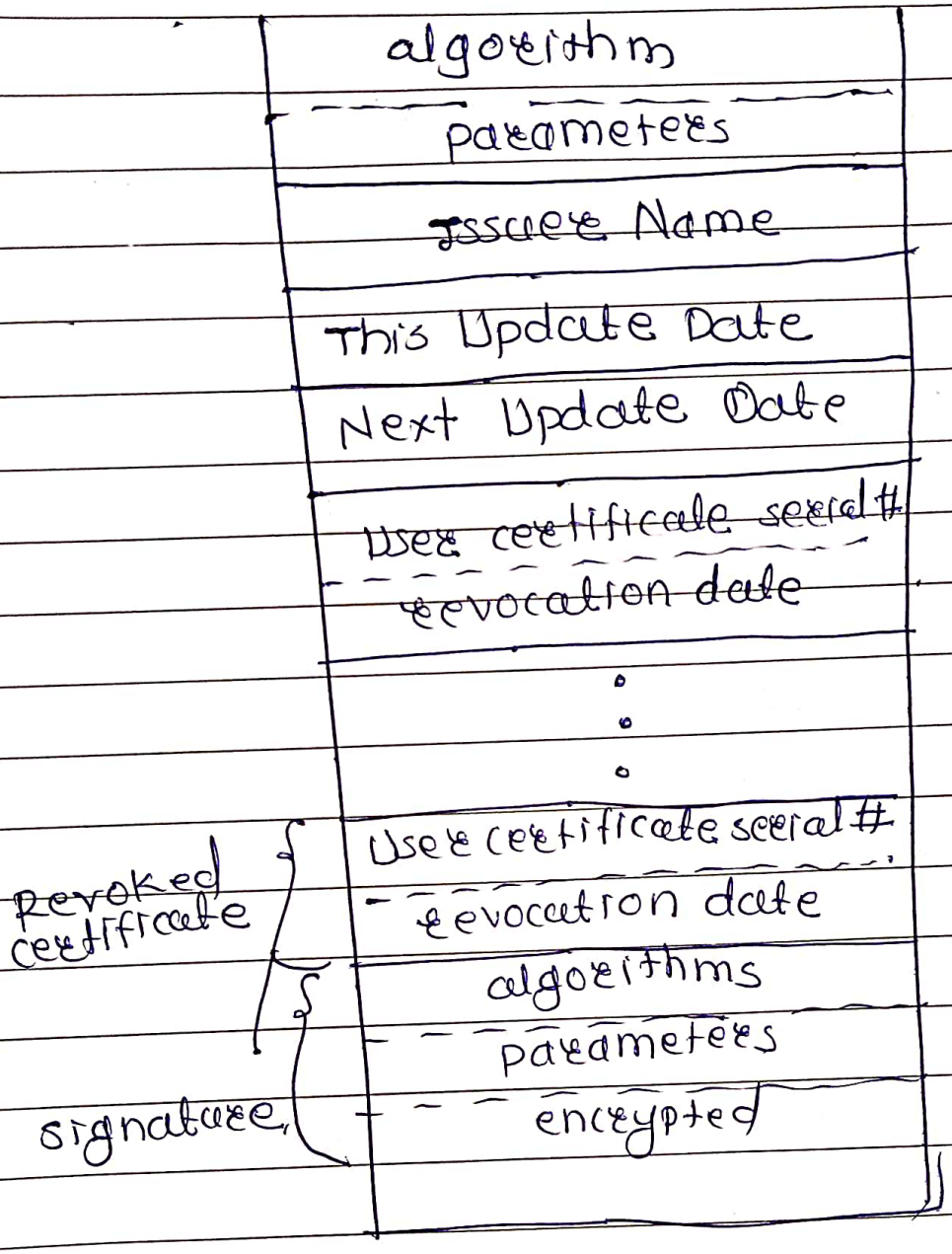
10) Extensions

A set of one or more extension fields. Extensions were added in version 3 & are

11) signature

- covers all the other fields of the certificate, it contains the hash code of the other fields, encrypted with the CA's private key.
- This field includes signature algo. identifier.
- The unique identifier fields were added in version 2 to handle the possible reuse of subject & issues name over time.





b) certificate Revocation List.