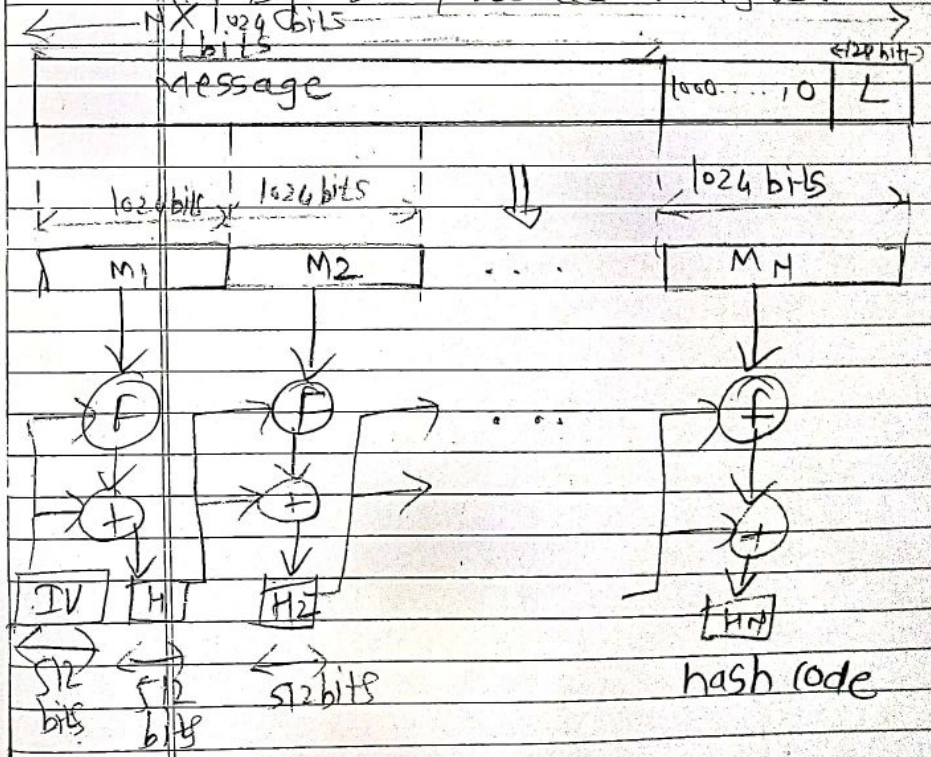
Sr. No	TITLE	Date	Teacher's Sign / Remark

Page No.
Date

Q.4:- a) - Explain the message digest generation in SHA-512.

1) The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as o/p a 512-bit msg digest. The i/p is processed in 1024 bit blocks. figure depicts the overall processing of a msg to produce a digest.



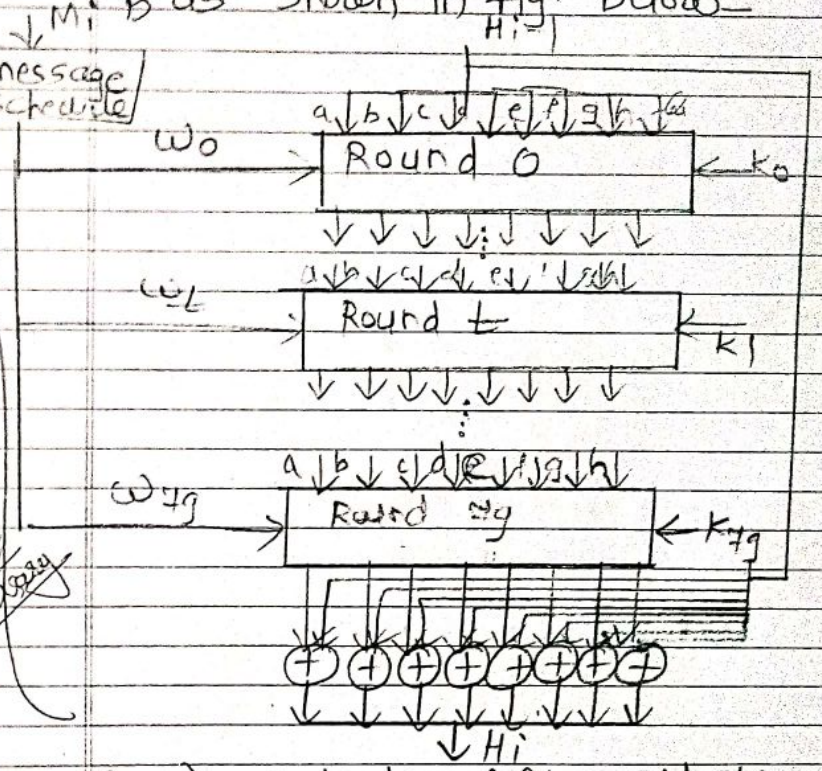
Scanned by CamScanner

Step i) – Append padding bits – The msg is padded so that its length is congruent to 896 modulo 1024 [length = 896 (mod 1024)] Padding is always added, even if the msg is already of the desired length. Thus, the no. of padding bits is in the range of 1 to 1024. The padding consists of a single 1 bit followed by the necessary number of 0 bits.

Step ii) – Append length – A block of 128 bits is appended to the msg. This block is treated as an unsigned 128-bit integer and contains the length of the original msg.

Step iii) Initialize hash buffer :– A 512 bit buffer is used to hold intermediate & final results of the hash fun. The buffer can be represented as eight 64 bit registers (a, b, c, d, e, f, g, h). These 64 bits are nothing but words where obtained by taking the first 64-bits of the fractional parts of the square roots of the first eight prime no.s

Step iv) – The heart of the algorithm is a module that consists of 80 rounds this module is as shown in fig. below –



Step v) – output :– After all N 1024 bit blocks have been processed, the o/p from the Nth stage is the 512 bit message digest.

We can summarize the behavior of SHA-512 as follows—

$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$

$MD = H_N$

Where—

IV = initial value of the abcdefgh buffer defined in step 3.

$abcdefgh_i$ = the o/p of the last round of processing of the ith msg block

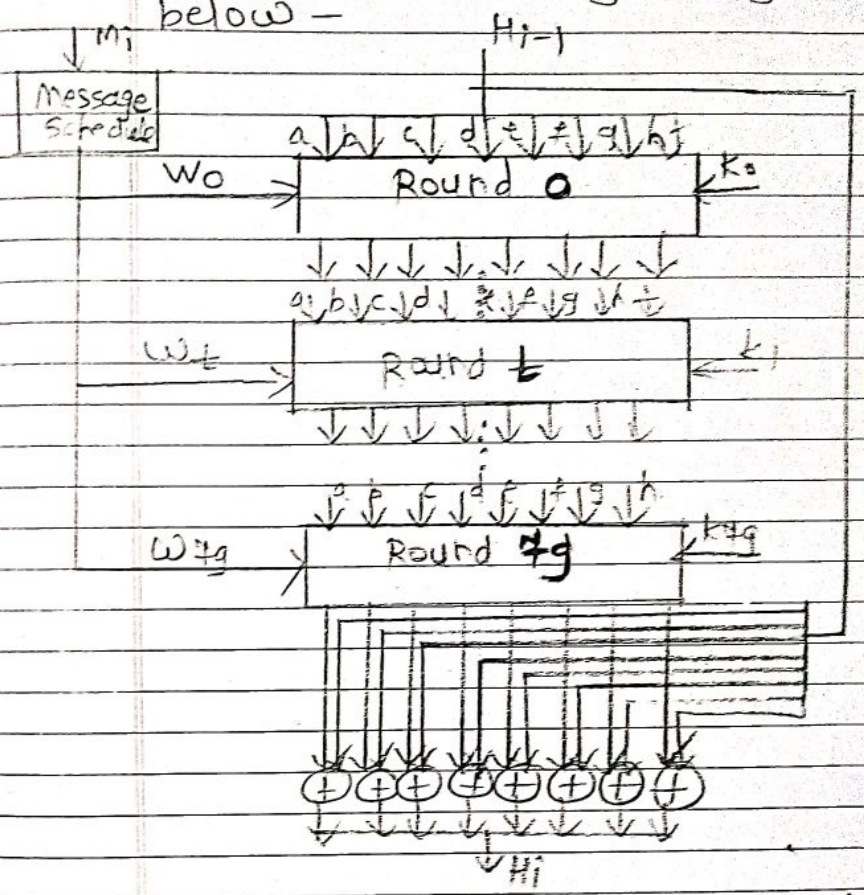
N = The no. of blocks in the msg

SUM_{64} = addition modulo 2^64 performed separately on each word of the pair of inputs.

MD = final msg digest value.

b)- Describe the SHA-512 bit processing of single 1024-bit block.

→ i) The Heart of an algorithm is a module that consists of 80 rounds; this module is labeled



is described in figure given below—

i) Each round takes as the input the 512 bit buffer value, abcd efgh, and updates the contents of the buffer. At input to the 1st round, the buffer has the value of the intermediate

hash value, H_{i-1}. Each round t makes use of 64 bit value W_t, derived from the current 1024-bit block being processed (M_i).

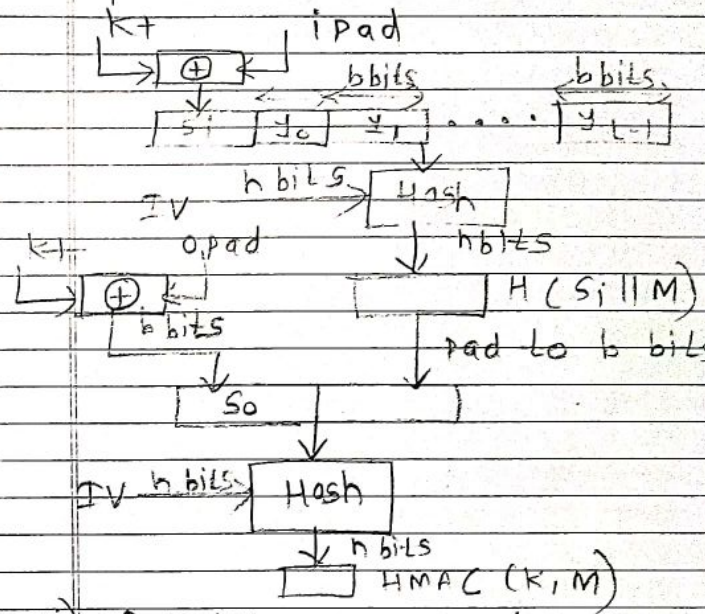
ii) These values are derived using a msg schedule described subsequently. Each round also makes use of an addive constants K_t, where $0 \le t \le 79$ indicates one of the 80 rounds.

iii) These words represent the first 64 bits of the fractional parts of the cube roots of the first 80 prime nos. The constants provide a "randomized" set of 64 bit patterns, which should eliminate any regular ities in the i/p data.

iv) The output of the eightieth round is added to the i/p to the first round (H_{i-1}) to produce H_i. The addition is done independently for each of the eight words in the buffer with each of the corresponding words in H_{i-1}, using addition modulo 2^{64}.

Q.5:- a) - Explain HMAC algorithm.

i) Figure illustrates the overall operation of the HMAC.



ii) Definitions of various terms used in it—

H = embedded hash function
IV = initial value i/p to hash func
M = message i/p to HMAC
Y_i = i-th block of M, $0 \le i \le (L-1)$
L = no. of block in M
b = no. of bits in a block
n = length of hash code produced

hash value, H_{i-1}. Each round t makes use of 64 bit value W_t, derived from the current 1024-bit block being processed (M_i).

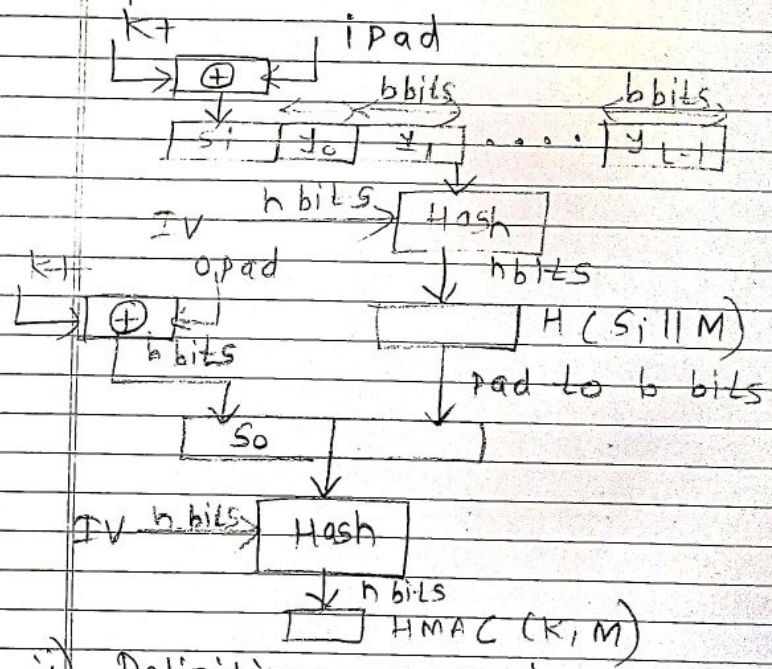
ii) These values are derived using a msg schedule described subsequently. Each round also makes use of an addive constants K_t, where $0 \le t \le 79$ indicate one of the 80 rounds.

iii) These words represent the first 64 bits of the fractional parts of the cube roots of the first 80 prime nos. The constants provide a "randomized" set of 64 bit patterns, which should eliminate any regular ities in the i/p data.

iv) The output of the eightieth round is added to the i/p to the first round (H_{i-1}) to produce H_i. The addition is done independently for each of the eight words in the buffer with each of the corresponding words in H_{i-1}, using addition modulo 2^{64}.

Q.5:- a) - Explain HMAC algorithm.

i) figure illustrates the overall operation of the HMAC.



ii) Definitions of various terms used in it—

H = embedded hash function

IV = initial value i/p to hash func

M = message i/p to HMAC

Y_i = i-th block of M, $0 \le i \le (L-1)$

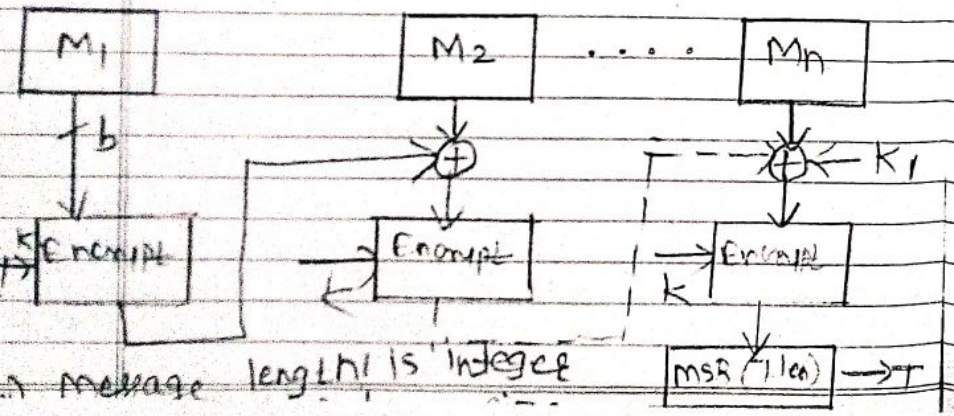
L = no. of block in M

b = no. of bits in a block

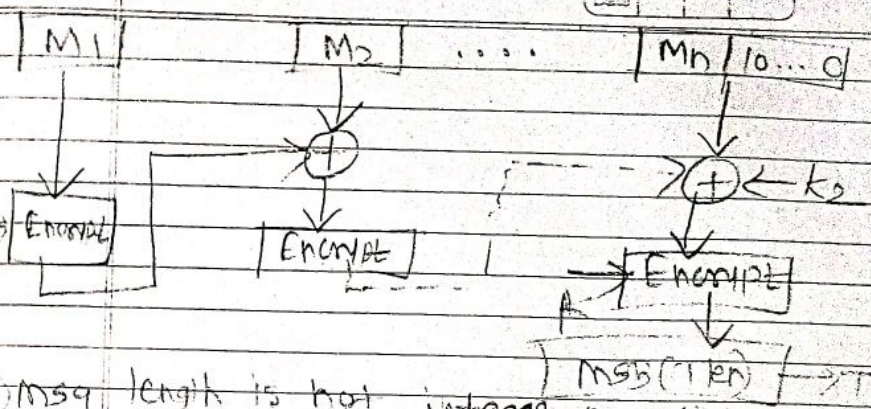
n = length of hash code produced by embedded hash function.

ment is adopted by the NIST is Cipher-based message Authentication code (CMAC) mode of operation for use with AES and triple DES.

ii) First let us define the operation of CMAC when the msg is an integer multiple n of the cipher block length b. For AES, b=128 and for triple DES, b=64. The msg is divided into n blocks ($M_1, M_2 \ldots M_n$). The algo makes use of a k-bit encryption key K & an b-bit constant K_1. For AES, the key size k is 128, 192 or 256 bits; for triple DES, the key size is 112 or 168 bits. CMAC is calculated as follows-



a) message length is integer



b) msg length is not integer multiple of block size

$C_1 = E(K, M_1)$

$C_2 = E(K, [M_2 \oplus C_1])$

$C_3 = E(K, [M_3 \oplus C_2])$

⋮

$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$

$T = MSB_{Tlen}(C_n)$

Where -

T = message authentication code

Tlen = bit length of T

$MSB_s(X)$ = the s leftmost bits of the bit string X

- If the msg is not an integer multiple of the cipher block length, then the final block is padded to the right with a 1 and as many 0's as necessary

K = secret key. recommended length is ≥ n. If key length is greater than b, the key is i/p to the hash function produce an n bit key

K+ = K padded with zeroes on the left so that the result is b bit in length

ipad = 00110110 (36 in hex)
opad = 01011100 (5C in hex)

Then HMAC can be expressed as
$$HMAC(K, M) = H[(K^+ \oplus opad) \| H[(K^+ \oplus ipad) \| M]]$$

We can describe the algorithm as follows—

1. Append zeroes to the left end of K to create a b-bit string K+.
2. XOR K+ with ipad to produce the b bit block S_i.
3. Append M to S_i
4. Apply H to the stream generated in step 3.
5. XOR K+ with opad to produce the b-bit block S_o.
6. Append the hash result from step 4 to S_o.
7. Apply H to the stream generated in step 6 & o/p the result.

Note that the XOR with ipad results in flipping one-half of the bits of K. Similarly, the XOR with opad results in flipping one-half of the bits of K, using a diff. set of bits. In effect by passing S_i and S_o through the compression function of the hash algorithm, we have pseudo randomly generated two keys from K.

HMAC should execute in approx. the same time as the embedded hash function for long messages. HMAC adds three executions of the hash compression function.

b) – Describe the CMAC algorithm with suitable diagram.

→ i) The two n-bit keys could be derived from the encryption key, rather than being provided separately. This refine

So that the final block is also of the length n. The CMAC operation then proceeds as before, except that a different n-bit key K_2 is used instead of K_1.

- The two n-bit keys are derived from the k-bit encryption key as follows:

$$L = E(K, 0^n)$$
$$K_1 = L \cdot x$$
$$K_2 = L \cdot x^2 = (L \cdot x) \cdot x$$

Q.6:- a) What are the requirements of digital signatures.

→ i) Following is the lists of types of attacks, in order to increasing severity. Here A denotes the user whose signature method is being attacked, & C denotes the attacker.

a) Key only attack:- C only knows A's public key.

b) Known message attack:- C is given access to a set of messages and their signatures.

c) Generic chosen message attack:- C chooses a list of messages before attempting to break A's signature scheme, independent of A's public key. C then obtains from A valid signatures for the chosen messages.

d) ~~attack~~ Directed chosen message attack:- similar to the generic attack, accept that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.

d) Total break:- C determines A's private key.

e) Universal forgery:- C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.

- Hence on the properties & attacks as above, we can formulate the following requirements for a digital signature:-

i) The signature must be a bit

Pattern that depends on the message being signed.

ii) The signature must use some information unique to the sender to prevent both forgery & denial.

iii) It must be relatively easy to produce the digital signature.

iv) It must be relatively easy to recognize & verify the digital signature.

v) It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.

vi) It must be practical to retain a copy of the digital signature in the storage.

b) Compare RSA and DSA approach for digital signature. Explain digital signature standard (DSS).

→ * Digital signature standard:-

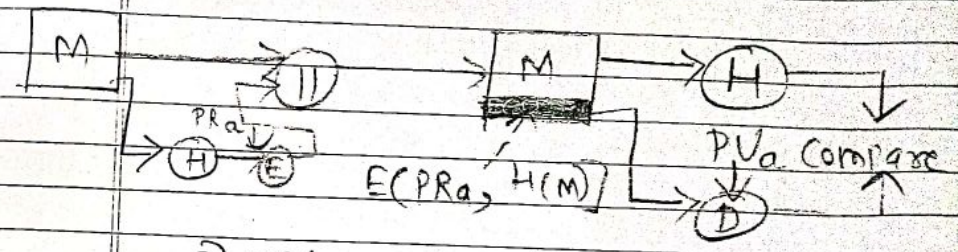
i) The DSS makes use of the Secure Hash Algorithm (SHA), and presents new digital signature technique, Digital Signature Algorithm (DSA).

ii) The DSS approach is as shown in the figure for generating digital signature to that used with RSA. In the RSA approach, the msg to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted. The receipent takes the msg and produces a hash code. The receipent also decrypts the signature using the sender's public key.

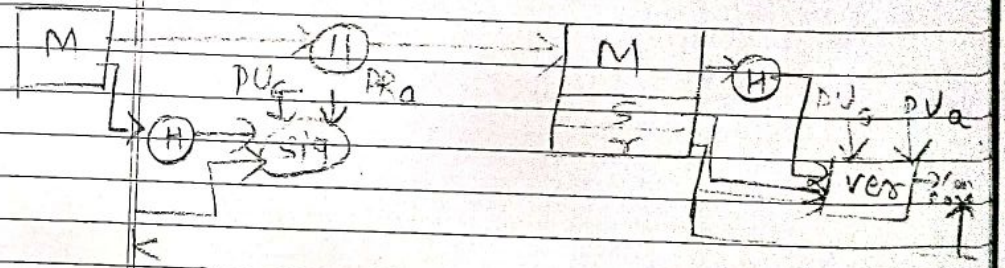
Scanned by CamScanner

If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only sender knows the private key, only the sender can produce the valid signature.

iii) The DSS approach also makes use of a hash function. The hash code is provided as input to a signature func along with a random number k generated for this particular signature. The signature function also depends on the sender's private key (PRa) and a set of parameters known to a grp of communicating principals. We can consider this set to constitute a global public key (PUG). The result is a signature consisting of two components, labeled as s & r.



a) RSA approach



b) The DSS approach

* Comparison betn RSA & DSA approach for digital signature:-

RSA	DSA
i) The RSA refers to the to the initials of the people who created it i.e. Ron Rivest, Adi Shamir, & Leonard Adleman.	i) The DSA refers to the digital signature algorithm.
ii) The RSA, on the	ii) The DSA was designed as an encryption algo.

Other hand looks at the difficulty of factoring nos as the main aspects of its development.	The main use is for digital signature which borrows heavily from ElGamal signature algo.
iii) RSA covers, signing in encryption and encryption of msg contained as well.	iii) This is a program that is mainly built for signing, and therefore it is quite popular with digital signatures. This however does not extend beyond the signature to the msg itself.
iv) when faster encryption is required RSA is preferred as it encrypt both message & signature for signing in.	iv) DSA produces keys very quickly. It preferred when faster key generation is needed.
v) while verification and encryption during decryption can be left to RSA.	v) It can be generally concluded that DSA is best suited for signing in ~~and~~ d

c) - Explain ElGamal digital signature scheme with suitable example.

→ i) Before proceeding, we need a result from number theory. For a prime no. q, if α is the primitive root of q, then -

$\alpha, \alpha^2, \ldots, \alpha^{q-1}$

are distinct (mod q). It can be shown that, if α is the primitive root of q, then -

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.

2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if & only if $i \equiv j \pmod{q-1}$.

ii) As with ElGamal encryption, the global elements of ElGamal digital signature are a prime no. q & α, which is a primitive root of q. User A generates a private/public key pair as follows -

1. Generate a random integer X_A such that $1 < X_A < q-1$.

2. Compute $Y_A = \alpha^{X_A} \bmod q$

3. A's private key is X_A; A's public key is $\{q, \alpha, Y_A\}$

To sign a message M, user A first computes the hash $m = H(M)$, such that m is an integer in the range $0 \le m \le q-1$. A then forms digital signature as follows-

1. Choose a random integer k such that $1 \le k \le q-1$ & $gcd(k, q-1) = 1$. That is k is relatively prime to $q-1$

2. Compute $S_1 = \alpha^k \bmod q$. Note that this is the same as the computation of C1 for ElGamal encryption

3. Compute $k^{-1} \bmod (q-1)$. That is, compute the inverse of k modulo $q-1$.

4. Compute $S_2 = k^{-1}(m - X_A S_1) \bmod (q-1)$

5. The signature consists of the pair (S_1, S_2).

Any user B can verify the signature as follows-

1. Compute $V_1 = \alpha^m \bmod q$
2. Compute $V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$

This signature is valid if $V_1 = V_2$

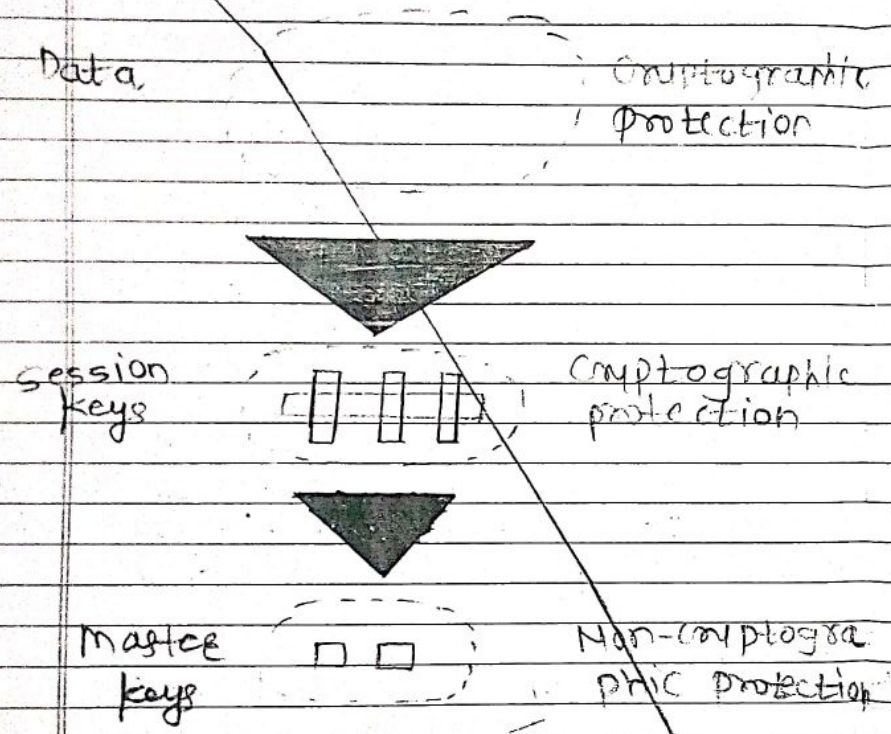
Q.7 a)- Describe different methods for symmetric key distribution

→ i) 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.

ii) Therefore the strength of any cryptographic system rests with key distribution technique, a term that refers to the means of delivering a key to two parties who wish to exchange data w/o allowing others to see the key.

Scanned by CamScanner

using a master key that is shared by the key distribution center & an end system or user.

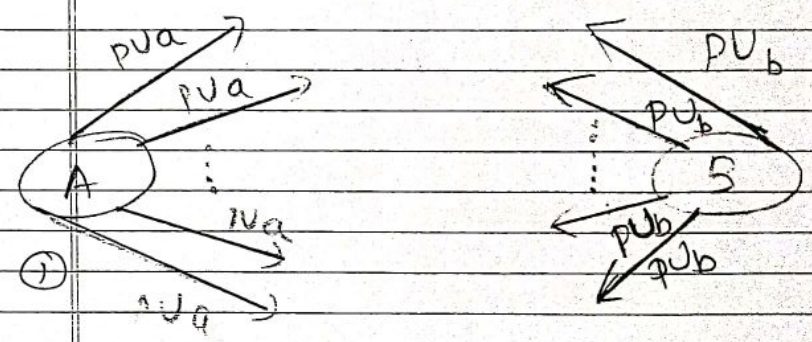


The use of key hierarchy

b)- List and explain different approaches for public key distribution.

i) Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes.

i) public announcement
ii) publicly available directory
iii) public key authority
iv) public key certificates

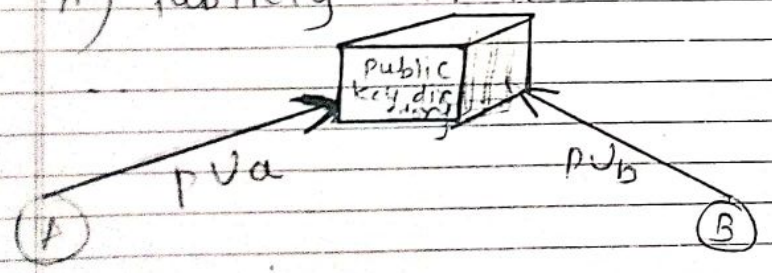


Uncontrolled public key distribution

i) public announcement:- on the face of it, the point of public key encryption is that the public key is public. Thus if there is some broadly accepted public key algo. 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. As (figure. 1). For example because for the PGP, which makes use of messages that they send to public forums, such as USENET news groups & Internet mailing lists.

Although this approach is convenient, it has a major weakness.

ii) Publicly Available Directory



- A greater security can be achieved by maintaining a publicly available dynamic directory of public key. It includes following elements -

1. The authority maintains a directory with a entry for each participant.
2. Each participant registers a public key with the directory authority.
3. participant may replace the existing key with a new one at any time either because of the desire to replace a public key that has already been used for a large amt of data, or because the corresponding private key has been compromised in some way.
4. participants could also access the directory electronically.

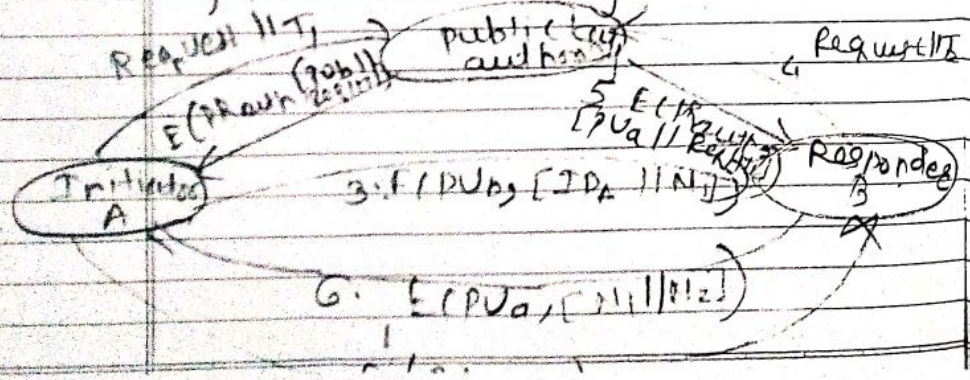
iii) public key authority -

Here in this approach following steps are occur -

1. A sends a time stamped

msg to the public-key authority containing a request for the current public key of B.

2. The authority respond with a msg that is encrypted using the authority's private key PRauth. Therefore, A is assured that the msg originated with the authority. The msg includes the following
- B's public key, PU_b, which A can use to encrypt messages destined for B.
- The original req. used to enable A to match this response with the corresponding earlier request & to verify that the original req. was not altered before reception by the authority.



- The original timestamp given so A can determine that this is not an old message from the authority containing a key other than B's current public key

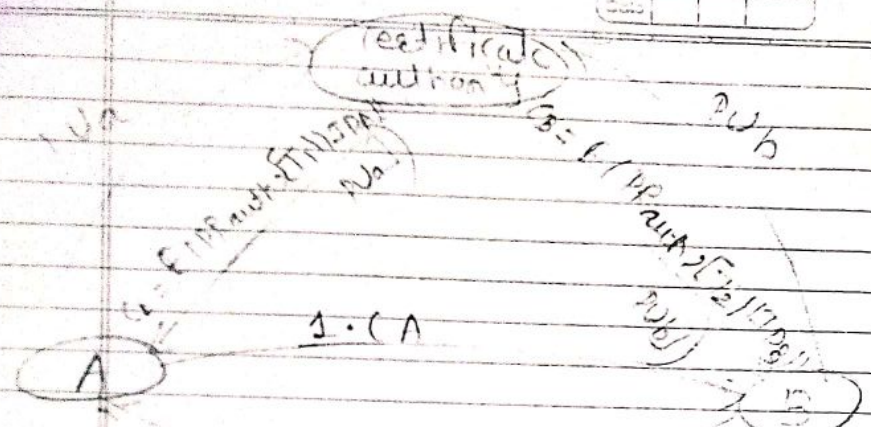
3. A stores B's public key & also uses it to encrypt a msg to B containing an identifier of A.

4,5. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

6. B sends msg to A encrypted with PU_a & containing A's nonce (N_1) as well as new nonce generated by B (N_2).

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

iv) public key Certificates:
The figure shows this approach-



1. Any participant can read a certificate to determine the name & public key of the certificate owner.

2. Any participant can verify that the certificate originated from the certificate authority & is not counterfeit.

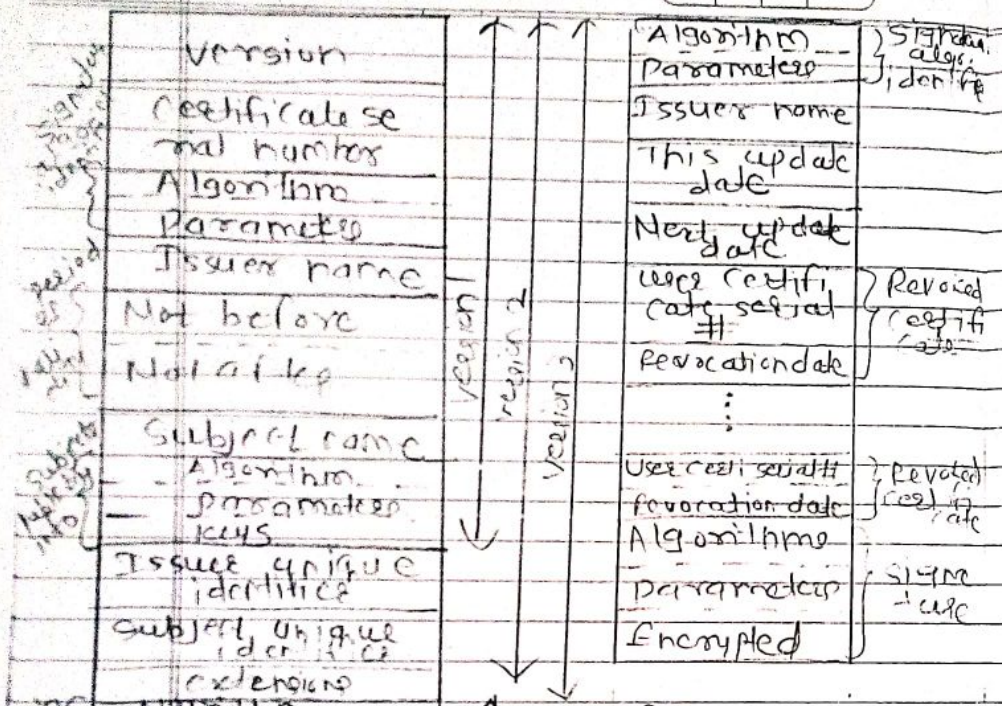
3. Only the certificate authority can create and update certificates.

4. Any participant can verify the currency of the certificate.

Q8 a) - Explain formate of X.509 certificate and certificate revocation list

→ i) 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 the CA or by the user.

ii) figure shows general format of a certificate which includes the following elements



b) Certificate Revocation list

X.509 format

i) Version:- Differentiates among successive versions of the certificate format; the default is version 1. If the issuer unique identifier or subject unique identifier are present, the value must be version 2. If one or more extensions are present, the version must be

ii) Serial number:- An integer value unique within the issuing CA that is unambiguously associated with this certificate

iii) signature algorithm identifier:- The algo. 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.

iv) Issuer name:- X.500 is the name of the CA that created & signed this certificate

v) period of validity:- consists of 2 dates:- The first & last on which the certificate is valid.

vi) Subject name:- The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.

Scanned by CamScanner

vii) 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.

viii) 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.

ix) 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 diff. entities.

x) Extensions – A set of one or more extension fields. extensions were added in version 3.

xi) Signature:– Covers all of 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 the signature algorithm identifier.

b) – Describe the concept of forward / reverse certificate & certification path with a suitable example.

→ i) Figure taken from X.509, is shown below. The connected circle indicates the hierarchical relationship among the CAs; the associated boxes indicate certificates maintained in the directory for each CA entry. The directory entry for each CA includes two types of certificates–

1) Forward certificate:– Certificates of X generated by other CAs.

2) Reverse certificates:– Certificates generated by X that are the certificates of other CAs.

ii) In the given example ie. shown in figure– A can acquire the following certificates from the directory to

establish a certification path to B

$X\langle\langle W\rangle\rangle W\langle\langle V\rangle\rangle V\langle\langle Y\rangle\rangle Y\langle\langle Z\rangle\rangle Z\langle\langle B\rangle\rangle$ →

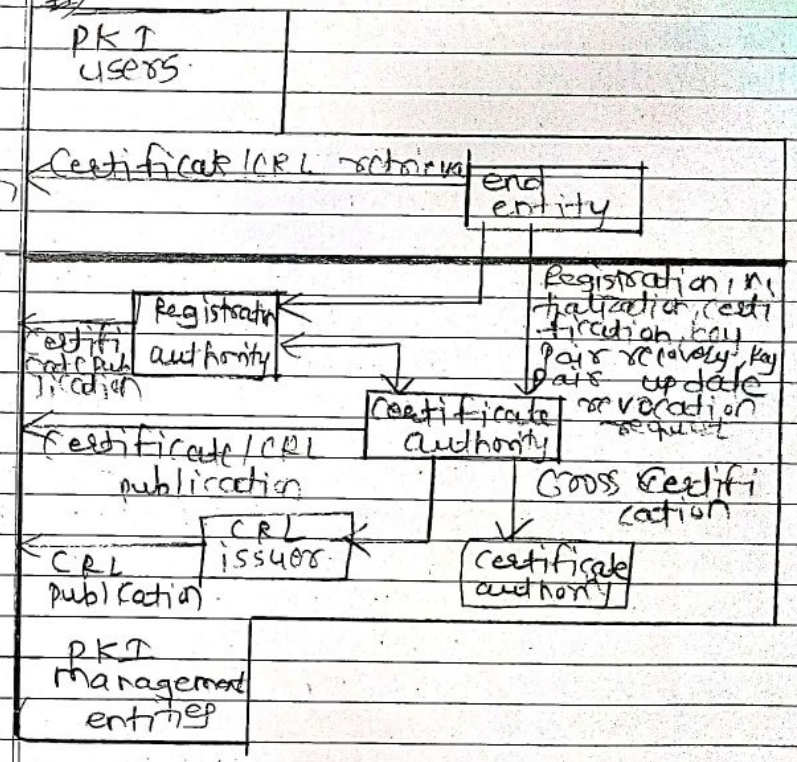
when A has obtained these certificates, it can unwrap the certification path in sequence to recover a trusted copy of B's public key. Using this public key, A can send encrypted messages to B. If A wishes to receive encrypted messages back from B, or to sign messages sent to B, then B will require A's public key, which can be obtained from the following certification path.

$Z\langle\langle Y\rangle\rangle Y\langle\langle V\rangle\rangle V\langle\langle W\rangle\rangle W\langle\langle X\rangle\rangle X\langle\langle A\rangle\rangle$

B can obtain this set of certificates from the directory or A can provide them as part of its initial message to B.

Q.9:- Describe public key infrastructure architectural model.

→ i) Figure shows the public key infrastructure X.509 (PKIX) model in detail.



ii) The various elements of this model are as follows →

i) End entity:- A generic term used to denote end users, devices, or any other entity that can be identified in the subject field of a public key certificate. End Entities typically consume and/or support PKI-related service

ii) Certification Authority (CA):- The issuer of certificates & certificate revocation lists (CRLs). It may also support a variety of administrative functions although these are often delegated to one or more registration authorities.

iii) Registration authority (RA):- An optional component that can assume a number of administrative functions from the CA. The RA is often associated with the end entity registration process but can assist in a no. of other areas as well.

iv) CRL issuer:- An optional component that a CA can delegate to publish CRLs.

v) Repository:- A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by end entities.

PKIX identifies a no. of management functions that potentially to be supported by management protocols. These are indicated in fig and include the following

① Registration:- This is the process whereby a user first makes itself known to a CA, prior to that CA issuing a certificate or certificates for that user.

② Initialization:- Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure.

③ Certification:- This is the process in which a CA issues a certificate for a user's public key, returns that certificate to the directory

user's client system, and/or posts that certificate in a repository.

(4) Key-pair recovery- Key pairs can be used to support digital signature creation and verification encryption & decryption, or both.

(5) Key-pair update:- All key pairs need to be updated regularly and new certificates issued. Update is required when the certificate lifetime expires and as a result certificate revocation.

(6) Revocation request:- An authorized person advises a CA of an abnormal situation requiring certificate revocation. Reasons for revocation include private key compromise, change in affiliation, and name change.

(7) Cross-certification: Two CAs exchange information used in establishing a cross-certificate. A cross certificate is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates.

Q.10:- Explain mutual authentication and one way authentication with suitable examples.

→ i) There are mainly 2 ways by which user - authentication takes place-
i) Mutual-Authentication
ii) One-way Authentication

(i) Mutual authentication:-
i) Such protocols enable communicating parties to satisfy themselves mutually about each others identity and to exchange session keys.
ii) Central to the problem of authenticated key exchange are 2 issues- Confidentiality & timeliness.
iii) To prevent masquerade and to prevent compromise of session keys, essential identification & session key information must be communicated in encrypted form.

This is the in which a CA issues a certificate for a user's public key, that certificate

iv) This requires prior existence of secret or public keys that can be used for this purpose. The second issue, timeliness, is important because of the threat of msg replay. lists of the following examples of replay attacks –

i) Simple replay
ii) Repetition that can be logged
iii) Repetition that can't be detected
iv) Backward replay w/o modification

v) one approach to coping with replay attack is to attach a sequence number to each msg used in an authentication exchange. A new msg is accepted only if its sequence no. is in proper order. But because of overhead, sequence no-s are generally not used for authentication and key exchange. Instead, one of the following two general approaches is used –

(i) Timestamps – party A accepts a msg as fresh only if the msg contains a time stamp, that in A's judgement is close enough to A's knowledge of current time.

(2) Challenge / response :- party A, expecting a fresh msg from B, first sends B a nonce and requires that the subsequent msg received from B contain the correct nonce value.

(ii) One-way authentication –

i) one appn for which encryption is growing in popularity is electronic mail. The very nature of electronic mail, & its chief benefit, is that it is not necessary for the sender & receiver to be online at the same time. Instead the email msg is forwarded to the receiver's electronic mail box, where it is buffered until the receiver is available to read it.

ii) The "envelope" or "header" of the email msg must be in the clear, so that the msg can be handled by the

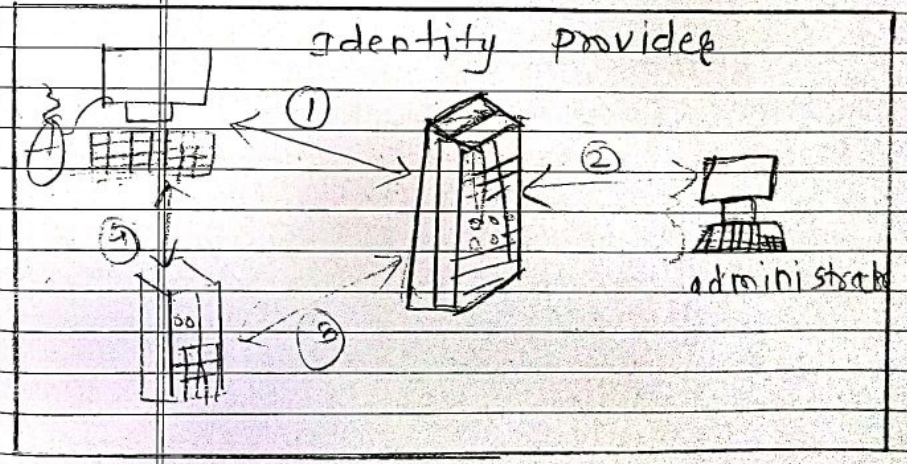
store & forward email protocol, such as SMTP or X.400.
A second requirement is that of authentication. Typically, the receipient wants some assurance that the msg is from the alleged sender.

Q.11:- a) Describe Kerberos version 4 message exchanges.

→ i) Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide the authentication service.

Q.12:- a) Explain federated Identity operation with suitable diagram.

→ i) Identity federation is, in essence, an extension of identity management to multiple security domains. Such domains include autonomous internal business units, external business partners, and other third party appns & services.
ii) Figure illustrates entities & data flow in a generic federated identity management architecture.
iii) The identity provider acquires attribute information through dialogue and protocol exchanges with users and administrators. For example, a user needs to provide a shipping address each time an order is placed at a new web merchant, & this the user to provide this information once, so that it is maintained in a single place & released to data consumers in accordance with authorization & privacy policies.



iv) service providers are entities that obtain & employ data maintained & provided by identity providers, often to support authorization decisions and to collect audit information. For example, a database server or file server is a data consumer that needs a client's credentials so as to know what access to provide to that client. A service provider can be in the same domain as the user & the identity provider. The power of this approach is for federated identity management in which the service provider is in a diff. domain.

In the figure —

① - End user's browser or other app engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.

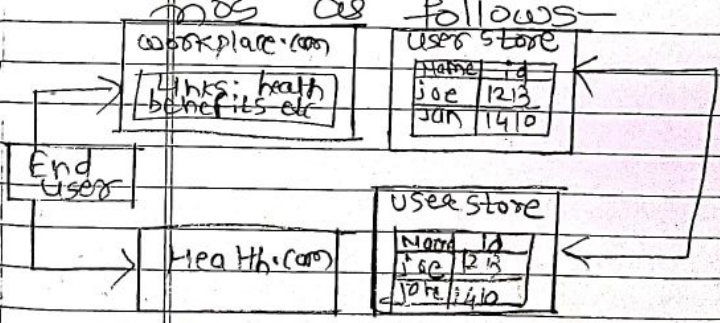
② Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.

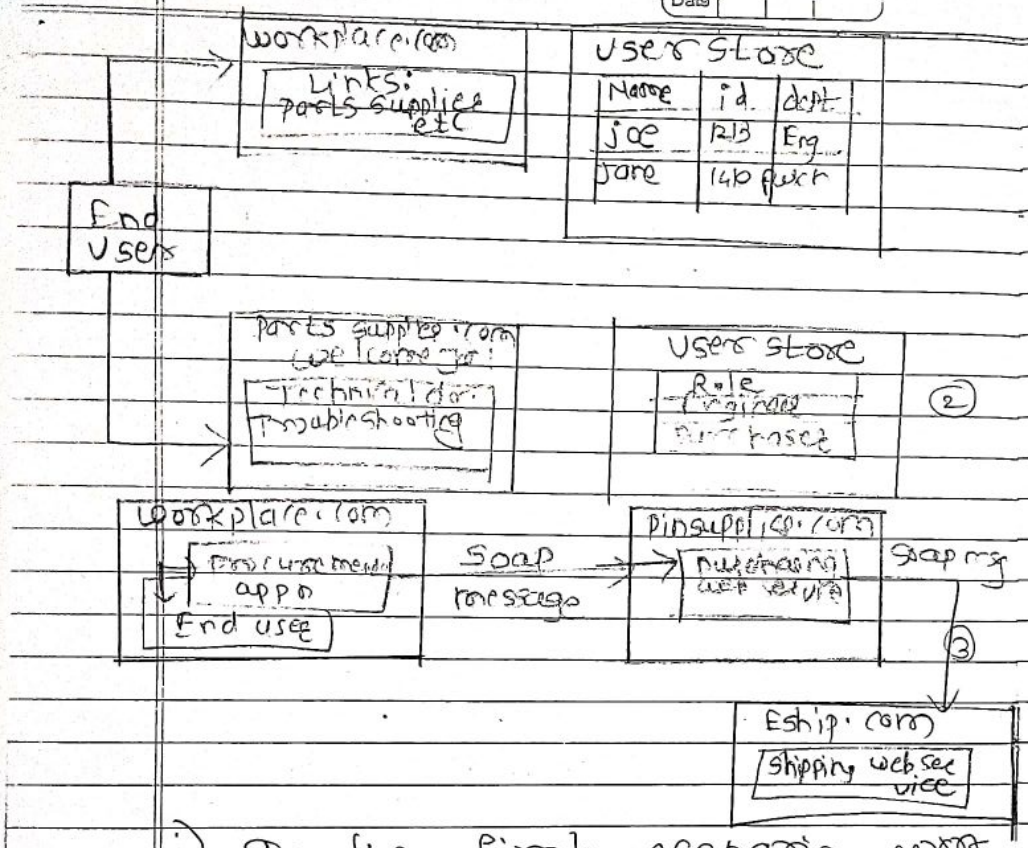
③ A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information & associated attributes from the identity provider in the source domain.

④ Service provider opens session with remote user and enforces access control restrictions based on user's identity & attributes.

b) Explain federated identity scenarios with diagrams.

→ i) To get some feel for the functionality of identity federation, we look at 3 scenarios as follows —





i) In the first scenario, workplace.com contracts with Health.com to provide employee health benefits. An employee uses a web interface to sign on to workplace.com & goes through an authentication procedure there. This enables the employee to access authorized services & resources at workplace.com. When the employee clicks on link to access health benefits, her browser is redirected to Health.com. At the same time workplace.com SIW passes the user's identifier to Health.com in a secure manner. The 2 organizations are part of a federation that cooperatively exchanges user identifiers.

ii) figure (2) shows second type of browser-based scheme. partssupplier.com is a regular supplier of parts to workplace.com. In this case, a role based access control (RBAC) scheme is used for access to information. An engineer of workplace.com authenticates at the employee portal at workplace.com & clicks on a link to access information at partssupplier.com. Because the user is authenticated in the role of an engineer he is taken to the technical documentation & trouble shooting portion of parts supplier.com's website w/o having to sign on. Similarly, an employee in a purchasing role signs on at workplace.com & is authorized in that role, to place purchases at

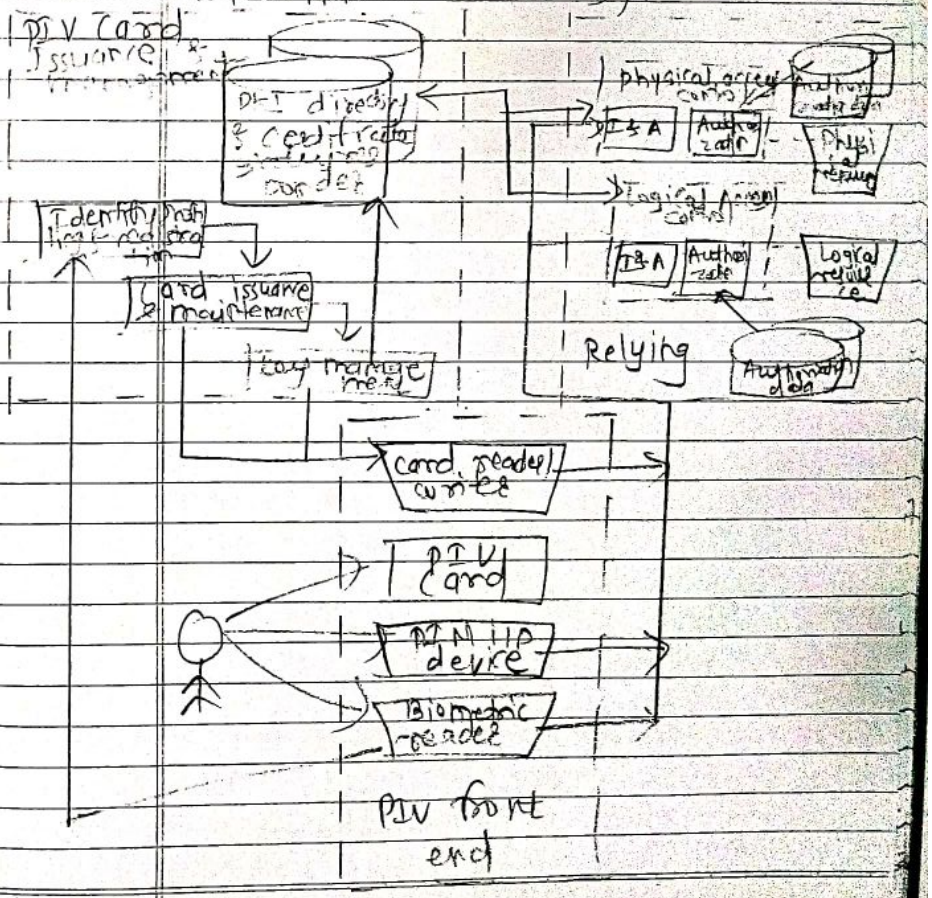
Parts supplier.com w/o having to authenticate to parts supplier.com. For this scenario, parts supplier.com does not have identity information for individual employees at workplace.com. Rather, the linkage betn the two federated partners is in terms of roles.

iii) The scenario illustrated in figure ③ can be referred to as document based rather than browser based. In this third example, workplace.com has a purchasing agreement with pinsupplier.com & pinsupplier.com has a business relationship with E-ship.com

Q.13 a)- Describe personal Identity Verification (PIV) System model.

→ i) figure illustrates the major components of FIPS 201-2 compliant systems. The PIV front end subsystem supports up to three factor authentication; the no. of factors used depends on the level of security required.

ii) The front end makes use of a smart card, known as a PIV card, which is a dual interface contact and contactless card. The card holds a cardholder photo graph, X.509 certificates, cryptographic keys, biometric data, and a card holder unique identifier (CHUID).



iii) The standard defines 3 assurance levels for verification of the card and the encoded data stored on the card, which in turn leads to verifying the authenticity of the person holding the credential. A level of some confidence corresponds to use of the card reader and PIN. A level of high confidence adds a biometric comparison of a fingerprint captured and encoded on the card during the card-issuing process and a fingerprint scanned at the physical access point. A very high confidence level requires that the process just described is completed at a control point attended by an official observer.

iv) The other major component of the PIV system is the PIV card issuance and management subsystem. This subsystem includes the components responsible for identity proofing & registration, card & key issuance and management & the various repositories & services (e.g., directory), required as part of the verification infrastructure.

v) The PIV system interacts with a relying subsystem which includes components responsible for determining a particular PIV card holder's access to a physical or logical resource.

vi) Unlike the typical card number/facility code encoded on most access control cards, the FIPS 201 CHUID takes authentication to a new level through the use of an expiration date & an optional CHUID signature.

b) - Explain authentication using PIV authentication key.

i) Using the electronic credentials resident on a PIV card, the card supports the following authentication mechanism:

a) CHUID:- The card holder is authenticated using the signed CHUID data element on the card. The PIN is not required. This mechanism is useful in environments where a low level of assurance is acceptable & rapid contactless authentication is necessary.

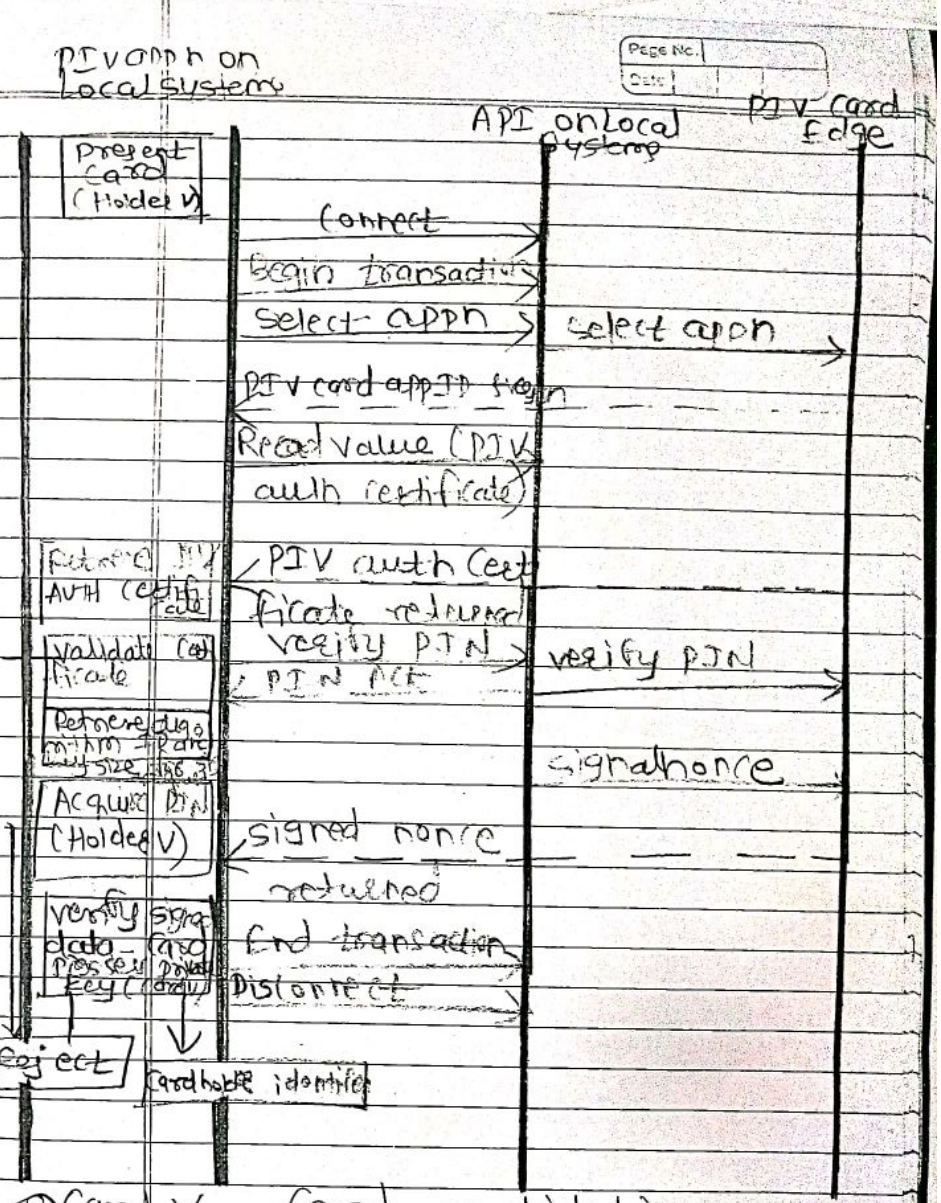
b) Card Authentication key – The PIV Card is authenticated using the card authentication key in a challenge response protocol. The PIN is not required.

c) BIO – The Cardholder is authenticated by matching his or her fingerprint sample to the signed biometric data element in an environment w/o human attendant in view.

d) BIO-A – The Cardholder is authenticated by matching his or her fingerprint sample to the signed biometric data element in an environment with a human attendant in view.

e) PKI – The Cardholder is authenticated by demonstrating (only) of the PIV authentication private key in challenge response protocol that can be validated using the PIV authentication certificate.

In each of the above use cases, except the Symmetric Card authentication key use case, the source & the integrity of the corresponding PIV Credential are



① Card V – Card Validation
② Cred V – Credential Validation
③ Holder V = Cardholder Validation

authenticated the

Key technique

validated by verifying the digital signature on the credential with the signature being provided by a trusted entity.

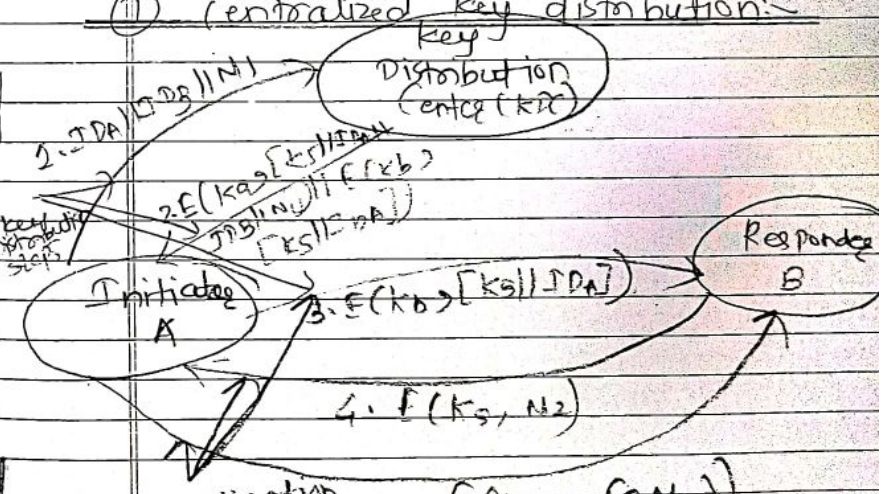
- figure illustrated as authentication scenario that includes the use of PIV authentication key. This provide high level assurance.
- The process begins when the local system detects the card either through an attached card reader or wirelessly. It then selects an appn on the card for authentication. The local system then requests the public key certificate for the card's PIV authentication key. If the certification is valid authentication continues otherwise card rejected.
- The scenario of fig. accomplishes 2 types of authentication. The combination of possession of the card and knowledge of the PIN service authenticates the cardholder. The PIV Authentication key certificate validates the cards credentials. The challenge-response protocol authenticates the card.

Q.7 a) ~~Describe Kerberos version 4 message exchanges~~

Q.7 - a) Describe different methods for symmetric key distribution.

i) The key distribution concept can be deployed in a number of ways. A typical scenerio is as illustrated below. This scenario assume that each user shares a unique master key with the key distribution center (KDC).

① Centralized key distribution:-



This is also known as centralized key technique.

Scanned by CamScanner

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 master key, K_a known only to itself & 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. Consist of identity of A & B & hence.

② The KDC responds with a msg encrypted using K_a. Thus A is the only one who can successfully read the msg, and A knows that it originated at the KDC.

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

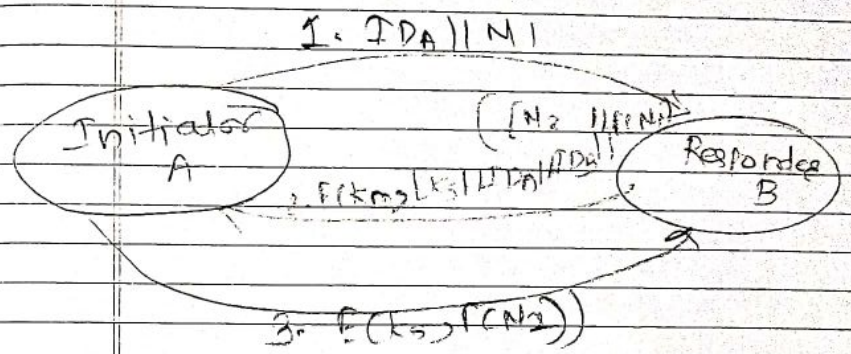
④ Using the newly minted session key for encryption, B sends a nonce N_2, to A.

⑤ Also using K_s, A responds with $f(N_2)$, where f is a function

1. $E(K_a, [N_1 || ID_B])$

that performs some transformation on N_2 (e.g. adding one

② Decentralized key distribution



This technique works as shown in the above figure.

- A decentralized approach requires that each end system be able to communicate in a secure manner with all potential partner end systems for purposes of session key distribution. Thus, there may need to be as many as $[n(n-1)]/2$ master keys for a configuration with n end systems.

- A session key may be established with the following sequence of steps-

①. A issues a request to B for a session key & includes a nonce N_1.

② B responds with a msg that is encrypted using the shared master key. The response include the session key selected by B, an identifier of B, the value of f(N1) & another nonce, N_2.

③ Using the new session key, A returns f(N_2) to B

③ Using Asymmetric distribution

1) simple secret key distribution

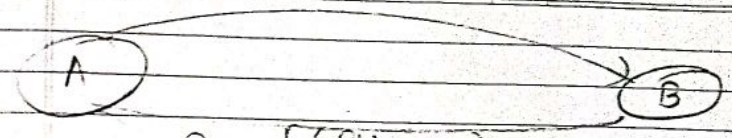
1. A generates public/private key pair [PUa, PRa] & transmits a msg to B consisting of PUa & an identifier of A, ID_A.

2. B generates a secret key, ks, and transmits it to A, which is encrypted with A's public key.

3. A computes D(PRa, E(PUa, ks)) to recover the secret key. Because only A can decrypt the msg, only A & B will know the identity of ks.

4. A discards PUa & PRa & B discards PUa.

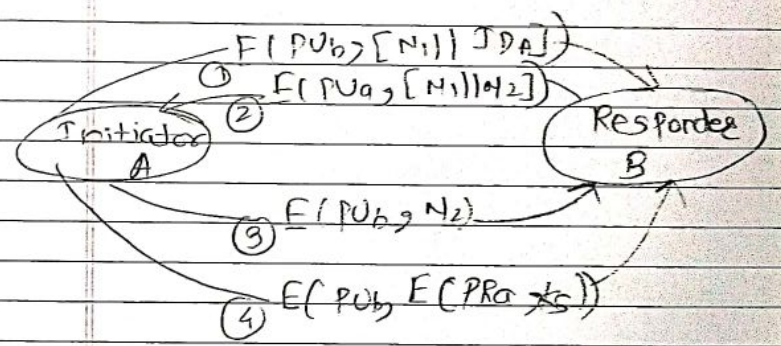
1. PUa || ID_A



2 E(PUa, ks)

But in this case man-in-the-middle attack is possible. Hence we go with 2nd technique as follows

ii) Secret key distribution with Confidentiality & Authentication



1. A uses B's public key to encrypt a msg 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 msg to A encrypted with PUa & containing A's nonce (N_1) as well as new nonce N_2.

2. AS → C: $E(K_c, \text{Ticket}_{tgs})$
3. C → TGS: $ID_c \| ID_v \| \text{Ticket}$

ing ticket using a key shared only by the AS and the TGS and verifies the

3. A returns N2, encrypted using B's public key, to assure B that its correspondent is A.
4. A selects a secret key K_s, and sends $M = E(PU_b, E(PR_a, K_s))$ to B. Encryption of this msg with B's public key ensures that only B can read it; encryption with A's private key ensure that only A could have sent it.

Q.11 a) - Describe Kerberos version 4 message exchanges

→ i) Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide the authentication service.
ii) In this message exchange we mainly have 2 scenarios. The first one is as follows-

Consider the following hypothetical dialogue -

① C → AS: $ID_c \| P_c \| ID_v$
AS → C: Ticket
C → V: $ID_c \| \text{Ticket}$
$\text{Ticket} = E(K_v, [ID_c \| AD_c \| ID_v])$

where, 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 & V.

- In this scenario user logs onto a work station & requests access to server V. The client module C in the user's workstation requests the user's password & then sends a message to the AS that includes the user's ID, the server ID & the user's password. The AS checks its database to see if the user has supplied the proper password for this user ID and whether this user is permitted access to server V. In this way process goes on till 3. But still this scenario involves 2 problems i.e. plain text transmission of password and - open n/w environment problem.
Hence we go with 2nd scenario →

1. C → AS: $ID_c \| ID_{tgs}$
2. AS → C: $E(k_c, Ticket_{tgs})$
3. C → TGS: $ID_c \| ID_v \| Ticket_{tgs}$
4. TGS → C: $Ticket_v$
5. C → V: $ID_c \| Ticket_v$

$Ticket_{tgs} = E(k_{tgs}, [ID_c \| AD_c \| ID_{tgs} \| TS_1 \| Lifetime])$

$Ticket_v = E(k_v, [ID_c \| AD_c \| ID_v \| TS_2 \| Lifetime_2])$

1. The client requests a ticket granting ticket on behalf of the user by sending its user ID to the AS, together with the TGS I.D, indicating a request to use the TGS service.

2. The AS responds with a ticket that is encrypted with a key that is derived from the user Password (k_c), which is already stored at the AS.

3. The client requests a service-granting ticket on behalf of the user.

4. The TGS decrypts the incoming ticket using a key shared only by the AS and the TGS (k_{tgs}) and verifies the success of the decryption by the presence of its ID.

5. The Client requests access to a service on behalf of the user.

But if mutual authentication is required then the server can reply as shown in msg 6.

6. V → C $E(k_{c,v}, [TS_5 + 1])$

$Ticket_v = E(k_v, [k_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(k_{c,v}, [ID_c \| AD_c \| TS_5])$

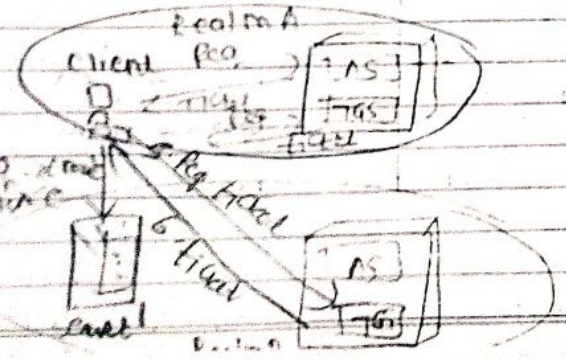
b) – Describe different elements of kerberos realm and explain Inter-realm authentication in kerberos version 4.

→ i) A full service of kerberos environment consisting of a kerberos server, a no. of clients and a no. of appn servers requires the following

① - The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.

② The Kerberos server must share a secret key with each server.

1. $C \rightarrow AS$: $ID_c \| ID_{tgs} \| TS_1$
2. $AS \rightarrow C$: $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$
3. $C \rightarrow TGS$: $ID_{tgs} \| Ticket_{tgs} \| Authenticator_c$
4. $TGS \rightarrow C$: $E(K_{c,tgs}, [K_{c,tgs} \| ID_{tgs} \| TS_4 \| Ticket_{tgs}])$
5. $C \rightarrow TGS_{rem}$: $ID_{vrem} \| Ticket_{tgs} \| Authenticator_c$
6. $TGS_{rem} \rightarrow C$: $E(K_{c,tgs}, [K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}])$
7. $C \rightarrow V_{rem}$: $Ticket_{vrem} \| Authenticator_c$



Such a environment is referred to as a Kerberos realm. The concept of realm is explained as follows. A Kerberos realm is set of managed nodes that share the same Kerberos database. This database resides on the Kerberos master computer system which should be kept in a physically secure room. A read only copy of the Kerberos database might also reside on other Kerberos computer systems. Changing or accessing the contents of a Kerberos database requires the Kerberos master password. A related concept is that of a Kerberos principal, which is a service or user that is known to the Kerberos system.

- Kerberos provides mechanism for supporting inter-realm authentication —

For two realms to support inter-realm authentication, a third requirement is added —

3. The Kerberos server in each interoperating realm shares a secret

key with the server in the other realm. The two Kerberos servers are registered with each other.

The scheme require that the Kerberos server in one realm must trust the Kerberos server in the other realm to authenticate its users

Q- Describe Kerberos version 5 message exchange.

→ 1. $C \rightarrow AS$ options || ID$_c$ || Realm$_c$ || ID$_{tgs}$ || Times || Nonce$_1$

2. $AS \rightarrow C$ Realm$_c$ || ID$_c$ || Ticket$_{tgs}$ || E$_{K_c}$[K$_{c,tgs}$ || Times || Nonce$_1$ || Realm$_{tgs}$ || ID$_{tgs}$]

3. $C \rightarrow TGS$ options || ID$_v$ || Times || Nonce$_2$ || Ticket$_{tgs}$ || Authenticator$_c$

4. $TGS \rightarrow C$ Realm$_c$ || ID$_c$ || Ticket$_v$ || E$_{K_{c,tgs}}$[K$_{c,v}$ || Times || Nonce$_2$ || Realm$_v$ || ID$_v$]

5. $C \rightarrow V$ options || Ticket$_v$ || Authenticator$_c$

6. $V \rightarrow C$ E$_{K_{c,v}}$[TS$_2$ || subkey || seq#]

- First consider the authentication service exchange. message (1) is a client request for a ticket granting ticket. The following new elements are added—

* Realm — Indicates realm of user
* Options — Used to req that certain flags to be set in the returned ticket.
* Times — 1) from
  2) till
  3) rtime
* Nonce

- message returns (2) a ticket granting ticket, identifying information for the client, and a block encrypted using the encryption key based on the user's password.
- message 3 in addition includes the requested times and options for the ticket and nonce— all with functions similar to those of msg (1).
- message (4) has the same structure that of msg (2). It returns a ticket plus information needed by the client, with the information encrypted using the session

key now shared by the client & the TGS

- Finally, for the client/server authentication exchange, several new features appear in version 5. In msg (5), the client may require as an option that mutual authentication is required. The authenticator include several new fields
(1) subkey
(2) sequence number

d) State & explain environmental & technical short comings of version 4.

→ - Kerberos version 4 was developed for use within the project athena environment and, accordingly, did not fully address the need to be of general purpose. This led to the following environmental short comings

(1) Encryption system dependence — version 4 requires the use of DES. Export restriction on DES as well as doubts about the strength of DES were thus of

Ans:- Both version

(2) Internet protocol dependence — version 4 requires the use of Internet protocol (IP) addresses. Other address types, such as the ISO NSAP address, are not accommodated.
(3) Message byte ordering — In version 4, the sender of a msg employs a byte ordering of its own choosing and tags the msg to indicate least significant byte in lowest address or most significant byte in lowest address.
(4) Ticket lifetime — Lifetime values in version 4 are encoded in an 8 bit quantity in units of five minutes. Thus the maximum time that can expressed is $2^8 \times 5 = 1280$ minutes.
(5) Authentication forwarding — version 4 does not allow credentials issued to one client to be forwarded to some other host & used by some other client.
(6) Inter realm authentication — In version 4, interoperability among N realms requires on the order of N^2 kerberos-to-kerberos relationships.

Apart from these environmental limitations, there are technical deficiencies in version 4 protocol itself.

1. - Double encryption:- Note in msg 2 & 4 that tickets provided to clients are encrypted twice - once with the secret key of the target server & then again with a secret key known to the client.

2. PCBC encryption:- Encryption in version 4 makes use of a nonstandard mode of DES known as propagating cipher block chaining (PCBC). It has been demonstrated that this mode is vulnerable to an attack involving the interchange of ciphertext blocks.

3. Session keys - Each ticket includes a session key that is used by the client to encrypt the authenticator sent to the service associated with that ticket. In addition, the session key may subsequently be used by the client and the server to protect messages passed during that session.

4. Password attacks:- Both versions are vulnerable to a password attack. The msg from the AS to the client includes material encrypted with a key based on the client's password. An opponent can capture this msg and attempt to decrypt it by trying various passwords. If the result of a test decryption is of the proper form, then the opponent has discovered the client's password and may subsequently use it to gain authentication credentials from Kerberos.

Q.3 a) Explain how encryption can be used for message authentication.

→ i) In this case ciphertext of the entire message serves as its authenticator.

ii) Message encryption by itself can provide a measure of authentication. The analysis differs for symmetric & public-key encryption schemes.

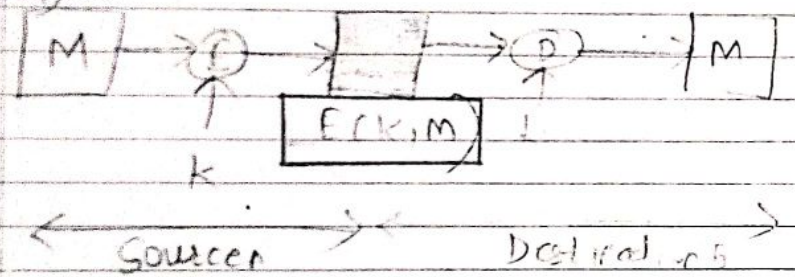
iii) Consider the straightforward use of symmetric encryption (a): A message M transmitted from source A to dest. B is encrypted using a secret

iv) If no other party knows the the key, then confidentiality is provided. No other party can recover the plain text of the message.

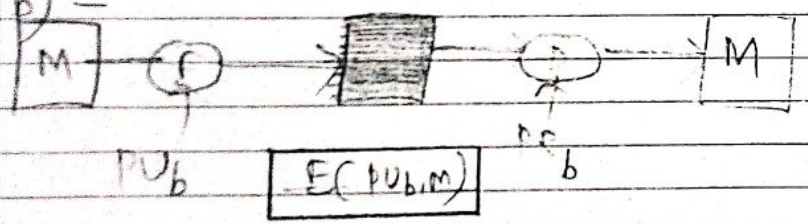
v) In addition B is assured that the message was generated by A.

vi) Following are the 4 ways by which you can achieve various properties using message encryption.

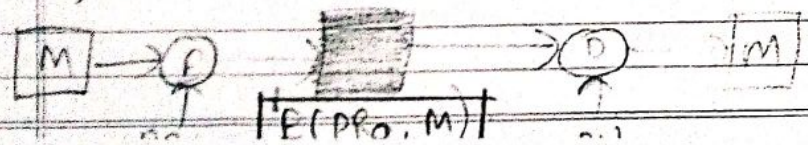
a)



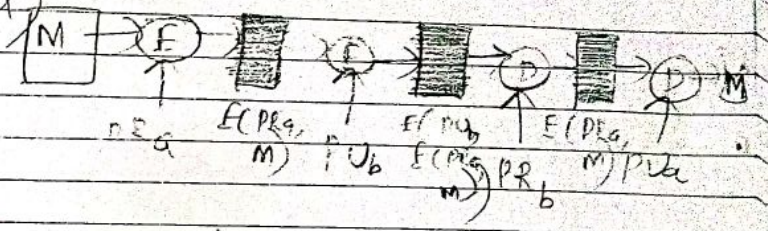
b)



c)



d)



a)- Symmetric encryption: Confidentiality & authentication

b)- public key encryption: Confidentiality

c)- public key encryption: authentication & signature

d)- public key encryption: confidentiality, authentication & signature.