T.Y. B.Sc. (IT) : Sem. V

Network Security

Time : $2\frac{1}{2}$ Hrs.] Prelim Question Paper Solution [Marks : 75

Q.1 Answer any TWO of the following: [10]

Q.1(a) Explain with example different approaches to implement security model. [5]

(A) An organization can take several approaches to implement its security model. These approaches are :

(1) No security :
In this approach, there is no security at all.

(2) Security through obsecurity :
In this model a system is secure simply because nobody knows about its existence and contents. This approach cannot work for too long as there are many ways an attacker can come to known about it.

(3) Host security :
In this scheme, the security for each host is enforced individually. This is very safe approach but the trouble in that it cannot scale well. The complexity and diversity of modern sites/organizations makes the task even harder.

(4) Network security :
Host security is tough to achieve as organizations grow and become more diverse. In this technique, the focus is to control network access to various hosts and this services, rather than individual host security. This is very efficient and scalable model.

Q.1(b) Encrypt the message 'Come Home Tomorrow' using [5]
(i) Ceaser Cipher
(ii) Simple Columner Transposition Techniques with four columns. Order is 3, 2, 4, 1

(A) 'Come Home Tomorrow'

(i) Ceaser cipher :
We need to shift 3 characters ahead in the alphabet.
C will be replaced with f
e " " h

PT = 'Come Home Tomorrow'
CT = FRPH KRPH WRPRUURZ

A b c d e f g h i j k l m n o p q r s t u v w x y z

(ii) Simple Columnar Transposition with four columns :

1	2	3	4
C	O	M	E
H	O	M	E
T	O	M	O
R	R	O	W

Column order is 3241
= MMM000REEOWCHTR
So the cipher text is
CT = MMMOOOOREEOWCHTR

Q.1(c) Explain how attackers misuse cookies to collect important information. [5]

(A) Cookies are small text files crested by your web browser and saved on your computer. Most cookies are session cookies, cresting only for the duration of your website visit and are deleted when you close your browser. No data is collected from cookies from which you can be personally identified.

Cookies are used in online or virtual shopping baskets cookies allowed for the contents of the basket to be changed based on the users actions and tracked items in the basket between browser session.

Cookies are also used when users log in to a website. Users enter their username and password into a login page and if they are authenticated a cookies is saved that allows the website to known the users are already logged in as they navigate around the site. This permits them access to any functionality that may be available only to logged in users probably the primary use of cookies at this time.

Cookies are used to track user actions across the site or domain. There are also some third party cookies that allow for tracking across multiple site. Most tracking done within site or domain is done to gather usage data for that site tracking over multiple sites is done by advertising companies to build usage profiles to allow for more targeted marketing.

Cookies can be exploited by connecting to the same site using the strlen, cookies, thus spoofing the original owner of the cookie. Attackers can modify the value of a cookie before sending them back to the server.

Q.1(d) List possibilities of attacks when the sender of a message encrypts the plain text message into its corresponding cipher text. [5]

(A) Possible types of attacks :

(1) Cipher–text only attack : In this type of attack the attacker does not have any due about plain text. She has some or all of the cipher text. The attacker analyses the cipher text at leisure to try and figure out the original plain text. Based on the frequency of letters, the attacker makes an attempt to guess the plain text.

(2) Known plain text attack :
In this case, the attacker known about some pair of plain text and corresponding cipher text for these pair. Using this information, the attacker tries to find other pairs and therefore know more and more of plain text. Example of such known plain text attack are company banners life headers etc. which are found commonly in all the documents of a particular company.

(3) Chosen plain–text attack :
Here the attacker selects a plain text block and tries to look for the encryption of the same in the cipher text. Here the attacker is able to choose the message to encrypt. Based on this the attacker intentionally picks patterns of cipher text that result in obtaining more information about the key.

(4) Chosen cipher text attack :
In the chosen cipher text attack the attacker knows the cipher text to be decrypted the encryption algorithm that was used to produce this cipher text and the corresponding plain text block. The attackers job is to discover the key used for encryption. However this type of attack is not very commonly used.

(5) The chosen text attack :
The chosen text attack is essentially a combination of chosen plain text attack and the chosen cipher text attack.

Q.2 Answer any TWO of the following : [10]

Q.2(a) List different types of cryptography algorithms. Explain with example. [5]

(A) Algorithm Types :

(1) Stream ciphers :
In stream ciphers, the plain text is encrypted one bit at a time. Suppose the original message is pay 100 in ASCII. When we convert these ASCII characters to their binary values, let us assume that it translates to 01011100.

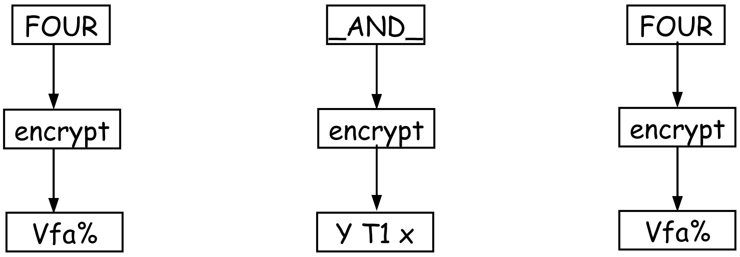
Let us also assume that we apply the XOR logic as the encryption algorithm. XOR is quite simple to understand. In simple terms, XOR produces on output of 1, only if one input is 0 and other is 1. The output is 0 if both the inputs are 0 or if both the inputs are 1.

Pay 100	0 1 0 1 1 1 0 0 1	P.T.
+ XOR	1 0 0 1 0 1 0 1 1	Key
	1 1 0 0 1 0 0 1 0	C.T.

Block cipher :

In block cipher rather then encrypting one bit at a time, a block of bits is encrypted at one go suppose we have a plain text FOUR_AND_FOUR that needs to be encrypted using block cipher, FOUR could be encrypted first followed by _AND_ and finally FOUR. Thus one block of characters gets encrypted at a time.

During decryption each block would be translated back to the original form. In actual practice the communication takes place only in bits. Therefore, FOUR actually means the binary equivalent of the ASCII characters FOUR . After any algorithm encrypts these the resultant bits are converted back into their ASCII equivalents. Therefore, we get funny symbols such as Vfa% etc. In actual practice, their binary equivalents are received which are decrypted back into binary equivalent of ASCII FOUR.



Q.2(b) Explain the steps in various rounds of AES. **[5]**

(A) Processes in each round :

(1) Apply S–box to each of the plain text_bytes :

The contents of the state array are looked up into the S–box. Byte–by–byte substitution is done to replace the contents of the state array with the respective entries in the S–box.

(2) Rotate row k of the plain text block by_k bytes :

Here each of the four rows of the state array are rotated to the left. Row 0 is rotated 0 bytes (i.e. not rotated at all) row 1 rotated by 1 byte, row 2 is rotated 2 bytes and row 2 is rotated 3 bytes. This helps in diffusion of data.

Original array Modified array

1	5	9	13	1	5	9	13
2	6	10	14	6	10	14	2
3	7	11	15	11	15	3	7
4	8	12	16	16	4	8	12

(3) Perform a mix column operation :

Now each column is mixed independent of the other. Matrix multiplication is used. The output of this step is the matrix multiplication of the old values and a constant matrix.

The multiplication is performed one column at a time. Each value in the column is eventually multiplied against every value of the matrix. The results of these multiplications are XORed together to produce only 4 resulting bytes for the next state. We together have 4 bytes of input, 16 multiplications, 12 XORs and 4 bytes of output. The multiplications is performed one matrix row at a time against each value of a state column.

b_1 = (b_1 * 2) XOR (b_2 * 3) XOR (b_3 * 1) XOR (by *1)
b_2 = (b_1 * 1) XOR (b_2 * 2) XOR (b_3 * 3) XOR (by *1)
b_3 = (b_1 * 1) XOR (b_2 * 1) XOR (b_3 * 2) XOR (by *3)
by = (b_1 * 3) XOR (b_2 * 1) XOR (b_3 * 1) XOR (by *2)

(4) XOR the state with the key block :
This step XORs the key for this round into the state array.

Q.2(c) Explain subkey generation process of blowfish algorithm. **[5]**

(A) Subkey Generation of Blowfish :
Let us understand the process step by step :

(1) Blowfish makes use of a very large number of sub keys : There keys have to be ready before encryption and decryption happen. The key size ranges from 32 bits to 448 bits. In other words, the key size ranges from 1 to 14 words each comprising a word of 32 bits. These keys are stored in an array.
$K_1, K_2 \ldots K_n$, $n \le 14$

(2) P array consists of 18 32 bit sub keys : $P_1, P_2 \ldots P_{18}$

(3) Four S boxes each consisting 256 32–bit entries :

S_1, 0	S_1, 1	…	S_1, 255
S_2, 0	S_1, 1	…	S_2, 255
S_3, 0	S_3, 1	…	S_3, 255
S_4, 0	S_4, 1	…	S_4, 255

Let us examine how all this information is used to generate sub keys :

(a) Initialize P array first followed by S boxes.

P_1 = 2345FA28
P_2 = 2754F289
⋮
S_4, 254 = 53217A56
S_4, 255 = 5AC2D561

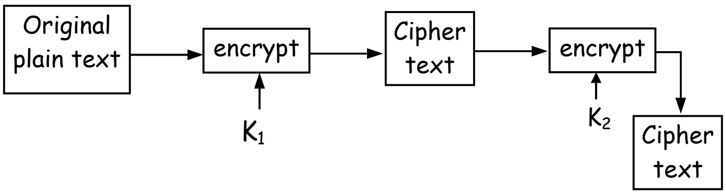
(b) Do bit wise XOR of P_1 and K_1, P_2 and K_2 etc. until please reuse K_1 to K_4.

P_1 = P_1 XOR K_1
P_2 = P_2 XOR K_2
⋮
P_{14} = P_{14} XOR K_{14}
P_{15} = P_{15} XOR K_1
P_{16} = P_{16} XOR K_2
P_{17} = P_{17} XOR K_3
P_{18} = P_{18} XOR K_4

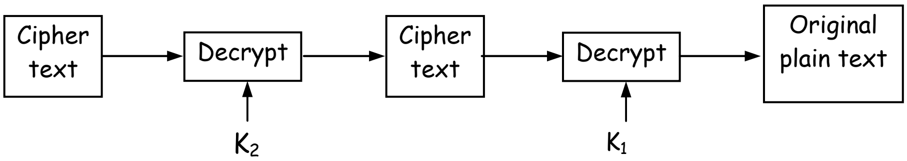
(c) Now take a 64 bit block, will all the 64 bits initialized to value of 0. Use the above P array and S box to run blowfish encryption process on a 64 bit all non zero block. To generate the sub key themselves the blowfish algorithm is used.

Q.2(d) Explain double DES algorithm. **[5]**

(A) Double DES :
Double DES is simple to understand. It does twice what DES normally does only once. Double DES uses 2 keys, K_1 and K_2 . It first performs DES on the original plain text using K_1 to get the encrypted text. It again perform DES on the encrypted text but this time with the other key K_2.



The decryption process would work in exactly the reverse order :



If we use a key of just 1 bit, there are two possible keys (0, 1). If we use a 2 bit key, there are four possible key values (00, 01, 10, 11). In general, if we use on n–bit key, the cryptanalyst has to perform 2^{nd} operations to try out all the possible keys. If we use two different keys, each constiting of n bits, the cryptanalyst would need 2^{2n} attempts to crack the key. Double DES would require a key search of ($2^{2 \times 56}$) i.e. 2^{112} keys. Merkle and Hellman introduced the concept of the meet_in_the_middle attack. This attack involves encryption from one end, decryption from the other and matching the results in the middle, hence the name meet_in_the_middle attack.

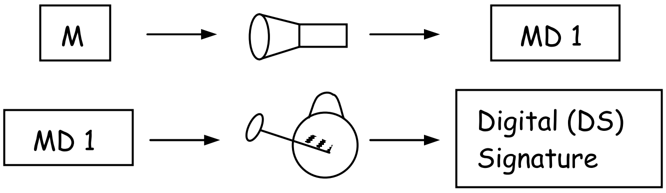
Q.3 Answer any TWO of the following: [10]

Q.3(a) How is RSA is used in digital signatures? Explain. [5]

(A) RSA and Digital Signatures :

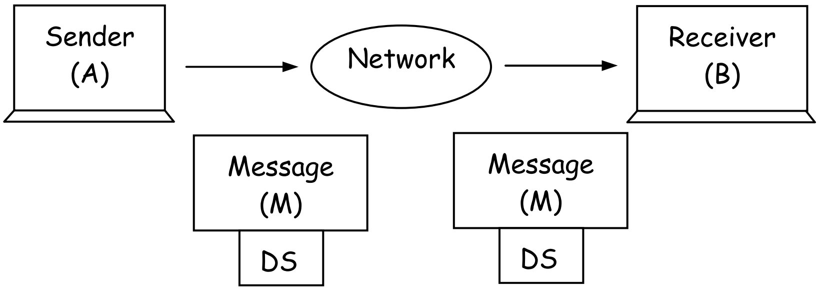
Step-1 : The sender (A) uses the SHA–1 message_digest algorithm to calculate the message digest (MD1) over the original message (M).

Step-2 : The sender (A) now encrypts the message digest with there private key. The output of this process is called the digital signature (DS) of A.



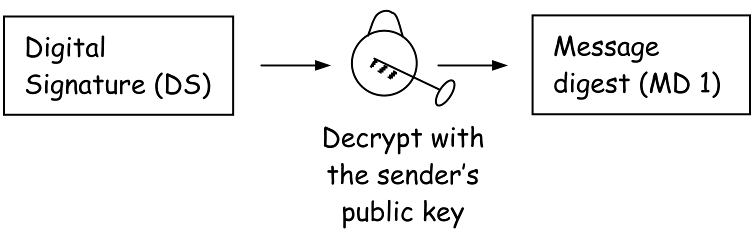
encrypt with the sender's private key

Step-3 : Now the sender (A) sends the original message (M) along with the digital signature (DS) to the receiver (B).



Step-4 : After the receiver (B) receives the original message (M) and the sender's (A's) digital signature, B uses the same message – digest algorithm was used by A and calculates its own message digest (MD 2).

Step-5 : The receiver (B) now uses the sender's (A's) public key to decrypt the digital signature.



Step-6 : If MD 1 = MD 2, the following facts are established :
B accepts the original message (M) as the correct unaltered message. B is also assured that the message come from A and not from some else.

Q.3(b) Explain the working of secure hash algorithm-512. **[5]**

(A) **SHA - 512**

The SHA 512 also takes a message of length 2^{128} bits and produces a message digest of size 512 bits. The input is divided into blocks of size 1024 bits each

Step-1 : Padding bits are added at the end of original message in such a way that the length of message is 128 bits short of a multiples of 1024 bits.

Step-2 : Append length, the length of the message excluding the length of the padding is now calculated and appended to the end of the padding as a 128 bit block. Hence the length of the message is exactly a multiple of 1024 bits.

Step-3 : Divide the input into 1024 bits block. The input message is now divided into blocks each of length 1024 bits. These blocks become the input to the message_digest processing logic.

Step-4 : Initalize chaining variables : Eight chaining variables a, b, c, d, e, f, g, h are initialised.

Step-5 : Process Blocks : Copy the chaining variables A – H into the variables a – h.

SHA– 512 has 4 rounds. Each round takes 20 constants.

Each round consists of

Temp 1 – h + ch(e, f, g) + Sum(ei for 1 to 512) + $W_\ell + K_\ell$

Temp 2 – Sum (a_i for i = 0 to 512) + Maj (a, b, c)

a = Temp 1 + Temp 2
b = a
c = b
d = c
e = d + Temp 1
f = e
g = f
h = g

Q.3(c) Explain the working of HMAC. **[5]**

(A) HMAC stands for Hash-based Message Authentication Code.
The fundamental idea behind HMAC is to reuse the existing message digest algorithms, such as MD5 or SHA-1.

The Working of HMAC

The various variables that will be used in HMAC.

MD	= The message digest/hash function used (e.g. MD5, SHA-1, etc.)
M	= The input message whose MAC is to be calculated

L	= The number of blocks in the message M
b	= The number of bits in each block .
K	= The shared symmetric-key to be used in HMAC
ipad	= A string 00110110 repeated b/8 times
opad	= A string 01011010 repeated b/8 times

Step 1 (Make the length of K equal to b) : The algorithm starts with three possibilities, depending on the length of the key K:

- Length of K< b In this case, need to expand the key (K) to make the length of K equal to the number of bits in the original message block (i.e. b). For this, add as many 0 bits as required to the left of K. For example, if the initial length of K = 170 bits, and b = 512 then add 342 bits, all with a value 0, to the left of K.
- Length of K = b In this case, do not take any action, and proceed to step 2.
- Length of K > b In this case, need to trim K to make the length of K equal to the number of bits in the original message block (i.e. b). For this, pass K through the message-digest algorithm (H) selected for this particular instance of HMAC, which will give a key K, trimmed so that its length is equal to b.

This is shown in figure 1.

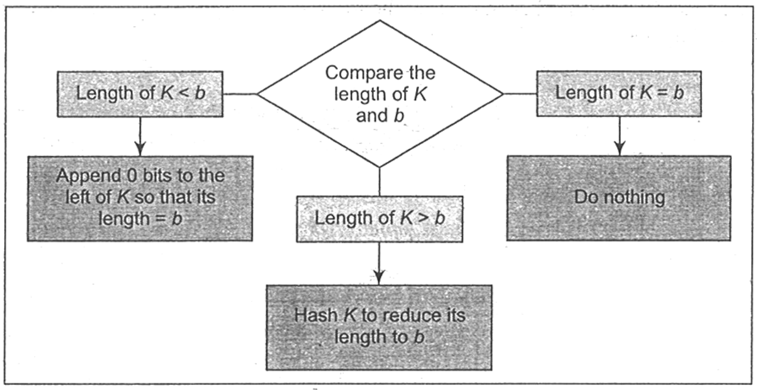

Compare the length of K and b
Length of K < b
Length of K = b
Append 0 bits to the left of K so that its length = b
Length of K > b
Do nothing
Hash K to reduce its length to b

Fig. 1 : Step 1 of HMAC

Step 2 (XOR K **with ipad to produce S1) :** XOR K (the output of step 1) and ipad to produce a variable called S1. This is shown in figure 2.

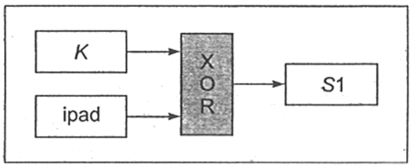


Fig. 2 : Step 2 HMAC

Step 3 (Append M to S1) : Take the original message (M) and simply append it to the end of S1 (which was calculated in step 2). This is shown in figure 3.

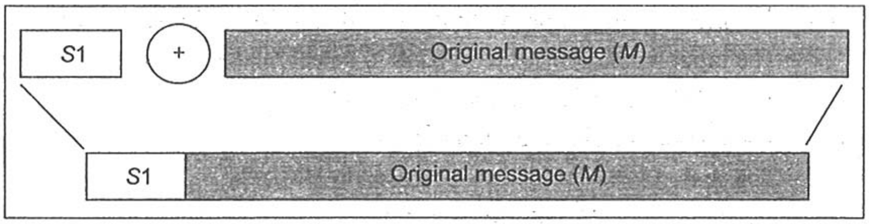


Fig. 3 : Step 3 of HMAC

Step 4 (Message-digest algorithm): Now, the selected message-digest algorithm (e.g.. MD5, SHA-1, etc.) is applied to the output of step 3 (i.e. to the combination of S1 and M). The output of this operation is called H. This is shown in figure 4.

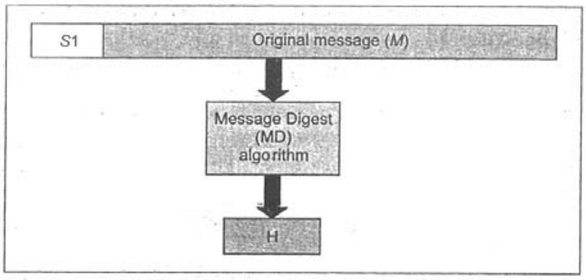


Fig. 4 : Step 4 of HMAC

Step 5 (XOR K with opad to produce S2) : Now, XOR K (the output of step 1) with opad to produce a variable called as .S2. This is shown in figure 5.

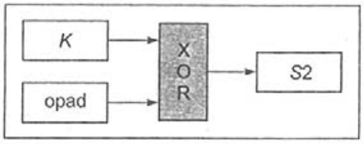


Fig. 5 : Step 5 HMAC

Step 6 (Append H to S2) : In this step, take the message digest calculated in step 4 (i.e. H) and simply append it to the end of S2 (which was calculated in step 5). This is shown in figure 6.

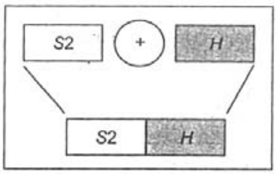


Fig. 6 : Step 6 of HMAC

Step 7 (Message-digest algorithm) : The selected message-digest algorithm (e.g. MD5, SHA-1, etc.) is applied to the output of step 6 (i.e. to the concatenation of S2 and H). This is the final MAC. This is shown in figure 7.

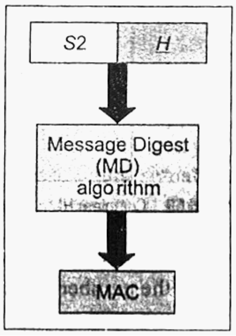


Fig. 7 : Step 7 of HMAC

Summarize the seven steps of HMAC, as shown in figure 8.

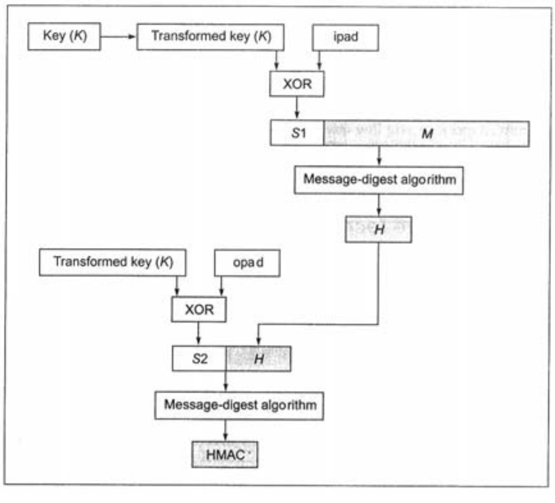


Fig. 8 : Complete HMAC operation

Q.3(d) Explain Elipitic Curve Cryptography and EIGamal. **[5]**

(A) **EIGamal Cryptography :**

EIGamal key generation : Steps

(1) Select a large prime number called P. This is the first part of the encryption key or public key.

(2) Select the decryption key or private key D.

(3) Select second part of the encryption key or public key E1.

(4) The third part of encryption key E2 is computed as E2 = $E1^D$ mod P.

(5) The public key is (E1, E2, P) and the private key is D.

EIGamal Key encryption : Steps

(1) Select a random integer R that fulfills some mathematical properties, which are ignored here.

(2) Compute the first part of the cipher text. C1 = $E1^R$ mod P.

(3) Compute the second part of cipher text C2 = (PT × $E2^R$) mod P, where PT is the plain text.

(4) The final cipher text is (C_1, C_2)

Elliptic Curve Cryptography :

It is an approach to public key cryptography based on the algebraic structure of elliptic curve over finite fields. ECC requires smaller keys compared to non ECC cryptography to provide equivalent security.

Elliptic curves are applicable for encryption, digital signatures, pseudo–random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography.

An elliptic curve is a plane curve over a finite field which consists of points satisfying the equation.

$$y^2 = x^3 + Qx + b$$

along with a distinguished point at infinity denoted.

Q.4 Answer any **TWO** of the following: **[10]**

Q.4(a) Describe of the various fields of X.509v3 digital certificate. **[5]**

(A) Fields of X.509v3 digital certificate :

(1) Authority key identifier – A CA may have multiple private public key pairs. This field defines which of these key pairs is used to sign this certificate.

(2) Subject key identifier – A subject may have multiple private key pairs. This field defines which of those key pairs is used to sign.

(3) Key usage : Defines the scope of operations of the public key of this particular certificate.

(4) Extended key usage : Can be used in addition to or in the place of the key usage field.

(5) Private key usage period : Allows defining different usage period limits for the private and the public keys corresponding to this certificate.

(6) Certificate policies : Defines the polices and optional qualifies information that the CA associates with a given certificate.

(7) Policy mapping : Used only when the subject of a given certificate is also a CA.

(8) Subject alternative name : Optionally defines one or more alternative names for the subject.

(9) Issues alternative name : Optionally define one or more alternative names for the issues.

(10)Subject directory attributes : Can be used to provide additional information about the subject, such as phone no.

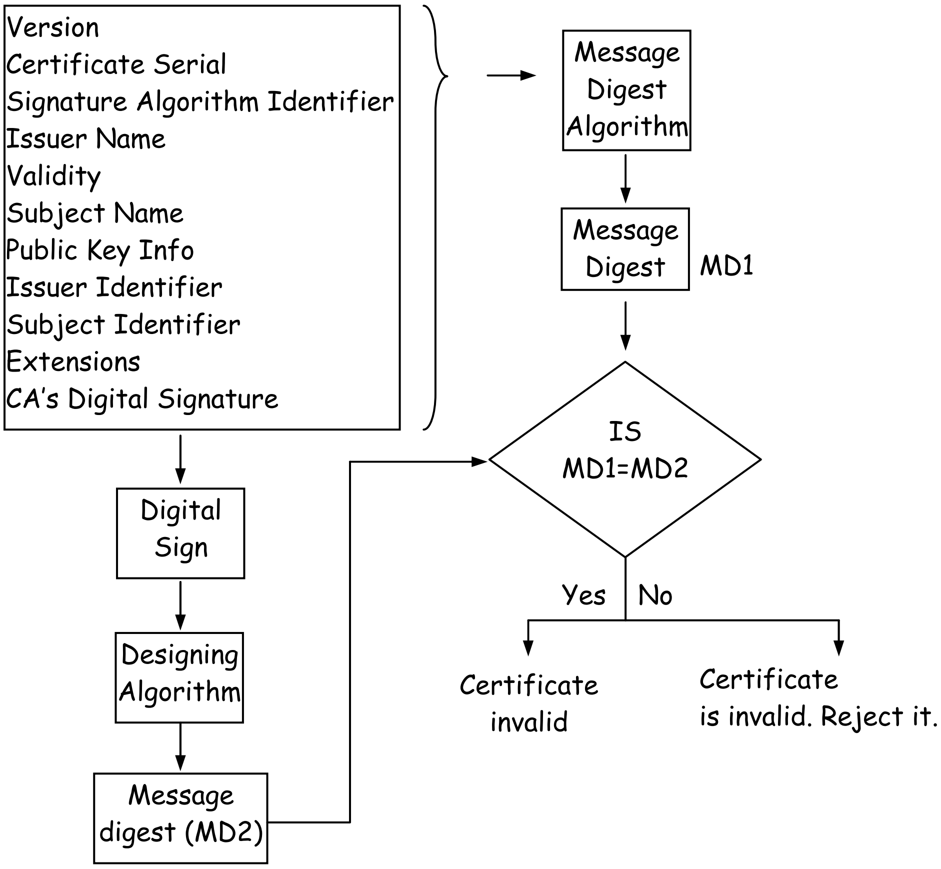
(11) Basic constraints : Indicates whether the subject in this certificate may act as a CA.

(12)Name constraints : Specifies name space.

(13)Policy constraints : Used only for CA's certificates.

Q.4(b) How is digital certificate is verified? Explain. [5]

(A) Having understood how the CA signs a digital certificate, let us now think how the verification of a certificate takes places. Suppose we receive a digital certificate of a user, which we want to verify. Clearly, we need to verify the digital signature of the CA.

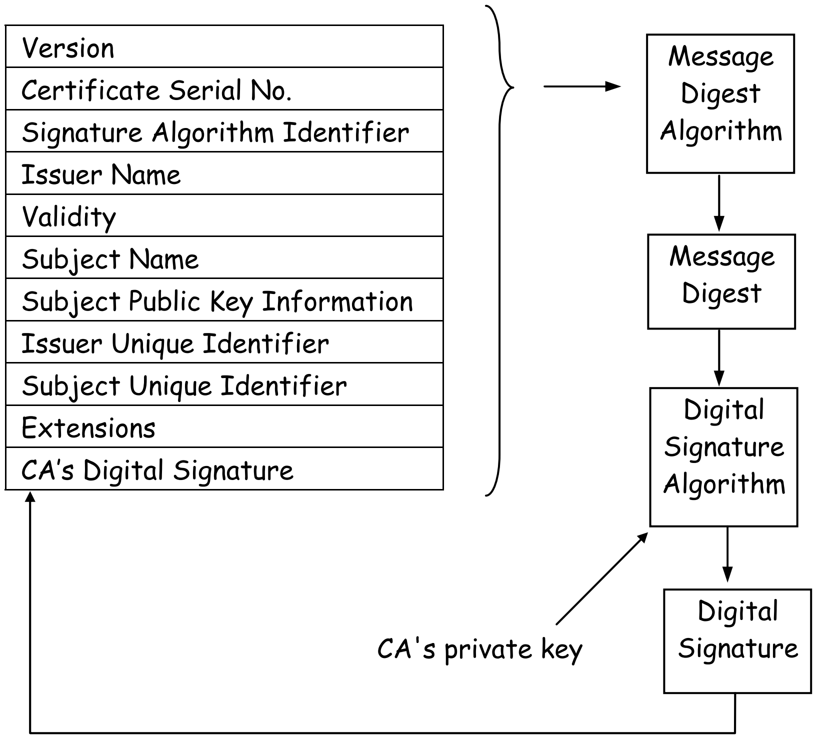


(a) The user passes all field except the last one of the received digital certificate to a message–digest algorithm.
(b) The message–digest algorithm calculates a message digest of all fields of the certificate called MD1.
(c) The user now extracts the digital signature of the CA from the certificate.
(d) The user de–signs the CA's signature (i.e. the user decrypts the signature with the CA's public key)
(e) This produces another message digest, which we called MD2.
(f) Now, the user compares the message digest it calculated (MD1) with the one, which is the result of designing the CA's signature (MD2). If the two match i.e. if MD1 = MD2, the user is convinced that the digital certificate was indeed signed by the CA with its private key. If this comparison fails, the user will not trust the certificate, reject it.

Q.4(c) Why do we trust digital certificate? [5]

(A) CA always signs a digital certificate with its private key. so, we can trust digital certificate. How CA signs a certificate :

The last field in a digital certificate is always the digital signature of the CA. That is every digital certificate not only contain the user's information (such as subject name, public key etc.) but also the CA's digital signature. Like passport therefore a digital certificate is always signed or certified.



Q.4(d) List and explain public key cryptography standards. **[5]**

(A) **PKCS # 1** : RSA encryption standard : defines the basic formatting rules of RSA public key function, more specifically the digital signature. It defines how digital signature should be calculated including the structure of the data to be signed as well as the format of the signature.

PKCS # 2 : RSA Encryption Standard for Message Digest : This standard outlined the message–digest calculation.

PKCS # 3 : Diffie-Hellmen key agreement standard. Defines a mechanism to implement Diffie–Hellman key agreement protocol.

PKCS # 4 : NA merged with Public key cryptograph 1.

PKCS # 5 : Password based Encryption (PBE) describes a method for encrypting an octet string with a symmetric key. The symmetric key is derived from a password.

PKCS # 6 : Extended certificate syntax Standard : Defines syntax for extending the basic attributes of an α : 509 digital certificate.

PKCS # 7 : Cryptographic Message Syntax Standard : Specifies a format for data that is the result of a cryptographic operation.

PKCS # 8 : Private Key Information Standard : Describes the syntax for private key information (i.e. the algorithm and attributes used to generate the private key.

PKCS # 9 : Selected attribute type : Defines selected attribute types for use in PKCS # 6 extended certificates.

PKCS # 10 : Certificate request syntax standard : Defines syntax for requesting for digital certificates.

PKCS # 11 : Cryptographic token interface standard : This standard also called cryptoki specifies on ABI for the single user devices that contain cryptographic information.

PKCS # 12 : Personal information exchange syntax standard : Defines syntax for personal identity information.

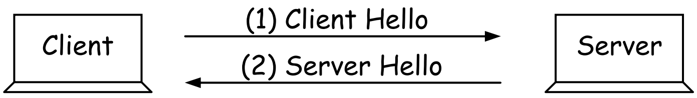
Q.5 Answer any **TWO** of the following: **[10]**

Q.5(a) Explain the handshake protocol. **[5]**

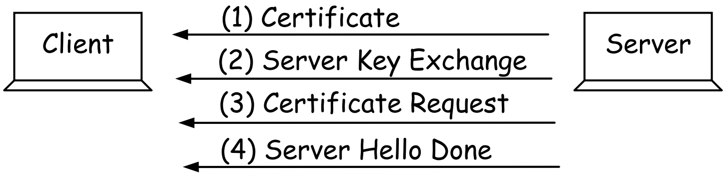
(A) The Handshake Protocol :

The Handshake Protocol is actually made up to four phases :

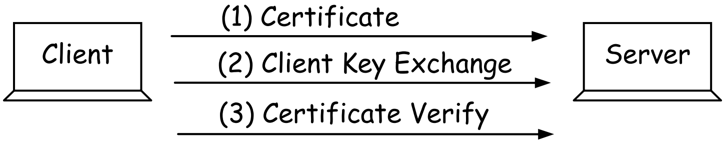
(1) Establish security capabilities : In this phase of the SSL handshake is used to initiate a logical connection and establish the security capabilities associated with that connection. This consists of the client hello and the server hello.



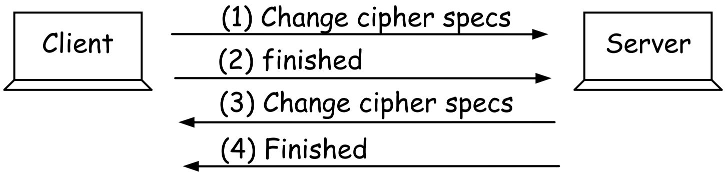
(2) Server Authentication and Key Exchange : The server initiates this second phase of SSL and sends message to the client. The client is sole receiver. This phase contains four steps.



(3) Client Authentication and key exchange : The client initiates this third phase of SSL handshake and is the hole sender of all the messages in this phase. The server is the sole receipent of all these message. This phase contains three steps :
Certificate, Client key exchange, Certificate verify.



(4) Finish : The client initiates the fourth phase of the SSL handshake, which the server ends. The client sends change cipher specs, finished. The server responds back with two identical message : change cipher specs, finished.



Q.5(b) Explain the Secure Electronic Transaction process. **[5]**

(A) The SET process :

(1) The customer opens an account : The customer opens an credit card accenent with a bank that support electronic payment mechanism and the SET protocol.

(2) The customer receives a certificate : After the customer's identify is verified, the customer receives a digital certificate from a CA.

(3) The merchant receives a certificate : A merchant that wants to accept a certain brand of credit cards must posses digital certificate.

(4) The customer places an order : This is a typical shopping card process where in the customer browser the list of items available search for specific item.

(5) The merchant is verified : The merchant also sends its digital certificate to the customer. This assures that the customer is dealing with the valid merchant.

(6) The order and payment details are sent : The customer sends both order and payment details to the merchant along with the customer's digital certificate.

(7) The merchant request payment antherization : The merchant forwards the payment details sent by the customer to the payment gateway via the acquires and request the payment gateway to authorize the payment.

(8) The payment gateway authorizes the payment : Using the credit card information received from the merchant, the payment gateway verifies the details of the customer's credit card with the help of the issues.

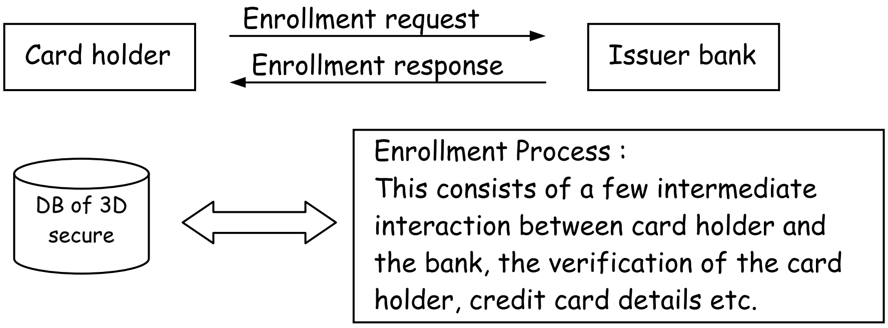
(9) The merchant confirms the order : Assuming that the payment gateway authorises the payment, the merchant sends a confirmation of the order to the customer.

(10) The merchant provides goods and services to the customer.

(11) The merchant request payment : The payment gateway receives a request from the merchant for making the payment.

Q.5(c) With neat diagram write the internal operaitons of 3-D secure protocol. **[5]**

(A) A new protocol developed by visa has emerged called 3 D secure.
To process is shown in figure :

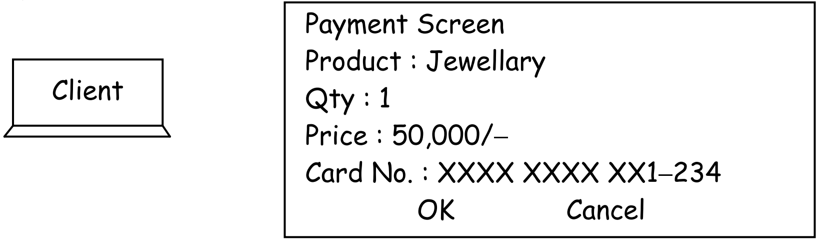


Enrollment Process :
This consists of a few intermediate interaction between card holder and the bank, the verification of the card holder, credit card details etc.

An entry for this card holders goes into this controlled users DB maintained by the issuer bank.

In 3–D secure protocol before a card holder makes a card payment be must enroll with the issuer bank's enrollment server. At the time of an actual 3–D secure transaction, when a merchant receives a payment instruction from the card holder, the merchant forwards this request to the issuer bank through the visa network. The issuer bank requires the card holder to provide the user id and password that were created at the time of user enrollment process. The card holder provides these details which the issues bank verifies against its 3–D secure enrolled user database. Only after the user is authenticated successfully does the issuer bank inform the merchant that it can accept the card payment instruction.

Step-1 : The user shops using the shopping cart on the merchant site and decides to pay the amount. The user enters the credit card details for this purpose and clicks on OK butter.

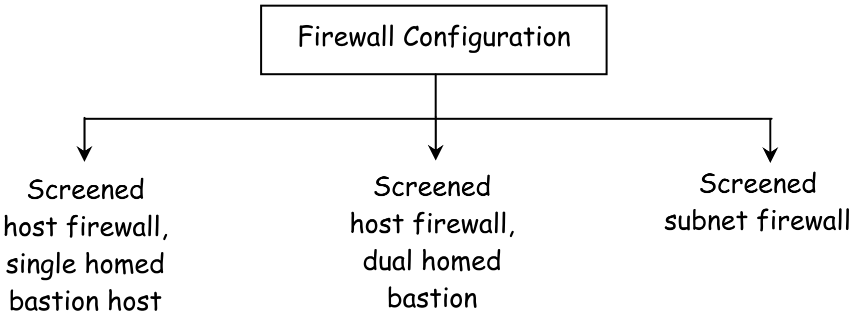


Payment Screen
Product : Jewellary
Qty : 1
Price : 50,000/–
Card No. : XXXX XXXX XX1–234
OK Cancel

Step-2 : When the user click on the OK button, the user will be redirected to the issuer bank site. The bank site will produce a pop up screen, prompting the user to enter password. Bank verifies the password and sends message.

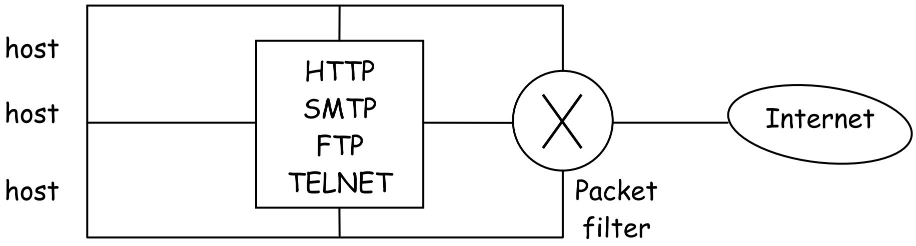
Q.5(d) List the different firewall configuraitons. Explain any two. **[5]**

(A)

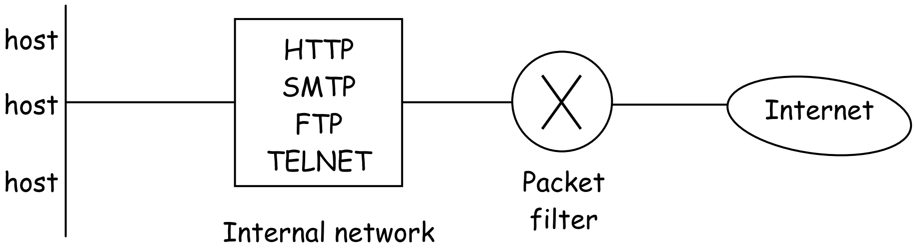

Firewall Configuration
Screened host firewall, single homed bastion host
Screened host firewall, dual homed bastion
Screened subnet firewall

(1) Screened host firewall single homes bastion : In this the firewall set up consists of two parts a packet filtering router and an application gateway. The packet filter ensures that the incoming traffic is allowed only it is destined for the application gateway by examining the destination address field of every incoming IP packet. Similarly it also ensures that the outgoing traffic is allowed only if it is originating from the application gateway.

The application gateway performs authentication and proxy functions.



Screened host firewall, duel homes bastion : Direct connections between the internet host and the packet filter are avoided. The packet filters connects only to the application gateway, which has a separate connection with the internet host.



Q.6 Answer any **TWO** of the following: **[10]**

Q.6(a) How does clear text password work? What are the problems with it? [5]

(A) Working of clear text password :

(1) Step-1 : Prompt for user Id and password : During authentication, the application sends a screen to the user, prompting for user Id and password.

(2) Step-2 : User enters Id and password : User enter Id and password and presses the OK button. This case the user Id and password to travel in clear text to the server.

(3) Step-3 : User Id and password validation : Server consults the user database to see if this particular user Id and password combination exists lase.

(4) Step-4 : Authentication result : Depending on the success or failure of the validation of the user Id and the password, the user–authentication program returns an appropriate result back to the server.

(5) Step-5 : Inform user accordingly : Depending on the outcome, the server sends back an appropriate message to the user. If the user authentication was successful. the server typically sends a menu of options for the user, which lists the actions the user is allowed to perform. If the result of the user authentication was a failure the server sends an error screen to the user.

Problems with the scheme :

(1) Database contains password in clear text : The user database contains user Id and passwords in clear text. Therefore, if an attacks succeeds in obtaining an access to the database, the whole list of user Id and password in available attackers.

(2) Password travels in clear text from the user computer to the server : Even if we store encrypted password in the database, the password would travel in clear text from the user to the server. If an attacker access the link, the attacker can easily obtain the clear text password.

Q.6(b) Write a short note on key distribution center. [5]

(A) Key distribution center (KDC) : It is a central authority dealing with keys for individual computer in a computer network. It is similar to the concept of the Authentication Server (AS) and Ticket Granting Server (TGS) in Kerberos. The basic idea is that every node shares unique secret key with the KDC. Whenever user A wants to communicate securely with user B, the following happens :

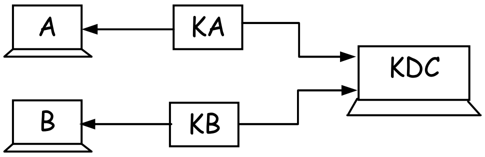
(1) The background is that A has a shared key KA with KDC. Similarly, B is assumed to share a secret key KB with the KDC.

(2) A sends a request to KDC encrypted with KA, which includes
a) Identifies of A and B. b) A random number R.

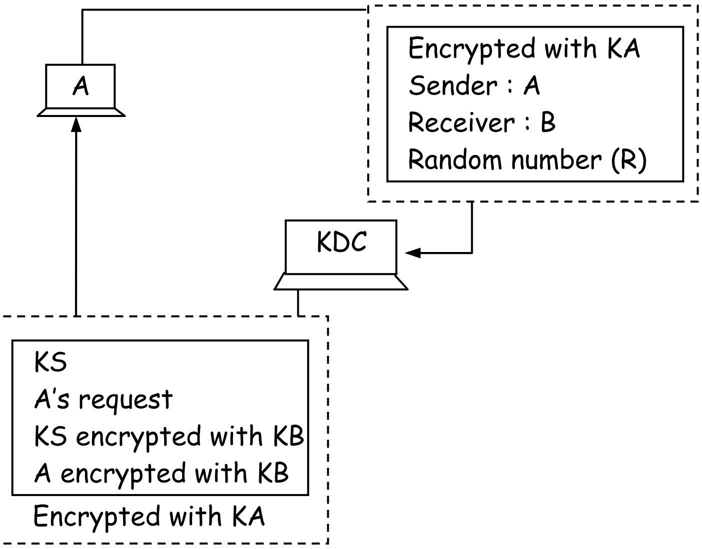
(3) KDC responds with a message encrypted with KA, containing.
a) One time symmetric key K.S.
b) Original request that was sent by a for verification.
c) Plus, KS encrypted with KB and ID of A encrypted with KB.

(4) A and B can now communicate by using KS for encryption.

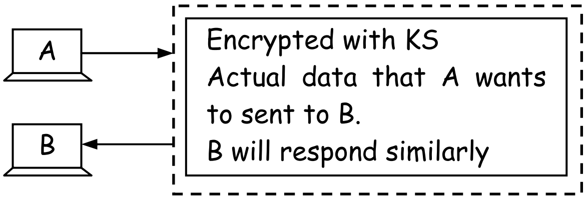
Step – 1 :



Step – 2 :



Step – 3 :

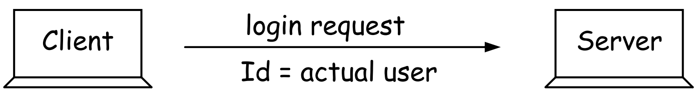


Q.6(c) Explain the working of how challenge/response tokens. [5]

(A) Challenge/response tokens :

This idea used in the challenge response tokens is actually a combination of the techniques, we have used. The seed pre-programmed inside an authentication token is secret and unique. This fact is the basis for the challenge / response token.

Step-1 : User sends a login request : In this technique, the user sends the login request only with his/her user Id (and not the one–time password)



Step-2 : Server creates a Random Challenge : When the server receives the user's login request containing the user Id alone, it first checks to see if the user Id is a valid one. If it is not, it sends an appropriate error message back to the user. If the user Id is valid, the server now creates a random challenge and sends it back to the user.

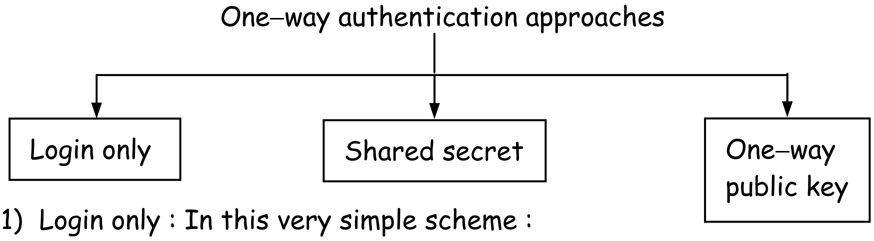
Step-3 : User signs the Random Challenge with message digest of the password.
The user gets a screen, which displays the user Id, the random challenge received from the server and data entry field, with the label password. The user reads the value and types it in the screen. This request is then sent to the server as login request.

Step-4 : Server verifies the encrypted Random Challenge received from the user : The server receives the random challenge, which was encrypted with the seed by the user's authentication taken. The server can decrypt the encrypted random challenge received from the user with the seed value for the server.
Alternatively, the server can simply encrypt it own version of the random challenge (i.e. the one which was sent earlier to the user) with the seed for the user.

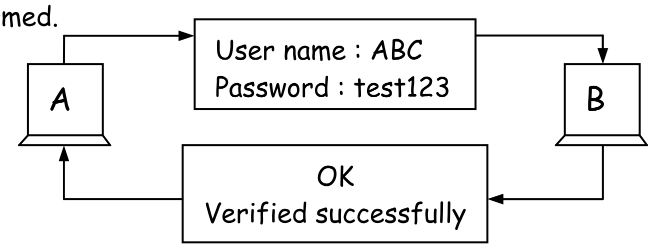
Step-5 : Server returns an appropriate message back to the user : finally the server sends an appropriate message back to the user, depending on whether there is success or failure.

Q.6(d) What are the One-way authenticaiton approaches? Explain any two. **[5]**

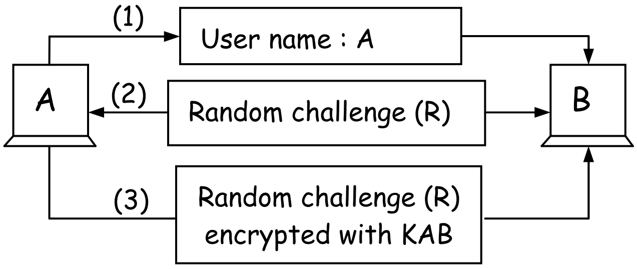
(A)



(1) Login only : In this very simple scheme :
  a) User A sends her user name and password in the plain text from to the other user B.
  b) B verifies the user name and password. If the user name and password are correct, communication starts happening between A and B. No further encryption or integrity checks are performed.



(2) Shared secret : there is an assumption that A and B have agreed on a shared symmetric key KAB, before the actual communication begins. Hence, we have the name shared secret for this approach. The protocol then works as follows :
  a) A sends her user name and password to B.
  b) B creates a random challenge R and sends it to A.
  c) A encrypts the random challenge (R) with the shared symmetric key between A and B (R) and sends the encrypted R to B. B also encrypts the original random challenge (R) with the same shared symmetric key (R). If this encrypted challenge matches with the one sent by A, B consider A to be authentic.



Q.7 Attempt any **THREE** of the following : **[15]**

Q.7(a) Write a short note on phishing. **[5]**

(A) Attackers set up fake websites, which look like real web sites. It is quite simple to do as creating a website using HTML, Java script, CSS etc. Learning and using these technology is quite simple. The attackers modus operandly works as follows :

(1) The attacker decides to create his/her own website, which looks very identical to the real website. For example, the attacker can clone ICICI bank website. The cloning such that human eye will not be able to distinguish between the real and fake site.

(2) The attacker can use many technique to attack the bank customers. The attacker sends an e–mail to the legitimate customer of the bank. The e–mail itself appears to have come from the bank. For ensuring, this the customer e–mail database is hacked. This fake e–mail warns the user that there has been some sort of attack on ICICI bank computer systems and the bank wants to issue new passwords to all its customers or verify there existing pin. For this purpose, the customer is asked to visit a URL mentioned in the e–mail.

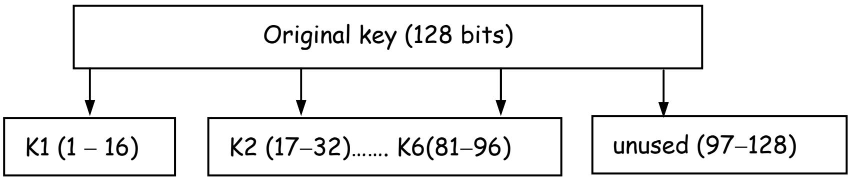
(3) When the customer innocently clicks on the URL specified in the e–mail, he/she is taken to the attacker's site and not the bank's original site. There the customer is prompted to enter confidential information such as password or pin. Since the attackers fake site looks exactly similar to the bank site, the customer provides all the information. The attacker gladly accepts this information and displays a "Thank you" message to the victim.

The attacker now user the victim's password or Pin to access the bank's real site and can perform any transaction as is he/she is the customer of the bank.

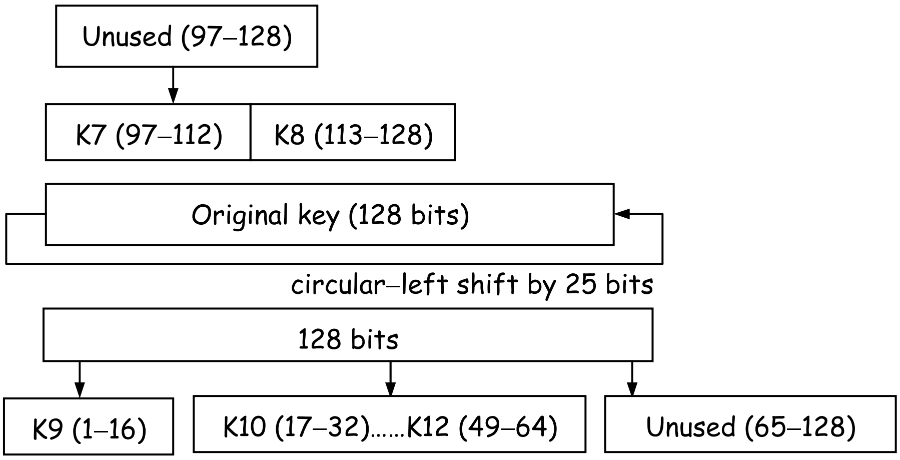
Q.7(b) Explain subkey generaiton process of each round of international data encryption algorithm. **[5]**

(A) Sub–key Generation of IDEA :

(1) First Round : Initial key consists of 128 bits, from which 6 sub–keys K1 to K6 are generated for the first round. Since K1 to K6 consists of 16 bits each of the original 128 bits, the first 96 bits (6 sub–key × 16 bits per sub–key) are used in the first round. Thus at the end of first round, bits 97 to 128 of the original key are unused.



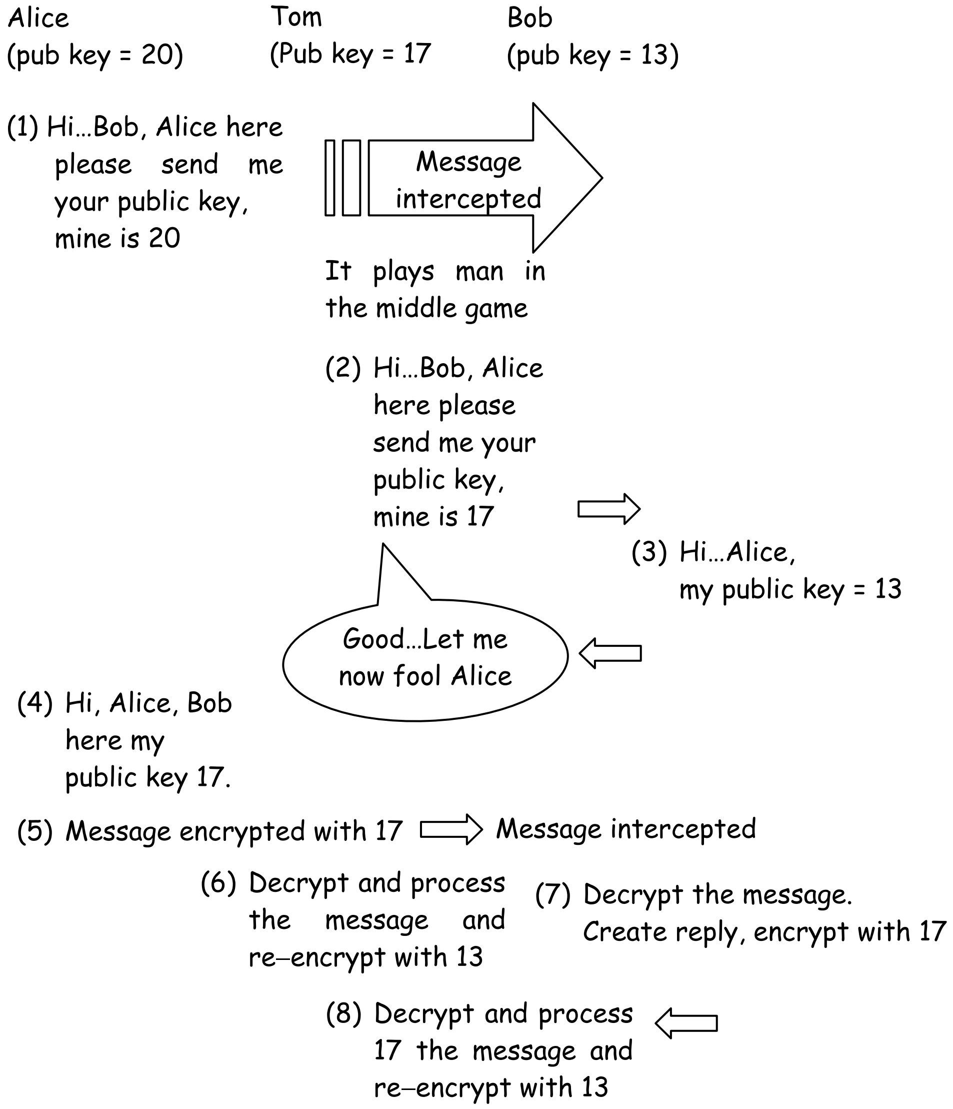
(2) Second Round : For the second round, we can utilize the 32 unused key bits at position 97 to 128 (which would give us two sub–keys, each of 16 bits). IDEA employs the technique of key shifting. The original key is shifted left circularly by 25 bits. That is, the 26th bit of the original key moves to the first position and becomes the first bit after the shift and the 25th bit of the original key moves to the last position and becomes the 128th bit after the shift. The whole process is shown in figure.



The unused bits of the second round will firstly be used in round 3 and then a circular–left shift of 25 bits will be performed once again. From the resulted we would extract the first 32 bits which covers the short fall is the key bits for this round.

Q.7(c) Explain with a neat diagram the man-in-middle attack. [5]

(A) Man–in–middle attack :



Alice (pub key = 20)

Tom (Pub key = 17

Bob (pub key = 13)

(1) Hi…Bob, Alice here please send me your public key, mine is 20

Message intercepted

It plays man in the middle game

(2) Hi…Bob, Alice here please send me your public key, mine is 17

(3) Hi…Alice, my public key = 13

Good…Let me now fool Alice

(4) Hi, Alice, Bob here my public key 17.

(5) Message encrypted with 17

Message intercepted

(6) Decrypt and process the message and re–encrypt with 13

(7) Decrypt the message. Create reply, encrypt with 17

(8) Decrypt and process 17 the message and re–encrypt with 13

Steps :

1) When Alice wants to send a message security to Bob, she sends Bob her public key (20) and asks Bob for Bob's public key.
2) Tom–the attacker intercepts Alice's message. He changes the public key value in Alice's original message from 20 to his own (17) and forwards this message to Bob.
3) Bob sends back his public key (13) in response to Alice's message.
4) Tom intercepts Bob's message, changes the public key value to 17 and forwards it to Alice.
5) Alice thinks that Bob's public key is 17. Therefore, she encrypts the confidential message to be sent to Bob with 17 and sends to Bob.
6) Tom intercepts this message, uses his private key to decrypt the message, processes it re-encrypt's the message with Bob's public key (13).
7) Bob decrypts the message coming from Tom with private key.

Q.7(d) Why is a self-signed certificate needed? **[5]**

(A) Self signed Certificate :

The root CA (and main times, even the second or the third level (A's) are automatically considered trusted CA's. For this, Alice's software (usually the web browser, but this can be any other piece of software that is above to store and verify certificates) contains a pre-programmed, hard–coded certificate of the root CA. Also, this certificate of the root CA is a self-signed certificate i.e. the root CA signs its own certificate. The self-signed certificate is shown in figure.

Digital Certificate

Issuer Name : Root
Subject Name : Root

The issuer name and the subject name both point to the root CA in this certificate.

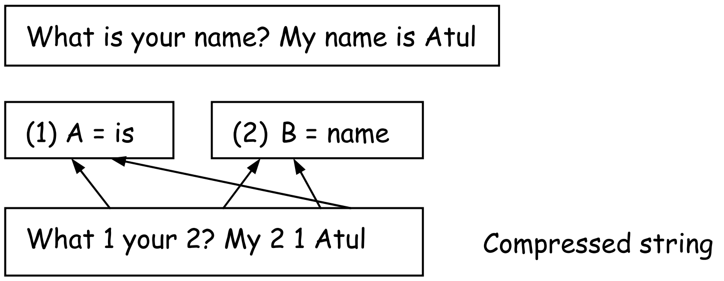
Since this certificate comes as a part of the basic software such as a web browser or web server. Alice need not worry about the authenticity of the root certificate, unless the basic software that she is using itself comes from an untrusted site. As long as, Alice restricts herself to industry standard, well accepted software applications (i.e. usually web browsers and web servers), she can be confident about the validity of the root CA's certificate.

Q.7(e) Explain the working of how pretty good privacy. **[5]**

(A) Working of PGP :

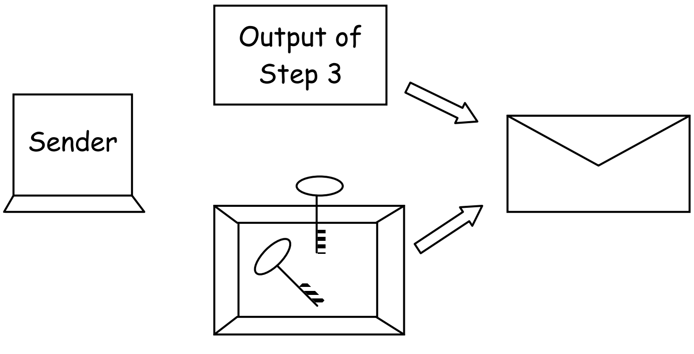
(1) Step-1 : Digital signature : It consists of the creation of a message digest of the e–mail message using the SHA-1 algorithm. The resulting message digest is then encrypted with the sender's private key.

(2) Step-2 : Compression : The input message as well as the digital signature are compressed together to reduce the size of the final message the will be transmitted. Z IP program is used.


What is your name? My name is Atul
(1) A = is
(2) B = name
What 1 your 2? My 2 1 Atul
Compressed string

(3) Step-3 : Encryption : The compressed output of step – 2 are encrypted with a symmetric key. For this, generally the IDEA algorithm is CFB mode is used.

(4) Step-4 : Digital enveloping : In this case, the symmetric key used for encryption in step 3 is now encrypted with the receivers public key. The output of step 3 and step 4 together form a digital envelope as shown figure :



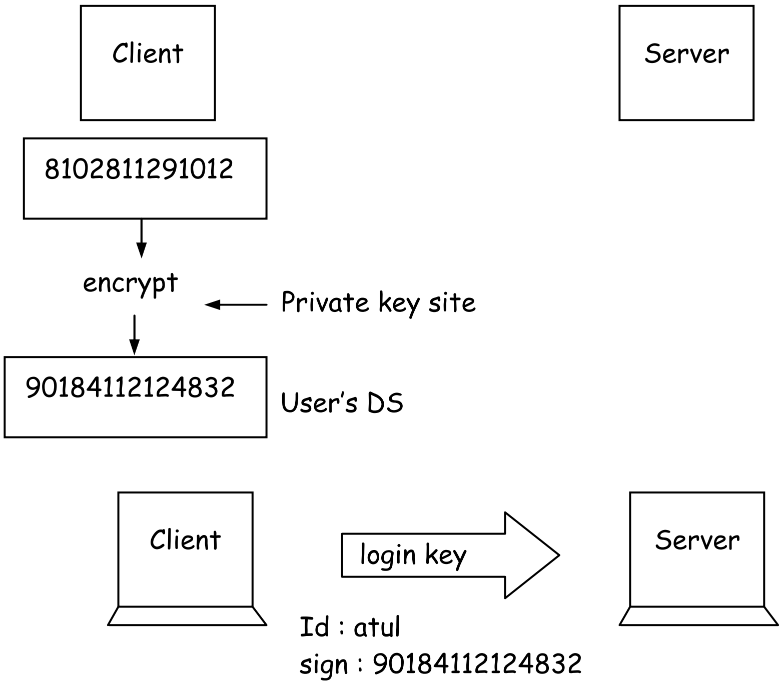
Symmetric key encrypted with the receiver's public key

(5) **Step-5 :** Base 64 encoding : The base 64 encoding process transforms arbitrary binary input into printable character output. In this technique, the binary input is processed in blocks of 3 octets or 24 bits. There are considered to be made up of 4 sets. Each such set of 6 bits is mapped into an 8 bit output character in this process.

Q.7(f) How does certificate-based authentication work? Explain. [5]

(A) Working of Certificate–based Authentication :

1) Step – 1 : Creation, storage and distribution of digital certificates : The first step in certificate–based authentication is actually a pre–requiste. Here the digital certificates are created by the CA.
2) Step – 2 : Log in request : During the login request, the user sends only his/her user Id to the server.
3) Step – 3 : Server creates a random challenges : The server employs the techniques that when the server receives the user's login request, it first checks to see if the user Id is a valid one. If it is not, it sends an appropriate error message back to the user.
4) Step – 4 : User signs the Random Challenge : The user has to now sign the random challenge with his private key. This private key corresponding to the user's public key.



5) Step – 5 : Server returns an appropriate message back to the user : Finally the server sends an appropriate message back to the user depending on whether the previous operations yielded success or failure.

❑ ❑ ❑ ❑ ❑