USN | 1 | C | R | 2 | 1 | I | S | | |

Internal Assessment Test II- Nov. 2024

Sub:	Cryptography and Network Security				Sub Code:	21IS71	Branch:	ISE
Date:	/11/2024	Duration:	90 mins	Max Marks: 50	Sem/ Sec:	VII/ A, B, C		OBE

	Answer any FIVE FULL questions	MARKS	CO	RBT
1	Explain with neat diagram control vector encryption and decryption.	[10]	CO3	L2
2	Explain Elgamal cryptosystem. Perform encryption and decryption using q = 19, a = 10, k = 6, M = 17, XA= 5 and YA = 3.	[10]	CO2	L3
3	Explain how symmetric key distribution works using symmetric encryption.	[10]	CO3	L2
4	Describe a typical key distribution scenario using a Key Distribution Center (KDC)	[10]	CO3	L2
5	Explain various techniques proposed for the distribution of public keys.	[10]	CO3	L2
6	a. How does the Diffie-Hellman algorithm work?	[6]	CO2	L2
	b. How can Diffie-Hellman mitigate Man-in-the-Middle attacks?	[4]	CO2	L2

CI **CCI** **HOD**

USN | 1 | C | R | 2 | 1 | I | S | | |

Internal Assessment Test II- Nov. 2024

Sub:	Cryptography and Network Security				Sub Code:	21IS71	Branch:	ISE
Date:	/11/2024	Duration:	90 mins	Max Marks: 50	Sem/ Sec:	VII/ A, B, C		OBE

	Answer any FIVE FULL questions	MARKS	CO	RBT
1	Explain with neat diagram control vector encryption and decryption.	[10]	CO3	L2
2	Explain Elgamal cryptosystem. Perform encryption and decryption using q = 19, a = 10, k = 6, M = 17, XA= 5 and YA = 3.	[10]	CO2	L3
3	Explain how symmetric key distribution works using symmetric encryption.	[10]	CO3	L2
4	Describe a typical key distribution scenario using a Key Distribution Center (KDC)	[10]	CO3	L2
5	Explain various techniques proposed for the distribution of public keys.	[10]	CO3	L2
6	a. How does the Diffie-Hellman algorithm work?	[6]	CO2	L2
	b. How can Diffie-Hellman mitigate Man-in-the-Middle attacks?	[4]	CO2	L2

CI **CCI** **HOD**

21IS71: Internal Assessment Test 2- November.2024

SCHEME & SOLUTION

Sub:	**Cryptography and Network Security**	**Sub Code:**	**21IS71**	**Branch:**	**ISE**	
	Answer any FIVE FULL questions			MARKS	CO	RBT
1	**Explain with neat diagram control vector encryption and decryption.**			2+4+4 [10]	CO3	L2

In this scheme, each session key has an associated control vector consisting of a number of fields that specify the uses and restrictions for that session key. The length of the control vector may vary.

The control vector is cryptographically coupled with the key at the time of key generation at the KDC. As a first step, the control vector is passed through a hash function that produces a value whose length is equal to the encryption key length. In essence, a hash function maps values from a larger range into a smaller range with a reasonably uniform spread. Thus, for example, if numbers in the range 1 to 100 are hashed into numbers in the range 1 to 10, approximately 10% of the source values should map into each of the target values. The hash value is then XORed with the master key to produce an output that is used as the key input for encrypting the session key. Thus,

Hash value = H = h(CV)

$$\text{Key input} = K_m \oplus H$$

$$\text{Ciphertext} = E([K_m \oplus H], K_s)$$

where is the master key and is the session key. The session key is recovered in plaintext by the reverse operation:

$$D([K_m \oplus H], E([K_m \oplus H], K_s))$$

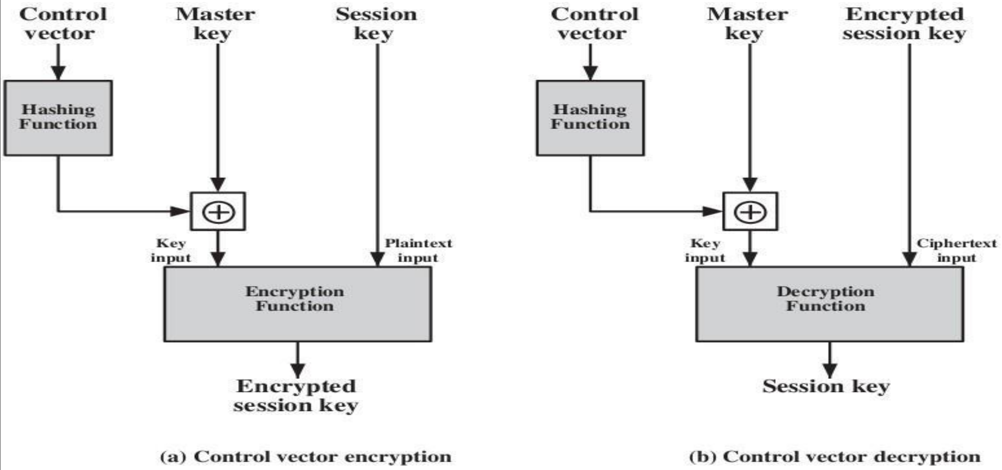


Figure 14.6 Control Vector Encryption and Decryption

When a session key is delivered to a user from the KDC, it is accompanied

	by the control vector in clear form. The session key can be recovered only by using both the master key that the user shares with the KDC and the control vector. Thus, the linkage between the session key and its control vector is maintained. Use of the control vector has two **advantages over use of an 8-bit tag.** ➢ First, there is no restriction on length of the control vector, which enables arbitrarily complex controls to be imposed on key use. ➢ Second, the control vector is available in clear form at all stages of operation. Thus, control of key use can be exercised in multiple locations.			
2	**Explain Elgamal cryptosystem. Perform encryption and decryption using q = 19, a = 10, k = 6, M = 17, XA= 5 and YA = 3.** Elgamal cryptosystem description. **Encryption:** K=7, k=6,C1=11,C2=5 Ciphertext= (11,5) **Decryption:** k=7, K inverse= 11 M=17	4+6 [10]	CO2	L3
3	**Explain how symmetric key distribution works using symmetric encryption.** For **symmetric** encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. Therefore, the term that refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key. For two parties A and B, key distribution can be achieved in a number of ways, as follows: 1. A can select a key and physically deliver it to B. 2. A third party can select the key and physically deliver it to A and B. 3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key. 4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B. Physical delivery (1 & 2) is simplest - but only applicable when there is personal contact between recipient and key issuer. This is fine for link encryption where devices & keys occur in pairs, but does not scale as number of parties who wish to communicate grows. 3 is mostly based on 1 or 2 occurring first. A third party, whom all parties trust, can be used as a **trusted intermediary** to mediate the establishment of secure communications between them (4). Must trust intermediary not to abuse the knowledge of all session keys. As number of parties grow, some variant of 4 is only practical solution to the huge growth in number of keys potentially needed.	5+5 [10]	CO3	L2

21IS71: Internal Assessment Test 2- November.2024

SCHEME & SOLUTION

4	**Describe a typical key distribution scenario using a Key Distribution Center (KDC).**	5+5 [10]	CO3	L2

Key distribution centre:

The use of a **key distribution center** is based on the use of a hierarchy of keys. At a minimum, two levels of keys are used.

Communication between end systems is encrypted using a temporary key, often referred to as a **Session key**.

Typically, the session key is used for the duration of a logical connection and then discarded

Master key is shared by the key distribution center and an end system or user and used to encrypt the session key.

Key Distribution Scenario:

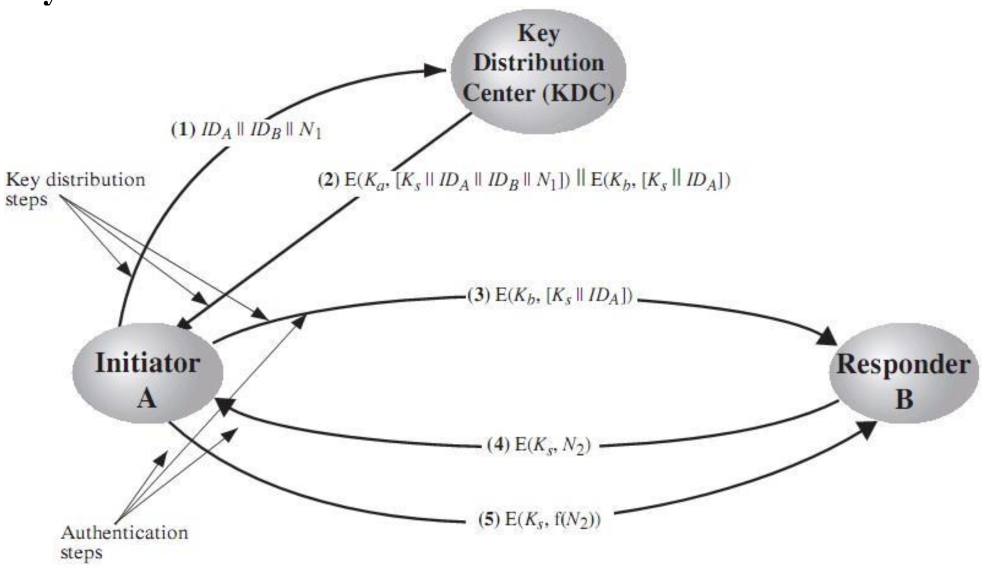


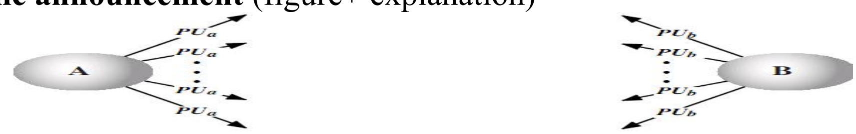
Figure 14.3 Key Distribution Scenario

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

1 A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier, N_1, for this transaction, which we refer to as a **nonce**. The nonce may be a timestamp, a counter, or a random number; the minimum equirement is **that it differs with each request.** Also, to prevent masquerade, it should be difficult for an opponent to guess the nonce. Thus, a random number is a good choice for a nonce.

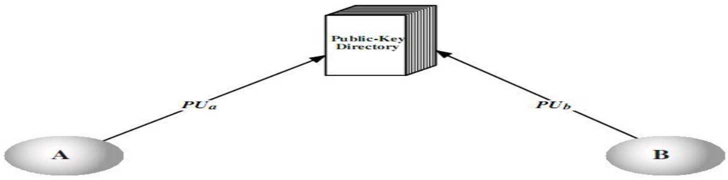
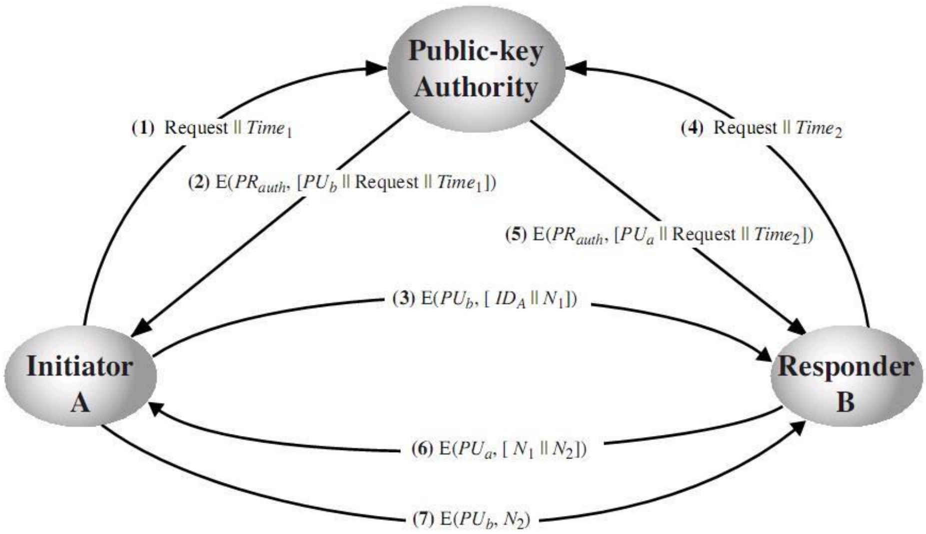
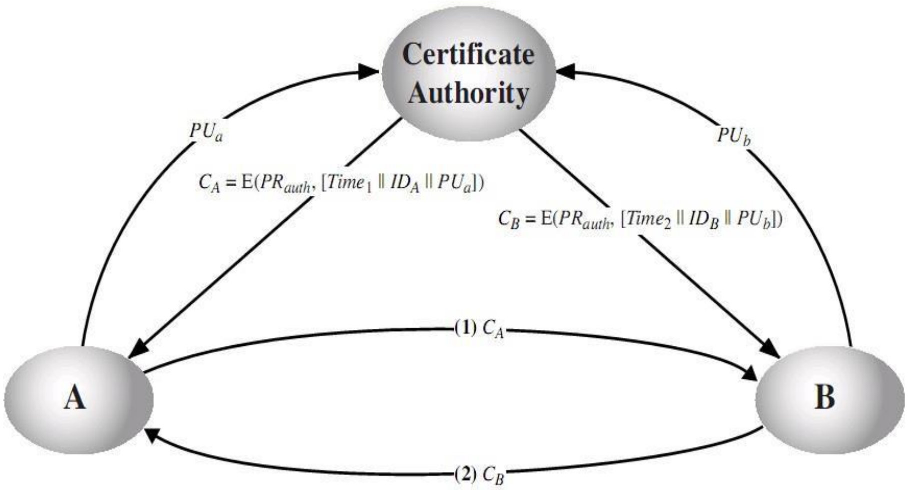
2. The KDC responds with a message encrypted using Ka Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC. The message includes two items intended for A:

The **one-time session key, Ks,** to be used for the session

	The **original request message,** including the nonce, to enable A to match this response with the appropriate request Thus, A can verify that its original request was not altered before reception by the KDC and, because of the nonce, that this is not a replay of some previous request. In addition, the message includes two items intended for B: The one-time session key, Ks to be used for the session An identifier of A (e.g., its network address), IDA These last two items are encrypted with Kb (the master key that the KDC shares with B). They are to be sent to B to establish the connection and prove A's identity. **3.** A stores the session key for use in the upcoming session and forwards to B the information that originated at the KDC for B, namely, $E(K_b, [K_s \| ID_A])$. Because this information is encrypted with K_b, it is protected from eavesdropping. B now knows the session key (K_s), knows that the other party is A (from ID_A), and knows that the information originated at the KDC (because it is encrypted using K_b). At this point, a session key has been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable: **4.** Using the newly minted session key for encryption, B sends a nonce, N_2, to A. **5.** Also using K_s, A responds with $f(N_2)$, where f is a function that performs some transformation on N_2 (e.g., adding one). These steps assure B that the original message it received (step 3) was not a replay. Note that the actual key distribution involves only steps 1 through 3 but that steps 4 and 5, as well as 3, perform an authentication function.			
5	**Explain various techniques proposed for the distribution of public keys.** Distribution of Public Keys: Several techniques have been proposed for the distribution of public keys, which can mostly be grouped into the categories shown. Public announcement Publicly available directory Public-key authority Public-key certificates **Public announcement** (figure+ explanation)  Figure 10.1 Uncontrolled Public Key Distribution **Publicly available directory** (figure+ explanation)	2.5 x 4 [10]	CO3	L2

21IS71: Internal Assessment Test 2- November.2024

SCHEME & SOLUTION

	 **Figure 10.2 Public Key Publication** **Public-key authority** (figure+explanation)  **Public-key certificates** (figure+explanation) 			
6	**How does the Diffie-Hellman algorithm work?** **Algorithm for Diffie-Hellman Key Exchange:** Step 1 two public known numbers q, α q= Prime number α= primitive root of q and α< q. Step 2 if A & B users wish to exchange a key	3+3[06]	CO2	L2

21IS71: Internal Assessment Test 2- November.2024

SCHEME & SOLUTION

a) User A select a random integer X_A<q and computes $Y_A = \alpha^{X_A} \bmod q$

b) User B independently select a random integer X_B <q and computes $Y_B = \alpha^{X_B} \bmod q$

c) Each side keeps the X value private and Makes the Y value available publicly to the outer side.

Step 3 User A Computes the key as $K = (Y_B)^{X_A} \bmod q$

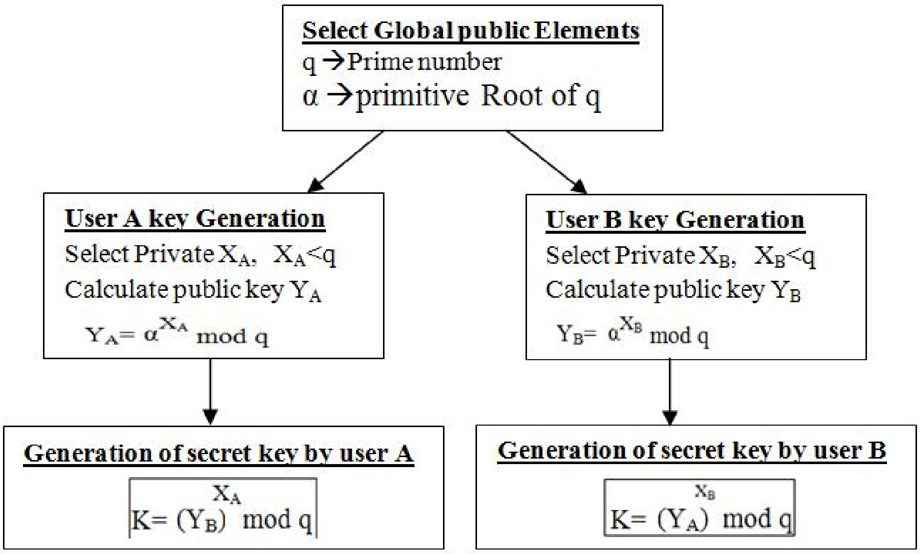
User B Computes the key as $K = (Y_A)^{X_B} \bmod q$

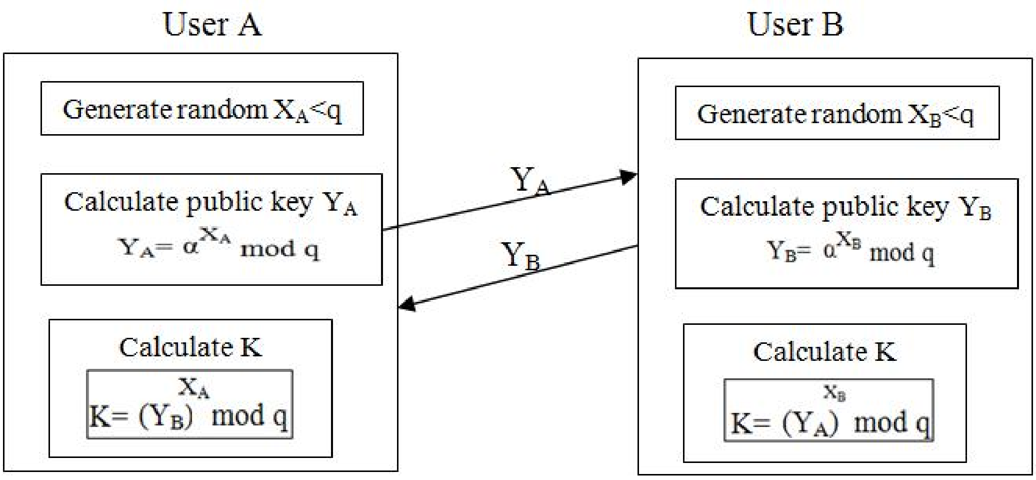
Step 4 two calculation produce identical results

(We know that)

(We know that)

The result is that the two sides have exchanged a secret key.





	How can Diffie-Hellman mitigate Man-in-the-Middle attacks? **Definition:** A man in the middle attack is a form of eavesdropping where communication between two users is monitored and modified by an unauthorized	2+2 [04]	CO2	L2

party.

Generally the attacker actively eavesdrops by intercepting (stoping) a public key message exchange.

The Diffie- Hellman key exchange is insecure against a “Man in the middle attack”.

Suppose user ‘A’ & ‘B’ wish to exchange keys, and D is the adversary (opponent). The attack proceeds as follows.

1. ‘D’ prepares for the attack by generating two random private keys X_{D1} & X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. ‘A’ transmits ‘Y_A’ to ‘B’
3. ‘D’ intercepts Y_A and transmits Y_{D1} to ‘B’. and D also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. ‘B’ receives Y_{D1} & calculate $K1 = (Y_{D1})^{X_B} \bmod q$.
5. ‘B’ transmits ‘Y_B’ to ‘A”
6. ‘D’ intercepts ‘Y_B’ and transmits Y_{D2} to ‘A’ and ‘D’ calculate K1 $K1 = (Y_B)^{X_{D1}} \bmod q$.
7. A receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key *K*1 and Alice and Darth share secret key *K*2.

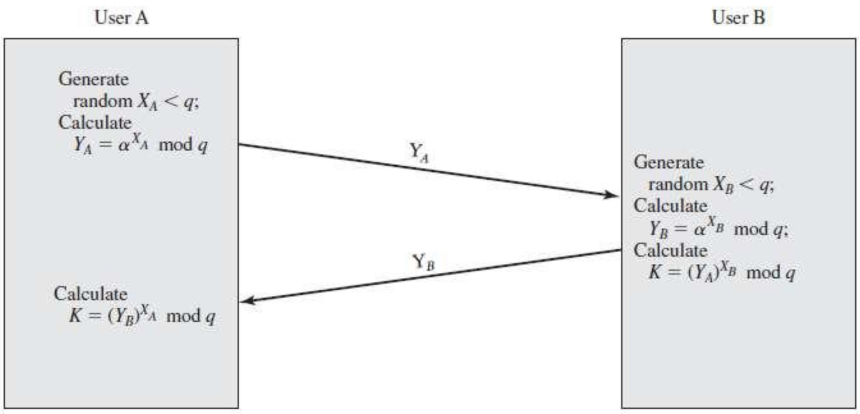


Figure 10.2 Diffie-Hellman Key Exchange