B.
Perform encryption and decryption using the RSA algorithm. You need to describe the detailed procedure, including using exponentiation modular arithmetic to compute x^y mod z

Answer:

Given Data:
Prime Numbers P = 3 , Q = 11
Encryption Key e = 7
Plain Text = 5
Decryption Key = ?
CipherText = ?

Step	**Example**
Choose two large prime numbers P and Q and compute N and Z	P=3 Q=11 N = P*Q = 33 Z = Φ(n) = (P-1)*(Q-1) = 20
Choose decryption key d that is relative prime to Z	**d = 3** Note: relative prime between d and Z means that d and Z have no prime factors in common, that is, their only common factors is 1. gcd(Z,d)=1; 1< d < Z;
Choose encryption key e so that "e*d mod Z=1", or "e = d^{-1} mod Z" Note: e < Z	This statement can be restated as e * d = Z* n + 1 e * d = Z * n + 1 where n is an integer e * 3 = 20 * n + 1 if n = 1, e * 3 = 21==> e = 7
Encryption E(M) = M^e mod N = C Note: • How to compute the mod of exponentiation 1. Use Google search 2. Use Widnows' calculator Starts ==> Programs ==> Accessaries ==> Calculator 3. Or you can use the Exponentiationprocedure described in Modular Arithmetic. Concept about congruent modulo n is also required for this approach.	Assume the message, M, is 19(Note: M must be smaller than N) C = cipher text = E(M) = M^e mod N = (5^7) mod 33 **C = 14**
Decryption D(C) = C^d mod N	Decrypt a message = D(C) = C^d mod N = 14^3 mod 33 = **5** <==== Thus the answer is correct

A.
Perform encryption and decryption using the RSA algorithm. You need to describe the detailed procedure, including using exponentiation modular arithmetic to compute x^y mod z

Answer:

Prime Numbers P = 7 , Q = 17
Decryption Key d = 5
Plain Text = 19
Encryption Key = ?
CipherText = ?

Step	**Example**
Choose two large prime numbers P and Q and compute N and Z	P=7 Q=17 N = P*Q = 119 Z = Φ(n) = (P-1)*(Q-1) =96
Choose decryption key d that is relative prime to Z	d = 5 Note: relative prime between d and Z means that d and Z have no prime factors in common, that is, their only common factors is 1. gcd(Z,d)=1; 1< d < Z;
Choose encryption key e so that "e*d mod Z=1", or "e = d^{-1} mod Z" Note: e < Z	This statement can be restated as e * d = Z* n + 1 e * d = Z * n + 1 where n is an integer e * 5 = 96 * n + 1 if n = 1, e * 5 = 97==> not a solution if n = 2, e * 5 = 193==> not a solution if n = 3, e * 5 = 289==> not a solution if n = 4, e * 5 = 385 ===> **e = 77**
Encryption E(M) = M^e mod N = C Note: ● How to compute the mod of exponentiation 1. Use Google search 2. Use Widnows' calculator Starts ==> Programs ==> Accessaries ==> Calculator 3. Or you can use the Exponentiationprocedure described in Modular Arithmetic. Concept about congruent modulo n is also required for this approach.	Assume the message, M, is 19(Note: M must be smaller than N) C = cipher text = E(M) = M^e mod N = (19^77) mod 119 **C = 66**
Decryption D(C) = C^d mod N	Decrypt a message = D(C) = C^d mod N = 66^5 mod 119 = 19 <==== Thus the answer is correct

Q2

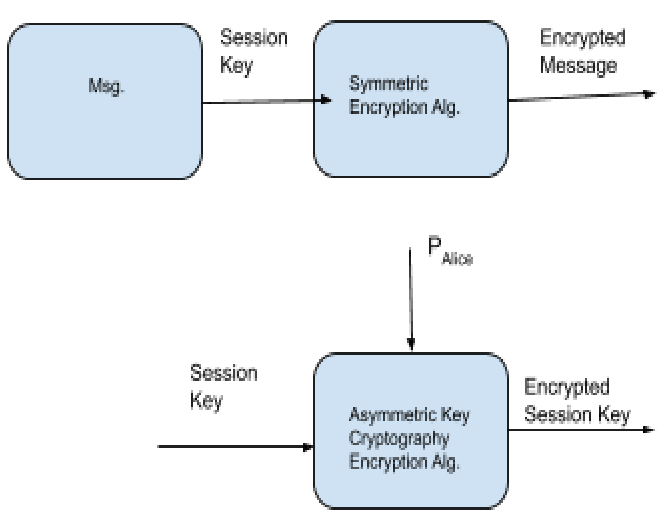
Draw a diagram to show the steps described in the following process (Page 81 of the book "Network Security Essentials, Applications and Standards, Second Edition")
This question assumes that Alice has a private key and a public key .
When Bob wishes to communicate with Alice, Bob can do the following:

1. **Bob prepares a message.**
2. **Bob creates a session key.**
3. **Bob encrypts the message using symmetric key cryptography and the session key.**
4. **Bob encrypts the session key using asymmetric key cryptography and Alice's public key.**
5. **Bob attaches the encrypted session key to the encrypted message and sends it to Alice.**
6. **Alice decrypts the session key using her private key.**
7. **Alice uses the decrypted session key to decrypt the message.**

The diagram should include Alice, Bob, and Eve.

Answer:

Msg => Message
S_{Alice} => Private Key of Alice
P_{Alice} => Public Key of Alice
S_{Bob} => Private Key of Bob
P_{Bob} => Public Key of Bob

What Bob knows and does	What Eve (the public) knows and does	What Alice knows and does
Msg , S_{Bob}, P_{Bob}, PAlice Bob ------------------------> Session key (creates) 	P_{Bob}, PAlice **Encrypted Msg** (i.e. Msg+SessionKey) **+ Encrypted Session key** (i.e.Session Key + Public Key of Alice(P_{Alice}))	S_{Alice} , P_{Alice}