Explanation of DH Algorithm

The Diffie–Hellman (DH) Key Exchange allows two people (say Alice and Bob) to establish a shared secret key even if they are communicating over an insecure public channel. The main thing which is maths lies in the use of modular arithmetic and exponentiation properties, which make it computationally very hard for an outsider to derive the secret key.

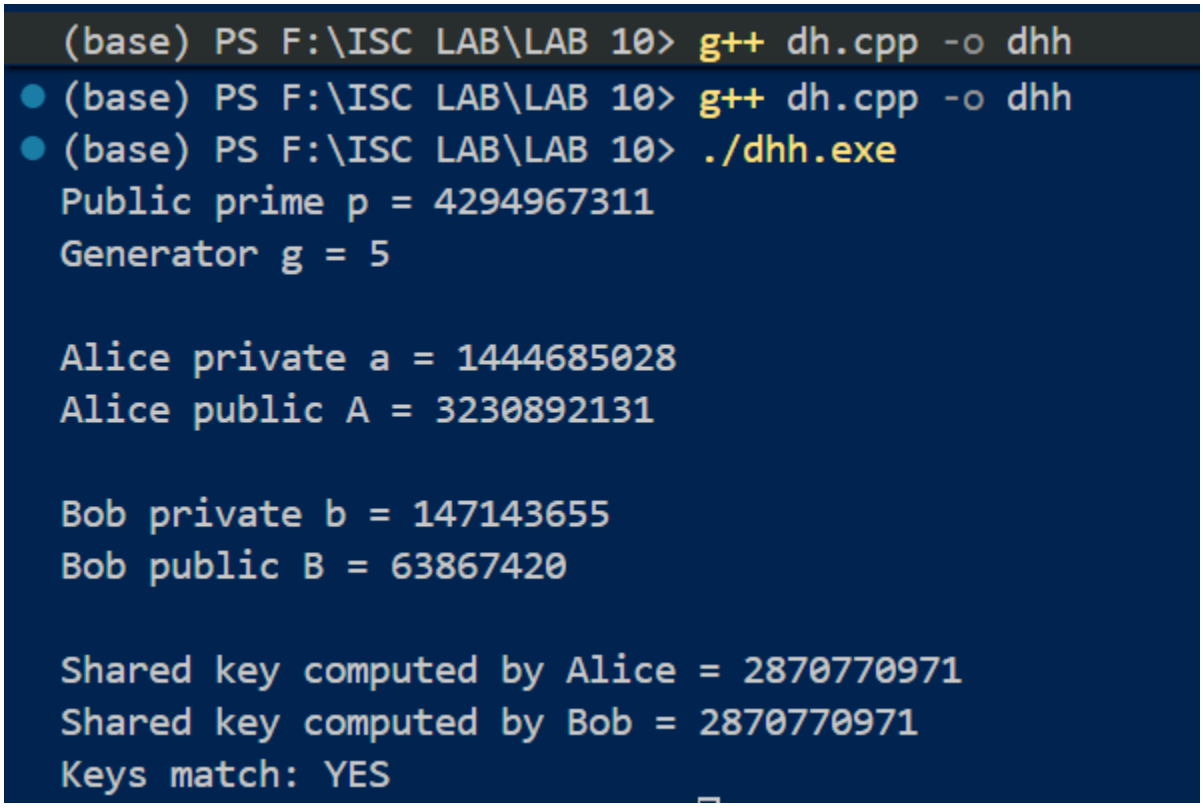
In the DH protocol, both Alice and Bob publicly agree on two numbers - a large prime number (p) and a primitive root (g) (also called a generator). These two numbers are not secret and can be known to everyone. Next, Alice randomly selects her private key (a) and computes her public value ($A = g^a \bmod p$). Similarly, Bob chooses his private key (b) and computes ($B = g^b \bmod p$). They exchange these public values (A) and (B) with each other.

Now, both can independently compute the same shared secret key. Alice calculates ($K_A = B^a \bmod p$), and Bob calculates ($K_B = A^b \bmod p$). Mathematically, both expressions result in the same value:

$K_A = K_B = g^{ab} \bmod p$.

This shared key can then be used as a secret encryption key for secure communication between Alice and Bob.

Output



Answer to Task 4

Even though an attacker can see all public values (p, g, A, B), they cannot compute the shared key because they do not know the private exponents (a) or (b). To find them, the attacker would need to solve the Discrete Logarithm Problem (DLP) - that is, given (g, p), and ($A = g^a \bmod p$), find (a). This problem is computationally infeasible for large primes since there's no efficient algorithm to solve it in a reasonable amount of time. Hence, the Diffie–Hellman key exchange remains secure against eavesdropping.