### ****Diffie-Hellman algorithm:****

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

* For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.
* P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a secret key, after which they have the same secret key to encrypt.

Step-by-Step explanation is as follows:

| **Alice** | **Bob** |
| --- | --- |
| Public Keys available = P, G | Public Keys available = P, G |
| Private Key Selected = a | Private Key Selected = b |
| Key generated =  x=GamodP*x*=*GamodP* | Key generated =  y=GbmodP*y*=*GbmodP* |
| Exchange of generated keys takes place | |
| Key received = y | key received = x |
| Generated Secret Key =  ka=yamodP*ka*​=*yamodP* | Generated Secret Key =  kb=xbmodP*kb*​=*xbmodP* |
| Algebraically, it can be shown that  ka=kb*ka*​=*kb*​ | |
| Users now have a symmetric secret key to encrypt | |

**Example:**

Step 1: Alice and Bob get public numbers P = 23, G = 9  
Step 2: Alice selected a private key a = 4 and  
 Bob selected a private key b = 3  
Step 3: Alice and Bob compute public values  
Alice: x =(9^4 mod 23) = (6561 mod 23) = 6  
 Bob: y = (9^3 mod 23) = (729 mod 23) = 16  
Step 4: Alice and Bob exchange public numbers  
Step 5: Alice receives public key y =16 and  
 Bob receives public key x = 6  
Step 6: Alice and Bob compute symmetric keys  
 Alice: ka = y^a mod p = 65536 mod 23 = 9  
 Bob: kb = x^b mod p = 216 mod 23 = 9  
Step 7: 9 is the shared secret.