**Diffie Hellman Code explanation**

|  |
| --- |
| class DiffieHellman(object): |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def \_\_init\_\_(self, ec, g): |
| Constructor to initialize ec which represents elliptic curve in a Finite Field . |
| |  | | --- | | self.ec = ec | |  |     ec is the instance of the EC class. |

|  |
| --- |
| self.g = g |
| g is the generator point on the ec. The point multiplication is done from this point. |
|  |

|  |
| --- |
| self.n = ec.order(g) |
| n is the prime order of G. The value of private key is from 1 to n-1. |
|  |

|  |
| --- |
| pass |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def gen(self, priv): |
|  |

|  |
| --- |
| this function is used to generate the public key. |

|  |
| --- |
| assert 0 < priv and priv < self.n |
| It checks whether the private key value is within the range from 1 to n-1 where n is the prime order of the generator point. |
|  |

|  |
| --- |
| return self.ec.mul(self.g, priv) |
| ec.mul function is which calculates point multiplication of g with the private key y. It is given as:  N=[p].g  g is the generator point and p is the private key. Value of N is returned which is the public key. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def secret(self, priv, pub): |
|  |

|  |
| --- |
| Secret function takes the private value and the public value and calculates the final key which both the sender and the receiver uses. |

|  |
| --- |
| assert self.ec.is\_valid(pub) |
| This function checks whether the public key which is exchanged between the sender and receiver is valid or not. |
|  |

|  |
| --- |
| assert self.ec.mul(pub, self.n) == self.ec.zero |
|  |

|  |
| --- |
| return self.ec.mul(pub, priv) |
| This function generates the final key which is given as:  N.[q] where N is [p].g which was calculated on the other side and was sent by the sender and g is the private key. Point multiplication in ECC is commutative therefore:  [p].n.[q] = [q].n.[p] therefore on the other side the public key will be [q].n and the private key will be [p] so the same key is generated. |
|  |

pass