Diffie Hellman Key Exchange

In [3]:

```
import random

#generate a prime number
def generate_prime():
    while True:
        p = random.randint(100,1000)
        for i in range(2, int(p**0.5)+1):
            if p % i ==0:
                break
        else:
            return p

#generate a public key
def generate_public_key(p,g,a):
    return pow(g,a,p)

#generate a shared secret
def generate_shared_secret(p,A,b):
    return pow(A,b, p)

#Example usage
p = generate_prime()
g = 2
a = random.randint(1,p-1)
b = random.randint(1,p-1)

A = generate_public_key(p,g,a)
B = generate_public_key(p,g,b)

shared_secret_1 = generate_shared_secret(p,A,b)
shared_secret_2 = generate_shared_secret(p,B,a)

print(f'Prime: {p}')
print(f'Generator: {g}')
print(f'Alice\'s private key: {a}')
print(f'Bob\'s private key: {b}')
print(f'Shared secret 1: {shared_secret_1}')
print(f'Shared secret 2: {shared_secret_2}')
```

```
Prime: 519
Generator: 2
Alice's private key: 141
Bob's private key: 162
Shared secret 1: 214
Shared secret 2: 214
```