In [ ]:
```
# Diffie-Hellman Key Exchange Implementation
```

In [2]:
```
def diffie_hellman(p, g, a, b):
    # Calculate public keys
    A = pow(g, a, p)
    B = pow(g, b, p)

    # Exchange and compute shared secret
    shared_secret_A = pow(B, a, p)
    shared_secret_B = pow(A, b, p)

    return {
        "Public key A": A,
        "Public key B": B,
        "Shared secret (Alice)": shared_secret_A,
        "Shared secret (Bob)": shared_secret_B
    }

# Example values
p = 23   # prime number
g = 5    # primitive root
a = 6    # Alice's private key
b = 15   # Bob's private key

result = diffie_hellman(p, g, a, b)

for key, value in result.items():
    print(f"{key}: {value}")
```

```
Public key A: 8
Public key B: 19
Shared secret (Alice): 2
Shared secret (Bob): 2
```

In [ ]:
```

```