Interlanguage Test Report - Python | Nodejs

1. **Creating Key Pairs in Python and verifying if the shared keys generated in both environments are the same.**

- **Creating a set of keypairs in Python:**

```
      keypair1=key_util.generate_key_pair()
      keypair2=key_util.generate_key_pair()

      println("keypair 1 ==> private_key:", keypair1.private_key, "public_key:",
keypair1.public_key)
      println("keypair 2 ==> private_key:", keypair2.private_key, public_key:",
keypair2.public_key)
```

- **Output:**

```
keypair 1 ==> private_key: MC4CAQAwBQYDK2VuBCIEIDCP34OGU48Ekots9wwpPflxZRdklzrm+J9RY/h/oIlJ public_key: MCowBQYDK2VuAyEAwb/2qbAF4noVnT4g3NWTkJsUyn
0BQv6gtTnS+hGIohU=
keypair 2 ==> private_key: MC4CAQAwBQYDK2VuBCIEIEBQw9WHztwR4SXSvvkm5BK7cG4WOZhHnNg2qAj4HIZy public_key: MCowBQYDK2VuAyEAzO5JWcZLPlf6LEogIV8RdsXsmf
Kse0vt5mGGb8Kfamw=
```

- **Generating Shared Keys for the above keypairs in Python:**

```
      // Generate Shared Key with User 1's Private Key and User 2's Public Key.
       shared_key1=key_util.generate_shared_key(keypair1.private_key,
keypair2.public_key)

        // Generate Shared Key with User 2's Private Key and User 1's Public Key.
         shared_key2=key_util.generate_shared_key(keypair2.private_key,
keypair1.public_key)
      println("shared_key1:",shared_key1)
      println("shared_key2:",shared_key2)
```

- **Output:**

```
shared_key1: XNVqd1qM7swar+SrOjitIafPTEaYmCnakYRGEsd2aXY=
shared_key2: XNVqd1qM7swar+SrOjitIafPTEaYmCnakYRGEsd2aXY=
```

- **Generating Shared keys for the same key pairs in NodeJS:**

```
// Generate Shared Key with User 1's Private Key and User 2's Public Key.
const sharedKey1 = generateSharedKey(
 "MC4CAQAwBQYDK2VuBCIEIDCP34OGU48Ekots9wwpPflxZRdklzrm+J9RY/h/oIlJ",
 "MCowBQYDK2VuAyEAzO5JWcZLPlf6LEogIV8RdsXsmfKse0vt5mGGb8Kfamw="
);
console.log("SharedKey1 ==> " + sharedKey1);

// Generate Shared Key with User 2's Private Key and User 1's Public Key.
const sharedKey2 = generateSharedKey(
 "MC4CAQAwBQYDK2VuBCIEIEBQw9WHztwR4SXSvvkm5BK7cG4WOZhHnNg2qAj4HIZy",
 "MCowBQYDK2VuAyEAwb/2qbAF4noVnT4g3NWTkJsUyn0BQv6gtTnS+hGIohU="
);
console.log("SharedKey2 ==> " + sharedKey2);
```

- **Output:**

```
SharedKey1 ==> XNVqd1qM7swar+SrOjitIafPTEaYmCnakYRGEsd2aXY=
SharedKey2 ==> XNVqd1qM7swar+SrOjitIafPTEaYmCnakYRGEsd2aXY=
```

Observation:

Shared keys generated from both utilities for the same set of keypairs generated from the Python utility are the same.

2. **Encrypting the text in Python and decrypting in Nodejs (Using above-shared keys)**

- **Encrypting the text in Python**

```
raw_data = "Hello This is ONDC Test Data"

println("-----------------------------------------------")
encrypted_data=encryption_util.encrypt_data(shared_key1,raw_data)
println("Encrypted Data ===> ", encrypted_data)
println("-----------------------------------------------")
```

- **Output:**

```
Encrypted Data ===> eyJub25jZSI6ICIraGZNNE1GamJDRDhzSXg5IiwgImVuY3J5cHRlZF9kYXRhIjogImk3c0FySGxiZmYyZXEvRXdTNkdBYVFtZU9wSTU4YzJpQkF2clpBPT0iLCAia
G1hYyI6ICI1VFhJVnNSTDc2aGI3d3JwWWFYR2lnPT0ifQ==
```

- **Decrypting the above text in nodejs:**

```
const encryptedData =
"eyJub25jZSI6ICIraGZNNE1GamJDRDhzSXg5IiwgImVuY3J5cHRlZF9kYXRhIjogImk3c0FySGxiZm
YyZXEvRXdTNkdBYVFtZU9wSTU4YzJpQkF2clpBPT0iLCAiaG1hYyI6ICI1VFhJVnNSTDc2aGI3d3JwW
WFYR2lnPT0ifQ==";

const sharedKey = "XNVqd1qM7swar+SrOjitIafPTEaYmCnakYRGEsd2aXY=";

const decryptedData = decryptData(sharedKey, encryptedData);

console.log(decryptedData);
```

- **Output:**

```
decrypted Data ===>  Hello This is ONDC Test Data
```

Observation:

Data encrypted in Python utility was decrypted successfully in Node.js Utility.

3. **Generating Key Pairs in Node.js and verifying whether the shared keys generated in both environments are the same:**

- **Generating a set of Key pairs in Nodejs:**

```
// Generate Key Pair for User 1.
const keyPair1 = generateKeyPair();
console.log("Key Pair 1 ==> ", keyPair1);

// Generate Key Pair for User 2.
const keyPair2 = generateKeyPair();
console.log("Key Pair 2 ==> ", keyPair2);
```

- **Output:**

```
Key Pair 1 ==>  {
  publicKey: 'MCowBQYDK2VuAyEAKSH2C3vjjeJykXKzhZGISY4Z5AhBX1F5pXTyxglgkkU=',
  privateKey: 'MC4CAQAwBQYDK2VuBCIEIIiE2k0kYcVjTbQ9BSWXbdtXQsWxRX+Er8aPxspif3xL'
}
Key Pair 2 ==>  {
  publicKey: 'MCowBQYDK2VuAyEAf4SFqEYYCop97r6XFo4g7bIgMps09mwdfyjMghb460s=',
  privateKey: 'MC4CAQAwBQYDK2VuBCIEIAimThnpMhx0QitdopYVb+W4jcd0dIhlpRFvM8xTBbJT'
}
```

- **Generating Shared key in Nodejs:**

```
// Generate Shared Key with User 1's Private Key and User 2's Public Key.
const sharedKey1 = generateSharedKey(keyPair1.privateKey, keyPair2.publicKey);
console.log("SharedKey1 ==> " + sharedKey1);

// Generate Shared Key with User 2's Private Key and User 1's Public Key.
const sharedKey2 = generateSharedKey(keyPair2.privateKey, keyPair1.publicKey);
console.log("SharedKey2 ==> " + sharedKey2);
```

- **Output:**

```
SharedKey1 ==> q6F5Fwb5O8bxptR8liG65eu4jSq1gLFjNg6EsZVuAWc=
SharedKey2 ==> q6F5Fwb5O8bxptR8liG65eu4jSq1gLFjNg6EsZVuAWc=
```

- **Generating shared key in Python for the above key pairs:**

```
      keypair1=key_util.DHKeyPair('MC4CAQAwBQYDK2VuBCIEIIiE2k0kYcVjTbQ9BSWXbdtXQsWxRX
+Er8aPxspif3xL','MCowBQYDK2VuAyEAKSH2C3vjjeJykXKzhZGISY4Z5AhBX1F5pXTyxglgkkU=')
      keypair2=key_util.DHKeyPair('MC4CAQAwBQYDK2VuBCIEIAimThnpMhx0QitdopYVb+W4jcd0dI
hlpRFvM8xTBbJT','MCowBQYDK2VuAyEAf4SFqEYYCop97r6XFo4g7bIgMps09mwdfyjMghb460s=')

      println("keypair 1 ==> private_key:",keypair1.private_key,
"public_key:",keypair1.public_key)
      println("keypair 2 ==> private_key:",keypair2.private_key,"public_key:",
keypair2.public_key)

      // Generate Shared Key with User 1's Private Key and User 2's Public Key.
      shared_key1=key_util.generate_shared_key(keypair1.private_key,
keypair2.public_key)
      // Generate Shared Key with User 2's Private Key and User 1's Public Key.
      shared_key2=key_util.generate_shared_key(keypair2.private_key,
keypair1.public_key)

   println("shared_key1:",shared_key1)
   println("shared_key2:",shared_key2)
```

- **Output:**

```
shared_key1: q6F5Fwb5O8bxptR8liG65eu4jSq1gLFjNg6EsZVuAWc=
shared_key2: q6F5Fwb5O8bxptR8liG65eu4jSq1gLFjNg6EsZVuAWc=
```

Observation:

The shared keys generated for both the utilities with the same set of keypairs generated from nodejs utility are the same.

4. **Encrypting the text in Nodejs utility and decrypting it in Python utility:**

- **Encrypting in Nodejs utility:**

```
// Initializing the raw text to be encrypted.
const MESSAGE_DATA = "Hello This is ONDC Test Data";

// Encrypting the raw data.
const encryptedData = encryptData(sharedKey1, MESSAGE_DATA);
console.log("Encrypted Data ===> " + encryptedData);
```

- **Output:**

```
Encrypted Data ===> eyJlbmNyeXB0ZWRfZGF0YSI6ImE3Q0o3QWJTMDBSNlBFUXZJVC81WmFsRUh6OW44Vko1Y2
1mcWtBPT0iLCJobWFjIjoibFBCbTQxWFB4ZzBIeC8ydzNhcW1HZz09Iiwibm9uY2UiOiJLTTZWb0M5YmNtb2Fzb3Qw
In0=
```

- **Decrypting the above text in python utility:**

```
    encrypted_data='eyJlbmNyeXB0ZWRfZGF0YSI6ImE3Q0o3QWJTMDBSNlBFUXZJVC81WmFsRUh6OW4
4Vko1Y21mcWtBPT0iLCJobWFjIjoibFBCbTQxWFB4ZzBIeC8ydzNhcW1HZz09Iiwibm9uY2UiOiJLTTZWb0M5Y
mNtb2Fzb3QwIn0='

    decrypted_data=encryption_util.decrypt_data(shared_key2,encrypted_data)
    println("decrypted Data ===> ",decrypted_data)
    println("------------------------------------------------")
```

- **Output:**

```
Decrypted Data ===> Hello This is ONDC Test Data
```

Observation:

Text encrypted in the Node.js utility was decrypted successfully in the Python utility.