2.2 DSA

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypt &amp; Decrypt</td>
</tr>
<tr>
<td>CKM_DSA_KEY_PAIR_GEN</td>
<td>✓</td>
</tr>
<tr>
<td>CKM_DSA_PARAMETER_GEN</td>
<td>✓</td>
</tr>
<tr>
<td>CKM_DSA_SHA1</td>
<td>✓</td>
</tr>
<tr>
<td>CKM_DSA_FIPS_186_4</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 2.2.1 Definitions

This section defines the key type “CKK_DSA” for type CK_KEY_TYPE as used in the CKA_KEY_TYPE attribute of DSA key objects.

Mechanisms:

- CKM_DSA_KEY_PAIR_GEN
- CKM_DSA
- CKM_DSA_SHA1
- CKM_DSA_PARAMETER_GEN
- CKM_FORTEZZA_TIMESTAMP
- CKM_DSA_FIPS_186_4

### 2.2.2 DSA public key objects

DSA public key objects (object class CKO_PUBLIC_KEY, key type CKK_DSA) hold DSA public keys. The following table defines the DSA public key object attributes, in addition to the common attributes defined for this object class:

**Table 1, DSA Public Key Object Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKA_PRIME(^{1,3})</td>
<td>Big integer</td>
<td>Prime (p) (512 to 1024 bits, in steps of 64 bits)</td>
</tr>
<tr>
<td>CKA_SUBPRIME(^{1,3})</td>
<td>Big integer</td>
<td>Subprime (q) (160 bits)</td>
</tr>
<tr>
<td>CKA_BASE(^{1,3})</td>
<td>Big integer</td>
<td>Base (g)</td>
</tr>
<tr>
<td>CKA_VALUE(^{1,4})</td>
<td>Big integer</td>
<td>Public value (y)</td>
</tr>
</tbody>
</table>

\(^1\) Refer to [PKCS #11-B] table 15 for footnotes
The **CKA_PRIME**, **CKA_SUBPRIME** and **CKA_BASE** attribute values are collectively the “DSA domain parameters”. See FIPS PUB 186-2 for more information on DSA keys.

The following is a sample template for creating a DSA public key object:

```c
CK_OBJECT_CLASS class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_DSA;
CK_UTF8CHAR label[] = "A DSA public key object";
CK_BYTE prime[] = {...};
CK_BYTE subprime[] = {...};
CK_BYTE base[] = {...};
CK_BYTE value[] = {...};
CK_BBOOL true = CK_TRUE;
CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &class, sizeof(class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &true, sizeof(true)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_PRIME, prime, sizeof(prime)},
    {CKA_SUBPRIME, subprime, sizeof(subprime)},
    {CKA_BASE, base, sizeof(base)},
    {CKA_VALUE, value, sizeof(value)}
};
```

### 2.2.8 DSA with SHA-1

The DSA with SHA-1 mechanism, denoted **CKM_DSA_SHA1**, is a mechanism for single- and multiple-part signatures and verification based on the Digital Signature Algorithm defined in FIPS PUB 186-2. This mechanism computes the entire DSA specification, including the hashing with SHA-1.

For the purposes of this mechanism, a DSA signature is a 40-byte string, corresponding to the concatenation of the DSA values $r$ and $s$, each represented most-significant byte first.

This mechanism does not have a parameter.

Constraints on key types and the length of data are summarized in the following table:

**Table 2, DSA with SHA-1: Key And Data Length**

<table>
<thead>
<tr>
<th>Function</th>
<th>Key type</th>
<th>Input length</th>
<th>Output length</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_Sign</td>
<td>DSA private key</td>
<td>any</td>
<td>40</td>
</tr>
<tr>
<td>C_Verify</td>
<td>DSA public key</td>
<td>any, 40(^2)</td>
<td>N/A</td>
</tr>
</tbody>
</table>

\(^2\) Data length, signature length.

For this mechanism, the `ulMinKeySize` and `ulMaxKeySize` fields of the **CK_MECHANISM_INFO** structure specify the supported range of DSA prime sizes, in bits.

### 2.2.9 FIPS 186-4

**CKM_DSA_FIPS_186_4** is identical to **CKM_DSA** except that the bit length of $p$ shall be 1024.