CKO_ .

CKO_PUBLIC_KEY, CKO_PRIVATE_KEY CKO_SECRET_KEY CK_OBJECT_CLASS CKA_CLASS.

-
-
  - CKO_PUBLIC_KEY, CKK_RSA ( )
  - CKO_PUBLIC_KEY, CKK_GOSTR3410 ( )
  - CKO_PUBLIC_KEY, CKK_GOSTR3410_512 ( )
-
  - CKO_PRIVATE_KEY, CKK_RSA ( )
  - CKO_PRIVATE_KEY, CKK_GOSTR3410 ( )
  - CKO_PRIVATE_KEY, CKK_GOSTR3410_512 ( )
-
  - CKO_SECRET_KEY, CKK_GENERIC_SECRET ( )
  - CKO_SECRET_KEY, CKK_GOST ( )
  - CKO_SECRET_KEY, CKK_GOST28147 ( )

, , , .

, , , ,

CKA_KEY_TYPE[1,2]	CK_KEY_TYPE	
CKA_ID[3]	Byte array	( )
CKA_START_DATE[3]	CK_DATE	( )
CKA_END_DATE[3]	CK_DATE	( )
CKA_DERIVE[3]	CK_BBOOL	CK_TRUE, ( ). CK_FALSE
CKA_LOCAL[4,5,6]	CK_BBOOL	CK_TRUE, **C_GenerateKey C_GenerateKeyPair C_CopyObject CKA_LOCAL**, CK_TRUE
CKA_KEY_GEN_MECHANISM[4,5,6]	CK_MECHANISM_TYPE	,
CKA_ALLOWED_MECHANISMS	CK_MECHANISM_TYPE _PTR, CK_MECHANISM_TYPE	, . *ulValueLen* CK_MECHANISM_TYPE.

1 **C_CreateObject.**

2 **C_UnwrapKey.**

3 **C_SetAttributeValue C_CopyObject**

4 **C_CreateObject**

5 **C_GenerateKey C_GenerateKeyPair**

6 **C_UnwrapKey**

CKA_ID . , , . , , .

CKA_ID .

CKA_START_DATE CKA_END_DATE . , .

CKA_DERIVE CK_TRUE , .

CKA_LOCAL CK_TRUE , **C_GenerateKey C_GenerateKeyPair**.

CKA_KEY_GEN_MECHANISM . , **CKA_LOCAL** CK_TRUE. **CKA_LOCAL** CK_FALSE, CK_UNAVAILABLE_INFORMATION.

- CKO_PUBLIC_KEY, CKK_RSA ( )
- CKO_PUBLIC_KEY, CKK_GOSTR3410 ( )
- CKO_PUBLIC_KEY, CKK_GOSTR3410_512 ( )

(**CKO_PUBLIC_KEY**) . , , .

( )

(Common Public Key Attributes)		
CKA_SUBJECT[1]	Byte array	DER- ( )
CKA_ENCRYPT[1]	CK_BBOOL	CK_TRUE, [2]
CKA_VERIFY[1]	CK_BBOOL	CK_TRUE, , [2]
CKA_VERIFY_RECOVER[1]	CK_BBOOL	CK_TRUE, ( )[2]
CKA_WRAP[1]	CK_BBOOL	CK_TRUE, ( )[2]
CKA_TRUSTED[3]	CK_BBOOL	, . () CKA_WRAP_WITH_TRUSTED CK_TRUE.
CKA_WRAP_TEMPLATE	CK_ATTRIBUTE_PTR	(). , . *ulValueLen* CK_ATTRIBUTE.
, (Rutoken Vendors Defined Public Key Attributes)		
CKA_CAPI_ID		(0)
CKA_PUBLIC_KEY_RSF_ID		RSF–, (0)
CKA_PRIVATE_KEY_RSF_ID		RSF–, (0)
CKA_VENDOR_KEY_JOURNAL	CK_BBOOL	CK_TRUE, PINPad

1 **C_SetAttributeValue**

2

3 CK_TRUE

CKA_SUBJECT CKA_ID **CKA_SUBJECT CKA_ID** . , .

CKA_CAPI_ID ID rtCSP. , **CKA_CAPI_ID, CKA_PUBLIC_KEY_RSF_ID CKA_PRIVATE_KEY_RSF_ID**, , 0.

CKA_VENDOR_KEY_JOURNAL PINPad.

keyUsage ISO/IEC 9594-8 (X.509) PKCS #11 .

.509 PKCS #11

X.509	PKCS #11
dataEncipherment	CKA_ENCRYPT
digitalSignature, keyCertSign, cRLSign	CKA_VERIFY
digitalSignature, keyCertSign, cRLSign	CKA_VERIFY_RECOVER
keyAgreement	CKA_DERIVE
keyEncipherment	CKA_WRAP
nonRepudiation	CKA_VERIFY
nonRepudiation	CKA_VERIFY_RECOVER

CKO_PUBLIC_KEY, CKK_RSA ( )

RSA (CKO_PUBLIC_KEY, CKK_GENERIC_SECRET) RSA.

CKO_PUBLIC_KEY CKK_RSA CKO_PUBLIC_KEY, , .

CKO_PUBLIC_KEY, CKK_RSA

RSA (RSA Public Key Attributes)		
CKA_MODULUS[1,2]	Big integer	*n*
CKA_MODULUS_BITS[3,4]	CK_ULONG	*n*
CKA_PUBLIC_EXPONENT[1]	Big integer	*e*

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_CreateObject**

4 **C_GenerateKey C_GenerateKeyPair**

RSA

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GENERIC_RSA;
CK_UTF8CHAR label[] = "An RSA public_key object";
CK_BYTE modulus[] = {...};
CK_BYTE exponent[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_WRAP, &IsTrue , sizeof(IsTrue )},
        {CKA_ENCRYPT, &IsTrue , sizeof(IsTrue )},
        {CKA_MODULUS, modulus, sizeof(modulus)},
        {CKA_PUBLIC_EXPONENT, exponent, sizeof(exponent)}
};
```

CKO_PUBLIC_KEY, CKK_GOSTR3410 ( )

CKO_PUBLIC_KEY CKK_GOSTR3410 34.10-2001 34.10-2012 512 .

CKO_PUBLIC_KEY CKK_GOSTR3410 CKO_PUBLIC_KEY, , .

CKO_PUBLIC_KEY, CKK_GOSTR3410

34.10-2001 / 34.10-2012 (GOST R 3410 Public Key Attributes)		
CKA_VALUE[1,2]	Byte array	64 : 32 (,) , (little endian)
CKA_GOSTR3410PARAMS[1,3]	Byte array	, 34.10-2001 (OID) DER-. , CKK_GOSTR3410 CKA_OBJECT_ID
CKA_GOSTR3411PARAMS[1,3,4]	Byte array	, 34.11-94 34.11-2012 (OID) DER-. , CKK_GOSTR3411 CKA_OBJECT_ID
CKA_GOST28147_PARAMS[4]	Byte array	, 28147-89 (OID) DER-. , CKK_GOSTR28147 CKA_OBJECT_ID.

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_GenerateKey C_GenerateKeyPair**

4 **C_SetAttributeValue**.

, , () C_CreateObject 28147-89, 34.10-2001 34.10-2012 (CKA_TOKEN = TRUE).

34.10-2001

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2001 public_key object";
CK_BYTE keyPairIdGost[] = {"GOST R 34.10-2001 sample key pair 1 ID (Aktiv Co.)"};
CK_BYTE gostR3410params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x00};
CK_BYTE gostR3411params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1e, 0x00};
CK_BYTE gost28147params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost, sizeof(keyPairIdGost)-1},
        {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
        {CKA_GOSTR3410PARAMS, gostR3410params_oid, sizeof(gostR3410params_oid)},
};
```

34.10-2012 (256)

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 public_key object";
CK_BYTE keyPairIdGost_256[] = {"GOST R 34.10-2012(256) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x01};
CK_BYTE parametersGostR3411_256[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x02};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost_256, sizeof(keyPairIdGost_256)-1},
        {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
        {CKA_GOSTR3410PARAMS, parametersGostR3410, sizeof(parametersGostR3410)},
        {CKA_GOSTR3411PARAMS, parametersGostR3411_256, sizeof(parametersGostR3411_256)},
        {CKA_VALUE, value, sizeof(value)}
};
```

CKO_PUBLIC_KEY, CKK_GOSTR3410_512 ( )

34.10-2012 (CKO_PUBLIC_KEY, CKK_GOSTR3410_512) 34.10-2012 1024 .

CKO_PUBLIC_KEY CKK_GOSTR3410 CKO_PUBLIC_KEY, , .

CKO_PUBLIC_KEY, CKK_GOSTR3410

34.10-2012		
CKA_VALUE[1,2]	Byte array	128 : 64 (,) , (little endian)
CKA_GOSTR3410PARAMS[1,3]	Byte array	, 34.10-2012 (OID) DER-. , CKK_GOSTR3410 CKA_OBJECT_ID
CKA_GOSTR3411PARAMS[1,3,4]	Byte array	, 34.11-2012 (OID) DER-. , CKK_GOSTR3411 CKA_OBJECT_ID
CKA_GOST28147_PARAMS[4]	Byte array	, 28147-89 (OID) DER-. , CKK_GOSTR28147 CKA_OBJECT_ID.

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_GenerateKey C_GenerateKeyPair**

4 **C_SetAttributeValue**.

, , () C_CreateObject 28147-89, 34.10-2001 34.10-2012 (CKA_TOKEN = TRUE).

34.10-2012 (512)

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410_512;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 public_key object";
CK_BYTE keyPairIdGost_512[] = {"GOST R 34.10-2012(512) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410_512[] = {0x06, 0x09, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x02, 0x01, 0x02, 0x01};
CK_BYTE parametersGostR3411_512[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x03};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[128] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost_512, sizeof(keyPairIdGost1_512)-1},
        {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
        {CKA_GOSTR3410PARAMS, parametersGostR3410_512, sizeof(parametersGostR3410_512)},
        {CKA_GOSTR3411PARAMS, parametersGostR3411_512, sizeof(parametersGostR3411_512)},
        {CKA_VALUE, value, sizeof(value)}
};
```

- CKO_PRIVATE_KEY, CKK_RSA ( )
- CKO_PRIVATE_KEY, CKK_GOSTR3410 ( )
- CKO_PRIVATE_KEY, CKK_GOSTR3410_512 ( )

(**CKO_PRIVATE_KEY**) . , , .

(Common Private Key Attributes)		
CKA_SUBJECT[1]	Byte array	DER- ( )
CKA_SENSITIVE[1,2]	CK_BBOOL	CK_TRUE, ( )[3]
CKA_DECRYPT[1]	CK_BBOOL	CK_TRUE, [3]

CKA_SIGN[1]	CK_BBOOL	CK_TRUE, , [3]
CKA_SIGN_RECOVER[1]	CK_BBOOL	CK_TRUE, [3]
CKA_UNWRAP[1]	CK_BBOOL	CK_TRUE, (, ..)[3]
CKA_EXTRACTABLE[1,4]	CK_BBOOL	CK_TRUE, [3]
CKA_ALWAYS_SENSITIVE[5,6,7]	CK_BBOOL	CK_TRUE, CKA_SENSITIVE CK_TRUE
CKA_NEVER_EXTRACTABLE[5,6,7]	CK_BBOOL	CK_TRUE, CKA_EXTRACTABLE CK_TRUE
CKA_WRAP_WITH_TRUSTED[2]	CK_BBOOL	CK_TRUE, CKA_TRUSTED CK_TRUE. CK_FALSE.
CKA_UNWRAP_TEMPLATE	CK_ATTRIBUTE_PTR	. , . *ulValueLen* CK_ATTRIBUTE.
CKA_ALWAYS_AUTHENTICATE	CK_BBOOL	CK_TRUE, PIN- . CK_FALSE.
, (Rutoken Vendors Defined Private Key Attributes)		
CKA_CAPI_ID		(0).
CKA_PUBLIC_KEY_RSF_ID		RSF–, (0).
CKA_PRIVATE_KEY_RSF_ID		RSF–, (0).
CKA_VENDOR_KEY_PIN_ENTER	CK_BBOOL	CK_TRUE, / PIN- PINPad.
CKA_VENDOR_KEY_CONFIRM_OP	CK_BBOOL	CK_TRUE, / PINPad.
CKA_VENDOR_KEY_JOURNAL	CK_BBOOL	CK_TRUE, PINPad

1 **C_SetAttributeValue**

2 CK_TRUE read-only

3

4 CK_FALSE read-only

5 **C_CreateObject**

6 **C_GenerateKey C_GenerateKeyPair**

7 **C_UnwrapKey**

CKA_SUBJECT CKA_ID CKA_SUBJECT CKA_ID . , .

CKA_SENSITIVE CK_TRUE **CKA_EXTRACTABLE** CK_FALSE, . .

CKA_ALWAYS_AUTHENTICATE (PIN-) . «» - , , . CK_TRUE, **CKA_PRIVATE** CK_TRUE.

C_Login *userType*, **CKU_CONTEXT_SPECIFIC**, , (, **C_SignInit**). , . **C_Login** CKR_OK, , , , (, **C_Sign, C_SignFinal**). **C_Login** CKR_PIN_INCORRECT , , **C_Login** . , PIN-. **C_Login** CKR_PIN_LOCKED . CKA_ALWAYS_AUTHENTICATE CK_TRUE CKR_USER_NOT_LOGGED_IN . **C_Login** CKR_OPERATION_NOT_INITIALIZED, , CKA_ALWAYS_AUTHENTICATE CK_FALSE.

CKA_CAPI_ID ID rtCSP. , **CKA_CAPI_ID, CKA_PUBLIC_KEY_RSF_ID CKA_PRIVATE_KEY_RSF_ID**, , 0.

CKA_VENDOR_KEY_PIN_ENTER, **CKA_VENDOR_KEY_CONFIRM_OP CKA_VENDOR_KEY_JOURNAL** PINPad.

CKO_PRIVATE_KEY, CKK_RSA ( )

RSA (CKO_PRIVATE_KEY, CKK_GENERIC_SECRET) RSA.

CKO_PRIVATE_KEY CKK_RSA CKO_PRIVATE_KEY, , .

CKO_PRIVATE_KEY, CKK_RSA

RSA (RSA Private Key Attributes)		

CKA_MODULUS[1,2,3]	Big integer	*n*
CKA_PUBLIC_EXPONENT[2,3]	Big integer	*e*
CKA_PRIVATE_EXPONENT[1,2,3,4]	Big integer	*d*
CKA_PRIME_1[2,3,4]	Big integer	*p*
CKA_PRIME_2[2,3,4]	Big integer	*q*
CKA_EXPONENT_1[2,3,4]	Big integer	*d* *p*-1
CKA_EXPONENT_2[2,3,4]	Big integer	*d* *q*-1
CKA_COEFFICIENT[2,3,4]	Big integer	CTR (,) RSA q^{-1} *p*

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_UnwrapKey**

4 **CKA_SENSITIVE**, CK_TRUE **CKA_EXTRACTABLE** CK_FALSE.

RSA PKCS #1.

RSA

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_RSA;
CK_UTF8CHAR label[] = "An RSA private_key object";
CK_BYTE subject[] = {...};
CK_BYTE id[] = {123};
CK_BYTE modulus[] = {...};
CK_BYTE publicExponent[] = {...};
CK_BYTE privateExponent[] = {...};
CK_BYTE prime1[] = {...};
CK_BYTE prime2[] = {...};
CK_BYTE exponent1[] = {...};
CK_BYTE exponent2[] = {...};
CK_BYTE coefficient[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
        {CKA_CLASS, &your_class, sizeof(your_class)},
        {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
        {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
        {CKA_LABEL, label, sizeof(label)-1},
        {CKA_SUBJECT, subject, sizeof(subject)},
        {CKA_ID, id, sizeof(id)},
        {CKA_SENSITIVE, &IsTrue, sizeof(IsTrue)},
        {CKA_DECRYPT, &IsTrue, sizeof(IsTrue)},
        {CKA_SIGN, &IsTrue, sizeof(IsTrue)},
        {CKA_MODULUS, modulus, sizeof(modulus)},
        {CKA_PUBLIC_EXPONENT, publicExponent, sizeof(publicExponent)},
        {CKA_PRIVATE_EXPONENT, privateExponent, sizeof(privateExponent)},
        {CKA_PRIME_1, prime1, sizeof(prime1)},
        {CKA_PRIME_2, prime2, sizeof(prime2)},
        {CKA_EXPONENT_1, exponent1, sizeof(exponent1)},
        {CKA_EXPONENT_2, exponent2, sizeof(exponent2)},
        {CKA_COEFFICIENT, coefficient, sizeof(coefficient)}
};
```

CKO_PRIVATE_KEY, CKK_GOSTR3410 ( )

34.10 (CKO_PRIVATE_KEY, CKK_GOSTR3410) 34.10-2001 34.10-2012 256 .

CKO_PRIVATE_KEY CKK_GOSTR3410 CKO_PRIVATE_KEY, , .

CKO_PRIVATE_KEY, CKK_GOSTR3410

34.10-2001 / 34.10-2012 (GOST R 3410 Private Key Attributes)		
CKA_VALUE[1,2,3,4]	Byte array	32 , (little endian)
CKA_GOSTR3410PARAMS[1,2,3]	Byte array	, 34.10-2001 (OID) DER-. , CKK_GOSTR3410 CKA_OBJECT_ID
CKA_GOSTR3411PARAMS[1,2,3,5]	Byte array	, 34.11-94 34.11-2012 (OID) DER-. , CKK_GOSTR3411 CKA_OBJECT_ID
CKA_GOST28147_PARAMS[2,3,5]	Byte array	, 28147-89 (OID) DER-. , CKK_GOSTR28147 CKA_OBJECT_ID.

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_UnwrapKey**

4 **CKA_SENSITIVE**, CK_TRUE **CKA_EXTRACTABLE** CK_FALSE.

5 **C_SetAttributeValue**.

, 34.10-2001/2012 . , 34.10-2001/2012 .

, , () `C_CreateObject` 28147-89, 34.10-2001 34.10-2012 (`CKA_TOKEN = TRUE`).

34.10-2001

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2001 private_key object";
CK_BYTE keyPairIdGost[] = {"GOST R 34.10-2001 sample key pair 1 ID (Aktiv Co.)"};
CK_BYTE gostR3410params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x00};
CK_BYTE gostR3411params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1e, 0x00};
CK_BYTE gost28147params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[32] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost, sizeof(keyPairIdGost)-1},
        {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
        {CKA_GOSTR3410PARAMS, gostR3410params_oid, sizeof(gostR3410params_oid)},
        {CKA_GOSTR3411PARAMS, gostR3411params_oid, sizeof(gostR3411params_oid)},
        {CKA_VALUE, value, sizeof(value)}
};
```

34.10-2012 (256)

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 private_key object";
CK_BYTE keyPairIdGost_256[] = {"GOST R 34.10-2012(256) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x01};
CK_BYTE parametersGostR3411_256[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x02};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[32] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost_256, sizeof(keyPairIdGost_256)-1},
        {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
        {CKA_GOSTR3410PARAMS, parametersGostR3410, sizeof(parametersGostR3410)},
        {CKA_GOSTR3411PARAMS, parametersGostR3411_256, sizeof(parametersGostR3411_256)},
        {CKA_VALUE, value, sizeof(value)}
};
```

CKO_PRIVATE_KEY, CKK_GOSTR3410_512 ( )

CKO_PRIVATE_KEY CKK_GOSTR3410 34.10-2012 512 .

CKO_PRIVATE_KEY CKK_GOSTR3410 CKO_PRIVATE_KEY, , .

CKO_PRIVATE_KEY, CKK_GOSTR3410

34.10-2001 / 34.10-2012 (GOST R 3410 Private Key Attributes)		
CKA_VALUE[1,2,3,4]	Byte array	64 , (little endian)
CKA_GOSTR3410PARAMS[1,2,3]	Byte array	, 34.10-2012 (OID) DER-. , CKK_GOSTR3410 CKA_OBJECT_ID
CKA_GOSTR3411PARAMS[1,2,3,5]	Byte array	, 34.11-2012 (OID) DER-. , CKK_GOSTR3411 CKA_OBJECT_ID
CKA_GOST28147_PARAMS[2,3,5]	Byte array	, 28147-89 (OID) DER-. , CKK_GOSTR28147 CKA_OBJECT_ID.

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_UnwrapKey**

4 **CKA_SENSITIVE**, CK_TRUE **CKA_EXTRACTABLE** CK_FALSE.

5 **C_SetAttributeValue**.

, 34.10-2012 . , 34.10-2012.

34.10-2012 (512)

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410_512;
CK_UTF8CHAR label[] = {"A GOST R34.10-2012 private_key object"};
CK_BYTE keyPairIdGost_512[] = {"GOST R 34.10-2012(512) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410_512[] = {0x06, 0x09, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x02, 0x01, 0x02, 0x01};
CK_BYTE parametersGostR3411_512[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x03 };
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
        {CKA_ID, &keyPairIdGost_512, sizeof(keyPairIdGost_512)-1},
        {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
        {CKA_GOSTR3410PARAMS, parametersGostR3410_512, sizeof(parametersGostR3410_512)},
        {CKA_GOSTR3411PARAMS, parametersGostR3411_512, sizeof(parametersGostR3411_512)},
        {CKA_VALUE, value, sizeof(value)}
};
```

- CKO_SECRET_KEY, CKK_GENERIC_SECRET ( )
- CKO_SECRET_KEY, CKK_GOST ( )
- CKO_SECRET_KEY, CKK_GOST28147 ( )

(**CKO_SECRET_KEY**) . , .

(Common Secret Key Attributes)		
CKA_SENSITIVE[1,2]	CK_BBOOL	CK_TRUE, ( ). CK_FALSE
CKA_ENCRYPT[1]	CK_BBOOL	CK_TRUE, [3]
CKA_DECRYPT[1]	CK_BBOOL	CK_TRUE, [3]
CKA_SIGN[1]	CK_BBOOL	CK_TRUE, ( ), [3]
CKA_VERIFY[1]	CK_BBOOL	CK_TRUE, ( ), [3]
CKA_WRAP[1]	CK_BBOOL	CK_TRUE, (, ..)[3]
CKA_UNWRAP[1]	CK_BBOOL	CK_TRUE, (, ..)[3]
CKA_EXTRACTABLE[1,4]	CK_BBOOL	CK_TRUE [3]
CKA_ALWAYS_SENSITIVE[5,6,7]	CK_BBOOL	CK_TRUE, CKA_SENSITIVE CK_TRUE
CKA_NEVER_EXTRACTABLE[5,6,7]	CK_BBOOL	CK_TRUE , CKA_EXTRACTABLE CK_TRUE
CKA_CHECK_VALUE	Byte array	
CKA_WRAP_WITH_TRUSTED[2]	CK_BBOOL	CK_TRUE, CKA_TRUSTED CK_TRUE. CK_FALSE.
CKA_TRUSTED[8]	CK_BBOOL	CKA_WRAP_WITH_TRUSTED CK_TRUE.
CKA_WRAP_TEMPLATE	CK_ATTRIBUTE_PTR	. , . *ulValueLen* CK_ATTRIBUTE.

CKA_UNWRAP_TEMPLATE	CK_ATTRIBUTE_PTR	. , . *ulValueLen* CK_ATTRIBUTE.
, (Rutoken Vendors Defined Secret Key Attributes)		
CKA_SECRET_KEY_RSF_ID		RSF–, (0)

1 **C_SetAttributeValue**

2 CK_TRUE read-only

3

4 CK_FALSE read-only

5 **C_CreateObject**

6 **C_GenerateKey C_GenerateKeyPair**

7 **C_UnwrapKey**

8 CK_TRUE

CKA_SENSITIVE CK_TRUE **CKA_EXTRACTABLE** CK_FALSE, . .

(key check value — KCV) **CKA_CHECK_VALUE**, 3 . , , .

:

1. .
2. CKA_CHECK_VALUE - .
3. . CKA_CHECK_VALUE. , .

CKA_CHECK_VALUE , , , . (CKA_ENCRYPT CK_FALSE).

(,), , ; CKR_ATTRIBUTE_VALUE_INVALID.

, « » ( ). C_GetAttributeValue. C_SetAttributeValue , « ».

, , (0x00) (ECB, electronic codebook), .

, **CKA_SECRET_KEY_RSF_ID**, , 0.

CKA_VENDOR_KEY_PIN_ENTER CKA_VENDOR_KEY_CONFIRM_OP PINPad.

CKO_SECRET_KEY, CKK_GENERIC_SECRET ( )

(CKO_SECRET_KEY, CKK_GENERIC_SECRET) , .

, HMAC. , .

CKO_SECRET_KEY CKK_GENERIC_SECRET CKO_SECRET_KEY, .

CKO_SECRET_KEY, CKK_GENERIC_SECRET

(Generic Secret Key Attributes)		
CKA_VALUE[1,2,3,4]	Byte array	()
CKA_VALUE_LEN[5,6]	CK_ULONG	

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_UnwrapKey**

4 **CKA_SENSITIVE**, CK_TRUE **CKA_EXTRACTABLE** CK_FALSE.

5 **C_CreateObject**

6 **C_GenerateKey C_GenerateKeyPair**

```
CK_OBJECT_CLASS your_class = CKO_SECRET_KEY;
CK_KEY_TYPE keyType = CKK_GENERIC_SECRET;
CK_UTF8CHAR label[] = "A generic secret key object";
CK_BYTE value[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_DERIVE,, &IsTrue, sizeof(IsTrue)},
    {CKA_VALUE, value, sizeof(value)}
};
```

CKO_SECRET_KEY, CKK_GOST ( )

CKO_SECRET_KEY CKK_GOST CKO_SECRET_KEY. ( ) CKK_GOST

CKO_SECRET_KEY CKK_GOST rtPKCS11 2.30.

CKO_SECRET_KEY, CKK_GOST

, (Rutoken Vendors Defined CKK_GOST Object Attributes)		
CKA_VALUE	Byte Array	
CKA_GOST_KEY_OPTIONS		28147-89 (: , ,)
CKA_GOST_KEY_FLAGS		28147-89

CKO_SECRET_KEY, CKK_GOST28147 ( )

GOST 28147-89 (CKO_SECRET_KEY, CKK_GOST28147) CKO_SECRET_KEY, , .

CKO_SECRET_KEY CKK_GOST28147 rtPKCS11 2.30.

CKO_SECRET_KEY, CKK_GOST28147

28147 (GOST 28147 Attributes)		
CKA_VALUE[1,2,3,4]	Byte array	32 , ,
CKA_GOST28147_PARAMS[1,5,6]	Byte array	28147 (OID) DER-. , CKK_GOST28147 CKA_OBJECT_ID
, (Rutoken Vendors Defined CKK_GOST Object Attributes)		
CKA_SECRET_KEY_RSF_ID		RSF-,

1 **C_CreateObject.**

2 **C_GenerateKey C_GenerateKeyPair**

3 **C_UnwrapKey**

4 CKA_SENSITIVE, CK_TRUE CKA_EXTRACTABLE CK_FALSE.

5 **C_GenerateKey C_GenerateKeyPair**

6 **C_UnwrapKey**

, , () C_CreateObject 28147-89, 34.10-2001 34.10-2012 (CKA_TOKEN = TRUE).

28147-89

```
CK_OBJECT_CLASS your_class = CKO_SECRET_KEY;
CK_KEY_TYPE keyType = CKK_GOST28147;
CK_UTF8CHAR label[] = "A GOST 28147-89 secret key object";
CK_BYTE value[32] = {...};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x01};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ENCRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_DECRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_GOST28147PARAMS, parametersGost28147,sizeof(parametersGost28147)},
    {CKA_VALUE, value, sizeof(value)}
};
```