Cryptosense
Master your Cryptographic Infrastructure

CS ref	Criterion	Criticality	Explanation of criticality	v2.20 page	v2.20 section
1	Default value of TOKEN should be FALSE	M	If not compliant, the application could inadvertently generate keys that will be preserved on the token, exposing them to a future (perhaps compromised) session.	p.71	10.4
2	Default value of MODIFIABLE should be TRUE	M	If not compliant, the application could generate a key that it cannot later protect (e.g. by marking as unextractable)	p.71	10.4
3	TRUSTED can only be set to CK_TRUE by SO	H	If not compliant, a compromised session could set the TRUSTED attribute, causing untrusted certificates to be treated as trusted, and WRAP_WITH_TRUSTED keys could be wrapped under untrusted ones	footnote 10 p.66, used p.73 (certificates), p.81 (public keys) and p.85 (secret keys)	10.6
4	LOCAL must be NONE for CreateKey, UnwrapKey and GenerateKey(Pair)	M	If not compliant, device may create keys marked as LOCAL using key material generated outside the device.	p.79	10.7
5	LOCAL true iff key was Generated local or is a copy of a LOCAL key	M	If not compliant, device may contain keys marked as LOCAL using key material generated outside the device.	p.79	10.7
6	EXTRACTABLE sticky off	H	If EXTRACTABLE can be changed to TRUE, a previously protected key can be exported, directly breaking the security properties described in section 7 of the standard.	p.82-85	10.9-10.10
7	SENSITIVE sticky on	H	If SENSITIVE can be changed to FALSE, a previously protected key can be read by the GetAttribute command, directly breaking the security properties described in section 7 of the standard.	p.82-85	10.9-10.10
8	NEVER_EXTRACTABLE must not be given (Create, Unwrap, Generate(Pair))	M	If not compliant, NEVER_EXTRACTABLE may be set with a misleading value giving a false assessment of security of a key.	p.82-85	10.9-10.10
9	ALWAYS_SENSITIVE must not be given (Create, Unwrap, Generate(Pair))	M	If not compliant, ALWAYS_SENSITIVE may be set with a misleading value giving a false assessment of security of a key.	p.82-85	10.9-10.10
10	WRAP_WITH_TRUSTED sticky on	H	If not compliant, a key which has previously been marked as WRAP_WITH_TRUSTED may later be wrapped under an untrusted key-encryption key.	p.82-85	10.9-10.10

11	ALWAYS_SENSITIVE true if key has always been sensitive	M	If not compliant, ALWAYS_SENSITIVE may have a misleading value giving a false assessment of security of a key.	p.82.85	10.9-10.10
12	NEVER_EXTRACTABLE true if key has never been extractable	M	If not compliant, NEVER_EXTRACTABLE may have a misleading value giving a false assessment of security of a key.	p.82-85	10.9-10.10
13	GetAttribute only returns values on non-SENSITIVE, EXTRACTABLE keys	H	If not compliant, device leaks the value of keys marked as SENSITIVE or un-EXTRACTABLE, breaking the security properties described in section 7 of the standard (see CVE-2010-3321)	p.30, p.194 (RSA), p.229 (DH), p.251 (Secret)	7, 12.1.3 (RSA), 12.4.4 (DH), 12.7 (Secret)
14	MODIFABLE is sticky on and off for SetAttribute	H	If non-compliant, the device may allow a key whose attributes have been set and protected in one session to be compromised in a future session.	p. 71	10.4
15	PRIVATE is sitcky on and off for SetAttribute	H	If non-compliant, the device may allow a key which has been marked as PRIVATE in one session to be compromised in a future session.	p. 71	10.4
16	TOKEN sticky on and off for SetAttribute	H	If non-compliant, the device may allow a key session key to be preserved and exposed in a future session, or a token key to be inadvertently removed.	p. 71	10.4
17	CERTIFICATE_TYPE must not be NONE when calling CreateObject on a certificate	L	The type (X509 public key, attribute or WTLS needs to be known).	p. 73	10.6
18	VALUE must not be NONE when calling CreateObject on X509 certificate	M	The value of the certificate needs to be defined.	p. 75	10.6.3
19	KEY_TYPE must be given when creating key with Unwrap or CreateObject	M	A non-compliant implementation may misinterpret given key material	p. 79	10.7
20	KEY_GEN_MECHANISM must be NONE when calling CreateObject, GenerateKey, UnwrapKey	L	The mechanism cannot be known for externally generated key material	p.80	10.7
21	KEY_GEN_MECHANISM has a value only if CKA_LOCAL is true	L	The mechanism cannot be known for externally generated key material	p.80	10.7
22	GetFunctionList can be called before calling C_Initialize	L	Functional compliance, no obvious security conesquences.	p106	11.5
23	C_InitPIN can only be called from a R/W SO Functions session	H	Only the security officer has the right to set a PIN without giving the old PIN	p.115	11.5

24	C_SetPIN can only be called from R/W public session or R/W SO functions or R/W User Functions state	H	If the device is in read-only mode, the PIN should not be changed	p. 116	11.5
25	ALWAYS_SENSITIVE is false on keys made by CreateObject	H	If the key material has come from outside the device, it cannot be considered ALWAYS_SENSITIVE. A non compliant implementation would give a false measure of security.	p.128	11.7
26	NEVER_EXTRACTABLE is false on keys made by CreateObject	H	If the key material has come from outside the device, it cannot possible be considered NEVER_EXTRACTABLE. A non-compliant implementation would give a false measure of security.	p.128	11.7
27	In a read-only session, only session objects can be made	H	A non-compliant device would allow unauthorised creation of token keys and certificates	p. 130	11.7
28	If the normal user is not logged in, only public objects can be created	H	A non-compliant device would allow unauthorised creation of private keys	p. 130	11.7
29	In a read-only session, only session objects can be modified by SetAttribute	H	A non-compliant device would allow unauthorised manipulation of token objects	p. 136	11.7
30	In a public session, FindObjects only reveals public objects	H	A non-compliant device would reveal private objects to unauthrized applications	p. 138	11.7
31	C_EncryptInit only succeeds if ENCRYPT is set to true	H	A non-compliant device would allow encryption to be carried out by a key which does not have the correct permission to do this operation.	p. 139	11.8
32	C_DecryptInit only succeeds if DECRYPT is set to true	H	A non-compliant device would allow decryption to be carried out by a key which does not have the correct permission to do this operation.	p.144	11.9
33	C_SignInit only succeeds if SIGN is true	H	A non-compliant device would allow signatures to be created by a key which does not have the correct permission to do this operation.	p.152	11.11
34	C_SignRecoverInit succeeds only if SIGN_RECOVER is true	H	A non-compliant device would allow signatures to be created by a key which does not have the correct permission to do this operation.	p.156	11.11
35	C_VerifyInit succeeds only if VERIFY is true	H	A non-compliant device would allow signature verification to be carried out by a key which does not have the correct permission to do this operation.	p.157	11.12

36	C_VerifyRecoverInit succeeds only if VERIFYRECOVER is true	H	A non-compliant device would allow signature verification to be carried out by a key which does not have the correct permission to do this operation.	p.177	11.12
37	C_Wrap succeeds only if wrapping key has WRAP set to TRUE	H	A non-compliant device would allow a key to be wrapped under a key-encrypting key which does not have the correct permission	p.178	11.14
38	C_Wrap succeeds only if wrapped key has EXTRACTABLE set tot TRUE	H	A non-compliant device would allow keys marked as being unextractable to be exported from the device, breaking the security properties described in section 7 of the standard.	p.178	11.14
39	C_Unwrap succeeds only if UNWRAP is set to true	H	A non-compliance device would allow encrypted keys to be imported under a key-encrypting key which does not have the correct permissions.	p.180	11.14
40	Keys created by C_UnwrapKey have NEVER_EXTRACTABLE false	H	If the key material has come from outside the device, it cannot possible be considered NEVER_EXTRACTABLE. A non-compliant implementation would give a false measure of security.	p. 180	11.14
41	Keys created by C_UnwrapKey have ALWAYS_SENSITIVE false	H	If the key material has come from outside the device, it cannot possible be considered ALWAYS_SENSITIVE. A non compliant implementation would give a false measure of security.	p. 180	11.14
42	Keys created by C_UnwrapKey have EXTRACTABLE true by default	M	A non-compliant device marking keys generated outside the device as unextractable by default may give a misleading assurance of security	p. 180	11.14
43	RSA private keys: modulus must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p. 194	12.1.3
44	RSA private keys: modulus must not be specified when using GenerateKeyPair	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3
45	RSA private keys: modulus must not be specified when calling Unwrap key	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3

46	RSA private keys: public exponent must not be specified when calling GenerateKeyPair	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3
47	RSA private keys: public exponent must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3
48	RSA private keys: private exponent must not be specified when calling GenerateKeyPair	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3
49	RSA private keys: private exponent must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 194	12.1.3
50	RSA private keys: private exponent must be specified when calling CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p. 194	12.1.3
51	RSA Public keys : modulus must be specified when calling CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p. 193	12.1.2
52	RSA Public keys : modulus must not be specified when calling GenerateKeyPair	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p. 193	12.1.2
53	RSA public keys: MODULUS_BITS must be specified when calling GenerateKey	L	A non-compliant device would have ill-defined behavior.	p. 193	12.1.2
54	RSA public keys: MODULUS_BITS must not be specified when calling CreateObject	L	A non-compliant device would have ill-defined behavior.	p.193	12.1.2
55	RSA public keys: public exponent must be specified when calling CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p. 193	12.1.2
56	RSA keys: the default public exponent (which can be omitted) for GenerateKey is 65537.	H	A default value of 3 leaves certain padding modes available in PKCS#11 open to attack.	p. 196	12.1.4
57	Diffie-Hellman public key: PRIME must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.227	12.4.2
58	Diffie-Hellman public key: PRIME must be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.227	12.4.2
59	Diffie-Hellman public key: BASE must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.227	12.4.2

60	Diffie-Hellman public key: BASE must be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.227	12.4.2
61	Diffie-Hellman public key: VALUE must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.227	12.4.2
62	Diffie-Hellman public key: VALUE must not be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.227	12.4.2
63	Diffie-Hellman private key: PRIME must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.229	12.4.4
64	Diffie-Hellman private key: PRIME must not be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4
65	Diffie-Hellman private key: PRIME must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4
66	Diffie-Hellman private key: BASE must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.229	12.4.4
67	Diffie-Hellman private key: BASE must not be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4
68	Diffie-Hellman private key: BASE must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4
69	Diffie-Hellman private key: VALUE must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.229	12.4.4
70	Diffie-Hellman private key: VALUE must not be specified when using GenerateKey	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.229	12.4.4
71	Diffie-Hellman private key: VALUE must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.229	12.4.4
72	Diffie-Hellman private key: VALUE_BITS must not be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4
73	Diffie-Hellman private key: VALUE_BITS must not be specified when calling UnwrapKey	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.229	12.4.4

74	DH domain parameters: PRIME must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.231	12.4.6
75	DH domain parameters: PRIME must not be specified when using Generate	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.231	12.4.6
76	DH domain parameters: BASE must be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.231	12.4.6
77	DH domain parameters: BASE must not be specified when using Generate	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.231	12.4.6
78	DH domain parameters: PRIME_BITS must not be specified when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a fixed key for all calls	p.231	12.4.6
79	DH domain parameters: PRIME_BITS must be specified when using Generate	M	A non-compliant device would have ill-defined behavior and may create a weak key	p.231	12.4.6
80	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): if base key has ALWAYS_SENSITIVE false, so does derived key	H	In DH key derivation, the base (private) key is combined with a public value, so if the base value is not ALWAYS_SENSITIVE, one cannot consider the derived key to be ALWAYS_SENSITIVE	p. 235	12.4.11
81	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): if base key has ALWAYS_SENSITIVE true, derived key gets ALWAYS_SENSITIVE equal to value of SENSITIVE	H	A non-compliant implementation may give a key ALWAYS_SENSITIVE true when SENSITIVE is false, leading a to a false evaluation of security	p.235	12.4.11
82	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): derived key may be given SENSITIVE true or false	L	Functional compliance, no obvious security conesquences.	p.235	12.4.11
83	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): derived key may be given EXTRACTABLE true or false	L	Functional compliance, no obvious security conesquences.	p.235	12.4.11
84	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): if base key has NEVER_EXTRACTABLE true, derived key gets NEVER_EXTRACTABLE equal to opposite value of EXTRACTABLE	H	A non-compliant implementation may give a key NEVER_EXTRACTABLE true when EXTRACTABLE is TRUE, leading to a false evaluation of security	p.235	12.4.11
85	PKCS#3 DH key derivation (CKM_DH_PKCS_DERIVE): if base key has NEVER_EXTRACTABLE false, so does derived key	H	In DH key derivation, the base (private) key is combined with a public value, so if the base value is not NEVER_EXTRACTABLE, one cannot consider the derived key to be NEVER_EXTRACTABLE	p.235	12.4.11

86	Wrapping a private key under a secret key requires CBC mode and PKCS padding	M	Use of another mode would lead to ill defined behaviour as key may not be an even number of blocks in length	p.250	12.6
87	Secret key (Generic, AES, Twofish, Blowfish..) : value must be given when using CreateObject	M	A non-compliant device would have ill-defined behavior and may create a weak key	p251 (generic), p269 (AES), p275 (DES), p368 (Blowfish), p370 (Twofish)	12.7 (generic), 12.12.3 (AES), 12.13.2 (DES)
88	Secret key: value must not be given when using GenerateKey	L	A non-compliant device would have ill-defined behavior.	p.251	12.7
89	Secret key: value must not be given when using UnwrapKey	L	A non-compliant device would have ill-defined behavior.	p.251	12.7
90	Secret key : VALUE_LEN must not be given when using CreateObject	L	A non-compliant device would have ill-defined behavior.	p.251	12.7
91	Secret key : VALUE_LEN must be given when using GenerateKey	L	A non-compliant device would have ill-defined behavior.	p.251	12.7
92	AES-CBC_PAD: when unwrapping, VALUE_LEN should be none	M	AES length should be determined from encrypted data, otherwise a long AES key could potentially be imported as two short AES keys	p.272	12.12.6
93	Keys made by DeriveKey should have LOCAL=FALSE	H	LOCAL must indicate that a key was generated on the device	p. 182	11.14
94	SHA_*_KEY_DERIVATION results in ALWAYS_SENSITIVE FALSE if base key has ALWAYS_SENSITIVE FALSE	H	In SHA_* key derivation, the base key is combined with a public value in a public operation, so if the base value is not ALWAYS_SENSITIVE, one cannot consider the derived key to be ALWAYS_SENSITIVE	p. 314	12.21.5
95	Keys created by SHA_*_KEY_DERIVATION can have SENSITIVE set to true or false	L	Functional compliance, no obvious security conesquences.	p. 314	12.21.5
96	Keys created by SHA_*_KEY_DERIVATION can have EXTRACTABLE set to true or false	L	Functional compliance, no obvious security conesquences.	p. 314	12.21.5
97	SHA_*_KEY_DERIVATION results in NEVER_EXTRACTABLE FALSE if base key has NEVER_EXTRACTABLE FALSE	H	In SHA_* key derivation, the base key is combined with a public value in a public opation, so if the base value is not NEVER_EXTRACTABLE, one cannot consider the derived key to be NEVER_EXTRACTABLE	p. 314	12.21.5

98	SHA_*_KEY_DERIVATION: if base key has ALWAYS_SENSITIVE TRUE, result has ALWAYS_SENSITIVE equal to value of SENSITIVE	H	A non-compliant implementation may give a key ALWAYS_SENSITIVE true when SENSITIVE is false, leading a to a false evaluation of security	p.314	12.21.5
99	SHA_*_KEY_DERIVATION: if base key has NEVER_EXTRACTABLE TRUE, result has NEVER_EXTRACTABLE equal to opposite value of EXTRACTABLE	H	In DH key derivation, the base (private) key is combined with a public value, so if the base value is not NEVER_EXTRACTABLE, one cannot consider the derived key to be NEVER_EXTRACTABLE	p.314	12.21.5
100	SHA_1_KEY Derivation - trying to derive a key longer than 20 bytes, e.g DES3, results in an error	M	Since the output of SHA-1 is 20 bytes, deriving a longer key might result in a weak key	p.314	12.21.5
101	PBKDF2: CKM_PKCS5_PBKD2 templates must contain KEY_TYPE	L	A non-compliant device would have ill-defined behavior.	p. 324	12.26.10
102	PBKDF2: CKM_PKCS5_PBKD2 templates must contain KEY_VALUE_LEN unless type is of fixed length	L	A non-compliant device would have ill-defined behavior.	p. 324	12.26.10
103	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: if either of the base keys are SENSITIVE, the result is SENSITIVE	H	Concatenation and XOR are easily reversible, hence a non-compliant device would expose a SENSITIVE base key	p.358	12.34.3
104	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY : if either of the base keys are EXTRACTABLE false, the result is EXTRACTABLE false	H	Concatenation and XOR are easily reversible, hence a non-compliant device would expose a non-EXTRACTABLE base key	p.358	12.34.3
105	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: ALWAYS_SENSITIVE is true if and only if both base keys have ALWAYS_SENSITIVE true	H	A non-compliant implementation could give ALWAYS_SENSITIVE TRUE to a key derived from at least one which is not ALWAYS_SENSITIVE, giving a false measure of security	p.358	12.34.3

106	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: NEVER_EXTRACTABLE is false if and only if both base keys have NEVER_EXTRACTABLE false	H	A non-compliant implementation could give NEVER_EXTRACTABLE TRUE to a key derived from at least one which is not NEVER_EXTRACTABLE, giving a false measure of security	p.358	12.34.3
107	CKM_CONCATENATE_BASE_AND_DATA, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: if base key is SENSITIVE, the result is SENSITIVE	H	Concatenation and XOR are easily reversible, hence a non-compliant device would expose a SENSITIVE base key	p.359	12.34.4
108	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: if base key is EXTRACTABLE false, the result is EXTRACTABLE false	H	Concatenation and XOR are easily reversible, hence a non-compliant device would expose a non-EXTRACTABLE base key	p.359	12.34.4
109	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: ALWAYS_SENSITIVE is true if and only if base key has ALWAYS_SENSITIVE true	H	A non-compliant implementation could give ALWAYS_SENSITIVE TRUE to a key derived from at least one which is not ALWAYS_SENSITIVE, giving a false measure of security	p.359	12.34.4
110	CKM_CONCATENATE_BASE_AND_KEY, CKM_CONCATENATE_DATA_AND_BASE, CKM_XOR_BASE_AND_DATA, EXTRACT_KEY_FROM_KEY: NEVER_EXTRACTABLE is false if and only if base key has NEVER_EXTRACTABLE false	H	A non-compliant implementation could give NEVER_EXTRACTABLE TRUE to a key derived from at least one which is not NEVER_EXTRACTABLE, giving a false measure of security	p.359	12.34.4
111	Session objects are really deleted at the end of a session	H	A non-compliant implementation could leave objects exposed to a future compromised session	p.16	6.5
112	Removing token closes session and removes session objects	H	A non-compliant implementation could leave objects exposed to a future compromised session	p. 22	6.7.4
113	A given object has a single value for each attribute it possesses	M	A non-compliant implementation would have ill-defined behaviour	p. 63	10
114	DestroyObject can only delete session objects in a read-only session	M	A non-compliant implementation would allow destruction of TOKEN objects in a read only session, a possible denial of service attack	p.131	11.7

115	DestroyObject cannot delete private objects unless the user is logged in	M	A non-compliant implementation would allow destruction of private objects in an unauthenticated session, a possible denial of service attack	p.131	11.7
116	When creating objects by GenerateKey, CreateObject, UnwrapKey or DeriveKey, the attributes of the object created should match those given in the template or an error TEMPLATE_INCONSISTENT should be given	H	If attributes do not match, an application may inadvertently create keys without adequate protection, or with excessive permissions	p.63	10.1.1
117	When creating objects by GenerateKey, CreateObject, UnwrapKey or DeriveKey, giving two different values for the same attribute should result in TEMPLATE_INCONSISTENT	M	If an error is not given, resulting behaviour is undefined and may lead to the application making errors with key protection and permissions	p.64	10.1.1
118	Trusted certificates cannot be modified	H	The TRUSTED attribute is assigned to a certificate and all its attributes. If one change sit can no longer be considered TRUSTED	p.73	10.6.2