Tanúsítási jelentés

Hung-TJ-0013-2004

az nShield F3 PCI,
az nShield F3 Ultrasign PCI és
az nShield F3 Ultrasign 32 PCI
kriptográfiai adapter

kriptográfiai modulokról

/nCipher Corporation Limited/

/hardver verzió: nC4032P-150, ,
nC4032P-300, nC4132P-300
förmver verzió: 2.0.0 és 2.0.2/

Tartalom

1. A Tanúsítási jelentés tárgya, feladata és hatóköre

Jelen Tanúsítási jelentés tárgya az nShield F3 PCI kriptográfiai adapter, melyet minősített hitelesítés-szolgáltatás nyújtásához kapcsolódó különböző feladatok ellátására kívánnak felhasználni, mint „biztonságos" kriptográfiai modult..

A minősített hitelesítés-szolgáltatókra vonatkozó funkcionális és biztonsági követelményeket meghatározó EU-s dokumentumok (CEN 14167-1 munkacsoport egyezmény: "Elektronikus aláírásokhoz tanúsítványokat kezelő megbízható rendszerekre vonatkozó biztonsági követelmények", ETSI TS 101 456: "Minősített tanúsítványokat kibocsátó hitelesítés-szolgáltatókra vonatkozó szabályozási követelmények") és hazai jogszabályok (köztük legrészletesebben a 2/2002 (IV.26) MeHVM irányelve a minősített elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó biztonsági követelményekről) irányadók jelen Tanúsítási jelentéshez.

Ezen követelmények közül az egyik meghatározó fontosságú (mely több más követelményre is hatással van) elvárja, hogy a minősített hitelesítés-szolgáltatók[1] által használt kriptográfiai modul tanúsítvánnyal igazoltan feleljen meg az alábbi szabványok legalább egyikének:

- [FIPS 140-1], 3-as (vagy magasabb) biztonsági szint,
- [CEN:HSM-PP] (CMCSO-PP és CMCKG-PP[2]),
- [ITSEC] E3/high (vagy magasabb) biztonsági szint.

Az nShield F3 PCI kriptográfiai adapter FIPS 140-2 3-as szintű tanúsítvánnyal rendelkezik.

A FIPS 140-2 3-as biztonsági szintje igen szigorú követelményrendszert támaszt az általános célú kriptográfia modulok részére. Ugyanakkor nem tartalmaz számos olyan funkcionális és biztonsági követelményt, melyet a minősített hitelesítés-szolgáltatóknak ki kell elégíteniük saját kriptográfiai moduljukkal.

A fentiekből következően a jelen Tanúsítási jelentés fő feladata annak megállapítása, hogy:

- az nShield F3 PCI kriptográfiai adapter alkalmas-e minősített hitelesítés-szolgáltatás nyújtásához való alkalmazásra, s ha igen, akkor mely kapcsolódó feladatokhoz használható,
- a FIPS 140-2 szerinti Tanúsítvány érvényessége, illetve a többi kielégítendő funkcionális és biztonsági követelmény teljesülése milyen korlátozásokat, feltételeket támaszt a kriptográfiai modul használatára.

Jelen Tanúsítási jelentés hatóköre ugyanakkor csak a minősített hitelesítés-szolgáltatás nyújtásához való alkalmasságra és ennek feltétel-rendszerének meghatározására szorítkozik. Nem terjed ki az nShield F3 PCI kriptográfiai adapter egyéb, köztük a FIPS 140-2 Tanúsítvánnyal igazolt tulajdonságaira, beleértve az alábbiakat:

- A FIPS 140-es Tanúsítvány érvényességébe tartozó, FIPS által jóváhagyott titkosító algoritmusra /AES, DES, DES MAC, Triple DES, Triple DES MAC, RSA (kódolás, dekódolás), HMAC-SHA-1/,
- az nShield F3 PCI adapter által megvalósított azon kriptográfiai algoritmusokra, melyek nem FIPS által jóváhagyott algoritmusok, s így már a FIPS értékelés sem terjedt ki rájuk /Arc Four, Blowfish, CAST 5, CAST6, Serpent, SEED, Twofish, Diffie-Hellman (kulcsegyeztetés), El-Gamal, KCDSA, HSA-160, MD2, MD5, RIPEMD 160, SHA-256, SHA-384, SHA-512, MD2, MD5, SSL master key derivation, PKCS #8 key wrapping/.

[1] A követelmény nem minősített hitelesítés-szolgáltatóra is vonatkozik.

[2] Ez utóbbinak csak akkor, ha a minősített hitelesítés-szolgáltató biztosít aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatást is.

A Tanúsítási jelentés további szerkezete a következő:

- Az nShield F3 PCI adapter legfontosabb tulajdonságainak összefoglalása (2. fejezet).
- A FIPS Tanúsítvány eredményeinek összefoglalása (3. fejezet).
- A FIPS 140-2-nek való megfelelőségből (3-as biztonsági szintből) adódó, kielégített követelmények /külön tárgyalva az értékelés követelményeit, s az értékeléshez megkövetelt fejlesztői bizonyítékokat / (4. és 5. fejezetek).
- A FIPS követelményrendszerén túlmutató, minősített hitelesítés-szolgáltatókra vonatkozó funkcionális és biztonsági követelmények (6. fejezet).
- A minősített hitelesítés-szolgáltatás nyújtáshoz való alkalmasság megállapítása, valamint az alkalmazás feltételeinek és korlátainak a meghatározása (7. fejezet).
- A jelen Tanúsítási jelentéshez figyelembe vett dokumentumok jegyzéke (8. fejezet).
- Felhasznált rövidítések jegyzéke (9. fejezet).

2. Az nShield F3 PCI legfontosabb tulajdonságainak összefoglalása

2.1 A kriptográfiai modul

Az nShield beavatkozás ellen védett Hardver Biztonsági Modulja olyan multitaszk hardver modul, mely modulo m aritmetikai műveleteket képes végezni nagy egész számokkal. Az nShield F3 PCI emellett még a kulcsmenedzsment protokollok teljes skáláját kínálja.

A modul az nCipher által nyújtott förmvert futtatja. A felhasználónak lehetősége van ezt a förmvert frissíteni. A NewEnquiry szolgáltatás segítségével határozható meg, hogy az adott modul milyen verziójú förmvert futtat. Ezen minősítés a 2.0.0 verziójú förmverre érvényes.

A modul inicializálható FIPS 140-2 3-as szinten, ami megfelel a minősítés követelményének. Ekkor a modul személyazonosság alapú hitelesítést használ. Az inicializációs paramétereket a NewEnquiry és a SignModuleState szolgáltatásokkal kérdezhető le. A modul futási módját a KeySafe vagy más parancssori programokkal lehet lekérdezni.

A modul olyan szolgáltatásokat is nyújt, amik nem kapcsolatosak tárolt kulcsokkal, így kívül esnek a FIPS 140-2 tanúsításon. Amikor ezek a szolgáltatások futnak, a modul nem FIPS 140-2 kompatibilis módban fut. Ilyen módról csak akkor beszélhetünk, ha a FIPS 140-2 által jóváhagyott szolgáltatás FIPS által elfogadott algoritmussal működik.

A modul PCI buszon keresztül csatlakozik a gazdaszámítógéphez. A modult különböző szoftvereken keresztül lehet elérni.

Az nShield modul a számítógép merevlemezén is tárol kulcsokat, ún. „key blobokban”. A modul nem-felejtő memóriát is tartalmaz. Léteznek olyan új szolgáltatások, melyek a memóriát, mint fájlt érnek el. A fájlokhoz Hozzáférési Engedély Listákon (ACL) keresztül lehet hozzáférni, mely egy fájlon belüli meghatározott bájtban állítható be. Ez a memória bizonyos bájtjait képessé teszi új szolgáltatások nyújtására. A modul tartalmaz még egy valósidejű órát is. Vannak olyan szolgáltatások, amik képesek azt olvasni és állítani.

A modul a következő operációs rendszerekkel keresztül használható:

- Windows NT 4.0
- Windows 2000
- Solaris
- HP-UX
- AIX
- Linux x86
- FreeBSD x86

2.2 Szerepkörök és szolgáltatások

Az nShield modul a következő szerepköröket támogatja:

User

Minden felhasználó először User módban csatlakozik az nShield modulhoz. Ebben az szerepkörben előzőleg készített tokeneket lehet betölteni és ezek segítségével a key blobok kulcsait felhasználni. Ezután az ACL-ben meghatározott keretek között lehet ezeket a kulcsokat kriptográfiai műveletekre felhasználni.

nCipher Security Officer

Az nCipher Security Officer (NSO) felelős a modul általános biztonságáért.

Az NSO egy kulcspár segítségével azonosítható, ez a K_{NSO}. A kulcs nyilvános részének lenyomatát az egység inicializáláskor eltárolja. Minden modul kulccsal vagy token írással kapcsolatos művelet végrehajtásához egy K_{NSO}-val aláírt tanúsítvány szükséges.

Az nCipher Security Officer felelős a felhasználók autentikációs tokeneinek (smart cardok) létrehozásáért, valamint ezek biztonságos kézbesítéséért.

Egy felhasználó akkor rendelkezik NSO szerepkörrel, ha nála van a K_{NSO} titkos kulcsa, valamint az a KeyID kulcsazonosító, ami ehhez a kulcshoz tartozik.

Junior Security Officer

Bizonyos esetekben a Security Officer át kívánja adni feladatait más adminisztrátoroknak (Junior Security Officer – JSO). Ezt egy kulcspár készítésével és a jogok hozzárendelésével meg tudja tenni. A JSO ezután minden olyan műveletet végre tud hajtani, ami a kulcsához tartozó ACL-ben le van írva. A JSO kulcsát egy más célra nem használt tokennel védett key blobban kell tárolni.

A JSO szerepkör igazolásához rendelkezni kell a JSO kulccsal, ennek KeyID kulcsazonosítójával, és amennyiben szükséges, a K_{NSO} kulccsal aláírt tanúsítvánnyal, ami az NSO feladatokat a JSO-nak delegálta.

A JSO hasonló módon továbbadhatja feladatai egy részét. Ha valamilyen fennhatóságot delegál a JSO, az új felhasználó is JSO lesz, ha nem, akkor csak User.

A modul a hagyományos kriptográfiai szolgáltatások mellett egyéb szolgáltatásokat is nyújt. Ezek a következők:

Állapotinformáció

A modul állapota a hátoldalán található LED-ekből kiolvasható.

Önteszt

A modul öntesztjei a törlés állapotában érhetők el. Ehhez vagy a Clear gombot kell a modulon megnyomni vagy a ClearUnit parancsot kell kiadni.

Modul állapot

A következő szolgáltatások változtatják meg a modul állapotát:

- Fail: olyan teszteredmény, ami a modul hibáját jelzi.
- Clear unit: törli a modult – megfelel a Clear gomb megnyomásának.[3]

Förmver frissítés

Az nShield modul förmverét csak az nCipher Förmver Integritás Kulccsal (Firmware Integrity Key) aláírt és a Förmver Bizalmasság Kulccsal (Firmware Confidentiality Key) titkosított új förmverrel lehet frissíteni. A modul továbbá ellenőrzi, hogy a telepítendő förmver újabb verziójú-e , mint az aktuálisan használt.[4]

2.3 Kulcsok

Az nShield modul minden kulcsra vagy annak leírójával, vagy egy tetszőleges számmal vagy az SHA-1 lenyomatával hivatkozik.

[3] Ezek a szolgáltatások csak adminisztrátori felhatalmazással érhetők el.

[4] FIPS által jóváhagyott módban csak akkor tud működni a modul, ha az új förmver is rendelkezik a FIPS minősítéssel.

Security Officer kulcs

Az nCipher Security Officer kulcsot az inicializáció során kell beállítani. Ez egy aszimmetrikus kulcspár, amit az nCipher Security Officer a kulcskezelés és más biztonsági műveletek engedélyezésére szolgáló tanúsítvány aláírására használ.

A kulcspár nyilvános részének SHA-1 lenyomatát a modul EEPROM memóriája tárolja.

A nyilvános kulcs nyílt szövegként megtalálható a tanúsítványban.

A modul a titkos kulcs tulajdonlása alapján tud valakit Security Officerként azonosítani.

Alapértelmezésben az nCipher modul DSA kulcsot használ erre a célra, azonban igény szerint beállítható RSA algoritmusú kulcsok használata is.

Junior Security Officer kulcs

Mivel az nCipher Security Officer kulccsal rengeteg feladat végezhető el, egyes feladatokat érdemes átadni a Junior Security Officer felhasználóknak.

A Junior Security Officer (JSO) felhasználó létrehozásához az NSO egy tanúsítványaláíró kulcsot generál. Ez a JSO kulcs. Ezt a kulcsot más alkalmazás kulcsokhoz hasonlóan logikai token védi.

A felelősség delegálásához az NSO-nak ezután egy Access Control Listet tartalmazó tanúsítványt kell készítenie, ami meghatározza az átadott jogokat és annak a JSO kulcsnak a lenyomatát, amire vonatkozik. A JSO ezután kulcsával és tanúsítványával minden olyan tevékenységet végre tud hajtani, ami az ACL-ben engedélyezve van számára.

Amennyiben a JSO kulcs ACL-e tartalmazza a Sign attribútumot, bizonyos tevékenységeit tovább tudja adni egy másik kulcs tulajdonosának, akinek ehhez rendelkeznie kell egy NSO és egy JSO által aláírt tanúsítvánnyal is. Amennyiben a JSO kulcs csak a UseAsCertificate tulajdonsággal rendelkezik, nem adhatja tovább hatásköreit.

Alapértelmezésben az nCipher modul DSA kulcsot használ erre a célra, azonban igény szerint beállítható RSA algoritmusú kulcsok használata is.

Hosszútávú aláíró kulcs

Az nShield modul egy 160 bites adatot tárol a Dallas 4320-as EEPROM memóriában, ami a modul förmverében tárolt diszkrét logaritmus csoport segítségével alkalmas a DSA kulcsok generálására.

Ez a kulcs a GenerateKLF szolgáltatás segítségével kicserélhető egy tetszőleges véletlen számra. Felhasználható a modul állapot tanúsítvány aláírására a SignModulState szolgáltatáson keresztül, nyilvános része pedig a Get Long Term Key szolgáltatáson keresztül lekérhető.

A kulcs nem használatos más adat titkosítására, csak arra, hogy a modul kriptográfiai azonosítását elvégezze, így felhasználhatóvá váljon a nyilvános kulcsú tanúsítványláncban. Az nCipher kibocsáthat olyan tanúsítványokat, amik bizonyítják, hogy a modul eredeti nCipher termék.

Modul aláíró kulcs

A modul inicializálásánál a modul egy DSA kulcspárt készít, ami a tanúsítvány aláírásnál használatos. A titkos kulcs az EEPROM-ban van tárolva, ahonnan semmilyen körülmények között nem lehet kiszedni. A nyilvános kulcs kinyerhető nyílt szövegként vagy titkosítva van tárolva különböző key blobokban. Ez a kulcs csak meghatározott modulok által generált tanúsítványok ellenőrzésére szolgál.

Modul kulcsok

A modul kulcsok Triple DES algoritmussal készülnek és a tokenek védelmére szolgálnak. Az első ilyen kulcs a modul inicializálásakor készül, ez a K_{M0}. A modul kulcs soha nem kerül ki a modulból.

A Security Officer további modul kulcsokat tölthet a modulba. Ezek készülhetnek a modulban, de akár külső forrásból is származhatnak. A modul kulcsnak kinevezett kulcsok az EEPROM-ban tárolódnak.

Miután egy kulcs modul kulcsnak lett kinevezve, nem lehet kinyerni a modulból. Egyedül a generáláskor lehet key blobként exportálni ezeket a kulcsokat.

Logikai tokenek

A logikai token egy Triple DES kulcs, amit a key blobokat védi.

A logikai token készítésénél meg kell határozni az olyan paramétereket, mint a megosztások száma, a token újragenerálásához szükséges megosztások száma és a használathoz szükséges megosztások száma. Az összes megosztás 1 és 64 között lehet, a használathoz szükséges megosztások száma pedig 1 és az összes megosztás között.

A logikai token mindig a modul véletlenszám generátora által generált véletlen érték.

Betöltés közben a logikai tokenek az objektum tárban helyezkednek el.

A token kulcsok csak fizikai vagy szoftveres tokenekre exportálhatók. Modul kulcs exportálása esetén a logikai token (a Triple DES kulcs és a token paraméterek) először a modul kulccsal lesz titkosítva. A titkosított token ezután a Shamir Secret Sharing[5] algoritmus segítségével egy vagy több megosztáshoz lesz rendelve. Ezután minden egyes megosztás a megosztás kulccsal lesz titkosítva és egy fizikai tokenre – intelligens kártyára – vagy szoftver tokenre kerül. A logikai tokenek egy vagy több fizikai token között oszthatók meg. A tokenek tulajdonságai határozzák meg, hogy hány megosztás kell a token újragenerálásához. Megosztások csak a tokenekkel együtt generálhatók. A förmver megakadályozza, hogy egy megosztás többször legyen létrehozva.

Megosztás kulcs

A megosztás kulcs a logikai token megosztásokat védi, amikor azok intelligens kártyára vagy szoftver tokenre kerülnek. A kulcsot jelszónak is védenie kell. A megosztást titkosító egyedi Triple DES kulcsot a jelszó, a modul kulcs, a megosztás szám és az intelligens kártya azonosítója segítségével alakítják ki. Ez a Triple DES kulcs nem tárolódik a modulban, egy megosztás betöltésénél mindig újraszámolódik. A megosztás adat tartalmaz egy MAC kódot, amit hibás jelszó esetén nem lehet megkapni, így a megosztáshoz való hozzáférést meg lehet tiltani.

Belső út kulcsok

A belső út két modul közötti biztonságos csatornát jelent.

Egy belső út felállításához a két modul biztonságos kulcscserét hajt végre, a Diffie-Hellman algoritmus segítségével.

A kulcscsere paraméterek a modul aláíró kulcsokkal vannak aláírva. Miután a modulok hitelesítették az aláírásokat, a modul négy szimmetrikus kulcsot generál az SHA-1

[5] Shamir Titkos Megosztás

algoritmuson alapuló protokoll segítségével. A szimmetrikus kulcsok Triple DES-nek megfelelőek.

A négy kulcs titkosításra, dekódolásra, MAC készítésre és MAC hitelesítésre használatosak. A protokoll garantálja, hogy az a kulcs, amit az 1. modul titkosításra használ, azt a 2. modul dekódolásra.

Minden belső út kulcs objektumként az objektum tárban van tárolva a DRAM-ban.

Kulcs objektumok

A titkosításhoz, dekódoláshoz, aláírás ellenőrzéshez és digitális aláíráshoz használatos kulcsok objektumként jelennek meg a DRAM memóriában. Minden objektum egy véletlen azonosítóval rendelkezik, ami függ a viszonytól és a felhasználótól.

Minden kulcs egy Access Control Listtel (ACL) van tárolva, ami meghatározza, hogy az adott kulccsal milyen műveletek végezhetők el.

Új kulcs generálása vagy nyílt szövegű kulcs importálásakor az adott kulcs típusának megfelelő ACL-t kell generálni. Az ACL a SetACL szolgáltatás segítségével állítható be. Amennyiben az ACL rendelkezik az ExpandACL engedéllyel, később több engedélyt is be lehet állítani hozzá.

Amennyiben az ACL engedélyezi, a kulcs objektumok key blobként exportálhatók. Minden blob egy kulcsot és egy ACL-t tárol. Az ACL határozza meg, hogy a kulccsal milyen művelet végezhető el. A blobban tárolt ACL-nek legalább annyira szigorúnak kell lennie, mint annak az ACL-nek, ami ahhoz a kulcshoz tartozik, amiből a key blob lett generálva. A key blobból betöltött új kulcsobjektum ACL-e a key blobból származik. A key blobok a logikai tokenekben, titkosítva vannak tárolva. A kulcs objektumok key blobként exportálhatók a modulból.

A kulcs objektumok csak akkor exportálhatók nyílt szövegként, ha az ACL engedélyezi ezt. FIPS 140-2 3-as szinten a titkos kulcs exportálása nyílt szövegként nem megengedett.

Viszony kulcs

A modul igény szerint generálja a viszony kulcsokat. Ezek a kulcsok a hozzá tartozó ACL-lel együtt kulcs objektumként vannak tárolva. A kulcs bármilyen támogatott algoritmussal készülhet.

Archiváló kulcsok

Előfordul, hogy a kulcsról olyan mentést kell készíteni, amit egy másik kulcs véd. A kulcsok védhetők:

- Triple DES kulcsokkal, vagy
- Triple DES és RSA kulcsok kombinációjával. Ebben az esetben egy véletlen Triple DES kulcs titkosítja az archiválandó kulcsot, majd ezt egy RSA kulccsal becsomagolják.

A kulcsarchiválásnál az eredeti kulcs ACL-e is el lesz mentve.

Az archíváló kulcs generálásakor vagy importálásakor be kell állítani a UseAsKeyBlob opciót az ACL-ben. A kulcsot a modul kulcs objektumként kezeli.

Az archiválandó kulcs generálásakor vagy importálásakor engedélyezni kell az Archival opciót az ACL-ben. Ezzel az opcióval lehetőség van az archiváló kulcsok lenyomatának beállítására, ami lehetővé teszi, hogy csak bizonyos kulcsokkal legyen elvégezhető az archiválás.

Tanúsítvány aláíró kulcsok

A kulcs objektumhoz kapcsolódó ACL egy tevékenység engedélyezéséhez elvárhatja egy tanúsítvány meglétét. A tanúsítvány aláírásához szükséges lehet például a Security Officer kulcsa. Az, hogy milyen kulccsal legyen aláírva a tanúsítvány, az ACL-ben van meghatározva a kulcs SHA-1 lenyomatával. A kulcs típusa szintén az ACL-ben van meghatározva. Ez akármilyen algoritmussal készült kulcs lehet, de az nCipher eszközök a DSA kulcsokat használják.

Sok szolgáltatás az NSO által aláírt tanúsítványt várja el a végrehajtáshoz.

Förmver Integritás kulcs

Minden förmver egy DSA kulccsal van aláírva. A modul csak akkor fogad el új förmvert, ha az olyan kulccsal van aláírva, amit a tárolt nyilvános kulcs dekódolni és ellenőrizni tud.

A titkos kulcs az nCipher cég birtokában van.

A nyilvános kulcs az összes förmverben megtalálható. A förmver a flash memóriában található kikapcsolt állapotban, bekapcsoláskor töltődik be a DRAM-ba.

Förmver Bizalmasság Kulcs

Minden förmver egy Triple DES kulccsal van titkosítva, hogy megakadályozzák a visszafejtését.

A titkosító kulcs az nCipher cégnél és a förmverekben is megtalálható.

A förmver a flash memóriában található kikapcsolt állapotban, bekapcsoláskor töltődik be a DRAM-ba.

nCipher Mester Tulajdonság Engedélyező kulcs

Üzleti okokból az nCipher modulban nincs az összes szolgáltatás engedélyezve. Ezen szolgáltatások engedélyezéséhez a felhasználónak rendelkeznie kell az nCipher Mester Tulajdonság Engedélyező kulcs által aláírt tanúsítvánnyal. Ezzel lehet egy-egy speciális bitet beállítani az EEPROM-ban.

Az nCipher Mester Tulajdonság Engedélyező kulcs egy DSA kulcs, melynek titkos része az nCipher cégnél van, nyilvános része pedig minden förmverben megtalálható. A förmver a flash memóriában található kikapcsolt állapotban, bekapcsoláskor töltődik be a DRAM-ba.

2.4 Szabályok

Azonosítás és hitelesítés

Az nShield modullal való összes kommunikáció egy gazdaszámítógépen futó szerver programon keresztül történik, szabványos folyamatok közötti kommunikációval.

A modul használatához a felhasználónak először be kell lépnie a gazdaszámítógépre, és el kell indítania egy nCipher alkalmazást. Az alkalmazás a szerverprogramhoz kapcsolódik, ahonnan kap egy 120 bites tetszőleges azonosítót.

Bármilyen szolgáltatás igénybevétele előtt a felhasználónak igazolnia kell a megfelelő jogosultságát. Ahol többlépcsős azonosításra van szükség, minden egyes lépésnek azonos kapcsolaton keresztül kell megtörténnie.

Az nShield modul személyazonosság alapú hitelesítést használ. Minden egyes felhasználó rendelkezik egy intelligens kártyával, amin a hitelesítéshez szükséges adatai – a logikai token

megosztás – található titkosított formában. Minden művelet elvégzéséhez először be kell tölteni a logikai tokent az intelligens kártyáról.

Hozzáférés engedélyezés

A gazdaszámítógép merevlemezén titkosított formában tárolt kulcsok a key blobok. Ezen kulcsok használatához először azt a tokent kell betölteni, amivel a blob titkosítva lett.

A tokenek szétoszthatók megosztásokba. Minden egyes megosztás tárolható intelligens kártyán vagy a merevlemezen szoftver tokenekben. A megosztások emellett még védhető jelszóval és a modul kulccsal. Ennek következtében a felhasználó csak akkor tudja betölteni a kulcsot, ha birtokában van a megfelelő intelligens kártyának, a szükséges jelszónak és a modul kulcsnak a modulban kell lennie.

A modul kulcsok az EEPROM-ban vannak tárolva. Ezeket csak az nCipher Security Officer tudja betölteni vagy törölni, akit a modul inicializálásakor betöltött titkos kulcs azonosít. A modul törlése nélkül ezt nem lehet megváltoztatni, ekkor azonban minden modul kulcs törlődik.

A key blobok mellett ACL-ek is megtalálhatók, amik szabályozzák, hogy az adott kulccsal milyen szolgáltatás és mennyiszer hajtható végre. A végrehajtások száma lehet előre beállított vagy megújítható. Az előre beállított műveletszám átlépése után a kulcs az adott szolgáltatáshoz többet nem használható fel. Megújítható műveletszám esetén a felhasználónak a token újból betöltésével kell újrahitelesítenie a kulcsát.

Minden objektumra egy kezelő hivatkozik. A kezelők a kliens azonosítókra hivatkoznak. Amennyiben egy kliens olyan kezelőre hivatkozik, ami másik klienshez tartozik, a parancs nem lesz végrehajtva. Léteznek olyan szolgáltatások, amik a kezelőket a kliens azonosítók között mozgatják.

Hozzáférési Engedély Listák

Minden kulcs objektum rendelkezik Hozzáférési Engedély Listákkal (ACL). A felhasználónak akkor kell az ACL-eket meghatároznia, amikor a kulcsokat generálja vagy importálja. Az ACL lista tartalmazza az összes olyan feladatot, amit a kulccsal végre lehet hajtani. Ami nincs az ACL-ben, azt a kulccsal nem lehet végrehajtani. Az ACL csak akkor változtatható meg, ha tartalmazza a SetACL kapcsolót. Az ACL a kulccsal együtt van tárolva a blobokban.

Az ACL-ek szabályozzák, hogy az adott kulccsal milyen szolgáltatás és mennyiszer hajtható végre. A végrehajtások száma lehet előre beállított vagy megújítható. Az előre beállított műveletszám átlépése után a kulcs az adott szolgáltatáshoz többet nem használható fel. Megújítható műveletszám esetén a felhasználónak a token újból betöltésével kell újrahitelesítenie a kulcsát.

Az ACL meghatározhat egy tanúsítót is. Ebben az esetben a felhasználónak egy olyan tanúsítvánnyal kell rendelkeznie, amit olyan kulccsal írtak alá, aminek lenyomata megtalálható az ACL-ben.

Az ACL elkészítése bonyolult feladat, a felhasználóknak ezért rendelkezésre állnak olyan eszközök, amik megkönnyítik ezt.

Objektum újrafelhasználás

Minden objektumra egy kezelő hivatkozik. A modul memóriakezelése biztosítja, hogy egy meghatározott memóriaszegmens csak egy kezelőhöz tartozhat. A kezelő meghatározza az objektum típusát, a modul pedig ellenőrzi ezeket a típusokat. Amikor a kezelő megszűnik, az általa használt memóriaszegmens lenullázódik.

Hibaállapotok

Amennyiben a modul valamilyen átmeneti hiba miatt nem tud végrehajtani egy utasítást, egy parancsblokkot ad vissza adat nélkül, végén egy állapotszóval, ami a megfelelő hibakódot tartalmazza. A felhasználó később újra végrehajthatja a parancsot. A szerverprogram naplóz minden ilyen típusú hibát.

Amennyiben az nShield modul valamilyen visszavonhatatlan hibát vét, hibaállapotba kerül. Ekkor a modul hátoldalán levő LED sor a Morse kód szerinti SOS üzenet szerint villog. Ahogy a modul hibaállapotba kerül, minden processzor abbahagyja a működést, és nem fogad el utasításokat, újra kell indítani.

Biztonsági határvonal

A biztonsági határvonal a nyomtatott áramkör határa.

A következő elemek nem tartoznak a FIPS 140-2 tanúsítás alá, mivel biztonsági szempontból nem fontosak:

- PCI csatlakozó
- Jumperek
- Törlés gomb
- Állapot LED-ek
- Ventilátor
- Soros csatlakozó

2.5 Fizikai biztonság

Az nShield modul minden biztonságilag kritikus komponense epoxy gyanta bevonattal van védve.

A modul rendelkezik egy törlés gombbal. Ez a gomb öntesztelő módba teszi a modult, mely minden tárolt kulcsobjektumot és logikai tokent töröl, majd lefuttatja az öntéseket. A hosszú ideig érvényes biztonságkritikus paramétereket, modul kulcsokat, modul aláíró kulcsokat és a Security Officer kulcsot a modul gyári állapotába való visszatéréssel lehet törölni.

A fizikai biztonságról a következő rendszeres ellenőrzéssel lehet megbizonyosodni:

- Meg kell vizsgálni az epoxy gyanta bevonatot, hogy van –e rajta valamilyen sérülés.
- Az intelligens kártya olvasó kábele közvetlenül a modulba csatlakozik, és sehol sincs megbontva.

2.6 Függvények erőssége

Object ID-k támadása

A kapcsolatokat a ClientID azonosító különbözteti meg, ami egy 32 bites véletlen szám.

Az objektumokat a KeyID azonosítja, melyeket inkább ObjectID-nak kéne nevezni, mert már nem csak a kulcsokat azonosítja, de kompatibilitási okok miatt ez a megnevezés érvényes. Ez szintén egy 32 bites véletlen szám.

Ahhoz, hogy egy másik felhasználó által feltöltött kulcsot el lehessen érni, két véletlen 32 bites számot kéne kitalálni. A modul percenként 2^{16} számú műveletet tud végrehajtani, így a sikeres támadás valószínűsége egy percen belül $2^{16}/2^{64} = 2^{-48}$.

A tokenek támadása

Amennyiben a felhasználó a logikai tokenhez egy megosztást választ, jelszó nélkül és az intelligens kártyáját az olvasóban hagyja, más felhasználó is be tudja tölteni a logikai tokent. A modul nem képes meghatározni azt, hogy melyik felhasználó dugta be az intelligens kártyát az olvasóba.

- nem marad a kártya az olvasóban
  ha az intelligens kártya nem marad az olvasóban, a támadó csak akkor tud hozzáférni a logikai tokenhoz, ha kitalálja a token ClientID-ját és az ObjectID-ját.

- jelszó megkövetelése
  jelszóval ellátott megosztáshoz való hozzáféréshez a felhasználónak rendelkeznie kell a jelszó SHA-1 lenyomatával. A lenyomat kombinálva van a modul kulccsal, a megosztás számmal és az intelligens kártya azonosítójával, így lehet előállítani azt a kulcsot, ami a megosztás titkosításának feloldásához szükséges. Amennyiben a támadó nem rendelkezik a jelszóval, 2^{80} számú próbálkozás szükséges a megosztás betöltéséhez. A modul a sikertelen próbálkozás után 5 másodperces szünetet tart, csak ezután lehet újra próbálkozni.

- egynél több megosztás megkövetelése
  amennyiben a logikai tokenhez egynél több intelligens kártyához tartozó megosztás tartozik, a támadónak minden egyes megosztáshoz meg kell ismételnie a támadást.

A logikai tokenek 168 bites Triple DES kulcsok. A megosztásokat a modul kulcs, a megosztás száma és a kulcs azonosítója segítségével generált titkosítás védi. Annak a valószínűsége, hogy a modul kulcs és a megosztást védő kulcs generálási eljárásának ismerete nélkül lehessen generálni egy megfelelő logikai token kulcsot a megfelelő formában, 2^{-168}.

3. A FIPS Tanúsítvány eredményeinek összefoglalása

Az nShield F3 Adaptert egy kriptográfiai modulok tesztelésére az Egyesült Államokban és Kanadában akkreditált laboratórium[6] megvizsgálta, értékelte és tesztelte az alábbi követelményrendszernek való megfelelőség szempontjából:

> *a FIPS 140-2-ből (Kriptográfiai modulokra vonatkozó biztonsági követelmények) származtatott teszt követelmények*
> */Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules/*

A (FIPS) értékelés eredményei az alábbiak voltak:

A kriptográfiai modul tervezése:	3-as szint
Modul portok és interfészek:	3-as szint
Szerepkörök, szolgáltatások és hitelesítés:	3-as szint
Véges állapotú automata modell:	3-as szint
Fizikai biztonság /több chipes, beágyazott/:	3-as szint
Kriptográfiai kulcsgondozás:	3-as szint
Elektromágneses interferencia és kompatibilitás:	3-as szint
Ön-tesztek:	4-es szint
Tervezési biztosíték:	3-as szint
Más támadások enyhítése:	nincs értékelve
Működési környezet:	nincs értékelve
Tesztelve a következő konfigurációkkal:	nincs értékelve

Az értékelés az alábbi digitális aláíráshoz kapcsolódó, FIPS által jóváhagyott algoritmusok megvalósítását vizsgálta, tesztelte: **DSA/SHA-1, HMAC-SHA-1, RSA (PKCS #1),**

Az értékelés az alábbi titkosításhoz kapcsolódó[7], FIPS által jóváhagyott algoritmusok megvalósítását vizsgálta, tesztelte: **DES, DES MAC, Triple DES, Triple DES MAC, AES**

Az elért általános biztonsági szint: 3-as

[6] a DOMUS IT Laboratory /NVLAP LAB CODE 200017-0/
[7] jelen Tanúsítási jelentés hatókörén kívül álló,

4. Az nShield F3 értékelési követelményei a FIPS 140-2 szerint

Az alábbiakban áttekintjük azokat a (FIPS 140-2 követelményrendszer 3-as szintjéből fakadó) biztonsági követelményeket, melyeknek való megfelelőséget az nShield F3 értékelését végző laboratórium vizsgálta és igazolta.

Az alábbi jelölést alkalmazzuk:

KÖV_x.y: a FIPS 140-2 x. fejezetének y. biztonsági követelménye.[8]

4.1. A kriptográfiai modul tervezése és dokumentálása

KÖV_01.01:
A kriptográfiai modulnak tartalmaznia kell hardverek, szoftverek, förmverek halmazát vagy ezek olyan kombinációját, mely kriptográfiai funkciókat vagy eljárásokat valósítanak meg, beleértve ebbe a kriptográfiai algoritmusokat és esetlegesen a kulcsgenerálást is, mindezt egy meghatározott kriptográfiai határon belül.

KÖV_01.02:
A kriptográfiai modulnak legalább egy FIPS által jóváhagyott biztonsági funkciót kell megvalósítania, melyet FIPS által Jóváhagyott működési módban kell használnia.

KÖV_01.03:
A kezelőnek értesülnie kell arról, hogy a Jóváhagyott működési mód lett kiválasztva.

KÖV_01.04:
A modulnak jeleznie kell, hogy a FIPS által Jóváhagyott működési mód lett kiválasztva.

KÖV_01.05:
A kriptográfiai határnak tartalmaznia kell egy pontosan meghatározott vonalat, ami a kriptográfiai modul fizikai határát jelenti.

KÖV_01.06:
Ha a kriptográfiai modul szoftvert vagy förmvert tartalmaz, a kriptográfiai határt úgy kell definiálni, hogy az tartalmazzon minden olyan processzort, amely végrehajtja a szóban forgó kódot.

KÖV_01.07:
A következő dokumentálási követelményeknek meg kell felelnie minden hardvernek, szoftvernek és förmvernek, amiket a kriptográfiai modul tartalmaz.

KÖV_01.08:
A dokumentációnak teljes mértékben meg kell határoznia a kriptográfiai modul minden hardver, szoftver és förmver komponensét, meg kell határoznia a modulnak a kriptográfiai határát, amely a komponenseket körülzárja, valamint teljes mértékben ismertetnie kell a modul fizikai konfigurációját.

KÖV_01.09:
A dokumentációnak meg kell említenie a modul minden olyan hardver, szoftver vagy förmver komponensét, amely nem tartozik a szabvány biztonsági követelményei alá, és bizonyítania kell, hogy ezek a részek nem befolyásolják a modul biztonságosságát.

KÖV_01.10:
A dokumentációnak tartalmaznia kell a kriptográfiai modul összes fizikai és logikai interfészét.

KÖV_01.11:

[8] Csak azokat a követelményeket adjuk meg, mely az nShield F3 kriptográfiai modulra ténylegesen vonatkoznak, ezért a követelmények sorszámozása nem mindig folyamatos.

A dokumentációnak tartalmaznia kell a kriptográfiai modul manuális és logikai kezelőit, a fizikai és logikai állapotjelzőit és a fizikai, logikai és elektromos karakterisztikáját.

KÖV_01.12:
A dokumentációnak fel kell sorolnia az összes biztonsági funkciót, mind a FIPS által Jóváhagyottakat, mind a nem Jóváhagyottakat, melyeket a kriptográfiai modulban felhasználnak, és meg kell határozni az összes működési módot, FIPS által Jóváhagyott és nem Jóváhagyott formában is.

KÖV_01.13:
A dokumentációnak tartalmaznia kell egy blokkdiagramot, amely leírja a modul minden fontos hardver komponensét és azok csatlakozásait, beleértve ebbe a mikroprocesszorokat, input/output puffereket, nyílt szöveg/rejtjelezett szöveg pufferek, vezérlési pufferek, kulcstárak, működési memória és program memória.

KÖV_01.14:
A dokumentációnak meg kell határoznia a hardver, szoftver és förmver komponensek tervezését. Magasszintű specifikációs nyelvet kell használni a szoftver/förmver vagy a hardver séma tervezésének leírására.

KÖV_01.15:
A dokumentációnak meg kell határoznia minden biztonsággal kapcsolatos információt, mint a titkos és magán kriptográfiai kulcsok (nyílt és titkosított formában), autentikációs adatok (pl. jelszavak, PIN kódok), és más védett információk (pl. naplózott események, naplóadatok), melyek közzététele vagy módosítása kompromittálja a modul biztonságát.

KÖV_01.16::
A dokumentációnak teljes mértékben meg kell határoznia a kriptográfiai modul biztonsági politikáját, vagyis mindazokat a biztonsági szabályokat, amelyek alatt a modulnak üzemelnie kell. Különösen fontos az, hogy a biztonsági politikának tartalmaznia kell azokat a biztonsági szabályokat, amelyek ezen szabvány[9] biztonsági követelményeiből illetve a gyártó által előírt járulékos biztonsági követelményekből származnak.

4.2 Modul interfészek

KÖV_02.01:
A modult úgy kell megszerkeszteni, hogy a modulhoz tartozó minden információ áramlás és minden fizikai hozzáférés olyan logikai interfészekre legyen korlátozva, amelyek valamennyi, a modulba való belépési- illetve a modulból való kilépési pontot meghatároznak.

KÖV_02.02:
A modul interfészeknek egymástól logikailag el kell különülniük, bár osztozhatnak egy fizikai porton (pl. a bejövő adat beléphet, a kimenő adat távozhat ugyanazon a porton) vagy el lehetnek osztva egy vagy több fizikai portra (pl. a bejövő adat érkezhet a soros vagy párhuzamos portról is).

KÖV_02.03:
A modulnak legalább a következő négy logikai interfészt tartalmaznia kell:

- adat input interfész,
- adat output interfész,
- vezérlési input interfész,
- státusz output interfész.

KÖV_02.04:
Minden adatot (kivéve a vezérlési adatot, mely a vezérlői input interfészen érkezik), mely bekerül a modulba, és az feldolgozza (ilyen a nyílt adat, a titkos adat, kriptográfiai kulcsok és CSP-k, autentikációs adatok és állapot információk más moduloktól), az adat input interfészen keresztül kell bevinni.

[9] FIPS 140-2

KÖV_02.05:
Minden adatot (kivéve a vezérlési adatot, mely a vezérlői output interfészen távozik), mely kikerül a modulból (ilyen a nyílt adat, a titkos adat, kriptográfiai kulcsok és CSP-k, autentikációs adatok és állapot információk más moduloknak), az adat output interfészen keresztül kell kiolvasni.

KÖV_02.06:
Az adat output interfészen keresztül történő minden adat outputot le kell tiltani hiba állapot vagy az öntesztek végrehajtása során.

KÖV_02.07:
Minden input parancs, jel, vezérlő adat (pl. a hívások és a manuális vezérlők, mint a kapcsolók, gombok és billentyűzetek), melyek a modul működését befolyásolják, a vezérlési input interfészen keresztül kell, hogy közlekedjen.

KÖV_02.08:
Minden output jel, jelző és állapotinformáció (pl. a visszatérő kódok, és a fizikai jelzők, mint a LED-ek és a kijelzők), melyek a modul állapotának jelzésére szolgál, a státusz output interfészen keresztül kell, hogy közlekedjen.

KÖV_02.09:
Minden külső elektromos áramforrásnak, mely a kriptográfiai modulba csatlakozik, az elektromos áram interfészen keresztül kell, hogy illeszkedjen.

KÖV_02.10:
A modulnak meg kell különböztetnie az input adatot és vezérlést valamint az output adatot és állapotot.

KÖV_02.11:
Minden input adat, mely bekerül a modulba az adat input interfészen keresztül, csak az input adat úton keresztül közlekedhet.

KÖV_02.12:
Minden output adat, ami az adat output interfészen keresztül hagyja el a modult, csak az output adat úton keresztül közlekedhet.

KÖV_02.13:
Az output adat utat logikailag le kell kapcsolni az áramkörről és a folyamatokról a kulcsgenerálás, a manuális kulcsbejegyzés és a kulcs törlése során.

KÖV_02.14:
Az érzékeny információk véletlen kiszivárgásának megakadályozása érdekében két független belső lépés szükséges az adat kiadásához bármely output interfészen, melyen nyílt szövegű kriptográfiai kulcsok vagy CSP-k, illetve érzékeny adatok távoznak (pl. két független szoftver flag beállítása, melyek egyikét a felhasználó állítja; két hardveres kapu, melyek sorosan hajtanak végre két intézkedést).

KÖV_02.15:
A dokumentációnak a modul minden fizikai portot, logikai interfészt, input és output adat utat ismertető, teljes specifikációt kell tartalmaznia.

KÖV_02.16:
Azon fizikai portoknak, melyeken nyílt szövegű kriptográfiai kulcsok, autentikációs adatok és CSP-k érkeznek vagy távoznak, fizikailag el kell különülniük az összes többi porttól a modulon belül, vagy eleget kell tenniük a KÖV_02.17-nek.

KÖV_02.17:
Azon logikai interfészeknek, melyeken nyílt szövegű kriptográfiai kulcsok, autentikációs adatok és CSP-k érkeznek vagy távoznak, fizikailag el kell különülniük az összes többi interfésztől megbízható adatút segítségével, vagy eleget kell tenniük a KÖV_02.16-nak.

KÖV_02.18:

Nyílt szövegű kriptográfiai kulcs komponenseket, autentikációs adatokat és más CSP-ket közvetlenül a kriptográfiai modulba kell bevinni (pl. megbízható adatúton vagy közvetlenül csatolt kábelen).

4.3 Szerepkörök és szolgáltatások

KÖV_03.01:
A kriptográfiai modulnak támogatnia kell az operátori szerepköröket és az ezekhez tartozó megfelelő szolgáltatásokat.

KÖV_03.02:
Ha a modul több egyidejű operátort támogat[10], akkor a modulnak belsőleg le kell kezelnie az egyes operátorok által végrehajtott jogosult szerepkörök és szolgáltatások szétválasztását.

4.3.1 Szerepkörök

KÖV_03.03:
A kriptográfiai modulnak minimálisan a következő jogosult szerepköröket kell támogatnia:

- Felhasználói szerepkör: a szerepkört egy olyan felhasználó tölti be, aki fel van jogosítva biztonsági szolgáltatások elérésére, kriptográfiai műveletek és egyéb jogosult funkciók végrehajtására,
- Kriptográfiai tisztviselő szerepkör: a szerepkört egy olyan kriptográfiai tisztviselő tölti be, aki fel van jogosítva az összes kriptográfiai inicializálás és menedzsment funkció végrehajtására (pl. kriptográfiai kulcsok és paraméterek beírása, kriptográfiai kulcsok katalogizálása, naplózási funkciók és alarm nullázások).

KÖV_03.06:
A dokumentációnak teljes specifikációt kell nyújtania mindazokról a jogosult szerepkörökről, amelyeket a modul támogat.

4.3.2 Szolgáltatások

KÖV_03.07:
A *szolgáltatások* fogalom minden olyan szolgáltatásra, műveletre és funkcióra vonatkozik, amit a modullal végre lehet hajtani.

KÖV_03.08:
A szolgáltatás bemenet tartalmaz minden olyan adatot és vezérlőműveletet, ami kezdeményez vagy elér bizonyos szolgáltatást, műveletet vagy funkciót.

KÖV_03.09:
A szolgáltatás kimenet tartalmaz minden olyan adatot és vezérlőműveletet, ami egy szolgáltatás, művelet vagy funkció eredménye, amit egy szolgáltatás bemenet kezdeményezett.

KÖV_03.10:
Minden szolgáltatás inputnak egy szolgáltatás outputot kell eredményeznie.

KÖV_03.11:
A kriptográfiai modulnak minimálisan a következő szolgáltatásokat kell nyújtania:

- státusz kijelzés: a modul aktuális státuszának outputja,
- ön-teszt: az ön-teszt inicializálása és futtatása a 11. fejezetben (Ön-tesztek) specifikáltaknak megfelelően.
- jóváhagyott biztonsági funkciók végrehajtása: legalább egy jóváhagyott biztonsági funkció végrehajtása Jóváhagyott működési módban.

KÖV03.14:
A dokumentációnak teljes specifikációt kell nyújtania minden olyan jogosult szolgáltatásról, műveletről és funkcióról, amelyet a modul segítségével végre lehet hajtani. Minden szolgáltatás esetén specifikálni

[10] Az nShield F3 PCI támogat több egyidejű operátort.

kell a szolgáltatás inputokat, a megfelelő szolgáltatás outputokat és azt a jogosult szerepkört ill. szerepköröket, amelyben a szóban forgó szolgáltatás végrehajtható.

KÖV03.15:
A dokumentációnak tartalmaznia kell minden olyan modul által nyújtott szolgáltatást, melynél nem szükséges az operátor bizonyos szerepköre, valamint annak a leírása, hogy ezek a szolgáltatások nem befolyásolják a kriptográfiai kulcsokat, CSP-ket, illetve a modul teljes biztonságát.

4.3.3 Operátori hitelesítés

KÖV_03.16:
A biztonság fokától függően a modulnak legalább a következők egyikét támogatnia kell: szerepkör alapú hitelesítés vagy azonosság alapú hitelesítés.

KÖV_03.19:
Azonosságon alapuló hitelesítés esetén a kriptográfiai modulnak hitelesítenie kell az operátor azonosságát, és ellenőriznie kell, hogy az azonosított operátor jogosult-e egy vagy több meghatározott szerepkör betöltésére. A modulnak a következő tevékenységeket kell végrehajtania:

- meg kell követelnie, hogy az operátor egyedileg azonosított legyen,
- hitelesítenie kell az operátor megadott azonosságát,
- meg kell követelnie, hogy az operátor közvetett vagy közvetlen módon kiválasszon egy vagy több szerepkört,
- A hitelesített azonosság alapján ellenőriznie kell, hogy az operátor jogosult betölteni a kiválasztott szerepkört, valamint jogosult végrehajtani az annak megfelelő szolgáltatásokat.

KÖV_03.20:
Az azonosságon alapuló hitelesítés esetén a modul engedélyezheti, hogy egy operátor szerepkört váltson anélkül, hogy szükséges lenne az operátor azonosságának újbóli hitelesítése, de a modulnak ellenőriznie kell, hogy a hitelesített operátor jogosult-e az új szerepkör végrehajtására.

KÖV_03.21:
Ha egy modult áram alá helyeznek miután előzőleg az áramellátás megszűnt (pl. villamos hálózati hiba következtében) vagy karbantartás, illetve javítás után, a megelőző hitelesítés eredményeit nem szabad megőrizni, azaz a modulnak újra hitelesítenie kell az operátor jogosultságát ahhoz, hogy a megkívánt szerepkört betölthesse.

KÖV_03.22:
A hitelesítő adatokat a modulon belül védeni kell a nyilvánosságra kerüléstől, a módosítástól és a helyettesítéstől.

KÖV_03.23:
A hozzáférés ellenőrző mechanizmusok megvalósításához szükséges hozzáférés ellenőrző információk inicializálására használt szolgáltatások esetében a modulhoz való hozzáférés szabályozására különböző módszerek használhatók, mint pl. ügyrendi ellenőrzés, vagy gyári alap (default) beállítású hitelesítési és jogosultsági információk.

KÖV_03.24:
A hitelesítési eljárások erősségének teljesítenie kell a következő követelményeket:

KÖV_03.25:
Minden hitelesítési próbálkozásnál a véletlen kitalálás vagy a hibás elfogadás valószínűsége legalább 1/1.000.000 kell, hogy legyen.

KÖV_03.26:
Egy perc alatti többszörös hitelesítési kérések véletlen kitalálásának vagy hibás elfogadásának a valószínűsége 1/100.000 kell, hogy legyen.

KÖV_03.27:
A hitelesítési adatot a hitelesítés során el kell takarni az operátor elől (pl. nem látszódnak a képernyőn a karakterek).

KÖV_03.28:
A hitelesítési próbálkozás visszajelzése az operátor felé nem gyengítheti a hitelesítési eljárást.

KÖV_03.29:
A dokumentációnak tartalmaznia kell a következőket:
- a modul által nyújtott hitelesítési eljárások,
- az autentikációs adatok típusa, ami a hitelesítési eljárások eléréshez szükségesek,
- azon hitelesítési eljárás, mely a modul első eléréséhez és inicializáláshoz szükséges, valamint
- a különböző hitelesítési eljárások erőssége.

KÖV_03.32:
A kriptográfiai modulnak azonosságon alapuló hitelesítési mechanizmusokat (pl. az operátor azonosításán alapuló mechanizmust) kell alkalmazni abból a célból, hogy az operátor jogosultságát ellenőrizze arra vonatkozóan, hogy a kívánt szerepköröket betölthesse és az annak megfelelő szolgáltatásokat igényelhesse. Ezeken túlmenően, nyílt formában megjelenő hitelesítési adatokat (pl. jelszavakat és PIN kódokat), nyílt formában megjelenő kriptográfiai kulcs komponenseket és más, nem védett kritikus biztonsági paramétereket olyan porton vagy portokon keresztül kell beadni, amelyek fizikailag el vannak különítve a többi porttól, és amelyek lehetővé teszik a direkt megadást /ahogyan azt a 2. fejezet (Modul interfészek) előírja/. Ide vonatkozó követelmények találhatók az KÖV_02.13 és KÖV_02.14-ben is.

4.4. Véges állapotú automata modell

KÖV_04.01:
Minden kriptográfiai modult egy olyan véges állapotú automata modell felhasználásával kell megtervezni, amely világosan meghatározza a modul minden üzemelés közbeni és hiba állapotát.

KÖV_04.02:
Egy kriptográfiai modult a következő állapot típusok alkalmazásával kell tervezni:
- Áram bekapcsolási-kikapcsolási állapot: primer, szekunder és tartalék áramellátási állapotok. Ezek az állapotoknak különbséget tehetnek a modul különböző részeinek ellátására szolgáló áramellátások között,
- Kriptográfiai tisztviselő állapotok: olyan állapotok, amelyekben a kriptográfiai tisztviselő funkciók kerülnek végrehajtásra (pl. kriptográfiai inicializálás és kulcs menedzsment funkciók),
- Kulcs beírási állapotok: olyan állapotok, amelyek kriptográfiai kulcsoknak és más kritikus biztonsági paramétereknek a modulba való beírási, és azok érvényességének ellenőrzésére szolgálnak,
- Felhasználói szolgáltatói állapotok: olyan állapotok, amelyekben az arra feljogosított felhasználók biztonsági szolgáltatásokhoz juthatnak, kriptográfiai funkciókat vagy más jogosult felhasználói funkciót hajthatnak végre,
- Ön-tesz állapotok: olyan állapotok, amelyek a modul ön-tesztjének végrehajtására szolgálnak /lásd 11. fejezet (Ön-teszt)/,
- Hiba állapotok: olyan állapotok, amelyekbe a modul hiba fellépésekor kerül (pl. sikertelen ön-teszt, titkosítás megkísérlése olyan esetben, amikor működéshez szükséges kulcsok vagy más kritikus biztonsági paraméterek hiányoznak, vagy kriptográfiai hibák lépnek fel). A hiba állapotok felölelhetnek működést kizáró (hard) hibákat, amelyek egy készülék hibáját jelzik és a modul karbantartását vagy javítását igénylik, és felölelhetnek helyreállítható (soft) hibákat, amelyek a modul inicializálását vagy “reset”-elését igényelhetik.

KÖV_04.03:
Minden hiba állapotnak olyannak kell lenni, hogy azt vissza lehessen állítani (reset) egy elfogadható működési állapotba vagy kezdeti állapotba, kivéve azokat a nem helyrehozható (hard) hibákat, amelyek a modul karbantartását, szervizelését vagy javítását igénylik.

KÖV_04.05:
Az állapot átmenetek leírásának tartalmaznia kell azokat a belső modul feltételeket, adat inputokat és vezérlő inputokat, amelyek egy állapotból egy másikba való átmenetet okoznak, és tartalmaznia kell

azokat a belső modul feltételeket, adat outputokat és státusz outputokat, amelyeket egy állapotból egy másikba való átmenet eredményez.

4.5. Fizikai biztonság

KÖV_05.01:
A kriptográfiai modulnak fizikai biztonsági eljárásokat kell alkalmaznia annak érdekében, hogy letiltsák a modul tartalmához való nem engedélyezett hozzáférést és hogy felfedezzék a modul nem nem engedélyezett működtetését és módosítását az telepítés során.

KÖV_05.02:
A kriptográfiai határon belül levő összes hardver, szoftver és förmver egységet védeni kell.

4.5.1 Közös követelmények[11]

KÖV_05.03:
A következő követelményeknek minden fizikai biztonsági alkotóra érvényesnek kell lenniük:

KÖV_05.04:
A dokumentációnak tartalmaznia kell a fizikai megvalósítás teljes specifikációját, valamint azt a biztonsági szintet, melyen a modul fizikai biztonsági eljárásai meg lettek valósítva.

KÖV_05.05:
A dokumentációnak tartalmaznia kell azoknak az alkalmazható biztonsági mechanizmusoknak a teljes leírását, amelyeket a modul alkalmazhat.

KÖV_05.12:
A modulnak rendelkeznie kell olyan gyártás során beépített alkatrésszel, ami megvédi a modult (pl. védőburkolat, mely a modul áramköreit veszi körül, ezzel védve a fizikai károsodástól).

KÖV_05.16:
A modulnak rendelkeznie kell olyan megoldással, ami lehetővé teszi a modulhoz való illetéktelen fizikai hozzáférés felfedését.

KÖV_05.17:
A modulnak a 3. biztonsági szinten a következő követelmények is eleget kell tennie:

KÖV_05.18:
Ha a kriptográfiai modul tartalmaz valamilyen nyílást vagy fedőt, vagy van karbantartási felülete, rendelkeznie kell olyan alkatrésszel, ami illetéktelen hozzáférés esetén kitörli az érzékeny adatokat a modulból.

KÖV_05.19:
Az illetéktelen hozzáférésre adott válasz során a törlő áramkörnek minden nyílt szövegű titkos kulcsot és CSP-t törölnie kell a nyílás kinyitásakor, a fedél elmozdításakor vagy a karbantartási felülelethez való hozzáféréskor.

KÖV_05.20:
Az illetéktelen hozzáférésre való válaszadásnak és a törlő áramkörnek mindig működnie kell, amikor nyílt szövegként titkos kulcs vagy CSP van a modulban.

4.5.2 Több chipes, beágyazott kriptográfiai modulra vonatkozó követelmények

KÖV_05.33:

[11] Vagyis a kriptográfiai modul mindhárom lehetséges fizikai konfigurációjára (egy chipből álló, több chipes, beágyazott, illetve több chipes, önmagában álló) vonatkozik.

Több chipes, beágyazott kriptográfiai modul esetén a modulban lévő chipeknek olyan termék minőségűeknek kell lenniük, amelyek magukban foglalnak standard passziválási technikát is.

KÖV_05.34:
Több chipes, beágyazott kriptográfiai modul esetében a modult egy nem átlátszó, beavatkozást kimutató anyaggal kell beburkolni.

KÖV_05.36:
Több chipes, beágyazott kriptográfiai modul esetében a következő három követelmény egyikét kell alkalmazni a modulra:

- egy kemény, nem átlátszó kiöntő anyagot kell alkalmazni,
- a modult egy erős, nem eltávolítható burkoló anyagnak kell tartalmaznia,
- a modult egy erős, eltávolítható burkolatba kell bezárni, és tartalmaznia kell beavatkozásra reagáló és nullázó áramköri egységet.

4.6 Az operációs rendszer biztonsága

Nincsenek követelmények[12].

4.7 Kriptográfiai kulcsgondozás

4.7.1 Általános követelmények

KÖV_07.01:
A titkos és magán kulcsokat védeni kell a jogosulatlan felfedéssel, módosítással és helyettesítéssel szemben.

KÖV_07.02:
A nyilvános kulcsokat védeni kell a jogosulatlan módosítással és kicseréléssel szemben.

KÖV_07.03:
Dokumentációnak kell specifikálnia a kriptográfiai modulra vonatkozó kulcsgondozás minden vonatkozását.

4.7.2 Véletlenszám generátorok (RNG)

KÖV_07.04:
Amennyiben a modul Jóváhagyott vagy Nem jóváhagyott RNG-t használ Jóváhagyott működési módban, az RNG-ből származó adatnak teljesíteni kell a folyamatos véletlenszám generálási tesztet.

KÖV_07.06:
A Jóváhagyott RNG-ket alá kell vetni a kriptográfiai algoritmus tesztnek.

KÖV_07.07:
A nem-determinisztikus RNG-knek meg kell felelnie az összes, szabványban foglalt, alkalmazható követelménynek.

KÖV_07.08:
Jóváhagyott RNG-t kell használni a Jóváhagyott biztonsági funkció kriptográfiai kulcsainak generáláshoz.

KÖV_07.09:
A mag (seed) és a kezdeti kulcs (seed key) soha nem lehet ugyanolyan értékű.

KÖV_07.10:

[12] Mivel az nShield F3 PCI kriptográfiai modulnak nincs saját operációs rendszere.

A dokumentációban fel kell sorolni a modul által használt összes véletlenszám generátort.

4.7.3 Kulcs generálásra vonatkozó követelmények

KÖV_07.11:
Egy kriptográfiai modul opcionálisan ki lehet egészítve egy belső kulcs generálási funkcióval[13]. A modulnak egy FIPS által jóváhagyott kulcs generálási algoritmust kell implementálni

KÖV_07.12:
Ha a kulcs generálási folyamatban egy véletlenszám generátor is alkalmazva van[14], minden értéket olyan módon kell véletlenszerűen vagy pszeudo-véletlenszerűen generálni, hogy a bitek minden lehetséges kombinációja és minden lehetséges érték egyenlő valószínűséggel generálódjon.

KÖV_07.13:
A kulcsgenerálási eljárás biztonságának veszélyeztetéséhez legalább annyi művelet szükséges, amennyiből a véletlen kulcs értékét ki lehet találni.

KÖV_07.14:
Ha egy kezdeti *(seed)* kulcs alkalmazva van[15], akkor azt ugyanolyan módon kell bevinni, mint a kriptográfiai kulcsokat.

KÖV_07.15:
Közbenső kulcs generálási állapotoknak és értékeknek nem szabad hozzáférhetőnek lenniük a modulon kívül nyílt vagy más nem védett formában.

KÖV_07.16:
A dokumentációnak tartalmaznia kell a modul által használt összes kulcsgenerálási eljárást.

4.7.4 Kulcs szétosztásra vonatkozó követelmények

KÖV_07.17:
Egy kriptográfiai modulnak FIPS által jóváhagyott kulcs szétosztási technikát kell implementálnia.

KÖV_07.19:
A kulcs szétosztási eljárás veszélyeztetéséhez legalább annyi művelet szükséges, amennyiből a továbbított kriptográfiai kulcs értékét ki lehet számolni.

KÖV_07.20:
Amennyiben létezik kulcstovábbítási eljárás, a teljesíteni kell a kulcs be- és kivitelre vonatkozó követelményeket

KÖV_07.21:
A dokumentációnak specifikálnia kell a modul által alkalmazott kulcs szétosztási technikát.

4.7.5 Kulcs bevitelére és kivitelére vonatkozó követelmények

KÖV_07.22:
Kézi úton szétosztott kriptográfiai kulcsok bevihetők a kriptográfiai modulba, illetve outputként kinyerhetők abból, tisztán kézi módszerekkel vagy elektronikus módszerekkel.

KÖV_07.23:
Amennyiben egy kezdeti (seed) kulcs kerül a modulba a kulcsgenerálás során, azt a kriptográfiai kulcsokkal azonos feltételek mellett kell bevinni.

[13] A nShield F3 PCI megvalósít belső kulcs generálási funkciót.
[14] A nShield F3 PCI alkalmaz véletlenszám generátort.
[15] A nShield F3 PCI véletlenszám generátora alkalmaz kezdeti *(seed)* kulcsot.

KÖV_07.24:
Minden titkosított titkos és nyilvános kulcsot, melyet a kriptográfiai modulba bevisznek vagy kivesznek, a FIPS által jóváhagyott módban egy FIPS által jóváhagyott algoritmussal kell titkosítani.

KÖV_07.25:
Eszközt kell szolgáltatni annak biztosítására, hogy a modulba bevitt vagy abból outputként kinyert kulcs azzal a megfelelő jogi személlyel legyen összekapcsolva (pl. személy, csoport vagy eljárás), akihez a kulcs hozzá van rendelve.

KÖV_07.26:
A kézi úton szétosztott kriptográfiai kulcsokat a kriptográfiai modulba való bevitel során ellenőrizni kell a helyesség szempontjából a 11 fejezetben (Ön-tesztek) meghatározott kézi kulcs beviteli teszt felhasználásával.

KÖV_07.27:
Ha kódolt kulcsok vagy kulcs komponensek kerülnek beírásra, az ebből származó nyílt formájú titkos vagy magán kulcsok nem jeleníthetők meg.

KÖV_07.28:
A dokumentációnak tartalmaznia kell minden olyan kulcs be- és kiviteli eljárást, melyet a kriptográfiai modul használ.

KÖV_07.30:
Az elektronikus úton szétosztott titkos és magán kulcsokat kódolt formában kell bevinni és kinyerni.

KÖV_07.31:
A kézi úton szétosztott titkos vagy magán kulcsokat nem szabad bevinni vagy outputként kinyerni a kriptográfiai modulból nyílt formában. Ha kézi úton szétosztott titkos vagy magán kulcsokat kell bevinni a kriptográfiai modulba vagy outputként kinyerni onnan, akkor ezeket a következő módszerek valamelyikével kell elvégezni:

- kódolt formában,
- osztott tudáson alapuló (azaz két vagy több nyílt formájú kulcs komponenst felhasználó) eljárás alkalmazásával.

KÖV_07.32:
Ha kézi úton szétosztott titkos vagy magán kulcsot osztott tudáson alapuló eljárás segítségével visznek be vagy nyernek ki, a modulnak lehetőséget kell nyújtania arra, hogy az operátort külön-külön hitelesítse minden egyes kulcs komponens esetében.

KÖV_07.33:
Osztott tudáson alapuló hitelesítés esetén a kulcs komponenseket közvetlenül a kriptográfiai modulba kell bevinni, illetve közvetlenül a kriptográfiai modulból kell kinyerni (pl. megbízható útvonalon vagy közvetlenül csatlakoztatott kábelen keresztül) anélkül, hogy az áthaladna valamilyen borításon vagy olyan közbenső rendszeren, ahol a komponensek tárolhatók, összekapcsolhatók vagy más módon feldolgozhatók.

KÖV_07.34:
Osztott tudáson alapuló eljárásoknál legalább két kulcs komponens szükséges az eredeti kriptográfiai kulcs újragenerálásához.

KÖV_07.35:
Osztott tudáson alapuló eljárások esetén a dokumentációban meg kell jelennie, hogy ha egy kulcs újragenerálásához n kulcs komponens kell, akkor n-1 kulcs komponens jelenléte nem elegendő az eredeti kulcshoz kapcsolódó bármilyen információ kinyeréséhez, kivéve a hosszát.

KÖV_07.36:
Osztott tudáson alapuló eljárások esetén a dokumentációnak tartalmaznia kell a modul által használt összes ilyen eljárást.

4.7.6 Kulcs tárolásra vonatkozó követelmények

KÖV_07.37:
Ha a titkos vagy magán kulcsokat a kriptográfiai modul tartalmazza, akkor azok tárolhatók nyílt formában.

KÖV_07.38:
A nyílt formájú kulcsok a modulon kívülről nem lehetnek hozzáférhetők.

KÖV_07.39:
Eszközt kell szolgáltatni annak biztosítására, hogy minden kulcs azzal a megfelelő jogi személlyel lett összekapcsolva (pl. személy, csoport vagy eljárás), akihez a kulcs hozzá van rendelve.

KÖV_07.40:
A dokumentációnak tartalmaznia kell minden kulcstárolás eljárást.

4.7.7 Kulcs megsemmisítésre vonatkozó követelmények

KÖV_07.41:
Egy kriptográfiai modulnak lehetőséget kell arra nyújtani, hogy minden nyíltan tárolt kriptográfiai kulcsot és egyéb nem védett kritikus biztonsági paramétert a modulon belül nullázni lehessen.

KÖV_07.42:
A dokumentációnak tartalmaznia kell minden kulcstörlési eljárást.

4.8 Elektromágneses interferencia, elektromágneses kompatibilitás

KÖV_08.01:
A kriptográfiai modulnak eleget kell tennie az alábbi követelményeknek:

KÖV_08.02:
A kriptográfiai modulok jeladó részének (rádiónak) minden alkalmazható FCC követelménynek eleget kell tenniük.

KÖV_08.03:
A dokumentációban nyilatkozatot kell tenni az EMI/EMC követelményeknek való megfelelőségről.

KÖV_08.04:
Egy kriptográfiai modulnak alkalmazkodnia kell az EMI/EMC követelményekhez, amelyek a 47 Code of Federal Regulations 15. részében, a B alfejezetben, B osztályában (azaz a házi alkalmazásra vonatkozó részben) vannak megadva.

4.9 Ön-tesztek

4.9.1 Általános követelmények

KÖV_09.01:
A modulnak végre kell tudnia hajtani bekapcsolási öntéseket és feltételes öntéseket, ami a helyes működést biztosítja.

KÖV_09.02:
Bizonyos ön-teszteket akkor kell végrehajtani, amikor a modul áram alá kerül (áram alá helyezéskor végrehajtandó tesztek).

KÖV_09.03:
Egyéb ön-teszteket különböző feltételek esetén kell végrehajtani, általában akkor, ha egy meghatározott funkció vagy művelet kerül végrehajtásra (feltételhez kötött tesztek).

KÖV_09.04:
Amennyiben a kriptográfiai modul valamelyik ön-tesztje sikertelen, a modulnak hiba állapotba kell kerülnie, és hiba jelet kell kiadnia a státusz interfészen keresztül.

KÖV_09.05:
A modul semmilyen kriptográfiai funkciót nem végezhet addig, amíg hiba állapotban van.

KÖV_09.06:
A modul semmilyen adatot nem adhat ki outputként az adat output interfészen keresztül, amíg a hiba feltétel fennáll.

KÖV_09.07:
Minden lehetséges bekapcsolási és feltételes öntesztnek, hiba feltételnek dokumentáltnak kell lenni mindazokkal a tevékenységekkel együtt, amelyek szükségesek a hiba törlésére és a normál működéshez való visszatéréshez (ez tartalmazhatja a modul karbantartását, szervizelését és javítását is).

4.9.2 Áram alá helyezési tesztek

4.9.2.1 Általános tesztek

KÖV_09.08:
Miután egy kriptográfiai modult áram alá helyeztek, a modulnak ön-teszt állapotba kell kerülnie.

KÖV_09.09:
Az áram alá helyezés utáni ön-tesztek nem igényelhetnek operátori közreműködést a futtatáshoz.

KÖV_09.10:
Amennyiben minden áram alá helyezés utáni teszt sikeres, akkor egy jelzést kell kiadni a "státusz output" interfészen keresztül.

KÖV_09.11:
Minden adat outputot le kell tiltani, amíg ezek a tesztek végrehajtás alatt állna.

KÖV_09.12:
A modulnak eszközöket kell biztosítania arra, hogy az áram alá helyezési teszteket igény esetén a modul periodikus tesztelésére is kezdeményezni lehessen.

KÖV_09.13:
A modulnak legalább a következő (áram alá helyezési) teszteket végre kell hajtania:

- kriptográfiai algoritmus teszt,
- szoftver/förmver teszt,
- a kritikus műveletek tesztje és

4.9.2.2 Kriptográfiai algoritmus tesztek

KÖV_09.16:
A kriptográfiai algoritmusokat tesztelni kell oly módon, hogy az algoritmust olyan adatokon kell végrehajtani, amelyekre vonatkozóan a helyes output már ismert ("ismert eredmény teszt"). Az ismert eredmény tesztet minden egyes kriptográfiai funkcióra vonatkozóan (pl. kódolás, dekódolás, hitelesítés) végre kell hajtani.

KÖV_09.17:
A teszt sikertelen, ha a kiszámított output nem egyezik meg a korábban generált outputtal.

KÖV_09.18:
Azon kriptográfiai algoritmusokat, melyek kimenete a bemenettől függ (pl. a DSA algoritmus), vagy az ismert eredmény teszttel vagy a pár konzisztencia teszttel kell ellenőrizni.

KÖV_09.19:
Az üzenet lenyomat készítő algoritmusok tesztelésére egy független ismert eredmény teszt vagy egy, a lenyomatoló algoritmushoz kapcsolódó kriptográfiai algoritmust tesztelő ismert eredmény teszt szükséges.

KÖV_09.20:
Ha a kriptográfiai modulnak két független megoldása van ugyanannak a kriptográfiai algoritmusnak a tesztelésére, akkor a két megvalósítás kimenetét folyamatosan össze kell hasonlítani.

KÖV_09.21:
Ha a kriptográfiai modulnak két független megoldása van ugyanannak a kriptográfiai algoritmusnak a tesztelésére, és a két megvalósítás kimenete nem egyezik, akkor a kriptográfiai algoritmus tesztnek nem felelt meg.

4.9.2.3 Szoftver/förmver teszt

KÖV_09.22:
A modulban (például az EEPROM-ban vagy RAM-ban) található minden beágyazott szoftver és förmver esetén számításba kell venni és tárolni kell egy hiba detektáló kódot (EDC) vagy FIPS által jóváhagyott hitelesítési technikát (pl. egy adat hitelesítési kód kiszámítását és ellenőrzését vagy egy FIPS által elfogadott digitális aláírási algoritmust). Ezt a hiba detektáló kódot, adat hitelesítési kódot ill. digitális aláírást ellenőrizni kell akkor, amikor az áram alá helyezési ön-tesztek futnak.

KÖV_09.23:
Amennyiben a kiszámolt eredmény nem egyenlő a korábban készített eredménnyel, a szoftver/förmver teszt nem felelt meg.

4.9.2.4 Kritikus funkciók tesztjei

KÖV_09.25:
Minden más, a modul biztonságos működése szempontjából kritikus funkció tesztelhető azon ön-tesztek részeiként, amelyeket az áram alá helyezéskor kell végrehajtani.

KÖV_09.26:
A meghatározott feltételek esetén végrehajtandó egyéb kritikus funkciókat a feltételhez kötött tesztek részeként kell végrehajtani.

KÖV_09.27:
A dokumentációnak teljes specifikációt kell szolgáltatnia a kritikus funkciókról és azon áram alá helyezési ön-tesztek természetéről, amelyek ezen funkciók számára ki vannak jelölve.

4.9.3 Feltételhez kötött tesztek

KÖV_09.29:
A feltételhez kötött teszteket a modulnak akkor kell végrehajtania, amikor a következő tesztek feltételei teljesülnek:

- páronkénti konzisztencia teszt,
- szoftver/förmver betöltési teszt,
- kézi kulcs bevitel teszt,
- folyamatos véletlenszám generátor teszt
- megkerülés teszt

4.9.3.1 Páronkénti konzisztencia teszt

KÖV_09.30:
Azon kriptográfiai modulok, amelyek nyilvános és magán kulcsokat generálnak, tesztelniük kell a kulcsokat a páronkénti konzisztencia szempontjából.

KÖV_09.31:
Ha a kulcsokat FIPS által jóváhagyott kulcstovábbításra használják, a nyilvános kulccsal kell titkosítani a nyílt szövegű értéket. Az eredményként kapott titkos szöveget kell összehasonlítani a nyílt szövegű értékkel. Ha a két érték egyezik, akkor a teszt sikertelen. Ha a két érték különbözik, akkor a titkos kulccsal dekódolni kell a titkos szöveget, majd a kapott értéket össze kell hasonlítani az eredeti nyílt szöveggel. Ha a két érték nem egyezik, akkor a teszt sikertelen.

KÖV_09.33:
Ha a kulcsokat csak digitális aláírás létrehozására és ellenőrzésére használják, akkor a kulcsok konzisztenciája tesztelhető egy aláírás létrehozásával és ellenőrzésével is.

4.9.3.2 Szoftver/förmver betöltési tesztek

KÖV_09.34:
Ha a modulba kívülről szoftver vagy förmver komponenst lehet betölteni, a következő szoftver/förmver teszteket kell végrehajtani.

KÖV_09.35:
Minden olyan érvényesített szoftver és förmver esetében, amelyet kívülről lehet betölteni a kriptográfiai modulba, alkalmazni kell egy olyan kriptográfiai mechanizmust, amely FIPS által jóváhagyott hitelesítési technikát (pl. adat hitelesítési kód vagy FIPS által elfogadott digitális aláírási algoritmus) használ.

KÖV_09.36:
A kiszámolt eredményt össze kell hasonlítani a korábban generált eredménnyel. Ha a két kiszámolt eredmény nem egyezik, akkor a szoftver/förmver integritás teszt nem felelt meg.

4.9.3.3 Kézi kulcs bevitel tesztje

KÖV_09.37:
Amennyiben egy kriptográfiai modulba kézi úton visznek be kriptográfiai kulcsokat vagy kulcs elemeket, a következő teszteket kell végrehajtani.

KÖV_09.38:
A kulcsoknak rendelkezniük kell egy hiba detektáló kóddal (pl. paritás ellenőrzési érték), vagy pedig kétszeres beírást kell alkalmazni a beírt kulcsok helyességének ellenőrzésére.

KÖV_09.39:
EDC használata esetén az EDC-nek legalább 16 bit hosszúnak kell lennie.

KÖV_09.40:

Ha az EDC-t nem lehet ellenőrizni, vagy a kétszeres beírás nem egyezik, a teszt nem felelt meg.

4.9.3.4 Folyamatos véletlenszám generátor teszt

KÖV_09.41:
Azon kriptográfiai moduloknak, amelyek egy véletlenszám vagy pszeudó véletlenszám generátort implementálnak, tesztelniük kell a generátort a sikertelenség szempontjából egy konstans értékig.

KÖV_09.42:
Ha a generátor n bitből álló blokkokat generál, ahol n>15, a bekapcsolás után generált első blokkot nem szabad felhasználni, de tárolni kell abból a célból, hogy összehasonlításra kerüljön a következő generálandó blokkal. Az egymást követő generálások során az újonnan generált blokk összehasonlításra kerül az előző generált blokkal. A teszt sikertelen, ha a két összehasonlított blokk azonos.

KÖV_09.43:
Ha a generátornak minden hívása 16 bitnél kevesebbet szolgáltat, akkor a bekapcsolás utáni első n bitet, valamilyen n>15-re, nem szabad felhasználni, de tárolni kell a következő n generált bittel való összehasonlításra. Minden egymást követő n-bit generálás összehasonlításra kerül a megelőzően generált n-bittel. A teszt sikertelen, ha két összehasonlított n-bites sorozat megegyezik.

4.10 Tervezési biztosíték

4.10.1 Konfiguráció kezelés

KÖV_10.01:
A modul kriptográfiai határán belül meg kell valósítani egy konfiguráció kezelő rendszert a kriptográfiai modul és a modul komponensek részére, és ezt a dokumentációban meg kell jeleníteni.

KÖV_10.02:
Minden, a konfigurációt érintő elemet, mely érinti a rendszer biztonságát és a dokumentációt, egy egyedi azonosítóval kell ellátni.

4.10.2 Továbbítás és működtetés

KÖV_10.03:
A dokumentációnak tartalmaznia kell a biztonságos telepítés, inicializálás és indítás műveleteit.

KÖV_10.04:
A dokumentációnak tartalmaznia kell a biztonság fenntartásának körülményeit a modul szétosztása és továbbítása során.

4.10.3 Fejlesztés

KÖV_10.06:
A dokumentációnak meg kell mutatnia a hardver, a szoftver és a förmver komponensek tervezése és a kriptográfiai modul biztonsági szabályzata közötti összhangot.

KÖV_10.07:
Amennyiben a modul tartalmaz szoftver vagy förmver komponenst, a dokumentációban meg kell jelennie ezek forráskódjának, világosan jelezve a tervezésnek való megfelelőségüket.

KÖV_10.08:
Ha a kriptográfiai modul tartalmaz hardver komponenseket, a dokumentációban meg kell határozni ezek sémáját és/vagy Hardver Leíró Nyelv (HDL) segítségével a komponensek listáját.

KÖV_10.10:
A dokumentációnak tartalmaznia kell olyan funkcionális specifikációt, mely informális módon leírja a modult, a külső portokat és az interfészeket, és az interfészek célját.

KÖV_10.12:
Minden szoftver és förmver komponenst magas-szintű nyelven kell megvalósítani, kivéve akkor, amikor teljesítmény vagy kivitelezési problémák miatt csak alacsony-szintű nyelv (assembly vagy mikrokód) használható.

KÖV_10.13:
Minden hardver komponenst magas-szintű specifikációs nyelvvel kell megtervezni.

4.10.4 Támogató dokumentáció

KÖV_10.21:
A kriptográfiai tisztviselő dokumentációjában le kell írni az adminisztratív funkciókat, biztonsági eseményeket, biztonsági paramétereket (és paraméter értékeket), fizikai portokat, és logikai interfészeket, amik a kriptográfiai tisztviselő számára elérhetők.

KÖV_10.22:
A kriptográfiai tisztviselő dokumentációjában le kell legyen írva, hogy hogyan lehet a kriptográfiai modult biztonságosan üzemeltetni.

KÖV_10.23:
A kriptográfiai tisztviselő dokumentációjának olyan, a felhasználók viselkedésével kapcsolatos elvárásokat is tartalmaznia kell, amik a biztonságos működéshez szükségesek.

KÖV_10.24:
A felhasználói dokumentációban meg kell határozni a Jóváhagyott biztonsági funkciókat, fizikai portokat, logikai interfészeket, melyek a felhasználó számára elérhetők.

KÖV_10.25:
A felhasználói dokumentációnak meg kell határoznia a felhasználó azon kötelességeit, melyek szükségesek a biztonságos működéshez.

5. Az nShield F3 PCI értékeléshez megkövetelt fejlesztői bizonyítékok

Az alábbiakban áttekintjük azokat a fejlesztői bizonyítékokat (dokumentálást, egyéb információ szolgáltatást), melyet a fejlesztő cég biztosított a vizsgálatok elvégzéséhez az nShield F3 PCI értékelését végző laboratórium számára.

Az alábbi jelölést alkalmazzuk:

FB_x.y.z: a FIPS 140-2 x. fejezetének y. biztonsági követelményére vonatkozó z. fejlesztői bizonyítékot meghatározó elvárása.

5.1. A kriptográfiai modul tervezése és dokumentálása

FB_01.03.01:
A fejlesztő által nyújtott biztonsági szabályzatnak tartalmaznia kell a FIPS által jóváhagyott működési mód leírását.

FB_01.03.02:
A fejlesztő által nyújtott biztonsági szabályzatnak tartalmaznia kell azokat az utasításokat, melyekkel a FIPS által jóváhagyott működési módot el lehet indítani.

FB_01.04.01:
A fejlesztő által nyújtott biztonsági szabályzatnak tartalmaznia kell annak a megoldásnak a leírását, ahogy a modul jelzi, ha FIPS által Jóváhagyott működési módban van.

FB_01.04.02:
A fejlesztő által nyújtott biztonsági szabályzatnak tartalmaznia kell, hogy a FIPS által Jóváhagyott működési mód jelzése hogyan érhető el.

FB_01.06.01:
A modulban lévő valamennyi processzorra a fejlesztőnek meg kell határoznia azt a szoftvert és förmvert, amelyet az adott processzor hajt végre, és azokat a memória egységeket, amelyek a végrehajtható kódot és adatokat tartalmazzák, és meg kell jelölni a szoftverek és förmverek fő funkcióját is.

FB_01.06.02:
Minden processzor esetén a fejlesztőnek meg kell határoznia minden olyan hardvert, amelyhez a szóban forgó processzor kapcsolódik.

FB_01.08.01:
A fejlesztői dokumentációban meg kell határozni minden olyan komponenst, amely kriptográfiai logikai áramkört vagy eljárást alkalmaz. A felsorolandó komponenseknek tartalmazniuk kell értelemszerűen a következőket:

- integrált áramköröket, beleértve a processzorokat, memóriákat és fogyasztói rendelésre készített integrált áramköröket,
- egyéb aktív elektronikai áramköri elemeket,
- villamos áram bemeneteket és kimeneteket, belső áramellátásokat vagy konvertereket,
- fizikai struktúrákat, beleértve az áramköri kártyákat vagy más szerelési alapfelületeket, foglalatokat és csatlakozókat,
- a szoftver és förmver modulokat,
- a modulban alkalmazott egyéb komponenseket.

FB_01.08.02:
A fenti komponens listának konzisztensnek kell lennie azokkal az információkkal, amelyek az 1. fejezet (A kriptográfiai modul tervezése és dokumentálása) egyéb követelményeinek kielégítésére szolgálnak.

FB_01.08.03:
A fejlesztői dokumentációnak meg kell határoznia a modul kriptográfiai határát. A kriptográfiai határnak egy olyan világosan meghatározott, összefüggő védelmi peremkerületnek kell lennie, amely a kriptográfiai modul fizikai határát alakítja ki. A védelmi peremkerület definíciónak meg kell határoznia a modul komponenseket és csatlakozókat (portokat), valamint a modul információ áramlási folyamatait, feldolgozó és input/output jeleit**.**

FB_01.08.04:
A kriptográfiai határnak tartalmaznia kell minden olyan hardvert vagy szoftvert, amely inputként fogad, feldolgoz, vagy outputként kiad olyan fontos biztonsági paramétereket, amelyek ha nincsenek kellően ellenőrizve, akkor ez érzékeny információk veszélyeztetéséhez vezethet.

FB_01.08.05:
A fejlesztőnek meg kell határoznia, hogy a modul fizikai konfigurációja a három lehetséges eset közül melyik: egyetlen chipből álló modul, több chipes, beágyazott modul vagy több chipes, önmagában álló modul.

FB_01.08.06:
A fejlesztői dokumentációnak vázolnia kell a modul belső elrendezését és összeszerelési módszereit (pl. rögzítők és szerelvények), beleértve a tervrajzokat is, amelyeknek méret-arányosaknak kell lenniük. Az integrált áramkörök belsejét nem kell ábrázolni.

FB_01.08.07:
A fejlesztői dokumentációnak ismertetnie kell a modul elsődleges fizikai paramétereit, beleértve a foglalatoknak, a hozzáférési pontoknak, az áramköri kártyáknak, az áramellátás elhelyezkedésének, az összekötő huzalok menetének, a hűtőberendezések elhelyezkedésének és más fontos paramétereknek a leírását.

FB_01.09.01:
Minden olyan komponenst, amely nem tartozik a biztonsági követelmények alá, tételesen fel kell sorolni a fejlesztői dokumentációban.

FB_01.09.02:
A FB_01.09.01 követelmény kielégítésére készített lista valamennyi elemére vonatkozóan a kizárás okát elfogadható módon meg kell magyarázni a fejlesztői dokumentációban. A fejlesztőnek bizonyítania kell, hogy ezen komponensek egyike sem okozhat veszélyeztetést elfogadható körülmények között, még hibás működés vagy rosszindulatú használat esetén sem.

FB_01.12.01:
A fejlesztőnek be kell mutatnia a FIPS által jóváhagyott kriptográfiai algoritmusokkal kapcsolatos tanúsítványait.

FB_01.12.02:
A fejlesztői dokumentációnak tartalmaznia kell minden FIPS által nem jóváhagyott biztonsági funkció listáját.

FB_01.13.01:
A fejlesztői dokumentációnak tartalmaznia kell egy olyan funkcionális blokkdiagramot, amely bemutatja a hardver komponenseket és azok csatlakozásait. A blokkdiagramnak tartalmaznia kell értelemszerűen a következő komponenseket:

- mikroprocesszorok,
- input/output bufferek,
- nyíltan tárolt szöveg / kódoltan tárolt szöveg bufferek,
- ellenőrző bufferek,
- kulcs tárolás,
- munka memória,
- program memória,
- minden más, fontos felhasznált komponens.

FB_01.13.02:
A blokkdiagramnak ezeken felül tartalmaznia kell minden más fogyasztói rendelésre készített integrált áramköröket, mint pl. előre megtervezett kriptográfiai áramköröket, kapu áramköröket vagy egyéb programozható logikai áramköröket.

FB_01.13.03:
A blokkdiagramnak be kell mutatnia a modul fő komponensei közötti, valamint a modul és a külső berendezés közötti kapcsolatokat.

FB_01.13.04:
A blokkdiagramnak be kell mutatnia a modul kriptográfiai határát.

FB_01.14.01:
A fejlesztői dokumentációnak tartalmaznia kell a hardver, szoftver és/vagy förmver komponensek részletes specifikációját. A dokumentációban meg kell jelennie egy véges állapot modellnek a 4.4 fejezetben meghatározott feltételeknek megfelelően. Amennyiben a kapcsolat a véges állapot modell és a tervezési specifikáció között nem világos, további dokumentációt kell benyújtani, ami tisztázza a kapcsolatot.

FB_01.15.01:
A fejlesztőnek dokumentálnia kell minden biztonsággal kapcsolatos információt, mint a titkos és nyilvános kulcsok, hitelesítő adatok, és más védett információk védelme, amik kiszivárgása vagy módosítása befolyásolja a modul biztonságát.

FB_01.16.01:
A fejlesztőnek gondoskodnia kell egy különálló dokumentumról vagy dokumentum fejezetről, amely meghatározza azt a biztonsági politikát (vagyis azokat a biztonsági szabályokat, amelyek mellett egy modulnak működnie kell), amelyet a kriptográfiai modul léptet hatályba.

5.2 Modul interfészek

FB_02.01.01:
A fejlesztői dokumentációnak meg kell határoznia minden fizikai portot és logikai interfészt , például:

- Fizikai portok és ezek tűkiosztásai
- Fizikai fedők, nyílások
- Logikai interfészek (pl. az API-k és más adat/vezérlő/állapot jelzések) és a jelzések nevei és funkciói
- Kézi vezérlők (gombok és kapcsolók), melyek a fizikai vezérlő bemenetre hatnak
- Fizikai állapotjelzők (pl. fényjelzések vagy kijelzők), melyek a fizikai állapot kimenetre érvényesek
- A logikai interfészek és a fizikai portok, kézi vezérlők, fizikai állapot jelzők közötti kapcsolatok
- Fizikai, logikai és elektromos karakterisztikák a fenti portokra és interfészekre

FB_02.01.02:
A fejlesztői dokumentációnak részleteznie kell a modul információ folyamait és hozzáférési pontjait azáltal, hogy az 1. fejezetben (A kriptográfiai modul tervezése és dokumentálása) megkövetelt blokkdiagram másolatait kiemelésekkel és jegyzetekkel látja el. Ezeken felül további dokumentációt is kell szolgáltatni, amely szükséges a logikai interfészek világos specifikálásához.

FB_02.01.03:
A modulhoz csatlakozó minden input és output esetében a dokumentációnak meg kell határoznia azt a logikai interfészt, amelyhez az adott input vagy output tartozik, és meg kell határoznia a megfelelő fizikai belépési/kilépési pontokat. Az ezen követelmény kielégítésére szolgáltatott információknak konzisztenseknek kell lenniük azokkal a komponens információkkal, amelyek az 1. fejezet (A kriptográfiai modul tervezése és dokumentálása) követelményei kielégítésére készültek, valamint a logikai portokra vonatkozó 2. fejezetbeli követelményekkel.

FB_02.02.01:

A fejlesztői tervnek a modul interfészeket logikailag elkülönített kategóriákra kell szétválasztani minimálisan azon kategóriák alkalmazásával, amelyek a KÖV_02.03 és a KÖV_02.09 követelményekben definiálva vannak. Az információknak konzisztensnek kell lennie a logikai interfészek és a fizikai portok KÖV_02.01-ben foglalt specifikációjával.

FB_02.02.02:
Amennyiben két vagy több interfész ugyanazon a fizikai porton osztozik, a fejlesztőnek meg kell határoznia, hogy a különböző interfész kategóriákból származó információk hogyan különíthetők el logikailag.

FB_02.03.02:
A fejlesztői dokumentációnak tartalmaznia kell annak bizonyítékát, hogy a következő négy logikai interfész megtalálható a modulban:

- adat input interfész (meghatározva a KÖV_02.04-ben),
- adat output interfész (meghatározva a KÖV_02.05-ben),
- vezérlési input interfész (meghatározva a KÖV_02.07-ben),
- státusz output interfész (meghatározva a KÖV_02.08-ban).

FB_02.04.01:
A modulnak rendelkeznie kell egy adat input interfésszel, amely definiálva van a fejlesztői dokumentációban, beleértve az alábbiakat:

- nyíltan tárolt adatok,
- kódolt szövegként tárolt adatok,
- kriptográfiai kulcsok,
- egyéb kulcsgondozási adatok,
- hitelesítési adatok,
- státusz információk,
- minden más input adat.

FB_02.04.02:
A fejlesztői dokumentációban meg kell határozni minden olyan külső beviteli eszközt, mely valamilyen adat bevitelére alkalmas az adat input interfészen keresztül. Ez lehet intelligens kártya, token, biometrikus eszköz, stb.

FB_02.05.01:
A kriptográfiai modulnak rendelkeznie kell adat output interfésszel. Minden adatot (kivéve az állapotadat, mely az állapot output interfészen jelenik meg), mely feldolgozás után kikerül a modulból, az adat output interfészen keresztül kell kiadni. Ilyen adatok:

- Nyíltszövegű adat
- Titkosított adat és elektronikus aláírás
- Kriptográfiai kulcsok és más kulcskezelési adatok (nyíltan vagy kódolva)
- Vezérlőinformációk külső eszközöknek
- Bármilyen más kimenő adat

FB_02.05.02:
A fejlesztői dokumentációban meg kell határozni minden olyan külső kimeneti eszközt, mely valamilyen adat fogadására alkalmas az adat output interfészen keresztül. Ez lehet intelligens kártya, token, biometrikus eszköz, stb.

FB_02.06.01:
A fejlesztői tervezetnek biztosítania kell, hogy az adat output interfészen keresztül történő minden adat output letiltásra kerüljön, amikor a modul hiba állapotba kerül, ahogyan azt a 4. fejezet (Véges állapotú automata modell) dokumentálja, és a fejlesztői dokumentációnak tartalmaznia kell, hogy ez hogyan valósul meg.

FB_02.06.02:

A fejlesztői tervezetnek biztosítania kell, hogy az adat output interfészen keresztül történő minden adat output letiltásra kerüljön, amikor a modul ön-teszt állapotba kerül, ahogyan azt a 9. fejezet (Ön-tesztek) dokumentálja, és a fejlesztői dokumentációnak tartalmaznia kell, hogy ez hogyan valósul meg.

FB_02.07.01:
A modulnak rendelkeznie kell egy vezérlési input interfészel, amely definiálva van a fejlesztői dokumentációban, és amelyet a modul működésének vezérlésére alkalmaznak, beleértve az input parancsokat, jelzéseket, adatokat és kézi inputokat.

FB_02.07.02:
A fejlesztői dokumentációban meg kell határozni minden olyan külső beviteli eszközt, mely valamilyen parancs, jel vagy vezérlő adat bevitelére alkalmas a vezérlő input interfészen keresztül. Ez lehet intelligens kártya, token, stb.

FB_02.08.01:
A modulnak rendelkeznie kell egy státusz output interfésszel, amely definiálva van a fejlesztői dokumentációban, és amelyet a modul státuszának megjelenítésére vagy kijelzésére alkalmaznak, beleértve az output adatokat, jelzéseket, kijelzőket és fizikai kijelzőket.

FB_02.08.02:
A fejlesztői dokumentációban meg kell határozni minden olyan külső kimeneti eszközt, mely valamilyen állapotinformáció, jel, logikai jelző vagy fizikai jelző fogadására alkalmas az állapot output interfészen keresztül. Ez lehet intelligens kártya, token, kijelző és/vagy tároló eszköz.

FB_02.09.01:
Ha a modul felvesz vagy szolgáltat külső áramot, rendelkeznie kell egy elektromos áram interfésszel, amely a fejlesztői dokumentációban megfelelő módon definiálva van, és amely tartalmazza az elektromos áram valamennyi belépési vagy kilépési pontját.

FB_02.10.01:
A fejlesztői dokumentációnak tartalmaznia kell annak leírását, hogy a modul hogyan tesz különbséget adat és vezérlés között az input, adat és állapot között az output interfészen, valamint hogy a bemenő adat és vezérlés útját meghatározó fizikai és logikai adatutak hogyan válnak szét a kimenő adat és állapot útját meghatározó fizikai és logikai adatutaktól.

FB_02.11.01:
A fejlesztői dokumentációnak minden fizikai és logikai input adat útvonalat megfelelő részletességgel ismertetnie kell abból a célból, hogy a modul input információinak minden fő kategóriája specifikálva legyen. Minden input adat, amely az adat input interfészen keresztül lép a modulba, csak az input adat útvonalat használhatja a belépéshez.

FB_02.12.01:
A fejlesztői dokumentációnak minden fizikai és logikai output adat útvonalat megfelelő részletességgel ismertetnie kell abból a célból, hogy a modul output információinak minden fő kategóriája specifikálva legyen. Minden output adat, amely az adat output interfészen keresztül lép ki modulból, csak az output adat útvonalon keresztül juthat ki.

FB_02.13.01:
A fejlesztői dokumentációban meg kell határozni, hogy a fizikai és logikai utak, melyeket a kimenő adatok fő kategóriái használnak, hogyan válnak le logikailag vagy fizikailag azokról a folyamatokról, melyek a kulcsgenerálást, a kézi kulcsbevitelt és a kriptográfiai kulcsok törlését végzik. A modul nem engedheti meg, hogy ezen folyamatok kulcs vagy CSP információkat adjanak ki, valamint a kimenő adatok ne zavarják meg a folyamatokat.

FB_02.14.01:
Ha bármilyen lehetősége fennáll annak, hogy a modul szerkezete valamelyik porton lehetővé teszi nyílt formában megjelenő kriptográfiai kulcsok vagy más kritikus biztonsági pataméterek outputját, a szerkezetnek két független belső tevékenységet kell megkövetelnie, mielőtt az output bekövetkezik egy ilyen porton. Ebben az esetben a fejlesztői dokumentációnak definiálnia kell, hogy mik ezek a tevékenységek és hogyan nyújtanak védelmet a kritikus biztonsági paraméterek gondatlan

közzétételével szemben. A dokumentációnak tartalmaznia kell a modul azon funkcionális részeinek a specifikációját (akár hardverben akár szoftverben van megvalósítva), amelyekben a két független tevékenység végrehajtásra kerül.

FB_02.16.01:
Amennyiben a modul szerkezete nem védett kritikus biztonsági paramétereket tesz szükségessé, beleértve nyíltan megjelenő kriptográfiai kulcsokat vagy nyíltan megjelenő hitelesítési adatokat, az ezen adatok inputjára vagy outputjára szolgáló adat portoknak fizikailag el kell különülniük a modul összes többi portjától. A fejlesztői dokumentációnak be kell mutatnia, hogy ez hogyan valósul meg.

FB_02.17.01:
Amennyiben a modul szerkezete nem védett kritikus biztonsági paramétereket tesz szükségessé, beleértve nyíltan megjelenő kriptográfiai kulcsokat vagy nyíltan megjelenő hitelesítési adatokat, az ezen adatok inputjára vagy outputjára szolgáló logikai interfészeknek logikailag el kell különülniük a modul összes logikai interfészétől. A fejlesztői dokumentációnak be kell mutatnia, hogy ez hogyan valósul meg.

FB_02.18.01:
Amennyiben a modul szerkezete nem védett kritikus biztonsági paramétereket tesz szükségessé, beleértve nyíltan megjelenő kriptográfiai kulcsokat, nyíltan megjelenő hitelesítési adatokat, az ezen paraméterek inputjára vagy outputjára szolgáló adat portokat közvetlenül a kriptográfiai határhoz kell csatolni, anélkül, hogy azok áthaladnának bármilyen, a kriptográfiai határon kívül eső processzoron, komplex logikai blokkon vagy a kulcs kezeléssel kapcsolatban nem álló funkciókat végrehajtó modul részen. A fejlesztői dokumentációnak be kell mutatnia a megvalósítás módját.

5.3 Szerepkörök és szolgáltatások

FB_03.02.01:
A fejlesztői dokumentációnak meg kell határoznia, hogy egyidejűleg több operátor engedélyezett-e. Amennyiben engedélyezett, a fejlesztőnek ismertetnie kell azt a módszert, amellyel az egyes operátorok által végrehajtott jogosult szerepkörök és szolgáltatások szétválasztása megvalósul. A fejlesztői dokumentációnak tartalmaznia kell az egyidejű operátorokra vonatkozó minden korlátozást (pl. nem engedélyezett egyidejűleg egy operátor karbantartói szerepkörben és egy másik operátor felhasználói szerepkörben).

5.3.1 Szerepkörök

FB_03.03.01:
A fenti FB_03.01.01 kielégítésére megkövetelt dokumentációba a fejlesztőnek legalább egy felhasználói és egy kriptográfiai tisztviselő szerepkört bele kell vennie.

FB_03.06.01:
A fejlesztői dokumentációnak meg kell határoznia minden megkülönböztethető jogosult szerepkört, beleértve annak megnevezését, célját és azokat a szolgáltatásokat, amelyek az adott szerepkörben végrehajthatók.

FB_03.11.01:
A dokumentációnak tartalmaznia kell a modul állapotának lekérdezési módját, valamint a felhasználó által meghívható ön-tesztek inicializációját és futtatását, az FB_03.14.01 és az FB_03.15.01-ben meghatározott szolgáltatásokkal együtt.

FB_03.14.01:
A dokumentációnak tartalmaznia kell minden szolgáltatás célját és funkcióját.

FB_03.14.02:
A fejlesztői dokumentációnak meg kell határoznia minden szolgáltatáshoz kapcsolódóan az inputokat, outputokat és azon felhatalmazott szerepet vagy szerepeket, amivel végre lehet hajtani. A szolgáltatás inputoknak tartalmaznia kell a modul összes adat vagy vezérlő inputját, amin keresztül inicializálni vagy működtetni lehet azt. A szolgáltatás outputoknak tartalmaznia kell minden olyan adat és állapot

outputot, melyen keresztül az inputon keresztül kezdeményezett szolgáltatások, eredményét le lehet kérni.

FB_03.15.01:
A dokumentációnak tartalmaznia kell minden szolgáltatás célját és funkcióját.

FB_03.15.02:
A fejlesztői dokumentációnak meg kell határoznia minden szolgáltatásra az inputokat és a hozzájuk tartozó outputokat. A szolgáltatás inputoknak tartalmaznia kell minden adat vagy vezérlő inputot, melyen keresztül a szolgáltatások inicializálhatók vagy működtethetőek. A szolgáltatás outputoknak tartalmaznia kell minden olyan adat és állapot outputot, melyen keresztül az input által kezdeményezett szolgáltatás eredménye lekérdezhető.

5.3.3 Operátori hitelesítés

FB_03.19.01:
A fejlesztőnek dokumentálnia kell azokat a mechanizmusokat, amelyeket az operátor azonosításának végrehajtására, az operátor azonosságának hitelesítésére, a szerepkör vagy szerepkörök közvetett vagy közvetlen kiválasztására és annak ellenőrzésére alkalmaznak, hogy az operátor jogosult-e a szerepkör(ök) felvételére. Meg kell jegyezni, hogy az azonosságon alapuló hitelesítés figyelembe veszi az operátornak az azonosságát, aki egy meghatározott szerepkört felvesz. Ez a hitelesítési módszer nemcsak a szerepkörök között tesz különbséget, de ugyanazon szerepkörön belül is; két operátor, aki ugyanazt a szerepkört kívánja betölteni, a modul számára különböző információt fog felmutatni, mivel azonosítójuk különböző. Például ha egy operátornak egy PIN kódot kell megadnia akkor, ha megkísérel egy szerepkört betölteni, minden egyes operátornak különböző PIN kóddal kell rendelkeznie, mivel a PIN kód a modul számára az operátort azonosítja.

FB_03.20.01:
A fejlesztőnek dokumentálnia kell, hogy a modul lehetővé teszi-e egy operátor számára, hogy szerepkört váltson anélkül, hogy azonosságát újra hitelesíteni kellene. Ha ez a lehetőség fennáll, a fejlesztői dokumentációnak ismertetnie kell, hogy az operátor számára fennáll az a lehetőség, hogy szerepkört váltson, és világosan ki kell jelentenie, hogy ellenőrizni kell az operátor jogosultságát az új szerepkörre.

FB_03.21.01:
A fejlesztői dokumentációnak ismertetnie kell, hogy egy áramellátás megszűnését követően a megelőző hitelesítések eredményei hogyan lesznek törölve.

FB_03.22.01:
A dokumentációnak tartalmaznia kell minden olyan védelmi eljárást, amit a modul a hitelesítő adatok védelmére felhasznál. A védelemnek olyan eljárásokat kell tartalmaznia, melyek védenek az illetéktelen hozzáféréstől, módosítástól és helyettesítéstől.

FB_03.23.01:
A fejlesztői dokumentációnak tartalmaznia kell azon eljárásokat, melyek a modulhoz való hozzáférést szabályozzák az inicializálás előtt.

FB_03.25.01:
A dokumentációban meg kell jelennie minden egyes hitelesítési módnak, valamint az elfogadható hibás engedélyezés és a véletlen kitalálás arányoknak.

FB_03.26.01:
A fejlesztői dokumentációban megtalálható minden hitelesítési mód, valamint ezek megfejtésének valószínűsége egy perc alatt.

FB_03.27.01:
A dokumentáció leírja, hogy hogyan lehet megakadályozni a hitelesítő adatok operátor általi kifigyelését.

FB_03.28.01:

A dokumentációban meg kell jelennie egy olyan eljárásnak, mely biztosítja az operátor által bevitt hitelesítő adatok visszajelzését.

5.4 Véges állapotú automata modell

FB_04.05.01:
A fejlesztőnek leírást kell adnia a véges állapotú automata modellről. Ezen leírásnak tartalmaznia kell a modul minden állapotának megadását és leírását, és le kell írnia a megfelelő állapot átmenetek mindegyikét. Az állapot átmeneteknek tartalmazniuk kell azokat a belső modul feltételeket, adat inputokat és vezérlő inputokat, amelyek egy állapotból egy másikba való átmenetet okoznak, és azokat a belső modul feltételeket, adat outputokat és státusz outputokat, amelyeket egy állapotból egy másikba való átmenet eredményez.

5.5 Fizikai biztonság

5.5.1 Közös követelmények

FB_05.04.01:
A fejlesztői dokumentációnak specifikálnia kell, hogy a modulra vonatkozóan az alábbi három fizikai megvalósítás melyike áll fenn: egyetlen chipből álló modul, több chipes, beágyazott modul vagy több chipes, önmagában álló kriptográfiai modul[16]. A specifikált fizikai megvalósításnak konzisztensnek kell lennie az aktuális modul fizikai tervével.

FB_05.05.01:
A fejlesztői dokumentációnak teljesen le kell írnia azokat az alkalmazható fizikai biztonsági mechanizmusokat, amelyeket a modul felhasznál. A modul összes összetevőjét, beleértve minden hardvert, szoftvert, förmvert és adatot (beleértve a nyíltan tárolt kriptográfiai kulcsokat és nem védett kritikus védelmi paramétereket) védeni kell.

FB_05.12.01:
A több chipes, beágyazott modul chipjeinek szabványos termék minőségű IC-knek kell lenniük, amelyeket úgy terveztek, hogy legalább a tipikus kereskedelmi minőségi specifikációknak feleljenek meg az áramellátás, hőmérséklet, megbízhatóság, ütés/rázkódás stb. tekintetében. Különösen fontos, hogy a modul standard passziválási technikát alkalmazzon minden egyes chipre vonatkozóan. A fejlesztői dokumentációnak ismertetnie kell az IC-k minőségét. Ha valamelyik alkalmazott IC nem szabványos, annak passziválási szerkezetét szintén ismertetni kell.

5.5.2 Több chipes, beágyazott kriptográfiai modulra vonatkozó követelmények

FB_05.34.01:
A modult tipikus termék szintű foglalatba vagy tokba kell beépíteni. A fejlesztői dokumentációnak ismertetnie kell a modulnak foglalat vagy tok leírását.

FB_05.36.01:
A modult egy nem átlátszó, beavatkozást kimutató burkolattal kell befedni, mint pl. egy, az alakot követő burkolat, vagy folyékony festék. Az anyagnak átlátszatlannak kell lennie a látható tartományon belül. A fejlesztői dokumentációnak meg kell adnia a beavatkozást kimutató, nem átlátszó burkolat fajtáját és annak karakterisztikáját.

5.6. Az operációs rendszer biztonsága

Nincsenek követelmények[17].

[16] Az nShield F3 PCI esetében ez: több chipes, beágyazott modul.
[17] Mivel az nShield F3 PCI kriptográfiai modulnak nincs saját operációs rendszere.

5.7. Kriptográfiai kulcsgondozás

5.7.1 Általános követelmények

FB_07.01.01:
A fejlesztői dokumentációnak ismertetnie kell minden, a modul számára belső titkos és/vagy magán kulcs védelmét. A védelemnek tartalmaznia kell olyan mechanizmusok implementálását, amelyek védelmet nyújtanak a jogosulatlan felfedéssel, módosítással és helyettesítéssel szemben.

FB_07.02.01:
Ha a modul támogat nyilvános kulcsokat, a fejlesztői dokumentációnak ismertetnie kell minden nyilvános kulcs védelmét. A védelemnek tartalmaznia kell olyan mechanizmusok implementálását, amelyek védelmet nyújtanak a jogosulatlan módosítással és helyettesítéssel szemben.

FB_07.03.01:
A dokumentációnak ismertetnie kell a kriptográfiai kulcsok, kulcs komponensek és CSP-k listáját.

5.7.2 Véletlenszám generátorok (RNG)

FB_07.08.01:
A fejlesztői dokumentációban szerepelnie kell egy állításnak, miszerint a kulcsgenerálás során FIPS által jóváhagyott véletlenszám generálás történik. Az erre vonatkozó követelmények a FIPS PUB 140-2 C mellékletében találhatók.

FB_07.09.01:
A fejlesztői dokumentációnak tartalmaznia kell egy olyan eljárást, ami biztosítja, hogy a mag és a kezdeti kulcs sosem egyezik meg.

FB_07.10.01:
A fejlesztői dokumentációban le kell írni az összes felhasznált RNG-t (akár FIPS által jóváhagyott, akár nem), ezek típusát és felhasználását a modulban.

5.7.3 Kulcs generálásra vonatkozó követelmények

FB_07.11.01:
A fejlesztőnek bizonyítékot is kell nyújtania arra vonatkozóan, hogy a kulcs generálási algoritmus FIPS által jóváhagyott.

FB_07.13.01:
A fejlesztőnek olyan dokumentumot kell benyújtania, ami megmutatja, legalább hány művelet szükséges ahhoz, hogy a generált kulcs értékét ki lehessen találni a kulcsgeneráló algoritmust kihasználva (pl. a kezdeti kulcsot kitalálva determinisztikussá tenni az RNG-t).

FB_07.15.01:
A dokumentációnak jeleznie kell, hogy a kulcsgenerálás során valamilyen átmeneti érték elhagyja-e a modult.

FB_07.15.02:
A kulcs generálási eljárások nem tehetnek lehetővé semmilyen outputot a kulcs generálási folyamat során, kivéve azokat az értékeket, amelyek kódolva vannak.

FB_07.16.01:
A dokumentációnak bizonyítékot kell szolgáltatnia arról, hogy a kulcsgenerálási eljárást a modul használja.

5.7.4 Kulcs szétosztásra vonatkozó követelmények

FB_07.17.01:
A dokumentációban a fejlesztőnek nyilvánosságra kell hoznia, hogy FIPS által jóváhagyott kulcsgondozási eljárást használ. A jóváhagyott kulcsgondozási eljárások a FIPS PUB 140-2 D mellékletében találhatók.

FB_07.19.01:
A fejlesztőnek olyan dokumentumot kell benyújtania, ami megmutatja, legalább hány művelet szükséges ahhoz, hogy a kriptográfiai kulcs értékét ki lehessen találni a kulcs továbbítása során.

FB_07.21.01:
A dokumentációnak tartalmaznia kell a modul által felhasznált kulcsszétosztási eljárásokat.

5.7.5 Kulcs bevitelére és kivitelére vonatkozó követelmények

FB_07.23.01:
A kulcsmenedzsment dokumentációnak tartalmaznia kell a kezdeti kulcs bevitelének módját.

FB_07.24.01:
A dokumentációban meg kell jelennie, hogy a magán és titkos kulcsokat, melyeket a modulba betöltenek vagy kivesznek, milyen FIPS által jóváhagyott algoritmusokkal titkosítják.

FB_07.25.01:
A dokumentált kulcs beviteli / kiviteli eljárásoknak ismertetniük kell azokat a mechanizmusokat vagy eljárásokat, amelyeket annak biztosítására alkalmaznak, hogy minden kulcs a megfelelő jogi személlyel legyen összekapcsolva.

FB_07.27.01:
A dokumentált kulcs beviteli eljárásnak lehetővé kell tennie a kódolt kulcsok és kulcs komponensek kijelzését a kulcs beírás folyamán, ha ez szükséges, de lehetetlenné kell tenni azoknak a nyílt formájú titkos és magán kulcsok kijelzését, amelyek a kódolt kulcsok és kulcs komponensek beviteléből származnak.

FB_07.28.01:
A fejlesztői dokumentációnak meg kell határoznia a modul által használt kulcsbeviteli és kivételi eljárásokat.

FB_07.32.01:
A fejlesztői dokumentációban meg kell határozni azt a modul által használt eljárást, amivel a kulcsbevitelért illetve kivételért felelős operátorokat külön-külön lehet azonosítani.

FB_07.34.01:
Ha kézi úton szétosztott titkos vagy magán kulcsokat osztott tudáson alapuló eljárás segítségével visznek be vagy nyernek outputként ki, a fejlesztői dokumentációnak a kulcs beviteli eljárás leírásában meg kell határoznia, hogy hány kulcs komponens szükséges a kulcs újragenerálásához.

FB_07.35.01:
A dokumentációnak le kell írnia, hogy n-1 kulcs komponens ismerete nem elegendő semmilyen, a kulccsal kapcsolatos információ felfedésére, kivéve a kulcs hosszát.
FB_07.36.01:
A fejlesztő által kiadott dokumentációban szerepelnie kell annak az állításnak, hogy a modul osztott tudáson alapuló eljárásokat használ.

5.7.6 Kulcs tárolásra vonatkozó követelmények

FB_07.39.01:

A kulcs tárolásról szóló fejlesztői dokumentációnak ismertetnie kell azokat a mechanizmusokat vagy eljárásokat, amelyeket annak biztosítására alkalmaznak, hogy minden kulcs a megfelelő jogi személlyel legyen összekapcsolva.

FB_07.40.01:
A fejlesztői dokumentációnak tartalmaznia kell a következő információt minden tárolt kulcsról:

- Típus és azonosító
- Tárolás helye
- A formátum, ahogy a kulcsot tárolják (nyílt szöveg, titkosított forma, osztott tudáson alapuló védelem). Amennyiben a kulcsot titkosított formában tárolják, meg kell határozni, hogy milyen FIPS által jóváhagyott algoritmus védi azt.

5.7.7 Kulcs megsemmisítésre vonatkozó követelmények

FB_07.41.01:
A fejlesztői dokumentációnak meg kell határozni a nyílt szövegű titkos és magán kulcsok valamint a CSP-k megsemmisítésével kapcsolatos információkat:

- Megsemmisítési technika
- Megkötések a nyílt szövegű titkos és magán kulcsok és a CSP-k megsemmisítésénél
- Nyílt szövegű titkos és magán kulcsok és a CSP-k, melyek megsemmisülnek
- Nyílt szövegű titkos és magán kulcsok és a CSP-k, melyek nem semmisülnek meg és ennek magyarázata
- Annak magyarázata, hogy a megsemmisítési eljárás annyi idő alatt megy végbe, amennyi nem elég a nyílt szövegű titkos és magán kulcsok és a CSP-k felfedésére

5.8 Elektromágneses interferencia, elektromágneses kompatibilitás

FB_08.02.01:
A fejlesztőnek meg kell neveznie azon FCC által akkreditált laboratóriumot, mely a tanúsítványát kiállította.

FB_08.02.02:
A fejlesztőnek be kell nyújtani a kriptográfiai modul FCC tanúsítványának számát.

FB_08.05.01:
A fejlesztőnek egy FCC bizonyítványt kell szolgáltatnia arra vonatkozóan, hogy a kriptográfiai modul alkalmazkodik azokhoz az EMI/EMC követelményekhez, amelyek az FCC 15 részében, a B alrészben és B osztályban vannak megadva.

5.9 Ön-tesztek

5.9.1 Általános követelmények

FB_09.04.01:
A fejlesztőnek dokumentálnia kell minden egyes ön-teszthez kapcsolódó minden hiba állapotot, és minden egyes hiba állapot esetén közölnie kell a várt hiba jelzést.

FB_09.05.01:
Lásd az FB_02.06.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően. A fejlesztői tervezetnek azt is biztosítania kell, hogy kriptográfiai műveletek nem hajthatók végre, amíg a modul hiba állapotban van.

FB_09.06.01:
Lásd az FB_02.06.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően. A fejlesztői tervezetnek azt is biztosítania kell, hogy kriptográfiai műveletek nem hajthatók végre, amíg a modul hiba állapotban van.

FB_09.07.01:
A fejlesztőnek listát kell szolgáltatni valamennyi, kötelező és opcionális ön-tesztről, amelyeket a modul végre tud hajtani. Ennek a listának egyaránt tartalmaznia kell az áram bekapcsolási teszteket és a feltételes teszteket.

FB_09.07.02:
A fejlesztői dokumentációnak minden egyes hiba feltételre vonatkozóan meg kell adnia annak megnevezését, azokat az eseményeket, amelyek kiváltják, azokat a tevékenységeket, amelyek szükségesek a hiba törlésére és a normál működéshez való visszatéréshez. Meg kell jegyezni, hogy a szükséges tevékenységek magukban foglalhatják azt is, hogy a modult a gyártóhoz kell elküldeni javításra.

5.9.2 Az áram alá helyezési tesztek

5.9.2.1 Általános tesztek

FB_09.09.01:
A fejlesztői dokumentációnak meg kell követelnie, hogy az áram alá helyezés utáni ön-tesztek nem vonhatnak maguk után semmilyen operátori inputot vagy operátori tevékenységet.

FB_09.10.01:
A fejlesztőnek dokumentálnia kell azt a jelzést, amelyet a modul kiad az áram alá helyezés után végrehajtandó tesztek sikeres végrehajtása esetén.

FB_09.12.01:
A fejlesztőnek ismertetnie kell azokat az eljárásokat, amelyek segítségével egy operátor elindíthatja az áram alá helyezéskor elvégzendő ön-teszteket.

FB_09.13.01:
Lásd az FB_09.07.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően.

5.9.2.2 Kriptográfiai algoritmus tesztek

FB_09.16.01:
Lásd az FB_09.07.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően.

FB_09.17.01:
A fejlesztőnek dokumentálnia kell az "ismert eredmény" tesztet, amelyet a kriptográfiai algoritmus tesztelésére végre kell hajtani.
FB_09.17.02:
A dokumentációban be kell mutatni azt, hogy amennyiben a két kimenet nem azonos, a modul hogyan megy át hibaállapotba, illetve milyen hibajelzés jelenik meg a kimenetén.

FB_09.18.01:
Lásd az FB_09.07.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően.

FB_09.18.02:
A fejlesztői dokumentációban meg kell határozni azokat a teszteket, amiket a modul felhasznál.

FB_09.19.01:
Lásd az FB_09.07.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően.

FB_09.19.02:
A fejlesztői dokumentációban meg kell határozni azokat a teszteket, amiket a modul felhasznál.

FB_09.20.01:
Lásd az FB_09.07.01-t a fejlesztői dokumentációra vonatkozó követelményeket illetően.

FB_09.20.02:

A fejlesztőnek meg kell határoznia, hogy ismert eredmény tesztet vagy két független kriptográfiai algoritmus megvalósítás kimenetének összehasonlító tesztjét alkalmazta a modul algoritmusainak tesztelésére. Amennyiben az összehasonlító tesztelést használja, ezt ki kell emelni a dokumentációban.

5.9.2.3 Szoftver/förmver teszt

FB_09.22.01:
A fejlesztői dokumentációnak meg kell határoznia, hogy a beágyazott szoftver és förmver sértetlenségének biztosítására hiba detektálási kódot (EDC) vagy pedig egy FIPS által jóváhagyott hitelesítési technikát (pl. FIPS által jóváhagyott adat hitelesítési kódot (DAC) vagy FIPS által elfogadott digitális aláírást) alkalmaznak-e.

FB_09.22.02:
A dokumentációnak ismertetnie kell az implementált sértetlenséget vizsgáló mechanizmust.

FB_09.22.03:
Ha a modul egy FIPS által jóváhagyott hitelesítési technikát implementál, a fejlesztőnek egy olyan bizonyítékot kell szolgáltatnia, amely tartalmaz egy FIPS értékelésre meghatalmazott (akkreditált) laboratóriumtól származó tanúsítványt, amely kijelenti, hogy a modulban implementált hitelesítési technika FIPS által jóváhagyott[18]. Egy ilyen bizonylat hiányában a fejlesztő cégnek írásos nyilatkozatot kell szolgáltatnia, amely kijelenti, hogy a modulban implementált hitelesítési technika FIPS által jóváhagyott.

5.9.2.4 Kritikus funkciók tesztjei

FB_09.27.01:
A fejlesztőnek minden kritikus funkcióról egy mátrixot kell szolgáltatnia. Minden egyes kritikus funkció esetén a fejlesztőnek fel kell tüntetnie:

- annak célját (pl. azt, hogy a szóban forgó funkció miért "kritikus"),
- melyek azok a kritikus funkciók, amelyeket az áram alá helyezési ön-tesztek tesztelnek,
- melyek azok a kritikus funkciók, amelyeket feltételhez kötött tesztek tesztelnek.

[18] Az nShield F3 rendelkezik ilyen tanúsítvánnyal (DSA, SHA-1, RSA).

5.9.3 Feltételhez kötött tesztek

5.9.3.1 Páronkénti konzisztencia teszt

FB_09.31.01:
Ha a modul nyilvános és magán kulcsokat használ FIPS által jóváhagyott kulcstovábbítási eljárásokra, a fejlesztői dokumentációnak ismertetnie kell egy páronkénti konzisztencia tesztet, amely a nyilvános kulcsot használja fel egy nyílt szöveg titkosítására. Az eredményül kapott kódolt szöveget össze kell hasonlítani az eredeti nyílt szöveggel, hogy különböznek-e.

- Ha a két érték egyenlő, a modult hibaállapotba kell rakni, az állapot interfészen egy hibajelzésnek kell megjelennie.
- Ha a két érték különbözik, a titkos kulcsot használva vissza kell fejteni a kódolt szöveget, majd az eredményt össze kell hasonlítani az eredeti nyílt szöveggel.
- Ha a két érték nem azonos, a teszt nem felelt meg.

FB_09.33.01:
Ha a kulcsokat a modul digitális aláírások számítására és ellenőrzésére használja, akkor vagy a kódolásra/dekódolásra használatos eljáráshoz hozzáadva, vagy azt helyettesítve, a fejlesztői dokumentációnak ismertetnie kell egy páronkénti konzisztencia tesztet, amely egy digitális aláírás létrehozásán és ellenőrzésén alapul.

5.9.3.2 Szoftver/förmver betöltési tesztek

FB_09.35.01:
A fejlesztői dokumentációnak ismertetnie kell a FIPS által jóváhagyott hitelesítési technikát, amelyet a kívülről betöltött szoftver és förmver sértetlenségének védelmére alkalmaznak.

FB_09.35.02:
A fejlesztőnek bizonyítékot kell szolgáltatnia arra vonatkozóan, hogy a technika FIPS által jóváhagyott. Ezen bizonyítéknak egy FIPS értékelésre meghatalmazott (akkreditált) laboratóriumtól származó érvényesítési bizonyítványból kell állnia, amely kijelenti, hogy a modulban implementált hitelesítési technika FIPS által jóváhagyott. Egy ilyen érvényesítési bizonylat hiányában a fejlesztő cégnek írásos nyilatkozatot kell szolgáltatnia, amely kijelenti, hogy a modulban implementált hitelesítési technika FIPS által jóváhagyott.

5.9.3.3 Kézi kulcs bevitel tesztje

FB_09.40.01:
A fejlesztőnek dokumentálnia kell a kézi kulcs bevitel tesztjét. Attól függően, hogy hiba detektáló kódot vagy duplikált kulcs bevitelt alkalmaznak, a kézi kulcs bevitel tesztje tartalmazhatja a következőket:

- hiba detektáló kódok (EDC):
  - a hiba detektáló kód számítási algoritmusának ismertetése,
  - az ellenőrzési eljárás ismertetése,
  - várható outputok sikeres vagy sikertelen teszt esetén,
- duplikált kulcs bevitel:
  - az ellenőrzési eljárás ismertetése
  - várható outputok sikeres vagy sikertelen teszt esetén

FB_09.40.02:
Ha a hiba detektáló kódot alkalmazzák, a fejlesztői dokumentáció azon részének, amely a kriptográfiai kulcsok formátumát ismerteti (lásd KÖV_07.03), tartalmaznia kell a hiba detektáló kódra vonatkozó részt is.

5.9.3.4 Folyamatos véletlenszám generátor teszt

FB_09.42.01:
Ha a modul hardver véletlenszám generátort implementál, a fejlesztőnek dokumentálnia kell a folyamatos véletlenszám generátor tesztet.

FB_09.43.01:
Ha a modul hardver véletlenszám generátort implementál, a fejlesztőnek dokumentálnia kell a folyamatos véletlenszám generátor tesztet.

5.10 Tervezési biztosíték

5.10.1 Konfiguráció kezelés

FB_10.01.01:
A fejlesztői dokumentációnak tartalmaznia kell a kriptográfiai modul, a modul komponensek és a modul dokumentáció által használt konfiguráció kezelési rendszer leírását.

FB_10.02.01:
A fejlesztő konfiguráció kezelési dokumentációjának tartalmaznia kell a konfigurációs elemek listáját, és azokat az eljárásokat, amik ezek egyedi megkülönböztetésére szolgálnak.

FB_10.02.02:
A fejlesztői dokumentációnak tartalmaznia kell azon eljárást, mely minden hitelesített konfigurációs elem verzióját egyedileg azonosítja.

5.10.2 Továbbítás és működtetés

FB_10.03.01:
A fejlesztői dokumentációnak tartalmaznia kell azokat a lépéseket, melyek a modul biztonságos telepítéséhez, inicializációjához és indításához szükségesek

FB_10.04.01:
A szállítási dokumentációnak tartalmaznia kell azon eljárásokat, melyek a kriptográfiai modul felhatalmazott operátornak való átadása közbeni biztonságának fenntartásához szükségesek.

5.10.3 Fejlesztés

FB_10.06.01:
A fejlesztői dokumentációnak tartalmaznia kell annak leírását, hogy a hardver, szoftver és förmver tervezése során hogyan tartották be a modul biztonsági szabályzatának előírásait.

FB_10.07.01:
A fejlesztőnek be kell nyújtania egy listát a modulban felhasznált összes szoftver és förmver komponensről.

FB_10.07.02:
A fejlesztőnek egy megjegyzésekkel ellátott forrás listát kell beadnia a listában felhasznált összes szoftver és förmver és komponensről.

FB_10.08.01:
A fejlesztőnek a modulban található összes hardver elemről készített listát kell készítenie.

FB_10.10.01:
A fejlesztő funkcionális specifikációjának le kell írnia a kriptográfiai modult annak minden külső interfészével és portjával.

FB_10.10.02:

A fejlesztő funkcionális specifikációjának meg kell határoznia minden külső interfész célját.

FB_10.12.01:
A fejlesztőnek azonosítania kell minden szoftver és szoftver komponenst, ami nem magas-szintű nyelven lett írva, és magyarázatot vagy indoklást kell szolgáltatni arról, hogy miért alacsony-szintű nyelven készültek. A magyarázat hivatkozhat a magas-szintű nyelv hiányára vagy a szoftver/förmver teljesítménynövelésének igényére.

FB_10.13.01:
A fejlesztőnek olyan dokumentációt kell készítenie, mely a felhasznált hardver komponenseket magas-szintű nyelven írja le.

5.10.4 Támogató dokumentáció

FB_10.23.01:
A fejlesztői dokumentációnak minden olyan információt tartalmaznia kell, mely a KÖV_10.21-ben, a KÖV_10.22-ben és a KÖV_10.23-ban megjelennek.

FB_10.23.02:
A kriptográfiai tisztviselő dokumentációjának rendelkezésre kell állnia a kriptográfiai tisztviselő számára.

FB_10.25.01:
A fejlesztői dokumentációnak minden olyan információt tartalmaznia kell, mely a KÖV_10.24-ben és a KÖV_10.25-ban megjelennek.

FB_10.25.02:
A felhasználói dokumentációnak rendelkezésre kell állnia a felhasználó számára.

6. A minősített hitelesítés-szolgáltatókra vonatkozó járulékos funkcionális és biztonsági követelmények

Az alábbiakban áttekintjük azokat az irányadó követelményrendszerekből adódó követelményeket, melyek egy minősített hitelesítés-szolgáltató által használt "biztonságos" kriptográfiai modulra vonatkoznak. Azokra a funkcionális és biztonsági követelményekre szorítkozunk, melynek teljesülését egy 3-as biztonsági szintű FIPS 140-2 értékelés/tanúsítás nem biztosítja automatikusan.
Az alábbiakban a CEN 14167-1 munkacsoport egyezmény jelöléseit alkalmazzuk, lábjegyzetként pedig egyenként utalunk a magyar jogszabályokban megfogalmazott megfelelő követelményekre.

6.1 Elektronikus aláírás hitelesítés szolgáltatásra vonatkozó követelmények

Ezen szolgáltatás keretében a követelmények a minősített hitelesítés-szolgáltató saját kulcsainak gondozására irányulnak. Az alábbiakban a kulcsok alábbi kategóriáit fogjuk megkülönböztetni[19]:

1. **Minősített tanúsítvány aláíró kulcsok.** A tanúsítvány előállítás kulcspárja minősített tanúsítványok létrehozásához.
2. **Infrastrukturális kulcsok**. Ezeket a kulcsokat a megbízható rendszerek olyan folyamatokhoz használják, mint pl. tanúsítvány állapot válaszok aláírása, kulcs-egyeztetés, alrendszer hitelesítés, napló aláírás, tárolt vagy továbbított adatok rejtjelezése stb. (A rövid életciklusú párbeszéd kulcsokat nem tekintjük infrastrukturális kulcsoknak.)
3. **Megbízható rendszervezérlési kulcsok.** Ezeket a kulcsokat személyek használják a megbízható rendszer használatára vagy kezelésére, és hitelesítési-, aláírási- vagy bizalmassági szolgáltatásokat biztosíthatnak a rendszerrel kölcsönhatásba kerülő személyek számára.
4. **Rövid életciklusú munkaszakasz kulcsok**. Egyszeri tranzakciókhoz, rövid ideig használatban lévő kulcsok.

[KM1.1][20]
A minősített tanúsítvány aláíró kulcsokat biztonságos kriptográfiai modulban kell előállítani.

[KM1.2][21]
A [KM1.1]-ben említett kriptográfiai modulnak tanúsítvánnyal igazoltan meg kell felelnie az alábbi szabványok legalább egyikének:

- [FIPS 140-1], 3-as (vagy magasabb) biztonsági szint,
- [CEN: CMCSO-PP, HSM-PP],
- [ITSEC][22].

[KM1.3][23]
A kriptográfiai modul a minősített tanúsítvány aláíró kulcsokat csak kettős ellenőrzés alatt állíthatja elő[24].

[KM1.4][25]
Az infrastrukturális kulcsokat biztonságos kriptográfiai modulban kell előállítani.

[19] Mely kulcs kategóriák megegyeznek a 2/2002 MeHVM 73. pontjában definiáltakkal.
[20] Lásd a 2/2002 MeHVM rendelet 75. pontját.
[21] Lásd a 2/2002 MeHVM rendelet 75. pontját.
[22] A kriptográfiai modul [ITSEC] szerint is kiértékelhető, amennyiben a gyártó/szolgáltató bizonyítja, hogy minimálisan ITSEC E3/high szerinti értékelést alkalmazva az [ITSEC]-ben használt biztonsági követelmények kielégítik a fenti szabványok egyikét. Ha ezek a kritériumok teljesülnek, el kell fogadni, hogy a modul teljesíti a [KM1.2], [KM1.5] és [TS4.2] előírásait is.
[23] Lásd a 2/2002 MeHVM rendelet 76. pontját.
[24] Megjegyzés: A kettős ellenőrzési követelmény teljesíthető akár közvetlenül a kriptográfiai modul által, akár úgy, hogy a hitelesítés-szolgáltató kettős személyi ellenőrzést alkalmaz.
[25] Lásd a 2/2002 MeHVM rendelet 77. pontját.

[KM1.5][26]
A [KM1.4]-ben említett kriptográfiai modulnak tanúsítvánnyal igazoltan meg kell felelnie legalább a [FIPS-140-1] 2-es szintjének, vagy más ennek megfelelő szabványnak [27].

[KM1.6][28]
A rendszervezérlési kulcsokat biztonságos kriptográfiai modulban kell előállítani[29].

[KM1.7][30]
Minden kulcselőállításnak meg kell felelnie az alábbiak valamelyikének:
- valódi (hardver) véletlen generálás legalább 128 bit szabadsági fokkal,
- pszeudó véletlen generálás egy legalább 128 bit hosszúságú "seed" kulcs mellett[31].

[KM6.1][32]
Minden magán- vagy titkos kulcsot biztonságosan kell tárolni.

[KM6.2][33]
A minősített tanúsítványokat aláíró kulcsot biztonságos kriptográfiai modulban kell tárolni, mely megfelel a [KM1.2]-ben rögzített tanúsítvánnyal történő igazolási követelményeknek.
A titkos/magán infrastrukturális kulcsokat biztonságos kriptográfiai modul(ok)ban kell tárolni, mely(ek) megfelel(nek) a [KM1.5]-ben rögzített tanúsítvánnyal történő igazolási követelményeknek.

[KM6.3][34]
A magán- vagy titkos rendszervezérlési kulcsokat biztonságos kriptográfiai modul(ok)ban kell tárolni.

[KM6.4][35]
Bármilyen, biztonságos kriptográfiai modulban tárolt kulcs modulból történő exportálásakor a modulnak gondoskodnia kell a kulcs védelméről. Érzékeny kulcsadatok nem védett módon történő tárolása tilos.
Minősített tanúsítvány aláíró kulcs csak további biztonsági mechanizmusok alkalmazása esetén tárolható és menthető. Ez megtehető például az "m az n-ből" technikák alkalmazásával, ahol m azon komponensek darabszáma a teljes n komponensből, amelynek ismeretében a kulcs inicializálása sikeresen elvégezhető. A hiba esetén alkalmazandó helyreállításra az m = 60% * n érték javasolt (azaz ha n=3, akkor m=2, ha n=4 akkor m=3, ha n=5 akkor m=3,…).
[CG1.4][36]
A minősített tanúsítvány aláírására használt kulcsot csak a minősített tanúsítványok, illetve esetlegesen a rájuk vonatkozó visszavonási listák aláírására szabad felhasználni.

[CG1.6] [37]
A megbízható rendszer által kibocsátott minősített tanúsítványnak meg kell felelnie a Törvény 2. mellékletében meghatározott követelményeknek. Különösen az alábbi tulajdonságoknak kell meglenniük[38]:
1….

[26] Lásd a 2/2002 MeHVM rendelet 77. pontját.
[27] Lásd [KM1.2] alatti megjegyzést.
[28] Lásd a 2/2002 MeHVM rendelet 78. pontját.
[29] Megjegyzés: Ennek a biztonságos kriptográfiai modulnak legalább a [FIPS-140-1] 1-es szintjének, vagy más megfelelő szabványnak kell megfelelnie.
[30] Lásd a 2/2002 MeHVM rendelet 79. pontját.
[31] Lásd a 2/2002 MeHVM rendelet irányelvek 1. sz. mellékletében felsorolt jóváhagyott kulcs generáló algoritmusok listáját.
[32] Lásd a 2/2002 MeHVM rendelet 104. pontját.
[33] Lásd a 2/2002 MeHVM rendelet 106. és 107. pontját.
[34] Lásd a 2/2002 MeHVM rendelet 108. pontját.
[35] Lásd a 2/2002 MeHVM rendelet 109. pontját.
[36] Lásd a 2/2002 MeHVM rendelet 94. és 160.pontját.
[37] Lásd a 2/2002 MeHVM rendelet 162/e alpontját.
[38] Csak az 5. Releváns a kriptográfiai modulra.

2…
3…
4…
5. A megbízható rendszer által a minősített tanúsítvány aláírásához használt aláírási algoritmusok/kulcsok az alábbiak valamelyike lehet:[39]
   - RSA (minimális modulus hosszúság (MinModLen): 1020 bit),
   - DSA (minimális p prímhosszúság (pMinLen): 1024 bit, minimális q prímhosszúság (qMinLen): 160 bit),
   - ECDSA-Fp (qMinLen = 160, r0Min = 10000, MinClass = 200),
   - ECDSA-F2m (qMinLen = 160, r0Min = 10000, MinClass = 200),
6…

6.2 Időbélyegzés szolgáltatásra vonatkozó követelmények

[TS4.1][40]
Az időbélyegzés-szolgáltató aláíró kulcsait biztonságos kriptográfiai modulban kell előállítani és tárolni.

[TS4.2][41]
A TS4.1-ben említett kriptográfiai modulnak tanúsítvánnyal igazoltan meg kell felelnie az alábbi szabványok legalább egyikének:
- [FIPS 140-1] 3-as (vagy magasabb) biztonsági szint,
- [CMCSO-PP, HSM-PP],
- ITSEC[42]

[TS4.3][43]
Az időbélyegzés-szolgáltató rendszervezérlési kulcsait biztonságos kriptográfiai modulban kell tárolni.

[TS4.4][44]
Az időbélyegzéshez használt aláíró kulcsokat kizárólag az adott időbélyegzés-szolgáltató által létrehozott időbélyegek aláírására szabad használni.

[TS4.6][45]
Az időbélyegzés-szolgáltató által használt aláíró algoritmusoknak/kulcsoknak, meg kell felelniük a [CG1.6] alatt felsorolt kriptográfiai követelményeknek.

[39] Lásd a 2/2002 MeHVM rendelet irányelvek 1. sz. mellékletében felsorolt jóváhagyott aláíró algoritmusok listáját.
[40] Lásd a 2/2002 MeHVM rendelet 75. és 212. pontját.
[41] Lásd a 2/2002 MeHVM rendelet 75. és 212. pontját.
[42] Lásd a [KM1.2] alatti megjegyzést.
[43] Lásd a 2/2002 MeHVM rendelet 104. és 213. pontját.
[44] Lásd a 2/2002 MeHVM rendelet 214. pontját.
[45] Lásd a 2/2002 MeHVM rendelet 216. pontját.

6.3 Aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatásra vonatkozó követelmények

[KM1.7][46]
Minden kulcselőállításnak[47] meg kell felelnie az alábbiak valamelyikének:
- valódi (hardver) véletlen generálás legalább 128 bit szabadsági fokkal,
- pszeudó véletlen generálás egy legalább 128 bit hosszúságú "seed" kulcs mellett[48].

[KM3.4][49]
Biztosítani kell, hogy az elektronikus aláírásra szolgáló aláírói kulcsok különbözzenek minden más funkcióra szolgáló kulcstól, mint például a titkosításra szolgálóktól.

[SP1.4][50]
Ha a kulcspár előállítása az aláírás-létrehozó eszközön kívül történik, a kulcspárt előállító kriptográfiai eszköznek tanúsítvánnyal igazoltan meg kell felelnie az alábbi szabványok, szabványjellegű dokumentumok legalább egyikének:
- [FIPS 140-1], 3-as (vagy magasabb) biztonsági szint,
- [CMCKG-PP, HSM-PP],
- [CEN SSCD][51].

[SP1.5][52]
Ha a kulcspár előállítása az aláírás-létrehozó eszközön kívül történik, a kulcspárt biztonságos módon kell az aláírás-létrehozó eszközbe juttatni. A kriptográfiai eszköz és az aláírás létrehozó eszköz között biztonságos útvonalnak kell lennie. Ennek az útvonalnak forráshitelesítést, sérthetetlenséget és bizalmasságot kell biztosítania megfelelő kriptográfiai mechanizmusok használatával.

[46] Lásd a 2/2002 MeHVM rendelet 79. pontját.
[47] Így az aláírás-létrehozó eszközön elhelyezendő aláíró magánkulcs generálása is.
[48] Lásd a 2/2002 MeHVM rendelet irányelvek 1. sz. mellékletében felsorolt jóváhagyott kulcs generáló algoritmusok listáját.
[49] Lásd a 2/2002 MeHVM rendelet 95. pontját.
[50] Lásd a 2/2002 MeHVM rendelet 226. pontját.
[51] Lásd a [KM1.2] alatti megjegyzést.
[52] Lásd a 2/2002 MeHVM rendelet 227. pontját.

7. A Tanúsítási jelentés eredménye, érvényességi feltételei

7.1 A Tanúsítási jelentés eredménye

Az nShield F3 PCI, az nShield F3 Ultrasign PCI és az nShield F3 Ultrasign 32 PCI adapter /nCipher Corporation Ltd./

tanúsítás tárgyát képező verziója
/hardver verzió: nC4032P, nC4032P-300, nC4132P-300,förmver verzió: 2.0.0, 2.0.2/

a Tanúsítás érvényességi feltételeinek[53] együttes teljesülése esetén

ALKALMAS

minősített hitelesítés-szolgáltató által végzett alábbi tevékenységek biztonságos elvégzéséhez:

Valamennyi szolgáltatásra vonatkozóan:

Infrastrukturális kulcsok generálására, tárolására és felhasználására az alábbi célokra:
- tanúsítvány állapot válaszok aláírása,
- tanúsítvány visszavonási listák aláírása,
- naplózott adatállomány aláírása,
- a minősített hitelesítés-szolgáltató megbízható rendszerében a különböző alrendszerek közötti hitelesítésre, kulcsegyeztetésre, tárolt vagy továbbított adatok aláírására.

Megbízható rendszervezérlési kulcsok generálására, tárolására és felhasználására az alábbi célokra:
- a minősített hitelesítés-szolgáltató megbízható rendszerével kölcsönhatásba kerülő személyek által a megbízható rendszer használatára irányuló hitelesítésre és aláírásra.

Elektronikus aláírás hitelesítés szolgáltatás keretén belül:

(Minősített) tanúsítvány aláíró kulcsok generálására, tárolására, (minősített) tanúsítványok létrehozásához való felhasználására, mentésére és helyreállítására.

Időbélyegzés szolgáltatás keretén belül:
Időbélyeg aláíró kulcsok generálására, tárolására, időbélyegző[54] aláírására történő felhasználására.

Aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatás keretén belül:
Az előfizetői (aláírói) kulcspár generálására[55].

7.2 Az eredmények érvényességi feltételei

Az nShield F3 PCI adapter egy bonyolult kriptográfiai eszköz, melyet fejlesztői úgy terveztek, hogy minél általánosabb feltételek között legyen használható, s a felhasználói igények minél szélesebb körét legyen képes kielégíteni. Ennek megfelelően számos biztonsági tulajdonság konfigurálható be, illetve ki rajta.

[53] Lásd a 7.2 “Az eredmények érvényességi feltételei” fejezet feltételeit.
[54] Mely időbélyegzőt a 2001 évi XXXV. törvény az elektronikus aláírásról minősített időbélyegzőként említi.
[55] Amennyiben a kulcspár előállítása az aláírás-létrehozó eszközön kívül történik.

A FIPS 140-2-nek megfelelő módú működtetés (mely a biztonságra helyezi a hangsúlyt, sokszor a hatékonyság és a felhasználói kényelem rovására) számos konfigurációs beállítást megkövetel, s ezek betartása feltétele a tanúsítás érvényességének.

Amennyiben az nShield F3 PCI adaptert egy minősített hitelesítés-szolgáltató kívánja felhasználni biztonságkritikus tevékenységeihez (az általa kibocsátott tanúsítványok aláírására, időbélyeg válaszai aláírására), további követelményeknek kell megfelelni, melyek a felhasználhatóságot tovább korlátozzák, kiegészítő feltételek betartását követelve meg.

Az alábbiakban összefoglaljuk azokat a feltételeket, melyek **együttes** betartása feltétele a Tanúsítvány érvényességének.

7.2.1 Általános érvényességi feltételek

Az alábbi feltételek minden felhasználási mód esetén (tehát a fejlesztő-gyártó cég által igen általánosra tervezett felhasználási kör egészében) szükségesek a megbízható és biztonságos működéshez.

1. Az nShield F3 PCI kriptográfiai modul szolgáltatásait igénybe vevő különböző munkaköröket (nCipher Security Officer, Junior Security Officer, User) betöltő személyek:
   - kompetensek, jól képzettek és megbízhatóak, valamint
   - betartják a különböző útmutatók (Getting Started Guide, nShield User Guide) által leírt, kötelező tevékenységeket.

7.2.2 A FIPS 140-2 megfelelőségből fakadó érvényességi feltételek

Az alábbi feltételek ahhoz elengedhetetlenek, hogy az nShield F3 PCI adapter megfeleljen a FIPS 140-2 3-as biztonsági szintjének.

Az nCipher által engedélyezett alkalmazásnak el kell végeznie a következő szolgáltatásokat, melyekről részletesen az nCipher Security Officer's Guide-ban és a Technical Reference Manualban lehet olvasni.

2. A modul inicializálása

1. Be kell állítani az inicializációs kapcsolót és újraindítani a modult.
2. Az Initialise parancs segítségével el kell érni az Inicializációs állapotot.
3. Egy kulcspárt kell generálni, mely a Security Officer kulcs lesz.
4. Egy logikai tokent kell generálni, mely a Security Officer kulcsát védi.
5. Ennek a logikai tokennek egy vagy több megosztását fizikai vagy szoftver tokenekre (pl. smart card) kell írni.
6. A Security Officer titkos kulcsát key blobként exportálni kell ezen tokenhez tartozóan.
7. A Security Officer nyilvános kulcsát nyílt szövegként kell exportálni.
8. A Set Security Officer szolgáltatás segítségével be kell állítani a modul Security Officer kulcsát és a működési szabályzatát. A FIPS-140 level 3 szintű működéshez legalább a következő jelzőket be kell állítani:
   - NSOPerms_ops_ReadFile
   - NSOPerms_ops_WriteFile
   - NSOPerms_ops_EraseShare
   - NSOPerms_ops_EraseFile
   - NSOPerms_ops_FormatToken
   - NSOPerms_ops_GenerateLogToken
   - NSOPerms_ops_SetKM
   - NSOPerms_ops_RemoveKM
   - NSOPerms_ops_StrictFIPS140
9. A tokeneket és a key blobokat biztonságosan kell tárolni
10. További modul kulcsok generálhatók a felhasználócsoportok megkülönböztetésére.
11. Ettől kezdve lehet munkakulcsokat generálni és a felhasználó engedélyezést elvégezni.

12. Ki kell kapcsolni az inicializációs kapcsolót és újraindítani a modult.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

- A KeySafe használatánál be kell állítani a StrictFIPS 140 jelzőt.
- A new-world program használatánál a –F kapcsolót kell használni.

3. A modul visszaállítása gyári állapotba

Ez az állapot törli a Security Officer kulcsát, a modul aláíró kulcsát és minden modulban tárolt kulcsot.

1. Be kell állítani az az inicializációs kapcsolót és újraindítani a modult.
2. Az Initialise parancs segítségével el kell érni az Inicializációs állapotot.
3. Egy véletlen értéket kell a Security Officer kulcsának lenyomataként betölteni.
4. A Set Security Officer szolgáltatás segítségével be kell állítani a modul Security Officer kulcsát és a működési szabályzatát.
5. Ki kell kapcsolni az inicializációs kapcsolót és újraindítani a modult.

Ezen működés után a modult megfelelően inicializálni kell, mielőtt FIPS módban működhetne.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

4. Új felhasználó létrehozása

1. Egy logikai tokent kell készíteni.
2. Ennek a tokennek egy vagy több megosztását hardver vagy szoftver tokenekre kell írni.
3. Minden kulcsot, mely szükséges a felhasználónak, key blobként exportálni kell ezen tokenekhez tartozóan.
4. Meg kell adni a felhasználó titkos jelszavát és a key blobját.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

5. Felhasználó felhatalmazása kulcskészítésre

1. Új kulcsot kell készíteni, olyan Access Control List (ACL) alatt, mely engedélyezi a UseAsSigningKey kapcsolót. Ezt csak felhatalmazott felhasználó engedélyezheti.
2. Ezt a kulcsot key blobként kell exportálni a felhasználó tokenéhez tartozóan.
3. Az nCipher Security Officer által aláírt tanúsítványt kell generálni, mely
   - tanúsítóként ennek a kulcsnak a lenyomatát tartalmazza,
   - engedélyezi a GenerateKey vagy a GenerateKeyPair működéseket attól függően, hogy mely kulcstípus szükséges,
   - amennyiben a felhasználónak szüksége van kulcstárolásra, engedélyezi a MakeBlob működést, de kizárólag a saját tokenjeikre.
4. Át kell adni a felhasználónak a key blobját és a tanúsítványát.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

6. Felhasználó felhatalmazása Junior Security Officerként való működésre

1. Egy logikai tokent kell generálni, mely védi a Junior Security Officer kulcsát.
2. Ennek a tokennek egy vagy több megosztását hardver vagy szoftver tokenekre kell írni.
3. Egy új kulcspárt kell generálni,

- melynek titkos kulcsa olyan ACL-ben van, mely engedélyezi a Sign és a UseAsSigningKey működést,
- nyilvános kulcsa pedig olyan ACL-ben van, mely engedélyezi az ExportAsPlainText működést.

4. A Junior Security Officer titkos kulcsát key blobként kell exportálni ezen tokenhez tartozóan.
5. A Junior Security Officer nyilvános kulcsát nyílt szövegként kell exportálni.
   - Olyan tanúsítványt kell készíteni, melyet az nCipher Security Officerének kulcsával írnak alá, és tartalmazza ennek a kulcsnak a lenyomatát, mint tanúsító,
   - felhatalmaz a GenerateKey és a GeneratKeyPair működésre,
   - felhatalmaz a GenerateLogicalToken, WriteShare és a MakeBlob működésre, de csak bizonyos modulkulcsokra vonatkozólag.
6. Át kell adni a Junior Security Officernek a szoftver tokenjét, a jelszavát, a key blobját és a tanúsítványát.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

7. A felhasználó azonosítása a tárolt kulcs használatához

1. A LoadLogicalToken szolgáltatás segítségével helyet kell csinálni a logikai tokennek.
2. A ReadShare szolgáltatás segítségével minden megosztást be kell olvasni a logikai tokenről.
3. A LoadBlob szolgáltatás segítségével a kulcsot be kell tölteni a key blobba.
4. A felhasználó innentől minden olyan szolgáltatást el tud érni, mi a kulcs ACL-jében le van írva.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

8. A felhasználó azonosítása új kulcs készítéséhez

1. Amennyiben a felhasználói token még nincs betöltve, a fenti módon kell azt megtenni.
2. A LoadBlob szolgáltatás segítségével kell a felhatalmazott kulcsot betölteni a key blobból.
3. A KeyId segítségével lehet aláírói kulcs tanúsítványt készíteni.
4. A Security Officer tanúsítványával aláírt tanúsítványt kell készíteni a GenerateKey, a GenerateKeyPair és a MakeBlob parancs segítségével.

Megjegyzés: az nCipher grafikus programja, a KeySafe, valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

7.2.3 A minősített hitelesítés-szolgáltatáshoz történő használhatóság kiegészítő feltételei

Egy minősített hitelesítés-szolgáltatónak az nShield F3 PCI felhasználása során az alábbi kiegészítő feltételeket is be kell tartania:

9. RSA aláírási algoritmus használata esetén a minimális modulus hosszúság (MinModLen): 1020 bit legyen.

10. DSA aláírási algoritmus használata esetén a minimális p prímhosszúság (pMinLen) 1024 bit, a minimális q prímhosszúság (qMinLen) 160 bit legyen.

11. Digitálisan aláírni csak 8-cal osztható bithosszúságú blokkot lehet

12. A minősített tanúsítvány aláírására használt kulcsot csak a minősített tanúsítványok, illetve esetlegesen a rájuk vonatkozó visszavonási listák aláírására szabad felhasználni.

13. Bármilyen, biztonságos kriptográfiai modulban tárolt kulcs modulból történő exportálásakor a modulnak gondoskodnia kell a kulcs védelméről. Érzékeny kulcsadatok nem védett módon történő tárolása tilos. Minősített tanúsítvány aláíró kulcs csak további biztonsági mechanizmusok alkalmazása esetén tárolható és menthető. Ez megtehető például az alábbiak valamelyikével is:

- az "m az n-ből" technika alkalmazásával (melyet az nShield F3 PCI támogat), ahol m azon komponensek darabszáma a teljes n komponensből, amelynek ismeretében a kulcs inicializálása sikeresen elvégezhető. A hiba esetén alkalmazandó helyreállításra az m = 60% * n érték javasolt (azaz ha n=3, akkor m=2, ha n=4 akkor m=3, ha n=5 akkor m=3,…).
- az alábbi módszerrel:
- a mentés intelligens kártyákra (tokenekre) történnek,
- a mentés kódolva van a Triple DES titkosító algoritmus alkalmazásával,
- a mentés kódolására alkalmazott titkosító kulcs (Key Encryption Key) legalább két véletlen komponensből van előállítva, s ennek megfelelően legalább két erre felhatalmazott személy együttes jelenléte szükséges a magánkulcs helyreállításához.

14. Az időbélyegzéshez használt aláíró kulcsokat csak időbélyegek aláírására szabad használni.

15. Amennyiben az aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatás keretén belül az előfizetői (aláírói) kulcspár generálása az aláírás-létrehozó eszközön kívül (az nShield F3 PCI kriptográfiai hardverben) történik, biztosítani kell, hogy az elektronikus aláírásra szolgáló aláírói kulcsok különbözzenek minden más funkcióra szolgáló kulcstól, mint például a titkosításra szolgálóktól.

16. Amennyiben az aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatás keretén belül az előfizetői (aláírói) kulcspár generálása az aláírás-létrehozó eszközön kívül (az nShield F3 PCI kriptográfiai modulban) történik, biztosítani kell, hogy az nShield F3 PCI kriptográfiai modul és az aláírás létrehozó eszköz között biztonságos útvonal legyen. Ennek az útvonalnak forráshitelesítést, sérthetetlenséget és bizalmasságot kell biztosítania megfelelő kriptográfiai mechanizmusok használatával.

17. A Tanúsítvány csak a jelenlegi hardver és förmver verzióra érvényes /hardver verzió: nC4032P-150, förmver verzió:2.0.0/. Új förmver verzió upgradje csak az alábbi követelmények együttes teljesülése esetén lehetséges:

- az új förmver verziót a fejlesztő-gyártó cég digitális aláírása hitelesíti,
- az új förmver verziót értékelte egy FIPS 140 értékeléssel meghatalmazott (akkreditált) laboratórium, s erről egy új FIPS tanúsítvány is készül,
- az új förmver verzió minősített hitelesítés-szolgáltatáshoz történő felhasználhatóságát egy erre kijelölt hazai tanúsító szervezet megfelelőségi tanúsítványba foglalja, s mint ilyen, az új verzió is bekerül az NHH biztonságos elektronikus aláírási termék nyilvántartásába.

8. A tanúsításhoz figyelembe vett dokumentumok

8.1 Termékmegfelelőségi követelményeket tartalmazó dokumentumok

Az elektronikus aláírásról szóló 2001. évi XXXV. törvény

16/2001. (IX.1.) MeHVM rendelet az elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó részletes követelményekről

2/2002 (IV.26) MeHVM irányelve a minősített elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó biztonsági követelményekről

FIPS 140-2: Security Requirements for Cryptographic Modules

Derived Test Requirements for FIPS 140-2

ETSI TS 101 456 Policy Requirements for Certification Authorities Issuing Qualified Certificates

CEN 14167-1 munkacsoport egyezmény: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures

8.2 A tanúsítási jelentéshez figyelembe vett egyéb dokumentumok

Kérelem /a tanúsítás elvégzésére/

Kérdőív a tanúsítás kérelmezéséhez

CEN 14167-2 munkacsoport egyezmény: Cryptographic Module for CSP Signing Operation – Protection Profile (CMCSO-PP, HSM-PP)

CEN 14167-3 munkacsoport egyezmény: : Cryptographic Module for CSP Key Genaration Services – Protection Profile (CMCKG-PP, HSM-PP)

FIPS 140-2 Validation Certificate No. 297 /nShield F3 PCI Cryptographic Adapter/

The nShield modules security policy /nShield F3 PCI, nshield F3 PCI Ultrasign v1.2.40/

Getting Started Guide /Version: 4.1.53, Date: 26 July 2002/

nShield User Guide Windows /Version: 4.7.51, Date: 1 August 2002/

Microsoft Windows Server 2003 PKI and deploying the nCipher nShield Hardware Security Module /Version: 1.0, Date: 15 December 2003/

9. Rövidítések

ACL	Access Control List
AES	Advenced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CEN	European Comittee for Standardization
CMCKG	Cryptographic Module for CSP Key Generation Services
CMCSO	Cryptographic Module for CSP Signing Operations
Cprov	Cryptoki (PKCS #11) Provider
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DAC	Data Authentication Code
DCP	Data Ciphering Processor
DES	Data Encryption Standard /FIPS PUB 46-3, FIPS PUB 74, FIPS PUB 81/
DSA	Digital Signature Algorithm /FIPS PUB 186-2/
ECB	Electronic Code Book
EDC	Error Detecting Code
EEPROM	Electrically Erasable Programmable Read Only Memory
EMI	Electromagnetic Interference
EMC	Electromagnetic Compability
ETSI	European Telecommunication Standards Institute
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards Publications
FIPS 140-1	Security Requirements for Cryptographic Modules
FIPS 186-2	Digital Signature Standard
HIK	Host Interface Key
HIMK	Host Interface Master Key
HMAC	Hashed (Keyed) Message Authentication Code
HSM	Hardware Security Module
IDEA	International Data Encryption Algorithm
ISSS	Information Society Standardization System
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria
JSO	Junior Security Officer
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NSO	nCipher Security Officer
OFB	Output Feedback Mode
PCI	Peripheral Component Interconnection
PIN	Personal Identification Number
PKCS	Public Key Cryptographic Standards
PKCS #1	RSA Cryptography Standard
PKCS #7	Cryptographic Message Syntax Standard
PKCS #10	Certification Request Syntax Standard
PKCS #11	Cryptographic Token Interface Standard
PP	Protection Profile
RAM	Random Access Memory
RISC	Reduced Instuction Set Computer
RSA	Rivest-Shamir-Adleman (public key cryptosystem) /ANSI X9.31/
RTC	Real Time Clock
SDRAM	Synchronous Dynamic Random Access Memory
SHA-1	Secure Hash Algorithm /FIPS PUB 180-1/
SSCD-PP	Secure Signature Creation Device – Protection Profile
Triple DES	/FIPS PUB 46-3, ANSI X9.52/
TS	Technical Specification