# Tanúsítási jelentés

# Hung-TJ-067-2014

**nShield F3 6000e /Hw: nC4033E-6K0/,**
**nShield F3 1500e /Hw: nC4033E-1K5/,**
**nShield F3 500e /Hw: nC4033E-500/,**
**nShield F3 10e /Hw: nC4033E-030/,**
**nShield F3 6000e for nShield Connect /Hw: nC4033E-6K0N/,**
**nShield F3 1500e for nShield Connect /Hw: nC4033E-1K5N/ és**
**nShield F3 500e for nShield Connect /Hw: nC4033E-500N/**

**Firmware verzió 2.50.16-3 és 2.51.10-3**

# kriptográfiai modulról

# /Thales e-Security Ltd./

| | |
|---|---|
| Verzió: | 1.0 |
| Fájl: | Hung-TJ-67_2014_v10.pdf |
| Minősítés: | Nyilvános |
| Oldalak: | 83 |

**Változáskezelés**

| Verzió | Dátum | A változás leírása |
|---|---|---|
| v0.1 | 2014.10.21 | A szerkezet felállítása |
| v0.2 | 2014.11.04 | Egyeztetésre kiadott változat |
| **v1.0** | **2014.11.11** | **Végleges verzió** |

A tanúsítási jelentést készítette:

Juhász Judit
HunGuard Kft
Tanúsítási divízió

**Tartalom**

# 1. A Tanúsítási jelentés tárgya, feladata és hatóköre

Jelen Tanúsítási jelentés tárgya az *nShield F3 6000e /Hw: nC4033E-6K0/, nShield F3 1500e /Hw: nC4033E-1K5/, nShield F3 500e /Hw: nC4033E-500/, nShield F3 10e /Hw: nC4033E-030/, nShield F3 6000e for nShield Connect /Hw: nC4033E-6K0N/, nShield F3 1500e for nShield Connect /Hw: nC4033E-1K5N/ és nShield F3 500e for nShield Connect /Hw: nC4033E-500N/* továbbiakban nShield F3 PCIe kriptográfiai adapter család, melyet minősített hitelesítés-szolgáltatás nyújtásához kapcsolódó különböző feladatok ellátására kívánnak felhasználni, mint „biztonságos" kriptográfiai modult..

Az Európai Parlament és a Tanács 910/2014/EU rendelete, a minősített hitelesítés-szolgáltatókra vonatkozó funkcionális és biztonsági követelményeket meghatározó EU-s dokumentum (CEN 14167-1 munkacsoport egyezmény: "Elektronikus aláírásokhoz tanúsítványokat kezelő megbízható rendszerekre vonatkozó biztonsági követelmények") és hazai jogszabályok 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]),
- [CC] EAL4 (vagy magasabb) biztonsági szint
- [ITSEC] E3/high (vagy magasabb) biztonsági szint.

Az nShield F3 PCIe kriptográfiai adapter család 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 PCIe kriptográfiai adapter család 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.

[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.

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 PCIe kriptográfiai adapter család egyéb, köztük a FIPS 140-2 Tanúsítvánnyal igazolt tulajdonságaira, beleértve az alábbiakat:

- A FIPS 140-es Tanúsítvány érvényességébe tartozó, FIPS által jóváhagyott titkosító algoritmusra.
- az nShield adapter család által megvalósított azon kriptográfiai algoritmusokra, melyek a FIPS tanúsítvány kiadásának időpontjában nem FIPS által jóváhagyott algoritmusok, s így már a FIPS értékelés sem terjedt ki rájuk.

A Tanúsítási jelentés további szerkezete a következő:

- Az nShield F3 PCIe kriptográfiai adapter család 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).
- Az nShield F3 PCIe kriptográfiai modul család sebezhetőség vizsgálata (7. fejezet).
- A minősített hitelesítés-szolgáltatás nyújtáshoz való alkalmasság megállapítása, valamint az alkalmazás feltételeinek és korlátainak a meghatározása (8. fejezet).
- A jelen Tanúsítási jelentéshez figyelembe vett dokumentumok jegyzéke (9. fejezet).
- Felhasznált rövidítések jegyzéke (10. fejezet).

# 2. Az nShield F3 PCIe kriptográfiai modul család legfontosabb tulajdonságainak összefoglalása

## 2.1 A Kriptográfiai modul

Az nShield F3 PCIe manipuláció ellen védett hardver biztonsági modulok olyan többfeladatos (multitaszkos) hardver modulok, melyeket nagy egész számokon végzett moduláris aritmetikai műveletek végrehajtására optimalizáltak. A modulok emellett még a kulcsmenedzsment protokollok teljes készletét is kínálják.

Az nShield F3 PCIe hardver biztonsági modulokat a FIPS PUB 140-2 által meghatározott többchipes beágyazott kriptográfiai modulokként definiálták.

| Egység ID | Modellszám | RTC VRAM | SEE | Védő-feltöltés | EMC | Crypto gyorsító | Értékelési szint |
|---|---|---|---|---|---|---|---|
| nShield F3 6000e | nC4033E-6K0 | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |
| nShield F3 1500e | nC4033E-1K5 | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |
| nShield F3 500e | nC4033E-500 | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |
| nShield F3 10e | nC4033E-030 | Igen | Opcionális | Igen | B | nincs | 3 |
| nShield F3 6000e for nShield Connect | nC4033E-6K0N | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |
| nShield F3 1500e for nShield Connect | nC4033E-1K5N | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |
| nShield F3 500e for nShield Connect | nC4033E-500N | Igen | Opcionális | Igen | B | Broadcom 5825 | 3 |

A modulok működési szempontból egyformák, számítási sebességben és a szoftver támogatottságban különböznek csupán.

Minden modul az „N” build szabvánnyal kerül szállításra annak jelzésére, hogy az RHOS szempontjából megfelelnek a legfrissebb EU szabványoknak.

A tanúsított förmver verzió: 2.50.16-3 és 2.50.10-3.

Az inicializációs paramétereket a **NewEnquiry** és a **SignModuleState** szolgáltatások biztosítják. Egy felhasználó a KeySafe GUI vagy parancssoros segédprogramokkal tudja meghatározni, hogy a modul milyen üzemmódban működjön, melyek a modullal együtt kerülnek szállításra, vagy pedig a saját kóddal – ezek a biztonsági határon kívül esnek.

A modulokhoz egyedi alkalmazások írásával kell hozzáférni. Az nCore API teljes dokumentációja hozzáférhető a Thales weblapján.

Az nShield modulok beépített nem-felejtő memóriát tartalmaznak. Léteznek olyan szolgáltatások, melyek lehetővé teszik a memória fájlként történő lefoglalását, elérését. A fájlokhoz hozzáférési engedélyezési listákon (ACL) keresztül lehet hozzáférni, ezek határozzák meg, hogy milyen műveletek a megengedettek a tartalmukon. Az nShield modulok tartalmaznak még beépített valós-idejű órát is.

Az nShield modulok tartalmaznak egy ún. biztonságos végrehajtási környezet (Secure Execution Environment, SEE) technológiát, ami lehetővé teszi a felhasználók számára egy SEE gép betöltését. Egy SEE gép olyan felhasználó által írott kód, amely adott szoftver megszakítás interfészt valósít meg. Ez lehetővé teszi a felhasználók számára nem kriptográfiai kód megvalósítását védett memóriatartományban a modulban, ami kívül esik a logikai biztonsági határokon.

Az SEE kód védett környezetben hajtódik végre. Valahányszor az SEE gép fut, az nCore kernel zárolódik. Amikor az nCore kernel aktív, akkor az SEE gép van zárolva. Az SEE gép nem esik bele a FIPS PUB 140-2 követelményekbe. Mindaddig, amíg az SEE gép aktív, a modul nem FIPS üzemmódban fut.

Az SEE gép nem rendelkezik közvetlen eléréssel a modulban tárolt objektumokhoz. A kriptográfiai funkciók használata érdekében át kell adnia egy jobot az nCore kernelnek az nCore API segítségével. A tesztelés megmutatja, hogy az nCore kernel és az SEE gép közötti interfész biztonságos és egy rosszindulatú SEE gép nem képes hozzáférni az nCore kernel által védett objektumokhoz.

Mielőtt egy felhasználó parancsokat küldhetne az SEE gépnek, létre kell hozni egy SEE World elnevezésű példányt. Egy SEE World-t a modul különálló felhasználóként kezeli, és megfelelően hitelesítenie kell magát, mielőtt parancsot adhatna az nCore kernelnek.

Az nShield F3 PCIe modulokat letiltott SEE funkciókkal adják ki. Ezen funkciók használata érdekében a vásárlónak egy szolgáltatás-engedélyező tanúsítványt kell vásárolnia, ami egy adott modulra biztosítja a funkciókat. Az SEE tulajdonság exportkorlátozás alá esik, így egyes helyeken nem hozzáférhető.

A modul a következő operációs rendszereket futtató számítógéphez csatlakoztatható:

- Windows
- Solaris
- HP-UX
- AIX
- Linux x86

Windows XP és Solaris operációs rendszert használták a FIPS140-2 tanúsításhoz.

## 2.2 Modul portok és interfészek

Az alábbi táblázat felsorolja a modul logikai és azok megfelelő fizikai interfészeit.

| **Logikai interfész** | **Fizikai interfész** |
|---|---|
| Data In | PCIe busz, Soros interfész, 16 utas csatlakozófej |
| Data Out | PCIe busz, Soros interfész, 16 utas csatlakozófej |
| Control In | PCIe busz, Hőmérséklet érzékelő, PSU Monitor, Reset kapcsoló, Üzemmód kapcsoló, 16-utas fej |
| Status Out | PCIe busz, LED, Soros interfész, 16 utas csatlakozófej, |
| Power | PCIe busz |

Az alábbi összetevők nem tartoznak a FIPS 140-2 tanúsítás hatályába:

- szabványos PCIe interfész
- PS-2 soros csatlakozó
- üzemmód kapcsoló
- reset kapcsoló
- állapotjelző LED
- SEE gép

## 2.3 Szerepkörök

A modul a következő szerepköröket támogatja:

**Jogosulatlan**

Kezdetben minden kapcsolat jogosulatlannak tekintett. Ha a modult hármas szintű üzemmódban inicializálják, egy jogosulatlan felhasználó tevékenysége a státusz parancsokra korlátozott, és parancsok kiadására van szükség az engedélyezési protokoll végrehajtásához.

**User**

Egy felhasználó *user* módba lép azáltal, hogy megadja egy szolgáltatás végrehajtásához szükséges jogosítványát. Az egyes szolgáltatásokhoz szükséges pontos akkreditációt a szolgáltatások táblázata tartalmazza.

Egy tárolt kulccsal végzendő művelet végrehajtásához a felhasználónak először be kell töltenie a kulcsblobot. Ha ezt logikai token védi, akkor a felhasználónak meg kell adnia a logikai tokent a megosztások intelligens kártyáról való betöltésével.

Amennyiben a modult 3-as szintű üzemmódban inicializálják, a felhasználó igényel egy tanúsítványt a biztonsági tisztviselőtől (security Officer) egy új kulcs importálásához vagy generálásához. Ez a tanúsítvány egy token által védett kulcshoz kapcsolódik.

Amikor a felhasználó szerepkörben lévő operátor betöltött egy kulcsot, akkor ez felhasználható kriptográfiai műveletek végrehajtására a kulccsal tárolt ACL által meghatározott módon.

Minden kulcsblob tartalmaz egy ACL-t, ami meghatározza, hogy mely szolgáltatás hajtható végre azon a kulcson. Ez az ACL tanúsítványt követelhet meg a tevékenységet engedélyező Security Officer-től. Bizonyos tevékenységekhez, beleértve a tokenek írását, mindig szükség van tanúsítványra.

**nCipher Security Officer**

Az nCipher Security Officer (nCipher biztonsági tisztviselő, NSO) felelős a modul általános biztonságáért. Az NSO egy kulcspár segítségével azonosítható, melynek neve $K_{NSO}$. 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 hitelesítési tokenjeinek (intelligens kártyáinak) létrehozásáért, valamint ezek megfelelő személynek történő biztonságos fizikai átadásáért.

Egy felhasználó akkor rendelkezik NSO szerepkörrel, ha nála van a $K_{NSO}$ magánkulcs része, valamint egy parancs engedélyezéséhez a kulcshoz tartozó **KeyID** kulcsazonosító.

**Junior Security Officer**

Amennyiben az nCipher Security Officer át kívánja adni valamely feladatát másnak, akkor létrehozhat egy kulcspárt, és odaadhatja a kijelölt egyénnek, aki a Junior Security Officer (JSO) lesz. Erre a kulcsra egy ACL fog hivatkozni, és a JSO ezután képes lesz a feladatot engedélyező tanúsítvány aláírására. A JSO kulcsait egy más célra nem használt tokennel védett kulcsblobban kell tárolni.

A JSO szerepkör igazolásához a felhasználónak be kell töltenie a JSO kulcsot, rendelkeznie kell a kulcs **KeyID** azonosítójával, és amennyiben szükséges, a $K_{NSO}$ kulccsal aláírt tanúsítvánnyal, ami a parancsok engedélyezéséhez a feladatokat a kulcshoz delegálja.

A JSO ugyanezen a módon továbbadhatja feladatai egy részét egy új felhasználónak. Az új felhasználó szintén JSO lesz, ha lesz delegálási jogköre, egyébként felhasználó lesz.

## 2.4 Az egyes szerepkörökhöz tartozó szolgáltatások

A szolgáltatásokkal kapcsolatban a tanúsítási jelentés keretein túlmutató további információk megtalálhatók az nShield Fejlesztői útmutatóban és az nShield Fejlesztői referenciakönyvben.

Az alábbi szolgáltatások felhasználói hitelesítést vagy kriptográfiai funkcionalitást biztosítanak. A rendelkezésre álló funkciók attól függnek, hogy a felhasználó a nem feljogosított szerepkörben van-e, engedélyezett felhasználó-e, ideértve a Junior Security Officert (JSO), vagy nCipher Security Officert (NSO). Minden műveletre láthatók a támogatott algoritmusok. A [szögletes zárójelben] lévő algoritmusok nem esnek az operátor határkörébe. Egy szolgáltatás opcionális részében használt algoritmusokat dőlt betűszedés jelzi.

*Megjegyzés: A csillaggal megjelölt algoritmusok csak akkor állnak rendelkezésre, ha a modult 2-es szintű üzemmódban inicializálták, 3-as szintű módban nem. Amennyiben kiválasztják ezeket használatra, akkor a modul nem FIPS-jóváhagyott módban fog működni.*

| **Kulcshozzáférés** | **Leírás** |
|---|---|
| Create | Létrehoz egy memórián belüli objektumot, de nem fedi fel az értékét. |
| Erase | Törli az objektumot a memóriából, intelligens kártyáról vagy nem felejtő memóriából az érték felfedése nélkül. |
| Export | Érték felfedése, de nem teszi lehetővé az érték megváltoztatását. |
| Report | Állapot információ visszaadása. |
| Set | Egy CSP (kritikus biztonsági paraméter) módosítása adott értékre. |
| Use | Létező CSP-vel művelet végrehajtása – a CSP felfedése vagy módosítása nélkül. |

| **Parancs/ Szolgáltatás** | **Szerepkör** | | | **Leírás** | **Kulcs/CSP hozzáférés** | **Kulcs típusok** |
|---|---|---|---|---|---|---|
| | **Unauth** | **JSO/ User** | **NSO** | | | |
| Bignum Operation | Igen | Igen | Igen | Egyszerű matematikai műveleteket hajt végre. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | [SHA-1 és AES vagy Triple DES] |

| | | | | | | |
|---|---|---|---|---|---|---|
| Change Share PIN | Nem | Jelszó | Jelszó | Módosítja a token megosztás rejtjelezéséhez használt jelszót. A felhasználó által megadott jelszót nem közvetlenül használja, hanem először lenyomatot készít róla, majd a modul kulccsal kombinálja. Ehhez a parancs dekódolja a meglévő megosztást a régi jelszóból származtatott régi megosztás kulcs, a modul kulcs és a intelligens kártya azonosítójának segítségével. Ezután származtat egy új megosztás kulcsot az új jelszó, modul kulcs és intelligens kártya azonosító alapján, törli a régi megosztást az intelligens kártyáról, majd kiírja az új megosztás kulccsal rejtjelezett új megosztást. | Beállítja a jelszót egy megosztáshoz, használja a modul kulcsot, használja a megosztás kulcsot, használja a modul kulcsot, létrehozza a megosztás kulcsot, használja az új megosztás kulcsot, exportálja a rejtjelezett megosztást, törli a régi megosztást. | |
| Channel Open | Nem | Handle, ACL | Handle, ACL | Kommunikációs csatornát nyit, amely bulk rejtjelezésre vagy megoldásra használható.<br>A DES-t vagy Triple DES-t CBC módban használó csatornák a Broadcom 5825-öt alkalmazzák a rejtjelezés végrehajtásához. | *Használ* egy kulcs objektumot. | AES, Triple DES |
| Channel Update | Nem | Handle | Handle | Rejtjelezést/megoldást hajt végre egy korábban megnyitott csatornán. A művelet és kulcs a **ChannelOpen**-ben van megadva. | *Használ* egy kulcs objektumot. | AES, Triple DES |
| CheckUser ACL | Nem | Handle | Handle | Meghatározza, hogy egy kulcsobjektumhoz tartozó ACL megenged-e adott felhasználói tevékenységet. | *Használ* egy kulcs objektumot. | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Clear Unit | Igen | Igen | Igen | Nulláz minden betöltött kulcsot, tokent és megosztást. A Clear Unit nem törli a hosszú távú kulcsokat, például a modul kulcsokat. | Objektumokat *töröl(nulláz)*. | Mind |
| Create Buffer | Nem | Cert [Handle] | Cert [Handle] | Adatok betöltéséhez lefoglal memóriaterületet. Ha az adat rejtjelezett, akkor ez a szolgáltatás specifikálja a használt rejtjelezési kulcsot és az IV-t.<br>Ez a szolgáltatás tulajdonság engedélyezés alá esik. A megoldás műveletet a **LoadBuffer** hajtja végre. | *Használ* egy kulcs objektumot. | AES, Triple DES |
| Create SEE World | Nem | Handle Cert | Handle Cert | Létrehoz egy SEE World-öt, egy pufferben tárolt inicializálási adatok átadásával. Ez a parancs ellenőrzi a DSA aláírásokat a pufferen, a biztosított nyilvános kulcs segítségével. Specifikálja azt is, hogy debuggolás megengedett-e vagy sem. A debugging engedélyezéséhez az nCipher Security Officertől szükség van tanúsítványra. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Decrypt | Nem | Handle, ACL | Handle, ACL | Megold egy rejtjelezett szöveget egy tárolt kulccsal és visszaadja a nyílt szöveget | *Használ* egy kulcs objektumot. | AES, Triple DES |

| | | | | | | |
|---|---|---|---|---|---|---|
| Derive Key | Nem | Handle, ACL | Handle, ACL | Ez olyan funkciókat biztosít, melyeket a FIPS 140-2 szabvány kulcs csomagolásnak és tudás megosztásnak ír le – nem a FIPS 140-2 által értett kulcs deriválást szolgáltatja. Létrehoz egy új kulcsobjektumot változó számú egyéb kulcsból, melyek már a modulban tárolódnak és visszaadja az új kulcs handle-jét. Ez a szolgáltatás használható rejtjelezési kulcsok felszeletelésére vagy kombinálására. A szolgáltatást a KDP-nek megfelelően kulcsok csomagolására használják, úgy, hogy egy kulcsszerver szét tudja osztani a becsomagolt kulcsokat a mikro-HSM eszközöknek. | *Használ* egy kulcsobjektumot, *létrehoz* új kulcsobjektumot. | AES, AES kulcscsomagolás, RSA, EC-DH, EC_MQV, Triple DES, PKCS#8*, TLS kulcs deriválás, XOR, DLIES (D/H és Triple DES vagy D/H és AES) |
| Destroy | Nem | Handle | Handle | Eltávolít egy objektumot; ha egy objektumnak több handle-je van a **RedeemTicket** szolgáltatás eredményeként, akkor törli az aktuális handle-t. | *Töröl* egy Impath (belső út), SEEWorld-t, logikai tokent vagy bármilyen kulcsobjektumot. | Mind |
| Duplicate | Nem | Handle, ACL | Handle, ACL | Létrehozza egy kulcsobjektum másodpéldányát ugyanazzal az ACL-el és visszaadja az új példány handle-jét. | *Létrehoz* egy új kulcsobjektumot. | Mind |
| Encrypt | Nem | Handle, ACL | Handle, ACL | Rejtjelez nyílt szöveget tárolt kulccsal és visszaadja a rejtjeles szöveget. | *Használ* egy kulcsobjektumot. | AES, Triple DES, RSA* |
| EraseFile | Csak 2-Es Szinten | Cert | Igen | Eltávolít egy fájlt intelligens kártyáról vagy szoftver tokenből, de logikai tokent nem. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Erase Share | Csak 2-Es Szinten | Cert | Igen | Eltávolít egy megosztást intelligens kártyáról vagy szoftver tokenből. | *Töröl* egy megosztást. | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Existing Client | Igen | Igen | Igen | Új kapcsolatot indít létező kliensként. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Export | Nem | Handle, ACL | Handle, ACL | Ha az egységet 3-as szintű FIPS 140-2-nek való szerepkör, szolgáltatás és kulcskezelés megfeleléssel inicializálták, akkor ez a szolgáltatás csak nyilvános kulcsokra áll rendelkezésre. | *Exportál* egy [nyilvános] kulcsobjektumot. | RSA, DSA, DSA2, ECDSA, ECDSA2, Diffie-Hellman, El-Gamal és ECDH nyilvános kulcsok |
| Feature Enabe | Nem | Cert | Cert | Engedélyez egy szolgáltatást. Az Master Feature Enable kulcs által aláírt tanúsítványt igényel. | *Használja* az Master Feature Enable Key nyilvános felét. | [DSA] |
| Firmware Authenticate | Igen | Igen | Igen | A förmver verziót adja meg. Végrehajt egy zéró tudás kérdés-válasz protokollt, ami HMAC alapú, és lehetővé teszi a felhasználónak annak biztosítását, hogy a modul förmver megegyezzen a Thales által szolgáltatott förmverrel.<br>A protokoll generál egy véletlen értéket HMAC kulcsként való használathoz. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | HMAC |
| Foreign Token Command (Bypass) | Nem | Handle | Handle | Küld egy ISO-7816 parancsot egy intelligens kártyának a **ForeignTokenOpen** által megnyitott csatornán keresztül. | *Használ* egy külső csatornát. | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Foreign Token Open (Bypass) | Nem | FE, Cert | FE | Megnyit egy csatornát a külső intelligens kártyához, ami ISO-7816 parancsokat fogad el. Ez a szolgáltatás nem használható, ha az intelligens kártyát a **FormatToken**-el formázták. A csatornát akkor zárja le, amikor a kártyát eltávolítják az eszközből, vagy a handle megsemmisül. Ez a szolgáltatás tulajdonság engedélyezés alá esik. | *Létrehoz* egy külső csatornát. | |
| FormatToken | Csak 2-Es Szinten | Cert | Igen | Használatra készre formáz egy intelligens kártyát vagy szoftver tokent. | *Használhat* modul kulcsot kérdés-válasz érték létrehozásához. | [AES, Triple DES] |
| Generate Key | Csak 2-Es Szinten | Cert | Igen | Specifikált ACL-el adott típusú szimmetrikus kulcsot generál és visszaadja a handle-t. Opcionálisan visszaad egy tanúsítványt, ami az ACL-t tartalmazza. | *Létrehoz* új szimmetrikus kulcs objektumot. *Beállítja* az ACL-t és az Application data-t (alkalmazási adatokat) az objektumra. Opcionálisan *használja* a modul aláíró kulcsot és *exportálja* a kulcs generáló tanúsítványt. | AES, Triple DES |
| Generate Key Pair | Csak 2-Es Szinten | Cert | Igen | Adott típusú kulcspárt generál specifikált ACL-el a pár mindegyik feléhez. Végrehajt egy páronkénti konzisztencia ellenőrzést a kulcspáron. Visszaadja a két kulcs handle-t. Opcionálisan visszaadja az ACL-t tartalmazó tanúsítványokat. | *Létrehoz* két új kulcsobjektumot. *Beállítja* az ACL-t és Application Data-t ezen objektumokhoz. Opcionálisan *használja* a modul aláíró kulcsot és *exportálja* a két kulcs generáló tanúsítványt. | Diffie-Hellman, DSA, DSA2, ECDSA, ECDSA2, EC-DH, EC-MQV, RSA, ElGamal* |
| Generate KLF | Nem | Fe | Fe | Egy új hosszú távú kulcsot generál. | *Törli* a modul hosszú távú aláíró kulcsát és *létrehoz* egy új hosszú távú aláíró kulcsot. | [DSA, ECDSA] |

| | | | | | | |
|---|---|---|---|---|---|---|
| Generate Logical Token | Csak 2-Es Szinten | Cert | Igen | Létrehoz egy új logikai tokent, amely majd megosztásként írható intelligens kártyákra vagy szoftver tokenekbe. | *Használja* a modul kulcsot, *létrehoz* egy logikai tokent. | [AES vagy Triple DES] |
| Get ACL | Nem | Handle, Acl | Handle, Acl | Visszaadja a megadott handle-re az ACL-t. | *Exportálja* az ACL-t egy kulcsobjektumra. | |
| Get Application Data | Nem | Handle, Acl | Handle, Acl | Egy kulccsal tárolt alkalmazási információkat ad vissza. | *Exportálja* az alkalmazási adatokat egy kulcsobjektumra. | |
| Get Challenge | Igen | Igen | Igen | Visszaad egy véletlen nonce-t, ami tanúsítványokban használható. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Get Key Info | Nem | Handle | Handle | Kompatibilitási okokból megmaradt, a **GetKeyInfoEx**tended az érvényes helyette. | Egy kulcsobjektum SHA-1 lenyomatát *exportálja*. | |
| Get Key Info Extended | Nem | Handle | Handle | Visszaadja egy kulcs lenyomatát ACL-ekben való használatra | Egy kulcsobjektum SHA-1 lenyomatát *exportálja*. | |
| Get Logical Token Info | Nem | Handle | Handle | Kompatibilitási okokból megmaradt, a **GetLogicalToken InfoExtended** az érvényes helyette. | Egy logikai token SHA-1 lenyomatát *exportálja*. | [SHA-1] |
| Get Logical Token Info Extended | Nem | Handle | Handle | Visszaadja a token lenyomatot és egy logikai token megosztások számát. | Egy logikai token SHA-1 lenyomatát *exportálja*. | [SHA-1] |
| GetModuleKeys | Igen | Igen | Igen | Visszaadja az nCipher Security Officer és minden betöltött modul kulcs lenyomatát. | A $K_{NSO}$ és modul kulcsok SHA-1 lenyomatát *exportálja*. | [SHA-1] |
| Get Module Long Term Key | Igen | Igen | Igen | Visszaad egy handle-t a modul aláíró kulcs nyilvános feléhez. Ez használható kulcs generálási tanúsítványok ellenőrzéséhez és modulon belüli útvonalak hitelesítéséhez. | *Exportálja* a modul hosszú távú kulcsának nyilvános felét. | [DSA, ECDSA] |
| Get Module Signing Key | Igen | Igen | Igen | Visszaadja a modul aláíró kulcs nyilvános felét. Ez használható a kulccsal aláírt tanúsítványok ellenőrzéséhez. | *Exportálja* a modul aláíró kulcsának nyilvános felét. | [DSA2] |

| | | | | | | |
|---|---|---|---|---|---|---|
| Get RTC | Igen | Igen | Igen | Megadja a beépített valós idejű óra szerinti időt. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Get Share ACL | Igen | Igen | Igen | Visszaadja egy megosztáshoz a hozzáférési listát. | *Exportálja* egy intelligens kártyán lévő token megosztás ACL-jét. | |
| GetSlot Info | Igen | Igen | Igen | Megadja egy slotban a fizikai token állapotát. Lehetővé teszi a felhasználó számára annak megállapítását, hogy a megfelelő token van-e benn, mielőtt kibocsát egy **ReadShare** parancsot. Ha a tokent egy kérdés-válasz értékkel formázták, a modul kulcsot használja az intelligens kártya hitelesítéséhez. | Egy modul kulcsot *használ,* ha a tokent kérdés-válasz értékkel formázták. | [*AES, Triple DES*] |
| Get Slot List | Igen | Igen | Igen | Megadja a modulból rendelkezésre álló slotok listáját. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| GetTicket | Nem | Handle | Handle | Kér egy ticketet – állandó azonosítót – egy kulcshoz. Ez adható tovább egy másik kliensnek vagy egy SEE World-nek ami aztán visszaveszi azt a **RedeemTicket** segítségével az objektumhoz új handle megszerzéséhez. | *Használ* egy kulcsobjektumot, logikai tokent, Impatht-t (belső utat), SEEWorld-t. | |
| Get World Signers | Nem | Handle | Handle | Visszatér azon kulcsok listájával, melyeket egy SEEWorld aláírásához használnak, amit a kulcslenyomat és a használt aláírási mechanizmus azonosít. | *Használ* egy SEEWorld objektumot. | |
| Hash | Igen | Igen | Igen | Egy érték lenyomatát elkészíti. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | SHA-1, SHA-256, SHA-384, SHA-512 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Impath Get Info | Nem | Handle | Handle | Megadja egy impath (belső út) állapotinformációját. | *Használ* egy Impath-t, *exportál* állapot információt. | |
| Impath Key Exchange Begin | Fe | Fe | Fe | Létrehoz egy új, modulon belüli útvonalat és visszaadja a kulcscsere paramétereket a társ modulhoz való elküldés érdekében. | Impath kulcsok készletét *hozza létre*. | [DSA2 vagy DSA és Diffie-Hellman AES, Triple-DES] |
| Impath Key Exchange Finish | Nem | Handle | Handle | Befejez egy impath kulcscserét. A kulcscsere paramétereket a távoli modultól kéri. | Impath kulcsok készletét *hozza létre*. | [DSA és Diffie-Hellman, AES, Triple-DES] |
| Impath Receive | Nem | Handle | Handle | Adatokat dekódol az Impath megoldó kulccsal. | Impath kulcsot *használ*. | [AES vagy Triple DES] |
| Impath Send | Nem | Handle | Handle | Adatokat rejtjelez az Impath rejtjelezési kulccsal. | Impath kulcsot *használ*. | [AES vagy Triple DES] |
| Import | Csak 2-Es Szinten | Cert | Igen | Betölt egy kulcsot és ACL-t a hosztról és visszaad egy handle-t. Ha az egység 2-es szintű FIPS üzemmódban működik, ezt a műveletet csak nyilvános kulcsokra szabad alkalmazni. Ha az egységet 3-as szintű FIPS 140-2-nek való szerepkör, szolgáltatás és kulcskezelés megfeleléssel inicializálták, akkor ez a szolgáltatás csak nyilvános kulcsokra áll rendelkezésre. | Új kulcsobjektumot *hoz létre*, *beállítja* a kulcsértéket, ACL-t és App data-t. | 2-es szintű mód – minden kulcs típus<br>3-as szintű mód – RSA, DSA, ECDSA, Diffie-Hellman, ECDSA vagy ECDH nyilvános kulcsok |

| | | | | | | |
|---|---|---|---|---|---|---|
| Initialise | Init | Inti | Init | Inicializálja a modult, gyári állapotba állítva azt vissza. Törli az összes NRAM állományt, minden betöltött kulcsot és minden modul kulcsot és a modul aláíró kulcsot.<br>Generál egy új KM0 és modul aláíró kulcsot. Az egyetlen nem nullázott kulcs a hosszú távú aláíró kulcs. Ez a kulcs arra szolgál, hogy kriptográfiai identitást biztosítson egy modulhoz, ami beilleszthető egy PKI tanúsítvány láncba. A Thales kibocsáthat ilyen tanúsítványokat, jelezvén hogy egy modul eredeti nShield modul. Ez a kulcs nem használt más típusú adatok rejtjelezésére. | Kulcsokat *töröl*, KM0-t és KML-t *létrehoz*. | [DSA2] |
| Load Blob | Nem | Handle | Handle | Betölt egy kulcsblobban tárolt kulcsot. A felhasználónak először a blob rejtjelezéséhez használt kulcsot vagy tokent kell betöltenie. | *Használja* a modul kulcsot, logikai tokent vagy archiváló kulcsot, *létrehoz* egy új kulcsobjektumot. | Triple DES és SHA-1, vagy AES, DH, vagy RSA és AES, SHA-1, és HMAC SHA-1 |
| Load Buffer | Nem | Handle | Handle | Aláírt adatokat tölt be egy pufferbe. Több load buffer parancsra lehet szükség az összes adat betöltéséhez, mely esetben a kliens program feladata annak biztosítása, hogy azok helyes sorrendben legyenek.<br>Szükség van a **CreateBuffer**-el létrehozott puffer handle-re. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Load Logical Token | Igen | Igen | Igen | Egy új logikai token számára foglal le területet – az egyedi megosztások ezután összegyűjthetők a **ReadShare** vagy **ReceiveShare** segítségével. Az összeállítás után a token használható a **LoadBlob** vagy **MakeBlob** parancsokban. | Modul kulcsot *használ*. | [AES vagy Triple DES] |
| Make Blob | Nem | Handle, Acl | Handle, Acl | Kulcsblobot hoz létre, ami tartalmazza a kulcsot és vissza is adja azt. Az exportálandó kulcsobjektum bármelyik algoritmus lehet. | *Használja* a modulkulcsot, logikai tokent vagy archiváló kulcsot, *exportálja* a rejtjelezett kulcsobjektumot. | Triple DES és SHA-1, vagy AES, DH, vagy RSA és AES, SHA-1, és HMAC SHA-1 |
| Mod Exp | Igen | Igen | Igen | Moduláris hatványozást hajt végre a parancs által megadott értékekre. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Mod Exp CRT | Igen | Igen | Igen | Moduláris hatványozást hajt végre a parancs által a kínai maradéktétel használatával megadott értékekre. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Module Info | Igen | Igen | Igen | Alacsony szintű állapotinformációt szolgáltat a modulról. Ezt a szolgáltatást a Thales tesztrutinokban való használatra tervezték. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| NewClient | Igen | Igen | Igen | Visszaad egy kliens ID-t. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| New Enquiry | Igen | Igen | Igen | Státusz információt szolgáltat. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| No Operation | Igen | Igen | Igen | Nem végez műveletet, annak megállapítására használható, hogy a modul válaszol-e a parancsokra. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |

| | | | | | | |
|---|---|---|---|---|---|---|
| NVMem Allocate | Nem | Cert | Igen | Fájlként területet foglal a nem felejtő memóriában és beállítja az ACL-eket erre a fájlra. Ez a parancs jelenleg ACL-el védett állományok intelligens kártyára írására használható. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| NVMem Free | Nem | Cert | Igen | Felszabadít nem felejtő memóriában tárolt fájlt. Ez a parancs jelenleg ACL-el védett állományok intelligens kártyára írására használható. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| NVMem List | Igen | Igen | Igen | A nem felejtő memóriában tárolt fájlok listáját adja vissza.<br>Ez a parancs jelenleg intelligens kártyán ACL-el védett állományok listázására használható. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| NVMem Operation | Nem | Cert, ACL | ACL | A nem felejtő memóriában tárolt fájlon hajt végre műveletet. A művelet lehet: olvasás, írás, inkrementálás, dekrementálás stb.<br>Ez a parancs jelenleg ACL-el védett állományok intelligens kártyára írására használható. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Random Number | Igen | Igen | Igen | Véletlen számot generál alkalmazás számára a beépített véletlen szám generátorral. Különálló szolgáltatások állnak rendelkezésre a kulcsok generálására. A véletlen szám szolgáltatást azért tervezték, hogy lehetővé váljon egy alkalmazás számára a véletlen szám forrás elérése saját célokra – például egy on-line kaszinó használhatja a **GenerateRandom** parancsot az alkalmazásaihoz. | Használja a DRBG kulcsot | [AES] |

| | | | | | | |
|---|---|---|---|---|---|---|
| Random Prime | Igen | Igen | Igen | Véletlen prímet generál. Ugyanazt a mechanizmust alkalmazza, mint az RSA és Diffie-Hellman kulcsgenerálás. A prímteszt ellenőrzés megfelel az ANSI X9.31-nek. | Használja a DRBG kulcsot | [AES] |
| Read File | 2-es szint | Cert | Igen | Intelligens kártyáról vagy szoftver tokenből fájlt olvas, de logikai tokent nem.<br>Ez a parancs csak ACL nélküli fájlokat tud olvasni. | Intelligens kártyáról vagy szoftver tokenből fájlt olvas, de logikai tokent nem.<br>Ez a parancs csak ACL nélküli fájlokat tud olvasni.<br><br>Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Read Share | Igen | Igen | Igen | Beolvas egy megosztást egy fizikai tokenről. Amikor elegendő megosztást töltött be, újragenerálja a tokent – ez több **ReadShare** vagy **ReceiveShare** parancsot igényel. | *Használja* a jelszót, modul kulcsot, *létrehozza* a megosztás kulcsot, *használja* a megosztás kulcsot, *létrehozza* a logikai tokent. | [SHA-1, AES vagy Triple DES] |
| Receive Share | Nem | Handle, Encrypted Share | Handle, Encrypted Share | Vesz egy **SendShare**-vel rejtjelezett megosztást és egy jelszót, és felhasználja ezeket a logikai token újralétrehozására - ez több **ReadShare** vagy **ReceiveShare** parancsot igényel. | *Használ* egy Impath kulcsot, *használ* jelszót, modul kulcsot, *létrehoz* egy megosztás kulcsot, *használ* megosztás kulcsot, *létrehoz* logikai tokent. | [AES, Triple DES] |
| Redeem Ticket | Nem | Ticket | Ticket | Handle-t kap az aktuális névtérben a **GetTicket** által létrehozott ticket által hivatkozott objektumra. | *Használ* egy kulcsobjektumot, logikai tokent, Impath–t vagy SEEWorld-t. | |
| Remove KM | Nem | Cert | Igen | Eltávolít egy betöltött modul kulcsot. | *Töröl* egy modul kulcsot. | |

| | | | | | | |
|---|---|---|---|---|---|---|
| SEE Job | Nem | Cert | Igen | Parancsot küld egy SEE World-nek. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Set ACL | Nem | Handle, Acl | Handle, Acl | Egy létező kulcshoz beállítja az ACL-t. A kulcshoz meglévő ACL-nek lehetővé kell tennie a műveletet. | *Beállítja* az ACL-t egy kulcs objektumhoz. | |
| Set Application Data | Nem | Handle, Acl | Handle, Acl | Információkat tárol egy kulccsal együtt. | *Beállítja* egy kulcs objektummal tárolt alkalmazási adatokat. | |
| Set KM | Nem | Cert | Igen | Modul kulcsként tölt be egy kulcs objektumot. | *Használ* egy kulcs objektumot, *beállít* egy modul kulcsot. | AES, Triple DES |
| Set NSO Perm | Init | Init | Nem | Betölt egy kulcslenyomatot az nCipher Security Officer kulcsaként és beállítja a modul által követendő biztonsági szabályzatot. Ez csak akkor végezhető el, amikor az egység inicializálási szakaszban van. | *Beállítja* az nCipher Security Officer kulcslenyomatot. | [DSA kulcs SHA-1 hash-e] |
| Set RTC | Nem | Cert | Igen | Beállítja a valós idejű órát. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Set SEE Machine | Nem | Cert Handle | Handle | Betölti egy puffer tartalmát (amit A **CreateBuffer/LoadBuffer** hozott létre) mint a modulra vonatkozó SEE gépet. Ez a parancs ellenőrzi a pufferen lévő aláírást. Az SEE gép a kriptográfiai határon kívül esik, amikor a modul FIPS módban fut. | *Használ* egy nyilvános kulcsot, ami a pufferben található. | DSA, DSA2*, Triple DES MAC, HMAC |
| Sign | Nem | Handle, Acl | Handle, Acl | Visszaadja a digitális aláírást vagy a nyílt szöveg MAC-ot egy tárolt kulcs használatával. | *Használ* egy kulcs objektumot. | RSA, DSA, DSA2, ECDSA, ECDSA2, Triple DES MAC, HMAC |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sign Module State | Nem | Handle, Acl | Handle, Acl | Aláír egy tanúsítványt ami a modul biztonsági szabályzatát írja le, ahogyan azt a SetNSOPerm beállította. | *Használja* a modul aláíró kulcsát. | [DSA] |
| Send Share | Nem | Hande, Acl | Handle, Acl | Beolvas egy logikai token megosztást és rejtjelezi egy impath kulccsal egy másik modulba való átvitel érdekében, ahol az betölthető a **ReceiveShare** segítségével. | *Használ* egy Impath kulcsot, *exportál* rejtjelezett megosztást. | [AES, Triple DES] |
| Statistics Enumerate Tree | Igen | Igen | Igen | Megadja a rendelkezésre álló statisztikákat. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Statistics Get Value | Igen | Igen | Igen | Visszaad adott statisztikát. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Trace SEE World | Nem | Cert | Igen | Visszaad debug kimenetet egy SEE World-ből. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Update Firmware Services (Hívás: Programming Begin Programming Begin Chunk Programming Load Block Programming End Chunk Programming End) | Monitor | Monitor | Monitor | Ezek a parancsok a förmver frissítés (upgrade) folyamat során használtak.<br>Az egyes parancsokra azért van szükség, hogy betöltsék a kijelölt förmver imaget, elég kis egységekben ahhoz, hogy az interfész által továbbíthatók legyenek.<br>A Thales a **LoadROM** segédprogramot biztosítja az adminisztrátor számára, ami kibocsátja a helyes parancsszekvenciát az új förmver letöltésére.<br>A modul csak akkor fog FIPS–jóváhagyott üzemmódban működni, ha NIST/CSE által érvényesített förmvert telepítünk.<br>Azoknak az adminisztrátoroknak, akik FIPS tanúsítást igényelnek, csak akkor szabad förmvert frissíteni, miután a NIST/CSE új tanúsítványt bocsátott ki.<br>A monitorfunkció ellenőrzi azt is, hogy a förmver verzió szekvencia szám (VSN) legalább olyan magas-e vagy magasabb, mint az aktuálisan telepített förmver VSN-je. | Firmware Integrity és Firmware Confidentiality kulcsot *használ*. *Beállítja* a Firmware Integrity és Firmware Confidentiality kulcsokat. | [DSA2,AES] |
| Verify | Nem | Handle, Acl | Handle, Acl | Tárolt kulcs alkalmazásával ellenőriz egy digitális aláírást. | Használ egy kulcs objektumot. | RSA, DSA, DSA2, ECDSA, ECDSA2, Triple DES MAC, HMAC |

| | | | | | | |
|---|---|---|---|---|---|---|
| Write File | 2-Es Szint | Cert | Igen | Fájlt ír intelligens kártyára vagy szoftver tokenbe, de logikai tokent nem. Ezen fájlok nem rendelkeznek ACL-el, az NVMEM parancsot kell használni ACL-el rendelkező fájlok létrehozására. | Nincs hozzáférés kulcsokhoz vagy CSP-khez (kritikus biztonsági paraméterekhez) | |
| Write Share | Nem | Cert Handle | Handle | Egy új megosztást ír intelligens kártyára vagy szoftver tokenbe. A létrehozható megosztások száma a token létrehozásakor dől el. Minden megosztást ki kell írni, mielőtt a token megsemmisül. | *Beállítja* a jelszót, *használja* a modul kulcsot, *létrehozza* a megosztást, *használja* a jelszót és modul kulcsot, *létrehozza* a megosztás kulcsot, *használja* a modul kulcsot, *használja* a megosztás kulcsot, *exportálja* a rejtjelezett megosztást. | [AES, Triple DES, SHA-1] |

| **Kód** | **Leírás** |
|---|---|
| Nem | A felhasználó nem tudja a szolgáltatást használni ebben a szerepkörben. |
| Igen | A felhasználó minden további engedélyezés nélkül végre tudja hajtani a szolgáltatást ebben a szerepkörben. |
| Handle | A felhasználó végre tudja hajtani a szolgáltatást, ha érvényes handle-t birtokol ezen erőforrásokhoz: kulcs, csatorna, impath (belső út), token, pufferek, **SEEWorld.**<br>A handle az objektum létrehozásakor generált tetszőleges szám.<br>Egy objektum handle-je egyedi módon jellemző az objektumot létrehozó felhasználóra.<br>A ticket szolgáltatások lehetővé teszik, hogy egy felhasználó egy ID-t továbbítson egy objektum felé, amelyet ők másik felhasználó vagy **SEEWorld** számára hoztak létre. |
| ACL | A felhasználó ezt a szolgáltatást csak akkor hajthatja végre egy kulccsal, ha a kulcshoz tartozó ACL ezt megengedi. Az ACL megkövetelheti, hogy a felhasználó egy Securtiy Officer által vagy más kulccsal aláírt tanúsítványt bemutasson.<br>Az ACL specifikálhatja, hogy tanúsítványra van szükség, ekkor a modul ellenőrzi a tanúsítványon lévő aláírást a művelet engedélyezése előtt. |
| jelszó | Egy felhasználó csak akkor tölthet be egy megosztást, vagy módosíthatja a megosztás PIN-t, ha birtokolja a megosztás rejtjelezéséhez használt jelszót. A modul kulcsnak, amivel a jelszót összemosták, szintén jelen kell lennie. |
| Cert | Egy felhasználó csak akkor hajthatja végre ezt a szolgáltatást, ha birtokol egy tanúsítványt az nCipher Security Officertől. Ez a tanúsítvány hivatkozni fog egy kulcsra. A modul ellenőrzi a tanúsítványon lévő aláírást, mielőtt engedélyezi a műveletet. |
| FE | Ez a szolgáltatás nem minden modulban áll rendelkezésre. Engedélyezni kell a **FeatureEnable** szolgáltatással használat előtt. |
| Csak 2-es szinten | Ezen szolgáltatások csak akkor állnak a nem hitelesített felhasználók rendelkezésére, amikor a modult FIPS 140-2 2-es szintű üzemmódban inicializálták. A modul inicializálható FIPS 140-2 3-as szintű szerepkör és szolgáltatás, kulcskezelésnek való megfeleléssel a **FIPS_level3_compliance** állapotjelzővel. Ha ez a jelző be van állítva: |

- a **GenerateKey**, **GenerateKeyPair** és **Import** parancsok az nCipher Security Offcier által aláírt tanúsítvány általi engedélyezést igényelnek.
- az **Import** parancs sikertelen, ha **Sign** vagy **Decrypt** üzenetekhez használható kulcstípust próbálunk meg importálni.
- a **GenerateKey, GenerateKeyPair, Import** és **DeriveKey** műveletek nem engedik meg, hogy ACL-t hozzunk létre egy titkos kulcshoz, amely lehetővé teszi a kulcs nyílt formában való exportálását.

| | |
|---|---|
| Encrypted share (Rejtjelezett megosztás) | A **ReceiveShare** parancs megköveteli, hogy Impath (belső útvonal) kulcs használatával legyen rejtjelezve egy logikai token megosztás, amit a **SendShare** parancs hozott létre. |
| Ticket | A **RedeemTicket** parancs megköveteli, hogy a ticketet a **GetTicket** parancs generálja. |
| Init | Ezen szolgáltatások a modul inicializálására szolgálnak. Csak akkor hozzáférhetők, ha a modul inicializálási üzemmódban van. Ezen modulok inicializálási módba tételéhez fizikai hozzáférés szükséges a modulhoz, és az üzemmód kapcsolót az inicializálás állásba kell kapcsolni. Annak érdekében, hogy visszaállítsuk a modult működési módba, a kapcsolót az *Operational* állásba kell helyezni. |
| monitor | Ezek a szolgáltatások a modul újraprogramozására alkalmazhatók. Csak monitor módban állnak rendelkezésre. A modul monitor módba való állításához fizikai hozzáférés szükséges, és az üzemmód kapcsolót a monitor állásba kell tenni. Annak érdekében, hogy visszaállítsuk a modult működési módba, újra kell inicializálni a modult, majd a kapcsolót az *Operational* állásba kell helyezni. |

## 2.5 Kulcsok

Az nShield modulok által használt minden egyes kulcstípusra az alábbi bekezdés írja le a felhasználói hozzáféréseket.

Az nShield modulok a kulcsokra vagy azok handle-jével, egy tetszőleges számmal vagy pedig az SHA-1 lenyomatával hivatkozik.

**n Cipher 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, biztonságot érintő művelet engedélyezésére szolgáló tanúsítvány aláírására használ.

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

A nyilvános kulcs nyílt szövegként megtalálható a tanúsítványban.

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

Amennyiben a modul inicializálásához a Thales által biztosított szabványos eszközöket használjuk, akkor ez a kulcs egy DSA kulcs, kulcsblobként tárolva, logikai token által védve az Adminisztrátori Kártya Készleten.

**Junior Security Officer kulcs**

Mivel az nCipher Security Officer kulccsal számos feladat végezhető el, egyes feladatokat érdemes átadni egy vagy több Junior Security Officer felhasználónak, mindegyikhez meghatározott felhatalmazást adva adott műveletekhez.

A Junior Security Officer (JSO) felhasználó létrehozásához az NSO egy tanúsítvány aláíró kulcsot generál. Ez a JSO kulcs. Ezt a kulcsot más alkalmazás kulcsokhoz hasonlóan logikai tokennek kell védenie.

A feladat JSO felé delegálásához az NSO-nak ezután egy hozzáférés ellenőrzési listát (ACL-t) tartalmazó tanúsítványt kell készítenie, amely meghatározza az átadott jogokat és annak a JSO kulcsnak a lenyomatát, amire vonatkozik.

A JSO ezután az ACL-ben felsorolt tevékenységeket engedélyezheti – mintha NSO lenne- azáltal, hogy megadja a JSO kulcsot és a tanúsítványt. Amennyiben a JSO kulcs ACL-e

tartalmazza a Sign engedélyt, a JSO 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.

Amennyiben a modul inicializáláshoz a Thales által biztosított szabványos eszközöket használjuk, akkor ez a kulcs egy DSA kulcs, kulcsblobként tárolva, logikai token által védve az Adminisztrátori Kártya Készleten.

**Hosszú távú aláíró kulcs**

Az nShield modul egy 160 és egy 256 bites adatot tárol az EEPROM-ban.

A 160 bites szám a modul förmverében tárolt diszkrét logaritmus csoporttal kombinálva DSA kulcs generálása használt. A 256 bites szám a NIST P521 görbe használatakor egy ECDSA kulcs privát exponense.

Ez a kulcs a **GenerateKLF** szolgáltatás segítségével kicserélhető egy új véletlen értékre. Felhasználható a modul állapot tanúsítvány aláírására a **SignModulState** szolgáltatáson keresztül, nyilvános része pedig a **GetLongTermKey** nem kriptográfiai szolgáltatáson keresztül lekérhető.

Ez az egyetlen kulcs, ami nem nullázódik a modul inicializálása során.

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

**Modul aláíró kulcs**

Az nShield modul inicializálásánál a modul automatikusan egy 3072 bites DSA2 kulcspárt készít, ami a tanúsítvány aláírásnál használatos. Az aláíráshoz SHA-256 lenyomatot használ. A kulcspár titkos része a modul belsejében, EEPROM-ban tárolódik, ahonnan semmilyen körülmények között nem lehet kiszedni. A nyilvános kulcs kinyerhető nyílt szövegként vagy titkosítva van tárolva különböző key blobokban. Ez a kulcs csak arra használt, hogy ellenőrizni lehessen, hogy egy tanúsítványt egy adott modullal generáltak.

**Modul kulcsok**

A modul kulcsok AES vagy Triple DES algoritmussal készülnek és a tokenek védelmére szolgálnak. Az nShield modul az első ilyen modul kulcsot –$K_{M0}$– a modul inicializálásakor készítik. A modul kulcs soha nem kerül ki a modulból. A $K_{M0}$ egy AES kulcs. Az nCipher Security Officer további modul kulcsokat tölthet a modulba. Ezek készülhetnek a modulban, de külső forrásból is származhatnak. A modul kulcsnak beállított kulcs a modulon belül EEPROM-ban tárolódik.

Miután egy kulcs modul kulcsnak lett kinevezve, nem lehet kinyerni a modulból. Kizárólag generáláskor lehet kulcsblobként exportálni az ilyen kulcsokat.

**Logikai tokenek**

A logikai token egy AES vagy Triple DES kulcs, ami a kulcsblobokat védi. A logikai tokenek a modul kulcsokhoz kapcsolódnak. A kulcstípusuk a modul kulcs kulcstípusától függ.

A logikai token készítésénél meg kell határozni az olyan paramétereket mint a megosztások száma, a token újragenerálásához szükséges megosztások száma, a használhatósághoz szükséges darabszám. A végső számérték 1 és 64 között lehet, e két értéket is beleértve. A használathoz szükséges megosztások száma pedig 1 és az összes megosztás közötti egész szám.

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

Betöltés közben a logikai tokenek az objektum tárban tárolódnak.

A token kulcsok sosem exportálódnak a modulból, kivéve a fizikai vagy szoftveres tokenekre történő exportálást. 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 Threshold Sharing 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 közt oszthatók meg. A tokenek tulajdonságai határozzák meg, hogy hány megosztásra van szükség a logikai token újrageneráláshoz. Megosztások csak a tokenekkel együtt generálhatók. A förmver megakadályozza, hogy egy megosztás többször legyen létrehozva.

A logikai tokenek nem használtak kulcs kialakításra.

**Megosztás kulcs**

A megosztás kulcs a logikai token megosztásokat védi, amikor azok hitelesítésre használt intelligens kártyára vagy szoftver tokenre kerülnek. A megosztás kulcs létrehozása a következőképpen történik: titkos prefixumból, modul kulcsból, megosztás számból, intelligens kártya egyedi ID-ből és egy opcionális, felhasználó által megadott 20 bájtból (ami várhatóan az alkalmazásnak megadott jelszó SHA-1 lenyomata) egy üzenet létrehozása, majd ennek inputként való felhasználásával a pRNG funkció felé a megosztás rejtjelezésére használt egyedi kulcs kialakítása – ez vagy egy AES vagy Triple DES kulcs, attól függően, hogy milyen a logikai token kulcstípusa, amit önmagában meghatároz a modul kulcs kulcstípusa. Ez a kulcs nem tárolódik a modulban, egy megosztás betöltésénél mindig újraszámolódik. A megosztás adat tartalmaz egy MAC kódot, melynek sikertelen ellenőrzése esetén a megosztáshoz való hozzáférés nem történik meg.

A megosztás kulcs nem használatos a CSP-k közvetlen védelmére. A logikai tokent újra kell építeni a megosztásokból a Shamir Threshold Sharing séma segítségével, majd dekódolni kell a modul kulccsal. Csak ezután lehet a logikai tokent alkalmazói kulcsok dekódolására felhasználni.

**Belső út kulcsok**

A belső út két modul közötti biztonságos csatornát jelent.

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

A Diffie-Hellman műveletek a CVL #1-es tanúsítás szerint validáltak. A CVL 1-es számú tanúsítás nem felel meg teljes mértékben az SP 800-56A-nak, mert a kulcs deriválási funkció nem tesztelt.

Az egyes modulokhoz a kulcscsere paraméterek az adott modul aláíró kulcsával vannak aláírva. Miután a modulok érvényesítették az aláírásokat, a modul négy szimmetrikus kulcsot származtat a kriptográfiai műveletekhez.

Az aktuálisan érvényes szimmetrikus kulcsok AES vagy Triple DES kulcsok. Az AES a használt, ha mindkét modul 2.50.16-os vagy későbbi förmverrel rendelkezik, a Triple DES pedig akkor, amikor a másik modul régebbi förmvert futtat. A négy kulcs rejtjelezésre, megoldásra, MAC létrehozásra és MAC ellenőrzésre szolgál. A protokoll biztosítja, hogy a Module 1 által rejtjelezésre használt kulcsot a Module 2 megoldásra használja.

**Kulcs objektumok**

A titkosításhoz, dekódoláshoz, aláírás ellenőrzéshez és digitális aláíráshoz használatos kulcsok objektumként jelennek meg az objektumtárban a RAM memóriában. Minden kulcs objektum egy véletlen azonosítóval rendelkezik, ami függ a sessiontől és a felhasználótól.

Minden kulcs egy hozzáférés ellenőrzési listával (ACL) van tárolva, ami meghatározza, hogy az adott kulccsal milyen műveletek végezhetők el. Valahányszor egy operátor új kulcsot generál vagy nyílt szövegű kulcsot importál, az adott kulcs típusának megfelelő ACL-t kell generálni. Az ACL a **SetACL** szolgáltatás segítségével állítható be. Amennyiben az eredeti ACL rendelkezik az **ExpandACL** engedéllyel, később több engedélyt is be lehet állítani hozzá.

Amennyiben az ACL engedélyezi, a kulcs objektumok kulcsblobként exportálhatók. Minden blob egy kulcsot és egy ACL-t tárol. Az ACL határozza meg, hogy a kulcs ezen másolatával milyen művelet végezhető el. A blobban tárolt ACL-nek legalább annyira szigorúnak kell

lennie, mint az az ACL, amely ahhoz a kulcshoz tartozik, amiből a kulcsblob lett generálva. A kulcsblobból betöltött új kulcs objektum ACL-je a kulcsblobból származik. A munka kulcsblobok a logikai tokenekben, titkosítva vannak tárolva. A kulcs objektumok kulcsblobként exportálhatók a modulból archiváló kulcs segítségével. A kulcsblobok tárolhatók a gazdagép lemezén vagy az NVRAM modulban.

A kulcs objektumok csak akkor exportálhatók nyílt szövegként, ha az ACL engedélyezi ezt. FIPS 140-2 3-as szintű szerepkör, szolgáltatás és kulcskezelés megfeleléssel inicializált modul esetén egy magán vagy titkos kulcshoz tartozó ACL nem tartalmazhatja a nyíltként exportálás funkciót. Egy felhasználó egy kulcsot egy másik felhasználónak – vagy egy SEE World-nek- a ticket mechanizmus segítségével adhat át. A **GetTicket** egy kulcsazonosítót vár, és egy ticketet ad vissza. Ez a ticket a kulcsazonosítóra hivatkozik – nem tartalmaz semmilyen kulcsadatot. A ticket bájtblokként adható át a másik felhasználónak, aki azután használhatja a **RedeemTicket** funkciót a sessionre érvényes ugyanazon objektumhoz való kulcsazonosító megszerzéséhez. Mivel az új azonosító ugyanarra az objektumra vonatkozik, a második felhasználót szintén korlátozza az eredeti ACL.

**Session kulcs**

A modul igény szerint generálja a session kulcsokat. Ezek a kulcsok a hozzá tartozó ACL-lel együtt objektumként vannak tárolva az objektumtárban. Ezek a kulcsok bármilyen támogatott algoritmussal készülhetnek.

**Archiváló kulcsok**

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

- Három kulcsos Triple DES kulcsokkal (csak a régi kulcsok kicsomagolására és nyilvános kulcsok becsomagolására használatosak).
- Három kulcsos Triple DES és RSA kulcsok kombinációjával. Ebben az esetben egy véletlen 168 bites Triple DES kulcs titkosítja az archiválandó kulcsot, majd ezt egy RSA kulccsal becsomagolják.
- RSA-t használó sémával:
  3072 bites RSA használt olyan titok kialakítására, amelyből a rejtjelező kulcsok generálódnak. Az RSA kulcsok nyilvános és magánrésze tulajdonosainak már létezniük kell operátorként a modulban.
  A generált kulcsok célja – amelyek akár AES, akár Triple DES kulcsok –, más kulcsok védelme. Az AES az alapértelmezetten alkalmazott a 2.50.16. förmver verzióban (míg a Triple DES visszafelé kompatibilitási célból áll rendelkezésre).
  A kulcsegyeztetési folyamat befejeződése után a modul további kulcsolt hashelési folyamatot futtat az archiválandó nCore Key objektum integritásának védelme érdekében, amelyet a kulcs típus, a kulcs érték és a hozzáférési lista (ACL) képezi. Ez a folyamat HMAC SHA-256-ot használ alapértelmezetten (a HMAC SHA-1 visszafelé kompatibilitási okból áll rendelkezésre).
- Diffie-Hellman-t használó sémával:
  3072 bites Diffie-Hellman, – amely a Jóváhagyott módban történő alkalmazásra engedélyezett –, olyan titok kialakítására használt, amelyből a rejtjelező kulcsok generálódnak. A Diffie-Hellman folyamat mindkét résztvevőjének léteznie kell operátorként a modulban. A generált kulcsok célja – amelyek akár AES, akár Triple DES kulcsok –, más kulcsok védelme. Az AES az alapértelmezetten alkalmazott a 2.50.16. förmver verzióban (míg a Triple DES visszafelé kompatibilitási célból áll rendelkezésre).
  A Diffie-Hellman magánkulcsokat a külső intelligens kártyán kell tárolni, ha az archivált kulcsokat egy későbbi időpontban vissza kell nyerni.
  A kulcsegyeztetési folyamat befejeződése után a modul további kulcsolt hashelési folyamatot futtat az archiválandó nCore Key objektum integritásának védelme érdekében, amelyet a kulcs típus, a kulcs érték és a hozzáférési lista (ACL) képezi. Ez a folyamat HMAC SHA-256-ot használ alapértelmezetten (a HMAC SHA-1 visszafelé kompatibilitási okból áll rendelkezésre).
  Bár a förmver biztosítja, ezt az opciót jelenleg egyik Thales eszköz sem használja. A modulhoz külsőleg kapcsolódó egyéb harmadik fél alkalmazások kihasználhatják ezen opció előnyeit, a fejlesztő belátása szerint.

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

Az archiváló kulcs generálásakor vagy importálásakor be kell állítani a **UseAsBlobKey** opciót az ACL-ben. Az archiváló 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 a megengedett archiváló kulcsok lenyomatának beállítására. Ha meg van adva a lenyomatok listája, akkor más kulcs nem lesz használható.

**Tanúsítvány aláíró kulcsok**

A kulcs objektumhoz kapcsolódó ACL egy tevékenység engedélyezéséhez elvárhatja egy tanúsítvány meglétét. Az elvárt kulcs lehet nCipher Security Officer kulcsa vagy bármilyen más kulcs. A kulcsok az ACL-ben vannak meghatározva egy kulcsot azonosító SHA-1 lenyomattal. A kulcs típusa szintén az ACL-ben van meghatározva, és bár a DSA a szabvány, bármilyen aláírási algoritmus használható; az összes Thales eszköz a DSA-t használja.

Bizonyos szolgáltatások az nCipher Security Officer által aláírt tanúsítványt várják el a végrehajtáshoz.

**Förmver Integritás kulcs**

Minden förmver egy 3072 bites DSA2 kulcspárral van aláírva. Az ezzel a kulccsal képzett aláírások SHA-256-ot használnak.

A modul ellenőrzi az aláírást, mielőtt az új förmver a flashre íródik. A modul csak akkor telepíti az új förmvert, ha az aláírás ellenőrzés sikeres.

A magán kulcs az Thales cég birtokában van.

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

**Förmver Bizalmasság Kulcs**

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

A titkosító kulcs a Thales cégnél és a förmverekben is megtalálható.

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

**Mester Tulajdonság Engedélyező kulcs**

Üzleti okokból nem minden nShield modul ajánlja ki az összes szolgáltatást. Ezen szolgáltatások külön engedélyezéséhez a felhasználónak rendelkeznie kell az Mester Tulajdonság Engedélyező kulcs által aláírt tanúsítvánnyal. Ezzel lehet egy-egy speciális bitet beállítani az EEPROM-ban.

A Mester Tulajdonság Engedélyező kulcs egy DSA kulcspár, melynek titkos része a Thales cég birtokában van, nyilvános része pedig a förmverben található meg. A förmver a flash memóriában található kikapcsolt állapotban, bekapcsoláskor töltődik be az RAM-ba.

**DRBG Kulcs**

A modul a CTR_DRBG-t használja az SP800-90-ből 256 bites AES kulccsal. Ezen kulcs kezdeti állapotának (seed) feltöltése az alaplapi entrópia forrásból történik, valahányszor a modul inicializálódik és az SP800-90-nek megfelelően újraseedelődik további 512 bites entrópiából minden egyes 2048 bájtos output után. Ezt a kulcsot kizárólag a DRBG használja, és soha nem kerül ki a modulból.

## 2.6 Szabályok

**Azonosítás és hitelesítés**

Az nShield modullal való összes kommunikáció egy gazdaszámítógépen futó szerver programon keresztül történik, folyamatok közötti szabványos kommunikációval, UNIX operációs rendszerben socketek, Windows alatt named pipes használatával.

A modul használatához a felhasználónak először be kell lépnie a gazdaszámítógépbe, és el kell indítania egy nShield alkalmazást. Az alkalmazás a szerverhez kapcsolódik, ahonnan kap egy 32 bites tetszőleges számot, a kliens ID-t.

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 egyes szolgáltatásokhoz szükséges engedélyezést az „Az egyes szerepkörökhöz tartozó szolgáltatások" fejezet tartalmazza. Egy felhasználó nem érhet el semmilyen szolgáltatást, amely kritikus biztonsági paraméterhez fér hozzá, anélkül, hogy először intelligens kártyát vagy szoftver tokent ne kellene bemutatnia a rendszerben.

Az nShield modul személyazonosság alapú hitelesítést használ. Minden egyes felhasználó rendelkezik egy intelligens kártyával, amelyen a hitelesítéshez szükséges adatai – a logikai token megosztás – találhatóak 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 kulcsblobok. A 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 jelszóval és a modul kulccsal titkosítva vannak. Ennek következtében a felhasználó csak akkor tudja betölteni a kulcsot, ha birtokában van a tokenben lévő megfelelő megosztásokat tartalmazó intelligens kártyának, a szükséges jelszónak és a modul kulcs a modulba van töltve.

A modul kulcsok a modulon belül EEPROM-ban tárolódnak. Ezeket csak az nCipher Security Officer tudja betölteni vagy törölni, akit a modul inicializálásakor létrehozott nyilvános kulcspár azonosít. A modul újrainicializálása nélkül az nCipher Security Officer kulcsát nem lehet megváltoztatni, ekkor azonban minden modul kulcs törlődik, így minden más kulcshoz való hozzáférés lehetősége elveszik.

A kulcsblob ACL-t is tartalmaz, amely szabályozza, hogy az adott kulccsal milyen szolgáltatás és hányszor 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 újbóli betöltésével kell újrahitelesítenie a kulcsát.

Minden objektumra egy ún. handle hivatkozik. A handle-ek és a kliens azonosítók (**ClientID**) között kereszthivatkozás áll fenn. Amennyiben egy parancs olyan handle-re hivatkozik, amely egy másik kliensnek lett kibocsátva, a parancs elutasításra kerül. Léteznek szolgáltatások, amelyek a handle-eket a kliens azonosítók között mozgatják.

**Hozzáférési ellenőrzési listák**

Minden kulcs objektum rendelkezik hozzáférés ellenőrzési listával (ACL). A felhasználónak akkor kell az ACL-eket meghatároznia, amikor a kulcsokat generálja vagy importálja. Az ACL tartalmazza az összes olyan feladatot, amelyet a kulccsal végre lehet hajtani. Amennyiben egy művelet 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, és az új kulcs objektumra a blob újra betöltésekor vonatkozik.

Az ACL-ek megadhatnak korlátozásokat a műveletekre (vagy művelet csoportokra), ezek globális korlátok vagy megújítható korlátok lehetnek. Az előre beállított (globális) műveletszám átlépése után a kulcs az adott művelethez többet nem használható fel. Megújítható műveletszám esetén el kell végezni a kulcsot védő logikai token újbóli betöltését a kulcs felhasználása előtt.

Az ACL meghatározhat egy tanúsítót is valamely művelethez. Ekkor a szolgáltatás használatához a felhasználónak egy olyan tanúsítvánnyal kell rendelkeznie, amit olyan kulccsal írtak alá, amelynek a lenyomata megtalálható a parancshoz tartozó ACL-ben.

Egy ACL tartalmazhatja a felhasználó által definiált tevékenységeket. Ezek a tevékenységek nem tesznek lehetővé bármilyen műveletet a modulon belül, de tesztelhetők a

**CheckUserAction** szolgáltatással. Így lehetővé válik az SEE programok számára az ACL rendszer saját céljaikra történő használata. Például a payShield ezt a tulajdonságot használja egy Triple-DES kulcs szerepkörének EMV-en belüli meghatározására.

Az ACL specifikálhat továbbá egy hoszt szolgáltatás azonosítót. Ekkor az ACL csak akkor teljesül, ha a szerver hozzáfűzi az illeszkedő szolgáltatás nevét. Ez a tulajdonság azért került bele, hogy korlátozott garanciaszintet biztosítson és a hoszt integritására épül, nem gondoskodik a biztonságról, ha a szerver kompromittálódott vagy nem használják azt.

Az ACL elkészítése bonyolult feladat, a felhasználóknak általában nem is kell maguknak elvégezni ezt a feladatot. A Thales biztosít olyan eszközöket, melyek segítségével szigorú ACL-el kulcsokat tudnak generálni.

**Objektum újrahasználat**

Minden, a modulban tárolt objektumra egy handle hivatkozik. A modul memóriakezeléssel kapcsolatos funkciói biztosítják, hogy egy meghatározott memóriaszegmens csak egy handle-höz tartozhat. A handle meghatározza az objektum típusát, és az összes modul szigorúan ellenőrzi ezen típusokat. Amikor a handle felszabadul, az általa használt memóriaszegmens ténylegesen lenullázódik.

**Hibaállapotok**

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

Amennyiben a modul valamilyen javíthatatlan hibát tapasztal, hibaállapotba kerül. Ezt az állapotjelző LED Morse kód SOS villogása jelzi. Amint a modul hibaállapotba kerül, minden processzor abbahagyja a működést, és az egység nem ad válaszokat. A hibaállapotban az egység nem reagál parancsokra. Az egységet újra kell indítani.

**Biztonsági határvonal**

A fizikai biztonsági határvonal az a műanyag tok, ami tartalmazza a nyomtatott áramkör mindkét oldalán a védőfeltöltést.

Minden kriptográfiai összetevőt védőfeltöltés fed.

Logikai biztonsági határvonal van az nCore kernel és az SEE között.

A „Hatókörön kívüli összetevők” alfejezet tartalmazza azokat az összetevőket, amelyek nem esnek a FIPS 140-2 tanúsítás hatálya alá, mivel biztonsági szempontból nem fontosak.

**Állapot információk**

A modul rendelkezik egy állapotjelző LED-el, ami a modul általános állapotát jelzi.

A modul állapotüzenettel tér vissza minden parancsra adott válaszában, ami a parancs állapotát jelzi.

Számos szolgáltatás van, ami státusz információt nyújt.

Amennyiben a modul egy nShield Connect-en belül van telepítve, ez az információ megjeleníthető az nShield Connect előlapján lévő LCD-n.

**Eljárás a modul 3-as szintű FIPS 140-2-nek megfelelő inicializálásához**

Az nShield használatára feljogosított alkalmazásnak az alábbi szolgáltatásokat kell végrehajtania: (további információk: nCipher Security Officer's Guide and Technical Reference Manual).

a. Állítsuk az üzemmód kapcsolót inicializálási pozícióba és indítsuk újra a modult.
b. A KeySafe grafikus interfész vagy a parancssoros **new-world** eszközt használatával specifikálni kell az Adminisztrátori kártyakészletben lévő kártyák számát, és a használandó rejtjelezés algoritmust, a Triple-DES-t vagy AES-t. Annak garantálása céljából, hogy a modul 3-as szint üzemmódba kerüljön, az alábbiakat kell tenni:

- a KeySafe-el válasszuk: „**Strict FIPS 140 Mode”=Yes**.
- a **new-world**-el adjuk meg a **–F**-et a parancssorban

c. Az eszköz kéri a kártyákat és minden kártyához a jelszót.
d. Az összes kártya létrehozása után, állítsuk az üzemmód kapcsolót működési pozícióba, és indítsuk újra a modult.

**Ellenőrzés, hogy a modul 3-as szintű üzemmódban van-e**

Egy operátor ellenőrizni tudja a modul inicializálás státuszát, hogy a modul 3-as szintű üzemmódban van-e:

- a KeySafe megjeleníti a „Strict FIPS 140-2 Mode”=Yes szöveget a modul információs paneljén.
- a parancssors **nfkminfo** eszköz tartalmazza a **StrictFIPS**-et a modul állapotjelzői között.

**A modul visszaállítása gyári állapotba**

1. Tegyük az üzemmód kapcsolót az inicializálási helyzetbe. Húzzuk az Initialisation kapcsolót felső állásba és indítsuk újra a modult.
2. Az **Initialise** parancs segítségével el kell érni az Inicializációs állapotot.
3. Egy véletlen értéket kell a nCipher Security Officer kulcsának lenyomataként betölteni.
4. A Set nCipher Security Officer szolgáltatás segítségével be kell állítani a modul nCipher Security Officer kulcsát és a modul működési szabályzatát.
5. Állítsuk az üzemmód kapcsolót a működési pozícióba. Húzzuk az Initialisation kapcsolót alsó állásba és indítsuk újra a modult.
6. Ezután a művelet után a modult megfelelően inicializálni kell mielőtt FIPS-jóváhagyott módban lehetne használni.

A modul gyári állapotba helyezése:

- megsemmisít minden betöltött logikai tokent, megosztás kulcsot, belső út kulcsot, kulcsobjektumot, munkaszakasz (session) kulcsot;
- törli az aktuális modul aláíró kulcsot és újat generál;
- törli az összes aktuális modul kulcsot, kivéve a Well Known modul kulcsot;
- egy új Zero modul kulcsot generál;
- az nCipher biztonsági tisztviselő kulcsát egy ismert értékre állítja;
- megakadályozza, hogy a modul bármilyen, kulcsblobban tárolt kulcsot betöltsön, mert a továbbiakban már nem áll rendelkezésre a megoldó kulcs.

A modul gyári állapotba állítása nem törli a förmver bizalmassági kulcsot, a hosszú távú aláíró kulcsot, valamint a förmver integritás kulcs és a mester tulajdonság engedélyező kulcs nyilvános részét, mivel ezek a modul kriptográfiai identitását biztosítják és a förmver betöltést vezérlik.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a **new-world** parancssori program ezeket a lépéseket automatikusan elvégzik.

**Új felhasználó létrehozása**

1. Készíteni kell egy logikai tokent.
2. Ennek a tokennek egy vagy több megosztását szoftver tokenekre kell írni.
3. A felhasználó által igényelt minden kulcs típus esetén, exportálni kell a kulcsot kulcsblobként ezzel a tokennel.
4. Meg kell adni a felhasználó titkos jelszavát és a kulcsblobját.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a new-world parancssori program ezeket a lépéseket automatikusan elvégzik.

**Felhasználó felhatalmazása kulcskészítésre**

1. Új kulcsot kell készíteni, olyan hozzáférés ellenőrzési listával (ACL-el), mely csak a **UseAsSigningKey** kapcsolót engedélyezi.
2. Ezt a kulcsot kulcsblobként kell exportálni a felhasználó tokenéhez tartozóan.
3. Az nCipher Security Officer által aláírt tanúsítványt kell generálni, mely
   a. tanúsítóként ennek a kulcsnak a lenyomatát tartalmazza;
   b. engedélyezi a **GenerateKey** vagy a **GenerateKeyPair** műveleteket attól függően, hogy milyen kulcstípus szükséges;
4. Ha az operátornak – szemben a session kulcsokkal – állandó kulcsokat kell létrehoznia, a tanúsítványnak tartalmaznia kell egy bejegyzést, amely lehetővé teszi a **MakeBlob** tevékenységet. A tanúsítvány korlátozhatja az operátort, hogy csak olyan blobokat hozhasson létre, amelyek az operátori kártyakészlettel védettek, azáltal, hogy bele van foglalva a logikai token lenyomata.
5. Át kell adni a felhasználónak a kulcsblobját és a tanúsítványát.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a **new-world** parancssori program ezeket a lépéseket automatikusan elvégzik.

**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 szoftver tokenekre kell írni.
3. Egy új kulcspárt kell generálni,
4. melynek titkos kulcsának ACL-je engedélyezi a **Sign** és a **UseAsSigningKey** működést,
5. nyilvános kulcsának ACL-je pedig engedélyezi az **ExportAsPlainText** műveletet.
6. A Junior Security Officer titkos kulcsát kulcsblobként kell exportálni ezen tokenhez tartozóan.
7. A Junior Security Officer nyilvános kulcsát nyílt szövegként kell exportálni. ·
8. Olyan tanúsítványt kell készíteni, melyet az nCipher Security Officerének kulcsával írnak alá, és tartalmazza ennek a kulcsnak a lenyomatát mint tanúsító,
   a. engedélyezi a **GenerateKey** és a **GeneratKeyPair** műveleteket,
   b. felhatalmaz a **GenerateLogicalToken**, **WriteShare** és a **MakeBlob** tevékenységekre, de ez korlátozható adott modulkulcsra.
9. Át kell adni a Junior Security Officernek a szoftver tokenjét, a jelszavát, a kulcsblobját és a tanúsítványát.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a **new-world** parancssori program ezeket a lépéseket automatikusan elvégzik.

**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 kulcsblobból.
4. A felhasználó ettől a pontttól kezdve minden olyan szolgáltatást el tud érni, mi a kulcs ACL-jében le van írva.
5. Az nCipher Security Officer szerepkör tölti be ezzel az eljárással az nCipher Security Offcier kulcsot. Az nCipher Security Officer kulcsa ezután használható tanúsítványokban további műveletek engedélyezésére.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a **new-world** parancssori program ezeket a lépéseket automatikusan elvégzik.

**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 az engedélyezési kulcsot betölteni a kulcsblobból.

3. A visszakapott **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.

A Thales által biztosított grafikus felhasználói interfész (KeySafe), valamint a **new-world** parancssori program ezeket a lépéseket automatikusan elvégzik.

## 2.7 Fizikai biztonság

A modul minden biztonságilag kritikus összetevője epoxy gyanta bevonattal van védve.

A modul rendelkezik egy törlés gombbal. Ez a gomb öntesztelő módba teszi a modult, mely minden tárolt kulcsobjektumot, logikai tokent és belső út kulcsot töröl, majd lefuttatja az összes öntesztet. A hosszú ideig érvényes biztonságkritikus paramétereket, modul kulcsokat, modul aláíró kulcsokat és a Security Officer kulcsot a modul gyári állapotába való visszaállításával lehet törölni (leírását lásd fentebb).

A modul ellenőrzése

A fizikai biztonságról az alábbiak rendszeres végrehajtásával lehet meggyőződni:

- Meg kell vizsgálni az epoxy gyanta bevonatot, hogy van–e rajta valamilyen látható sérülés.
- Az intelligens kártya olvasó közvetlenül a modulba csatlakozik vagy egy porthoz, amelyet egy olyan eszköz biztosít, amelybe a modul integrálódik és a kábel nem sérült, nem történt rajta beavatkozás. Ahol a modul egy eszközben van, akkor a csatlakozás biztonságát az eszköz védheti pecsétekkel vagy más babrálási bizonyítékkal.

## 2.8 Funkciók erőssége

**Object ID-k támadása**

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

Az objektumokat ObjectID azonosítja, ami szintén egy véletlen 32 bites szám.

Ahhoz, hogy egy másik felhasználó által feltöltött kulcshoz véletlenszerűen hozzá lehessen férni, két darab 32 bites véletlen számot kellene kitalálni. $2^{64}$ lehetőség van, így az $1{:}10^6$ követelmény teljesül.

A modul percenként $2^{16}$ számú műveletet tud végrehajtani, így az egy percen belüli sikeres támadás valószínűsége $2^{16}/2^{64}=2^{48}$, ami lényegesen kevesebb, mint a megkívánt valószínűség $1{:}\ 10^5$ ($\sim 2^{-17}$).

**A tokenek támadása**

Amennyiben a felhasználó a logikai tokenhez csak 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. Ez megakadályozható az alábbiak betartásával:

- nem marad a kártya az olvasóban
  Ha az intelligens kártya nincs az olvasóban, a támadó csak akkor tud hozzáférni a logikai tokenhez, ha kitalálja a token **ClientID**-jét és az **ObjectID**-jét.
- 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 másodperc 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 vagy 256 bites AES kulcsok. A megosztásokat a modul kulcs, a megosztás száma és a kártya azonosítója segítségével generált titkosítás védi. Annak a valószínűsége, hogy egy támadó létre tud hozni egy logikai token megosztást helyes formátumban a modul kulcs és a megosztás kulcs származtatásának pontos ismerete nélkül, és ez véletlenül egy érvényes token lesz, $2^{-168}$ vagy $2^{-256}$.

**Kulcsblobok**

A kulcsblobok a modulon kívül védik a blobokat. Kétféle blobformátum van: közvetett és közvetlen.

Amennyiben a modul AES modul kulccsal van beállítva, a tokent vagy modulkulcsot magába foglaló blobot védő kulcs, egy 256 bites AES kulcsot vesznek és egy nonce-t, továbbá SHA-1 et használnak ahhoz, hogy egy AES rejtjelező kulcsot származtassanak, ami rejtjelezésre szolgál, és egy HMAC SHA-1 kulcsot, ami integritás ellenőrzéshez használt.

Amennyiben a modul Triple DES modul kulccsal konfigurált, a tokent vagy modulkulcsot magába foglaló blobot védő kulcs Triple DES-t használnak és SHA-1et a rejtjelezéshez és integritás biztosításához.

Amennyiben a modul egy új biztonsági környezetben („security world”) van inicializálva, a kulcs helyreállításhoz és jelmondat helyreállításhoz használt blobok egy 3072 bites RSA kulcs nyilvános felét és egy nonce-t vesznek inputként, és SHA-256-ot használnak egy 256 bites AES rejtjelezési kulcs származtatásához, ami rejtjelzésre szolgál, és egy HMAC SHA-256 kulcsot az integritásvédelem biztosításához.

Ha a modul egy régi biztonsági környezetben jött létre, a 2.50.16 förmver kibocsátása előtt, akkor a kulcs helyreállításhoz és jelmondat helyreállításhoz használt blobok egy 1024 bites RSA kulcs nyilvános felét és egy nonce-t vesznek inputként, és SHA-1-et használnak egy 168 bites Triple DES vagy 256 bites AES rejtjelező kulcs származtatásához – a modul kulcshoz választott opciótól függően –, valamint egy HMAC SHA-1 kulcsot az integritás védelemhez.

A förmver támogatja az integrált rejtjelezési sémán alapuló kulcsblobokat is, melyben a Diffie-Hellman szolgál egy mester titok és HMAC SHA-256 kialakítására integritásvédelemre, az AES CBC üzemmódban a rejtjelezésre; vagy HMAC SHA-1 az integritásvédelemre és Triple DES CBC módban a rejtjelezéshez. Azonban, ez az opció jelenleg egyetlen Thales eszközben sem alkalmazott.

Az SP800-131-nek megfelelő biztonsági környezetben használt minden séma legalább 128 bites biztonságot ad. A visszafelé kompatibilitás miatt használt biztonsági környezetben alkalmazott mechanizmusok pedig legalább 80 bites biztonságot.

**Belső út kulcsok**

A belső út kulcsok védik a modulok közötti titkosított megosztások átvitelét.

Belső út kialakításakor, feltéve, hogy mindkét modul az SP800-131-nek megfelelően konfigurált, a modul ellenőrzi a 3072 bites DSA aláírást SHA-256 lenyomattal a másik modul identitásának ellenőrzése céljából. Ezután 3072 bites Diffie-Hellman kulcscserét hajt végre egy 256 bites AES rejtjelezési és MAC kulcsok kialakításához, amelyek a csatornát fogják védeni. Ez a megoldás a rejtjelzett csatorna minimum 128 bites biztonságát garantálja.

Egyébként mindkét modul 1024 bites DSA aláírást használ a másik modul identitásának ellenőrzéséhez. Majd egy 1024 bites Diffie-Hellman kulcscserét hajtanak végre a 168 bites Triple DES rejtjelezési kulcsok kialakításához, a csatorna védelmét biztosítandó. Ez minimum 80 bites biztonságot garantál a rejtjelezett csatornának.

*Megjegyzés: A csatornán átküldött megosztások titkosítottak a megosztási kulccsal, dekódolás csak a fogadó oldalon történik.*

**KDP kulcs ellátás**

A kulcsoknak a modultól egy mikro HSM-hez való továbbítására használt KDP protokoll 1024 bites DSA aláírást használ a végpont azonosítására és 2048 bites Diffie-Hellman kulcscserével végzi a Triple-DES vagy AES kulcsok egyeztetését, melyeket az átvitelben használt kulcsok titkosítására használnak és legalább 100 bit biztonsági erősséget nyújtanak a rejtjelezett csatornára.

**Származtatott kulcsok**

Az nCore API a kulcsszármaztatás és csomagolási lehetőségek széles választékát nyújtja, melyeket a felhasználó a protokolljában alkalmazhat.

Minden kulcsra igaz, hogy ezek a mechanizmusok csak akkor állnak rendelkezésre, ha az operátor explicit módon engedélyezi azokat a kulcs ACL-jében, amikor a kulcsot generálják vagy importálják.

Az ACL nemcsak a használható mechanizmust specifikálhatja, de a használható specifikus kulcsokat is, ha ezek ismertek.

| Mechanizmus | Használat | Megjegyzés |
|---|---|---|
| Kulcsszeletelés | Egy szimmetrikus kulcsot különálló részekre bont szétosztott tudás kulcs exportáláshoz | A komponensek bájtblokkok. |
| TLS mester kulcs származtatás | TLS session létesítése | |
| Kulcs becsomagolás | Rejtjelez egy kulcsobjektumot egy másikkal a becsomagolt kulcs exportálásának lehetővé tétele céljából. | Használható bármelyik támogatott rejtjelezési mechanizmus, ami elfogad egy bájtblokkot. Az operátornak biztosítania kell, hogy olyan becsomagoló kulcsot válasszanak, ami egyforma erősségű az átviendő kulccsal. |

## 2.9 Öntesztek

A modul áram alá helyezésével önteszt állapotba kerül. A modul önteszt állapotba kerülhet továbbá az egység újraindításával (reset), ami a törlés gomb megnyomásával tehető meg.

Az önteszt módban a modul törli a fő RAM-ot, biztosítva ezáltal, hogy minden betöltött kulcs vagy engedélyezési információ törlődjön, majd végrehajtja az önteszt lépéseket, melyek tartalmazzák az alábbiakat:

- a hardver komponensek működési tesztje
- a förmver integritás ellenőrzése, az SHA-1 lenyomat ellenőrzése
- statisztikai próba a véletlen szám generátoron
- ismert válasz és páronkénti konzisztencia ellenőrzés az összes jóváhagyott és engedélyezett algoritmusra minden jóváhagyott üzemmódban és a DRBG-re
- MAC ellenőrzés az EEPROM tartalmára, biztosítandó, hogy helyesen inicializálódott.

Ez a lépéssorozat pár másodpercet vesz igénybe, miután a modul a Pre-Maintenance (karbantartás előtti), Pre-Initialisation (inicializálás előtti), Uninitialised (inicializálatlan) vagy Operational (üzemi) állapotba kerül, az üzemmód kapcsoló helyzetétől és az EEPROM tartalom érvényességétől függően.

A bekapcsolás alatt a modul folyamatosan figyeli a belső hőérzékelő által rögzített hőmérsékleteket. Amennyiben ez eltér az üzemi működés során megengedettől, akkor hibaállapotba lép.

A modul szintén folyamatosan figyeli a hardver entrópia forrást és a jóváhagyott AES-256 alapú DRBG-t. Ha bármelyik helytelen, a modul hibaállapotba kerül.

Amikor a förmvert frissítik, a modul ellenőrzi a DSA aláírást az új förmver image-n, mielőtt a flash memóriába írja.

A hibaállapotban a modul LED folyamatosan villog a Morse kód SOS jelzésével, amit a hibát jelző állapotkód követ. Minden egyéb input vagy output le lesz tiltva.

**Förmver betöltési teszt**

Amikor egy adminisztrátor új förmvert tölt be, a modul beolvassa a kijelölt image-t a munkamemóriába. Majd a következő teszteket hajtja végre az image-n, mielőtt az aktuális alkalmazást lecserélné:

- Az image érvényes aláírást tartalmaz-e, amit a modul a Firmware Integrity Key (förmver sértetlenségi kulcs) segítségével tud ellenőrizni.
- Az imaget a modulban tárolt Firmware Confidentality Key-vel (förmver bizalmassági kulcs) rejtjelezték.
- Az image-re vonatkozó Version Security Number-nek legalább olyan magas értéknek kell lennie, mint a tárolt érték.

A förmver csak akkor íródik az állandó tárba, ha mindhárom teszt sikeres.

A förmver frissítése törli az nCipher Security Officer kulcsát és minden tárolt modul kulcsot. A modul nem lép át működési módba addig, amíg az adminisztrátor megfelelően újra nem inicializálja.

## 2.10 Támogatott algoritmusok

FIPS által jóváhagyott algoritmusok:

Szimmetrikus rejtjelezés

- AES
  Certificate #397 CBC üzemmód (Csak „Channel Open” és „Channel Update” szolgáltatások – ezek rátöltve a Broadcom processzorra)
  Certificate #1579 (minden más szolgáltatás)
  ECB, CBC GCM és CMAC üzemmódok
- Triple DES
  Certificate #435 CBC üzemmód (Csak „Channel Open” és „Channel Update” szolgáltatások szolgáltatások - ezek rátöltve a Broadcom processzorra)
  Certificate #1579 (minden más szolgáltatás)
  ECB, CBC GCM and CMAC modes

Hash és üzenethitelesítés

- AES CMAC AES Certificate #1579
- AES GMAC Certificate #1579
- HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384 és HMAC SHA-512 Certificate #925
- SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512
  Certificate #1398
- Triple-DES MAC
  Triple-DES Certificate #1035 fejlesztői nyilatkozat

Aláírás

- DSA
  Certificate #487
  FIPS 186-2 és FIPS 186-3 aláírás létrehozás és ellenőrzés
  1024 bites modulus alcsoport 160 bites SHA-1
  2048 bites modulus alcsoport 224 bites SHA-224
  2048 bites modulus alcsoport 256 bites SHA-256
  3072 bites modulus alcsoport 256 bites SHA-256

- ECDSA
  Certificate #192
  FIPS186-2: Aláírás létrehozás és ellenőrzés
  P-192 P-224 P-256 P-384 P-521 K-163 K-233 K-283 K-409 K-571 B-163 B-233 B-283 B-409 és B-571 görbék
- RSA
  Förmver 2.50.16: Certificate #770;
  Förmver 2.51.10: Certificate #1092
  FIPS 186-2: RSASSA-PKCS1_V1_5 aláírás létrehozás és ellenőrzés
  FIPS 186-3: Kulcsgenerálás; RSASSA-PKCS1_V1_5 és RSASSA-PSS aláírás létrehozás és ellenőrzés

Kulcsegyeztetés

- Diffie-Hellman
  CVL Cert. #1 (kulcsegyeztetési, kulcs kialakítási módszer 80 és 256 bit közötti rejtjelezési erősséget biztosít)
- Diffie-Hellman elliptikus görbe
  CVL Cert. #1 (kulcsegyeztetési, kulcs kialakítási módszer 80 és 256 bit közötti rejtjelezési erősséget biztosít);
- EC-MQV
  (kulcs kialakítási módszer 80 és 256 bit közötti rejtjelezési erősséget biztosít)
- RSA
  (kulcs csomagolási, kulcs kialakítási módszer 80 és 256 bit közötti rejtjelezési erősséget biztosít)
- AES
  (AES Certificate #1579, kulcs csomagolási, kulcs kialakítási módszer 128 és 256 bit közötti rejtjelezési erősséget biztosít) AES Key Wrap, AES CMAC számláló mód az SP800-108-nak megfelelően, AES CBC mód

Egyéb

- Deterministic Random Bit Generator
  Certificate #72
  SP 800-90 AES-256 számláló módban

# 3. A FIPS Tanúsítvány eredményeinek összefoglalása

A tanúsítás tárgyát képező kriptográfiai adapter családot egy kriptográfiai modulok tesztelésére az Egyesült Államokban és Kanadában akkreditált laboratórium[3] 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 eredménye az alábbi:**

**Az elért általános biztonsági szint: 3-as**

Az értékelés az alábbi digitális aláíráshoz kapcsolódó, FIPS által jóváhagyott algoritmusok megvalósítását vizsgálta, tesztelte: **SHS (Cert. #1398); DSA (Cert. #487); ECDSA (Cert. #192); RSA (Cert. #770 and #1092); DRBG (Cert. #72);**

Az értékelés az alábbi titkosításhoz kapcsolódó[4], FIPS által jóváhagyott algoritmusok megvalósítását vizsgálta, tesztelte: **AES (Certs. #397 and #1579); Triple-DES (Certs. #435 and #1035); HMAC (Cert. #925); Triple-DES MAC (Triple-DES Cert. #1035, vendor affirmed); CVL (Cert. #1)**

[3] CSC /NVLAP LAB CODE 200426-0 /
[4] jelen Tanúsítási jelentés hatókörén kívül álló,

# 4. Az nShield F3 PCIe modulok 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 PCIe kriptográfiai adapter család értékelését végző laboratórium vizsgálta és igazolta.

Az alábbi jelölést alkalmazzuk:

KÖV_x.y: a FIPS 140-2 x. fejezetének y. biztonsági követelménye.[5]

## 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.

[5] Csak azokat a követelményeket adjuk meg, mely az nShield F3 PCIe kriptográfiai adapter család ténylegesen vonatkoznak, ezért a követelmények sorszámozása nem mindig folyamatos.

**KÖV_01.11:**
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[6] 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.

[6] 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[7], 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

[7] Az nShield F3 PCIe kriptográfiai adapter család támogat több egyidejű operátort.

specifikálni 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[8]

**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.

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

### 4.5.2 Több chipes, beágyazott kriptográfiai modulra vonatkozó követelmények

**KÖV_05.33:**
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[9].

## 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.

[9] Mivel az nShield F3 PCIe kriptográfiai adapter család működési környezete nem képezi az értékelés részét.

**KÖV_07.10:**
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[10]. 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[11], 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[12], 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.

[10] Az nShield F3 PCIe kriptográfiai adapter család megvalósít belső kulcs generálási funkciót.
[11] Az nShield F3 PCIe kriptográfiai adapter család alkalmaz véletlenszám generátort.
[12] Az nShield F3 PCIe kriptográfiai adapter család 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_08.05:**
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 PCIe kriptográfiai adapter család 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 PCIe kriptográfiai adapter család 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[13]. 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.

[13] Az nShield F3 PCIe kriptográfiai adapter család esetében ez: több chipes, beágyazott modul.

## 5.6. Az operációs rendszer biztonsága

Nincsenek követelmények[14].

## 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.

[14] Mivel az nShield F3 PCIe kriptográfiai adapter család működési környezete nem képezi az értékelés részét.

**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 öszehasonlító tesztelést haszálja, 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. 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.

### 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 eredti 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 a 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:

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.
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]**
A minősített tanúsítvány aláíró kulcsokat biztonságos kriptográfiai modulban kell előállítani.

**[KM1.2]**
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][15].

**[KM1.3]**
A kriptográfiai modul a minősített tanúsítvány aláíró kulcsokat csak kettős ellenőrzés alatt állíthatja elő[16].

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

**[KM1.5]**
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 [17].

**[KM1.6]**

A rendszervezérlési kulcsokat biztonságos kriptográfiai modulban kell előállítani.

[15] 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.
[16] Megjegyzés: A kettős ellenőrzési követelmény teljesíthető akár közvetlenül a kriptográfiai modul által, akár úgy, hogy a hitelesítés-szolgáltató kettős személyi ellenőrzést alkalmaz.
[17] Lásd [KM1.2] alatti megjegyzést.

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

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

**[KM6.2]**
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]**
A magán- vagy titkos rendszervezérlési kulcsokat biztonságos kriptográfiai modul(ok)ban kell tárolni.

**[KM6.4]**
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]**
Egy minősített tanúsítvány aláírásához használt kulcsot csak minősített tanúsítványok és opcionálisan a kapcsolódó visszavonási státusz adatok aláírására szabad használni.

**[CG1.6]**
- A megbízható rendszer által kibocsátott minősített tanúsítványnak meg kell felelnie a Törvény 2. mellékletében meghatározott követelményeknek.

## 6.2 Időbélyegzés szolgáltatásra vonatkozó követelmények

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

**[TS4.2]**
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[18]

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

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

[18] Lásd a [KM1.2] alatti megjegyzést.

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

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

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

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

**[SP1.4]**
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][19].

**[SP1.5]**
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.

[19] Lásd a [KM1.2] alatti megjegyzést.

# 7 Az nShield F3 PCIe kriptográfiai modul család sebezhetőség vizsgálata

Az nShield F3 PCIe kriptográfiai modul család vizsgálata kiterjedt a nyilvános adatbázisokban fellelhető sebezhetőségek ellenőrzésére.

Az ellenőrzés eredménye, hogy a *nShield F3 6000e /Hw: nC4033E-6K0/, nShield F3 1500e /Hw: nC4033E-1K5/, nShield F3 500e /Hw: nC4033E-500/, nShield F3 10e /Hw: nC4033E-030/, nShield F3 6000e for nShield Connect /Hw: nC4033E-6K0N/, nShield F3 1500e for nShield Connect /Hw: nC4033E-1K5N/ és nShield F3 500e for nShield Connect /Hw: nC4033E-500N/ /firmware verzió 2.38.4-3 és 2.38.7-3*/ modulokról nyilvános adatbázisban fellelhető sebezhetőség jelen tanúsítás időpontjában nem található.

# 8. A Tanúsítási jelentés eredménye, érvényességi feltételei

## 8.1 A Tanúsítási jelentés eredménye

**A Thales e-Security Ltd. fejlesztésű,**
**nShield F3 6000e /Hw: nC4033E-6K0/,**
**nShield F3 1500e /Hw: nC4033E-1K5/,**
**nShield F3 500e /Hw: nC4033E-500/,**
**nShield F3 10e /Hw: nC4033E-030/,**
**nShield F3 6000e for nShield Connect /Hw: nC4033E-6K0N/,**
**nShield F3 1500e for nShield Connect /Hw: nC4033E-1K5N/ és**
**nShield F3 500e for nShield Connect /Hw: nC4033E-500N/**

**Firmware verzió 2.50.16-3 és 2.51.10-3**

**a Tanúsítás érvényességi feltételeinek[20] 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[21] aláírására történő felhasználására.

**Aláírás-létrehozó eszközön az aláírás-létrehozó adat elhelyezése szolgáltatás keretén belül:**
Az előfizetői (aláírói) kulcspár generálására[22].

[20] Lásd a 8.2 "Az eredmények érvényességi feltételei" fejezet feltételeit.
[21] Mely időbélyegzőt a 2001 évi XXXV. törvény az elektronikus aláírásról minősített időbélyegzőként említi.
[22] Amennyiben a kulcspár előállítása az aláírás-létrehozó eszközön kívül történik.

## 8.2 Az eredmények érvényességi feltételei

Az nShield F3 PCIe kriptográfiai adapter család 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.

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 PCIe kriptográfiai adapter család egy elemét 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.

### 8.2.1 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 PCIe kriptográfiai adapter család szolgáltatásait igénybe vevő különböző munkaköröket (nCipher Security Officer, Junior Security Officer, User) betöltő személyek:

- kompetensek, jól képzettek és megbízhatóak, valamint
- betartják a különböző útmutatók által leírt, kötelező tevékenységeket.

### 8.2.2 A FIPS 140-2 megfelelőségből fakadó érvényességi feltételek

2. Az nShield F3 PCIe kriptográfiai adapter család, hogy a FIPS 140-2 3-as biztonsági szintjének megfelelően működjön, az alábbi módon kell inicializálni:

a. Állítsuk az üzemmód kapcsolót inicializálási pozícióba és indítsuk újra a modult.
b. A KeySafe grafikus interfész vagy a parancssoros new-world eszközt használatával specifikálni kell az Adminisztrátori kártyakészletben lévő kártyák számát, és a használandó rejtjelezés algoritmust, a Triple-DES-t vagy AES-t. Annak garantálása céljából, hogy a modul 3-as szint üzemmódba kerüljön, az alábbiakat kell tenni:
   - a KeySafe-el válasszuk: „**Strict FIPS 140 Mode”=Yes**.
   - a new-world-el adjuk meg a **–F**-et a parancssorban
c. Az eszköz kéri a kártyákat és minden kártyához a jelszót.
d. Az összes kártya létrehozása után, állítsuk az üzemmód kapcsolót működési pozícióba, és indítsuk újra a modult.

Amennyiben egy modult 3-as szinten inicializáltak

- a KeySafe megjeleníti a „Strict FIPS 140-2 Mode”=Yes szöveget a modul információs paneljén.
- a parancssors **nfkminfo** eszköz tartalmazza a **StrictFIPS**-et a modul állapotjelzői között.

### 8.2.3 A minősített hitelesítés-szolgáltatáshoz történő használhatóság kiegészítő feltételei

Egy minősített hitelesítés-szolgáltatónak az nShield F3 PCIe kriptográfiai modul család felhasználása során az alábbi kiegészítő feltételeket is be kell tartania:

3. RSA aláírási algoritmus használata esetén a minimális modulus hosszúság (`MinModLen`): 2048 bit legyen.

4. DSA aláírási algoritmus használata esetén a minimális p prímhosszúság (`pMinLen`) 2048 bit, a minimális q prímhosszúság (`qMinLen`) 224 bit legyen.

5. Az ECDSA aláírási algoritmus használata esetén a következő paraméter feltételek teljesítése szükséges: `qMinLEN=256` SHA256 használata mellett, továbbá `r0Min` nagyobb mint $10^4$ és `MinClass` legalább `200`, ahol a paraméterek jelölése megfelel az ETSI TS 102 176-1 v 2.1.1 –ben leírtaknak.

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

7. SHA-1 vagy annál gyengébb lenyomatoló algoritmus használata tilos.

8. A minősített tanúsítvány (QC) aláírásához használt kulcsot csak minősített tanúsítványok és opcionálisan a kapcsolódó visszavonási státusz adatok (beleértve az azok ellenőrzésére szolgáló tanúsítványt) aláírására szabad használni.

9. 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, 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 vagy AES 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.

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

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

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

13. A Tanúsítvány csak a 8.1 fejezetben megadott hardver és förmver verzióra érvényes. 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 NMHH biztonságos elektronikus aláírási termék nyilvántartásába.

### 8.2.4 Egyéb, az érvényességet befolyásoló megjegyzések

14. A National Institute of Standards and Technology (NIST) által kibocsátott tanúsítványok visszavonásig érvényesek. Így a tanúsítványokban szereplő hardver, förmver és szoftver konfigurációk változatlan formában használhatók.

15. Nyilvános források között jelenleg nem található olyan információ, mely befolyásolná a modul biztonságos működését. Ezt a vizsgálatot legalább 3 évente szükséges elvégezni.

# 9. A tanúsításhoz figyelembe vett dokumentumok

## 9.1 Termékmegfelelőségi követelményeket tartalmazó dokumentumok

Az Európai Parlament és a Tanács 910/2014/EU rendelete ( 2014. július 23. ) a belső piacon történő elektronikus tranzakciókhoz kapcsolódó elektronikus azonosításról és bizalmi szolgáltatásokról

Az elektronikus aláírásról szóló 2001. évi XXXV. törvény

3/2005. (III.18.) IHM rendelet az elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó részletes követelményekről

FIPS 140-2: Security Requirements for Cryptographic Modules

Derived Test Requirements for FIPS 140-2

ETSI TS 102 176-1 V2.1.1 Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms

CEN 14167-1:2003 munkacsoport egyezmény: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures

## 9.2 A tanúsítási jelentéshez figyelembe vett egyéb dokumentumok

Kérelem /a tanúsítás elvégzésére/

FIPS 140-2 Consolidated Certificate No.0018 /1742/

The nShield security policy / v3.2/

# 10. Rövidítések

| | |
|---|---|
| ACL | Acces Control List |
| AES | Advenced Encryption Standard |
| ANSI | American National Standards Institute |
| API | Application Programming Interface |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CEN | European Comittee for Standardization |
| CMCKG | Cryptographic Module for CSP Key Generation Services |
| CMCSO | Cryptographic Module for CSP Signing Operations |
| CSP | Critical Security Parameter |
| CPU | Central Processing Unit |
| 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 140-2 | Security Requirements for Cryptographic Modules |
| FIPS 186-2 | Digital Signature Standard |
| HMAC | Hashed (Keyed) Message Authentication Code |
| HSM | Hardware Security Module |
| IDEA | International Data Encryption Algorithm |
| ITSEC | Information Technology Security Evaluation Criteria |
| MAC | Message Authentication Code |
| NIST | National Institute of Standards and Technology |
| OFB | Output Feedback Mode |
| PCI | Peripheral Component Interconnection |
| PIN | Personal Identification Number |
| PKCS | Public Key Cryptographic Standards |
| PKCS #11 | Cryptographic Token Interface Standard |
| PP | Protection Profile |
| PRNG | Pseudo Random Number Generator |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| RSA | Rivest-Shamir-Adleman (public key cryptosystem) /ANSI X9.31/ |
| RTC | Real Time Clock |
| SEE | Secure Execution Environment |
| SHS | Secure Hash Standard |
| SSCD-PP | Secure Signature Creation Device – Protection Profile |
| Triple DES | /FIPS PUB 46-3, ANSI X9.52/ |
| TS | Technical Specification |