Tanúsítási jelentés

Hung-TJ-012-2003

a
SafeGuard Sign&Crypt Software Development Kit

elektronikus aláíró alkalmazás fejlesztő készletről

/Utimaco Safeware AG/

/verzió: 2.0 /

Tartalom

1. A SafeGuard Sign&Crypt SDK v2.0 legfontosabb tulajdonságainak összefoglalása

1.1 Áttekintés

A SafeGuard Sign&Crypt SDK (a továbbiakban SDK) egy fejlesztő készlet, melynek segítségével különböző felhasználói igényeket kielégítő alkalmazói programok (aláírás alkalmazás összetevők) fejleszthetők, s építhetők be egy elektronikus üzenetkezelő rendszerbe. A segítségével fejleszthető alkalmazói programok digitális aláírások kezelésére, aszimmetrikus kulcsok felhasználására, illetve üzenetek titkosítására és dekódolására is használhatók.

Az SDK a SafeGuard Sign&Crypt alaptermék funkcionalitására épül, mely digitális aláírásra és üzenet titkosításra szolgáló rendszer (s egy párhuzamos ITSEC értékelés és tanúsítás tárgya volt). Az SDK segítségével egy fejlesztő az SDK részét képező CryptWare Client Server API által biztosított funkcionalitásokat saját rendszerébe integrálhatja.

Az SDK segítségével elérhető funkciók segítségével egy feladótól a fogadóhoz továbbított adatokra egyaránt biztosítható az adatok hitelessége, sértetlensége, letagadhatatlansága és bizalmassága.

A SafeGuard Sign&Crypt SDK egy kliens-szerver rendszernek tekinthető, melyben a CryptWare Client Server API és a mögötte álló SafeGuard Sign&Crypt kernel működik szerverként, a CryptWare Client Server API funkciókat meghívó alkalmazás pedig kliensként.

A SafeGuard Sign&Crypt SDK a következő feladatokhoz biztosít funkciókat:

- a kliens alkalmazás és a SafeGuard Sign&Crypt CryptWare Client Server API közötti kommunikáció elindítása és leállítása,
- intelligens kártyáról vagy titkosított lokális fájlból származó magánkulcsok használata,
- a dokumentum fogadói számára biztosított adatbázisokban tárolt nyilvános kulcsok használata,
- digitális aláírás létrehozása, speciális protokolloknak megfelelő formattálás és a fájl tartalom titkosítása,
- a fájl tartalom dekódolása és a digitális aláírás ellenőrzése.

Digitális aláírás létrehozására és ellenőrzésére egy aszimmetrikus kulcsrendszert (aszimmetrikus titkosító algoritmust) használ az SDK.

Titkosításra és dekódolásra egy szimmetrikus titkosító algoritmust használ az SDK.

1.2 Biztonsági funkciók

Az SDK az alábbi biztonsági funkciókat valósítja meg:

Digitális aláírás létrehozás (SF1)

A digitális aláírás a fájl bináris tartalmára készül.
Az aláírások az adatokra számolt lenyomat (hash) értékre vonatkoznak, s maga az aláírás az aláíró (intelligens kártyán vagy helyi adatbázisban tárolt) magánkulcsával történő aszimmetrikus titkosítás.
A dokumentum aláírásán kívül még a hitelesítés-szolgáltató által kibocsátott tanúsítványt is a dokumentumhoz csatolódik, mely bizonyítja az aláíró azonosságát. Ez a tanúsítvány tartalmazza az aláíró nyilvános kulcsát is. Az aláíró hitelessége azzal bizonyított, hogy jogosult felhasználója az intelligens kártyának vagy ismeri az adatbázisban tárolt magánkulcshoz való hozzáférés jelszavát.
A felhasználó tájékoztatást kap az adatok helyes aláírásáról, s ezt az információt meg is kell erősítenie.

/A digitális aláírásra az SHA-1, MD-5 és RIPEMD-160 lenyomatoló (hash) függvények, valamint az RSA aszimmetrikus titkosító algoritmus (legalább 768 bites kulcsméret mellett) használhatók./

Digitális aláírás ellenőrzés (SF2)

Az SF1 által létrehozott aláírást is tartalmazó dokumentum ellenőrzése egyrészt a lenyomat érték dekódolását jelenti a fogadott aláírásból, a küldő nyilvános kulcsának felhasználásával, másrészt a fogadott dokumentumra új lenyomat érték számítását. A két összetartozó lenyomat érték összehasonlítása dönti el, hogy a dokumentum valóban hiteles és változatlan-e (a két érték azonos-e).
A felhasználó tájékoztatást kap a fogadott dokumentum helyes ellenőrzéséről, s ezt az információt meg is kell erősítenie.
Egy dokumentum aláírójának azonosítója és nyilvános kulcsa a dokumentumhoz csatolt tanúsítványból nyerhető ki, mely a hitelesítés-szolgáltató (intelligens kártyán vagy helyi adatbázisban tárolt) nyilvános kulcsával ellenőrizhető.

Szimmetrikus adat titkosítás/dekódolás (SF3)

Az aláírás után a dokumentum adatai és az aláírás titkosításra kerül egy szimmetrikus titkosító algoritmussal.
Az adat dekódolására még a fogadó általi aláírás ellenőrzés előtt kerül sor.
A titkosításra felhasznált kulcs véletlenekből előállított munkaszakasz kulcs, melyet a dokumentum részeként a fogadóhoz továbbítanak, a fogadó nyilvános kulcsával titkosítva. Egyedül a fogadó képes dekódolni ezt a munkaszakasz kulcsot saját magánkulcsával, majd ennek segítségével a dokumentumot és annak aláírását. Több fogadó (címzett) is lehetséges, ilyenkor több titkosított nyilvános kulcs mezőt (benne a munkaszakasz kulcs különböző titkosított képével) kell a dokumentumhoz csatolni.

/Adat titkosításra a DES, Triple-DES és IDEA szimmetrikus titkosító algoritmusok választhatók./

2. A SafeGuard Sign&Crypt SDK v2.0 biztonsági előirányzata

Az alábbiakban áttekintjük a SafeGuard Sign&Crypt SDK v2.0 biztonsági előirányzatának /Security Target/ lényegi részeit, melyeknek való megfelelőséget vizsgálta és igazolta az ITSEC szerinti értékelés és tanúsítás.

2.1 Az értékelés tárgyának meghatározása

Az értékelés tárgya a következő termék:

SafeGuard Sign&Crypt Software Development Kit version 2.0

Az értékelés tárgyát a következő termék komponensek alkotják:

- SafeGuard Sign&Crypt version 2.0
- CryptWare Client Server API (CCS API) támogató szoftver
- SafeGuard Sign&Crypt felhasználói útmutató (írásos dokumentum)
- SafeGuard Sign&Crypt SDK műszaki kézikönyv (írásos dokumentum)
- SafeGuard Sign&Crypt SDK programozói útmutató biztonságos alkalmazásokhoz (írásos dokumentum)

Az értékelés tárgya a következő operációs rendszer platformokat támogatja:

- Microsoft Windows 95
- Microsoft Windows NT 4.0

Az operációs rendszer platformok támogatása az értékelés tárgya telepítésekor valósul meg, amikor is a különböző operációs rendszerekben különböző komponensek telepítésére kerül sor.

2.2 Az értékelés tárgyának leírása és tervezett felhasználási módja

2.2.1 Áttekintés

A SafeGuard Sign&Crypt SDK egy fejlesztő készlet, melynek segítségével különböző felhasználói igényeket kielégítő alkalmazói programok fejleszthetők, s építhetők be egy elektronikus üzenetkezelő rendszerbe.
A SafeGuard Sign&Crypt SDK a különböző feladatokhoz biztosít funkciókat:

- A kliens alkalmazás és a SafeGuard Sign&Crypt CryptWare Client Server API közötti kommunikáció elindítása és leállítása,
- Intelligens kártyáról vagy titkosított lokális fájlból származó magánkulcsok használata,
- A dokumentum fogadók számára biztosított adatbázisokban tárolt nyilvános kulcsok használata,
- Digitális aláírás létrehozása, speciális protokolloknak megfelelő formattálás és a fájl tartalom titkosítása,
- A fájl tartalom dekódolása és a digitális aláírás ellenőrzése.

Digitális aláírás létrehozására és ellenőrzésére egy aszimmetrikus kulcsrendszert (aszimmetrikus titkosító algoritmust) használ az SDK.

Titkosításra és dekódolásra egy szimmetrikus titkosító algoritmust használ az SDK.

A funkciócsoportok részletesebb leírását a következő három alfejezet tartalmazza.

2.2.2 Életciklus kezelés

Ez a funkciócsoport a kliens alkalmazás, valamint a SafeGuard Sign&Crypt CryptWare Client Server API közötti kommunikáció elindítását, illetve leállítását végzi.

Ebbe a funkciócsoportba a következő funkciók tartoznak:

- Az API-t használó kliens alkalmazás és a SafeGuard Sign&Crypt CryptWare Client Server API (CCS API) közötti kommunikáció elindítása,
- A kliens és a CCS API közötti kommunikáció leállítása.

Ezek a funkciók nem valósítanak meg semmilyen biztonsági mechanizmust, kizárólag kezelési célokat szolgálnak. Az SDK használata előtt, illetve után kell meghívni őket.

2.2.3 Munkaszakaszok (session) kezelése

A munkaszakasz kezelés funkciói kezelik az aszimmetrikus kulcsrendszer kulcsait. Ebben a kulcsrendszerben minden felhasználónak van egy kulcspárja: egy kizárólag a jogosult felhasználó által elérhető magánkulcs, valamint az ehhez tartozó, bárki által elérhető nyilvános kulcs.

Ebbe a funkciócsoportba a következő funkciók tartoznak:

- Egy felhasználó magánkulcsát aktivizáló funkciók (egyszerre csak egy magánkulcs aktivizálható). Az ide tartozó funkciók egy fejrészt állítanak elő, melyek leírják a magánkulcs tulajdonságait és tárolási helyét.
  A magánkulcsok különböző helyen tárolhatók:
  - Lemezen tárolt magánkulcsok: ebben az esetben valamennyi kulcs egy titkosított adatfájlban, a merevlemezen tárolódik. Az adatfájl egy jelszóval nyitható meg, melyből a dekódoláshoz szükséges magánkulcs előállítható.
  - Intelligens kártyán tárolt magánkulcsok: ebben az esetben a magánkulcs egy intelligens kártyán tárolódik. Ilyenkor egy CardMan típusú intelligens kártyaolvasó berendezést kell a rendszerhez kapcsolni. A magánkulcs soha nem hagyja el az intelligens kártyát, az értékelés tárgya valamennyi magánkulcsot igénylő műveletet az intelligens kártyán hajt(at)ja végre.

- A fogadó nyilvános kulcsait aktivizáló funkciók: egy vagy több nyilvános kulcs aktivizálható, mindegyiket egyértelműen azonosít egy felhasználói azonosító. Az ide tartozó funkciók egy fejrészt állítanak elő, melyek leírják a nyilvános kulcsok tulajdonságait és tárolási helyüket. A nyilvános kulcsok tanúsítványokba foglalva, egy speciális belső szerkezetű adatbázisban tárolódnak. Ezen adatbázis különböző forrásból való feltölthetőségét egy külön Utimaco hitelesítés-szolgáltató szoftver biztosítja.

- A fenti funkciók által előállított kulcs fejrészek deaktivizálása.

Mindkét kulcs aktivizálást végző funkció (alkalmazói program általi) meghívása esetén az SDK funkció kijelzi a felhasználó számára, hogy válassza ki a kívánt kulcsot, illetve helyezze be az olvasóba a megfelelő intelligens kártyát és PIN kódjával aktivizálja a magánkulcsát.

2.2.4 Fájlok kezelését végző biztonsági függvények

2.2.4.1 Áttekintés

A fájlokra vonatkozó legtöbb biztonsági művelet négy API függvény (funkció) kombinációjaként áll elő. Ebből kettő „fájl becsomagolás" jellegű, és a következő műveleteket tartalmazza:

- digitális aláírás létrehozás,
- adat titkosítás (esetleg adat tömörítéssel kombinálva),
- protokoll-specifikus formátum generálás.

A másik kettő az alábbi inverz műveleteket („fájl kicsomagolás") tartalmazza:

- protokoll adatok értelmezése,
- adat dekódolás (szükség esetén a dekompresszió elvégzése),
- digitális aláírás ellenőrzés.

Minden függvénycsoportban az egyik függvény egyetlen fájlra, a másik függvény pedig több fájlra hajtja végre (egymás után) a műveleteket.

A fájl „becsomagolást" végző függvényekre a meghívó programnak egy üzemmód paraméterben kell meghatároznia, hogy mely műveleteket (digitális aláírás létrehozás, tömörítés és/vagy titkosítás) kell végrehajtania a meghívott függvénynek. Csak a paraméter helyes beállítása biztosítja a függvényt meghívó alkalmazás biztonságos működését.

A fájl „kicsomagolást" végző függvények esetében az érintett fájlok tartalma meghatározza az igényelt műveleteket.

A telepített SafeGuard Sign&Crypt termék installációs paraméterei határozzák meg, hogy a későbbiekben milyen protokollt és algoritmusokat fog a termék használni protokoll-specifikus és kriptográfiai műveleteiben. A választható kriptográfiai algoritmusok a következők[1]:

- lenyomatoló (hash) függvény
  - SHA-1,
  - MD-5,
  - RIPEMD-160,
- aszimmetrikus titkosító algoritmus
  - RSA (legalább 768 bites kulcsméret mellett)
- szimmetrikus titkosító algoritmus
  - DES (CBC, 56 bites kulcsméret),
  - Triple-DES (CBC, 112 bites kulcsméret),
  - IDEA (CBC, 128 bites kulcsméret),
- adat tömörítés
  - LZSS
  - ZLIB.

[1] Csak azokat a kriptográfiai algoritmusokat soroljuk fel, melyeket érintett az ITSEC értékelés.

Az adat dekódolást és digitális aláírást végző függvények meghívása előtt aktivizálni kell a szükséges magánkulcsot, és fejrészét paraméterként át kell adni a függvénynek. A titkosítás művelet kiválasztása esetén a titkosítandó fájl címzettjeinek nyilvános kulcsaira vonatkozó érvényes (aktivizált) kulcs fejrészt is át kell adni a függvénynek.

Alapvetően a digitális aláírás létrehozása és ellenőrzése, valamint az adatok titkosítása és dekódolása függvények alkotják az értékelés tárgyának biztonságot érvényre juttató funkcióit.

A fájlok kezelést végző biztonsági függvények által végrehajtott fontosabb műveleteket a következő alfejezetek részletezik.

2.2.4.2 Digitális aláírás létrehozás

Egy digitális aláírás létrehozására az értékelés tárgyának ez a függvénye először egy lenyomat (hash) értéket számol az aláírandó dokumentum bináris tartalmára. Ezután a lenyomat értéket titkosítják az aláíró magánkulcsának és egy aszimmetrikus titkosító algoritmusnak a segítségével. Amennyiben a magánkulcsot egy intelligens kártyán tárolják, maga a titkosítás (aláírás) is a kártyán hajtódik végre.

A titkosított lenyomat értéket nevezik digitális aláírásnak, melyet a függvény az aláírt dokumentumhoz csatol. Kiegészítésül egy tanúsítványt is a dokumentumhoz csatol, mely azonosítja a dokumentum aláíróját, tartalmazza annak nyilvános kulcsát, s egy hitelesítés-szolgáltató által digitálisan alá is van írva.

Az értékelés tárgya ezután kijelzi a felhasználónak a dokumentum aláírási folyamatának sikeres befejezését, akinek meg kell erősítenie az üzenet aláírási szándékát a folytatáshoz.

Amennyiben a függvényt a tömörítési opcióval hívták meg, s a választott protokoll támogatja a tömörítést, a dokumentum tartalmának tömörítésére az aláírási folyamatot követően kerül sor.

2.2.4.3 Adat titkosítás

A titkosítás kiválasztása esetén a dokumentumot (az eredeti vagy a tömörített adatokat, beleértve a digitális aláírást is) egy szimmetrikus titkosítási algoritmus, s egy véletlenül generált munkaszakasz kulcs (session key) segítségével titkosítják. Magát a kulcsot is titkosítják, a fogadó(k) nyilvános kulcsa(i) felhasználásával, majd a titkosított kulcso(ka)t és a fogadó(k) azonosító(já/i)t hozzácsatolják a titkosított dokumentumhoz. Ehhez a lépéshez a „fájl becsomagoló” függvényt a fogadó(ka)t azonosító nyilvános kulcsokat tartalmazó kulcs fejrésszel kell meghívni.

Az aláírás létrehozás és/vagy a fent leírt titkosítás mechanizmusainak feldolgozása után a függvény az alkalmazásnak visszatérési értékként megadja a folyamat (sikeres vagy hibás) eredményét.

Sikeres visszatérési érték esetén a dokumentum elküldésre kész. A címzett(ek)nek való elküldés az értékelés tárgya hatókörén kívül esik.

2.2.4.4 Adat dekódolás

Amikor egy fogadott dokumentumot valamelyik „fájl kicsomagoló” függvény dolgoz fel, ellenőrzésre kerül, hogy titkosított-e. Amennyiben igen, az első lépés a dekódolás elvégzése. Ehhez először a munkaszakasz kulcsot kell dekódolni, a fejrész aktuális fogadójának[2] megfelelő részéből. Ezt a fogadó aktivizált magánkulcsával lehet megtenni, melynek érvényes (aktivizált) fejrészét paraméterként át kell adni a „kicsomagoló” függvénynek. A munkaszakasz kulcs dekódolását az aszimmetrikus titkosító algoritmus végzi. Ezt követően a dekódolt munkaszakasz kulcs a dokumentum adat részének (köztük a digitális aláírás(ok) is) szimmetrikus titkosító algoritmussal való dekódolására szolgál.

Amennyiben a dokumentum tömörített, a dekódolás után végrehajtódik a dekompresszió művelete.

2.2.4.5 Digitális aláírás ellenőrzés

A dekódolás után (amennyiben a dokumentum titkosított) a „kicsomagoló” függvényben a digitális aláírás ellenőrzése következik.

Először az aláíró tanúsítványát nyerik ki a dokumentumból, s ezt ellenőrzik a hitelesítés-szolgáltató nyilvános kulcsa segítségével. Ezt követően a tanúsítványból kiveszik a (hitelességében már ellenőrzött) nyilvános kulcsot. Ennek felhasználásával (az aszimmetrikus titkosítás dekódolásával) előállítják a dokumentumra számolt lenyomat (hash) értéket. Ugyanakkor a dokumentum tartalmára megismétlik a lenyomatoló eljárást, ugyanazon lenyomatoló függvény alkalmazásával. A dekódolt és a helyben kiszámolt lenyomat értékek összehasonlítása adja az ellenőrzés eredményét: megegyezés esetén az aláírás ellenőrzés sikeres, ellenkező esetben sikertelen.

Az ellenőrzés eredménye a felhasználó számára kijelzésre kerül, akinek megerősítése szükséges a folyamat folytatásához.

Az adat dekódolás és/vagy aláírás ellenőrzés fent leírt mechanizmusainak feldolgozása után a függvény az alkalmazásnak visszatérési értékként megadja a folyamat (sikeres vagy hibás) eredményét.

[2] Egy dokumentumot lehet egyszerre több fogadónak is címezni. Ilyenkor mindegyik fogadó nyilvános kulcsával titkosítva van a (közös) munkaszakasz kulcs, s a fejrész valamennyi titkosított változatot tartalmazza.

2.2.5 A termék telepítése és használata

A SafeGuard Sign&Crypt SDK telepítésére először abban a fejlesztő környezetben van szükség, ahol az API függvényeket meghívó alkalmazásokat fejlesztik.

A SafeGuard Sign&Crypt SDK használatához először a SafeGuard Sign&Crypt terméket kell telepíteni. Ezt egy hajlékony lemezről kell végrehajtani egy telepítő program segítségével.

A telepítő program többször megszakítja működését különböző választási lehetőséget felkínálva:

- a program elérhetőségi útvonala,
- a kulcs adatbázis elérhetőségi útvonala,
- a választott adattovábbítási protokoll,
- azok az alkalmazói programok, melyekbe a "SafeGuard Sign&Crypt"-et integrálni kell,
- intelligens kártyaolvasó.

Ezt követően a telepítés teljesen automatikus.

Ezt követően a SafeGuard Sign&Crypt CryptWare Client Server API támogató szoftver speciális tartalmát kell telepíteni egy külön hajlékony lemezről. Ez a rész az alábbiakat tartalmazza:

- header fájl (.H) és egy statikus, kapcsolható könyvtár (*.LIB) a C kódhoz,
- leíró fájl (.BAS) a VisualBasic kódhoz,
- leíró fájl (.PAS) a Pascal kódhoz,
- minta programok (C, BASIC, PASCAL) az API felhasználására.

A header fájlok, illetve a leíró fájlok beilleszthetők a különböző felhasználó-specifikus alkalmazói programok C, VisualBasic vagy Pascal forráskódjába.

Az SDK segítségével fejlesztett alkalmazói programok végrehajthatósága érdekében a célrendszerben szükség van a következőkre:

- SafeGuard Sign&Crypt Kernel és CryptWare Client Server API (a SafeGuard Sign&Crypt termék fő kompenensei),
- Az SDK segítségével fejlesztett alkalmazói program.

2.3 Az elvárt környezet

2.3.1 Hardver követelmények

Az értékelés tárgya, valamint a segítségével fejlesztett alkalmazói programok standard, (az Intel Pentium mikroprecesszorral kompatibilis) személyi számítógépeken futtathatók.
Nincs egyéb speciális hardver feltétel, mint például rögzített lemez egység vagy egyebek (a telepítéshez szükséges szabad helyen kívül).
Intelligens kártya használata esetén a személyi számítógépnek rendelkeznie kell egy szabad soros port-tal a CardMan/CardMan Compact kártyaolvasóval való összekapcsoláshoz. A CardMan billentyűzet szintén egy szabad soros port-ot igényel a kártyaolvasóval való összekapcsoláshoz. CardMan Mobile használata esetén nincs szükség rögzített szabad soros port-ra, de a négy elérhető soros port egyikének ilyenkor is szabadnak kell lennie.

Az értékelés tárgya biztonságos működéséhez az alábbi típusú intelligens kártyákat várja el:

- SLE CR80S T-COS operációs rendszerrel és a kártyán 768 bites RSA műveletekkel,
- SLE 44CR80S CardOS operációs rendszerrel és a kártyán 1024 bites RSA műveletekkel.

(Az értékelés tárgya más intelligens kártya típusokat is támogat, de a tanúsított működés a listában szereplő két kártyára korlátozódik.)

2.3.2 Szoftver követelmények

A SafeGuard Sign&Crypt SDK a következő operációs rendszereken elérhető:

- Windows 95,
- Windows NT 4.0 munkaállomás és szerver.

A SafeGuard Sign&Crypt SDK a következő platformokat támogatja alkalmazói programok fejlesztéséhez:

- valamennyi ANSI kompatibilis C fordító, a támogatott operációs rendszereken,
- Microsoft VisualBasic,
- Borland (Inprise) DELPHI Pascal fejlesztő környezet.

Ezen kívül az API függvényei tetszőleges, 32-bites Windows DLL függvények meghívására alkalmas alkalmazói programból is meghívhatók.

2.3.3 Biztonságos konfigurálás

Biztonságos alkalmazások fejlesztése érdekében az API függvényeit a „SafeGuard Sign&Crypt SDK programozói útmutató biztonságos alkalmazásokhoz” dokumentum utasításainak megfelelően kell használni. Ez garantálja, hogy a függvények végrehajtása során az egyes biztonsági mechanizmusok nem deaktiválják, nem kerülik meg, illetve nem hiúsítják meg egymást. Az említett utasításokat jelen Tanúsítási jelentés 3. számú érvényességi feltétele (4. fejezet) részletesen felsorolja.

2.4 Alanyok (szubjektumok), objektumok és műveletek

Az alábbiakban a dokumentum egy olyan fájlt jelent, melyet egy küldő egy vagy több fogadónak (címzettnek) küld.
Az aláírás pedig egy olyan adathalmaz, melyet a küldő a dokumentumhoz csatol, s mely a fogadó(k) által ellenőrizhető.

2.4.1 Alanyok

S1: A dokumentum küldője.

S2: A dokumentum jogosult fogadója (a küldő által szándékolt fogadók egyike).

S3: Jogosulatlan személy (akár egy dokumentum tartalmát próbálja megismerni, akár az aláírt dokumentum tartalmát próbálja módosítani).

S4: Olyan (a küldőtől eltérő) személy, melynek azonosságát a dokumentum küldője állítja magáról (akár szándékosan, akár nem szándékosan).

2.4.2 Objektumok

O1: Egy dokumentum tartalma, azaz bináris tartalma (a hozzácsatolt tanúsítványok és aláírások nélkül).

2.4.3 Műveletek

A1: Egy dokumentum aláírása.

A2: Egy dokumentum aláírásának az ellenőrzése.

A3: Egy dokumentum tartalmának módosítása.

A4: Egy dokumentum tartalmának (részleges vagy teljes) megismerése.

2.5 Biztonsági cél és feltételezett fenyegetések

2.5.1 Biztonsági cél

Az értékelés tárgyát arra tervezték, hogy segítségével biztonságos elektronikus üzenetkezelő rendszert megvalósító alkalmazói programokat fejleszthessenek, bennük digitális aláírások létrehozásával / ellenőrzésével, illetve titkosítással / dekódolással.

Ezek az alkalmazások a küldő által aláírt dokumentumok letagadhatatlanságát biztosítják. Egyaránt bizonyítható a küldő hitelessége és a dokumentum sértetlensége, az értékelés tárgyának aláírási függvényeinek felhasználásával. Az ehhez szükséges valamennyi kriptográfiai mechanizmust és kulcs kiválasztási műveletet biztosítják az értékelés tárgya függvényei.

Mindez annak feltételezésével állítható, hogy a felhasználót nyilvános és magánkulcsokkal, valamint tanúsítványokkal ellátó hitelesítés-szolgáltató biztonságosan működik.

A fenti feltétel teljesülése esetén a dokumentum küldője biztos lehet abban, hogy egyedül ő képes saját aláírásával ellátni a dokumentumokat, s abban is, hogy csak a pontosan általa aláírt dokumentum ellenőrzése lesz sikeres a fogadónál.

A fenti feltétel teljesülése esetén a dokumentum fogadója is biztos lehet abban, hogy a dokumentum küldője pontosan azt a tartalmat írta alá, melyet az általa végzett ellenőrzés helyesnek értékelt.

Az értékelés tárgya ezeken kívül függvényeket biztosít a dokumentumok bizalmasságának megőrzésére az aláírás és annak ellenőrzése közötti időszakra.

A (hitelesítés-szolgáltatóra és a biztonságos alkalmazás fejlesztésére vonatkozó) feltételek figyelembe vételével, az értékelés tárgya segítségével fejlesztett alkalmazások képesek lesznek az alábbiakban meghatározott fenyegetések kivédésére.

2.5.2 Feltételezett fenyegetések

T1: Az adatok sértetlensége elleni támadás

Az S1 küldő által aláírt (A1) O1 dokumentumot manipulálja (A3) egy S3 jogosulatlan személy, s ennek ellenére az S2 fogadó mégis helyesnek fogadja el azt ellenőrzése (A2) alapján.

T2: A küldő hitelessége elleni támadás

Az O1 dokumentum valódi S1 küldője helyett egy S4 küldőt (aláírót) jelez ki az S2 számára az aláírás ellenőrzése (A2). Ez akkor következhet be, ha a dokumentumot kibocsátó S1 (szándékosan, vagy véletlenül) azt állítja magáról, hogy ő S4.

T3: Az adatok bizalmassága elleni támadás

Az O1 dokumentum elolvasható (A4) egy S3 jogosulatlan személy által, az adatok S1 és S2 alanyok közötti továbbítása során.

2.6 A biztonságot érvényre juttató funkciók és mechanizmusok

A biztonságot az alábbi biztonsági funkciók juttatják érvényre:

- Digitális aláírás létrehozás (SF1)
- Digitális aláírás ellenőrzés (SF2)
- Szimmetrikus adat titkosítás/dekódolás (SF3)

A fenti biztonsági funkciók részletes leírását az 1.2 alfejezet tartalmazza.

Az alábbi táblázat áttekintést ad arról, hogy mely biztonsági funkció, mely fenyegetés kivédéséhez járul hozzá.
Ahol több biztonsági funkció jut egy fenyegetésre, ott e funkciók együttesen lépnek fel a fenyegetéssel szemben.

	T1	T2	T3
SF1	+	+	
SF2	+	+	
SF3			+

A biztonsági funkciók az alábbi biztonsági mechanizmusokat valósítják meg:

- SHA-1 lenyomatoló függvény /SF1 és SF2/
- MD5 lenyomatoló függvény /SF1 és SF2/
- RIPEMD-160 lenyomatoló függvény /SF1 és SF2/
- RSA aszimmetrikus titkosító algoritmus (768 bites kulccsal) /SF1 és SF2/
- RSA aszimmetrikus titkosító algoritmus (1024 bites kulccsal) /SF1 és SF2/
- DES szimmetrikus titkosító algoritmus (CBC, 56 bites kulcsméret) /SF3/
- Triple-DES szimmetrikus titkosító algoritmus (CBC, 112 bites kulccsal) /SF3/
- IDEA szimmetrikus titkosító algoritmus (CBC, 128 bites kulccsal) /SF3/
- LZSS adat tömörítő algoritmus /SF3/
- ZLIB adat tömörítő algoritmus /SF3/

A mechanizmusok minimális erőssége (a 2.3.3 alatt meghatározott helyes konfigurálás esetén): **közepes**[3]

[3] Egy biztonsági mechanizmus minimális erőssége:
alap szintű, ha bizonyíthatóan védelmet nyújt a biztonság véletlen megsértése ellen, de kellő ismeretekkel rendelkező támadók hatálytalaníthatják,
közepes szintű, ha bizonyíthatóan védelmet nyújt korlátozott erőforrással és lehetőséggel rendelkező támadók ellen,
magas szintű, ha bizonyíthatóan csak olyan támadó képes hatálytalanítani, aki magas szintű szaktudással, erőforrással és lehetőséggel rendelkezik.
A mechanizmusok minimális erősségére vonatkozó „közepes szintű” állítás azt jelenti, hogy minden mechanizmusra ez a minimális erősség legalább közepes szintű (azaz közepes vagy magas szintű).

3. Az ITSEC értékelés eredményeinek összefoglalása

Az értékelés sikerrel zárult. Az értékelés megerősítette, hogy a biztonságot érvényre juttató funkciók a biztonsági előirányzatnak megfelelően működnek.

Biztonsági funkcionalitás: **digitális aláírás létrehozás,**
digitális aláírás ellenőrzés,
szimmetrikus adat titkosítás/dekódolás

Garanciaszint: **E2**

A mechanizmusok erőssége: **közepes szintű**

4. A Tanúsítási jelentés eredménye, érvényességi feltételei

Az alábbiakban összefoglaljuk azokat a feltételeket, amelyek betartása hozzájárul a SafeGuard Sign&Crypt SDK v2.0 által kezelt aláírások fokozott biztonságához.
A feltételek között vannak kötelezően betartandó, a tanúsítvány érvényességére kiható feltételek, és vannak olyan feltételek, amelyek az aláírások biztonságára jelentős befolyással bírnak, ezért fokozott (nem minősített) aláíráshoz, ahol a lehetőség adott, ezen feltételek betartása erősen ajánlott.

4.1 Kötelezően betartandó feltételek

1. Az elvárt hardver környezet biztosítása

Amennyiben a SafeGuard Sign&Crypt SDK-hoz (és a segítségével fejlesztett alkalmazói programokhoz) intelligens kártyát használnak, akkor:
az alábbi típusú kártyaolvasót kell alkalmazni:

- CardMan vagy
- CardMan Compact, illetve

az alábbi típusú intelligens kártyákat kell alkalmazni:

- SLE CR80S (T-COS operációs rendszerrel, a kártyán 768 bites RSA műveletekkel),
- SLE 44CR80S (CardOS operációs rendszerrel, a kártyán 1024 bites RSA műveletekkel).

2. Az elvárt szoftver környezet biztosítása

A SafeGuard Sign&Crypt SDK a következő operációs rendszereken használható:

- Windows 95,
- Windows NT 4.0 munkaállomás és szerver.

3. Biztonságos konfigurálás

Biztonságos alkalmazások fejlesztése érdekében az API függvényeit a „SafeGuard Sign&Crypt SDK programozói útmutató biztonságos alkalmazásokhoz” dokumentum alábbi utasításainak megfelelően kell használni:

- Valamennyi SDK függvény hívás esetén (a CCSClose kivételével) a visszatérési kódot ellenőrizni kell,
- Abban az esetben, ha egy SDK függvény visszatérési kódja problémát jelez, a felhasználót tájékoztatni kell a problémáról, az alkalmazás kapcsolatát a SafeGuard Sign&Crypt Kernel felé le kell zárni, s egy üresjárat állapotba kell visszatérni.
- Egy fájl kezelő biztonsági függvénynek paraméterként átadandó magánkulcsot (pontosabban az ezt meghatározó adatokat) az SDK-tól közvetlenül a meghívás előtt kell lekérni, s érvényességét ellenőrizni kell. Érvénytelenség esetén a függvényt nem szabad meghívni.
- Egy fájl kezelő biztonsági függvénynek paraméterként átadandó nyilvános kulcsot (pontosabban az ezt meghatározó adatokat) az SDK-tól közvetlenül a meghívás előtt kell lekérni, s érvényességét ellenőrizni kell. Érvénytelenség esetén a függvényt nem szabad meghívni.

- Különösen a digitális aláírás létrehozás / ellenőrzés függvények üzemmód paraméterét[4] kell helyesen beállítani ahhoz, hogy a dokumentumokra biztonságos digitális aláírás létrehozás és ellenőrzés valósuljon meg. (A paraméternek tartalmaznia kell az „S”, illetve „E” karaktereket, a dokumentum digitális aláírása, illetve titkosítása érdekében.)

4 Az ITSEC által értékelt algoritmusok használata

Csak az alábbi, az ITSEC értékelés során megvizsgált kriptográfiai algoritmusokat hívják meg[5]:

- lenyomatoló (hash) függvény
  - SHA-1,
  - MD-5,
  - RIPEMD-160,
- aszimmetrikus titkosító algoritmus
  - RSA (legalább 768 bites kulcsméret mellett)
- szimmetrikus titkosító algoritmus
  - DES (CBC, 56 bites kulcsméret),
  - Triple-DES (CBC, 112 bites kulcsméret),
  - IDEA (CBC, 128 bites kulcsméret),

4.2 Ajánlások az aláírás alkalmazások fejlesztéséhez

A SafeGuard Sign&Crypt SDK /”aláíró alkalmazást fejlesztő munkaállomásokon”/ alapvetően elszigetelt működtetési környezetben használandó, de kiegészítő feltételek garantálása esetén védett működtetési környezetben is lehet fejleszteni vele (sőt a teljes funkcionalitás végső tesztelése csak ilyen körülmények között valósítható meg).

Elszigetelt működtetési környezet (kisebb fejlesztéseknél ez a tipikus eset) esetén a fejlesztői készletet (és egyúttal a fejlesztés alatt álló aláíró alkalmazást) az védi, hogy nincs (sohasem) kapcsolódás kommunikációs hálózatokra (Internet, Intranet), és a működtetési környezetben olyan védelmi intézkedéseket valósítanak meg, melyek kivédik a jogosulatlan manuális hozzáférésen és adathordozóról történő adatbevitelen alapuló támadásokat is/. A SafeGuard Sign&Crypt SDK-val alapvetően ilyen környezetben ajánlott fejleszteni, tesztelni. Ugyanakkor az ebből fejlesztett, teljes funkcionalitást biztosító végterméket nem lehet teljeskörűen ebben a működtetési környezetben tesztelni, mivel az SDK bázisán kifejlesztett aláíró alkalmazás:

- aláírás létrehozásánál nem képes tesztelni bizonyos funkcióit (pl. időbélyegzésnél nem kerülhet hálózati kapcsolatba egyetlen időbélyeg-szolgáltatóval sem),
- aláírás ellenőrzéséhez nem képes tesztelni az érvényes CRL-t letöltő funkcióját.

Védett működtetési környezet (kisebb fejlesztések esetén nem ez a tipikus eset) esetén a fejlesztői készletet (és egyúttal a fejlesztés alatt álló aláíró alkalmazást) a működtetési környezet nagy bizonyossággal megvédi a kommunikációs hálózatok (Internet, Intranet) irányából érkező, valamint a jogosulatlan manuális hozzáférésen és az adathordozóról történő adatbevitelen alapuló támadásoktól.

[4] (lásd 2.2.4.1)
[5] Ezeken kívül az SDK támogat számos egyéb kriptográfiai algoritmust is.

Általános (mindkét működtetési környezetre vonatkozó) feltételek

5 Eljárásrendi/szervezeti védelmi intézkedések

Eljárásrendi/szervezeti védelmi intézkedésekkel kell támogatni az aláíró alkalmazások fejlesztését megvalósító számítógép(ek)re irányuló olyan támadások kivédését, melyek manuális hozzáférésen, illetve adathordozóról történő adatbevitelen alapulnak. Garantálni kell, hogy a fejlesztés technikai környezete, valamint a fejlesztett programok és az ehhez felhasznált fejlesztő készlet funkcióit ne lehessen manipulálni, melyet különösen vírus és Trójai faló bejuttatása okozhat. Minden újonnan telepített szoftvernek manipulációtól mentesnek kell lennie.
/A fenti intézkedések döntően ahhoz kellenek, hogy a fejlesztendő aláíró alkalmazás és az ennek bázisát képező fejlesztő készlet ne manipulálódjon./

6. konfigurációmenedzselési eljárások

Az aláíró alkalmazásokat fejlesztő környezetben konfigurációmenedzselési eljárások kidolgozásával és betartásával kell garantálni a fejlesztett termékek sértetlenségét azzal, hogy fegyelmet és ellenőrzést követelnek meg a fejlesztendő termék és más ezzel összefüggő információ pontosításában és módosításában.
/ A fenti intézkedések akadályozzák meg a fejlesztés alatt álló alkalmazások egyes verzióinak jogosulatlan módosítását, bővítését vagy törlését, illetve járulnak hozzá a mégis bekövetkező felhatalmazás nélküli változtatások észleléséhez (a verziónként elkülönítetten is letárolt példányok időszakos összehasonlításával)./

A védett működtetési környezetben történő felhasználás járulékos feltételei

7. Mester példányok alkalmazása

Amennyiben a fejlesztő környezetnek hálózati kapcsolatai is vannak a 6. feltételben elvárt konfigurációmenedzselési eljárásokon kívül rendszeres időnként ellenőrizni kell a fejlesztő készlet és a fejlesztett aláíró alkalmazás verziók sértetlenségét, az elkülönítetten is letárolt mester példányok időszakos összehasonlításával.

Az elszigetelt működtetési környezetben történő felhasználás járulékos feltételei

Az elszigetelés számos fenyegetést eleve kizár (hálózati támadások), a fenyegetések más részét pedig az általános működtetési feltételek lefedik.
Nincs járulékos feltétel.

5. A követelményeknek való megfelelést ellenőrző független vizsgálat garancia szintje

A jelen Tanúsítási jelentéshez figyelembe vett, a fejlesztőktől független ellenőrző vizsgálat garancia szintje az ITSEC **E2**-es szintje volt. /Ez a CC EAL3 garanciaszintjének megfelelő, a fejlesztőktől függetlenül garantált biztonság közepesen erős szintjét biztosítja./

Ez az alábbi vizsgálatokat jelentette:

Az ellenőrző vizsgálat a SafeGuard Sign&Crypt SDK v2.0 biztonsági viselkedésének megértése érdekében elemezte a biztonsági funkciókat és biztonsági mechanizmusokat, melyhez felhasználta az alábbi, SafeGuard Sign&Crypt SDK v2.0-re vonatkozó fejlesztői dokumentációkat:

- biztonsági előirányzat,
- az architektúra informális leírása,
- a részletes tervek informális leírása,
- teszt dokumentáció,
- a teszteléshez használt tesztprogramok és tesztelő eszközök,
- felhasználásra vonatkozó útmutatók,
- az értékelés tárgyának verzióját azonosító konfigurációs lista,
- a konfigurációkezelésre használt rendszer leírása,
- a fejlesztői környezet biztonságára vonatkozó leírás,
- felhasználói útmutató,
- adminisztrátori útmutató[6].

Ezeken kívül:

- funkcionális teszteket végzett annak kimutatására, hogy az értékelés tárgya megfelel biztonsági előirányzatának,
- tesztelést végzett az eljárásokban esetlegesen meglévő hibák felderítésére,
- értékelte a biztonsági mechanizmusok erősségét.

[6] mely jelen esetben a műszaki kézikönyv és a programozói útmutató

6. A SafeGuard Sign&Crypt SDK v2.0 biztonsági funkcióinak értékelt erőssége

Még ha az értékelés tárgyának (jelen esetben a SafeGuard Sign&Crypt SDK v2.0 fejlesztő készletnek) a biztonsági funkcióit nem is lehet megkerülni, kiiktatni vagy tönkretenni, akkor is lehet lehetőség kijátszani azokat, ha a mögöttes biztonsági mechanizmusok sebezhetőek. E funkciók biztonsági viselkedése minősíthető a mechanizmusok biztonsági viselkedésének mennyiségi vagy statisztikai alapú elemzési eredményeinek felhasználásával és az ilyen mechanizmusok legyőzésére vonatkozó erőfeszítések segítségével.

A biztonsági funkciókat a biztonsági mechanizmusok valósítják meg. Például egy jelszókezelő mechanizmus az azonosítás és hitelesítés biztonsági funkciók megvalósításában használható fel.

A biztonsági funkciók erősségének elemzése a biztonsági mechanizmusok szintjén zajlott.

/A biztonsági mechanizmus erőssége értékelése keretében azt vizsgálták meg, hogy az értékelés tárgya által megvalósított biztonsági mechanizmusok mennyire képesek ellenállni egy képzett támadó közvetlen támadásának, illetve a támadónak milyen szintű erőforrásokra szaktudásra és lehetőségekre van szüksége a sikeres támadás megvalósításhoz. Itt annak mérésről van szó, hogy a mechanizmus mennyire képes a védelem megkerülése mellett a közvetlen támadásokat is megakadályozni.

A biztonsági mechanizmus erősségét alap, közepes és magas szintű minősítésekkel osztályozzák.

- *Alap szintű* a védelmi mechanizmus erőssége, ha bizonyíthatóan védelmet nyújt a biztonság véletlen megsértése ellen, de kellő ismeretekkel rendelkező támadók hatálytalaníthatják
- *Közepes szintű* a védelmi mechanizmus erőssége, ha bizonyíthatóan védelmet nyújt korlátozott erőforrással és lehetőséggel rendelkező támadók ellen
- *Magas szintű* a védelmi mechanizmus erőssége, ha a védelmet bizonyíthatóan csak olyan támadó képes hatálytalanítani, aki magas szintű szaktudással, erőforrással és lehetőséggel rendelkezik. Ilyen erősségű védelmi mechanizmus mellett a sikeres támadás valószínűsége rendkívül csekély./

A biztonsági mechanizmusok erősségének elemzése az alábbi eredményt adta:

A biztonsági funkciók erőssége: **közepes szintű**

7. A tanúsításhoz figyelembe vett dokumentumok

7.1 Termékmegfelelőségi követelményeket tartalmazó dokumentumok

Az elektronikus aláírásról szóló 2001. évi XXXV. törvény

CEN/ISSS/E-Sign; Area G1, 14170 munkacsoport egyezmény: Security Requirements for Signature Creation Systems

CEN/ISSS/E-Sign; Area G2, 14171 munkacsoport egyezmény: Procedures for Electronic Signature Verification

CEN/ISSS/E-Sign; Area V, 14172-4 munkacsoport egyezmény: Signature Creation Application and Procedures for Electronic Signature Verification

7.2 A tanúsítási jelentéshez figyelembe vett egyéb dokumentumok

7.2.1 A tanúsításhoz figyelembe vett egyéb fejlesztői dokumentumok

Kérelem a tanúsítás elvégzésére

Kérdőív a tanúsítás kérelmezéséhez

7.2.2 A tanúsításhoz figyelembe vett, fejlesztőktől független dokumentumok

Certification Report /SafeGuard Sign&Crypt Software Development Kit version 2.0, Utimaco Safeware AG/ debisZERT-DSZ-ITSEC-04008-1999

Certification Report /SafeGuard Sign&Crypt version 2.2, Utimaco Safeware AG/ debisZERT-DSZ-ITSEC-04007-1999

8. Rövidítések

A	Action
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher Block Chaining
CC	Common Criteria
CCS API	CryptWare Client Server API
DES	Data Encryption Standard /FIPS PUB 46-3, FIPS PUB 74, FIPS PUB 81/
DLL	Dynamic Link Library
EAL	Evaluation Assurance Level
IDEA	International Data Encryption Algorithm
ITSEC	Information Technology Security Evaluation Criteria
LZSS	/Compression Library/
MD5	Rivest: “The MD5 Message Digest Algorithm”
O	Object
PIN	Personal Identification Number
RSA	Rivest-Shamir-Adleman (public key cryptosystem)
S	Subject
SDK	Software Development Kit
SF	Security Function
SHA-1	Secure Hash Algorithm
T	Threat
Triple-DES	/FIPS PUB 46-3, ANSI X9.52/
ZLIB	/Compression Library/