Šifrovacie algoritmy radu SEA na použitie v šifrovej ochrane informácií

Autor: Milanp

Dátum vydania: 2. augusta 2025

Obsah

1. Špecifické pojmy použité v dokumentácii

OT – otvorený text
ŠT – šifrovaný text
OS – operačný systém
BŠ – Bezpečnostný štandard vydaný Národným Bezpečnostným Úradom SR pre vývoj ***PŠOI,*** ktorý podlieha stupňu utajenia „***D***“.
NBÚ – Národný Bezpečnostný Úrad Slovenskej republiky
AES – séria verejných šifrovacích algoritmov AES128, AES192, AES256 a AES512
SEA – séria súkromných utajovaných šifrovacích algoritmov SEA64, SEA512 a SEA1024
NSA – The National Security Agency – Národná bezpečnostná agentúra, obdoba nášho NBÚ
ÚŠO – Ústredný Šifrovací Orgán
ECB – Electronic Code Book (elektronická kódová kniha) – jadro šifrovacieho algoritmu
KT rozbor – kryptologicko technický rozbor algoritmu
KH – kľúčové hospodárstvo
PT ZD – stupeň utajenia „Prísne tajné zvláštnej dôležitosti“
PŠOI_OUS – prostriedky šifrovej ochrany informácií na ochranu utajovaných skutočností pre stupne utajenia „***V***“, „***D***“, „***T***“, „***PT***“
PŠOI_CUV – prostriedky šifrovej ochrany informácií na ochranu citlivých údajov pre širokú verejnosť
Šifra – šifrovací algoritmus všeobecne, prípadne zašifrovaný text - ***ŠT***
Runda – iterácia, jedno kolo substitučno-permutačných operácií pri spracovaní podkľúča
Renonc – prehrešok proti pravidlám, v kryptografii dosť závažný a ohrozujúci bezpečnosť
RFID – Radio Frequency Identification - rozhranie pre identifikačné karty
Režim – popis štandardu v režime fyzickej a objektovej bezpečnosti: Bezpečnostný štandard
MOSR – Ministerstvo obrany Slovenskej Republiky
Kryptoschéma – čisto hardvérová realizácia kryptografického algoritmu
Šumátor – fyzikálny generátor šumu na generovanie náhodných údajov
SAV – Slovenská akadémia vied
SEA64, SEA512, SEA1024 – rad silných a kvalitných neverejných šifrovacích algoritmov s dĺžkami kľúčov 256, 512 a 1024 bitov
AES128, AES192, AES256 – verejný šifrovací algoritmus s dĺžkami kľúčov 128, 192 a 256 bitov

2. Úvod do problematiky šifrovej ochrany informácií

V prvom rade je potrebné rozlišovať medzi potrebami pre realizáciu prostriedkov na šifrovú ochranu citlivých údajov používaných v širokej verejnosti (ďalej len ***PŠOI_CUV***) a potrebami pre realizáciu prostriedkov na šifrovú ochranu utajovaných skutočností pre stupne utajenia „V“, „D“, „T“, „PT“, (ďalej len ***PŠOI_OUS)***.

V prvom prípade pri citlivých informáciách neplatia také prísne pravidlá, ako v druhom prípade, kde sa jedná o utajované skutočnosti, tam sa pravidlá riadia podľa:

§ 70 ods. 1 písm. c) bod 5 zákona NR SR č. 215/2004 Z. z. o ochrane utajovaných skutočností a o zmene a doplnení niektorých zákonov v znení neskorších predpisov (ďalej len „zákon o ***OUS***“) a v súlade s § 2 ods. 19 v spojení s ods. 22 vyhlášky NBÚ č. 340/2004 Z. z., ktorou sa ustanovujú podrobnosti o šifrovej ochrane informácií (ďalej len „vyhláška o ***ŠOI***“).

To platí pre systémy spôsobilé chrániť utajované informácie Slovenskej republiky, ktoré môžu byť použité pre zabezpečenie ochrany a spracovávania utajovaných skutočností klasifikovaných podľa vnútroštátnych noriem stupňov utajenia v zmysle platnej legislatívy Slovenskej republiky.

3. Úvodné pojednanie o šifrovacích algoritmoch a ich nasadení v praxi

Existuje množstvo šifrovacích algoritmov, ktoré môžeme rozdeliť do dvoch skupín. V prvej sú verejné šifrovacie algoritmy a v druhej sú utajované šifrovacie algoritmy. O tej druhej skupine sú znalosti v laickej verejnosti, myslí sa tým verejnosť s nízkym povedomím v oblasti kryptológie, pomerne malé, alebo žiadne. Známe sú hlavne verejné algoritmy skupiny dnešnej doby, napríklad AES. (AES128, AES192, AES256, prípadne AES512) Sú to deriváty pôvodného algoritmu Rijndael, ktoré boli vytvorené odvodením a úpravami stanovenými americkou NSA a schválené ako bezpečnostný štandard Národným inštitútom pre štandardy a technológie (NIST). Nakoľko AES je dosť starý algoritmus, má už skoro 25 rokov a blíži sa doba postkvantová, zaoberáme sa zvyšovaním bezpečnostnej odolnosti šifrovacích algoritmov a vývojom nových riešení v oblasti kryptografie. V tejto oblasti máme skúsenosti a aj praktické výsledky z vyše 45 rokov výskumu a vývoja na poli bezpečnostných aplikácií, hlavne pre štátny sektor. Okrem iného, tiež z implementácií šifier do hardvéru aj softvéru. Pracovali sme nielen v oblasti utajenia citlivých údajov ale hlavne v oblasti utajenia pre štátny sektor na stupne utajenia „V-vyhradené“, „D-dôverné“ „T-tajné“, „PT-prísne tajné“ „PT ZD-prísne tajné zvláštnej dôležitosti“. „PTZD“ už po revolúcii neexistuje, bol to stupeň utajenia pre prezidentskú kanceláriu a vrchného veliteľa ozbrojených síl.

V dobe pred prijatím AES algoritmu ako štandardu, ktorý vyžadujú používať ako USA, tak aj EÚ, sme pri vývoji zariadení pre štátny sektor, používali algoritmus SEA64. To bolo obdobie od roku 1985 až do doby, keď NBÚ SR ako ***ÚŠO*** – Ústredný Šifrový Orgán, zapracoval do ním vydaného **BŠ** aj povinnosť používania algoritmu AES256 (v žiadnom prípade nie AES128 alebo AES192). Nie je ale vylúčené použitie aj SEA64 algorimu, ale technická dokumentácia predkladaná na NBÚ pri procese certifikácie musí v tom prípade obsahovať aj **KT rozbor** a bezpečnostný rozbor implementácie SEA64. Pre NATO a EÚ je nanútené použiť aj AES, pokiaľ sa použije aj iný algoritmus, napríklad SEA. Pri AESe je od predloženia týchto rozborov upustené, lebo je to verejný algoritmus ale platia tu zasa striktné obmedzenia podľa ***BŠ*** pri jeho implemenácii a použití na stupne utajenia "V" až "PT".

Riešením je použiť v aplikáciách viacero šifrovacích algoritmov a viac úrovňové **KH** a pritom AES256 ako hlavný algoritmus. Zrejme nanútením AESu, má americká NSA na to dôvod. Treba si to povedať na rovinu. Nikdy predtým, ako bolo použitie algoritmu AES nariadené v ***BŠ***, sa v realizácii ***PŠOI_OUS*** nepoužíval žiadny verejný algoritmus. Jedine utajovaný algoritmus. Na Slovensku a v Česku to bol algoritmus SEA64 a bol utajovaný, hlavne kvôli jeho S-boxom.

Veľká skupina popredných kryptológov za posledných 40 rokov potvrdila, že algoritmus SEA64 je silnejší ako AES256. Veď SEA64 sme používali až do stupňa utajenia PT ZD. Samozrejme s príslušnými opatreniami pri jeho implementácii a výhradne jeho realizáciu v hardvéri, na vyššie stupne utajenia ktorými sú "D", "T" a "PT". SAE64 bol vyvinutý skupinou kryptológov na bývalom FMV pod vedením kryptológa Lorenca. Bol použitý v hardvérových komponentoch v projekte rezortnej úlohy Fénix na šifrovanie prenosu hlasovej prevádzky (obdoba toho, čo je dnes v mobilných telefónoch ale bolo to na omnoho vyššej bezpečnostnej úrovni) pre prezidentskú kanceláriu a to na stupeň utajenia PT ZD. S tým rozdielom, že spojenie prebiehalo po klasických telefónnych linkách.

Vývoj šifrovacích zariadení sme realizovali vo Vojenskom výskumnom ústave VÚ060. Bol to vývoj realizovaný na úrovni asi o 10 rokov dopredu pred civilným výskumom a vývojom. Po prevrate v roku 1989 však VÚ060, vtedy už premenovaný na VTU – Vojenský technický Ústav bol čoskoro zlikvidovaný a veľa duševného vlastníctva sa doslova rozkradlo a to čo sa delilo po rozdelení Československa medzi Slovensko a Česko uniklo kdekoľvek (pomer delenia bol 2:1, Česko-2, Slovensko-1). Dostalo sa to do rúk aj Rusom v rámci predchádzajúcej spolupráce v rámci RVHP a Varšavskej zmluvy, ktorí z algoritmu SEA64 spravili verejný algoritmus GOST. Spravili to degradáciou pôvodných 1024 bajtových S-boxov (štyri 256 bajtové tabuľky) v SEA64 na 64 bajtov (128 štvorbitových položiek) v GOSTE, čiže urobili zmenu v uzloch zámeny a ostatné v algoritme ponechali. Museli to tak urobiť aby ich nikto neobvinil o odtajnení pôvodného tajného algoritmu. Veď vlastne S-boxy sú vždy jadrom celého šifrovaciehio algoritmu, určujúceho jeho vlastnosti. Pokiaľ nie je algoritmus verejný, tak aj S-boxy sú utajované a považované ako dlhodobo platný kryptografický prvok. Zmenou S-boxov sa zmenia vlastnosti algoritmu a je to vlastne potom už iná šifra.

My sme po 40-tich rokoch praxe a skúseností postupne vyvinuli aj zdokonalené algoritmy SEA512 a SEA1024. Tie nevznikli spodaricky, ale po získavaní skúseností v oblasti konštrukcií PŠOI zariadení, analýzach iných algoritmov a samozrejme aj pod vedením skúsených a renomovaných kryptológov. Postupne sme v posledných dvoch našich algoritmoch zvyšili dĺžku hlavného šifrovacieho kľúča a veľkosť spracovávaného bloku a

použili sme nové S-boxy, úmerne zväčšené a prispôsobené k požiadavkám týchto prvkov algoritmu.

4. Pojednanie o hlavných požiadavkách na nové algoritmy SEA

V prvom rade nejde ani tak o to, ako je daný algoritmus zrealizovaný, ale o jeho matematický popis a popis diagramami graficky, prípadne naprogramovaním.

Až v druhom rade je nastolená požiadavka vlastnej realizácie a spôsobu použitia. Realizovať konkrétny SEA algoritmus je možné prakticky x-torakými spôsobmi. A to napísaním v rôznych programovacích jazykoch pre rôzne operačné systémy, avšak pri dodržaní určitých prevádzkových podmienok. A to všetko podľa skutočného nasadenia a požiadaviek na použitie v daných aplikáciách alebo hardvérovom prostredí a jeho firmvéri. Napríklad implementáciou v programovateľných poliach alebo signálových procesoroch, s čím už máme skúsenosti a máme realizované aj výsledky našej práce, ktoré sú dodnes fuknkčné a používané.

A od toho, ako sa algoritmus naprogramuje, v akom OS aleho hardvéri, firmvéri bude fungovať, potom závisí aj rýchlosť jeho spracovania a celková bezpečnosť šifrovania. No a v neposlednom rade aj od spôsobu realizácie **KH** a typu zabezpečenia (viacúrovňové **KH**, bezpečnostné watchdogy a pod.).

4.1 Formy realizácie algoritmov

Konečná realizácia v podobe stvárnenia algoritmu binárnym kódom spadá pod ochranu autorským zákonom, ale matematický popis a popis diagramami, teda popísanie myšlienky realizácie, je už duševné vlastníctvo, ktoré sa autorským zákonom nedá chrániť, jedine v USA sa dá patentovať a tak získať ochranu.

Ako príklady možnej realizácie sú v technickej dokumentácii SEA algoritmov priložené aj zdrojové texty algoritmu v dvoch verziách, vzorových testov a benchmarkov. Avšak naprogramovanie šifier je možné ďalej zdokonaľovať, napríklad naprogramovať ich s využitím viacerých jadier procesora alebo vytvoriť programový modul pre použitie v jadre OS s IOPL ringu 0, kde je možné šifru prevádzkovať s real time prioritou jadra, prípadne použiť na vykonávanie kritických sekcií algoritmu funkcie "SpinLock", ktoré zastavia na krátky čas v danom toku spin multiplexu procesov a ich threadov v rámci preemptívneho multitaskingu OS, čo podstatne zrýchli spracovanie šifry.

My neponúkame definitívne konečné stvárnenie realizácie algoritmov, to závisí od každého používateľa, ako si ten ktorý algoritmus naprogramuje, dodávame len jeho matematický popis, diagram a zdrojové texty s projektami ako dôkaz funkčnosti a pre inšpiráciu pri vývoji a implementácii. Ale aj dopĺňame technickú dokumentáciu o dva spôsoby realizácie dvomi programovacími jazykmi. Tieto spôsoby je možné ďalej zdokonaľovať čo sa týka rýchlosti

spracovania algoritmu tiež pomocou ďalších optimalizácií, čo do objemu binárneho kódu a rozhrania pre komunikáciu aplikácie s algoritmom, v ktorej je implementovaý. Prípadne prejsť na iný programovací jazyk. Pri pochopení matematiky a použití rôznych typov realizácí musí byť výsledeok spracovania algoritmu rovnaký. Hoci ho aj napíšete v primitívnom jazyku BASIC, výsledná šifra - ***ŠT*** je zhodná napríklad s realizáciu v assembleri. Akurát rýchlosť spracovania v jazyku BASIC by bola žalostne nízka. Ale aj to niekedy môže postačovať, napríklad, keď som vyvíjal biochemický softvér.

5. O šifrovacích technológiách v praxi za posledných 45 rokov dodnes až po algoritmus SEA1024

Odolnosť metód šifrovania je prvoradá pre ochranu citlivých a utajovaných údajov pred čoraz sofistikovanejšími hrozbami, čo je požiadavkou budúcich algoritmov. Jedným zo šifrovacích algoritmov, ktorý by sa mohol považovať za budúci šifrovací algoritmus na šifrovanie utajovaných skutočností, je šifrovací algoritmus SEA1024 od spoločnosti Salutis systems. SEA1024 svojím inovatívnym návrhom predstavuje novú hranicu v kryptografickej bezpečnosti, ktorá je špeciálne navrhnutá tak, aby odolala výzvam, ktoré predstavuje kvantová výpočtová technika.

Táto štúdia neskôr ďalej popisuje šifrovací algoritmus SEA1024 so zameraním sa na kľúčové aspekty, ako je zložitosť zabezpečenia, odolnosť voči kryptoanalýze, odolnosť voči útokom hrubou silou a odolnosť voči kvantovým výpočtom. Totižto práve na S-boxoch daného algoritmu, okrem hlavného kľúča, stojí alebo padá celá kvalita výstupného ***ŠT*** a teda aj sila zašifrovania. To je kryptografickou axiomou. S-boxy musia byť generované špeciálne v závislosti na konštrukcii algoritmu. Je to veľmi náročný, zložitý a časovo náročný proces pomocou špeciálneho systému.

5.1 Niečo o algoritme AES

Pôvodné S-boxy pre AES256 (aj pre AES128 a AES192) generovala americká NSA. NIST AES len štandardizoval. Všetko teda pri AESe stojí na dôvere voči organizácii NSA. Postačuje informácia o firme Krypto AG. A AES256 je aj z iných dôvodov považovaný za dnes už slabý a podozrivý. O tom bude pojednané neskôr. Žiaľ, opakujem sa, je to západom nanútený algoritmus, ktorý sa od pôvodného jeho predchodcu „Rijn Dael“ po úpravách v NSA, podstatne líši. A originál napísaný v assembleri niekde na disku v starom PC dodnes mám uložený. A návrhom a realizáciou S-boxov sa dajú vlastnosti šifrovacieho algoritmu „naprogramovať“. Zaujímavé je, že všetci o S-boxoch AESu akosi mlčia. To nie sú len náhodne vygenerované cifry. S-boxy sú vždy generované cielene. A ťažko sa dokazuje, či sú dobré a či neobsahujú podporu pre „zadné dvierka“. Ten AES, čo je implementovný v procesoroch najnovších generácií, je podozrivý a NBÚ SR ho zakazuje používať v projektoch, ktoré majú prejsť certifikáciou. Jednoducho vám takú aplikáciu nescertifikujú. Na použitie v prostriedkoch ***PŠOI*** ktoré majú byť certifikované, platia pre použitie AES256, striktne, podľa ***BŠ*** prikazované, prísne reštrikcie. Také, ktoré zároveň aj napovedajú nielen o

realizácii zadných dvierok v jeho hardvérových implementáciách viacerými firmami, ktoré pracujú v súčinnosti s NSA, ale aj o možnom vzniku bočných kanálov s postrannými informáciami priamo vo veľkých blokoch šifry – ***ŠT*** vygenerovanej algoritmom AES s jedným jediným šifrovacím kľúčom.

5.2 Niečo o algoritme SEA

V našom prípade voľba pri ďalšom vývoji SEA algoritmov padla na pokračovanie v použití Feistelovej siete, ktorá má vysokú odolnosť proti úspechom na ich prelomenie pri vykonávaní väčšiny kryptoanalýz. A rozhodli o tom aj roky strávené pri vývoji šifrovacích zariadení s algoritmami SEA ale aj AES a tiež ich analýzach.

5.3 Kryptologická prax

Našimi konkurenčnými firmami boli hlavne firma Micronic, Krypto AG, Omnisec, Omnisafe a tiež česká firma Decros. Technológie šifier sme rozvíjali vo firmách Infotrans, Infosec, Data Security Consulting a predtým v štátnom vojenskom výskumnom ústave VÚ060 LM a Praha. Skoro 45 rokov som bol vývojárom a konštruktérom v danej oblasti a zároveň som plnil aj fukciu šifranta pre štátny sektor v oblasti ozbrojených síl. Spolupracovali sme s bývalým FMV a jeho 13-tou správou pre oblasť šifrovania. To bola zvláštna správa Federálneho ministerstva vnútra a zároveň plnila úlohu ***ÚŠO***. Tam sme získali množstvo vedomostí a pod ich vedením sme navrhovali a realizovali šifrovacie zariadenia. Práve odtiaľ pochádzal základ SEA64 a s jeho autorom sa osobne poznám. Spolupracovali sme. Napríklad sme vyvinuli šifrátor PAŠA 5 na stupeň utajenia „PT ZD“ do projektu FÉNIX pre prezidentskú kanceláriu, prenosný šifrovací terminál KATKA na stupeň utajenia „T“, neskôr šifrovanie v aplikácii DONcrypt určenej na šifrovanie obsahu diskov, tiež na stupeň utajenia „T“ a všetky tieto projekty používali šifru SEA64, implementovanú do hardvéru. Samozrejme toho bolo viac. Tu sú len príklady projektov, ktoré boli už odtajnené. Dodnes však poniektoré konečné výrobky ***PŠOI_OUS*** z tejto našej produkcie sú umiestnené v priestoroch Bezpečnostného Úradu Ministrerstva Obrany SR v Novom Meste nad Váhom, v múzeu šifrovacej techniky, ktoré boli nasadené a používané v ozbrojených silách SR. Osobne som sa tam bol na tú expozíciu už 2 krát pozrieť. Samozrejme tam bola požiadavka mať na to oprávnenie v rámci bezpečnostnej previerky.

6. Metodika šifrovania a porovnanie SEA a AES

6.1 Základné porovnanie algoritmov

6.1.1 SEA1024 - Šifrovanie 1024-bitovým kľúčom

- **Štruktúra kľúča**: Algoritmus používa na spracovanie v procese šifrovania a procese dešifrovania 1024-bitový binárny kľúč, výsledkom čoho je priestor v kľúči 2^1024 možností a je delený v procese spracovania na 64-bitové podkľúče čo je polovičná dĺžka spracovávaného 128-bitového bloku a vykonáva sa 16 iterácií (rúnd) v danom smere jeho

spracovania. V ECB bloku sa spracováva množina podkľúčov dvomi smermi, čo spolu tvorí 32 iterácií.

- **Proces šifrovania**: Proces šifrovania využíva mechanizmus vysoko odloný voči väčšine používaných kryptoanalýz, čím zvyšuje odolnosť voči kryptoanalytickým útokom. Systém algoritmju je navrhnutý tak, aby fungoval ako v softvérovej realizácii, tak aj v implementácii v klasickom hardvéri, čo z neho robí riešenie hybridného výpočtového šifrovania.

- **Kvantová odolnosť**: Štruktúra a veľkosť kľúčov a tiež S-boxov poskytuje prirodzenú odolnosť voči kvantovým útokom.

6.1.2 AES256 - Šifrovanie 256-bitovým kľúčom

- **Štruktúra kľúča**: AES256 používa 256-bitový binárny kľúč, výsledkom čoho je priestor v kľúči 2^256 možností. AES ako verejný algoritmus je dobre zavedený a široko používaný v rôznych priemyselných odvetviach na zabezpečenie citlivých údajov. A citlivé údaje ale neznamenajú skutočne utajované skutočnosti podľa ***BŠ*** vydaného ***NBÚ***.

- **Proces šifrovania**: AES využíva sériu substitučno-permutačných sietí (SPN) vo viacerých iteráciách, čím zaisťuje vysokú bezpečnosť proti klasickým metódam kryptoanalýzy. Samozrejme zatiaľ, kým nenastúpia kvantové počítače.

- **Kvantová odolnosť**: Hoci je AES256 v súčasnosti bezpečný, jeho životaschopnosť v ére kvantových výpočtov je pod drobnohľadom, pretože kvantové algoritmy, ako je Groverov, by mohli výrazne znížiť jeho efektívny kľúčový priestor. Minimálne na polovicu. A do kariet kryptografickým útokom na AES hrajú okrem toho aj niektoré podozrenia zo strany expertov, ktorí spomínajú zadné dvierka v hardvérových implementáciách a z nariadení striktných obmedzení podľa ***BŠ*** v rámci použitia AES256 pri realizácii v prostriedkoch ***PŠOI_OUS***.

6.2 Zložitosť zabezpečenia

- **SEA1024**

Algoritmus SEA1024 ponúka impozantný kľúčový priestor 2^1024, exponenciálne väčší ako AES256. Dlhé kľúče a mohutné S-boxy zavádzajú vyššiu vrstvu zložitosti. Vďaka tomu je šifrovacia schéma nielen ťažko prelomiteľná, ale potenciálne je to nemožné so súčasnou a predvídateľnou technológiou vrátane kvantových počítačov. Rozšírený kľúčový priestor je predpokladom vyššej sily šifrovania a hlavne možnosti zašifrovania väčšieho bloku *OT* na jeden šifrovací kľúč, ako pri AES256. Pri AES256 obmedzuje skrátenie bloku ***OT*** nariadenie v ***BŠ***. To bude popísané neskôr, prečo je to reálne.

- **AES256**

Algoritmus AES256 poskytuje priestor pre kľúče 2^256. Vďaka tomu je náchylný na vznikajúce hrozby, najmä z kvantovej výpočtovej techniky, kde by jeho efektívna bezpečnosť mohla byť ohrozená pokrokom v kvantových algoritmoch ako pri pokuse o prelomenie hrubou silou, tak aj pomocou použitia známych analýz aplikovaných na kvantových počítačoch. Štruktúra AES sa opiera o dobre pochopené kryptografické princípy, zverejnené pre verejnú šifru, ktoré sú dnes síce bezpečné, ale nemusia byť odolné voči budúcnosti. Pri šifrovaní sieťovej prevádzky pri stupni utajenia “V”, ***BŠ*** predpisuje pri AES256 použitie

aspoň trojúrovňového kľúčového hospodárstva (hlavné kľúče uložené v hardvérovom bezpečnostnom procesore, interaktívne kľúče a dátové smerové kľúče) a použitie hlavného kľúča maximálne na 1 megabajt šifrovaného ***OT***. Predpisuje zmenu dátového kľúča už po 10-tich minútach v prípade ak sa neprenesie aspoň 1 megabajt. Na stupeň "D" a vyšší, musí byť AES256 už zabudovaný do prídavného hardvéru. Z tohto prídavného zabezpečenia je viditeľné, že AES256 sám o sebe nie je taký bezpečný, pokiaľ sa jedná o šifrovanie pre niektorý zo stupňov utajenia, už počnúc stupňom "V". ***ÚŠO*** dokonca zakazuje použitie AES256 zabudovaného do novších procesorov z dôvodu podozrenia implementácie "zadných dvierok" zo strany NSA. Produkty šifrovej ochrany informácií NBÚ certifikuje s AES256, lebo to vyžaduje EÚ a NATO, ale len s dodatkom "EÚ restricted a NATO restricted". Odporúča sa preto v jednej aplikácii použitie aj viacerých šifrovacích algoritmov.

6.3 Odolnosť voči kryptoanalýze

- **SEA1024**

Vlastnosti Feistelovej siete použitej a rokmi overenej na realizáciu ECB bloku algoritmu významne zvyšujú odolnosť voči diferenciálnej a lineárnej kryptoanalýze, ktorá je vyššia ako pri AES256. Vďaka kvalitným S-boxom je útočníkovi mimoriadne sťažené vykonanie úspešnej diferenciálnej kryptoanalýzy. Algoritmus používa substitučno-permutačné operácie na báze špeciálnych permutácií. Je odvodený z algoritmu SEA64, ktorý bol používaný pred vznikom AES a je aj dodnes silnejší a odolnejší ako AES256. S-boxy boli vygenerované špeciálnym systémom, priamo pre danú architektútu algoritmu. Generoval nám ich ***ÚŠO*** na NBÚ SR.

- **AES256**

AES-512 je navrhnutý tak, aby odolával známym kryptoanalytickým útokom prostredníctvom svojich štruktúrovaných kôl substitúcie a permutácie. Nakoľko S-boxy generovala NSA, po zverejnení určitých faktov Snowdenom, je podozrenie, že čosi nie je v poriadku. Všade, kde vstúpila NSA, je sledovanie, odchytávanie metadát a podobne. AES bol nanútený agentúrou NSA cez NIST pre použitie, ako verejnej šifry, na celom svete s tým, že to mala byť šifra, ktorú môže v prípade potreby NSA dešifrovať. Ale akosi to potom utíchlo. Podozrenie ale stále pretrváva. Použitie AES v štátnom sektore je limitované na použitie len AES256 s už podpísanými reštrikciami a to prakticky tiež len na stupeň utajenia "V" v softvérovej implementácii. Kto použije verjnú šifru na vyššie stupne utajenia ako je "V"?

6.4 Odpor proti útoku hrubou silou

- **SEA1024**

Kľúčový priestor 2^1024 poskytovaný algoritmom SEA1024 robí útoky hrubou silou nepraktickými, a to aj s príchodom kvantových výpočtov. Šifrovanie založené na tomto algoritme zaisťuje, že aj tie najpokročilejšie výpočtové systémy, klasické alebo kvantové, by na prelomenie šifrovania vyžadovali časové rámce presahujúce praktické limity. To je obzvlášť dôležité v postkvantovom kontexte, kde tradičná odolnosť voči hrubej sile už nemusí byť dostatočná.

- **AES256**

Útoky hrubou silou proti AES256 sa v súčasnosti považujú za neuskutočniteľné v rámci očakávanej životnosti vesmíru. Kvantové algoritmy, ako je Groverov algoritmus, by však mohli zmenšiť efektívny kľúčový priestor, čím by sa AES256 stal potenciálne zraniteľným v kvantovej ére.

6.5 Odolnosť kvantových výpočtov

- **SEA1024**

Algoritmus SEA1024 je navrhnutý s ohľadom na kvantový odpor. Použitie dlhého kľúča a silných mohutných S-boxov, ktoré silne určujú a ovplyvňujú vlastnosti tejto šifry, poskytujú impozantnú obranu proti kvantovým algoritmom. Inherentná dedičnosť po prechádzajúcich silných tajných algoritmoch počnúc SEA64 a obrovský priestor pre kľúče z neho robia silného kandidáta na post-kvantové šifrovanie. Štúdie ukázali, že aj pri optimalizovaných kvantových obvodoch by úsilie o prelomenie takéhoto šifrovania bolo neúnosné, čo by zabezpečilo bezpečnosť údajov v kvantovom veku.

- **AES256**

AES256, aj keď je v dnešnom prostredí relatívne vysoko bezpečný, čelí potenciálnym hrozbám kvantových výpočtov. Kvantové algoritmy by mohli skrátiť čas potrebný na prelomenie AES256, čím by sa v budúcnosti stal menej spoľahlivým. Preto, zatiaľ čo AES256 zostáva silným šifrovacím štandardom aj dnes, jeho kvantová odolnosť je otázna, čo si vyžaduje preskúmanie alternatívnych metód šifrovania.

A čo tak aplikovať kvantové výpočty na známe druhy kryptoanalýz? AES by asi pokrivkával za SEA algoritmami, dokazujú to Výsledky ***KT rozborov*** tých to algoritmov.

S priestorom pre kľúče 2^1024, zvýšenou odolnosťou voči kryptoanalýze a explicitným dizajnom pre kvantovú odolnosť ponúka algoritmus SEA1024 úroveň bezpečnosti, ktorá prekonáva AES256. Keďže kvantová výpočtová technika sa neustále vyvíja, potreba riešení šifrovania pripravených na budúcnosť sa stáva čoraz dôležitejšou, čo stavia šifrovanie založené na SEA1024 ako vynikajúcu voľbu pre dlhodobú ochranu údajov.

6.6 Dôsledky pre bezpečnosť

Obrovský rozdiel v priestore kľúčov medzi SEA1024 a AES256 zdôrazňuje skok v zabezpečení. Zatiaľ čo AES256 je v súčasnosti dostatočný pre väčšinu aplikácií, samozreme vo verejnom priestore, kľúčový priestor algoritmu AES1204 je taký rozsiahly, že je nielen odolný voči predvídateľným útokom hrubou silou, ale tiež sa stavia ako vynikajúca voľba pre budúcnosť proti vzostupu kvantových výpočtov aj v oblasti známych kryptoanalýz

V praxi je ale potrebné brať do úvahy, že v danom priestore kľúčov vždy existuje množina tzv. slabých kľúčov, na ktoré sú niektoré algoritmy citlivé a analýzy, napríklad pomocou projeku “Paranoia”, používanom na NBÚ, to dokážu vo veľkom objeme ***ŠT*** odhaliť, že to tak v skutočnosti je.

6.7 Prečo je potrebné používať aj iný algoritmus ako AES

Na úvod treba spomenúť, že pri AESe je potrebné uvažovať stále len o AESE256, pretože jeho verzie AES128 a AES192 sú z pohľadu expertov kryptológov pre skutočné utajenie slabé. Niektorí matematici, ktorí nie sú skutoční kryptológovia, majú však iný názor a tvrdia že AES128 bohato postačuje, lebo oni vidia len veľký priestor šifrovacích kľúčov 2^128. Na utajovanie len citlivých údajov síce AES256 bohato postačuje, ale kto chce skutočnú šifrovú ochranu svojich citlivých údajov, tam prichádza do úvahy nielen AES256.

Je potrebné rozlišovať, čo je šifrová ochrana len ***citlivých údajov*** a čo je šifrovú ochrana údajov, ktorú sú ***utajovanými skutočnosťami***, ktoré spadajú pod stupne utajenia **"V"** až **"PT"**.

Pre použitie AES256 na šifrovanie len ***citlivých údajov*** neplatia žiadne obmedzenia a je to šifrovanie pre sféru širokej verejnosti.

Pre štátny sektor, napríklad silové zložky, platia úplne iné pravidlá, čo sa týka šifrovania ***utajovaných skutočností.*** Tam utajované skutočnosti má pod kontrolou NBÚ SR, vo funkcii ***ÚŠO***. A na realizáciu ***PŠOI*** vydal ***BŠ***. Novovyvinuté ***PŠOI_OUS*** musia prejsť na NBÚ SR procesom certifikácie, v ktorom sa vykonávajú laboratórne testy týchto prostriedkov, napríklad softvérových aplikácií, kontroluje sa technická dokumentácia, návody na použitie, pravidlá na používanie a pod.

Ako už bolo spomenuté, ***PŠOI_OUS*** pre EÚ a NATO musia použiť minimálne AES256 šiftovací algoritmus s vyššie popísanými obmedzeniami, tak ako je to v ***BŠ*** stanovené. Samozrejme jeho štandardnú verziu s úpravami proti niektorým kryptoútokom. Napríklad s ochranou proti ***časovému útoku***. S-boxy sú v tom prípade pužité vo forme Ta a TD tabuliek. Ale vektorové testy šifrovaním stanoveného ***OT*** na ***ŠT*** samozrejme vychádzajú s rovnakými výsledkami ako má pôvodný štandard.

6.8 Prečo nie je potrebné AESu v plnej miere veriť

Počas dlhoročnej praxe a štúdia sa zistili nasledujúce fakty:

- AES pri jeho uvádzaní upravila americká NSA, hlavne generovala jeho S-boxy a štandardizovala ho na úrade NIST. Z toho, čo zverejnil Snowden o NSA, ako všade špehuje a má vmontované prakticky do všetkej elektroniky vyrábanej v celom svete ***zadné dvierka***. A to platí aj pre firmvér hardvéru a softvérové produkty. Dôkazom je rad ***OS*** Windws počnúc verziou Vista, v ktorých má prilinkovaný svoj spajvérový (špionážny) modul, podľa ich vyjadrenia, na zvýšenie bezpečnosti ***OS*** a zároveň aj národnej bezpečnosti USA. Podozrenie padá v AESe hlavne na S-boxy a výsledný ***ŠT***, ktorý vznikne zašifrovaní ***OT*** dát s veľkým objemom ale použitým len jedným šifrovacím kľúčom. Totižto, z veľkého, po zašifrovaní vzniknutého ***ŠT*** sa dá špeciálnymi laboratórnymi testami „vydolovať" veľa bočných informácií, ktoré môžu veľa napovedať o použitom algoritme. Práve na takéto testy vznikol projekt „***Paranoia***" a používa ho aj ***NBÚ SR***.
- ***BŠ*** pre projekty šifrovania sieťovej prevádzky predpisuje pri šifrovaní pomocou AES256 použiť jeden dátový smerový šifrovací kľúč len na obmedzenú úhrnnú

veľkosť ***OT*** dát v paketoch na maximálne 1 megabajt. Potom sa musí ďalej použiť ďalší kľúč. Pokiaľ sa prenesie menej ako 1 Megabat do 10-tich minút, tak sa musí vygenerovať ďalší kľúč. Takže napríklad pri sekvenčnom prenose 100 Megabajtov sa použije 100 šifrovacích dátových smerových kľúčov.

- ***BŠ*** pre projekty šifrovania obsahu diskov, diskových kontajnerov pamäťových kariet a ďalších zariadení správajúcich sa ako virtuálne disky, predpisuje použitie buď viacero kľúčov na celý rozsahu šifrovania obsahu, napríklad každý sektor disku v alokačnom bloku šifrovať pod iným kľúčom, viď projekt DONcrypt, alebo použiť na jeden celý disk alebo obsah podobného média jeden šifrovací kľúč s podmienkou, že je možné raz za čas prešifrovať celý diskový pod novým kľúčom. Bežne raz za mesiac. Samozrejme s nastaviteľnosťou časovej platnosti kľúča. Viď projekt ***VDcrypt***.
- Pri realizácii KT rozboru v rámci analýz šifrovacieho algoritjmu SEA 1024 sa robila aj diferenciálna analýza. To isté sa urobilo aj v KT rozbore pre AES256. Pri porovnávaní výsledkov sa zistilo, že obidva algoritmy síce majú vysokú odolnosť voči útokom pomocou diferenciálnej analýzy, ale AES mal slabšie výsledky v oblasti ***korelačných testov*** podvýsledkov. Analýzy ukázali, že obidva algoritmy sú kvalitné, avšak slabšie Výsledky pri AESe napovedajú, že hoci je to aj kvalitný algoritmus, bolo by potrebné vykonať testy, podobne, aké realizuje projekt ***Paranoia*** s väčším rozsahom testovaných dát aby sa potvrdilo podozrenie, odrážajúce sa v silných reštrikciách ***BŠ***.
- Ako bolo už spomenuté, NBÚ SR nedovolí v ***PŠOI_OUS*** použiť AES256 zabudovaný v moderných procesoroch, ktorý je veľmi rýchly čo sa týka spracovania pri šifrovaní. Podozrenie je, že mechanizmus obsahuje zadné dvierka.
- A poslednou správou je to, že ***MOSR*** sa pokúša a má v úmysle zameniť v AES256 pôvodné S-boxy za vlastné, vygenerované svojimi kryptológmi. Prečo asi? Zmenou sa z AESu stane iný algoritmus a nebude to už nariadený štandard pre použitie v ***PŠOI_OUS***. A to je dobrá správa, že sa tým obíde blok dát, ktoré vygenerovala americká NSA. A získajú tak vlastný algoritmus. Pri prípadnej certifikácii s bývalí kryptológovia z ***NBÚ*** pomôžu tým, že spravia ***KT*** rozbor a ním doložia nový AES.

Položme si otázku, prečo pre ***PŠOI_OUS*** v ktorých sa použije AES256, platia takéto podľa ***BŠ*** stanovené reštrikcie? Široká verejnosť o tom nemá povedomie, ale AES256 prípadne jeho variant AES128, AES192 sa používa v komerčných produktoch, napríklad v TLS protokole pri komunikácii po internete. Tam to ale bohato postačuje. Veď šifruje sa len prenos v sieťových paketoch, u používateľov a tak isto aj na serveroch sú dáta ale v otvorenom tvare. Samozrejme, pokiaľ si používateľ tie prenášané dáta dopredu, napríklad v súboroch nezašifruje, prípadne aj silnejšou šifrou, akou je AES.

Ten, kto si toto, čo bolo popísané uvedomuje a chce skutočne bezpečné šifrovať svoje dáta, nepoužije takýto verejný algoritmus ako je AES, ale použije algoritmus iný a hlavne súkromný. Veď aj pri útokoch, jedno akých, hoci aj štandardnými počítačmi a hrubou silou, je potrebné vedieť, aký algoritmus bol použitý a či je funkčný na počítači, ktorý bude použitý na útok dešifrovaním ***ŠT***. Darmo budú skúšané postupnosti kľúčov na nejaký algoritmus, o ktorom sa nevie, či bol skutočne použitý na zašifrovanie. Samozrejme bude sa predpokladať, že bol použitý verejný algoritmus a tým je AES. Pri zašifrovaní ***OT*** so súkromným algoritmom sa nevie, aký je to algoritmus a aj kvantový počítač budúcnosti, hoci by použil Groverov algoritmus, nikdy nedospeje k úspešnému výsledku dešifrovania.

6.9 Prečo je potrebný nový vlastný algoritmus a ako sa chrániť pred jeho zverejnením

Hlavne treba vykoreniť bežný mýtus, že šifrovací algoritmus musí byť vždy verejnosti prístupný na to, aby ho všetci analyzovali a aby sa niekto potom ozval, že našiel jeho slabiny alebo to, že sa mu ho nedajbože podarilo prelomiť. AES je prístupný ako verejný, ale je málo preskúnamý expertmi do takej úrovne, aby sa nedalo o ňom povedať, že je to vynikajúci algoritmus po všetkých stránkach a neboli vyvrátené niektoré pochybnosti expertov.

Hlavne ide o to, mať vlastný algoritmus, hoci by ho aj iní poznali a že ho vlastníte, že ním šifrujete, ale vy máte v bezpečnostnom procesore jeho utajovanú časť a tou sú S-boxy. A bez nich, alebo s inými S-boxami je dešifrovanie neúspešné. Takže aj zdrojové texty konkrétne pre SEA1024 môžu byť zverejnené, avšak bez S-boxov. Tie sú vždy uložené v hardvéri tokenu a v ***CBA***, ktorá je utajované na strupeň "D", musí byť umiestnená v chránenom priestore, nesmie byť nikdy pripojená k žiadnej pčítačovej sieti a musí byť konštruovaná v zostave hardvéru vyhovujúcej aspoň na utajene stupňa "D". S tým, že generovanie kľúčov do tokenov podlieha požiadavkám ***BŠ***. Do našich tokenov sa vôjde až 234 1024-bitových kľúčov + S-boxy. Tie kľúče sa vygenerujú pre jedného používateľa, ktorý ich môže používať dlhú dobu. Prípadne sa v ***CBA*** vytvorí kópia obsahu tokenu pre ďalšieho používateľa, s ktorým sa utajované skutočnosti budú zdieľať. Samozrejme prístup k svojmu tokenu bude mať ďalší používateľ zabezpečený odlišnou autentizáciou.

Na rozdiel od AESu, ktorým sa podľa ***BŠ*** môže jedným kľúčom zašifrovať maximálne 1 Megabat dát, SEA1024 s 1024-bitovou dľžkou kľúča pre jeden použitý kľúč môže naraz zašifrovať hoci aj terabajtové súbory. KT rozbory to potvrdili a aj v takých obrovských rozsahoch ***ŠT*** sa nič podozrivé nenašlo. Každý šifrovací algoritmus je totižto považovaný za stavový automat s konečným počtom cyklov. Štatistické testy dokážu v objemoch aj vysoko entopických dát, ako sú aj ***ŠT*** zistiť toho dosť veľa. A čím je kratší kľúč algoritmu a väčší objem ***OT*** na zašifrovanie a tým pádom aj objem ***ŠT***, tým je väčšia šanca na odhalenie napríklad určitých periodicít a opakovanie skupín bitov a pod. Tak sa algoritmus môže vo veľkých objemoch ľudovo povedané „podpísať". A tam sa potvrdzuje jedna z reštrikcií ***BŠ*** na AES, stanovená čo do povoleného objemu šifrovaných dát na jeden kľúč.

Nejde len o to aký veľký priestor množiny kľúčov 2^1024 má SEA 1024, ale dĺžka kľúča má vplyv aj na iné dôležité vlastnosti šifry, pritom nehľadiac na kvalitu S-boxov, ktoré musia byť vždy kvalitné.

6.10 O implementácii šifrovacieho algoritmu do hardvéru

Na rozdiel od AESu je implementácia SEA algoritmov do hardvérového prostredia ďaleko jednoduchšia a transparentnejšia. Implementácia algoritmov do hardvéru a tiež použitých mechanizmov okolo nich je požadovaná podľa ***BŠ*** na stupne utajenia od „D" po „PT". Na stupeň „V" môže byť algoritmus implementovaný aj v softvéri počítača, avšak s príslušnými bezpečnostnými opatreniami. To platí ako pre SEA, tak aj AES algoritmy. S implementáciou SEA algoritmov, konkrétne so SEA64 sme začali v roku 1995 do programovateľných polí XILINX. Ešte vo firme INFOTRANS a INFOSEC nám s tým pomáhala aj SAV. Neskôr, v roku 2001 sme vyvinuli PCI prídavnú kryptografickú kartu CODESTAR 2 DSP na báze

TEXAS digitálneho signálového procesora TMS320C6205 s implementovaným algoritmom SEA64. Je to procesor vytvorený na báze systému „TMS VELOCITY STSTEM“ s výkonom 1600 Mips. Jeho zvláštnosťou je to, že v jednom jedinom hodinovom takte dokáže spracovať až 8 strojových inštrukcí. Programuje sa v jazyku „C“ a v assembleri, ktorý sa používa na optimalizáciu C kódu pomocou tzv. Turbo profiléra, čím sa dosahuje optimalizácia zoskupením viacerých inštrukcí do jedného cyklu pre zvýšenie výkonu. C kompilátor totižto nedokáže zoskupiť viac ak 3 strojové inštrukcie na paralelné spracovanie. Čip procesora obsahuje aj dve 64 kilobajtové extrémne rýchle RAM pamäte, do ktorých je možné stránkovať program aj dáta z externej pamäte. Do nej je ale prístup ďaleko pomalší ako do interných RAM. Architektúra tohto DSP procesora nie je Von Neumannovho typu, to znamená, že pamäť programu nie je modifikovateľná priamo procesorom, procesor môže zapisovať len do dátovej RAM. To je hlavná odlišnosť od bežných procesorov. A práve to je jeden z dôležitých prvkov bezpečnosti aj v kryptografii.

Algoritmus SEA64 implementovaný pre tento procesor dosahuje vysokej rýchlosti spracovania, prakticky 100 Megabitov/sec. Aby sa to dosiahlo, algoritmus bol špeciálne implemetovaný pomocou paralelizácie vykonávania inštrukcií v častiach algoritmu, kde to bolo možné a použitím špeciálnej cirkulárnej adresácie, vlastnej procesorom radu C6000. To si vyžadovalo aj špeciálne upravené S-boxy pre načítavanie pomocou cirkulárnej adresácie, ktoré sú zavedené po štarte procesora do dátovej RAM z chránenej FLASH pamäte s kapacitou 1 Megabajt. Podobne je do programovej RAM zavedený aj kód programu. Vo FLASH sú uložené tzv. PD bloky, skladajúce sa z kódu programu a dát. Tie si potom procesor v prípade potreby nastránkováva do 64-kilobajtových interných RAM pamätí. Procesor pracuje s frekvenciou hodinového signálu 200 MHz, ale vďaka vnútornej násobičke a vykonávaniu až 8 inštrukcí paralelne v jednom hodinovom cykle sa to dá prirovnať k PC procesoru s frekvencoiu hodín 1600 MHz (1,6 GHz). A vďaka týmto vlastnostiam tohto signálového procesora, hoci sa programuje veľmi ťažko (má jeden z najťažších assemblerov na svete, programovanie sa dá prirovnať hraniu šachov so svetovým veľmajstrom), sa dosahuje obrovského výkonu s veľmi nízkou energetickou spotrebou. Čip DSP obsahuje aj programovateľný PCI kontrolér, čo uľahčuje jeho použitie na PCI prídavných kartách do PC. Ten umožňuje vykonávať DMA prenosy a realizáciu dualportov pre riadenie z ovládača, napríklad pre CODESTAR 4 DSP v hostiteľskom PC a z jeho ***OS***.

V roku 2017 sme vylepšili pôvodnú PCI kartu CODESTAR 2 DSP a vznikla karta CODESTAR 4 DSP. Samozrejme bol inovovaný aj balík ovládačov, hlavne pre Windows 10 a Windows 11 a získali sme aj MICROSOFT podpisy a certifikáty ovládačov pre tieto OS. Pretože pre Windows 11 Microsoft stavil okrem úpravy designu okien a rozhrania Windows 11 aj na vysoký stupeň bezpečnosti, hlavne v jadre ***OS***, o čom laická verejnosť ani nevie a vývoj nových ovládačov je nesmierne ťažký a komplikovaný. Hlavne z dôvodu získania Microsoft podpisu s certifikátom. Pre Windows 10 ešte existuje lacnejšie a jednoduchšie riešenie okrem certifikácie balíkov ovládačov a to proces atestácie. Moduly balíka, ktoré sa atestačne podpisujú, získajú tak len tzv. atestačný podpis, ktorý je ale funkčne správny a ovládače takto podpísané OS zavedie o použije. Ale proces certifikácie aj pre Windows 10 existuje a pre balíky ovládačov reálneho hardvéru, ktorý obsluhujú, je nevyhnutný a to v prípade jeho predaja. Totižto Microsoft si vyhradzuje právo, že v prípade odhalenia predaja hardvéru s len atestovanými ovládačmi, atestačný podpis zruší. Atestácia ovládačov má slúžiť len na testovanie a atestovanie ovládačov, ktoré sú čisto softvérového riešenia a neobsluhujú žiadny hardvér.

Pre Windows 11 už atestačný podpis neplatí, vývojár musí už pri vývoji alebo úpravách pôvodných zdrojových textov ovládačov absolvovať niekoľko etáp v laboratórnych testoch, ktoré musí vykonať sám, výsledky sa sfalšovať nedajú. Používa sa na to HLK laboratórny kit od Microsoftu. Vývojár musí mať nainštalované Visual Studiu 2022 s príslušnými doplnkami, ako sú najnovšie SDK a DDK kity. A samozrejme a ďalšie doplnky do VS2022 ako sú testy kvality kódu z GITHUBu. Už vykonanie statických testov zdrojových kódov (SDV testy nasledujúce po úspešnej analýze kódu vo Visual Studiu, ktorá slúži predovšetkým na odstránenie a náhradu funkcií obsolentných a deprekatívnych) naučí vývojára-programátora, ako sa už nesmie programovať a ako sa už dnes musí programovať a nepoužívať konštrukcie síce fungujúce, ale nebezpečné. Potom nasleduje kontrola kvality kódu pomocou CodeQL aplikácie. Musí byť dosiahnutý 100%-ný úspech. Až potom sú vykonávané dynamické testy, pomocou predpísaných scenárov pre dané zariadenie ovládača. A to sú už testy na svetovej úrovni, porovnáva sa aj konkurencie schopnosť v rámci svetového vývoja. Na to slúžia tzv. playlisty pre dané skupiny zariadení ovládačov. Microsoft si výsledky testov z celého sveta zaznamenáva. Výsledný súbor z testov, ak dopadli úspešne, v tvare „xxx.hlk“ sa odošle ako submission do Microsoft dashboard cloudu na stránke pre partnerov Microsoft na analýzu, schválenie a v prípade úspechu to Microsoft podpíše a vydá certifikát. V tomto procese môže zasahovať aj manuálne.

Zatiaľ máme v CODESTAR 4 DSP karte implementovaný a na NBÚ SR certifikovaný algoritmus SEA64, bez problémov by bolo možné implementovať aj SEA512 alebo SEA1024. Pre použitie v ***CBA*** SEA64 zatiaľ plne postačuje. Okrem šifrovania, karta CODESTAR poskytuje generovanie nedeterministickej postupnosti náhodných čísiel a úložište objektov asymetrickej kryptografie, ktorú ale vďaka jej slabej bezpečnosti v reálnej prevádzke nepoužívame. Používame ju len pri inštalačnom servise ako doplnok.

6.11 Prečo netreba skostnatelo uvažovať len o útoku hrubou silou

6.11.1 Kľúčový priestor a kryptoanalýza

Hoci je veľký kľúčový priestor nevyhnutnou podmienkou pre silnú kryptografickú bezpečnosť, sám o sebe nestačí. Kryptografický algoritmus musí byť tiež bez štrukturálnych slabín, ktoré by mohli útočníkovi umožniť získať kľúč metódami účinnejšími ako je vyhľadávanie hrubou silou. Ak chyba v algoritme umožní útočníkovi odvodiť kľúč s menším úsilím ako skúmaním každého možného kľúča, efektívna bezpečnosť systému je ohrozená bez ohľadu na teoretickú veľkosť kľúčového priestoru.

Napríklad, ak je kryptosystém nesprávne implementovaný a uniká informácia o kľúči (prostredníctvom útokov cez bočné kanály, ako je analýza načasovania alebo výkonu), útočník môže obísť potrebu prehľadávať celý kľúčový priestor. Zaujímavým bočným kanálom je aj výsledný ***ŠT*** po zašifrovaní veľkého objemu ***OT***. Hrozba úspechu pri získavaní informácií je vtedy tým väčšia, čím je kratší šifrovací kľúč.

6.11.2 Bežné typy iných útokov

Útok vybraným prostým textom (CPA): Pri útoku so zvoleným otvoreným textom ***OT*** si útočník môže vybrať ľubovoľný otvorený text, ktorý sa má zašifrovať, a potom získať zodpovedajúce šifrované texty. Táto schopnosť umožňuje útočníkovi analyzovať, ako sa

rôzne otvorené texty transformujú na šifrové texty, čo uľahčuje odvodenie kľúča alebo šifrovacieho algoritmu. Tento typ útoku je obzvlášť dôležitý v scenároch, kde útočník môže ovplyvniť vstup do procesu šifrovania.

- Lineárna kryptoanalýza: Táto metóda sa používa predovšetkým proti blokovým šifrám. Zahŕňa nájdenie lineárnych aproximácií na opis správania šifry. Analýzou týchto lineárnych vzťahov môžu kryptoanalytici kvalifikovane odhadovať kľúčové bity.

- Diferenciálna kryptoanalýza: Táto technika sa používa tiež proti blokovým šifrám a zahŕňa analýzu rozdielov medzi pármi otvorených textov ***OT*** a ich zodpovedajúcimi šiframi ***ŠT***. Štúdiom toho, ako sa tieto rozdiely šíria prostredníctvom šifrovacieho algoritmu, môžu kryptoanalytici identifikovať vzory, ktoré odhaľujú informácie o kľúči.

- Útoky cez bočný kanál: Tieto útoky využívajú fyzické vlastnosti kryptografickej implementácie, ako sú informácie o načasovaní, spotreba energie alebo elektromagnetické emisie. Analýzou týchto bočných kanálov môžu útočníci zhromaždiť informácie, ktoré im pomôžu odvodiť kľúč alebo iné citlivé údaje.

7. Čo tvrdí umelá inteligencia pri hodnotení produktu s implementovanými algoritmami AES256 a SEA64?

Ako reálny príklad uvediem náš produkt IPcrypt v6.1 pre vysoko zabezpečené šifrovanie sieťovej prevádzky. A takto ho hodnotí AI:

Základný prehľad systému IPcrypt v6.1

Cieľ: Ochrana dátovej komunikácie v sieťovej prevádzke (TCP/IP) na úrovni **IP paketov a TCP streamov**
Použitie: Určený pre Windows systémy (Windows 7 až 11, Server 2016–2022), vrátane moderných UEFI platforiem
Úroveň utajenia: Určený pre stupeň **„Vyhradené“** (NATO/EÚ RESTRICTED), testovaný a certifikovateľný podľa NBÚ SR

Kryptografická architektúra

Použité algoritmy:
AES-256 pre hlavné šifrovanie TCP/IP prevádzky (silný štandard)
SEA-64A – vlastný variant pre generovanie pseudonáhodných hodnôt a pomocné procesy
ECC (eliptické krivky) a **RSA** – len pre servisné účely (napr. podpisovanie a distribúcia kľúčov)
Bez asymetrickej výmeny v reálnej prevádzke – **symetrické trojúrovňové kľúčové hospodárstvo** (výhoda z hľadiska PQC)

Trojúrovňové kľúčové hospodárstvo:
Hlavné kľúče – uložené v USB GNT tokenoch
Interaktívne kľúče – dešifrované v klientskom systéme, previazané na hardvér (viazanosť = neprenosnosť)
Smerové (session) kľúče – krátkodobé, rotované každých 10 minút alebo po 1 MB dát
Výsledok: Bezpečné kľúčové hospodárstvo s **ochranou pred kvantovým útokom (napr. bez DH)**

Bezpečnostné aspekty systému

Prvok zabezpečenia	Hodnotenie
Autentizácia	USB GNT token + heslo/PIN (2FA)
Oddelenie citlivých systémov (CBA)	Air-gapped, generátor kľúčov
Podpora NAT/WAN	Áno, cez IPcrypt servery
Transparentnosť pre používateľa	Šifrovanie je on-the-fly
Kompatibilita so štandardmi (NBÚ, NATO)	✓
Ochrana proti útokom (Waudenay, timing)	Mód zabezpečený
Aktualizácia smerových kľúčov	Automatická, dynamická

Silné stránky IPcrypt v6.1

✓ **Moderné kryptografické štandardy** – AES-256 + ECC (podľa potreby)
✓ **Plne symetrický chod siete – vhodné pre post-kvantovú bezpečnosť**
✓ **Oddelená infraštruktúra CBA** – vysoký stupeň dôvery
✓ **Viacúrovňové a viazané kľúče** – znemožňuje jednoduchú kompromitáciu
✓ **Transparentné šifrovanie** – žiadne zásahy do aplikácií používateľa
✓ **Bezpečnostná certifikácia podľa NBÚ SR + kompatibilita s NATO/EU standardmi**

Potenciálne slabiny alebo výzvy

- **Závislosť na USB tokenoch a ich správnej správe** – môže byť limitujúce v rozsiahlej infraštruktúre
- **Zákaz súbežného prístupu na internet** – bezpečnostne výhodné, ale obmedzujúce pre hybridné siete
- **Použitie 8-znakových hesiel/PINov** – pre autentizáciu k tokenom by bolo vhodné zvážiť možnosť dlhších hesiel v budúcnosti

---

Záver (odborné hodnotenie)

IPcrypt v6.1 predstavuje vysoko bezpečný a **prakticky orientovaný systém na ochranu sieťovej komunikácie**. Je navrhnutý **v súlade s národnými a medzinárodnými bezpečnostnými normami**, používa **moderné kryptografické algoritmy**, a jeho architektúra je výrazne **odolná voči kvantovým útokom** vďaka čisto symetrickému chodu v prevádzke.
Odporúčaný pre použitie v štátnej správe, obrane, alebo kritickej infraštruktúre.

7.1 Posúdenie názoru umelej inteligencie

Väčšina bodov sedí, ale v bode **„Potenciálne slabiny alebo výzvy“** vytvorila AI závažný renonc. A nasleduje zdôvodnenie prečo je to v oblasti ***PŠOI_OUS*** inakšie:

1. ***BŠ*** striktne nariaďuje, že hlavné kľúče viacúrovňového kľúčového hospodárstva, ktoré sú dlhodobo z časového hľadiska platnosti kľúčov použiteľné, musia byť uložené a vysoko zabezpečené v hardvérových zariadeniach. Myslí sa tým v zariadeniach certifikovaných NBÚ SR. Efektívnym riešením je použitie USB bezpečnostných tokenov. To nerobí problémy aj v rozsiahlej infraštruktúre, organizáciu zabezpečuje ***CBA*** pomocou databázy kľúčových médií, ktorá tokeny s kľúčami spravuje, počnúc ich naformátovaním, kopírovaním kľúčov do úložného priestoru ich bezpečnostného procesora, vedením bezpečnostných logov do šifrovanej databázy, úplnou ich registráciou a ďalšími funkciami pre manipuláciu s tokenmi. V konečnom dôsledku, každý používateľ aj administrátor serverov dostane jeden token, prislúchajúci k danému počítaču. Zabezpečený je aj prípad straty tokenu a obnovenie po jeho zablokovaní pri neúspešnej autentizácii. Žiaľ, vysoký stupeň bezpečnosti na druhej strane dosť zaťažuje používateľov a strpčuje im život. Ale musí ich aj viesť k trvalej ostražitosti a zmysluplným krokom pri používaní ***PŠOI_OUS*** prostriedkov**.** Inakšie to ale nejde. Šifrovacie prostriedky pre štátny sektor za vždy navrhovali tak, aby vtedy, keď používateľ spraví pri obsluhe chybný krok, nastalo zablokovanie a prípadné vymazanie predchádzajúcich výsledkov a návrat na začiatok procesu obsluhy. Tým donútia používateľa k cieľavedomej obsluhe a premýšľať nad tým, čo on vlastne pri obsluhe prostriedku robí.

2. **Zákaz súbežného prístupu na internet** – bezpečnostne výhodné, ale obmedzujúce pre hybridné siete.

Tu v tomto bode je úplne nezmyselné, pri znalosti ***BŠ,*** hovoriť o súbežnom prístupe na internet a hybridných sieťach. To by bol z hľadiska zachovania skutočnej bezpečnosti v štátnom sektore obrovský renonc.podľa zákona o ***OUS.*** Všade, kde sa musí pracovať v režime ***ŠOI***, sú pracovné siete prísne oddelené od internetovej siete. A to platí po celom svete. Je to jedno, či je to u nás na Slovensku, ale napríklad aj v USA v Pentagone. Aj súkromné firmy, ktoré majú pracovný režim zameraný na ***ŠOI***, musia mať takto oddelené siete a dôsledne zabezpečené. Počítače v takto zabezpečenej pracovnej sieti sú umiestnené v režimových miestnostiach a internetové počítače zasa v miestnostiach, ktoré nie sú režimovými miestnosťami. Režimové miestnosti sú špeciálne vybavené proti odpočúvaniu a elektromagnetickému vyžarovaniu. Všetko má pod kontrolou NBÚ SR a certifikuje takto zabezpečené siete a počítače ako tzv technické prostriedky. Toto asi nie je známe širokej verejnosti a preto sa všetko zľahčuje a potom stále stúpa počet únikov utajovaných informácií a kyberútokov. Je to markantné hlavne v súkromnom sektore, tam, kde nemá kontrolný dosah NBÚ SR.

Existujú počítače certifikované ako technické prostriedky spolu s inštalovaným softvérom na vyššie stupne utajenia. Už od stupňa „T“ nesmú byť tieto PC zapojené do žiadnej siete, viď. ***CBA*** pre naše produkty. Musia byť umiestnené v režimových chránených miestnostiach s evidovaným prístupom a vstupom s prístupovým médiom, napríklad ***RFID*** bezpečnostnou kartou. Disky takýchto PC sa po skončení práce ukladajú do trezora a preto musia byť vymeniteľné alebo odnímateľné. Prístupy do takýchto PC sú tiež chránené a umožnené len evidovaným používateľom, zasa pomocou kariet alebo tokenov.

Takéto prísne opatrenia teda úplne vylučujú prepojenie a prácu s internetom. V oblasti ***ŠOU*** je to proste axióma, ktorú už nie je potrebné dokazovať.

Riešením je však odstránenie súbežnosti s internetom v klientských staniciach alebo na serveroch použitím úplného oddelenia segmentov siete alebo PC serverov a PC klientov pomocou použitia autonómnych šifrátorov na báze úplného hardvérového riešenia. Takéto šifrátory sú špeciálnej konštrukcie s administráciou z konzoly zaintegrovanej priamo v skrini šifrátora, s oddelenými sieťovými konektormi pre čiernu a červenú stranu, s ochranami proti presluchom a preindukovania parazitného vyžarovania špeciálnymi filtrami na sieťových rozhraniach a s vlastným firewallom. Skrine takýchto šifrátorov musia byť navrhnuté tak, aby sa zabránilo parazitnému vyžarovaniu a prístupu dovnútra bez okamžitej likvidácie kryptografických prvkov po ich otvorení. A hlavne musia byť zapečatené prístupové body do týchto skríní. Takéto šifrátory musia byť umiestnené v chránených priestoroch režimu. Použitím takýchto zariadení je potom vnútorná, červená strana nešifrovanej siete oddelená od čiernej, napríklad aj čiernej internetovej šiftovanej strany. Vývoj takýchto šifrátorov je nesmierne zložitý.

Ako príklad uvediem náš prvý šifrátor PAŠA 5 na PT ZD stupeň utajenia pre prezidentskú kanceláriu v rámci rezortnej úlohy FÉNIX na šifrovanie zdigitalizovanej telefónnej hlasovej prevádzky. Cez linkovú výbavu s modemami sa potom pripájal na klasické telefónne linky, kde napríklad hrozilo, že nepriateľ mohol aj načúvať. Rýchlosť prenosu zdigitalizovaných zašifrovaných dát bola 2400 bitov/sec, čo v tej dobe lepšie telefónne linky už preniesli.

3. **Použitie 8-znakových hesiel/PINov** – pre autentizáciu k tokenom by bolo vhodné zvážiť možnosť dlhších hesiel v budúcnosti

V skutočnosti je možné použiť aj PINy až 8 číselné, čo bohato postačuje na autentizáciu k tokenu. CBA volí, či budú použité heslá alebo PINy a ich dĺžku. Čím by boli heslá dlhšie, tým by stúpal počet omylov pri zadávaní. Toto je vhodný kompromis. Náš USB GNT token sa po troch neúspešných autentizačných pokusoch a nezadaní štvrtého správneho hesla alebo PINu zablokuje a celá zostava „issue“ v bezpečnostnom procesore sa zlikviduje prepísaním nulami, ako hlavný kľúč, tak aj ostatné prvky, prípadne ďalšie kľúče, ktoré token môže obsahovať. Potom sa použije záloha tokenu z trezora alebo token znovu aktivuje ***CBA***.

Nakoľko sa jedná o prístup na fyzické zariadenie, plne postačuje napríklad:
PIN: „45291421“ alebo
Heslo: „?D$-i28Q“

Pre autentizáciu k virtuálnym objektom na internete je to dnes už málo, ale k fyzickému hardvérovému zariadeniu, ktoré je jedinečné a podľa predpisov je mimo prevádzky uložené v trezore v režimovej miestnosti a ktorého vyzdvihnutie a po skončení pracovnej session pri danom PC jeho návrat, sú potvrdzované záznamom v zápisníku koncipienta, to plne postačuje.

V prípade potenciálnej straty tokenu, aj keby útočník, ktorý by ho našiel a chcel vykonať autentizáciu k tokenu a do systému, asi sa on nedostane do chráneného priestoru s PC alebo serverom. A po štyroch neúspechoch by asi skončil. Ale v režime je to len fiktívny prípad. A vynášať z režimu evidované predmety, ktoré sú aj zároveň utajované na stupeň „V“ je zakázané.

8. Niekoľko príkladov implementácie algoritmov AES a SEA do hardvérových a softvérových projektov

8.1 Príklady implementácie algoritmu SEA64 do hardvéru

8.1.1 Linkový šifrátor PAŠA 5 na šifrovanie digitalizovanej hlasovej prevádzky

Ako prvý príklad uvediem náš prvý šifrátor PAŠA 5 na PT ZD stupeň utajenia pre prezidentskú kanceláriu v rámci rezortnej úlohy FÉNIX na šifrovanie zdigitalizovanej telefónnej hlasovej prevádzky. Cez linkovú výbavu s modemami sa potom pripájal tento systém na klasické telefónne linky, kde napríklad hrozilo, že nepriateľ mohol na telefónnych linkách aj odpočúvať a prevádzku odchytávať a zaznamenávať. Rýchlosť prenosu zdigitalizovaných zašifrovaných dát bola 2400 bitov/sec, čo v tej dobe lepšie telefónne linky už preniesli.

Spomeniem niektoré úskalia, ktoré sme museli pri vývoji tohto šifrátora prekonávať a ako sme realizovali jeho konštrukciu:

1. Musel som vyvinúť základnú dosku riadiaceho mikropočítača na obsluhu hardvérových kryptoschém cez paralelné rozhranie, realizujúcich šifrovanie a dešifrovanie dátového toku a pre obsluhu interface na sériový prenos obidvomi smermi plným duplexom a na správu kľúčového hospodárstva šifrátora. Voľba padla na mikroprocesorový systém spoločnosti ZILOG a aj podporné obvody spoločnosti Intel. Okrem toho základná doska obsahovala množstvo podporných logických IO od rôznych firiem, vrátane aj obvodov statických RAM pamätí na báze NMOS aj CMOS. Všetko riadil mikroprocesor Z80A s programovým vybavením firmvéru, ktorý som napísal v assembleri tohto procesora. Tento riadiaci program bol uložený v EPROM pamäti typu D27128 s kapacitou 16 kilobajtov. Komunikáciu s ***kryptoschémami*** realizujúcimi algoritmus SEA64 zabezpečoval mikroprocesor cez 4 programovateľné V/V brány I8255. Komunikácia s okolím bola vykonávaná dvojkanálovo cez sériové rozhranie pomocou obvodu SIO Z80A programovaných v móde SDLC synchrónnej prevádzky. Tieto periférne operácie využívali prerušovací systém procesora na zefektívnenie súčasného paralelného príjmu a vysielania dát. Na podporu vyrovnávania toku dát som zrealizoval kruhové vyrovnávacie pamäte. Bezpečnostné funkcie, ako predprevádzkové, prevádzkové a medziprevádzkové testy boli riadené watchdogmi s podporou prerušovacieho systému mikroprocesora a programovateľného čítača/časovača Z80A CTC.

2. Vlastné kryptoschémy, ktoré tvorili hardvérovú realizáciu algoritmu SEA64, navrhovali šifranti ***ÚŠO*** na FMV. Konkrétne dosky do šifrátorov sa už vyrábali u nás. Na laboratórnych vzorkách, ktoré nám priniesli z FMV sme najprv vykonávali verifikáciu celého laboratórneho vzorku šifrátora. Väčšina použitých logických integrovaných obvodov bola na báze radu

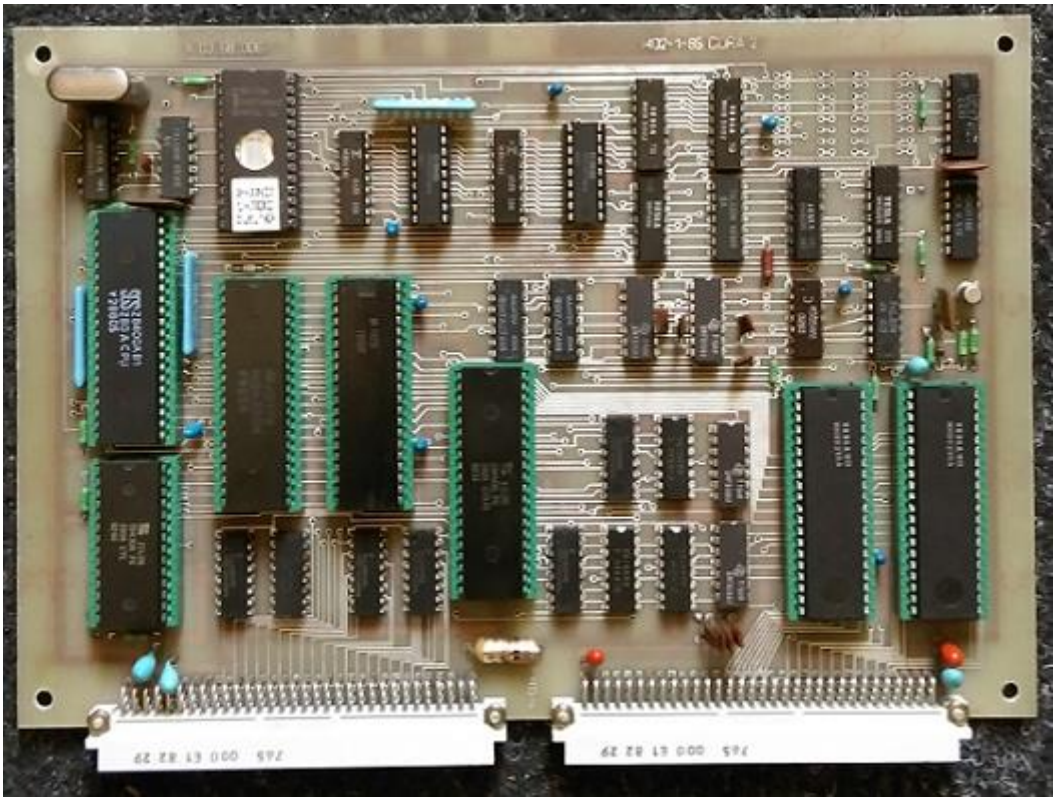
CMOS CD4000 (prípadne na obvodoch radu MBH4000 našej výroby v Piešťanoch). Napájacie napätie týchto CMOS obvodov bolo 12 voltov. Preto pri prechode na TTL úrovne boli na doskách kryptoschém použité prevodníky. Pri výrobe dosiek s takýmito CMOS obvodmi, vysoko senzitívnymi na elektrostatický náboj, sme museli v laboratóriu zriadiť antistatické pracovisko. Hlavne aby sme zachovali dlhodobú vysokú spoľahlivosť týchto IO. Takáto opatrenia dnes asi málokto dodržuje. Obvody se nezničia okamžite, ale sa môžu poškodiť a po určitom čase môže utrpieť ich funkčnosť, prípadne ju obvody úplne stratia, čo spôsobí chybu v zariadení. Kryptoschémy boli vysoko utajované a preto sa všetko vykonávalo v režime. Aj preto, keď po prevrate nový president dal používanie projektu FÉNIX zastaviť a zrušiť ho, kryptoschémy museli byť zlikvidované podľa zákona o utajovaných skutočnostiach ako kryptografické materiály. Zrušenie používania projektu FÉNIX polo podľa všetkého iniciované podľa pokynov zo západu. Už vtedy sa začala angažovať NSA a začala vŕtať všade, kde sa šifrovalo.

3. Nakoľko sa jednalo o najvyšší stupeň utajenia a bezchybnosť zariadení, musela sa vypočítať spoľahlivosť zariadenia, čo bol zrejme najťažší oriešok. Robil to kolega a skoro ho to doviedlo do zúfalstva. S dostupnými súčiastkami mu to stále nevychádzalo v rámci požiadaviek na takéto zariadenia. Nakoniec nám pomohol nápad, použiť teplotné cyklovanie súčiastok pred osádzaním plošných spojov, aby sme simulovali „umelé starnutie“ a odhalili nespoľahlivé komponenty. Týždeň pri +150 stupňov celzia, potom týždeň pri -50 stupňov celzia a to opakovanie. Tým sme zvýšili spoľahlivosť súčiastok, ktoré to vydržali. Tam už bola istota vyššej spoľahlivosti. Väčšina IO to vydržala. Bolo to v rámci skladovacích teplôt.

4. Riešila sa otázka vyžarovania zo skrine šifrátora. Tá musela mať špeciálnu konštrukciu. Bola to rebrovaná čierna skriňa s víkami podobnými a skrutkovanými podobne ako príruby vodovodných armatúr a so špeciálnymi kovovými tesneniami. Totižto vyžarovanie siahalo vyššími harmonickými frekvenciami vysoko do oblasti mikrovĺn. Len silné priskrutkovanie pomáhalo doslova a do písmena utesniť unikanie elektromagnetického vyžarovania. Museli sme podľa predpisu robiť merania vyžarovania a dostať sa do normy. Úporná drina to bola. Chodilo sa merať do Faradayovej klietky na Záhorie na pracovisku nášho výskumného ústavu. Bolo to ďaleko v lese, kde bol nízky elektromagnetický smog, ktorý mohol inde skresľovať merania. Dokonca vlákno žiarovky rušilo merania. Zvládli sme to nielen takto špeciálne konštruovanou skriňou, ale aj filtrami na komunikačnom rozhraní šifrátora.

Celý vývoj aj výroba bola zvládnutá za 1,5 roka. Samozrejme sa na tom pracovalo od nevidím do nevidím. Za tým stálo obrovské nadšenie. Ale úspech sa v bývalom Československu neodpúšťal. Konkurenciu sa nám snažila robiť Konštrukta Brno s ich obdobným šifrátorom Gejša. Tí na nej pracovali cca 2 roky. Začali tvrdiť, že to čo sme zrealizovali my, nemôže byť v poriadku a že niečo obdobné sa za takú krátku dobu nemôže poradiť zrealizovať. A tak sme ich pozvali do nášho výskumného ústavu na exkurziu. V rámci nej sme sa aj čosi dozvedeli o ich šifrátore Gejša. Nakoniec museli náš úspech uznať. A odvtedy už náš šifrátor PAŠA 5 nekritizovali. Ono totižto všetky obvody ich šifrátora realizovali z diskrétnych IO, čo zvýšilo pracnosť ako návrhu obvodov, tak aj čas na oživovanie, testovanie a výrobu prototypov. Oni mikroprocesorový systém na báze I8080 požili len na testovacie a kontrolné účely v tom ich šifrátore. Aj to mierne komplikovalo riešenie obvodov a hlavne zdroja. Pokiaľ my sme používali na napájanie väčšiny IO len 5V napájanie, s výnimkou 12 voltových kryptoschém ale s malým odberom, Oni I8080 systém museli napájať až tromi napájacími napätiami a to -5V, 5V a 12V. A komplikovanejší zdroj určite aj viac vyžaroval elektromagnetického žiarenia, nakoľko aj spotreba takých IO bola vyššia. Aj pracovníci FMV spočiatku tvrdili že sme predbehli dobu, pretože aj oni najprv

nechceli použiť mikroprocesor. To boli začiatky a z ich strany bola veľká nedôvera voči softvérovej realizácii väčšiny funkcií šifrátora. Ale presvedčili sme ich.



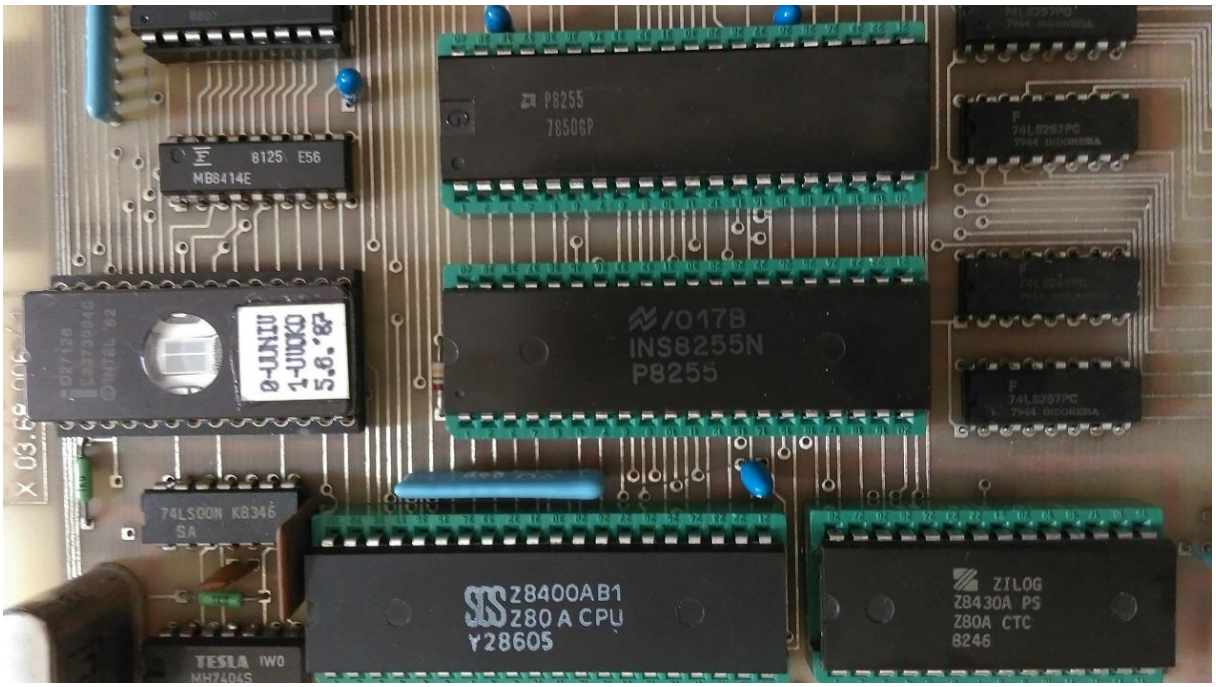
Obr. 1: Hlavná riadiaca doska šifrátora PAŠA 5 s mikroprocesorom, podpornými obvodmi, komunikačným interface a V/V bránami pre kryptoschémy – len funkčný prototyp

Pôvodne na Pražskom pracovisku jeden náš kolega začínal s úlohou FÉNIX na klasických IO, ale nezvládol to. Tak to potom preložili k nám DO LM. Výskumno-vývojovú úlohu na šifrátore PAŠA 5 sme nakoniec zvládli a začlenili sme ho spolu s linkovou výbavou, ktorú realizoval náš kolega z vedľajšieho úseku, do hlavného systému s digitalizáciou hlasu, ktorý realizovali kolegovia z Pražského pracoviska. Funkčnosť šifrátora a splnenie podmienok vývoja boli zverifikované na ***ÚŠO*** FMV.

Na pamiatku zostala hlavná riadiaca doska s mikroprocesorom a interface, demonštračne osadená, ako prototyp, ale funkčná. Na koncovo vyrobených doskách samozrejme pätice pod hlavnými IO, z dôvodu dosiahnutia vysokej spoľahlivosti, neboli osadzované.

Na obr. 1 je zobrazená riadiaca doska šifrátora PAŠA 5. V pamäti EPROM je riadiaci program celého šifrátora.

Na obr. 2 je zobrazený detail z riadiacej dosky šifrátora PAŠA 5 s čipmi hlavných obvodov a pamäťou EPROM s firmvérom.



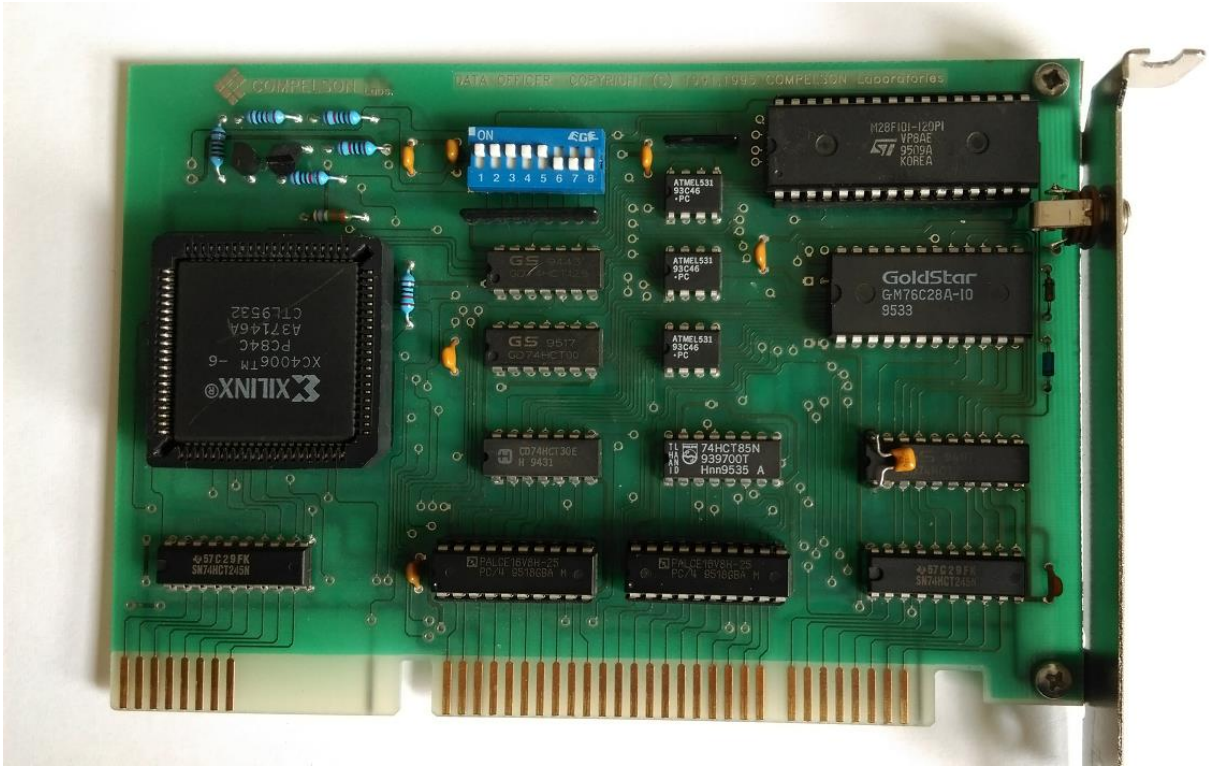
Obr. 2: Detailný pohľad na mikroprocesor Z80A CPU, čítač-časovač Z80A CTC, V/V brány typu I8255 pre kryptoschémy a pamäť EPROM 27128 s riadiacom programom

8.1.2 Prídavná ISA karta systému DATA OFFICER s programovateľným poľom XILINX pre šifrovací systém DONcrypt

Ďalším príkladom je naša implementácia algoritmu SEA64 do programovateľného poľa XILINX na bezpečnostnej karte DATA OFFECER firmy COMPELSON s ktorou sme spolupracovali v roku 1995. Karta realizovala podporu pre šifrovanie dát v reálnom čase. Bolo to za éry používania OS Microsoft MSDOS. Táto ISA karta tvorila z našej strany podporu pre našu aplikáciu DONcrypt určenú na šifrovanie dát na harddiskoch a v počítačovej sieti. Systém bol vyvinutý pre použitie v ozbrojených silách na Generálnom štábe Ministerstva Obrany SR a bol použiteľný až do stupňa utajenia „T“.

Na obr. 3 je zobrazená doska prídavnej karty ISA do PC typu AT s obvodom XILINX. Na obr. 4 je zobrazený detail z tejto karty s poľom XILINX.

Karta mala implementovanú ďalšiu bezpečnostnú funkcionalitu určenú pre ochranu pracovných PC typu AT, realizovanú českou partnerskou firmou COMPELSON.



Obr. 3: Prídavná karta DATA OFFICER s podporou šifrovania na báze hardvérovej implementácie šifrovacieho algoritmu SEA64 do programovateľného poľa XILINX

Obr. 4: Detail z prídavnej ISA karty s poľom XILINX

8.1.3 Kufríkový šifrátor KATKA na prípravu a šifrovanie textov

Toto zariadenie vzniklo v roku 1989 z pôvodne pripravovaného vreckového kódovacieho zariadenia požadovaného vojenskými spravodajcami. Názov KATKA vznikol z „Kapesný tlačítkový kódovací aparát“. Avšak my ako šifranti-konštruktéri sme presadili, že kódovaním sa nezaoberáme a teda budeme vyvíjať šifrátor. Podaril sa nám husársky kúsok tým, že sme presvedčili jednak ***ÚŠO*** na FMV a jednak vedenie VÚ060, že vyvinieme niečo lepšie na báze šifrovania. Zariadenie KATKA bolo prenosné v kufríku a tiež pripojiteľné k PC cez sériové alebo paralelné rozhranie. Taktiež obsahovalo na základnej doske modem pre pripojenie k telefónnym linkám alebo k rádiostanici. Na komunikáciu som vyvinul vlastný komunikačný protokol MPX. Okrem toho bolo zariadenie pripojiteľné aj k diaľnopisným linkám alebo diaľnopisu pomocou prúdovej slučky. V ďalšom kufríku bola prenosná tlačiareň pripojiteľná k zariadeniu KATKA cez LPT rozhranie. Zariadenie bolo použiteľné až do stupňa „T“. Funkčnosť zariadenia a splnenie podmienok vývoja boli zverifikované na ***ÚŠO*** FMV.

Pri vývoji padla voľba na mikroprocesorový systém ZILOG Z80. Šifrovací algoritmus SEA64 bol realizovaný softvérovo vo firmvéri. Zadanie na vývoj softvéru vypracovali kryptológovia z FMV. Strojový binárny kód algoritmu bol naprogramovaný vo dvojici EPROM pamätí, ktoré boli na plošnom spoji realizovanom vo forme zásuvného modulu. Tento modul bol mimo prevádzky ukladaný do trezora. Tak, ako to určoval v tej dobe ***ÚŠO*** FMV.

Návrh obvodového riešenia som realizoval okrem modemu ja a taktiež okrem testov som firmvér vyvíjal v assembleri mikroprocesora Z80. Testy vyvinul kolega. Napájanie bolo možné buď zo sieťového adaptéra alebo z palubnej siete. V skutočnosti sme výskum aj vývoj prototypov zvládli piati ľudia. Na vývoj a výrobu v prototypových dielňach VÚ060 LM bolo v skutočnosti písaných až 80 ľudí. Po skončení vývoja, výroby a odovzdaní príslušného počtu vyrobených kusov ozbrojeným zložkám som pokračoval na vývoji systému KATEST pre servisné účely opráv a testov zariadenia KATKA. To fungovalo na platforme PC AT s prídavnou ISA kartou pripojiteľnou cez kábel k testovaciemu servisnému konektoru KATKy. Testovací softvér bežal pod OS Microsoft MSDOS. Rozhranie programu bolo grafické s možnosťou emulácie mikroprocesora a krokovania jeho softvéru. Aj tento softvér KATESTu som realizoval v assembleri.

Jeden kus zariadenia KATKA je umiestnený v priestoroch Bezpečnostného Úradu Ministrerstva Obrany SR v Novom Meste nad Váhom, v múzeu šifrovacej techniky. Po delení majetku bývalého Československa sa jednotlivé kusy zariadení delili v pomere 1:2. Po 2 kusy do Česka.

8.1.4 Sieťový šifrátor IPcrypt QS32

Toto zariadenie vznikalo po roku 2002 ako šifrovací automat na šifrovanie sieťovej prevádzky s utajením na stupeň „D“. Šifrovanie je vykonávané bez akejkoľvek súbežnosti s nešifrovanou prevádzkou. Šifrujú sa dátové obsahy paketov pre všetky IP adresy aj porty. Preto šifrátor obsahuje dva sieťové adaptéry. Jeden pre červenú stranu s otvorenou sieťovou prevádzkou v chránených priestoroch a druhý pre čiernu stranu so šifrovanou sieťovou prevádzkou vonkajšej siete WAN napríklad aj vládnej siete GOVNET. Oddelenie sietí je realizované bez akéhokoľvek súbehu týchto sietí, striktným hardvérovo realizovaným oddelením. Ako riadiaci systém je použitý systém Windows XP Embedded bežiaci na

základnej doske priemyselného počítača Nova-8522-G s procesorom Intel Celeron. Ako celok je to hardvérový šifrátor pre rýchlosť toku dát 100 megabitov/sec.

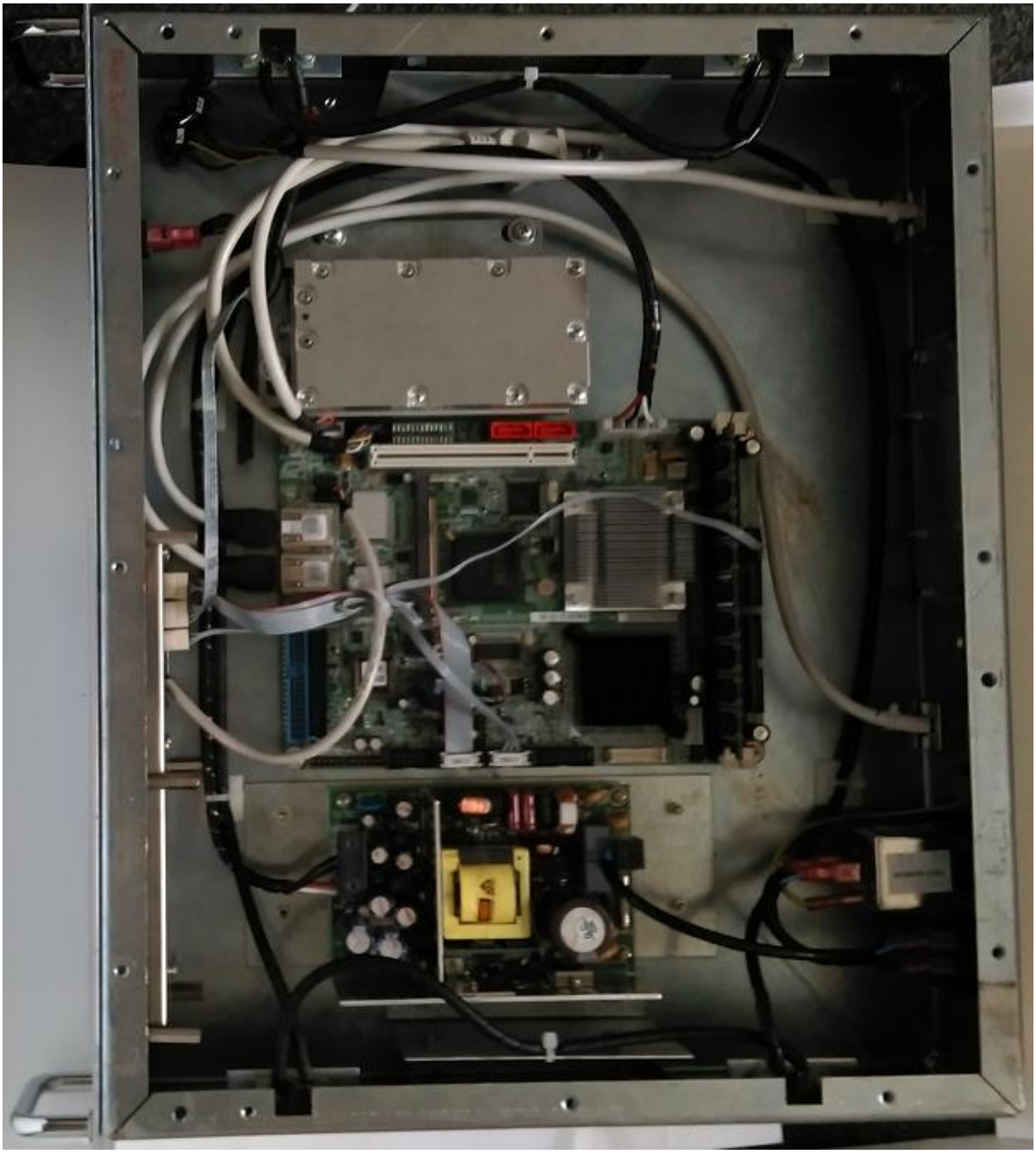
Šifrovanie toku dát realizuje na základnej doske PCI prídavná karta CODESTAR 2 DSP s algoritmom SEA64. Algoritmus je spracovávaný firmvérom signálového digitálneho procesora TMS320C6205. Rýchlosť spracovania algoritmu SEA64 plne postačuje pre 100 megabitové siete. Samozrejme, pokiaľ sa použije tento algoritmus. AES256 je tak pomalý, že je potom šifrátor pod hranicou použiteľnosti. Hoci AES256 nie je vôbec silnejší ako SEA64. A práve na nanútený prechod na AES256 podľa ***BŠ*** v čase vývoja sme doplatili a takýto šifrátor by bol nepoužiteľný. Projekt bol preto zastavený. Dokonca aj priamo Celeron na tejto doske priemyselného PC vykonáva AES256 úžasne pomaly. AES256 bol pomalý aj na procesoroch PENTIUM a pre šifrovanie sieťovej prevádzky podľa nariadení ***BŠ*** bol na hranici použiteľnosti. Vysvetlenie bude popísané neskôr. To čo bolo popísané o AES256 už predtým, navodzuje úvahu, prečo je to tak. A tak šifrátor zostal s kartou CODESTAR v ktorej je implementovaný algoritmus SEA64. A konštrukcia hardvéru zodpovedá utajeniu dokonca na stupeň utajenia „D“. Na obr. 5 je pohľad na modul šifrátora.

Obr. 5: Rack modul šifrátora IPcrypt QS32 – pohľad spredu

Na obr. 6 je ilustračný pohľad do odkrytého šifrátora. Jadrom je riadiaca hlavná doska priemyselného počítača NOVA-8522-G. Vedľa nej je ochranný blok PCI karty CODESTAR 2 DSP, ktorá nie je v šifrátore na fotografi momentálne práve osadená aby bol viditeľný PCI slot základnej dosky. Po stranách skrine v hornej časti sú umiestnené mikrospínače pre detekciu odkrytia horného víka šifrátora. Na pravom boku skrine sa nachádza napájací zdroj.

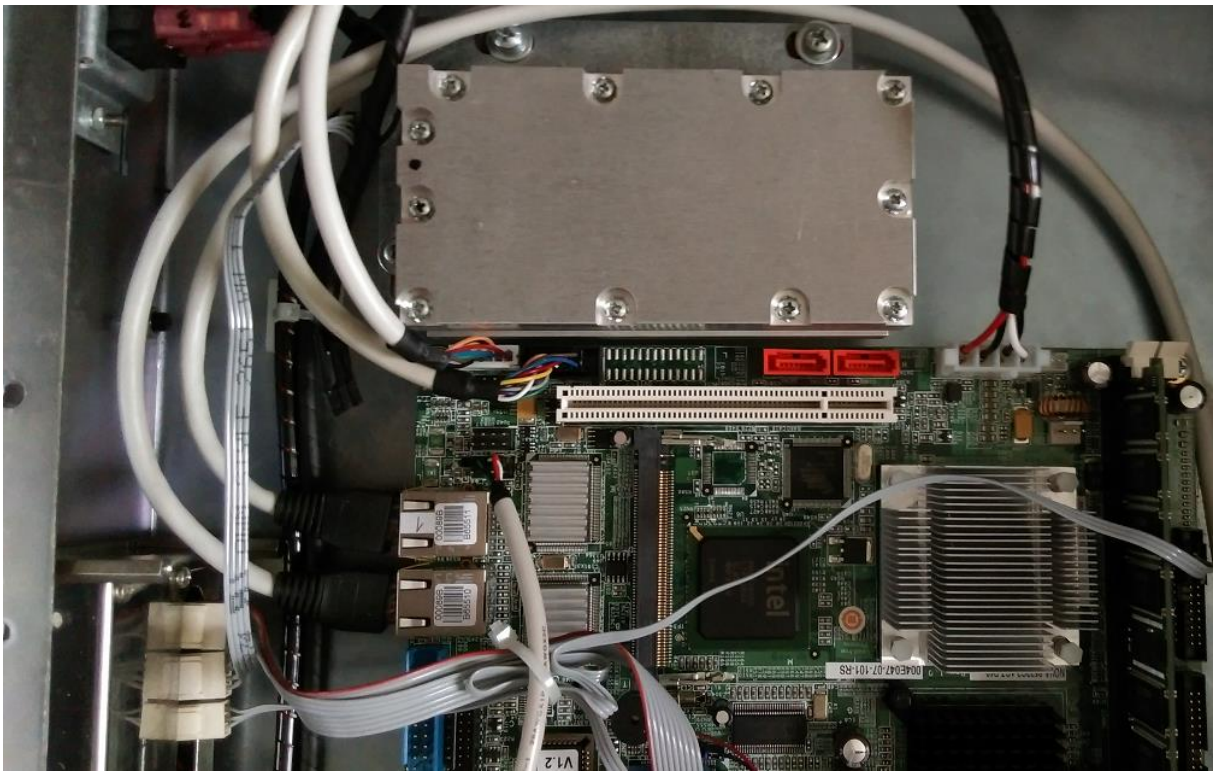
Na obr. 7 je detail s ochranným blokom PCI karty CODESTAR 2 DSP. Ten je vyfrézovaný z hliníka. Tvorí ochranu jednak proti elektromagnetickému vyžarovaniu smerom von z PCI modulu a tiež ochranu proti rušivému vyžarovaniu z hlavnej dosky šifrátora. Vyžarovanie smerom von z PCI karty by tvorilo nežiadúci kanál pre bočné informácie z DSP počas procesu šifrovania. PCI karta má realizované aj ďalšie stupne ochrany o ktorých bude pojednané neskôr.

Ochranný blok má zároveň aj úlohu zabrániť naindukovaniu rušenia do ***šumátora*** na PCI karte počas jeho činnosti. O jeho funkcii bude pojednané neskôr pri popise karty CODETSR 2 DSP.

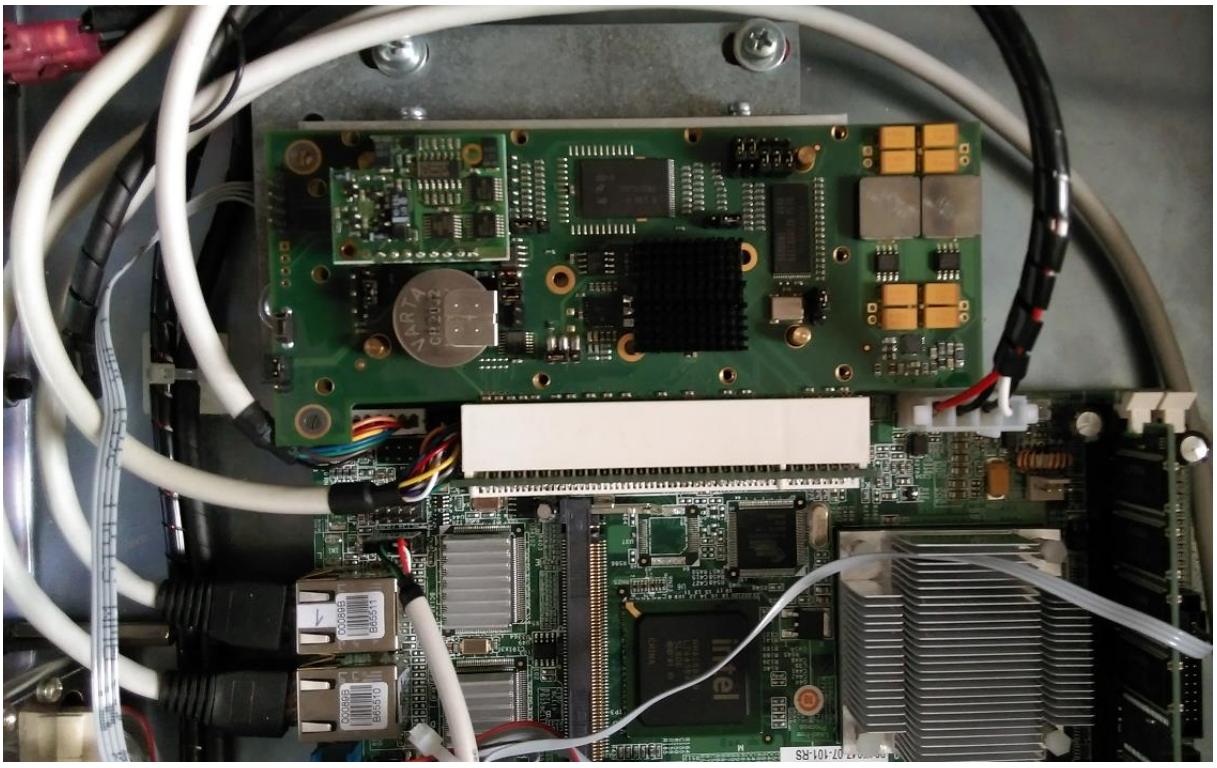


Obr. 6: Pohľad do odkrytého šifrátora IPcrypt QS32

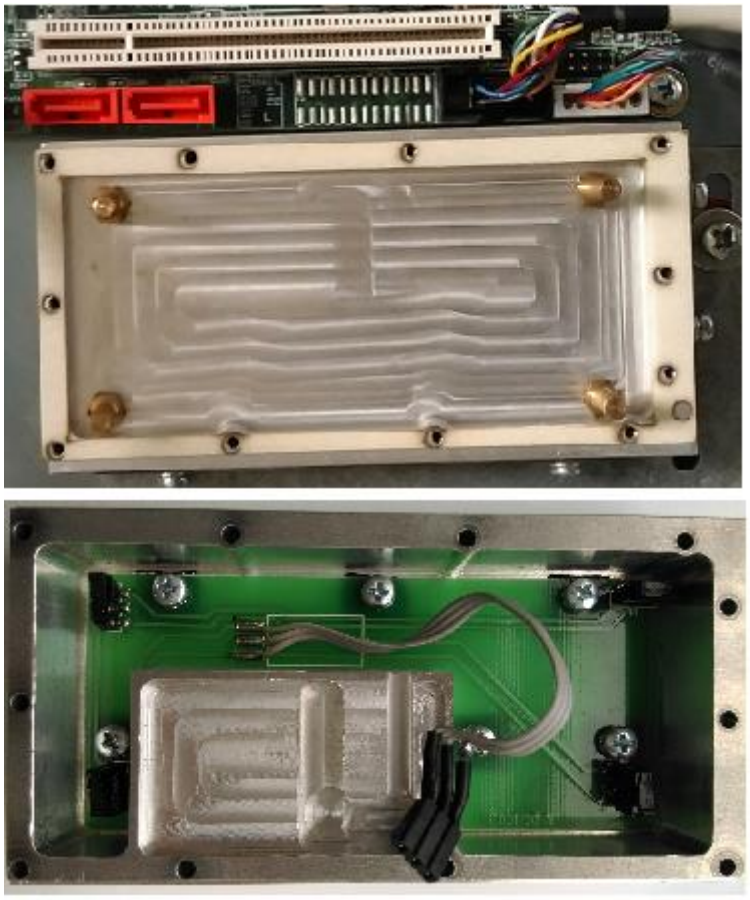
Na obr. 8 je pohľad na otvorený ochranný blok s kartou CODESTAR 2 DSP pripojenou k PCI slotu cez rohovú riser kartu. Horné víko bloku je odskrutkované. Detaily obidvoch polovíc bloku sú zobrazené na obr. 9. Na spodnom víku bloku sú viditeľné 4 kolíky, ktoré pri zaskrutkovanom hornom víku tlačia na 4 ochranné mikrospínače, ktoré majú za úlohu detekovať pracovný stav karty a jej uzatvorenia v tomto bloku. Pri pokuse o odkrytovanie sa vymažú kryptografické prvky z CMOS RAM jej odpojením od zálohovacieho napájania a vyskratovaním prívodov tohto napájania k čipu.



Obr. 7: Pohľad na ochranný blok karty CODESTAR 2 DSP bez vloženej PCI karty



Obr. 8: Otvorený ochranný blok s kartou CODESTAR 2 DSP pripojenou k PCI slotu cez rohovú riser kartu



Obr. 9: Detaily ochranného bloku karty CODESTAR 2 DSP. V hornej časti obrázku je spodná časť priskrutkovaná k spodnému víku šifrátora hneď vedľa základnej dosky tak, aby do jej PCI slotu zasunutá rohová riser karta poskytla konektor PCI slotu pre kartu CODESTAR 2 DSP.

Na hornom víku je viditeľná vyfrézovaná časť subkrytu pre generátor šumu, bežne nazývanému „***šumátor***“. Obsahuje aj plošný spoj s umiestnením 4-och mikrospínačov a na obrázku viditeľný káblik s konektorom pripojiteľným ku karte CODESTAR 2 DSP.

*Obr. 10: Zadný panel šifrátora so sieťovými konektormi RJ45 pre červenú (otvorené nešifrované dáta) a čiernu (šifrované dáta) stranu siete podľa **BŠ**.*

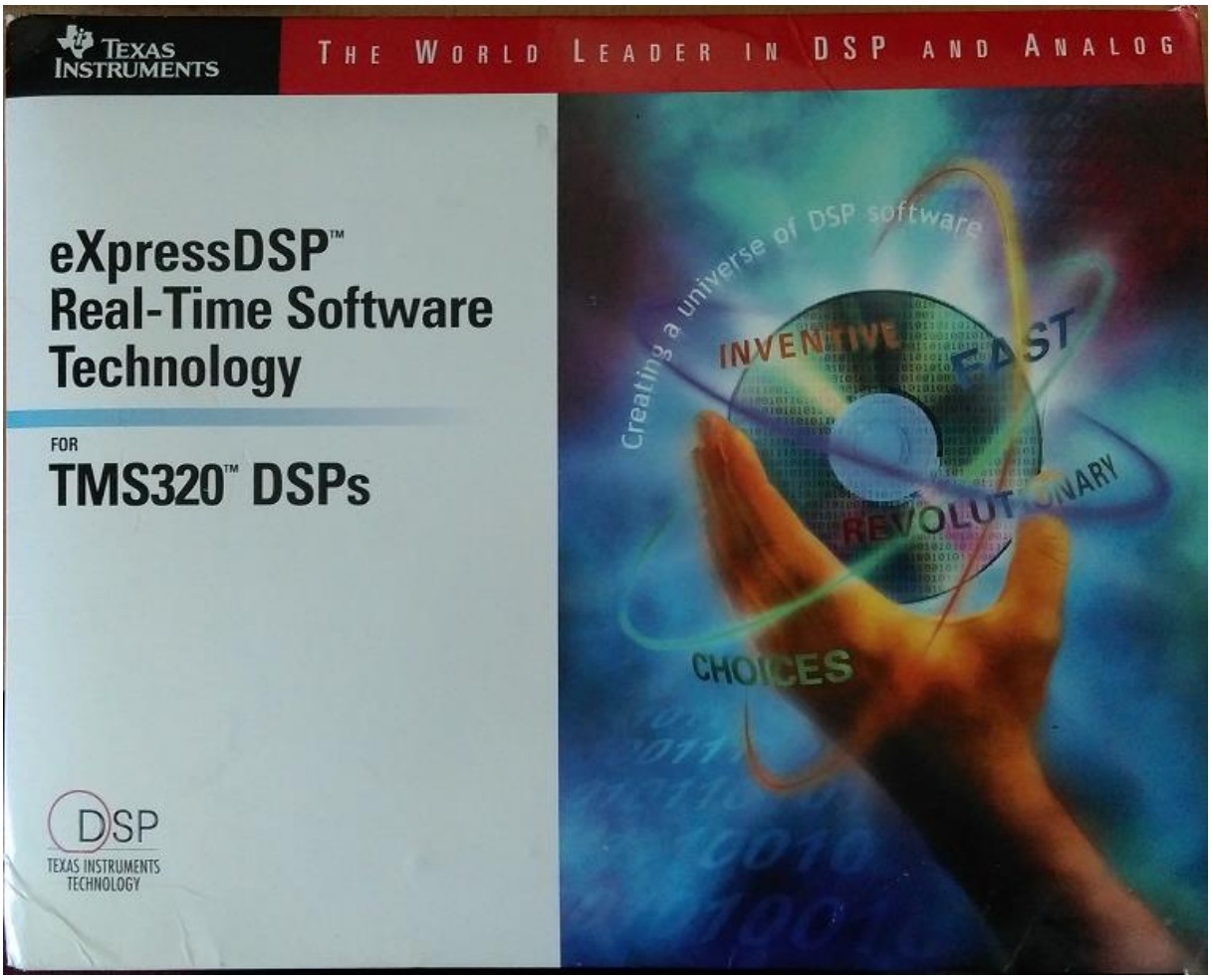
9. Vývoj kryptografických prídavných PCI kariet CODESTAR

9.1 PCI karta CODESTAR – vývojový kit

Tieto PCI karty sa vyvíjali a zdokonaľovali za účelom poskytovania šifrovania v hardvéri algoritmom SEA64, generovania kvalitnej náhodnej postupnosti na báze nedeterministického fyzikálneho generátora, poskytovania chráneného úložišťa kryptografických prvkov, ako sú kľúče, objekty asymetrickej kryptografie a realizácie ďalších bezpečnostných prvkov. Karta je určená do počítačov typu PC a tiež aj do serverov na báze PC so zbernicami typu štandardného PCI komunikačného rozhrania s operačnými systémami Windows alebo Linux. Pre príslušný OS musí byť použitá aj vlastná sada ovládačov tejto karty. My sme sa zamerali na OS Windows pre PC aj servery. Táto PCI karta, už aj v najnovšom prevedení, je kľúčovým prvkom v ***CBA*** pre naše bezpečnostné aplikácie. Posledná verzia CODESTAR 4 DSP má vybudovanú podporu okrem predchádzajúcich verzií OS Windows aj pre Windows 11 na PC aj serveroch. Sada ovládačov s väzbou na hardvér PCI karty, je na základe výsledkov úspešne vykonaných laboratórnych testov scertifikovaná spoločnosťou Microsoft pre OS Windows 10 aj OS Windows 11. Vývoj začal v roku 2001. Na obr.11 je prvý laboratórny kit určený na otestovanie signálového procesora firmy TEXAS. Na tejto PCI karte sa otestovali a začali sa vyvíjať časti programového vybavenia firmvéru. Začínalo sa na klasickom PC s procesorom Pentium a pod OS Windows 98. Tam tvoril základ ešte aj MSDOS, pod ktorým sa v RINGu 3 dalo ešte komunikovať s hardvérom. Tak sa odskúšala komunikácia cez PCI rozhranie pomocou magickej I/O adresy „0xC8“, najprv nájdenie zariadenia DSP na PCI zbernici podľa jeho DEVICE ID a VENDOR ID a následne základné PCI transakcie. Potom som prešiel na Windows NT 4.0, kde začal vznikať aj základný RING 0 kernel modul „Ism.sys“.

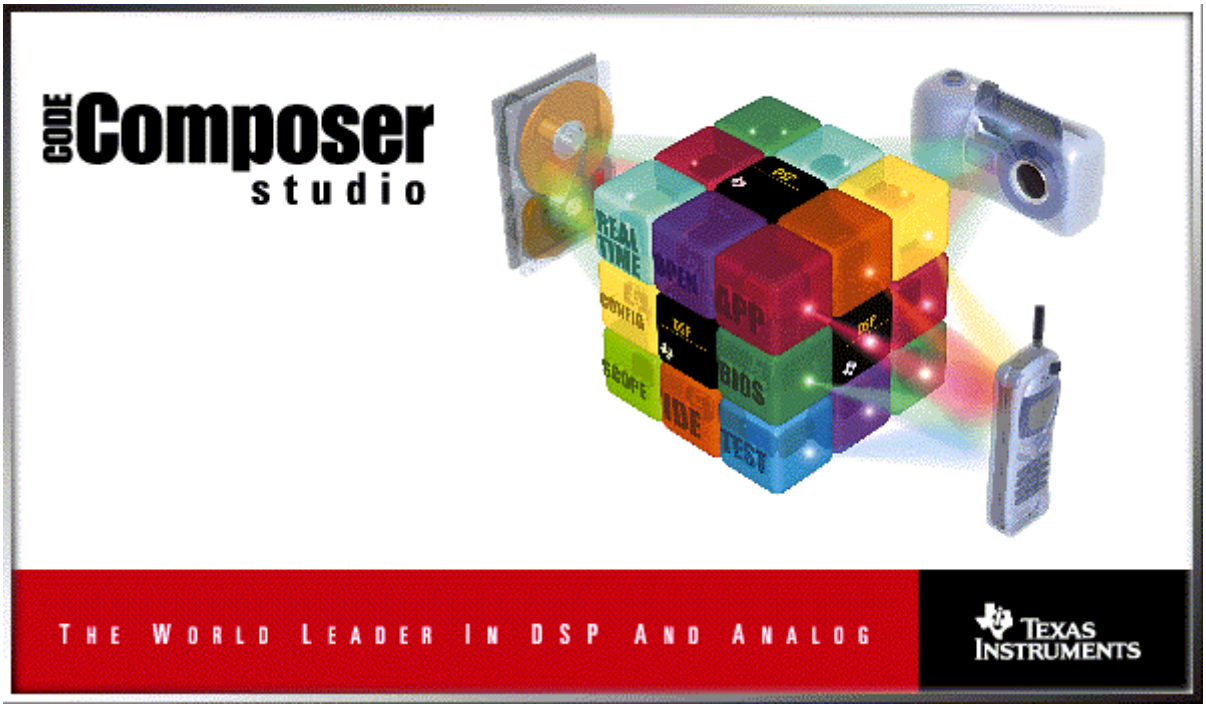
*Obr. 11: Laboratórny kit prvej PCI karty CODESTAR s konektormi rozhrania určenými na ladenie a testy. Pod chladičom je DSP, vpravo je spínaný zdroj, ktorý z bezpečnostných dôvodov oddeľuje napájanie od napájania zbernice PCI a zabezpečuje dodržiavanie zásad stanovených **BŠ**.*

Na tomto laboratórnom DSP kite som začal aj s implementáciou šifrovacieho algoritmu SEA64 v assembleri procesora TMS320C6205. Ostatná časť firmvéru je vyvíjaná v jazyku „C“. Hardvér bol vyvíjaný v spolupráci so ***SAV*** Bratislava s pracoviskom na Dúbravskej ceste 9. Aj doska plošného spoja bola vyrobená a osadená v ***SAV***. Softvérové vybavenie na vývoj bolo legálne zakúpené s licenciou. Aj PCI-SIG (PCI Special Interest Group) licencia bola zakúpená a použitá legálne. Licencie zakúpila firma INFOSEC, s.r.o. Balík softvérových vývojových kitov pre DSP procesory TEXAS radu TMS320 zobrazuje obr. 12.

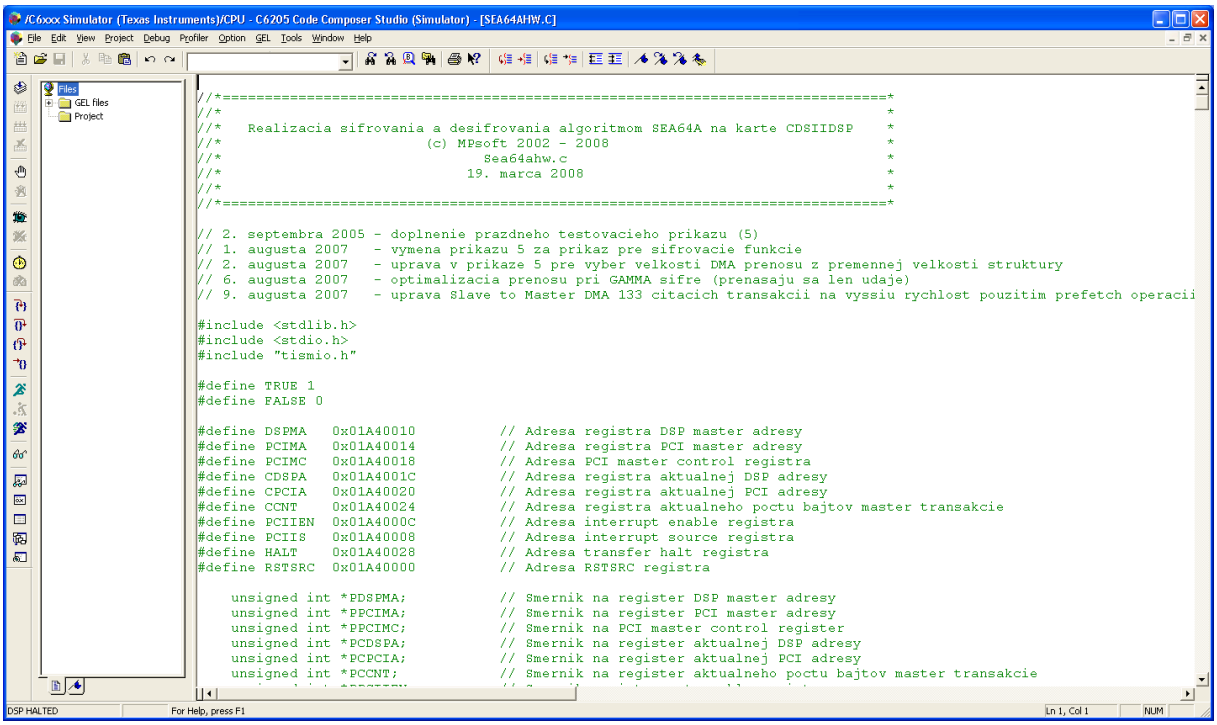


Obr. 12: Softvérové vývojové kity pre DSP procesory TEXAS radu TMS320

Balík obsahuje okrem rozsiahlej dokumentácie aj „***Code Composer studio***“, ako obdobu Microsoft Visual studia. Na obr. 13 je úvodné logo štartu Code Composer studia pre DSP procesory TEXAS radu TMS320. Okrem kompilátora jazyka „C“ a linkera, v štúdiu, kity obsahujú aj tzv. ***Turbo profiller***, ktorým sa optimalizuje výsledný kód assembleru vyprodukovaný kompilátorom jazyka „C“. Pretože vyšší jazyk nedokáže zoradiť strojové inštrukcie dosť optimmálne vzhľadom na ich paralelné spracovanie, čo bolo spomínané už v predchádzajúcich kapitolách. Okrem shellu firmvéru napísaného v jazyku „C“ som kód vlastného šifrovacieho algoritmu SEA64 napísal a optimalizoval priamo v assembleri DSP procesora. Jednak bol tak kód optimalizovaný čo do jeho veľkosti a jednak sa podarilo zvýšiť paralelizáciu strojových inštrukcií vykonávaných procesorom DSP. Na obr. 14 je zobrazený ako príklad úvod zdrojového kódu v jazyku „C“ shellu algoritmu SEA64 v textovom editore Code Composer studia pre DSP procesory TEXAS radu TMS320. Jadro algoritmu je napísané v assembleri a jeho binárny kód je linkovaný ku binárnemu kódu tohto modulu.



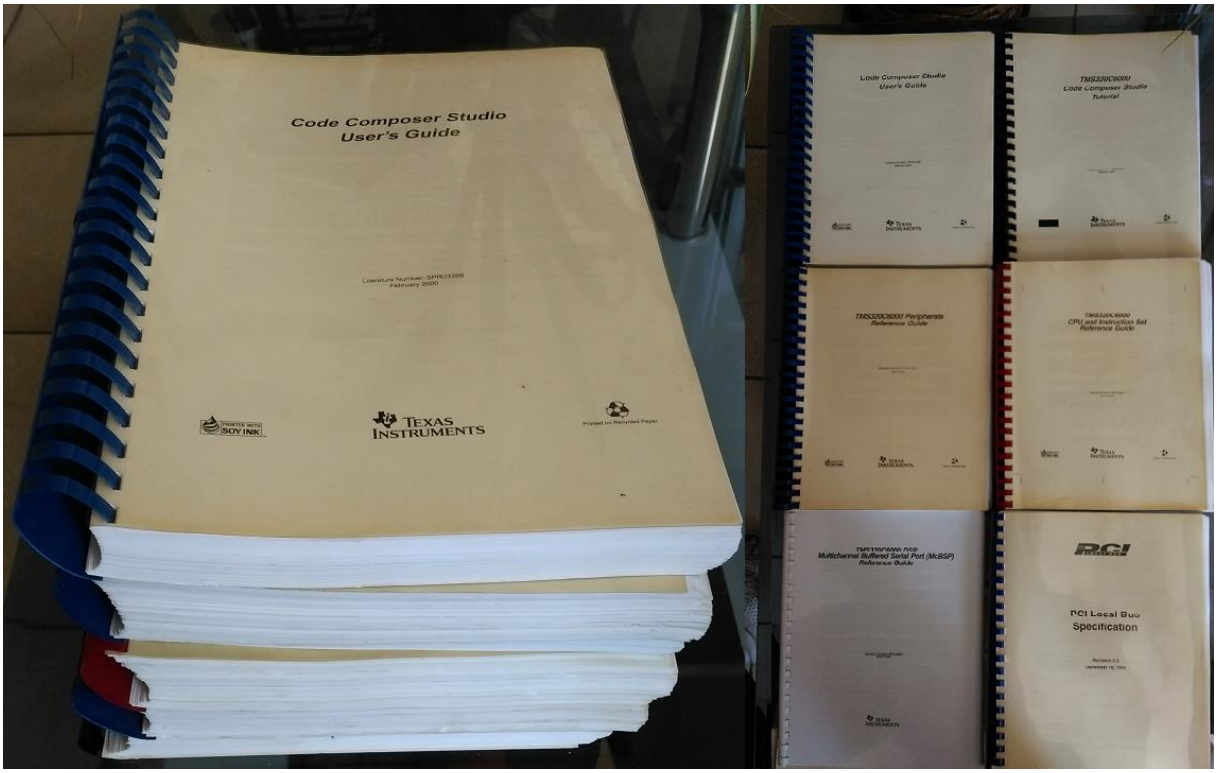
Obr. 13: Logo štartu Code Composer studia pre DSP procesory TEXAS radu TMS320. Toto vývojové štúdio bolo použité na vývoj firmvéru šifrátora s algoritmom SEA64 naprogramovaného v assembleri DSP procesora TMS320C6205.



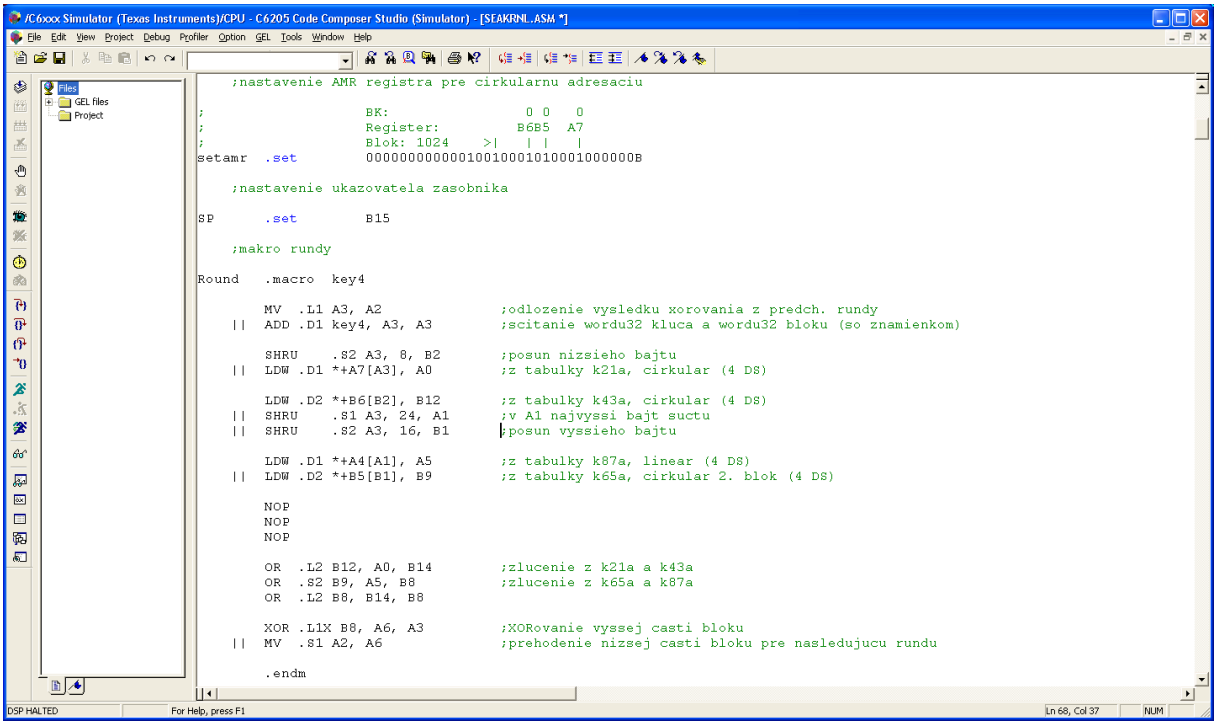
Obr. 14: Príklad zobrazenia zdrojového kódu algoritmu SEA64 v textovom editore Code Composer studia pre DSP procesory TEXAS radu TMS320

V predchádzajúcich kapitolách už bolo spomenuté, akú má tento DSP procesor architektúru a ako sa programuje. Je to digitálny signálový procesor s okolitou hardvérovou podporou pre PCI komunikáciu a hlavne s veľkým počtom programovateľných registrov v ktorých sa pomocou bitových nastavení konfiguruje táto hardvérová podpora DSP. PCI transakcie sú programované v rámci nastavenia PCI karty v režime „***Bus Master***“ a DMA prenosov s pomocou riadenia cez dualporty.

Periférne rozhrania tohto DSP sa programujú pomocou spomínaných registrov na spôsob programovateľných polí s logickými členmi so základnými funkciami (AND, NAND, OR, NOR, XOR…). Napríklad, v prípade potreby rozhrania typu UART, všetky jeho funkcie sa naprogramujú.



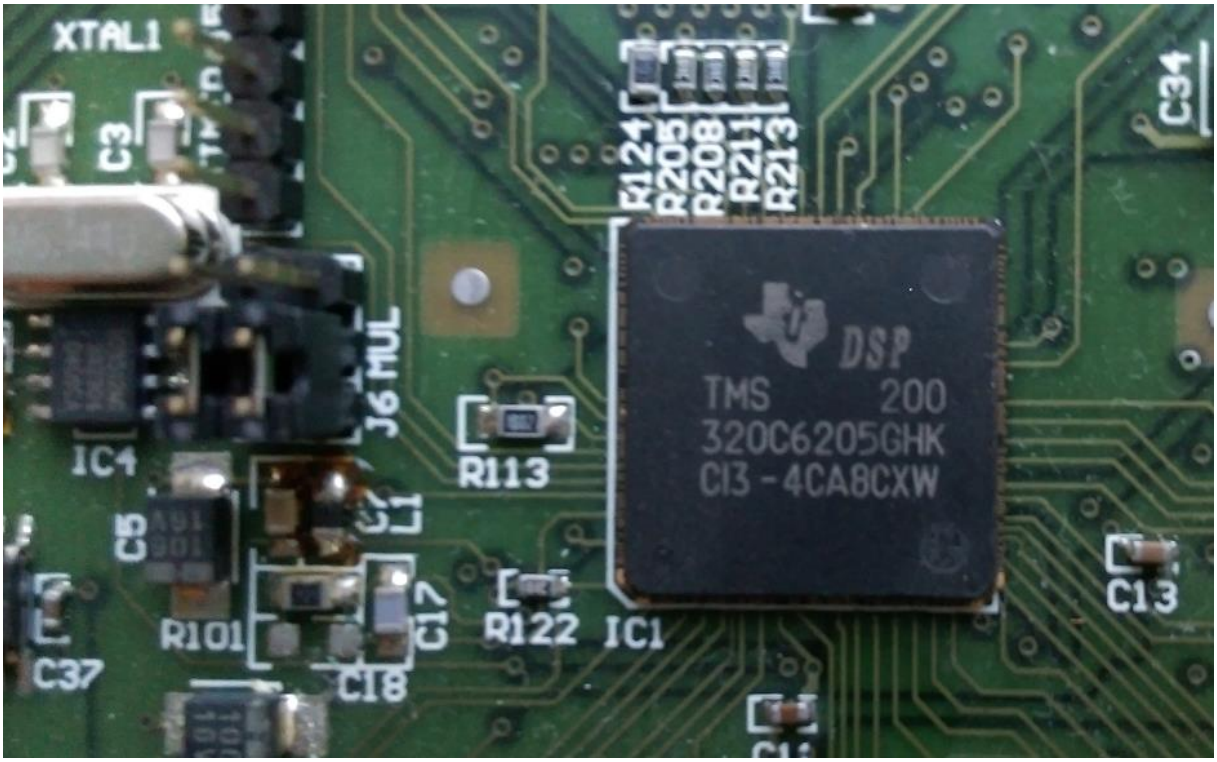
Obr. 15: Základná dokumentácia použitá na vývoj firmvéru pre PCI karty CODESTAR



Obr. 16: Ukážka programovania makroinštrukcie jednej rundy algoritmu SEA64 v assembleri DSP procesorov radu C6000 v Code Composer Studiu

Na vývoj softvéru pre tento DSP je potrebné naštudovať veľké množstvo dokumentácie dodávanej spoločnosťou TEXAS INSTRUMENTS. Taktiež všetko kolo PCI zberníc obsahuje špecifikácia „***PCI Local Bus***“, ktorú sme zakúpili v licencii, pozostáva z vyše 300 strán dôležitých informácií pre zvládnutie komunikácie cez PCI interface. Manuály základnej dokumentácie sú zobrazené na obr. 15.

Na obr. 16 je ukážka programovania makroinštrukcie jednej rundy algoritmu SEA64 v assembleri DSP procesorov radu C6000 v Code Composer Studiu. V zdrojovom kóde sa používa cirkulárna adresácia, čo podstatne zjednodušuje zdrojový kód a tým sa aj urýchľuje vykonávanie takýchto strojových inštrukcií procesora DSP. Avšak tak stúpajú aj nároky na logickú konštrukciu programu a dátovej časti S-boxov. Pomocou symbolov „||“ sa zoraďujú inštrukcie do jedného hodinového cyklu procesora. Vo výpise sú viditeľné zoradenia pre paralelné spracovanie dvoch až troch inštrukcií jedným hodinovým cyklom. Cirkulárnou adresáciou a súčasným paralelným zoradením inštrukcií do jedného cyklu sa drasticky zvyšuje výkon procesora. Kód SAE64 bol v assmbleri vysoko zoptimalizovaný. Tak procesor DSP zabezpečuje rýchlosť spracovania algoritmu SEA64 (šifrovanie/dešifrovanie) vysoko nad 100 megabitov/sec.

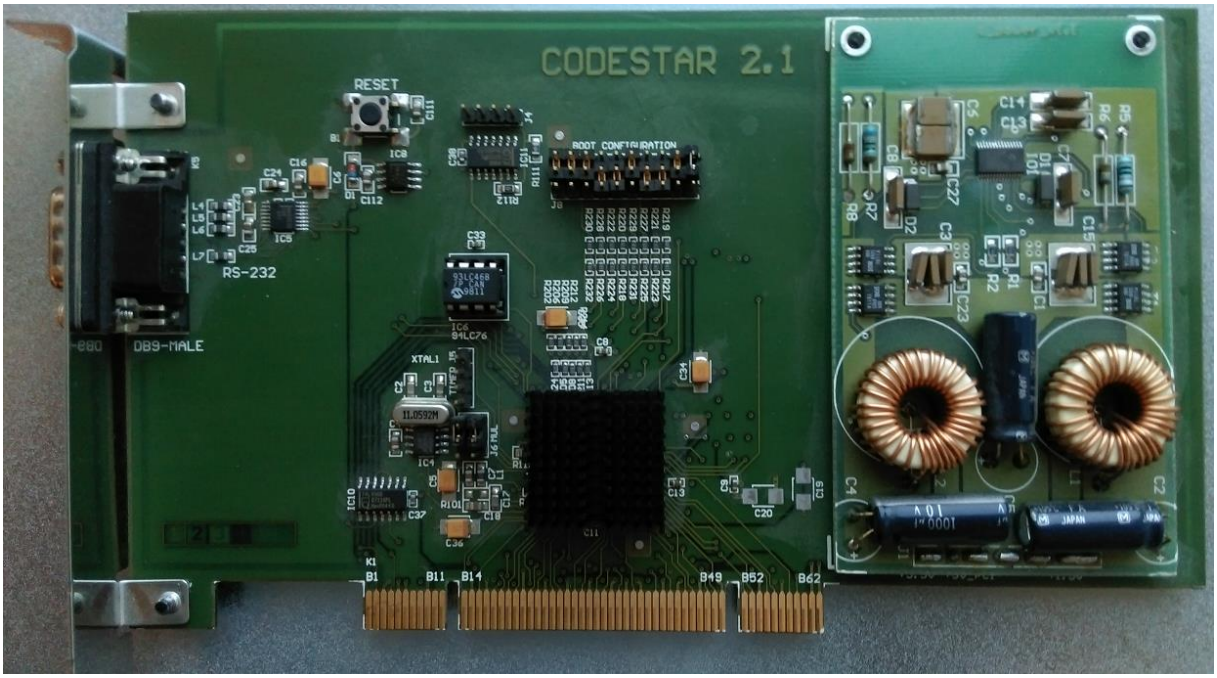


Obr. 17: DSP procesor TMS320C6205 v prevedení GHK bez chladiča na karte CODESTAR

Na obr 17. je pohľad na prispájkovaný DSP procesor na karte CODESTAR. Zatiaľ je bez chladiča, čo pre väčšinu hardvérových realizácií projektov postačuje ale pri jeho vyťažení na maximálny výkon, kedy vyvíja veľké teplo, nalepený chladič zvyšuje jeho životnosť a hlavne spoľahlivosť celého zariadenia. To je aj náš prípad. Pri vývoji nám už niekoľko procesorov zhorelo.

9.2 PCI karta CODESTAR 2.1 DSP

Na obr. 18 je pohľad na prototyp PCI karty CODESTAR 2.1. Táto karta nemá ešte integrovaný fyzikálny generátor šumu na generovanie náhodných údajov.



Obr. 18: PCI karta CODESTAR 2.1 DSP

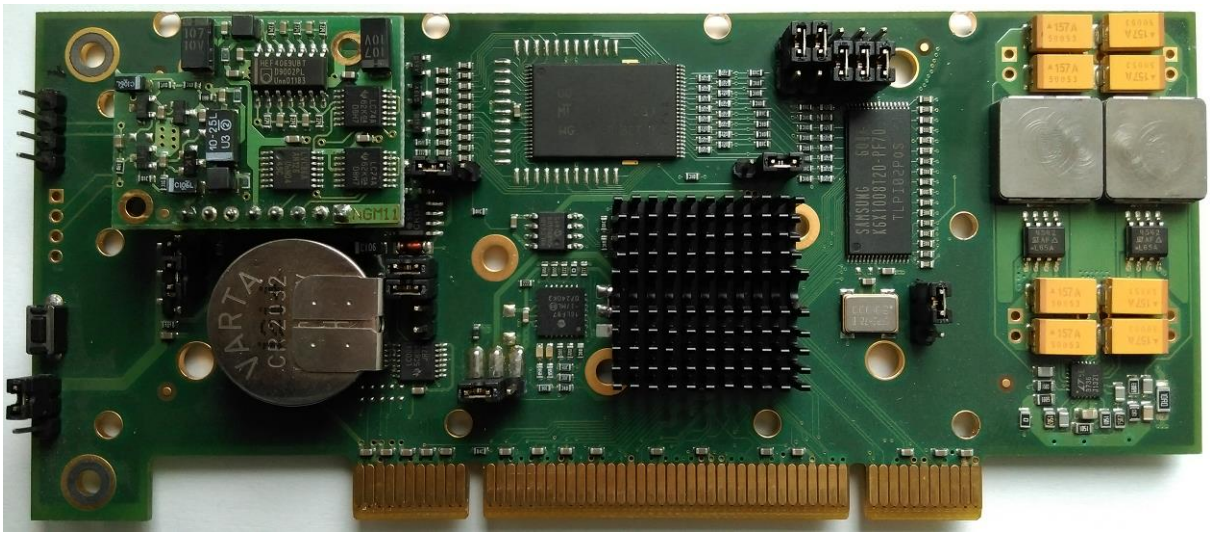
Karta poskytuje šifrovanie a dešifrovanie algoritmom SEA64. Tak ako aj jej nasledujúce ďalším vývojom zdokonalené verzie.

Do úvahy neprichádzalo použitie aj šifrovacieho algoritmu AES256, pretože jeho spracovanie procesorom DSP by nebolo dostatočne rýchle a priestor interných superrýchlych RAM procesora by nebol dostatočný na jeho programovú realizáciu

9.3 PCI karta CODESTAR 2.2 DSP

Na obr. 19 je pohľad na PCI kartu CODESTAR 2.2, čo je zdokonalená verzia. Táto karta má už integrovaný fyzikálny generátor šumu na generovanie náhodných údajov. Je to ten malý plošný spoj naľavo hore na základnej karte.

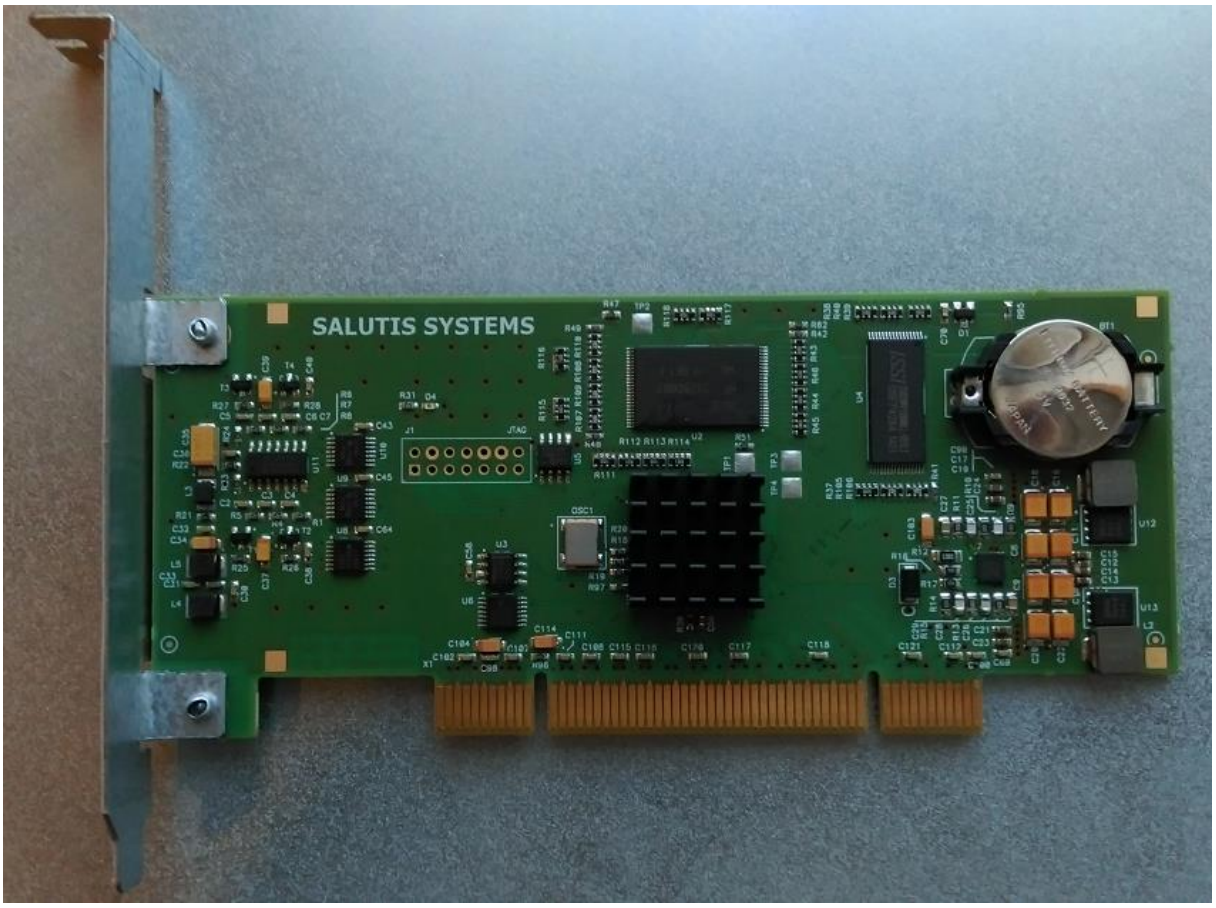
Má vylepšený napájací zdroj, pridanú CMOS RAM pamäť so zálohovaním batériou CR2032. Táto karta je určená do šifrátora IPcrypt QS32. Na obr. 8 je táto karta v šifrátore v otvorenom ochrannom bloku pripojená k PCI slotu hostiteľského systému cez rohovú riser kartu.



Obr. 19: PCI karta CODESTAR 2.2 DSP

9.4 PCI karta CODESTAR 4 DSP

Na obr. 20 je pohľad na PCI kartu CODESTAR 4, čo je finálna verzia CODESTAR, určená do PC s nainštalovanými aplikáciami Centrálnej Bezpečnostnej Autority pre viaceré naše bezpečnostné aplikácie.



Obr. 20: PCI karta CODESTAR 4 DSP - finálna verzia certifikovaná pre Windows 10 aj 11

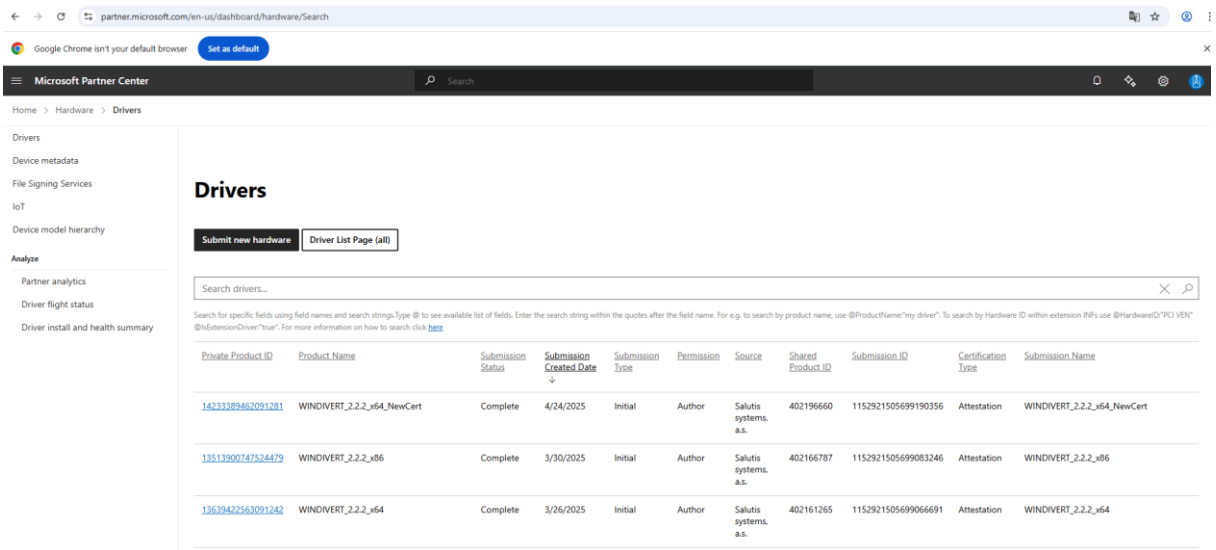
9.5 Softvérová podpora kariet CODESTAR DSP v hostiteľských PC

Komunikácia všetkých modelov kariet CODESTAR medzi hostiteľským PC a vlastnou kartou prebieha cez PCI zbernicu pomocou PCI transakcií. PCI transakcie sú inicializované z kernel modulu OS Windows. Beh transakcií je vykonávaný obidvomi smermi pomocou dualportov a DMA prenosov za účasti prerušovacieho systému. Jadro kernel modulu ovládača karty v hostiteľskom PC beží v RING 0 režime procesora. To sú I/O privilégiá cháneného systému procesora IOPL = 0. Priamy prístup z bežného používateľského režimu, ktorý beží v RING 3 režime procesora k hardvéru nie je možný. Preto sa nad jadrom v RING 0 používajú knižnice funkcií v RING 3 s volaniami pomocou IOCTL riadiacich kódov.

Ovládač karty CODESTAR je realizovaný modulom „***Ism.sys***“, ktorého funkcie sú volané pomocou knižnice „***Ismx.dll***“, prípadne ďalším používateľským softvérom, pre servis a diagnostiku. Cez funkcie tejto knižnice sú potom prístupné kryptografické funkcie PCI karty.

Vývoj modulu ovládača karty „***Ism.sys***“ bol zameraný v poslednej dobe na OS Windows 10 a Windows 11. Špeciálne pre Windows 11 sú laboratórne testy na certifikáciu, požadované Microsoftom, zamerané na vysoký stupeň bezpečnosti ako hardvéru, tak aj softvéru. A preto aj nároky na vývoj hardvéru pre Windows 11 sú nesmierne vysoké. Pod hardvérom Microsoft myslí ako vlastný hardvér, tak aj RIGN 0 softvér ovládačov.

Laboratórne testy musí vývojár vykonať sám s prostriedkami popísanými v predchádzajúcich kapitolách. Na to sa použije Microsoft HLK štúdio. Testy prebiehajú na svetovej úrovni s konkurencieschopnosťou v rámci výsledkov testov podobných zariadení z celého sveta. Len po 100 percentnom úspechu testov je v HLK štúdiu vygenerovaný výsledkový súbor „***xxx.hlkx***“, kde „xxx“ je názov testovaného hardvérového projektu. Ten sa potom zasiela ako „submission“ do partnerského cloudu Microsoftu, obr. 21.



Obr. 21: Microsoft Partner Center pre zaslanie výsledkov laboratórnych testov aj s balíkom celého testovaného projektu ovládačov

14293126337217699	CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 21H2	Complete	5/26/2023	Initial	Author	Salutis systems, a.s.	401349563	1152921505696343889	HLK	CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 21H2
14598742574858851	CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 22H2	Complete	5/26/2023	Initial	Author	Salutis systems, a.s.	401349393	1152921505696343341	HLK	CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 22H2

Obr. 22: Certifikačné procesy ovládačov CODESTAR 4 DSP pre Windows 11 verzií 21H2 a 22H2

Na obr. 22 sú úspešne vykonané HLK certifikácie ovládačov CODESTAR 4 DSP pre Windows 11 verzií 21H2 a 22H2 ktoré ale v rámci kompatibility budú fungovať aj vo vyšších verziách Windows 11.

CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 22H2

Shared product ID: 401349393
Private product ID: 14598742574858851

Package Acceptance — Preparation — Scanning — Validation — Catalog creation — Manual review — Sign — Finalize

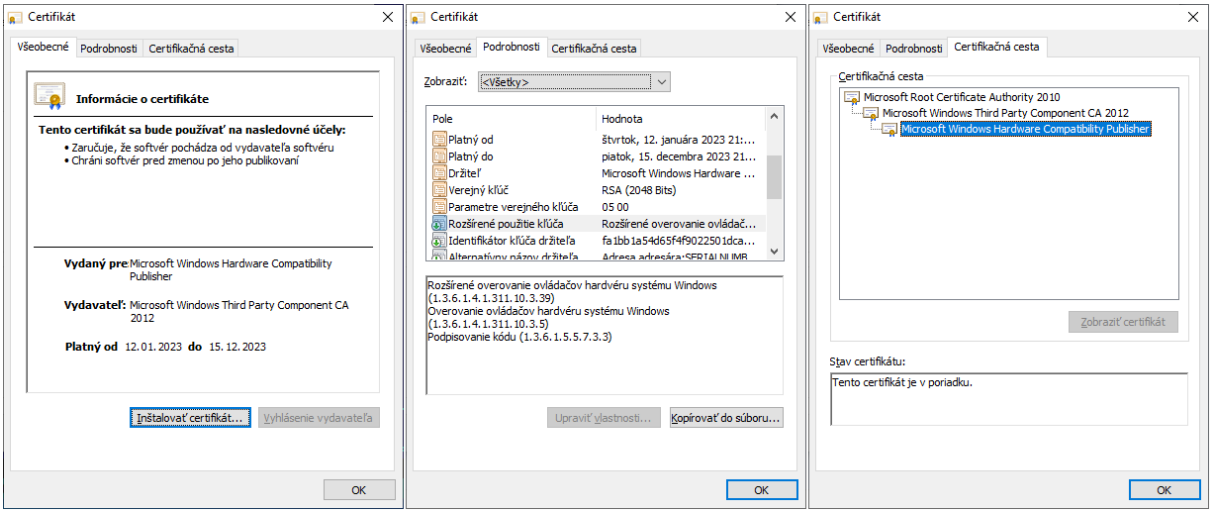
Your submission is certified! Your signed packages are available below, where you can add DUA packages and create shipping labels. Any shipping labels you created earlier are processing.

Packages and signing properties

Upload new | Download DUA shell

Name	Date	Submission Id	Status	Created By	
CODESTAR 4 DSP PCI CARD FOR WINDOWS 11 22H2 - Initial	5/26/2023	1152921505696343341	Finalize	MilanP	More

Obr. 22: Priebeh certifikačného procesu balíka modulov ovládača CODESTAR 4 DSP pre Windows 11 verzie 22H2 s úspešným dokončením a podpísaním modulov



Obr. 23: Digitálny podpisový certifikát balíka modulov ovládača CODESTAR 4 DSP pre Windows 11

Obr. 24: Certifikát balíka modulov ovládača CODESTAR 4 DSP pre Windows 11 verzie 22H2 vydaný Microsoftom

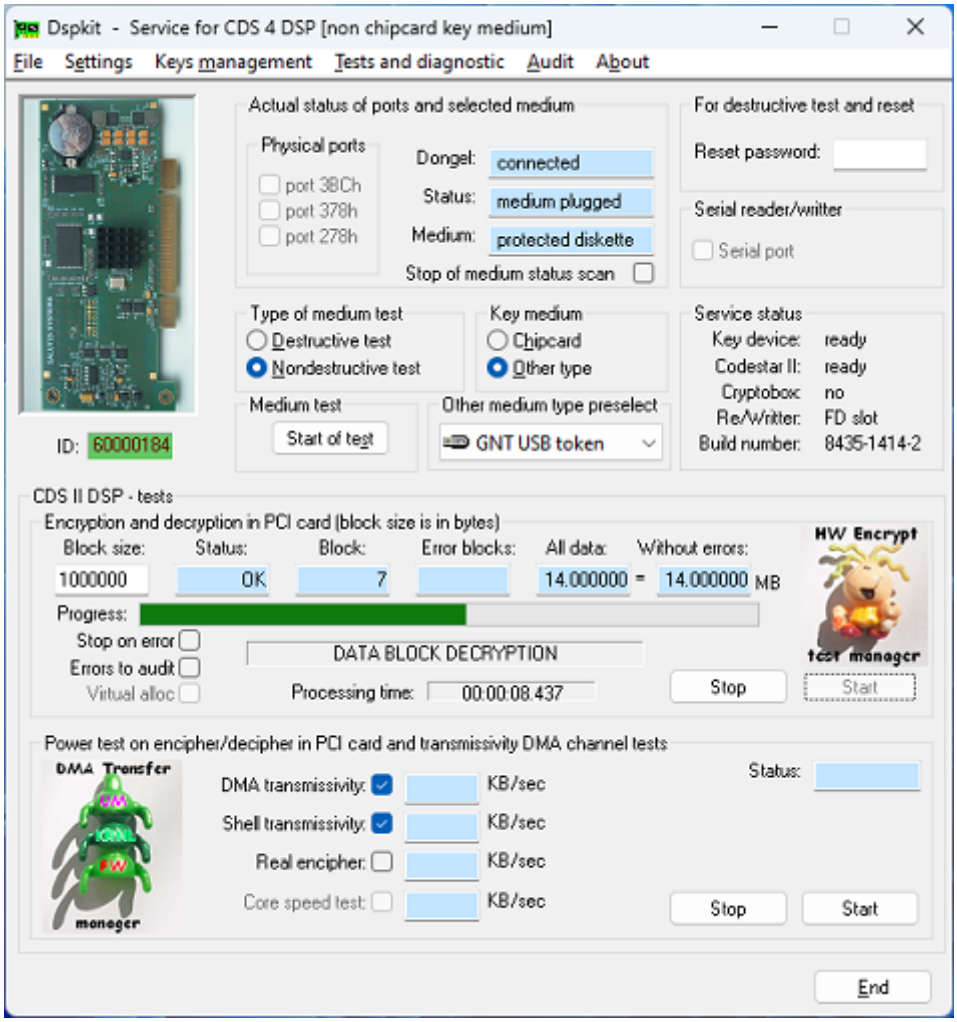
Na obr. 23 je digitálny podpisový certifikát hlavného modulu ***Ism.sys*** pre podporu CODESTAR 4 DSP pre Windows 11. Na obr. 24 je certifikát ovládača CODESTAR 4 DSP pre Windows 11 verzie 22H2.

Až po dosiahnutí tohto výsledku, je balík modulov ovládača CODESTAR 4 DSP funkčne spôsobilý pre jeho zavedenie do systému Windows 11 a jeho štart.

Firmvér s algoritmom SEA64 je pre Microsoft v rámci testovania a certifikácie nedostupný. Certifikuje sa balík ovládačov ale nie firmvér. Testuje sa len funkcionalita firmvéru cez ovládač hardvéru „***ism.sys***“.

9.5.1 Servisný kit PCI kariet CODESTAR DSP

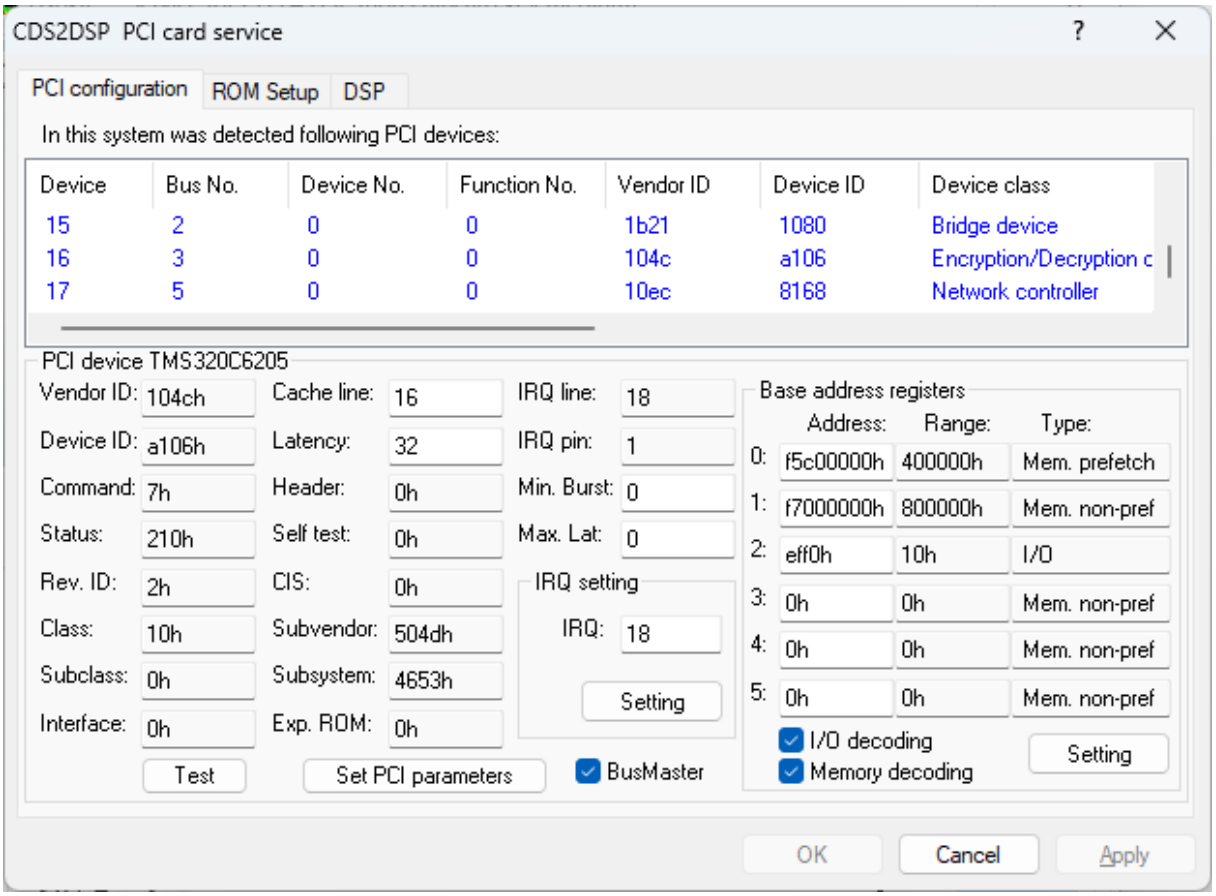
Na oživovanie, servis a testovanie PCI kariet CODESTAR bol vyvinutý program „***DSPKIT.EXE***“ ktorý má cez kernel hostiteľského ***OS*** špeciálny prístup k prostriedkom karty CODESTAR.



Obr. 25: DSP kit pre servis a testovanie PCI karty CODESTAR 4 DSP cez funkcie jej ovládača „Ism.sys“

Na obr. 25 je zobrazené hlavné okno aplikácie DSPKITu na servis a testovanie PCI karty CODESTAR. Tento laboratórny kit zabezpečuje a pokrýva všetko potrebné pre vývoj a laboratórne testy týchto PCI kariet s možnosťou auditovania všetkých úkonov a priebehu operácií s prostriedkami karty a jej testov do denníka udalostí bezpečnostných aplikácií.

Okrem servisu CODESTAR kariet DSPKIT zabezpečuje aj prácu s kľúčovým hospodárstvom s podporou čipových kariet a hlavne USB bezpečnostných tokenov na uloženie a ochranu kryptografických prvkov kľúčového hospodárstva.

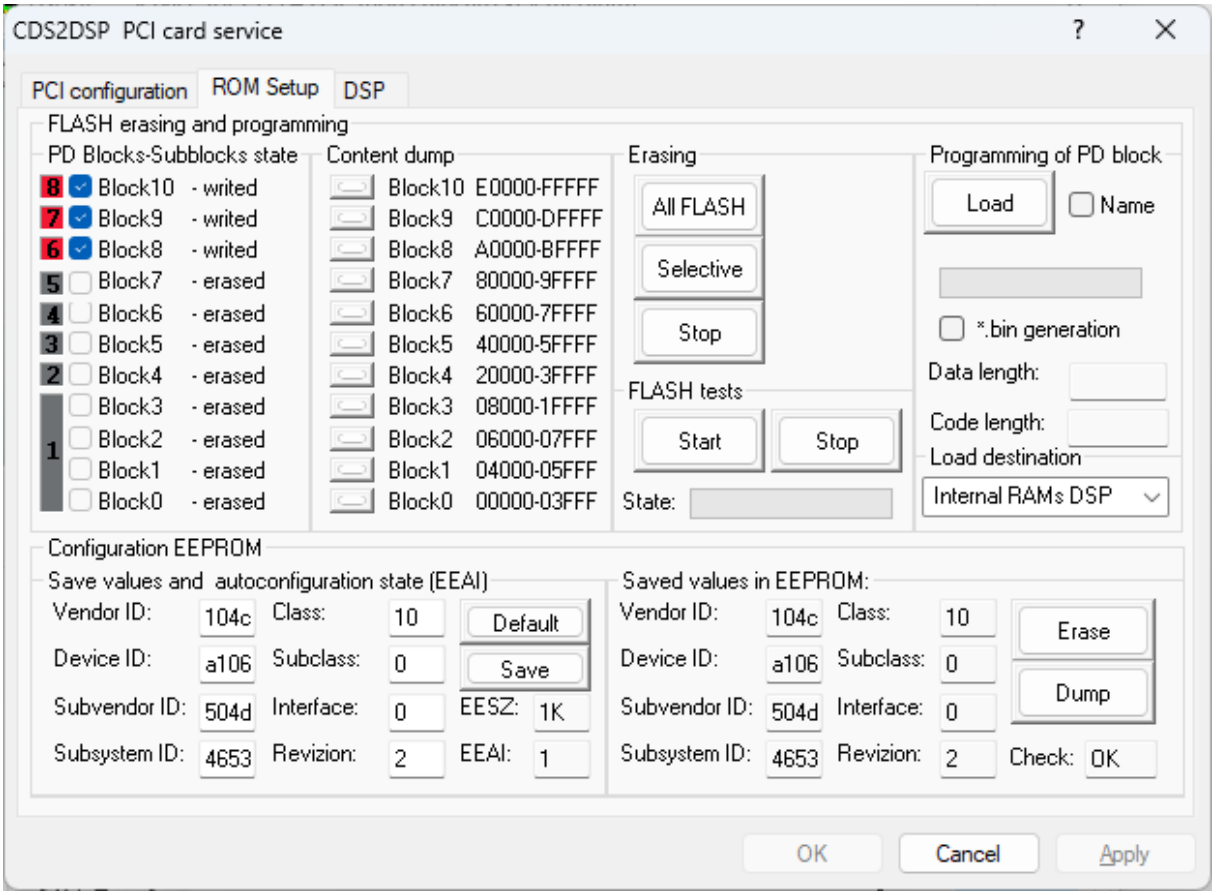


Obr. 26: DSP kit pre servis a testovanie PCI karty CODESTAR DSP – karta PCI konfigurácie pre karty CODESTAR 2 DSP až CODESTAR 4 DSP

Na obr. 26 je zobrazená karta dialógu konfigurácie a nastavení PCI modulu v zariadení procesora DSP. V listboxe je zariadenie „16“ nájdené na zbernici „3“ s Vendor ID = 104c a Device ID = a106. V „Encryptoin/Decription“ triede zariadení je nájdená PCI karta CODESTAR. Ďalej sú vypísané všetky potrebné parametre PCI modulu procesora TMS320C6205. To sú parametre pre PNP ovládač karty CODESTAR modulu „*Ism.sys*“.

Na obr. 27 je zobrazená karta dialógu ROM setupu flash pamäte na karte CODESTAR pre stránkované PD bloky do RAM DSP procesora. Štandardne sú v ROM zapísané 3 PD bloky potrebné pre reálnu prevádzku firmvéru karty. Na karte je aj EEPROM setup.

Zavedené PD bloky do FLASH CDS 4 DSP:
==========
8. SEA64AHW.OUT - binárny kód šifrovacieho algoritmu SEA64 aj s S-boxami
7. CDSTST.OUT - binárny kód pre nastránkovanie testov celého hardvéru s DSP
6. SEAPOWER.OUT - binárny kód pre nastránkovanie benchmarkových testov rýchlosti šifrovania a na servis karty



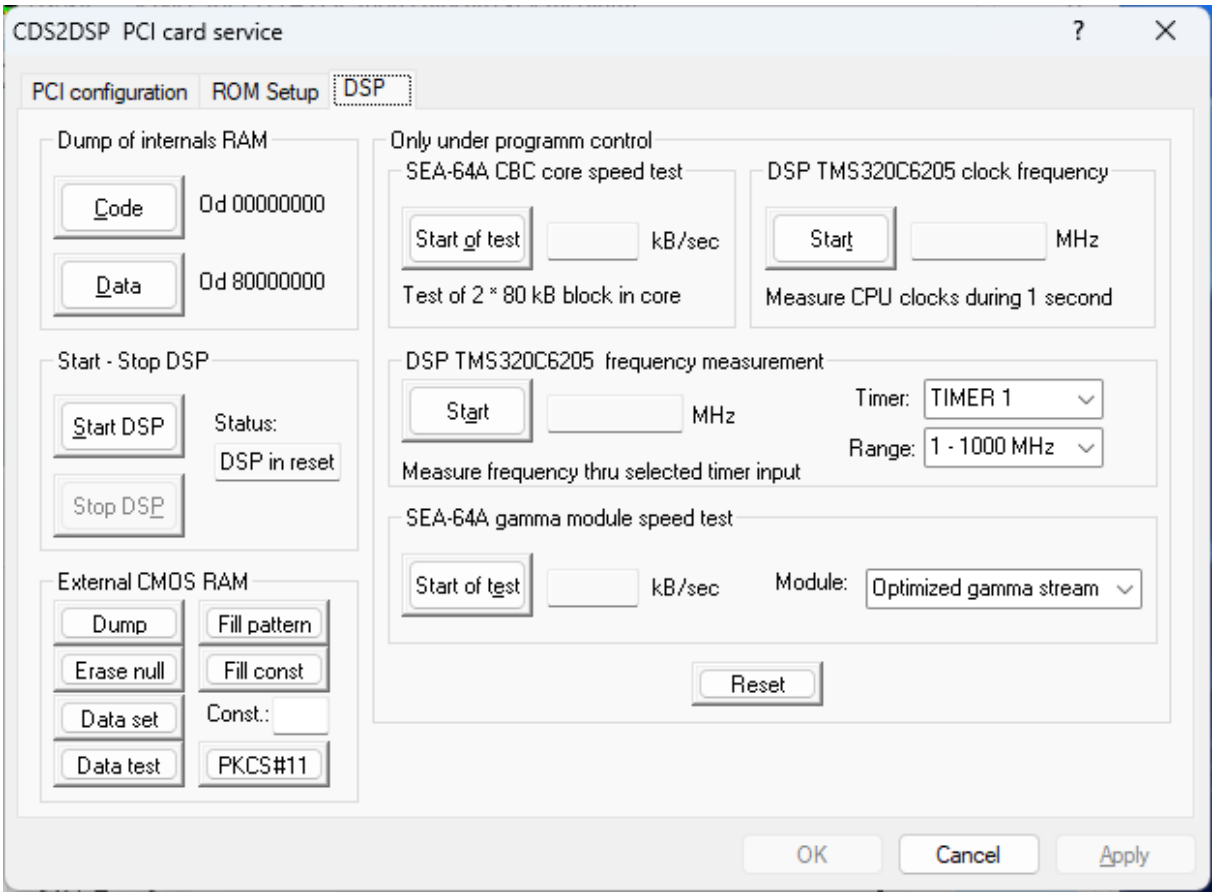
Obr. 27: DSP kit pre servis a testovanie PCI karty CODESTAR DSP – karta PCI ROM setupu pre karty CODESTAR 2 DSP až CODESTAR 4 DSP

EEPROM je konfiguračná pamäť PCI karty pre upresňujúcu konfiguráciu typu triedy zariadenia karty pre Device manažéra ***OS***.

Na obr. 28 je zobrazená karta dialógu benchmarkových testov rýchlosti šifrovania, výpisu obsahu RAM DSP, CMOS RAM, štart a stop procesora a pre ostatné funkcie procesora na kartách CODESTAR 2 DSP až CODESTAR 4 DSP. Cez dualporty a DMA kanál je možné vypísať obsah RAM procesora, externej CMOS RAM na karte a testovať CMOS RAM. V tomto servisnom režime je možné zastaviť beh procesora a aj ho spustiť.

EEPROM setup je naprogramovaný do štandardnej EEPROM pamäte typu 93LC46B. Slúži pre Spávcu zariadení ***OS*** Windows. Hlavne pre nastavenie „Subvendor ID“ a „Subsystem ID“ identifikátorov, ktoré určuje vývojár. „Vendor ID“ a „Device ID“ sú pevne uložené v PCI module DSP procesora jeho výrobcom.

Verzia firmvéru je v1.3 a táto verzia bola použitá aj pri laboratórnych testoch. Kód „SEA64AHW.OUT“ je binárny kód šifrovacieho algoritmu SEA64 aj s S-boxami ako PD blok vo FLASH pamäti pre nastránkovanie do interných RAM procesora pre reálnu prevádzku. „AHW“ v názve určuje, že kód šifrovacieho algoritmu je napísaný a zrealizovaný v assembleri procesora DSP a v konečnom dôsledku sa používa ako hardvérová šifra.



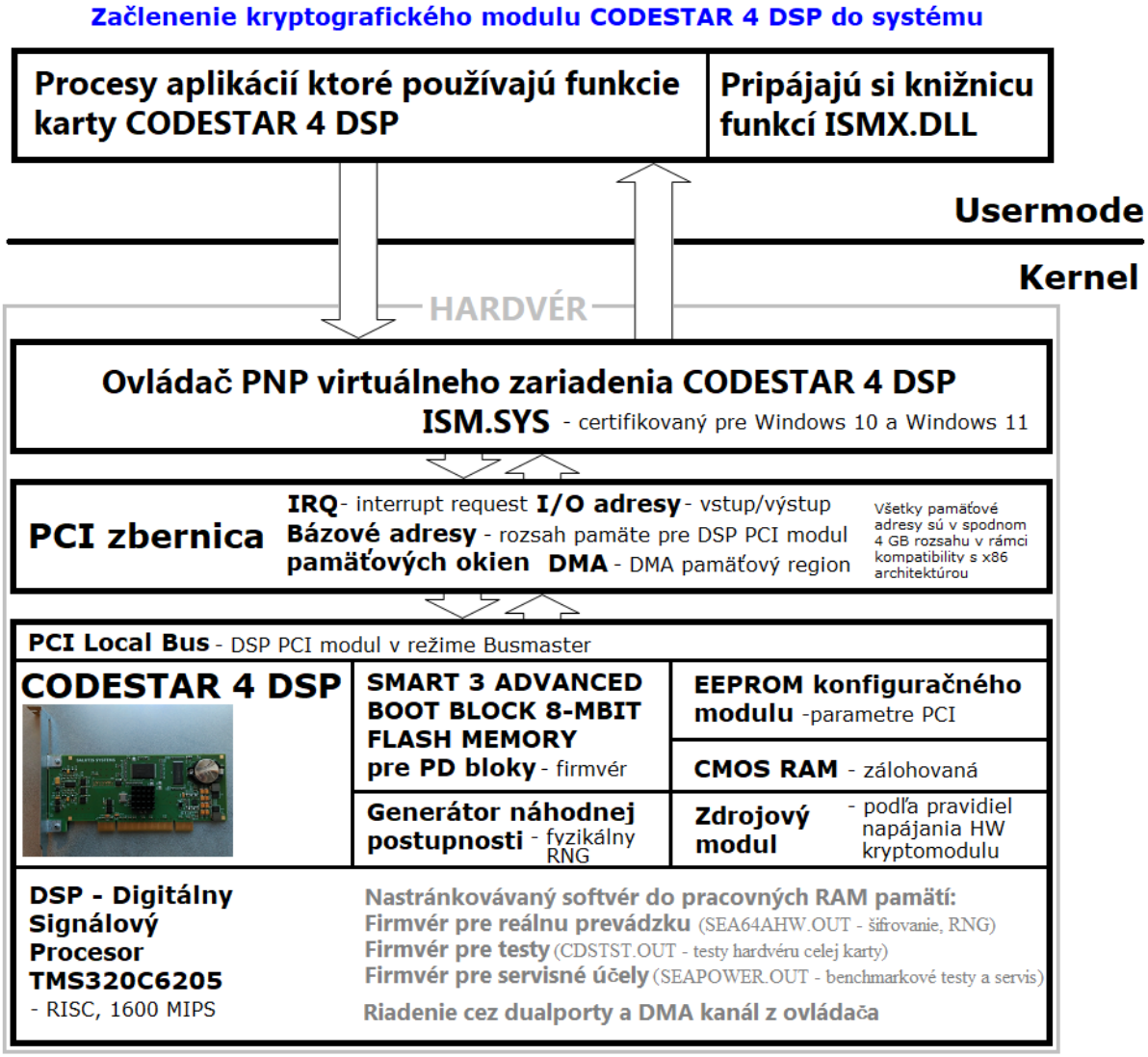
Obr. 28: DSP kit pre servis a testovanie PCI karty CODESTAR DSP – karta benchmarkových testov rýchlosti šifrovania, výpisu obsahu RAM DSP, CMOS RAM, start a stop procesora a pre ostatné funkcie procesora na kartách CODESTAR 2 DSP až CODESTAR 4 DSP

9.6 Začlenenie zariadenia CODESTAR do systému Windows

Hardvér CODESTAR sa v konečnom dôsledku javí v systéme Windows ako virtuálne zariadenie, pre každý proces bežiaci v ***OS***. Úlohou ovládača karty CODESTAR je virtualizovať funkcie karty tak, že sa to pre každý proces javí, ako keby bolo toto zariadenie len jeho. To jedna zo základných vlastností PNP ovládačov virtuálnych zariadení systému Windows s preemptívnym multitaskingom.

Na obr. 29 je diagram začlenenia zariadenia karty CODESTAR 4 DSP do systému Windows a štruktúry architektúry hardvérovej podpory. Karta aj s celou podporou je určená predovšetkým pre našu ***CBA***, ale aj ostatné naše aplikácie, napríklad IPcrypt QS32. Celý kryptografický modul pozostáva z knižnice funkcií modulu, balíka modulov ovládača a hardvéru karty a jeho firmvéru. Je to viditeľné na diagrame. Ovládač a firmvér Microsoft už považuje za hardvér.

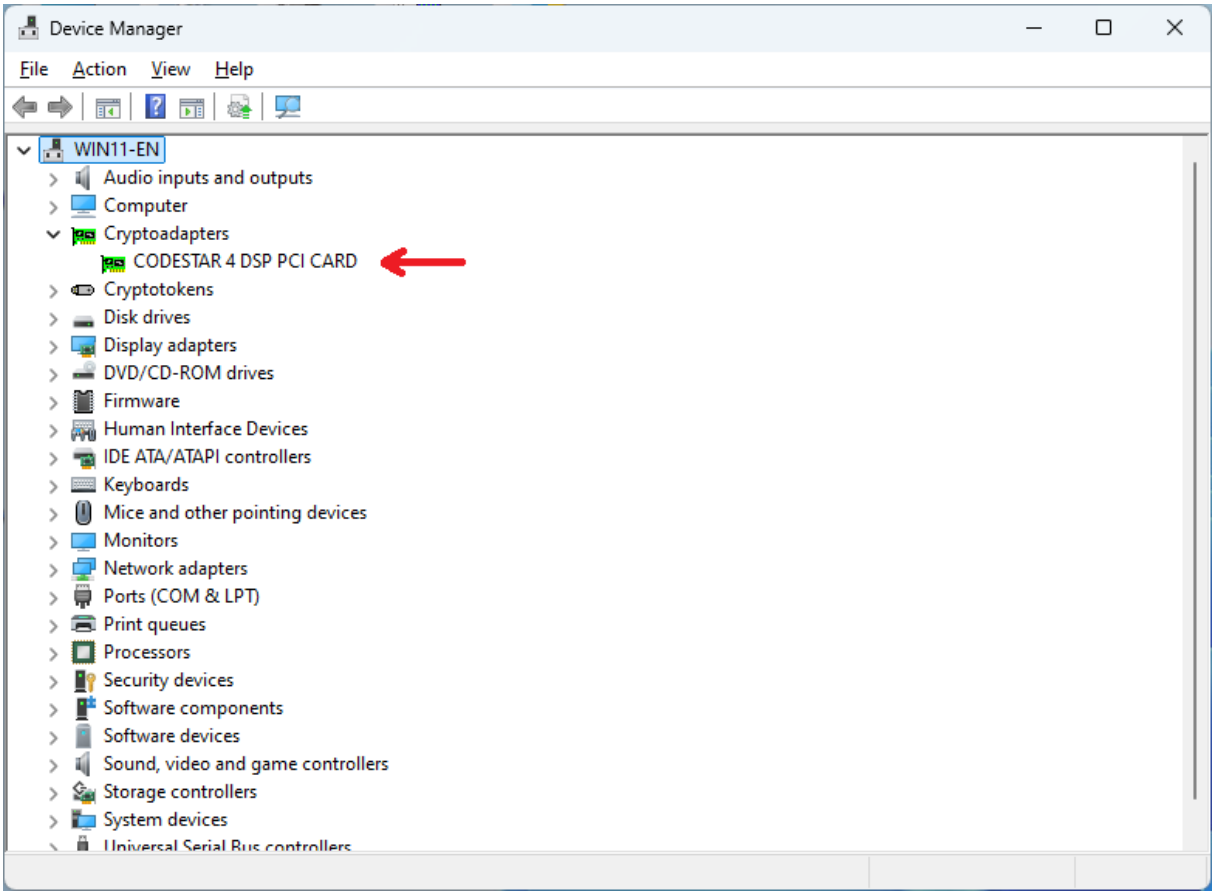
Na obr. 30 je okno dialógu Manažéra zariadení systému Windows, kde je karta CODESTAR 4 DSP je inštalovaná v triede zariadení „Cryptoadapters“.



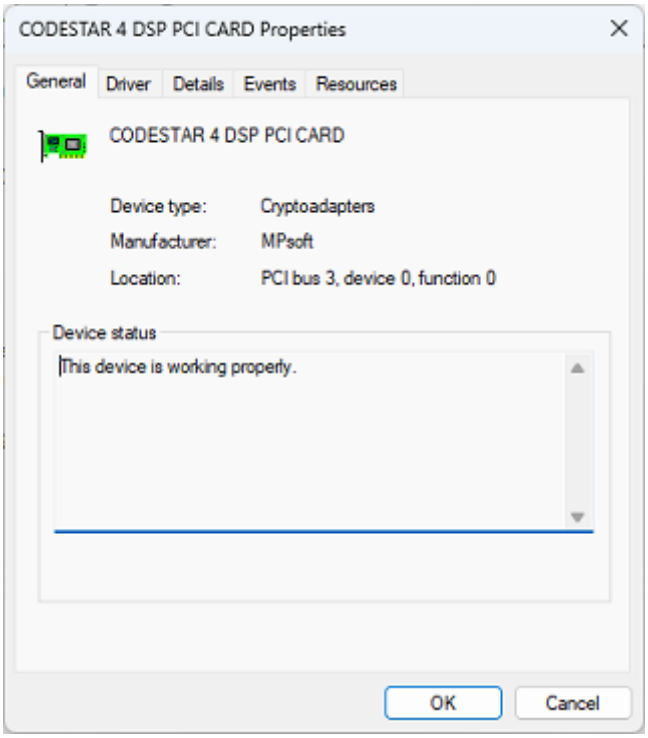
Obr. 29: Diagram začlenenia zariadenia karty CODESTAR 4 DSP do systému Windows a štruktúry architektúry hardvérovej podpory. Všetko v oblasti Kernel, ako hardvér tak aj softvér, Microsoft nazýva hardvérom.

V dialógu vlastností Manažéra zariadení systému Windows, kde je karta CODESTAR 4 DSP inštalovaná v triede zariadení „Cryptoadapters“, je definovaná ako typ zariadenia „Cryptoadapters“ a lokalizovaná na PCI zbernici číslo 3, ako zariadenie číslo 0 s funkciou 0. Zobrazuje to obr. 31.

Ďalej nasleduje výpis súboru s informáciami o inštalácii „Cds4dsp.inf“, ktorý obsahuje všetky informácie, ktoré inštalačné súčasti zariadenia používajú na inštaláciu balíka ovládačov do systému Windows. V spodnej časti INF súboru je nápoveda pre manuálne nastavenia typu transferu komunikácie s kartou cez PCI zbernicu, zmenou hodnoty "TransferMode"
v kľúči „Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ISM\Parameters\TransferMode“ v registry tak, ako to zobrazuje obr. 32. Inakšie sa nastavuje typ transferu automaticky pri inštalácii kernel modulu pre optimálnu komunikáciu na:
„COMMAND_MODE_AUTO_SET 255 - automatic mode detection and setting“.
.



Obr. 30: Karta CODESTAR 4 DSP v Manažérovi zariadení systému Windows je inštalovaná v triede zariadení „Cryptoadapters“



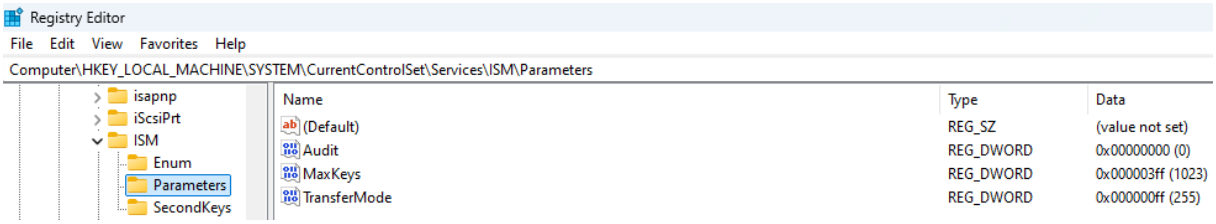
Obr. 31: Zariadenie karty CODESTAR 4 DSP v Manažérovi zariadení systému Windows

Výpis súboru s informáciami o inštalácii ovládača „*Ism.sys*“, „Cds4dsp.inf“:

```
;-------------------------------------------------------------------------
;  (c) Copyright 2016 – 2023, MPsoft
;-------------------------------------------------------------------------

; Installation of ISM.SYS x64 driver package for CODESTAR 4 PCI card

[Version]
Signature="$Windows NT$"
Class=Cryptoadapters
ClassGUID={F0C0BDED-936E-4d33-A667-476CC99065D0}
provider=%MPSOFT%
CatalogFile=cds4dsp.cat
DriverVer=02/05/2023,0.6.20.1

[Manufacturer]
%MPSOFT%=MPSOFT, NTamd64,

[MPSOFT.NTamd64]
%Cds4DSPDesc%=cds4dsp.Dev.NTamd64, PCI\VEN_104c&DEV_A106&SUBSYS_4653504D

[ControlFlags]
ExcludeFromSelect = PCI\VEN_104c&DEV_A106&SUBSYS_4653504D

[DestinationDirs]
cds4dsp.Files.Ext.NTamd64 = 10,System32\Drivers
cds4dsp.Files.Dll.NTamd64 = 10,System32
cds4dsp.Files.Dllx.NTamd64 = 10,System32

[SourceDisksFiles]
ism.sys=1
eppcds.dll=1
ismx.dll=1

[SourceDisksNames]
1=%INSTDISK%,,,""

[ClassInstall32]
AddReg=UpDateRegistry

[UpDateRegistry]
HKR,,,,Cryptoadapters
HKR,,EnumPropPages32,,"Eppcds.dll,CdsPropPages"
HKR,,SilentInstall,,1
HKR,,NoInstallClass,,1
HKR,,Icon,,"101"

[cds4dsp.Dev.NTamd64]
CopyFiles= cds4dsp.Files.Ext.NTamd64
CopyFiles= cds4dsp.Files.Dll.NTamd64
CopyFiles= cds4dsp.Files.Dllx.NTamd64

[cds4dsp.Dev.NTamd64.Services]
Addservice = Ism, 0x00000002, cds4dsp.AddService.NTamd64

[cds4dsp.AddService.NTamd64]
DisplayName    = %cds4dsp.SvcDesc%
ServiceType    = 1                  ; SERVICE_KERNEL_DRIVER
StartType      = 3                  ; SERVICE_DEMAND_START
```

```
ErrorControl   = 1              ; SERVICE_ERROR_NORMAL
LoadOrderGroup = Extended base
ServiceBinary  = %10%\System32\Drivers\ism.sys

AddReg = cds4dsp.AddReg.NTamd64
AddReg = parameters_subkey
AddReg = parameters_add

[cds4dsp.AddReg.NTamd64]

;  ******* Hodnoty podkluca Parameters *******

[parameters_subkey]
HKR, "Parameters", "TransferMode", %REG_DWORD%, 0x000000FF
HKR, "Parameters", "Audit", %REG_DWORD%, 0x00000000
HKR, "Parameters", "MaxKeys", %REG_DWORD%, 1023

;  ******* Hodnota v hlavnom kluci *******

[parameters_add]
HKR,, "LoadType", %REG_DWORD%, 0x00000001


[cds4dsp.Files.Ext.NTamd64]
ism.sys

[cds4dsp.Files.Dll.NTamd64]
eppcds.dll

[cds4dsp.Files.Dllx.NTamd64]
ismx.dll

[Strings]
INSTDISK="CDS4DSP Installation Disk"
MPSOFT="MPsoft"
Cds4DSPDesc="CODESTAR 4 DSP PCI CARD"
cds4dsp.SvcDesc="CDS4DSP service"

SERVICE_BOOT_START = 0x0
SERVICE_SYSTEM_START = 0x1
SERVICE_AUTO_START = 0x2
SERVICE_DEMAND_START = 0x3
SERVICE_DISABLED = 0x4

SERVICE_KERNEL_DRIVER = 0x1
SERVICE_ERROR_IGNORE = 0x0
SERVICE_ERROR_NORMAL = 0x1
SERVICE_ERROR_SEVERE = 0x2
SERVICE_ERROR_CRITICAL = 0x3

REG_EXPAND_SZ = 0x00020000
REG_DWORD = 0x00010001

;   Codes of communication modes with DSP adjustable in the "Parameters\TransferMode" value:
;
;   COMMAND_MODE_SETUP_DMA          0   - setup mode (character default value)
;   COMMAND_MODE_POOL_DMA           1   - pooling mode
;   COMMAND_MODE_INT_DMA            2   - interrupt mode (preferred mode, set as default by
                                          translation)
;   COMMAND_MODE_FAST_DMA           3   - fast pooling mode
```

```
;  COMMAND_MODE_POOL_DMA_DIAG      4   - diagnostic slow pooling mode
;  COMMAND_MODE_NOINIT_INT_POOL_DMA_DIAG  5   - diagnostic mode with prohibition of HW
                                                interrupt system initialization,
                                                "COMMAND_MODE_FAST_DMA" mode
;  COMMAND_MODE_AUTO_SET              255 - automatic mode detection and setting
;
;  Codes of the required level in the "Audit" value for diagnostic purposes of the "ISM" module:
;
;  KDA_RESERVE                        0   - is not recorded in the event log
;  FATAL_ERROR                        1   - to the level of a fatal error
;  CRITICAL_ERROR                     2   - to the critical error level
;  RECOVERABLE_ERROR                  3   - to the level of a recoverable state error
;  GENERAL_EVENT_SUCCESS              4   - to the level of general success
;  DIAG_EVENT_SUCCESS                 5   - to the diagnostic level of general success
;  DIAG_FOR_IRRELEVANT_EVENTS         6   - to the diagnostic level for irrelevant events
;  DIAG_ONLY_FOR_REAL_OPERATION       7   - only for real operation to verify the operation routine
                                            function
```

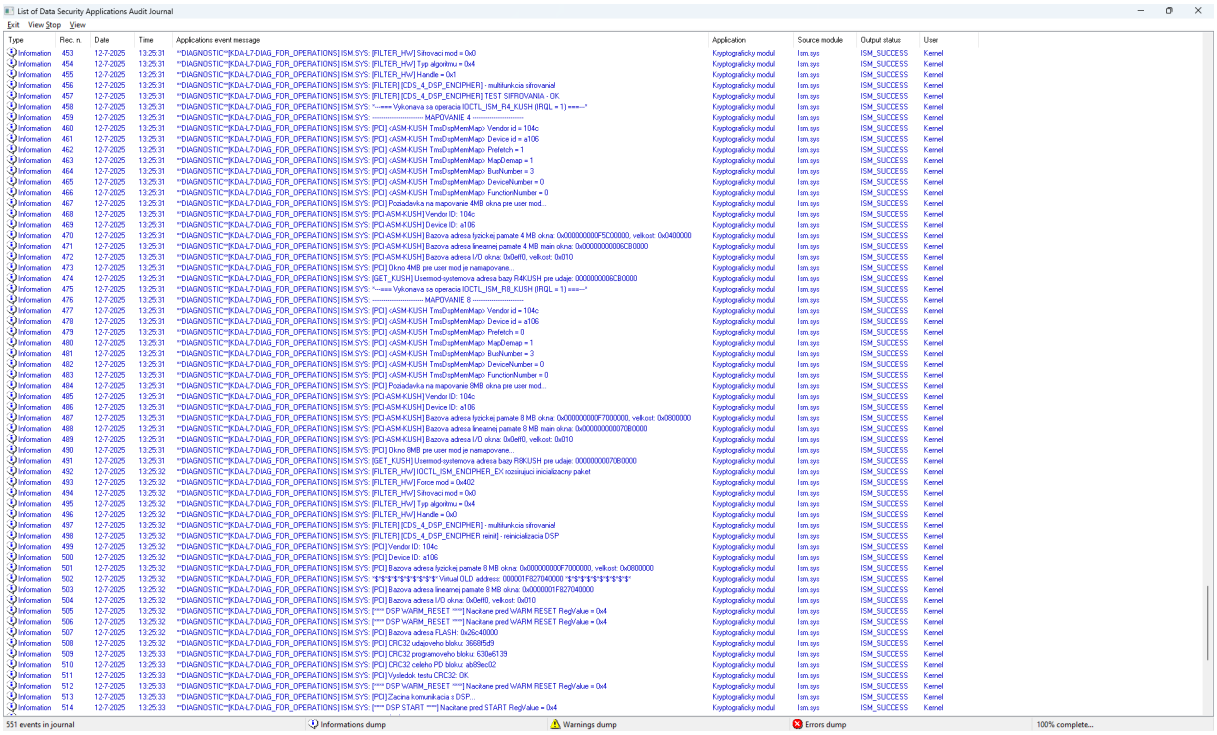


Obr. 32: Možnosť manuálnych zmien v podkľúči registry „Parameters\TransferMode“ a „Audit“.

Manuálne nastavenie typu auditovania funkcí kernel modulu „Ism.sys“ pre diagnostické účely celej zostavy cez PCI zbernicu, sa vykoná zmenou zmenou hodnoty "Audit" v kľúči „Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ISM\ Parameters\Audit“ v registry tak, ako to zobrazuje obr. 32. Inakšie sa inštaláciou nastavuje ako default typ auditu do denníka udalostí automaticky na:
„KDA_RESERVE 0 - is not recorded in the event log“.
Pre plný kernel audit diagnostického testovania reálnej prevádzky sa nastaví:
„DIAG_ONLY_FOR_REAL_OPERATION 7 - only for real operation to verify the operation routine function“

Diagnostické testy požaduje ***BŠ***. Veľkému množstvu kernel modulov iných firiem pracujúcich v RING0 procesora takéto testy chýbajú.

Na obr. 33 je zobrazená malá časť záznamu v denníku udalostí, na obr. 34 je z neho detail, spusteného diagnostického kernel testu karty CODESTAR 4 DSP pri nastavení hodnoty „DIAG_ONLY_FOR_REAL_OPERATION“ v „Parameters\Audit“ určenej na diagnostické auditovanie reálnej prevádzky počas diagnostických testov. Je to vyčerpávajúci audit prevádzky šifrovania signálovým procesorom v súčinnosti s ovládačom pracujúcim v RING0 chráneného režimu procesora hostiteľského systému. Beh programu to síce nesmierne spomaľuje, ale podáva dôležité informácie, hlavne pre vývojára a experta. Na obr 34 je pre lepšiu čitateľnosť zobrazený detail z tejto malej časti záznamu v denníku udalostí. Konkrétne sa vykonávalo mapovanie do pamäťových oblastí usermódu pre vytvorenie pamäťových okien z hardvéru v kerneli.



Obr. 33: Fragment záznamov v denníku udalostí z diagnostického kernel testu karty CODESTAR 4 DSP spusteného v aplikácii DSPKIT.

Information 456 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [FILTER] [CDS_4_DSP_ENCIPHER] - multifunkcia sifrovania!
Information 457 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [FILTER] [CDS_4_DSP_ENCIPHER] TEST SIFROVANIA - OK
Information 458 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: "---=== Vykonava sa operacia IOCTL_ISM_R4_KUSH (IRQL = 1) ====---"
Information 459 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: ---------- MAPOVANIE 4 ----------
Information 460 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Vendor id = 104c
Information 461 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Device id = a106
Information 462 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Prefetch = 1
Information 463 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> MapDemap = 1
Information 464 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> BusNumber = 3
Information 465 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> DeviceNumber = 0
Information 466 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> FunctionNumber = 0
Information 467 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] Poziadavka na mapovanie 4MB okna pre user mod...
Information 468 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI-ASM-KUSH] Vendor ID: 104c
Information 469 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI-ASM-KUSH] Device ID: a106
Information 470 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI-ASM-KUSH] Bazova adresa fyzickej pamate 4 MB okna: 0x000000000F5C00000, velkost: 0x0400000
Information 471 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI-ASM-KUSH] Bazova adresa linearnej pamate 4 MB main okna: 0x000000000006CB0000
Information 472 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI-ASM-KUSH] Bazova adresa I/O okna: 0x0eff0, velkost: 0x010
Information 473 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] Okno 4MB pre user mod je namapovane...
Information 474 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [GET_KUSH] Usermod-systemova adresa bazy R4KUSH pre udaje: 0000000006CB0000
Information 475 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: "---=== Vykonava sa operacia IOCTL_ISM_R8_KUSH (IRQL = 1) ====---"
Information 476 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: ---------- MAPOVANIE 8 ----------
Information 477 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Vendor id = 104c
Information 478 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Device id = a106
Information 479 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> Prefetch = 0
Information 480 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> MapDemap = 1
Information 481 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> BusNumber = 3
Information 482 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> DeviceNumber = 0
Information 483 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] <ASM-KUSH TmsDspMemMap> FunctionNumber = 0
Information 484 12-7-2025 13:25:31 **DIAGNOSTIC**[KDA-L7-DIAG_FOR_OPERATIONS] ISM.SYS: [PCI] Pozadavka na mapovanie 8MB okna pre user mod...

Obr. 34: Detail z fragmentu záznamu v denníku udalostí z diagnostického kernel testu karty

Týmto popisom bol uvedený aspoň stručný príklad implementácie šifrovacieho algoritmu SEA64 do hardvéru, konkrétne do prídavnej PCI karty s digitálnym signálovým procesorom. Podobným spôsobom je možné implementovať aj algoritmy SEA512 a SEA1024 do tohto istého hardvéru, len zavedením príslušného firmvéru. Rozsah a prostriedky tohto hardvéru na báze DSP radu C6000 boli navrhované predovšetkým pre naše algoritmy SEA. Algoritmus AES však nie je možné implementovať do tohto hardvéru z dôvodu veľkého rozsahu binárneho kódu algoritmu AES. AES je kódovo kapacitne ďaleko rozsiahlejší ako algoritmy SEA, hoci SEA512 a SAE1024 majú objemovo väčšie S-boxy ako AES.

10. Príklady implementácie algoritmov SEA a AES256 do čisto softvérových projektov a projektov softvérovo-hardvérových

10.1 IPcrypt v 6.1 na šifrovanie sieťovej prevádzky

Je to projekt, ktorý nie je bežným „krabicovým softvérom“ a je určený ako prostriedok ***PŠOI_OUS*** určený na šifrovú ochranu informácií utajovaných skutočností prenášaných po sieti pre stupeň utajenia „***V***“ - ***Vyhradené*** a je certifikovaný Národným Bezpečnostným Úradom Slovenskej republiky. Hlavným šifrovacím algoritmom podľa ***BŠ*** je AES256 a ako ďalší algoritmus je použitý aj SEA64. Niektoré časti projektu podliehajú utajeniu až do stupňa „***D - dôverné***“, preto nie je možné sa o nich zmieňovať a opis musí byť len stručný. Je to projekt so softvérovo-hardvérovou implementáciou algoritmov AES256 a SEA64.

Do doby, kým algoritmus AES nebol prijatý ako štandard, verzia IPcryptu „***IPprot***“ nasadená v ozbrojených silách používala ako hlavný šifrovací algoritmus SEA64. V tej dobe sa používali v PC staršie typy procesorov PENTIUM alebo CELERON. Rýchlosť spracovania algoritmom SEA64 bola pre sieťové prenosy po sieťach 100 Megabitov plne postačujúca. Po zavedení verejného algoritmu AES256 do ***BŠ***, čo požadovali EÚ a NATO pre ***PŠOI_OUS***, nastali problémy s rýchlosťou spracovania AES256 na starých procesoroch. Bolo to na hranici použiteľnosti. Bolo to tiež spôsobené striktnou, prikazovanou v ***BŠ***, potrebou zmeny dátového smerového šifrovacieho kľúča po každom prenesenom jednom megabajte dát, prípadne po 10-tich minútach nečinnosti alebo malého prenosu dát. A tiež pri šifrovaní interaktívnej výmeny smerového dátového kľúča (nahrádza zastaralý Diffie-Hellman algoritmus založený na asymetrickej kryptografii). AES totižto potrebuje pred použitím kľúča prípravu šifrovacieho kontextu, čo sa stalo tiež kameňom úrazu na pomalých procesoroch typu PENTIUM alebo CELERON. To sa v prípade SEA64 nemuselo vykonávať, hoci už vtedy podľa nového ***BŠ*** sa menil kľúč tiež po každom jednom prenesenom megabajte. Až príchodom výkonnejších procesorov sa stal AES256 v IPprote a IPcrypte už použiteľným.

IPcrypt je určený pre profesionálne siete hlavne v štátnom sektore. Požaduje hlavný šifrovaný server, monitorovací a distribučný server. Tie dva posledné môžu byť zlúčené do jedného hardvéru, v jednom PC servera. Je to ďaleko pokročilejšie a vysoko bezpečnejšie riešenie ako realizácia šifrovania bežných prenosov po sieti s použitím aj asymetrickej kryptografie.

Dôležitou súčasťou týchto produktov je ***CBA*** – Centrálna Bezpečnostná Autorita, realizovaná ako špeciálny PC, vyhovujúci až do stupňa utajenia „***D***“. Jadrom kryptografie v CBA je práve hardvér PCI karty CODESTAR 4 DSP, v ktorej je implementovaný algoritmus SEA64 realizovaný firmvérom bežiacim na procesore DSP a tiež zabudovaný RNG generátor náhodných dát a chránené úložište PKI objektov, prístupné cez rozhranie PKCS#11. ***CBA*** nesmie byť pripojená k počítačovej sieti a musí byť inštalovaná v PC prispôsobeným danému stupňu utajenia. Umiestnená musí byť v zabezpečenej režimovej miestnosti, chránenom priestore podľa pokynov NBÚ SR.

Algoritmus SEA64 sa v CBA používa napríklad na šifrovanie záznamov v ***Databáze kľúčových médií*** (USB tokeny s bezpečnostným procesorom) a záznamov v ***Systémovom denníku udalostí*** s výsledkami auditu funkcií CBA. Tieto záznamy obsahujú citlivé informácie, napríklad heslá a PINy k tokenom.

Konkrétne CBA sa inštaluje, konfiguruje a vyrába odborne "na mieru", preto neexistuje jednoliaty inštalačný softvér, len čiastkové inštalácie jednotlivých komponentov CBA.

Podobne prebieha aj inštalácia komponentov celej siete IPcrypt. Celý systém potom funguje úplne transparentne bez akejkoľvek záťaže používateľov a navyše je potrebný len jeden USB prístupový token pre každého používateľa v sieti. Klienti IPcryptu môžu pracovať ako v chránených, tak aj nechránených priestoroch. V nechránených priestoroch sa bezpečnosť upevňuje spôsobom používania USB tokenov a ďalšími vrstvami softvérovej bezpečnosti, čo sa volí pri inštalácii. Servery sú vždy umiestnené v chránených priestoroch.

10.2 CRYPTOOL512 a CRYPTOOL1024 aplikácie na šifrovanie súborov

Aplikácia CRYPTOOL512 je vývojovým medzistupňom pre vývoj aplikácie CRYPTOOL1024, ktorá je špeciálne vyvíjaná pre Windows 11, kde Microsoft stavil vo veľkej miere na bezpečnosť systému. CRYPTOOL1024 poskytuje však spätnú kompatibilitu aj pre Windows 10 a aj nižšie verzie Windows. Ďalší popis sa preto bude venovať len aplikácii CRYPTOOL1024. Aplikačné rozhrania obidvoch aplikácií sú veľmi podobné.

10.2.1 Základný popis silného a bezpečného šifrovania súborov

Prostriedok šifrovej ochrany informácií CRYPTOOL1024 Professional v2.0 je aplikácia, ktorá spolu s prostriedkom šifrovej ochrany informácií Centrálnej Bezpečnostnej Autority CBA v3.4 tvorí systém určený na veľmi silné a bezpečné šifrovanie súborov. Systém CRYPTOOL1024 je vybudovaný na báze kvantovo bezpečnej kryptografie. Týmto sa radí medzi prostriedky post-kvantovej kryptografie, čo je označenie pre kryptografiu, ktorá je odolná voči kvantovým výpočtom.

Ide o systém ochrany údajov v súboroch uložených na pamäťových médiách určených na ukladanie súborov, prípadne v zariadeniach so súborovým systémom prístupným po ich pripojení k operačnému systému Windows. Ochrana je vykonávaná šifrovaním s použitím veľmi silného šifrovacieho algoritmu a ďalších kryptografických funkcií určených pre post-kvantovú kryptografiu.

Použitie systému CRYPTOOL1024 predpokladá použitie dobre zabezpečených technických prostriedkov, konkrétne počítačov pre CBA a aj inštalácie aplikácie CRYPTOOL1024 v2.0 na šifrovanie súborov. Taktiež je potrebné použitie zariadení na uchovávanie šifrovacích kľúčov a inicializačno-aktivačnej postupnosti pre šifrovací algoritmus SEA1024 použitý v aplikácii. Na tento účel sa používajú USB GNT tokeny s kryptoprocesorom.

Systém CRYPTOOL1024 umožňuje bezpečne šifrovať aj veľmi veľké súbory, nakoľko používa šifrovací kľúč s dĺžkou 1024 bitov. Šifrovaný text v cieľových súboroch, ktoré sú výsledkom procesu zašifrovania, neobsahuje žiadne postranné informácie a tiež ani stopy

periodicity a iných stôp šifrovania, ktoré by pomohli pri kvantových výpočtoch realizovaného útoku na prelomenie šifrovania. Tento benefit je dosahovaný aj vďaka vysokej kvalite šifrovacieho algoritmu.

10.2.2 Popis systému CRYPTOOL1024

Systém CRYPTOOL 1024 pozostáva z dvoch prostriedkov šifrovej ochrany informácií:

- Centrálna Bezpečnostná Autorita v3.4, ďalej len ***CBA***
- Používateľská aplikácia CRYPTOOL1024 Professional v2.0

CBA je nainštalovaná na počítači, ktorý je ako špeciálny technický prostriedok upravený tak, že je oddelený od siete, je osadený prídavnou PCI kartou CODESTAR 4 DSP a má nainštalované bezpečnostné nastavenia vydané NBÚ SR. Vlastný softvér CBA je nainštalovaný v adresári „CRYPTSRV“, v ktorom sú umiestnené systémové súbory CBA. Okrem toho sú v danom PC nainštalované ovládače pre podporu PCI karty, kryptomodulu, USB GNT tokenov a ISO 7816 emulácie čipkariet. PC s CBA nikdy nesmie byť pripojený k sieti a internetu. Pravidelné updatovanie aktualizáciami nie je potrebné a z dôvodu bezpečnosti je aj nežiadúce. Okrem softvéru CBA nie je povolená ani inštalácia ďalšieho softvéru do PC. Sú to požiadavky stanovené v rámci bezpečnosti. Sieťová karta na hlavnej doske PC musí byť zablokovaná hardvérovo a v BIOSe.

CBA obsahuje nedeterministický hardvérový generátor náhodnej postupnosti, realizovaný na prídavnej PCI karte CODESTAR 4 DSP. Z tejto postupnosti, prísne testovanej, sa vyberajú náhodným spôsobom šifrovacie kľúče, ktoré sa ukladajú priamo do GNT tokenov. Okrem softvérového modulu realizujúceho túto funkciu, CBA obsahuje modul ***Databázy Kľúčových Médií*** a modul ***Denníka Udalostí*** auditu všetkých vykonávaných funkcií CBA. Hlavný exekutabilný súbor „CRYPTSRV.EXE“, realizuje základné funkcie CBA, medzi ktoré patria aj ***Automatizované Funkcie*** pre prácu s kľúčovými médiami GNT tokenov. Tieto sú previazané s modulom databázy kľúčových médií.

CBA obsluhuje poverený administrátor, ktorý pripraví a vydáva používateľom aplikácie CRYPTOOL1024 Professional v2.0 USB GNT tokeny, v ktorých sú uložené šifrovacie kľúče a inicializačná postupnosť pre aktiváciu šifrovacieho kontextu klienta, tvorená parciálnymi klúčami pre S-boxy.

Používateľská aplikácia CRYPTOOL1024 Professional v2.0 je nainštalovaná v technickom prostriedku počítača, na ktorom sa vykonáva šifrovanie alebo dešifrovanie súborov. Aplikácia je vyvinutá s ohľadom na bezpečnosť, ktorej základom, tak ako pri CBA, je realizácia spĺňajúca požiadavky kritérií bezpečnosti podľa Bezpečnostného Štandardu (***BŠ***) vydaného NBÚ SR. Navyše je použitý šifrovací algoritmus SEA1024, ktorý je silnejší ako AES256 a má šifrovací kľúč so štvornásobnou dĺžkou. Z dôvodu realizácie post-kvantovej bezpečnosti sú niektoré požiadavky ***BŠ*** doplnené o realizáciu ďalších bezpečnostných funkcií.

Obsluha aplikácie CRYPTOOL1024 je jednoduchá a aplikácia výpismi vedie používateľa k dosiahnutia výsledku zvolenej kryptografickej funkcie. Počas vykonávania funkcií aplikácie sa vykonáva aj auditný záznam do denníka udalostí, ktorý je šifrovaný. Postranné informácie o spracovávaných súboroch sa ukladajú do šifrovanej databázy, ktorá sa automaticky a na pozadí používa pri dešifrovaní. Všetky postranné informácie tak zostávajú v PC klienta. Výsledný obsah šifrovaného súboru neobsahuje žiadne postranné informácie. Je to dôležité

z hľadiska požiadaviek post-kvantovej kryptografie. Šifrovaný text nie je štrukturovaný, tak ako pri použití iných aplikácií. Je vytváraný procesom xorovania streamu gamma blokov, generovaných použitím GAMMA módu šifry SEA1024. Je to vo výsledku podobnosť so šifrovaním VERNAMOVEJ šifry, ktorá sa javí ako najsilnejšia šifra. Nie je to však priamo Vernamova šifra, pri ktorej sa prixórováva heslo, ktoré je vytvorené mimo šifrátora a má dĺžku otvoreného textu. Aplikácia CRYPTOOL1024 generuje toto heslo ako stream blokov vlastnej tvorby hesla. Je to symetrické šifrovanie blokovou šifrou v gamma režime. Dešifrovanie prebieha opätovným prixórovaním streamu toho istého hesla, vygenerovaného tým istým kľúčom, ktorý bol použitý pri zašifrovaní. Je to v konečnom dôsledku jednoduchý proces, avšak mimoriadne silný.

Stručný popis vlastností systému CRYPTOOL1024 pre skupinu systémov Windows 7 až Windows 11 pre architektúru procesorov x64:

- programové vybavenie systému CRYPTOOL1024 klienta je určené pre 64-bitové operačné systémy typu NT (New Technology), ako Windows 7 až po Windows 11 pre 64 bitovú architektúru procesora,
- programové vybavenie CBA je určené pre 64-bitové operačné systémy Windows 10 a 11,
- šifrujú sa obsahy súborov bez postranných informácií,
- šifrovanie je sprístupnené až po úspešnej autentizácii k používanému GNT tokenu a zadaní čísla používaného šifrovacieho kľúča a následnej buď automatickej alebo manuálnej inicializácii šifrovacieho kontextu, čo určuje obsah tokenu generovaný v CBA,
- v prípade voľby tzv. delenej šifry, CBA určí použitie dvojice USB GNT tokenov,
- šifrovaním kvalitným a silným šifrovacím algotitmom SEA1024 s kľúčom uchovávaným mimo PC s nainštalovaným systémom CRYPTOOL1024 sa zabezpečí vysoký stupeň ochrany informácií uložených v zašifrovaných súboroch,
- šifruje sa algoritmom implementovaným podľa ***BŠ***,
- systém šifrovania používa na šifrovanie softvérom realizovaný algoritmus SEA1024 s 1024-bitovým kľúčom,
- ovládanie systému CRYPTOOL1024 je umožnené cez ikonu na pracovnej ploche, ikonu na systémovej lište, alebo cez „Štart“ v zozname programov, prípadne z príkazového riadku príkazom „CRYPTOOL1024“,
- vykonávané činnosti a udalosti v softvéri klienta CRYPTOOL1024 sú zaznamenávané do denníka udalostí Security auditu, ktorý je šifrovaný, lebo obsahuje aj citlivé informácie,
- systém má plnú kompatibilitu s kľúčovým hospodárstvom poskytovaným softvérom KRYPTOSERVIS pre Centrálnu Bezpečnostnú Autoritu (CBA.

Prístupové heslá pre autentizáciu používateľov s USB tokenom majú maximálnu dĺžku 8 znakov. Namiesto hesiel môžu byť použité aj PIN kódy s dĺžkou 4 až 8 číslic. To plne pre hardvérové zariadenie tokenu na viacfaktorovú autentifikáciu postačuje.

Celá bezpečnosť CRYPTOOL1024 systému CBA je vybudovaná v jadre Windows. Funkcie klienta CRYPTOOL1024 sú realizované pomocou natívnych funkcií jadra Windows z dôvodu bezpečnosti a s vylúčením buffrovania a kešovania spracovávaných súborov.

10.2.3 Filozofia potreby použitia nového šifrovacieho algoritmu

Symetrické šifrovacie algoritmy, používané v súčasnej dobe, ako sú AES a SEA64 (plnohodnotný nezdegradovaný predchodca GOSTu), používajú šifrovacie kľúče s maximálnou dĺžkou 256 bitov (32 bajtov). V dnešnej dobe, keď neustále narastá výpočtový vykon počítačov a spomínajú sa aj superpočítače a kvantové počítače, vzniká aj požiadavka na vývoj silnejších šifrovacích algoritmov, ktoré by mali tvoriť základ tzv. post-kvantovej kryptografie. V súvislosti s post-kvantovou kryptografiou sa predpokladá, že šifrovací algoritmus by mal pracovať s aspoň 1024-bitovým šifrovacím kľúčom, hlavne pre dosiahnutie vyššej kvality šifry, aby bol výsledný ***ŠT*** odolný aj proti rôznym vyspelým typom kryptoanalýz. Nie je problémom použiť aj kľúč s väčšou dĺžkou, avšak čím je kľúč dlhší, kladú sa pri spracovaním algoritmu čoraz väčšie nároky na čas spracovania procesorom. Z kryptografického hľadiska je výhodné použiť na realizáciu algoritmu Feistelovu schému a to z viacerých dôvodov, ktoré sú opísané v odborných publikáciách. Vhodným kompromisom medzi rýchlosťou spracovania algoritmu a dĺžkou kľúča je použitie práve kľúča s dĺžkou 1024 bitov.

Použitie kvalitného šifrovacieho algoritmu je podmienkou nevyhnutnou, avšak zďaleka nie postačujúcou. Na zabezpečenie bezpečnostných kritérií pri šifrovaní, stanovených celosvetovo uznávanými štandardmi je potrebné zabezpečiť viaceré factory pri návrhu a vývoji ako samostatného algoritmu, tak aj pri implementácii algoritmu do prostredia aplikácie a hlavne pri spôsobe realizácie kľúčového hospodárstva. Sila šifrovania je závislá okrem kvality algoritmu aj na kvalite generovaných kľúčov. Kvalitné kľúče musia byť generované nedeterministickým generátorom realizovaným na báze hardvéru. Dôležitým faktorom je použitie Centrálnej Bezpečnostnej Autority (***CBA***), ktorá generuje kľúče, dohliada na kvalitu generátora kľúčov, vykonáva štatistické testy generovanej postupnosti podľa štandardov (NIST 800-22) a FIPS, manažuje prácu s hardvérovými bezpečnostnými zariadeniami (tokenmi) a distribuuje kľúče na použitie v prostriedkoch s nainštalovaným šifrovacím softvérom.

Takto navrhnuté systémy je potom možné použiť pre jednotlivé stupne utajenia. Pre prvý stupeň utajenia “V” postačuje použitie realizácie šifrovacieho algoritmu v softvérovej forme. Avšak pre druhý stupeň utajenia „D“ a vyššie stupne sa požaduje realizácia šifrovacieho algoritmu v hardvérovom zariadení, napríklad v PCI prídavnej karte so signálovým procesorom.

Preto bol vyvinutý nový symetrický šifrovací algoritmus SEA1024 (Super Encryption Algorithm) s dĺžkou používaného šifrovacieho kľúča 1024 bitov. Je nasledovníkom algoritmu SEA64, ktorý je takisto realizovaný na báze Feistelovej schémy, ale s dĺžkou používaného kľúča len 256 bitov, podobne ako AES256. SEA1024 je kandidátom na použitie v post-kvantovej kryptografii.

10.2.4 Podporované druhy údajových médií

Z dôvodu bezpečnosti sa doporučuje šifrovať a dešifrovať súbory uložené na lokálnych harddiskoch počítača s nainšlatovanou aplikáciou CRYPTOOL1024. Je možné použiť aj USB pripojiteľné USB kľúče alebo USB točivé disky. Šifrovanie po sieti nie je z bezpečnostných dôvodov povolené. Povolená je len nasledujúca distribúcia zašifrovaných súborov kopírovaním po sieti. Pokiaľ za z bezpečnostno-profesionálnych dôvodov vyžaduje vysoká

bezpečnosť, tak sa aplikácia CRYPTOOL1024 inštaluje do počítačov oddelených od siete. A šifrované súbory sa distribuujú na dátových nosičoch. Na týchto počítačoch sa predpokladá, že sú nainštalované bezpečnostné nastavenia, čo však nie je podmienkou funkčnosti aplikácie CRYPTOOL1024, ale pokiaľ používateľovi záleží na bezpečnosti, tak je to nutnosťou.

10.3. Aplikácia WEBPROT ako príklad aplikácie, ktorá nemohla dostať certifikát od NBÚ SR

Bola to jedna z našich prvých aplikácií určená pre ***Úrad pre štátnu službu***. Požiadavky tejto inštitúcie boli do bodky splnené, bola nainštalovaná a sprevádzkovaná aj úspešná testovacia prevádzka. Aplikácia šifrovala sieťovú internetovú prevádzku medzi klientami a webovým serverom na dohodnutom porte pre danú webovú stránku, ktorá sa na strane klienta pripájala v štýle „***doména:číslo portu***“. Zároveň sa požadovala súbežná nešifrovaná prevádzka s inými webovými servermi. Pokúsili sme sa aj o našu prvú certifikáciu na novo zriadenom NBÚ SR. Dopadlo to ale s totálnym fiaskom. Pochválili nás, že sme vyvinuli perfektne fungujúci systém, ktorý je ale len detskou hračkou. A tak sme sa začali učiť, ako dosiahnuť splnenie podmienok stanovených ***BŠ***.

Dôvody, prečo systém aplikácie WEBPROT nemohol dostať certifikát:

- nemali sme ešte prístup k ***BŠ*** a tým pádom sme nepoznali požiadavky na vývoj aplikácie aspoň pre stupeň utajenia „***V***“ (prístup k ***BŠ*** vyžaduje bezpečnostnú previerku aspoň na stupeň “***D***”),
- ***BŠ*** nepovoľuje súbežnú nešifrovanú prevádzka po sieti akú mal WEBPROT,
- ***BŠ*** vyžaduje použitie viacúrovňového kľúčového hospodárstva na šifrovanie sieťovej prevádzky,
- ***BŠ*** vyžaduje generovanie kľúčového hospodárstva dobre zabezpečenou ***CBA***,
- nakoľko sme už v aktuálnej verzii použili ASE256, nepoužili sme striktnú požiadavku ***BŠ*** na použitie viacerých kľúčov, používali sme len dva kľúče na zašifrovanie veľkého objemu prenášaných dát v paketoch,
- použili sme na distribúciu a zavádzanie kľúčov vtedy čipové karty XICOR o ktorých sa čoskoro zistilo, že má v nich zadné dvierka americká NSA, takže sme následne museli prejsť na vlastné USB tokeny s bezpečnostným procesorom.

To boli hlavné dôvody aj na vznik prvej úspešne scertifikovanej verzie systému IPprot vyvíjanej v spolupráci s ***NBÚ SR***.

11. Záverečné porovnanie a zhodnotenie algoritmov

Na základe poznatkov získaných z vlastnej dlhoročnej praxe vo funkcii vývojára prostriedkov šifrovej ochrany informácií a z funkcie šifranta v silových zložkách som opísal použitie a skúsenosti s aplikáciou a používaním niektorých šifrovacích algoritmov v profesionálnej sfére. Veľa som sa naučil od skúsených expertov kryptológov pod vedením ktorých som zo začiatku vyvíjal zariadenia na šifrovanie dát pre štátny sektor. A z týchto poznatkov vyplývajú pre reálne použiteľné algoritmy AES a SEA v danej sfére nasledujúce skutočnosti:

11.1 Skutočnosti uvádzané pre použitie algoritmu AES

- Používať len AES 256, nikdy nie AES s kratším kľúčom (AES128, AES192),
- AES 128 a AES 192 sú z matematického hľadiska posudzované síce ako dostatočne silné, ale nie použiteľné na šifrovanie v ***PŠOI_OUS***, pretože tak stanovuje ***BŠ***,
- Pri použití AES 128 a AES 192 ale len na šifrovanie citlivých informácií, hrozí vzhľadom na malú dĺžku kľúča pri zašifrovaní jedným kľúčom nadmieru veľkého bloku ***OT***, nízka kvalita výsledného ***ŠT***,
- To isté, nízka kvalita ***ŠT***, hrozí aj pri použití AES256, avšak až pri vyšších objemoch šifrovaných ***OT*** dát s jedným šifrovacím kľúčom, nad cca 1 megabajt ***OT*** dát, preto je potrebné použitie viacerých šifrovacích kľúčov a toto obmedzenie striktne stanovuje ***BŠ***,
- AES256 sa javí zdanlivo ako kvalitný šifrovací algoritmus, ale hrozia spomínané úskalia,
- Pri použití AES256 na šifrovanie v ***PŠOI_OUS*** je potrebné používať rozsiahle kľúčové hospodárstvo s veľkým počtom kľúčov, na každý megabajt jeden ďalší kľúč a používať viacúrovňové kľúčové hospodárstvo pri šifrovaní streamov dát, tak ako je tomu pri šifrovaní sieťovej prevádzky. A tiež je potrebné vykonávať aj časový dohľad a meniť kľúč aspoň každých 10 minút, aj keď je kľúč nepoužitý na zašifrovanie objemu až do jedného megabajtu dát,
- AES256 je štandard verejného algoritmu, ktorý sa dnes používa prakticky všade,
- AES256 je však podozrivo spätý s americkou NSA, ktorá sleduje prakticky všetko a má zadné dvierka vo veľa zariadeniach, firmách, spoločnostiach zaoberajúcich sa kybernetickou bezpečnosťou,
- AES256 zabudovaný vo forme hardvéru aj do novších verzií procesorov je podozrivý a ***NBÚ SR*** ho zakazuje používať v ***PŠOI_OUS***,
- Pri porovnaní AES256 napríklad so SEA1024 je pri diferenciálnej analýze AES256 slabší, síce je dosť odolný, ale v oblasti korelačných testov diferenciálnej analýzy zaostáva za SEA1024, ktorému nahráva vyššia dĺžka kľúča a tým aj vyššia kvalita ***ŠT***, čo môže pri algoritme AES256 napomôcť útokom pomocou informácií získavaných z bočných kanálov,
- Verejným algoritmom, ako je AES, hrozí možnosť útoku na ***ŠT*** použitím napríklad Groverovho algoritmu na pokus o prelomenie šifry, hlavne v postkvantovej ére,
- Algoritmus AES v pôvodnej forme realizácie je náchylný na lokálny časový útok s pomocou keší procesora, preto je potrebné používať upravenú verziu s S-boxami vo

forme „Te“ a „Td“ tabuliek, čo je ochranou proti neinvazívnemu pasívnemu útoku - analýze doby výpočtu šifry **AES**,

- Ďalej napríklad v móde CBC hrozí možnosť lokálneho Waudenayovho útoku pri použití štandardných softvérových knižníc pre AES, v ktorých sa používa PKCS výplň,
- Pri prechode na algoritmus AES256 z algoritmu SEA64 v projekte systému na šifrovanie sieťovej prevádzky z dôvodu potreby certifikácie pre potreby EÚ a NATO vznikol na vtedajších starších počítačoch typu PENTIUM, poťažmo CELERON, časový problém pri spracovaní algoritmu AES a celý systém s trojúrovňovým kľúčovým hospodárstvom bol na hranici použiteľnosti, lebo šifrovanie spomalilo sieťové prenosy, hlavne pri prístupe väčšieho množstva klientov na servery a riešením bolo len použitie modernejších PC s novšími typmi procesorov,
- Taký istý problém s pomalosťou AES algoritmu nastáva pri použití na niektorých doskách priemyselných počítačov, napríklad NOVA 8522 s procesorom CELERON,
- Aj z algoritmu AES, sa zmenou S-boxov za iné, vlastné, napríklad armáda, pokúša vytvoriť si tým pádom už vlastne iný algoritmus, ktorý je možné považovať za vlastný utajený, ktorý už ale nebude verejným štandardom,
- Aj iné inštitúcie, ako napríklad Pentagon, CIA, NSA, FBI určite nepoužívajú na účely vysokého stupňa utajenia verejný algoritmus AES a majú svoje verzie, prípadne iné tajné algoritmy,
- Použitím neštandardu či už vytvoreného z AESu alebo iného algoritmu sa bráni útokom hrubou silou, pretože sa nevie o aký algoritmus sa jedná a teda ani jeden kľúč z celej množiny priestoru kľúčov nevyhovie, pretože každý pokus o dešifrovanie ***ŠT*** neuspeje,
- AES256 by sa mal používať len na utajenie citlivých informácií a kto chce mať vyššiu istotu pri ochrane svojich informácií, mal by používať súkromný algorimus.

11.2 Skutočnosti uvádzané pre použitie algoritmov SEA

- Pokiaľ by sme sa pohybovali na úrovni AES256, čiže s dĺžkou kľúča 256 bitov, tak už SEA64 algoritmus je rýchlejší a podľa expertov aj silnejší,
- Dlho ho používal verejný sektor pokiaľ nebol na svete AES, bolo veľa pokusov o jeho prelomenie či už hrubou silou alebo pomocou kryptoanalýz, ale doteraz sa ho nepodarilo skompromitovať a ani nevznikli návody na jeho prelomenie, možno len niečo na úrovni teórie, v praxi sa to neudialo,
- Zámenou iných tabuliek S-boxov, samozrejme kvalitných a vygenerovaných špeciálnym autorizovaným softvérom, ktorý vlastní napríklad ***ÚŠO***, vzniká už iný typ algoritmu SEA,
- Vlastnosti a predovšetkým kvalita SEA algoritmov závisia okrem dĺžky spracovávaného kľúča hlavne od S-boxov, čo sa týka hodnôt v tabuľkách a tiež rozsahu S-boxov, teda aj od ich veľkosti,
- ***KT rozbor*** analýzami všetkých typov SEA algoritmov, dokázal ich vysokú kvalitu a pre SEA1024 aj kvalitatívnu prevahu nad AES256 algoritmom pri vykonaní prvkov diferenciálnej analýzy v rámci KT rozboru SEA1024 algoritmu, kde pri zmene jediného bitu v referenčnom kľúči mal lepšie výsledky pri vyhodnocovaní v korelačných testoch, ďalšie testy a analýzy možno potvrdia aj viac,
- Vysoká kvalita ***ŠT*** potvrdzuje, že pri šifrovaní veľkých objemov ***OT*** nie sú potrebné reštrikcie stanovené podľa ***BŠ*** a využíva to hlavne CRYPTOOL512 a CRYPTOOL1024 na šifrovanie veľkých súborov, aj rádovo v Gigabajtoch pomocou jediného kľúča (512 alebo 1024 bitov),

- Utajiť postačuje len S-boxy, ktoré sú v našich aplikáciách uložené v bezpečnostnom procesore tokenu,
- S-boxy algoritmov SEA sa považujú za dlhodobo platný kryptografický prvok kľúčového hospodárstva,
- Na utajené algoritmy nie je možné použiť Groverov útok, totižto nikto nezrealizuje dešifrovanie s akýmkoľvek kľúčom,
- Kto chce mať vyššiu istotu pri ochrane svojich informácií, mal by používať súkromný algoritmus, napríklad z radu SEA, a aby získal aj vysokú odolnosť voči útokom pomocou kryptoanalýz, tak by mal použiť algoritmus SAE1024, prípadne iný kvalitný súkromný algoritmus, čo je zrejmé aj z predchádzajúceho popisu,
- Čím menej informácií existuje o použití súkromného algoritmu, tým je hrozba úspechu pri pokuse o prelomenie menšia,
- Čo sa týka algoritmov SEA, existuje kompletná technická dokumentácia s ich matematickým popisom, diagramami, KT rozborom s jeho výsledkami, S-boxami, príkladmi ich realizácie naprogramovaním vo viacerých programovacích jazykoch a so vzorovými vektorovými testami a benchmarkmi,
- Táto dokumentácia je z pochopiteľných dôvodov utajovaná a utajované sú tiež S-boxy,
- S-boxy zavádza do bezpečnostného tokenu CBA,
- Šifrovacie kľúče zavádza do bezpečnostného tokenu tiež CBA,
- Samozrejme, je na používateľovi, ako prevádzkuje celý proces šifrovania a kľúčové hospodárstvo, to bolo popísané už v predchádzajúcich kapitolách,
- Pokiaľ sa striktne nedodržia predpísané pravidlá pre kryptografiu symetrického šifrovania, a to platí pre AES256 aj SEA algoritmy, potom naozaj nemá šifrovanie význam a šifra sa stáva prelomiteľnou hlavne na základe získaných postranných informácií z bočných nezabezpečených kanálov, čo už tiež bolo popísané a vysvetlené,
- Použitie súkromného algoritmu prináša pri ochrane informácií bezpečnostné benefity.

12. Použitá odborná literatúra a dokumentácia

Väčšina použitej a stále používanej literatúry a ostatnej dokumentácie pri vývoji je vytvorená na báze vývojárskej technickej dokumentácie z oblasti vývoja hardvéru aj softvéru. To platí aj pre napísanie tejto štúdie. V dnešnej dobe sú informácie získavané hlavne cez internet zo stránok na serveroch dôveryhodných technologických spoločností, ako sú Microsoft, Texas Instruments, Intel, IBM, Siemens a pod. Za 45 rokov bolo preštudované množstvo dokumentácie a možné je uviesť aspoň príklady, avšak veľké množstvo kryptografickej dokumentácie a zadaní sa uchováva v režime utajenia a nie je možné to uvádzať. Napríklad je to dokument ***BŠ*** a množstvo manuálov a ďalšej dokumentácie, na ktorú sa ***BŠ*** odvoláva.

Všetky vývojové prostriedky s príslušnými dokumentáciami a ďalšiu technickú dokumentáciu je možné rozdeliť do viacerých oblastí a časových období použitia. Ako príklady uvediem len základ, na ktorom sa začínalo.

12.1 Vývojové prostriedky a dokumentácia z obdobia MS DOS a Windows 95 až Millenium operačných systémov

1. Microsoft: *Microsoft Macro Assembler 5.1*, for MS-DOS and Microsoft Windows, USA, Redmond, Washington

Verzia 6.0, vydaná v roku 1991, okrem už existujúcich záznamov podobných vysokej úrovni pridala okrem už existujúcich záznamov podobných vysokej úrovni aj "invoke" a niektoré ďalšie konštrukcie podobné vysokej úrovni. 6.0 bolo možné spustiť na procesore 8086, ale dokázali generovať plochý 32-bitový 386 kód. V roku 1992 bola vydaná verzia 6.1, ktorá pridala podporu pre formát COFF používaný v systéme Windows NT a odstránila podporu pre OS/2. 6.1 bol vytvorený ako bimodálny binárny súbor pred dokončením rozhrania Win32 API a je nekompatibilný so spustením v systéme Windows NT z dôvodu chýbajúcich exportov.

Používal sa hlavne na vývoj VxD ovládačov, virtuálnych zariadení Windows 95, 98 až Millenium. Špecialitou bolo, že musel byť použitý inkrementálny linker. Vyvíjal som v ňom VxD modul „Ism.VxD“ s implementovaným algoritmom SEA64 a X76F.VxD, čipkartový drajver pripojiteľný donglom na paralelný LPT port pre čipkarty X76F64.

2. Microsoft Press, Walter Waney: *Systems Programming for Windows 95*, A Division of Microsoft Corporation, Redmond, Washington 98052-6399

Odborná publikácia pre expertov v oblasti programovania VxD ovládačov, tiež na báze PNP a systémových modulov pre Windows 95 a ďalšie verzie Windows vyvíjané ešte pred príchodom Windows NT s novou technológiou jadra ***OS***.

3. Manley P. Ludwig: *IC master*, katalóg integrovaných obvodov, rok vydania 1978

Prvý 2175 stranový kompletný katalóg integrovaných obvodov z celého sveta používaný v začiatkoch nášho vývoja v oblasti mikroprocesorovej techniky ktorá sa vtedy začínala rozvíjať v rámci celého sveta.

4. Charles Petzold: *Programovanie vo Windows*, CPress, Legendárna publikácia o programovaní Win32 API

Odborná publikácia pre programovanie Win32 rozhrania v používateľskom režime.

12.2 Odborná literatúra z obdobia začiatkov NT operačných systémov

5. Computer Press, Jeffrey Richter: *Windows pre pokročilých a expertov, 32 bitové programovanie*, prvé vydanie v Microsoft press

   Veľa z popisu tejto technológie platí aj pre 64 bitové programovanie. Ako príklady publikuje aj zdrojové kódy popisovaných aplikácií na demonštráciu 32 bitového API s natívnymi funkciami jadra ***OS***.

6. Microsoft Press, David A. Solomon: *Windows NT pre administrátorov a vývojárov*

Oficiálny sprievodca architektúrou a jadrom operačného systému.

12.3 Dokumentácia a vývojové prostriedky z obdobia Windows NT až Windows 11 operačných systémov

Začínalo to ako **Microsoft Developer Network (MSDN)** - Sieť vývojárov spoločnosti Microsoft. **MSDN** bola divízia spoločnosti Microsoft zodpovedná za riadenie vzťahov firmy s vývojármi a testermi, ako sú vývojári hardvéru, ktorí sa zaujímajú o operačný systém a vývojári softvéru vyvíjajúci na rôznych platformách ***OS*** alebo pomocou API prípadne skriptovacích jazykov aplikácií spoločnosti Microsoft. Riadenie vzťahov sa odohrávalo v rôznych médiách: webové stránky, konferencie vývojárov, obchodné vzťahy, blogy a distribúcia DVD. Od januára 2020 bola webová lokalita plne integrovaná s ***Microsoft Docs***, ktorá bola v roku 2022 integrovaná do služby ***Microsoft Learn***. Tam bolo získané obrovské množstvo informácií a prostriedky vývoja, ako Visual Studio s množstvom ***OS*** a ich verzií na vývoj a testovanie.

Dôležité informácie boli čerpané aj z RFC dokumentácie. **RFC** je skratka z anglického výrazu (**request for comments** – žiadosť o komentáre), ktorá sa používa pre označenie radu štandardov a ďalších dokumentov opisujúcich internetové protokoly, systémy a pod. Ako už názov napovedá, RFC sú oficiálne považované skôr za odporúčania ako normy v tradičnom zmysle, napriek tomu sa podľa nich riadi väčšina internetu.

Jednotlivé RFC dokumenty vydáva editor RFC podľa príkazov Internet Architecture Board. Každé RFC má pri zverejnení pridelené číslo. Žiadne vydané RFC sa nikdy neruší, len sa v budúcnosti môže upraviť vydaním novšieho RFC. Všetky RFC sú k dispozícii na stiahnutie na viacerých miestach na internete. Každé RFC je dostupné v podobe čistého ASCII textu (v angličtine). Na rozdiel od klasických noriem a štandardov vydávaných klasickými normotvornými inštitúciami (ako napr. ISO, ANSI a pod.) vznikajú RFC trochu iným spôsobom. Pôvodnými autormi jednotlivých RFC sú zvyčajne konkrétni experti, ktorí sa snažia riešiť konkrétny problém, ktorého riešenie ponúknu vo forme návrhu RFC internetovej verejnosti (ako tzv. Internet Draft). Ak je dané riešenie (často už dobre fungujúce v rámci nejakej pilotnej prevádzky) uznané za prínosné, dokument sa vydá ako RFC. Toto pragmatické riešenie štandardov zostavovaných jednotlivcami alebo malými skupinami na základe praktických skúseností má mnohé výhody oproti formálnejším procesom štandardizačných komisií pri úradoch typu ISO. Štandardy vytvorené pomocou RFC sú (vzhľadom na absenciu akejkoľvek skutočnej moci na ich presadzovanie) až na výnimky dodržiavané, pričom pomohli rozšíreniu IT do dnešných celosvetových rozmerov.

12.4 Dokumentácia z vývoja šifrovacích algoritmov, PŠOI a realizovaných produktov

7. Salutis systems: *CRYPTOOL512 Professional v2.0, Špecifikácia produktu,* Kvantovo bezpečná kryptografia na šifrovanie súborov
8. Salutis systems: *CRYPTOOL512 Professional v2.0, Silné šifrovanie súborov systémom CRYPTOOL512 Professional – Používateľský Manuál,* Kvantovo bezpečná kryptografia na šifrovanie súborov

9. Salutis systems: *Cryptool512 CBA v4.3 – Dokumentácia pre generovanie kľúčov v CBA*
10. Salutis systems: *Cryptool512 CBA v4.3 – Databáza kľúčových médií pre automatizované funkcie v CBA*
11. Salutis systems: *Šifrovací algoritmus SEA512, Špecifikácia produktu - základné údaje*
12. Salutis systems: *Šifrovací algoritmus SEA512, Špecifikácia produktu,* popis šifrovacieho algoritmu a jeho použitia
13. Salutis systems: *CRYPTOOL1024 Professional v2.0, Špecifikácia produktu,* Kvantovo bezpečná kryptografia na šifrovanie súborov
14. Salutis systems: *CRYPTOOL1024 Professional v2.0, Silné šifrovanie súborov systémom CRYPTOOL1024 Professional – Používateľský Manuál,* Kvantovo bezpečná kryptografia na šifrovanie súborov
15. Salutis systems: *Cryptool1024 CBA v4.3 – Dokumentácia pre generovanie kľúčov v CBA*
16. Salutis systems: *Cryptool1024 CBA v4.3 – Databáza kľúčových médií pre automatizované funkcie v CBA*
17. Salutis systems: *Šifrovací algoritmus SEA1024, Špecifikácia produktu - základné údaje*
18. Salutis systems: *Šifrovací algoritmus SEA1024, Špecifikácia produktu,* popis šifrovacieho algoritmu a jeho použitia

Toto je kompletná dokumentácia prostriedkov šifrovej ochrany informácií vyvinutých na báze nových šifrovacích algoritmov SEA512 a SEA1024 a kompletná technická dokumentácia popisujúca tieto algoritmy z hľadiska ich matematiky a tiež ich realizácie vo forme zdrojových textov a celých softvérových projektov s implementáciou do vzorových testov a benchmarkov.

19. Salutis systems: *Komparatívna analýza šifrovacích algoritmov SEA a AES*

Porovnanie šifrovacích algoritmov z hľadiska dĺžky šifrovacích kľúčov.

20. Salutis systems: *Kryptologicko - technický rozbor symetrického šifrovacieho algoritmu SEA1024*

KT rozbor vykonaný pomocou kryptoanalýz, výsledky ktorých odzrkadľujú kvalitu analyzovaného šifrovacieho algoritmu. Pre porovnanie bol vykonaný aj KT rozbor algoritmu AES256.

12.5 Dokumentácia z vývoja hardvéru

Na vývoj hardvéru na báze signálového procesora bola použitá dokumentácia popísaná v predchádzajúcich kapitolách. Vývoj ostatného hardvéru, časť ktorého bola popísaná v predchádzajúcich kapitolách, vyžadoval štúdium kvanta odbornej literatúry hlavne z odboru mikroprocesorovej techniky, datasheetov použitých typov čipov, vývojových systémov a technológií procesorov a konštrukcie počítačových systémov na báze PC.

13. Záver

Cieľom bolo ukázať, že nestačí použiť len kvalitný šifrovací algoritmus, čo je síce podmienka nevyhnutne potrebná, ale je zďaleka nepostačujúca. V predchádzajúcich kapitolách bolo popísané, ako je možné algoritmus implementovať a ako vybudovať celú bezpečnosť okolo šifrovania informácií. A hlavne bolo cieľom dostať do povedomia, čo všetko je potrebné vykonať na to, aby sa šifrovalo bezpečne a malo to vôbec zmysel. To čo sa na šifrovanie používa verejne, je nie zďaleka bezpečnostne konkurenčné skutočným ***Prostriedkom šifrovej ochrany informácií***, pretože sa nedodržiava viacero zásad kybernetickej bezpečnosti. Napríklad, používatelia síce pomocou TLS layerov medzi sebou alebo pri komunikácii so servermi, šifrujú, ale na serveroch je to už zasa v otvorenej forme a u klientov taktiež. A NSA to sleduje, pretože to odôvodňuje zákonným nariadením. Prípadne to nikto nikdy nezistí a ani nevie. NSA už špionážou zničila viacero zahraničných firiem, keď sa prevalilo, že z NSA im bolo nariadené implementovať „***zadné dvierka***“ do ich produktov. A to aj do prostriedkov ***ŠOI***. Jednoznačne je preto potrebné pre pocit skutočného súkromia, mať vlastný šifrovací algoritmus. A tiež mať aj vlastné aplikácie na šifrovanie a dodržať pravidlá, ktoré boli popísané v predchádzajúcich kapitolách. Túto štúdiu dopĺňa firemná technická a odborná dokumentácia, ktorá bola spomenutá v predchádzajúcej kapitole.