Videó titkosítása

BME - TMIT
VITMA378 - Médiabiztonság
feher.gabor@tmit.bme.hu

Titkosítás és adatrejtés

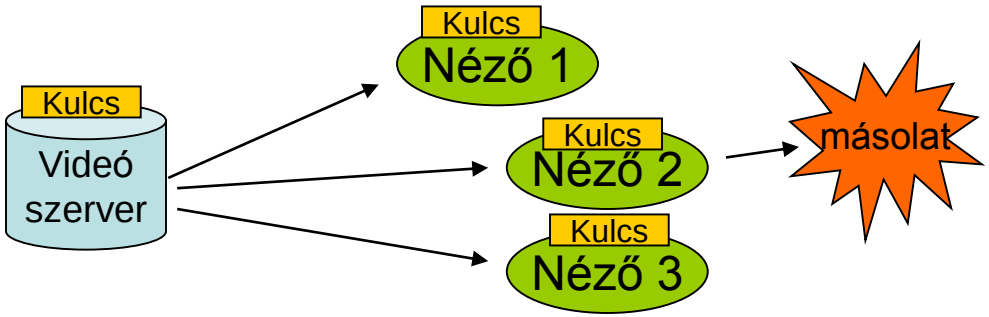
- Steganography – Fedett írás
- Cryptography – Titkos írás

- Adatrejtés
  - Az adat a szemünk előtt van, csak nem vesszük észre
- Titkosítás
  - Az adatot titkosítjuk. Látszik, hogy titkos, képtelenek vagyunk visszafejteni

- A jó titkosítás és adatrejtés esetén az algoritmus közismert (nem titkos)
  - A kulcs titkos, nagyon sok lehetőség
  - Az titkosító/adatrejtő algoritmus egyszerű – valós idejű futás
  - Matematikailag bonyolult, időigényes feltörés – cél, hogy csak kimerítő kulcskereséssel lehessen megoldani

Videófolyam titkosítása

- Alkalmazása
  - Kódolt TV adás
    - Video on Demand
    - Élő közvetítés
    - DVB
  - Video-telefon
  - Titkos közvetítés



- A vevő készüléke feloldja a titkosítás (szükséges a megjelenítéshez)
  - Készülhet titkosítás nélküli másolat

Videófolyam titkosítása - célok

- Megvalósítási célok
  - Valós időben kódolás és dekódolás
    - A videófolyamot nem tudjuk/akarjuk hosszan tárolni
  - Ne növelje túlságosan a folyam méretét
    - Növelő tényezők lehetnek: kulcskezelés, jelzések (marker), redundancia, blokkos kódolás
  - Ne okozzon hibás videó dekódolást
  - Biztonságos legyen
    - Különböző alkalmazásoknál különböző biztonsági szintek
    - Katonai alkalmazásnál elvárható a teljes felismerhetetlenség
    - TV adásnál nem gond, ha csak annyira titkosított, hogy élvezhetetlen az adás. Sokszor jobb is, mert így a nem előfizető motivált az előfizetésre

Algoritmusok

- Naiv algoritmus
  - Nem érdekel milyen az adat, mindent tikosítunk
- Szelektív titkosítás
  - Multimédia specifikus
    - Csak bizonyos képkockákat titkosítunk
    - Pl.: csak I képek
  - Formátum specifikus
    - DCT komponensek
    - Tömörítési eljárások (EC, DPCM)

Video scrambling

- A videó jel összezavarása
  - Nagyon korai technológia, nagyon gyenge védelemmel
  - A kábelTV esetén használták (analóg inkább)
- Technológiák
  - Analóg
    - Frekvencia konverzió
    - Zavaró jel
  - Digitális
    - Helyettesítés
    - Eltolás

Naiv algoritmus (1995)

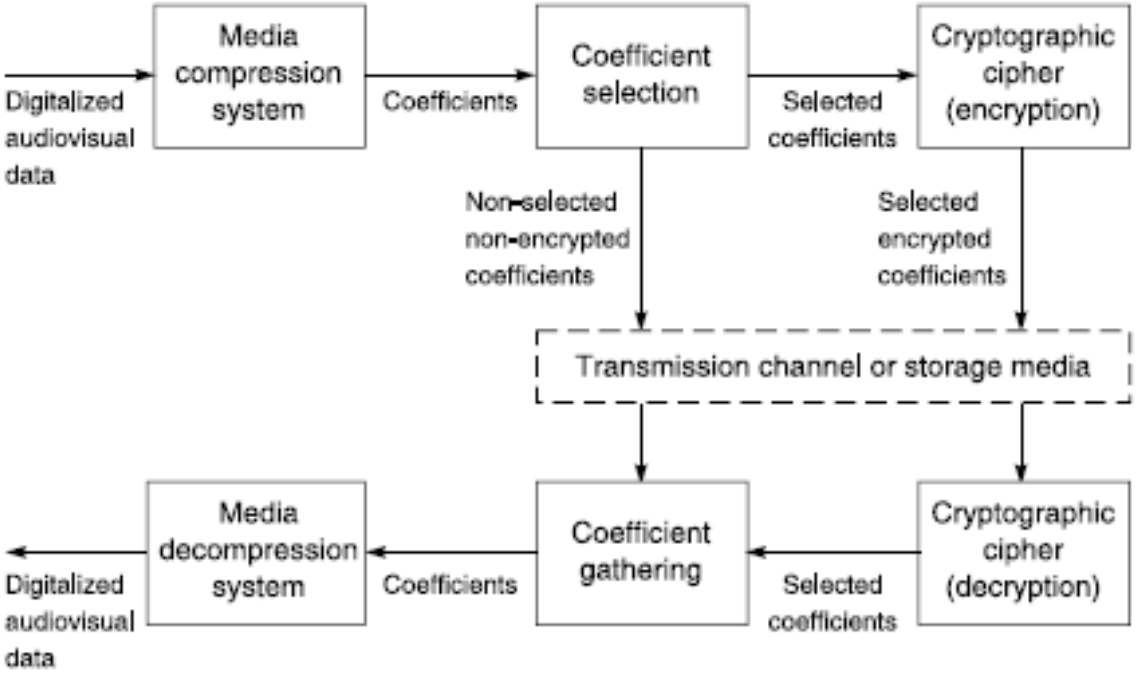
- A videófolyamot nem különböztetjük meg más adattól
  - Nem használjuk ki a videófolyam előállításánál használt eljárásokat
  - Nem használjuk ki a videófolyamok szerkezet felépítését
- Minden ismert titkosító használható
  - Tradicionális: DES; Korszerű és gyors: AES, RC4, …
  - Érdemes folyamtitkosítót használni, hogy ne okozzon növekedést a folyamban (mint blokkos kódolás esetén)
  - Annyira biztonságos, mint az alkalmazott titkosítás. A kulcs még ismert nyílt szöveg esetén sem fejthető vissza
- Hátránya, hogy időigényes (mindkét oldalon)
- Alkalmazása
  - Nincs extrém sávszélesség
  - Maximális biztonság kell

Permutációs algoritmus

- Pure permutation
  - A folyam bájtjait álvéletlen sorozat alapján permutáljuk
  - A permutálás egyszerűbb művelet, mint a titkosítás
- Hátránya, hogy kevésbé biztonságos
  - Ha van ismert videó-részlet, akkor a permutáció megfejthető

Szelektív algoritmusok

- Csak bizonyos adatrészek titkosítása



SECMPEG (1995)

Jürgen Meyer and Frank Gadegast

- Secure MPEG
  - 4 szintű biztonság
    - 1. szint:Csak a fejlécek + MV
    - 2. szint:Releváns I blokkok (DC + 3-8 AC)
    - 3. szint:Minden I kép és I blokk
    - 4. szint:Minden
  - Titkosítás a DES algoritmus segítségével
    - A teljes titkosítás az 50%-a videó dekódolásnak
    - A fejlécek az entrópia kódolás előtt vannak, így azt el kell távolítani
- Integritás védelmével is foglalkoztak
  - CRC kód használata

AEGIS titkosítás (1995)
T. B. Maples és G. A. Spanos

- Csak az I képkockák titkosítása + fejlécek
  - A GOP –tól függően a videófolyam valamekkora része
  - Pl.: MP4, I képek minden 24. képkockában, I képek mérete: 13-21%
- A teljes I képet titkosítom
  - Az összes itt található információt
  - Az illegális vevő nem tudja az I képeket dekódolni

I képek titkosítása 2.

- A P és B képek is hordoznak információt
  - Mozgás után a mozgáskompenzáció blokkjai

- TV adás titkosítására még megfelelő
- Ha növelem az I képek gyakoriságát
  - Javul a titkosság (csak bizonyos mértékben)
  - Romlik a hatékonyság, egyre több adatot titkosítok

Zig-zag algoritmus (1996)

Tang

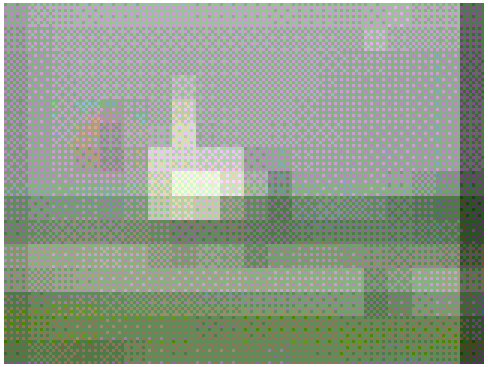
- Zig-zag permutáció
  - Az ötlet az, hogy a DCT együtthatók cikk-cakk összegyűjtése helyett használjuk álvéletlen permutációt
    - A DC komponens a legnagyobb érték, ezt úgy tüntetjük el, hogy kettévágjuk (4felső – 4 alsó bit) és az alsó biteket a legutolsó AC komponens helyére tesszük
    - Még nagyobb biztonság, ha előbb a DC értékeket titkosítjuk
    - Még további biztonság, ha több permutációs listát alkalmazunk

Zig-zag algoritmus 2.

- Hátrány:
  - Az algoritmus hatására a DCT együtthatók kevésbé tömöríthetőek, nő a folyam mérete
  - Az algoritmus könnyen feltörhető
    - Ismert videó-részlet esetén (pl. intró) a permutáció kitalálható
    - Ha több permutáció van, akkor is az elrendeződésből kitalálható, melyiket alkalmazták
    - A DC komponensek elhagyásával még mindig nézhető marad a videó (de nagyon rossz minőség)

Zig-zag algoritmus 2.

- Hátrány (folyt.)
  - Feltörés ismert videó-részlet nélkül
    - A DCT szerkezet sejthető: Nagy értékek a bal felső sarokban, pici, közel 0 értékek a jobb alsóban
    - Statisztika készíthető a DCT blokkokról
    - A DC és a bal felső AC értékek helye megtalálható viszonylag kis számú kombinációval



Csak DC

DC + 2 AC

DC + 5 AC

Video Encryption Algorithm (1997)

Lintian Qiao & Klara Nahrstedt

- MPEG folyamok statisztikus vizsgálata
  - Digrammok a folyamban előforduló bájt-párok
    - Lehetnek / távolságra. Legegyszerűbb az egymás utáni bájtokat vizsgálni
  - Ha a digrammok egyenletes eloszlásúak és egy kis tartományban csak kis valószínűséggel ismétlődnek, akkor a digramm egyik bájtjából nem tudunk következtetni a másikra
    $\Rightarrow$ Elegendő csak az egyiket titkosítani
  - Ha nem nézzük a fejléceket, akkor teljesül az egyenletes eloszlás, kis valószínűségű ismétlődés (1/16 I képkocka esetén 3% valószínűséggel ismétlődik)

Video Encryption Algorithm 2.

- Video Encryption Algorithm: VEA
  - Bájt-párokat alkotunk
    - egymás után lévő bájtok: a_1-a_2, a_3-a_4, …

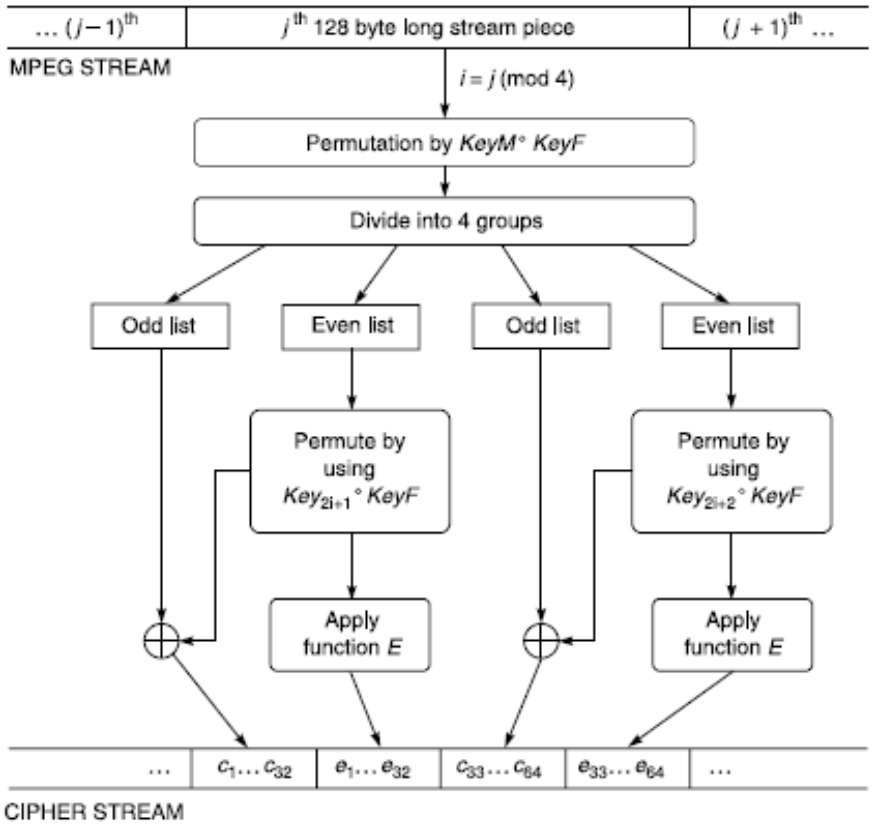
    vagy
    - I kép egyik fele és másik fele: a_1-$a_{n/2}$, a_2-$a_{n/2+1}$
  - Új értékek kizáró vagy kapcsolattal származtatva:
    - $c_1=a_1$ XOR a_2, $c_2=a_3$ XOR a_4, …
    - Az XOR művelet nagyon könnyen számítható
  - Elegendő az XOR művelet eredményét és az egyik bájtot titkosítva továbbítani:
    - $c_1,c_2,\ldots,c_{n/2},E(a_1,a_3,\ldots a_{n-1})$
  - A költséges titkosítás felét megspóroljuk! A hatás ugyanaz, mintha az egész I képet tikosítottuk volna
- A megoldás csak akkor működik helyesen, ha nincs ismétlődő minta. Ez MPEG2 esetén csak az 1/16 I képkockára igaz.
  - A 8 (16 képkocka darab fele) darab titkosítást más kulccsal kell megtenni.

Video Encryption Algorithm 3.

- Az eljárás támadható, ha a támadó ismeri a titkosítás egyik felét:
  - Támadó ismeri a_1, a_3, a_5… -t
  - c_1, c_2, c_3, … -ből ki tudja számolni a_2, a_4, … -t
- A biztonság javítható, ha a bájt-párok egy álvéletlen alapján állnak elő
  - Pl.: 256 bites álvéletlen sorozat 128 db 0 és 1 bittel
  - A 0 és 1 érték szerint páros vagy páratlan listába sorolás
  - A támadó így nem tudja, melyik érték hova tartozik

VEA kulcsok

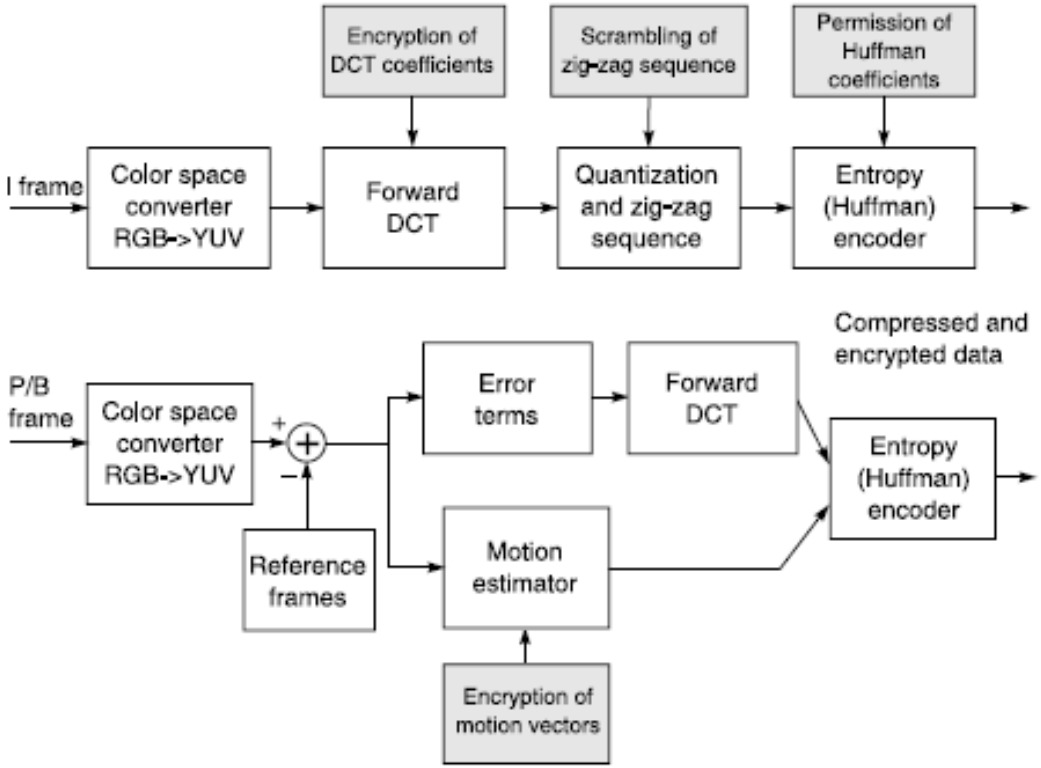
- KeyM, KeyF, Key1-8



Video Encryption Algorithm (1998-9)

Shi & Wang & Bhargava

- Algorithm I-IIVEA
- Algorithm IIIMVEA
- Algorithm IVRV



VEA Algorithm I

Shi & Wang & Bhargava

- Az I képek Huffman kódjainak permutációja
  - Azért, hogy ne legyen gond a tömörítéssel, csak azonos hosszú kódszavak cseréje
- Támadható
  - Ismert nyílt szöveg esetén a permutáció megfejthető
  - Csak titkosított szöveg esetén a permutációnál a gyakori részek megfejthetőek. A fontos DC és AC komponensek felderíthetőek

VEA Algorithm II

Shi & Wang & Bhargava

- A DCT együtthatók előjelei vannak titkosítva
  - GOP méretű titkos kulcs segítségével XOR művelet (szinkronizáció)
- A kulcs mérete meghatározza mennyire biztonságos

VEA Algorithm III - MVEA

Shi & Wang & Bhargava

- A 2. algoritmus kibővítése
  - A mozgásvektorok előjeleit is titkosítjuk
    - A mozgásvektorok differenciálisan vannak kódolva
    - Torzítottabb képek
  - Ebben az esetben már nem szükséges az összes DCT együttható előjelének titkosítása
    - Csak a DC értékek titkosítása

- Támadások
  - A DC és AC értékek összefüggésben vannak. Az AC értékekből kikövetkeztethető a DC érték
    - Ebben az esetben érdemes az összes DCT együtthatót titkosítani

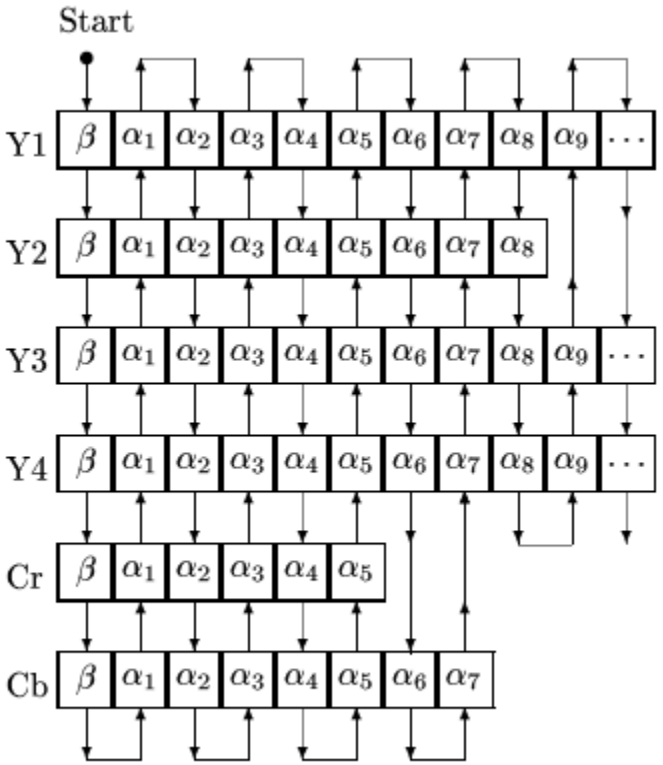
VEA Algorithm III - RVEA (1999)

Shi & Wang & Bhargava

- Real-Time Video Encoding Algorithm: RVEA
  - Nem kell az összes információt kódolni, elegendő csak a legfontosabbakat
    - DC kódolásánál a differencia előjele
    - AC együtthatók előjelei
    - Mozgásvektorok előjelei (szintén differenciális kódolás)
    - Elegendő a legfontosabb 64 ilyen érték
  - Titkosítás:
    - Kódoljuk az előjelek sorozatát, a kódolt értéket tároljuk előjelként
    - Blokk titkosítás használata!

RVEA algoritmus 2.

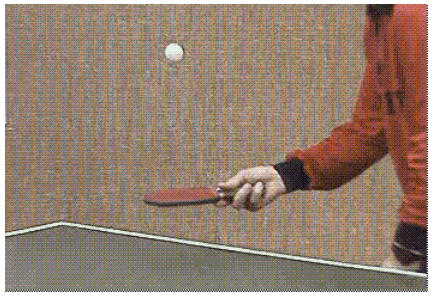
- Titkosított bitek egy makró-blokkban:
  - I kép: a 64 DC és AC előjel bit
  - P kép: x és y mozgás előjelei, 62 DC és AC előjel bit
  - B kép: x és y mozgások előjelei, 60 DC és AC előjel bit



B: DC differencia
α: AC komponensek

RVEA algoritmus 3.

- A titkosítás hatásfoka



Kódolatlan

MVEA

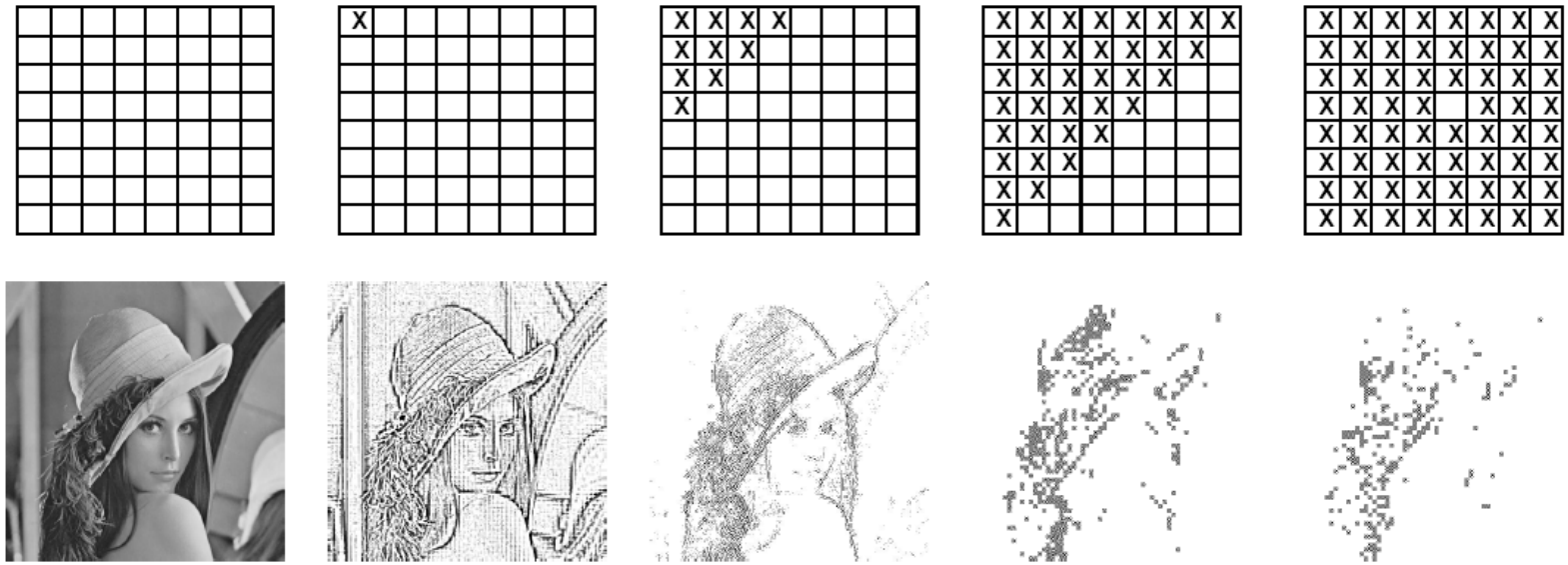
RVEA

- Mivel keveset kódol, ezért nagyon gyors
- Az eredeti kép nem állítható vissza (túl erőforrás-igényes)

RVEA algoritmus 4.

- Törés:
  - Ha az AC előjeleket pozitívra állítjuk és DC értéke 128, akkor a kép felismerhetővé válik (első kép)
  - Szintén felismerhető még, ha a fontos AC értékeket a törésnél 0-ra állítjuk (további képek, 2, 3 és 4 legfontosabb helyérték)

Titkosított DC és AC értékek



- DC és AC értékek hiányában is nézhető marad a kép
  - Az x jelöli a titkosított (most ismeretlen) értékeket
    - Helyettesítés: DC: 128, AC: 0

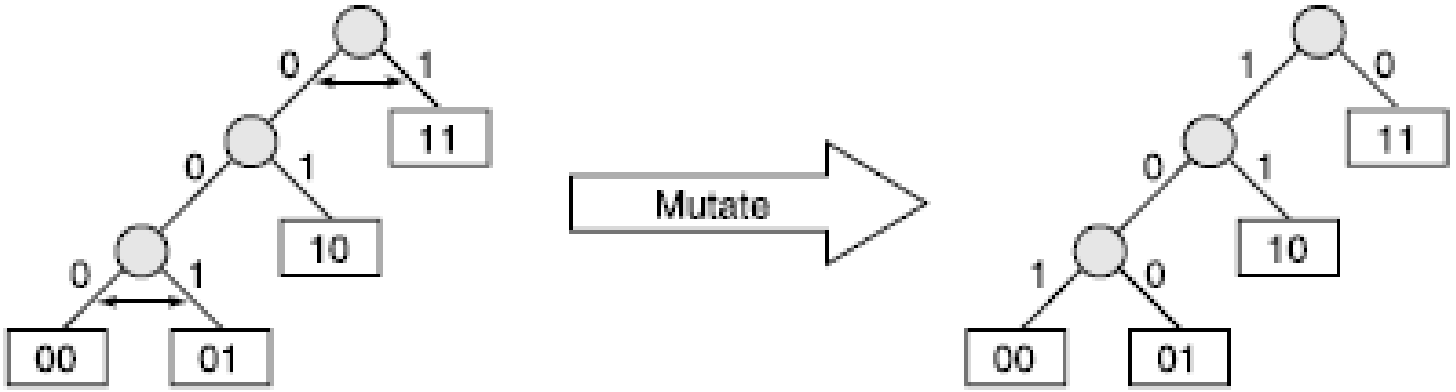
MHT-Encryption Scheme (2000)

Wu and Kuo

- Titkosított entrópia kódolás
  - Multiple Huffman Tables
- A felhasznált Huffman tábla kulcs alapú cseréje
  - Titkosítás
    - 2^k Huffman tábla generálása
    - Az aktuális szimbólum az 1…k véletlen sorszámú táblával lesz kódolva
  - Hogy ne változzon a videó mérete, azonos teljesítményű táblázatok kellenek
    - Általános, optimális táblázatok használata
    - Huffman fa mutáció

Huffman fa mutáció

- Egyes szimbólumok kódszavának a cseréje (mutáció)
  - Egy kulcs alapján a fában megváltoztatjuk az ágakat
  - Nem változik az optimálisság

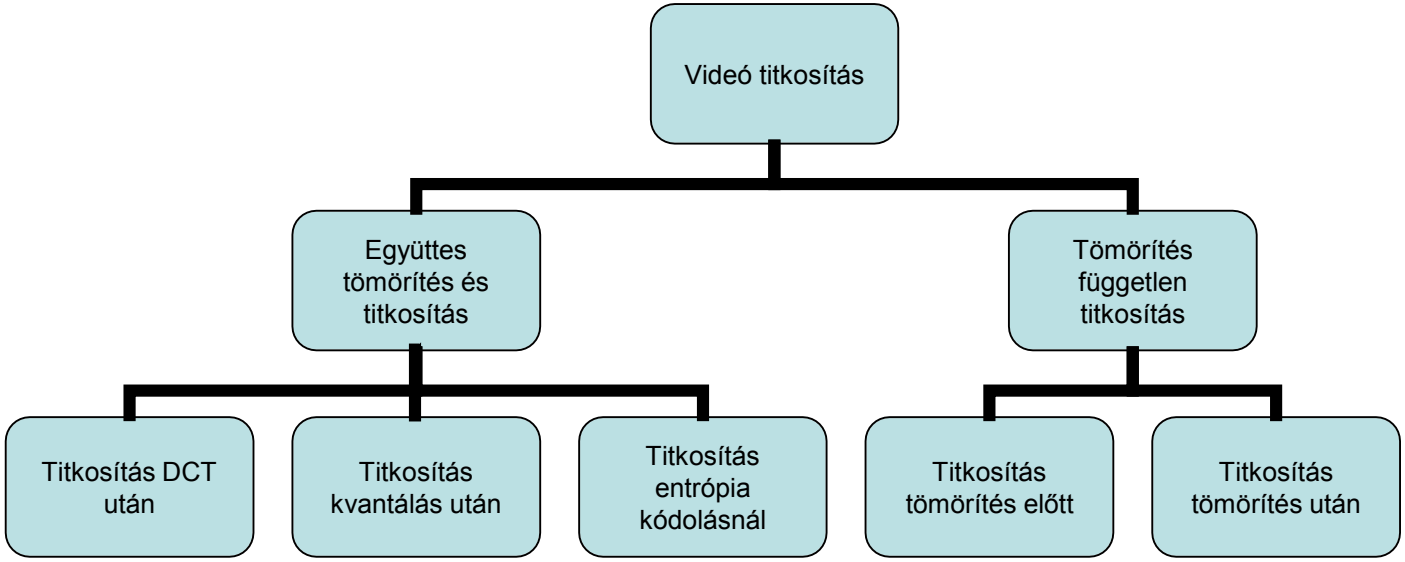


Format-Compliant Configurable Encryption (2002)
Wen és társai

- Lényeges lehet, hogy a videófolyam megőrizze értelmezhetőségét
- Titkosítás
  - Az adatokat két részre lehet osztani
    - Információt hordoz
    - Nem hordoz információt
  - Amik információkat hordoznak, azokat a biteket összeszedi és együtt titkosítja
  - A titkosított bitek visszakerülnek a folyamba
  - Gond lehet VLC értékekkel (FLC is néha)
    - Ekkor előbb indexelni kell ezeket az értékeket és az indexet kell titkosítani, majd e szerint kell visszaírni az értékeket

Videó titkosítások

- Videó titkosítás lehetséges helye
  - Nyers videó (tömörítés előtt)
  - Tömörítés közben
  - Tömörített videó (tömörítés után)

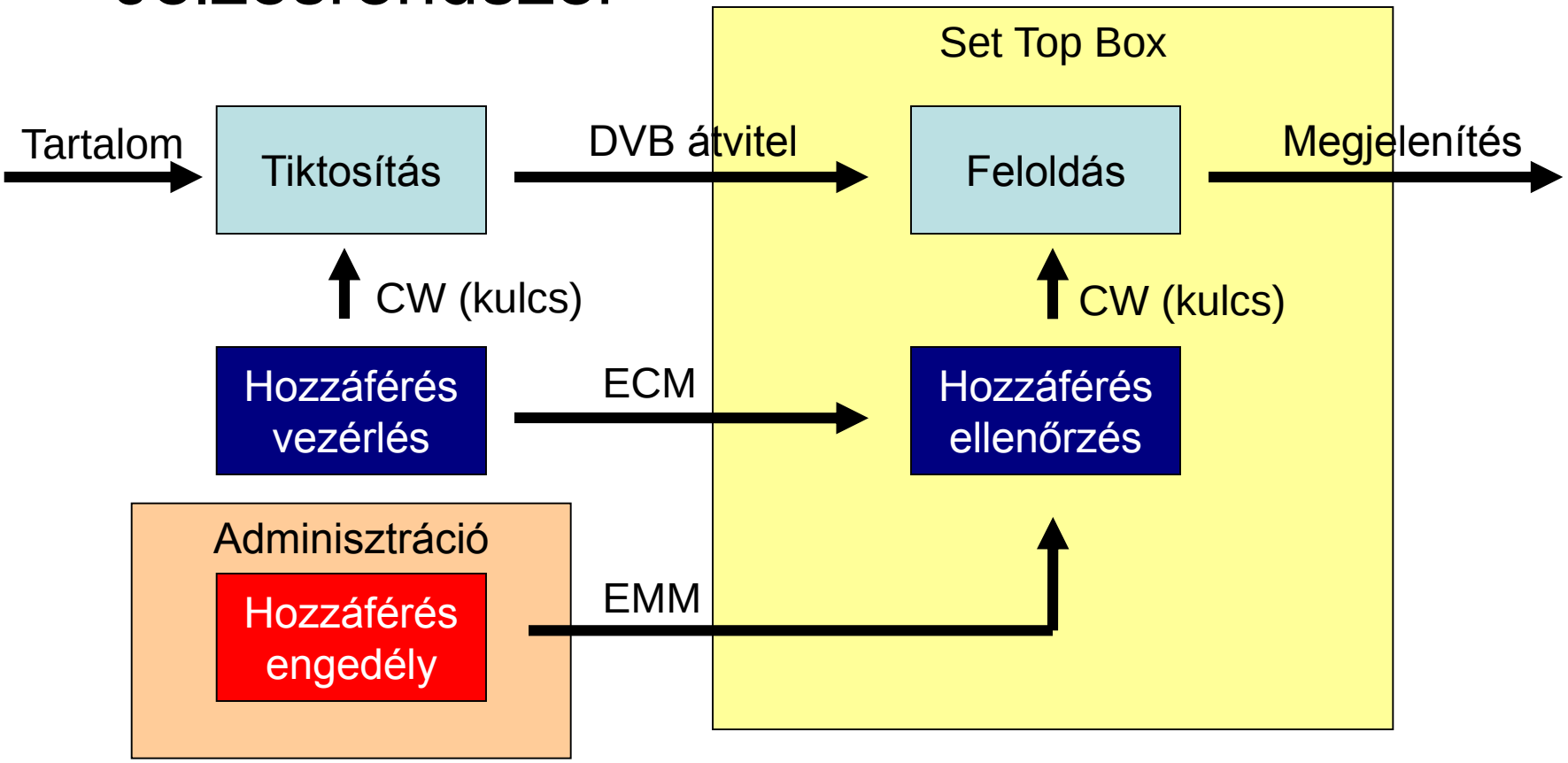


DVB-C/S/T/H megoldások

- DVB CAS - DVB CSA - DVB -CI
- CAS: Hozzáférés korlátozó rendszerek
  - CAM: Conditional Access Module
    - PCMCIA szabványú bővítő
    - Kódkártya olvasás
    - CAS alkalmazás (Cond. Acc. System)
  - CI: Common Interface
    - A vevő/képalkotó és a modul közötti interfész

Hozzáférés korlátozás

- Jelzésrendszer



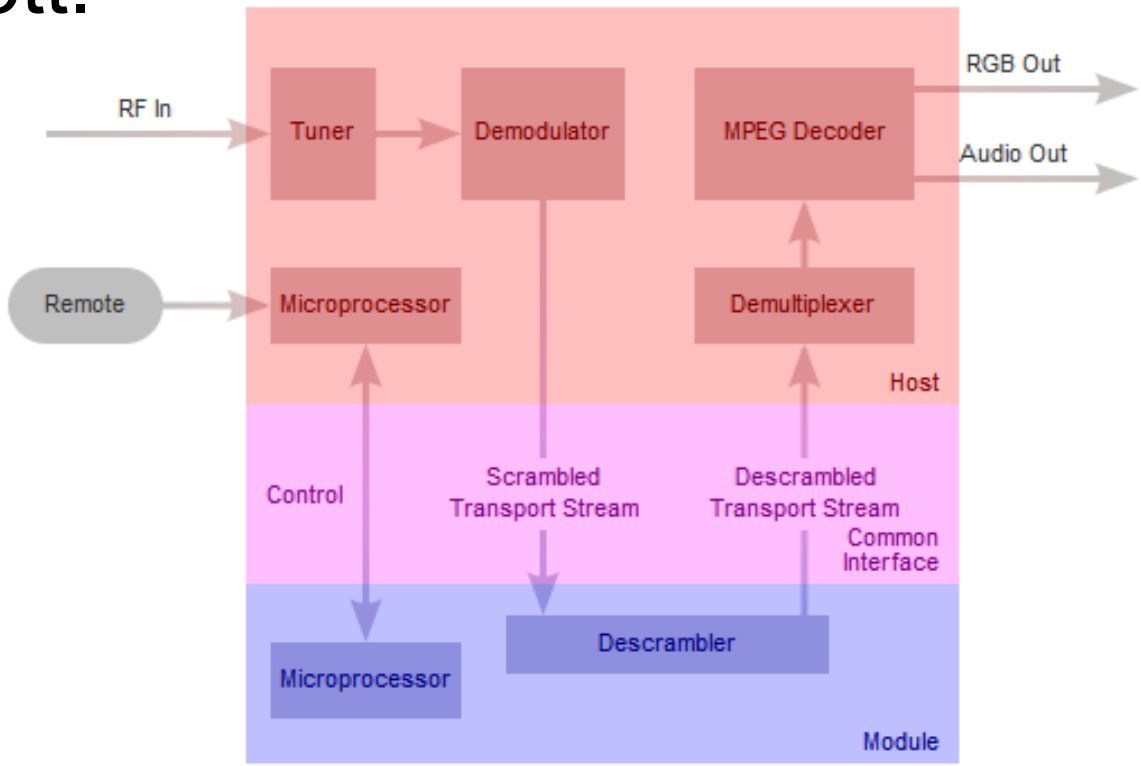
ECM és EMM

- Entitlement Management Message (EMM)
  - Kódkártyát engedélyező üzenet
    - memória feltöltés/frissítés
  - Aktuális jogosultsági adatok, kulcsok
  - Titkosított üzenet

- Entitlement Control Message (ECM)
  - Control Word (kulcs) átvitele (10-30 másodpercenként csere, maximum 120 másodperc). A kulcsok hossza 64 bit.
  - Jogosultsági információk
  - Titkosított üzenet

Common Interface

- Szabványos interfész a kártya és a TV között.



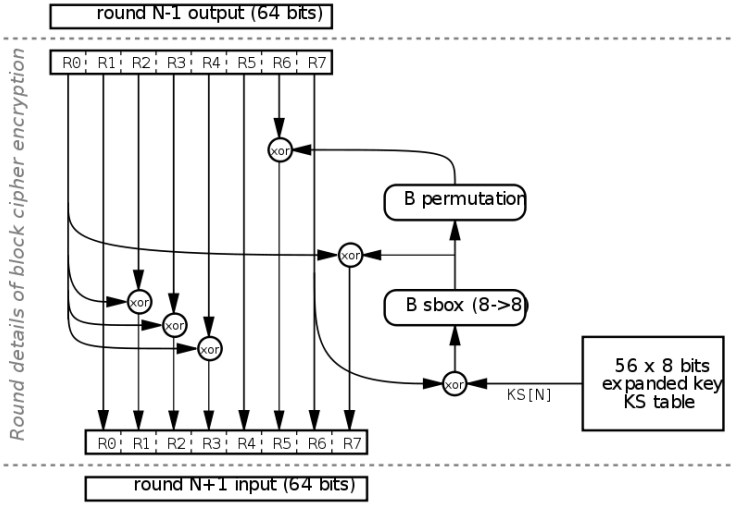
Kódok előállítása

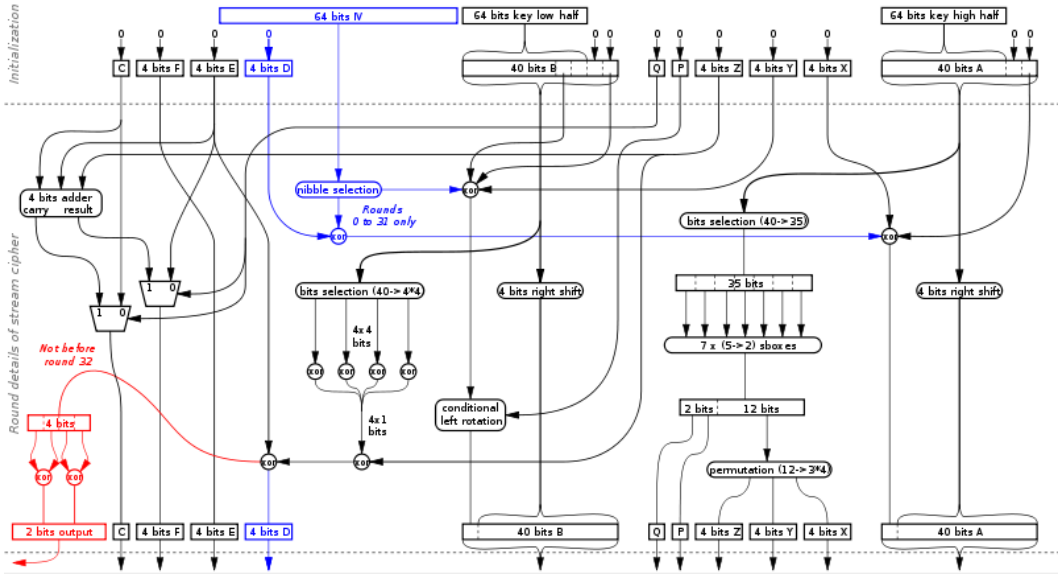
- Európában alkalmazott rendszerek:
  - Conax (Telenor)
    - MinDigTV Extra
  - CryptoWorks (Philips)
    - UPC Direct
  - Nagravision (Nagra-Kudelski)
    - Digi TV
  - Viaccess (France Telecom), Videoguard (NDS), Irdeto (Irdeto), Mediaguard (SECA), PowerVu (Scientific Atlanta), VideoCrypt (News Datacom)

DVB titkosítás

- Common Scrambling Algorithm (CSA) - 1994
- 2002 óta nyilvánosan is ismert (libdvbcsa)
- Kulcsok segítségével a videófolyam titkosítása (MPEG2 TS)
  - Blokk titkosítás és folyam titkosítás együttes használata (kivéve az első és utolsó blokkokat)
  - CBC Blokk titkosítás hátulról, utolsó egész blokktól
  - Folyam titkosítás elölről, a 64. bit után
- 64 bites kulcsok (de ebből csak 48 bitet használ)

DVB CSA blokk és folyam titkosító





DVB CSA3

- A DVB CSA (1 és 2) egyelőre nem feltörhető, bár vannak hibák
  - A 48 bites kulcsokat FPGA segítségével lehet brute force támadni
  - Ismert nyílt szöveg esetén szivárvány táblák segítségével gyorsan törhető
- 2008 DVB CSA3 megjelenése
  - Egyelőre nincs széles körben elterjedve. A cél, hogy mire kifut a CSA2, már meglegyen az új alternatíva
  - 128 bites kulcs, folyam titkosítás
  - AES128 titkosító + "eXtended emulation Resistant Cipher (XRC)", az utóbbi nem publikus
  - Kulcs előállítása az IDEA titkosító alapján + nem publikus cserék
  - Nem a jelenlegi problémákra ad megoldást

Card sharing

- A hozzáférés ellenörző által előállított kódok megoszthatóak az Interneten!
  - 64 bit információ, rövid frissítéssel
  - A felhasználók megoszthatják egymás között az előfizetéseiket. (Akár 1 felhasználó is a saját előfizetését több készülékkel)
  - Illegális előfizetések kód szolgáltatásra

- Védekezés
  - Irdeto és NDS belenyúl a lejátszó szoftverébe is. Így a CW titkosított és csak legitim lejátszó fedheti fel.
  - A CW cserék száma megnövelhető, ezáltal nehezebbé tehető a megosztás.
  - Jogi lépések a megosztók ellen.