KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Vytautas Simanaitis

Vaizdo duomenų srauto apsaugos metodas, pagrįstas MPEG-2 transporto srauto daliniu šifravimu.

Magistro darbas

Darbo vadovas
doc. A. Liutkevičius

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Vytautas Simanaitis

Vaizdo duomenų srauto apsaugos metodas, pagrįstas MPEG-2 transporto srauto daliniu šifravimu.

Magistro darbas

Recenzentas

2012-05-

Darbo vadovas

doc. A. Liutkevičius

2012-05-

Atliko

IFN-0/3 gr. stud.
Vytautas Simanaitis

2012-05-23

Kaunas, 2012

Turinys

SUMMARY

This study analyzes security techniques used in video and video streaming intelectual property protection. Review of H.264 video coding standard used for video encoding and widely used RTP and MPEG-2 transport stream protocols. Analyze of similar works in problematic area, with different approach for securing video content. Protection could be implemented in two different methods, while encoding video, or encrypting video stream.

The proposed partial video stream encryption method is suitable for limited resource devices such as STB, mobile devices. Security method provides necessary level of protection, secures video stream faster then full stream encryption.

Key words: MPEG-2 transport stream, chipering, security, video streaming, STB, low resource.

1 ĮVADAS

Skaitmeniniame amžiuje vis didėjanti intelektualinės nuosavybės problema kelia didelį susirūpinimą. Išpopuliarėjus skaitmeninei televizijai vis dažniau pasitaiko pažeidimų, dėl kurių nukenčia kūrėjai, kad to išvengtume yra naudojama DRM(skaitmeninių teisių valdymas) struktūra, t.y. skaitmeninių teisių valdymas, leidžiantis tiekti vaizdo paslaugas tik autorizuotiems vartotojams. Internetinėse televizijos transliacijose naudojamos įvairios topologijos, kuriose skiriasi DRM sprendimai, kyla skirtingos suderinamumo problemos norint tinkamai apsaugoti turinį, kurti paslaugas orientuotas į klientą ir jo norų patenkinimą. Vaizdo transliacijose vis dažniau naudojamas MPEG-4 H.264/AVC formatas, kuris turi daug pranašumų lyginant su senesniu, bet vis dar naudojamu, MPEG H.261 formatu. MPEG-4 formatu koduotas vaizdas reikalauja mažesnio interneto srauto, geresnė atkuriamo vaizdo kokybė, atsparesnis klaidoms pasitaikančioms transliuojant vaizdą. Formatas apibrėžia vaizdo kodavimą įvairaus formato vaizdams, galimybę koduoti vaizdą atsparesnė pasitaikančioms klaidoms, ar be nuostolių, t.y. Vaizdas suspaudžiamas jo nepakeičiant. H.264/AVC vaizdo dekodavimui/užkodavimui sukurti aparatiniai spartintuvai gebantys vaizdą dekoduoti naudojant vos kelis vatus energijos. Tokie, spartintuvai plačiai naudojami mobiliuose įrenginiuose, IP televizijos priedėliuose. DRM struktūroje duomenys yra šifruojami, naudojantis saugiais ir patikrintais kriptografijos metodais. Kriptografijoje naudojamas AES šifravimo algoritmas yra greitas ir patikimas, nors daugelis dar neatsisako ir pasenusio DES algoritmo, ar vaizdo šifravime abejotinai saugaus XOR. Ne visi šifravimo metodai yra vienodai patikimi, kai kurie užtikrina aukštą saugumo lygį, tačiau reikalauja didelių skaičiavimo resursų, tai nėra didelė problema jei klientas vaizdą žiūri per kompiuterį ar panašų įrenginį turintį didelius skaičiavimo resursus, sudėtingiausia yra televizijos priedėliuose, nes jie neturi daug resursų, o vaizdo dekodavimui naudojami spartintuvai, dėl to ne visi šifravimo metodai su jais yra suderinami dėl dešifravimo žingsnių, kuriuos reikia atlikti vaizdo dekodavimo etapuose.

1.1 Tikslas ir uždaviniai

Atsižvelgiant į sukurtas technologijas ir standartus, pasiūlyti tiesioginės transliacijos H.264/AVC formato vaizdo turinio šifravimo metodą, kuris nereikalautų didelių skaičiavimo resursų, ir būtų suderinamas su kliento sistemos vaizdo dekodavimo spartinimo sistemomis. Suteiktų pakankamą saugumo lygį, taip pat būtų nesunkiai suderinamas su skaitmeninių teisių valdymo struktūra.

- Išanalizuoti naudojamus standartus transliacijose, suprasti ir išsiaiškinti H.264/AVC vaizdo šifravimo algoritmą, atsparumą atakoms, sugadinto vaizdo atstatymo galimybes.
- Išanalizuoti ir palyginti egzistuojančius vaizdo transliacijų ir H.264/AVC vaizdo šifravimui skirtus apsaugos metodus.
- Sukurti tiesioginės transliacijos vaizdo turinio apsaugos metodą, kuris užtikrindamas reikalingą apsaugos lygį, tuo pat metu užtikrintų didesnę užšifravimo ir iššifravimo greitaveiką ir reikalautų mažiau skaičiavimo resursų, nei šiuo metu visuotinai naudojamas pilnas vaizdo turinio šifravimas.
- Siūlomo metodo pagrindu, realizuoti vaizdo turinio užšifravimo ir iššifravimo paprogrames.
- Eksperimentiškai įvertinti siūlomo metodo efektyvumą, įvertinant jo realizacijos greitaveiką bei duomenų pralaidumą, ir palyginti su standartiniu pilnu vaizdo turinio šifravimu.

2 VAIZDO TRANSLIACIJOS APSAUGOS METODŲ IR PROTOKOLŲ ANALIZĖ

Siekiant geriau suprasti grėsmes, problemas kylančias norint apsaugoti internetinę televiziją ir vaizdo transliacijas, reikia išsiaiškinti transliavimo principus ir esamus metodus vaizdo pristatymui, naudojamus protokolus. Vaizdo transliacijose pagal pristatymo principą skirstomos į: transliacijos vienam, transliacijos keletui ir retransliacijas.

Mokamo turinio transliacijoms apsaugoti naudojai šifravimo algoritmai transliacijai šifruoti arba vaizdo suspaudimo formatai numatantys kodavimo metu atlikti vaizdo įslaptinimą naudojant paslaptį(pvz. šifro raktą) be kurios vaizdas nebus dekoduojamas teisingai. Transliacijos apsaugai naudojama paslaptis turi būti saugiai perduota tik tiems klientams kurie turi teisę vaizdą dekoduoti, tuo rūpinasi licencijų serveris.

2.1 Vaizdo transliacijos ir tiesioginės transliacijos

Vaizdo transliacijos gali būti suskirstomos į dvi pagrindines grupes:

- Tiesioginės realaus laiko vaizdo transliacijos
- Paruošto turinio transliavimas klientui.

Pagrindinis skirtumas tarp tiesioginės realaus laiko transliacijos ir paruošto turinio transliavimo yra tai, kad transliuojant paruoštą turinį klientas gali buferizuoti duomenis priedėlyje, pvz rodyti 15min filmo ir turėti parsiųstą dar 5min į priekį. Buferizuojant duomenis išvengiama nevienodo srauto pralaidumo, transliacijos serverio apkrovos problemų. Realaus laiko tiesioginėse transliacijose nėra ką buferizuoti, nes transliacija vykdoma gyvai, iškart perduodant duomenis iš kodavimo įrenginio, siekiant išvengti nedidelių tinklo sutrikimų įvedamas kelių sekundžių vėlinimas, pvz 5s vėlinimas kuris yra buferizuojamas.

Vaizdo kodavimo algoritmai yra kuriami taip, kad vaizdo dekodavimo laikas būtų kuo trumpesnis ir paprastesnis, dėl to vaizdo užkodavimas tampa sudėtingas ir imlus resursams procesas.

Geriausiai žinomas vaizdo transliacijos klientui pavyzdys yra internetinė televizija(toliau IPTV). Transliuojamas turinys turi būti paruoštas taip, kad klientas galėtų jį peržiūrėti su turima įranga, IPTV klientai dažniausiai turi IPTV priedėlius, kurie yra prijungti prie televizoriaus ir transliacijos kanalo, pvz interneto. IPTV priedėlis priima tiekėjo siunčiamą srautą, žinodamas kokiu formatu vaizdas yra transliuojamas, jį dekoduoja ir perduoda į televizorių. Vaizdo transliavimui Lietuvoje naudojant IPTV dabar plačiai naudojamas H.263 formatas, tačiau greitai jis bus pakeistas, geresnę vaizdo kokybę užtikrinančiu H.264/AVC formatu.

2.2 Vaizdo transliacijos pagal srauto pristatymo principą

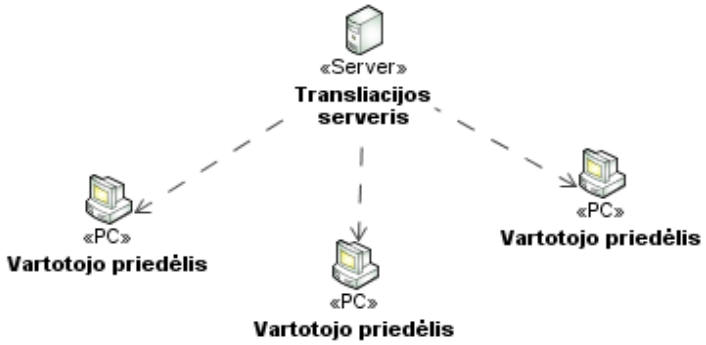
Pagal transliacijos principą vaizdo transliacijas galima suskirstyti į 3 pagrindines sritis:

- Transliacija vienam klientui
- Transliacija keletui klientų
- Transliacija visiems

Naujausia transliacijos sritis: p2p transliacija, tokio tipo transliacijoje klientai retransliuoja srautą kitiems, taip sukurdami tinklą. Naudojant p2p transliacijas sudėtinga skirstyti klientus į grupes siekiant valdyti jų žiūrimą turinį.

2.2.1 Transliacija vienam klientui

Televizijos transliacijos vienam klientui nėra naudojamos dažnai, naudojama kai reikia vaizdą koduoti kitu formatu, arba jei tinklas nepalaiko transliacijos keletui galimybės.



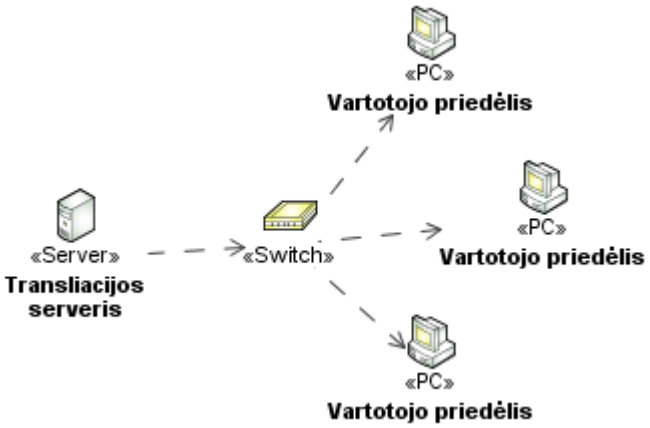
1 pav. Transliacijos vienam klientui metodo principinė schema

Kiekvienas klientas iš serverio reikalauja transliacijos resursų. Klientas užmezga ryšį su serveriu ir serveris transliacijai išskiria resursus(1 pav.). Jei vaizdas jau yra paruoštas transliacijai t.y., užkoduotas, serverio išskirti resursai kiekvienam klientui yra identiški.

Šio principo privalumas – kiekvienas klientas yra betarpiškai kontroliuojamas serverio. Pakitus tinklo charakteristikoms transliacija gali būti pakeista, vaizdas perkoduotas neįtakojant kitų klientų transliacijų.

2.2.2 Transliacija keletui klientų

Transliacijos keletui klientų yra naudojamos plačiausiai. Transliuojant tokiu metodu serveris yra apkraunamas daug mažiau nei transliuojant kiekvienam klientui atskirai. Transliacijos metu vaizdo srautas yra siunčiamas keletui klientų visiškai toks pat, nauja serverio jungtis sukuriama tik tada jei klientas nepasiekiamas esamu keliu. Transliuojant vaizdą 300kbit/s greičiu 1000 klientų. Transliacijos vienam klientui metodu reikėtų 300Mbit/s spartos tinklo ir serverio gebančio apdoroti 1000 lygiagrečių sesijų. Transliacijos keletui klientų metodu pakanka 300kbit/s, nes visi klientai priima identišką srauto kopiją.



2 pav. Transliacijos keletui klientų metodo modelis

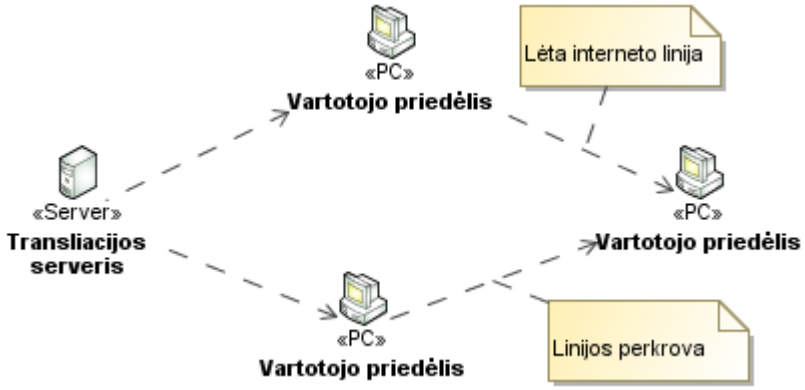
Norint naudotis transliacijos keliems metodą interneto tinklas turi palaikyti multicast režimą(2 pav.), t.y turi palaikyti D klasės adresų zoną. Transliacijai gali būti naudojami RTP, RTCP, SRTP SRTCP, MMS, MPEG-2 ir panašūs protokolai.

2.2.3 Transliacija visiems

Transliacijos visiems metodas plačiai paplitęs, naudojama įvairi fizinė terpė: radijo bangos, internetiniai tinklai(plačiajuosčiai, šviesolaidžiai, ISDN ir pan.). Šio transliacijos principas – transliuoti visus kanalus visiems, klientai norimus kanalus išfiltruoja ir atkuria. Siekiant atskirti nemokamą ir mokamą transliacijas, nėra kito būdo kaip tik naudoti srauto įslaptinimą koduojant ar šifruojant.

2.2.4 Peer-to-Peer transliacija

Peer-to-Peer transliacija tai metodas, kai klientas yra naudojamas vaizdo srauto retransliavimui. Serveris transliuoja vaizdo srautą klientams turintiems spartų interneto ryšį, kurie gali priimtą vaizdą transliuoti kitiems klientams. Šiuo metodu galima dar labiau sumažinti reikalingus serverio pajėgumus vaizdui transliuoti. Tačiau naudojantis šiuo metodu iškyla daug sunkumų, nes ne kiekvienas klientas gali retransliuoti vaizdo srautą(3 pav.). Klientui pakeitus žiūrimą kanalą, išjungus kompiuterį ar IPTV priedėlį, dingus elektrai ar pan., gali kilti televizijos trukdžių kitiems klientams. Klientas retransliuojantis vaizdo srautą turi būti patikimas, kad neiškraipytų nepakeistų ar kitaip nesugadintų srauto, taip pat nebūtų įterpiamas papildomas kodas galintis sutrikdyti kitų klientų darbą.



3 pav. Peer-to-Peer transliacija

2.2.1 ir 2.2.2 aptarti metodai gali tarpusavyje maišytis sudarydami mišrius transliavimo metodus, tokiu atveju sumažinamas serverio resursų poreikis transliuojant televizija daugeliui klientų. Vaizdo transliacijos reikalauja didelių transliuojančio serverio pajėgumų, bei spartaus interneto ryšio, pvz.: valanda filmo užkoduoto 300kbit/s, 320x240 taškų reikalauja apie 128MiB serverio vietos diske. Sakykim filmas yra transliuojamas 1000 klientų naudojant transliacijos vienam metodą, reikalaus 300Mbit/s interneto greičio. Naudojant neefektyvius vaizdo apsaugos metodus, gali tekti tokiu pat greičiu ir perkoduoti vaizdą. Siekiant sumažinti resursų kiekį kurio reikia serveriui yra naudojami transliacijos keletui klientų ar retransliacijos metodai.

2.3 Skaitmeninių teisių valdymas tiesioginėse vaizdo transliacijose

Skaitmeninio teisių valdymas(Digital Rights Managemen – DRM). 2.2 skyriuje aprašyti modeliai naudojami tiesioginėms televizijos transliacijoms perduoti. Kiekvienas modelis turi savų privalumų ir trūkumų. Vaizdo srautas pasiekia klientus, jie mėgaujasi teikiamomis paslaugomis, tačiau ne visas vaizdo turinys yra nemokamai platinamas. Dažniausiai už filmus ir kitą vaizdo medžiagą reikia mokėti filmų kūrėjams, autoriams, įrašų kompanijoms. Norint užtikrinti vaizdo srauto saugumą ir privatumą, kad jį galėtų peržiūrėti tik už tai sumokėję klientai, yra pasitelkiama skaitmeninių teisių valdymas. DRM įgyvendintas naudojant kriptografiją ir šifravimo algoritmus, kuriais vaizdo srautas yra šifruojamas. Tačiau, be šifro rakto vaizdo negalėtų peržiūrėti net ir sumokėję mokestį už turinį klientai, DRM struktūra išsprendžia saugaus raktų apsikeitimo problemas, klientų autentikaciją ir autorizavimą.

2.3.1 Licencijų serveris

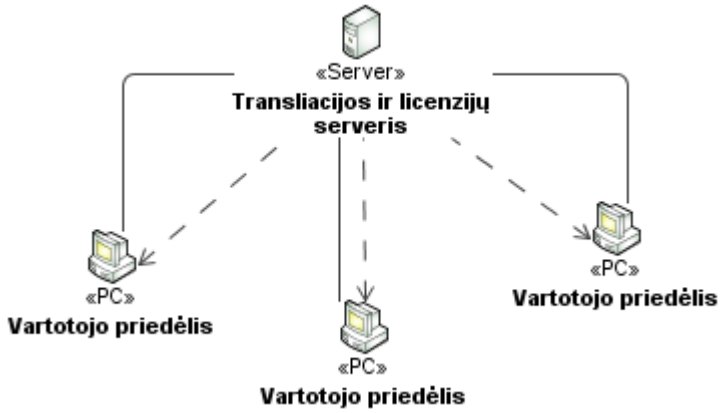
Licencijų serveris yra kompiuteris, kuris saugo vartotojų licencijas ir jiems suteiktus leidimus bei teises į licencijuotą turinį. Pagrindinis licenzijų serverio funkcija yra patvirtinti ir teikti reikalingus kodus ar informaciją klientams ar sistemoms, turinčios galimybę apdoroti licencijuotą

turinį. Licencijų serveris naudoja licencijavimo taisykles, apspręsti vartotojams ar jų įrenginiams, turintiems priėjimą prie turinio. Licencijų taisyklės yra teisės arba ribojimai kurių laikomasi suteikiant priėjimą prie turinio[23].

Licencijų serveris gali būti fiziškai viename kompiuteryje su transliacijos serveriu, jeigu transliuojama nedideliam kiekiui klientų, taip pat serveryje yra apsprendžiamas protokolas saugiam licencijos perdavimui užtikrinti. Licencijos į turinį gali būti teikiamos papildomu saugiu kanalu, arba perduodamas kartu su turiniu. Licencijos suteikimas, šifro raktas, šifravimo algoritmas ar panaši informacija turi būti perduodama saugiu kanalu arba šifruota, nes perėmus šifro raktą turinys gali būti atskleistas trečiųjų asmenų.

2.3.2 Šifruota transliacija vienam klientui

2.1.1 skyriuje aprašytas transliacijos vienam principą, jį papildžius skaitmeninių teisių valdymo infrastruktūra pavaizduotą 4 pav. diagramoje. Kiekvienam klientui transliacija yra užšifruojama tik jam būdingu metodu ir tik tas klientas gali transliaciją iššifruoti. 4 pav. ištisinėmis linijomis pažymėtas saugus duomenų, reikalingų turiniui iššifruoti, apsikeitimas. Tokio modelio pranašumas yra tas, kad ne identifikuotiems klientams turinys gali būti net neteikiamas, nes kiekvienas klientas palaiko susijungimą su serveriu. Net ir perimtas turinys bus šifruotas, ir neturint duomenų kaip jį iššifruoti, jo peržiūrėti nepavyks. Paslaugos tiekimas klientui gali būti bet kada nutrauktas, jam nebesiunčiant licencijos informacijos. Tokioje topologijoje klientai gali būti suskirstyti į grupes, kuriuos sieja bendra licenzija, taip sumažinant serverių apkrovą, nes ne kiekvienam klientui turinį reikia užšifruoti.

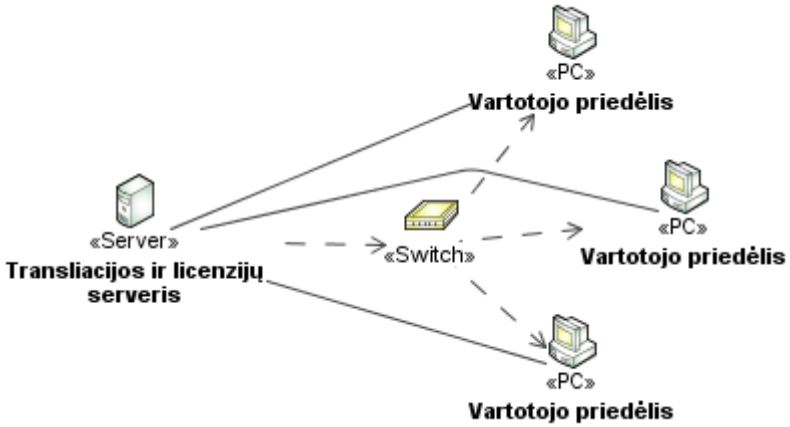


4 pav. Šifruota transliacija vienam klientui

2.3.3 Šifruota transliacija keletui klientų

Transliacijos keletui klientų metu, duomenys yra šifruojama visiems klientams žinomu algoritmu ir licencinė informacija reikalinga turiniui iššifruoti. Šis metodas yra plačiai naudojamas, taip pat jo modifikacija, kai klientai skirstomi į pogrupius su bendromis licencinėmis taisyklėmis.

Užkoduotą transliacijos srautą gali priimti visi, tačiau iššifruoti tik licencijų serverio(2.2.1 pastraipa) identifikuoti klientai, gavę reikalingus raktus ir algoritmus saugiu(**Error! Reference source not found.** pav. ištisinė linija) kanalu.



5 pav. Transliacijos keletui klientų metodas

Šis metodas plačiai paplitęs dėl santykinai pigių transliacijos kaštų, pasinaudojant skirtingais kriptografijos algoritmais pasiekiamas didelis saugumo ir patikimumo lygis. Transliuojant šiuo metodu dažniausiai naudojamas SRTP ir SRTCP arba MPEG-2 transliacijos protokolas. Analogiški saugumo mechanizmai taikomi transliacijos visiems metodu.

2.3.4 Peer-to-Peer transliacija

Peer-to-Peer transliacijos metodas yra sudėtinga užtikrinti saugią transliaciją, [24] straipsnyje yra pasiūlytas DRM metodas. Naudojant p2p tinklus, licencijos serveris turi būti paskirstytas, paskirsčius licencijavimą yra sunkiau užtikrinti saugumą. [24] modelyje yra naudojamas serveris MARS(Multimedia Application Routing Server), šiame serveryje atliekamas klientų identifikavimas, transliacijos kelių parinkimas. MARS serveris taip pat užtikrina transliacijos vientisumą ir integralumą, priimamas srautas tik iš identifikuotų klientų. MARS serveris yra pirmame IPTV lygyje, antrame lygyje naudojamas p2p tinklas ir suskirstytų jame klientų. Saugumui užtikrinti transliacijos tarp klientų turi būti patvirtintos licencinių serverių.

2.4 Tiesioginėse transliacijose naudojami protokolai

2.3 skyriuje trumpai aprašyti pagrindiniai principai tiesioginėms televizijos vaizdo transliacijoms perduoti, ir DRM pritaikymą šioms struktūroms. Ne visi tiesioginių transliacijų protokolai gali būti taikomi bet kuriam transliavimo metodui. Kaip ir transliavimo principai protokolai yra skirti transliuoti vienam ir keletui klientų su numatytu šifravimu arba be.

Transliacijai naudojamas RTP protokolas, kartu su RTCP protokolu kuris kontroliuoja RTP srautą[25]. RTP protokole pridėjus duomenų šifravimą buvo sukurtas SRTP ir SRTCP protokolai[22], naudojant šį protokolą, video transliacija yra pilnai šifruojama(žr. 2.4.1).

RTSP, realaus laiko transliacijos protokolas[21] aprašo sąlygas kurios turi būti tenkinamos tiesioginei transliacijai, paketų praradimą, per mažą srauto pralaidumą. RTSP teikia funkcijas vartotojui, tokias kaip, groti, pristabdyti, sustoti ir pan. (žr. 2.4.3)

MPEG2 standartai aprašo transliacijos srauto perdavimo protokolus, naudojamus tiesioginėms ir realaus laiko tiesioginėms transliacijoms. Vaizdo ir garso užkoduoto MPEG kodavimo algoritmais perdavimas paketų komutavimo tinklais(žr. 2.4.5)

Transliacijoms perduoti dažniausiai naudojamas UDP protokolas. TCP protokolas praktiškai nenaudojamas, nes nesėkmingo siuntimo atveju, tcp paketai yra pakartotinai persiunčiami, o tai sustabdytų vaizdo atkūrimą kliento pusėje, kol blogi paketai bus priimti teisingai.

2.4.1 RTP protokolas

RTP protokolas pagal OSI modelį yra 7 – aplikacijų lygmenyje. Transliacija perduodama naudojant UDP protokolą, jei UDP nepalaikomas, naudojamas TCP protokolas. RTP protokolas apibrėžia vaizdo, garso ir papildomų duomenų transliacijas vienam ar keletui klientų naudojant vieną srautą. RTP protokolas neužtikrina paketų pristatymo laiku, jų kokybės, ar paketų pristatymo tvarkos, tai turi užtikrina žemesni sluoksniai. [25]

RTP protokolo pakete nurodoma laiko žymė kuri naudojama paketams sinchronizuoti, paketo unikalus numeris, RTP versija. RTP sesijoms atskirti naudojama unikali SSRC reikšmė. Naudojant RTCP protokolą gauta statistinė informacija atskiriama pagal SSRC žymę.

1 lentelė. RTP protokolo paketo struktūra[26]

00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
V		P	X	CC				M	PT							Sequence Number															
Timestamp																															
SSRC																															
CSRC [0..15] :::																															

- V – RTP protokolo versija
- P – Kamšalo indikatorius
- X – Antraštės praplėtimo vėliavėlė
- CSRC count – CSRC indikatorius.
- M – Žymeklis, naudojamas pagal profilį.

- Payload type – turinio tipas, vaizdas, garsas ir t.t.
- Sequence number – sekos numeris, padidinamas vienetu kiekvienam naujam paketui.
- Timestamp – laiko žymė, naudojama paketų sinchronizacijai.
- SSRC – unikalus sesijos žymeklis.
- CSRC – naudojamas atskirti sulietiems srautams.

2.4.2 RTCP protokolas

RTCP protokolas neperduoda vaizdo ar garso srauto. Protokolas skirtas statistinės ir kontrolės informacijos perdavimui apie RTP srauto sesiją. Pagrindinė RTCP protokolo funkcija užtikrinti servisų kokybės kontrolę(QoS).

Protokolas grąžina tokius parametrus kaip: persiųstas, klaidingų, pamestų paketų kiekius, tinklo perkrovimus.

Kiekviena sesija turi unikalią SSRC reikšmę, pagal kurią yra identifikuojama, serveris gavęs informaciją apie nesklandžia transliaciją gali pristabdyti siuntimą, ar mažinti transliuojamo vaizdo kokybę klientui.

2.4.3 SRTP protokolas

SRTP - Secure Real-Time Protocol, tai RTP protokolo plėtinys, užtikrinantis šifravimą, autentikaciją ir integralumą. Šifravimas ir autentikacija yra nebūtina, norit transliuoti SRTP protokolu. Šifravimui ir iššifravimui naudojamas AES metodas ir vienas iš dviejų algoritmų

- Išskaidytas skaitliuko algoritmas (Segmented Integer Counter Mode) – Šis algoritmas leidžia iššifruoti bet kuria bloko vietą, nepriklausomai nuo o ar prieš tai buvęs blokas buvo iššifruotas teisingai, tai labai svarbu nes siunčiant duomenis UDP tinklu jie gali būti pakitę.
- f8 algoritmas – retai naudojamas, nes *i+1* blokas šifruojamas naudojant šifruotą *i* bloką.

SRTP standarte naudojamas 128 bitų ilgio slaptas raktas ir 112 atsitiktinis raktas.

Integralumas užtikrinamas naudojant HMAC-SHA1 maišos funkciją, ji sugeneruoja 160 bitų ilgio seką. Authentication tag yra SHA-1 maišos rezultatas sutrumpintas iki 80 arba 32 bitų. Seka skaičiuojama iš perduodamų duomenų ir dalies antraštės įskaitant ir paketo sekos numerį.

SRTP protokole nenumatytas raktų apsikeitimo mechanizmas. Tam naudojami kiti algoritmai ir metodai, pvz. ZRTP, SDES, MIKEY.

2 lentelė. SRTP protokolo struktūra[27]

00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
RTP extension :::																															
Payload :::																					Pad								Pad count		

MKI
Authentication tag

- RTP extensions – RTP paketo antraštė
- Payload – RTP paketo duomenys
- Pad – kamšalas užpildyti likusią vietą, lyginiam baitų kiekiui
- Pad count – kamšalo ilgis
- MKI – Master Key Identifier, naudojamo rakto identifikatorius, pagal kurį turės būti dešifruotas paketas, tai ne pats raktas!
- Authentication tag – tai RTP paketo SHA1 suma, suma turi būti skaičiuojam prieš užšifruojant paketą.

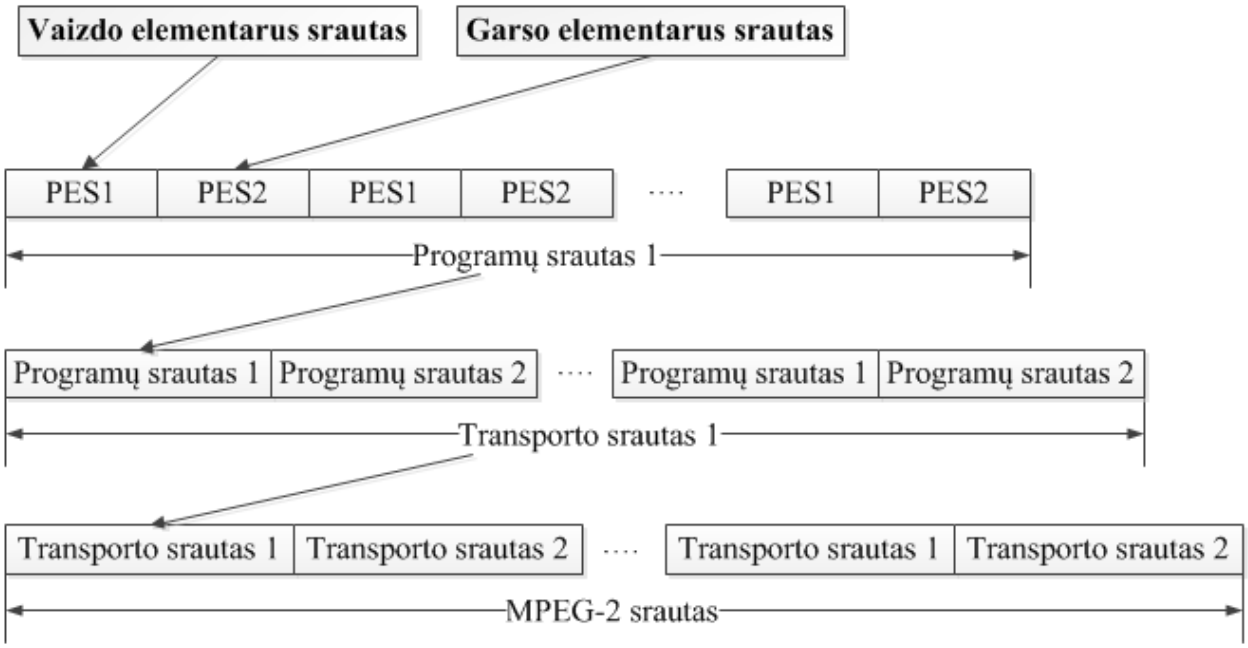
2.4.4 SRTCP protokolas

SRTCP protokolas buvo sukurtas RTCP protokolo pagrindu. SRTCP protokolas turi visą RTCP protokolo funkcionalumą, papildytą perduodamos informacijos šifravimo, autentikacija ir integralumo tikrinimu. Integralumas užtikrinamas naudojant HMAC-SHA1 maišos funkciją. Kaip ir SRTP atveju naudojamas AES šifravimo metodas.

2.4.5 MPEG-2 transliacijos protokolas

2.4.5.1 MPEG-2 Programų srautas

MPEG-2 transliacijos formatas aprašo, kaip reikia sudėti vaizdo, garso, ir kitus duomenis į vieną transliacijos transporto srautą. Transporto srautas - sudarytas iš keleto programų srauto, programos srautas sudarytas iš elementarių srautų, tokiu kaip vaizdo garso ar duomenų informacijos(6pav.). Perduodant transliacijas komunikavimo kanalais, srautas suskaidomas į 188, 204 ar 210 baitų ilgio blokelius, paketai perduodami ryšio kanalu. Ryšio kanalas gali būti ATM, paketų komutavimo, radijo bangomų. MPEG-2 sraute paketo dydis priklauso, nuo ryšio kanalo paketo duomenų perduodamo ilgio.



6 pav. MPEG-2 transliacijos formato diagrama.

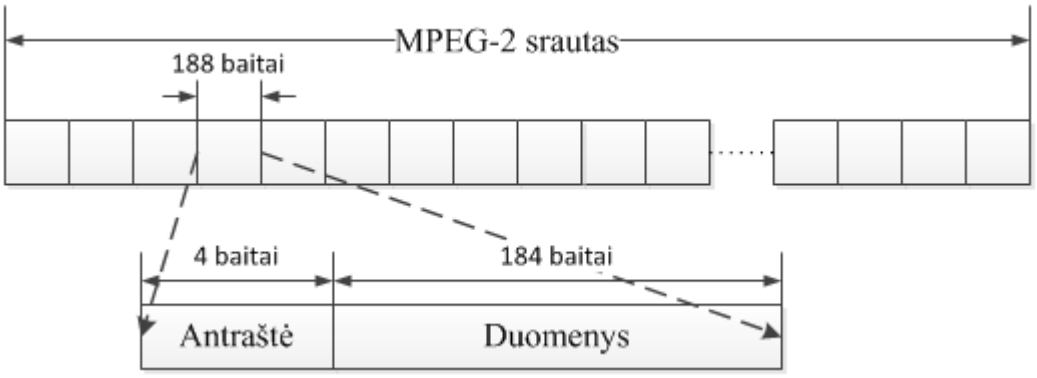
Elementarieji srautai sudėti į programų srautą, turi bendrai nusakomą aplinką, bendra laiko žymė. Naudojant programų srautą dekoduojančiam įrenginiui yra lengviau susieti garso ir vaizdo sinchronizaciją. Praktikoje programų srautas perduoda tik vieną kanalą, vaizdas garsas ir tarnybinė informacija.

Programų srautas sudaromas iš suskaidytų į paketus elementariųjų srautų. PES(Packetised Elementary Stream – PES) naudojamas vaizdo ir garso sinchronizacijai, turinio apsaugai, kodavimo parametrams perduoti. PES gali būti fiksuoto arba kintamo ilgio iki 65536 baitų bloke, turi 6 baitų antraštę. PES turi sveiką skaičių elementarių srautų. PES antraštė prasideda 3 baitų ilgio pradžios kodu(packet start code prefix), toliau 1 baito ilgio srauto identifikatorius(stream id), MPEG-2 standarte numatyti tokios id reikšmės: 110xxxxx – garso srautas, 1110xxxx – vaizdo srautas, 11110010 – kontrolinis paketas. MPEG-2 transporto srautas

MPEG-2 transliacijos srautas perduodant suskaidomas į 188B dydžio paketus. 188B dydžio paketai pasirinkti dėl suderinamumo su ATM tinklais. Labai mažo patikimumo perdavimo kanaluose naudojami 204B ar 208B ilgio paketai, papildomi baitai skirti klaidų taisymui. Transliacijos srautu galima perduoti kelis programų srautus su nepriklausoma laiko žyme. 7 pav. pateikta MPEG-2 transporto paketo struktūra, kiekvienas paketas susidedantis iš 4 baitų antraštės ir 184 baitų duomenų. Antraštėje yra :

- Sync – 0x47 neunikalus baitas žymintis transporto paketo pradžią
- TPR – transporto prioriteto bitas
- PUSI – duomenų pradžios žymeklis, nurodo kad esamas paketas yra PES paketo pradžia

- EI – klaidos indikatorius
- PID – paketo identifikatorius PES paketo ID
- SCR – paketo kodavimas, 10 – lyginis raktas, 11 – nelyginis raktas
- AF – ar yra papildomų vėliavėlių(jos užima bendra 188 baitų vietą).
- CC – paketų indikatorius, kiekvienam paketui kuris turi duomenų(payload) padidinamas vienetu.



7 pav. MPEG-2 transportavimo paketo struktūra.[34]

Dekoduojant vaizdą iš transporto srauto, pagal transporto paketo PID reikšmę dekoderis žino kuriam srautui, vaizdo ar garso, priklauso paketo turinys.

Transporto pakete, kai PID=0 yra perduodama programų lentelė(PAT – Program Association Table), kiekvienas įrašas perduoda informaciją apie elementariuosius srautus kurie sudaro programų srautą, ir jam priklausančių elementariųjų srautų PID reikšmes. PAT lentelėje taip pat gali būti perduodama informacija apie tinklo fizinius išteklius, spartą, pralaidumą, paketų pristatymo patikimumą. Aukštesnio lygio programos gavusios tokius duomenis gali mažinti vaizdo kokybę, siekiant užtikrinti nenutrūkstamą transliaciją.

2.5 H-264/AVC vaizdo kodavimo formatas

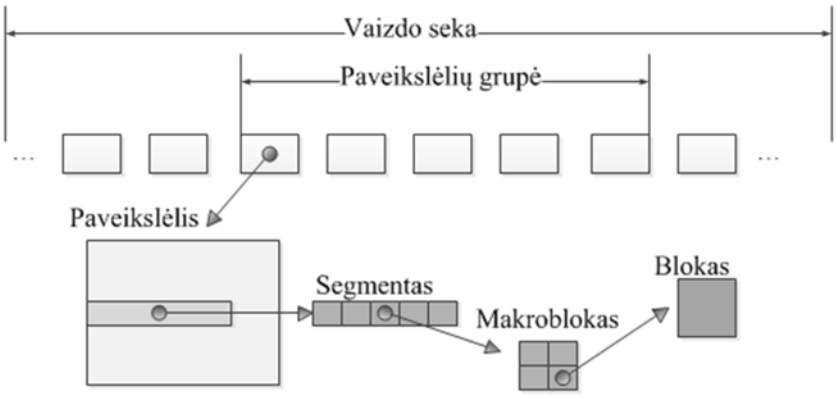
Norint perduoti geros kokybės vaizdą, nedidės spartos linijomis, jį būtina efektyviai suspausti. H.263 plačiai naudotas formatas, keičiamas efektyvesniu H.264 vaizdo kodavimo formatu. Norint perduoti 720x576 25 kadrai/s vaizdą, H.263 formatu, reikia apie 4Mb/s spartos linijos, tokį pat vaizdą koduojant H.264/AVC formatu, reikia tik 1.5-2 Mb/s spartos. H.264 formatas yra efektyvesnis dėl jame naudojamų suspaudimo algoritmų. Nekoduotas vaizdo įrašas yra paveikslėlių rinkinys, einantis vienas po kito.

Vaizdo seka skirstoma į paveikslėlių(kadrų) sekas(8 pav.). Vaizdo skaidymų į sekas pasiekiamas aukštesnis atsparumo klaidoms laipsnis, yra galimybė transliaciją pradėti nuo paveikslėlių sekos pradžios, nes jei visas srautas butų spaudžiamas nuo 0 laiko, klientas privalėtų turėti visus

kadrus, nuo kurių pradėtas vaizdo spaudimas, norėdamas dekoduoti esamus. Kiekviena paveikslėlių seka prasideda kadru kuris pilnai koduotas vien I tipo makro blokais.

2.5.1 Kadro skirstymas į sritis

Naudojant H.264 formatą, kiekvienas paveikslėlis yra suskirstomas į *sritis* kuriose bus atliekamas tolesnis suspaudimas. Skaidymas į sritis užtikrina geresnį atsparumą atsiradusioms



8 pav. Kadro skirstymas į sritis.

klaidoms. Jei klaida atsiranda srities I makro bloke, bus sugadintas tik toje srityje esančių P ar B makro bloku(žr. 2.5.2) atkūrimas, kurie rodo į sugadintąjį I makro bloką. (8 pav.).

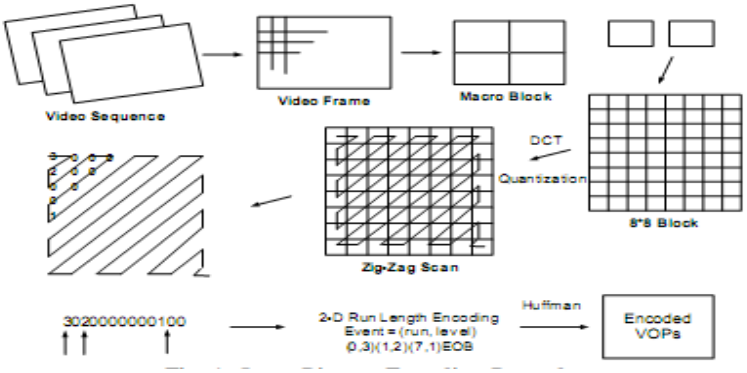
2.5.2 Segmentų suspaudimas

Segmentai dalinami į makro-blokus(MB). Pagal standartą makro blokų blokai gali būti 4x4, 4x8, 4x16, 8x8, 8x4, 8x16, 16x16, 16x4, 16x8 dydžio. H.264 nepraplėstame standarte yra trijų tipų makro-blokai:

- I – blokas užkoduotas naudojant 9 pav. pavaizduotus žingsnius. Bloko iškodavimui nereikalinga jokia informacija apie kitus makro blokus.
- P – makro-blokas yra tarsi nuoroda į pasikartojantį makro bloką, kuris yra I tipo. Pirma turi būti iškoduotas atraminis MB.
- B – yra nuoroda į du makro blokus, atkuriamas blokas bus suma atraminių makro blokų. Turi būti iškoduoti abu atraminiai MB.

Koduojant vaizdą naudojama YCbCr spalvų gama, Y – šviesumas, Cb – mėlynumas, Cr – raudonumas. Srityje išrenkami MB kurie yra panašūs į kitus tame pačiame kadre, arba kituose kurie priklauso paveikslėlių grupei, jie užkoduojami kaip P arba B tipo MB, likę koduojami kaip I tipo. Kiekvienam makro bloko blokui pritaikomas DCT kodavimas, atlikus šį žingsnį blokas yra matricos

pavidalo. Panaudojant Zig-Zag metodą(11 pav.), matrica paverčiama į skaitmeninę eilutę. Eilutė suspaudžiama panaudojant Run Length Encoding, ir pritaikomas Huffman metodas.



9 pav. H.264/AVC kodavimo žingsniai.[17]

Gauta seka yra užkoduotas kadras, seka gali būti perduodama ar saugoma panaudojant įvairius konteinerių ar perdavimo protokolus.

2.5.3 Vaizdo dekodavimas

Dekoduojant I tipo makro bloką srautas išspaudžiamas iki matricos kuri buvo gauta DCT kodavimo metu. Gauta matrica dekoduojama iki spalvų gamos. P ir B blokai dekoduojami tik tada, kai jau dekoduoti reikalingi atraminiai kadrai. Iškodavus visus makro blokus priklausančius sričiai gaunamas vaizdas. Sugadintų makro blokų aptikimas ir atstatymas plačiau aprašytas 2.5.4 skyriuje.

2.5.4 Atsparumas klaidoms

H.264 standarte numatytas nesudėtingų klaidų taisymas galintis atsirasti transliacijos metu. Didžiausią įtaką vaizdo sugadinimui turi sugadinto I makro bloko dekodavimas. Sugadinti makro blokai aptinkami lyginant dekoduotą sritį su šalia esančia, ieškant per didelio spalvinio perėjimo, kontūrų nevientisomo. Trys pagrindiniai metodai naudojami sugadintų MB ištaisymui:

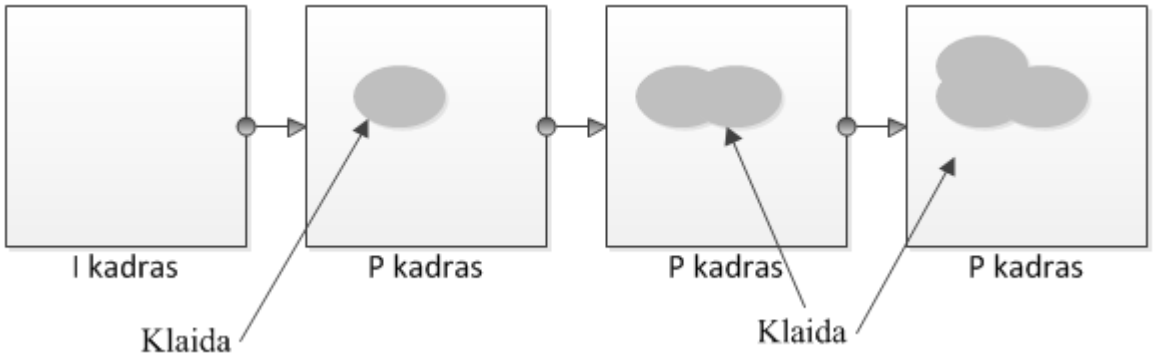
- Erdvinė interpoliacija – MB bandoma ištaisyti pagal aplinkinius MB juos suliejant, šiuo metodu atstatinėjant MB patogiau imti, kad jis ne nulinių koeficientų matrica, o vidurkis aplinkinių MB, taip pasiekiamas tikslesnis MB atstatymas.
- Kadro sustabdymas – MB nukopijuojamas iš prieš tai buvusiame kadre teisingai dekoduoto MB, šis metodas labai veiksmingas nedaug judesio turintiems filmams.
- Judesiu kompensuota interpoliacija – pirmų dviejų apjungtas metodas, tai atstatytu statinį MB vaizdą, toliau bandoma surasti judesio vektoriaus kryptį, pagal aplinkinių vektorių kryptis.

Pateikti metodai pakankamai gerai atkuria prarastą vieną ar kelis ne greta esančius MB, tačiau atsirandant klaidoms jų paprastai būna per daug, kad jas ištaisyti, nes į vieną prarasta paketą telpa keletas MB.

Daugiausia klaidų gali įsivelti bandant atkurti MB Huffman ir Run Length Encoding, nes bent vienas neteisingas bitas sukelia neteisingą bloko atstatymą.

Klaidos plinta dėl naudojamų P ir B tipo MB, jei toks MB yra sudaromas remiantis sugadintu makro bloku(10 pav.), jis pats tampa neteisingas, ir iš to seka kad visi kiti MB kurie remiasi klaidingais, bus klaidingi.

Siekiant išvengti tokių klaidų, yra naudojama keletas metodų, paprasčiausias būtų koduoti visus MB kaip I, t.y pilnus, tačiau šiuo atveju, vaizdo suspaudimo lygis yra labai mažas.



10 pav. Klaidos plitimas dekoduojant vaizdo kadrus.

2.6 Vaizdo medžiagos šifravimo būdai

Suspaustas vaizdo srautas yra neatsparus atsiradusioms klaidoms, todėl MPEG-2 transliavimo formatas numato nedidelių klaidų taisymą. Šifruoti duomenys taip pat gali būti traktuojami kaip transliacija su klaidomis, todėl dekoduojant duomenis jie bus neteisingi, ir vaizdas bus iškraipytas.

2.6.1 Pilnas transliacijos šifravimas

Vaizdo turiniui apsaugoti ir perduoti yra išanalizuota p2p principu veikianti sistema[1], tačiau ji aktuali tik tuo atveju jei vartotojai TV transliacijoms žiūrėti naudoja kompiuterį, taip pat bent dalis turi turėti spartų internetą turinio skleidimui. Kita vaizdo transliavimo DRM struktūra analizuota[2] šaltinyje pateikia gera raktų apsikeitimo mechanizmą, ir naujų raktų užtikrintą pristatymą vaizdo turiniui atkoduoti. Taip pat plačiai naudojamas ir standartizuotas RTP ir RTSP[21][6] skirtas nekoduoto realaus laiko vaizdo transliacijoms, bei RTP ir RTSP pagrindu sukurti SRP ir SRTCP protokolai kurie koduoja visus perduodamus duomenis[22][6]. Visi išvardyti transliavimo metodai turi savus privalumus ir trūkumus, gerai yra tai, kad nėra ribojamas formatas kuriuo turės būti perduodamas vaizdas, mažinamas centrinio serverio apkrova vykdant retransliaciją, bet pagrindinis trūkumas, kartu ir privalumas yra viso srauto šifravimas, neapibrėžtas šifravimo metodas, numatytas šifravimo rakto keitimas vykstant srautinei transliacijai. Pilnas srauto kodavimas užtikrina aukščiausią

saugumo lygį, tačiau vaizdo srautui iššifruoti reikia didelių resursų, kurių mobilieji įrenginiai ir vaizdo priedėliai neturi.

2.6.2 Koduoto vaizdo pasirenkamasis šifravimas

Naudojant pasirenkamąjį šifravimo būdą yra išskiriami 4 šifravimo lygmenys, užtikrinantys skirtingus saugumo lygius.

Pirmas ir mažiausiai saugumo teikiantis metodas yra visų antraščių kodavimas. Užkodavus antraštes dekodavimo įrenginys nežino kokiu formatų yra perduodami duomenys, tačiau tokia apsauga nevertėtų pasitikėti, nes tiesioginėse transliacijose yra naudojami visiems žinomi standartai, tokiu būdu galima antraštes nuspėti.

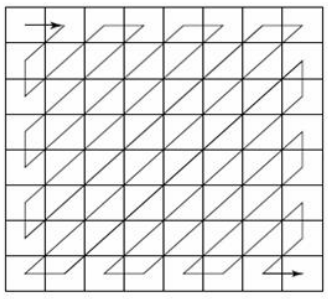
Antro lygmens saugumui užtikrinti šifruojamos visos antraštės ir I kadrai.

Trečiuoju metodu šifruojami visi kadrai ir makro blokai, antraštės nėra šifruojamos, šis metodas užtikrina pakankamą saugumo lygį, nes net ir žinant kaip užkoduotas vaizdas, neįmanoma jo atstatyti nes turinys yra šifruotas.

Ketvirtas ir didžiausią saugumą užtikrinantis metodas yra pilnas šifravimas[5].

2.6.3 Zig-Zag šifravimas

Apsauga įgyvendinat viename iš H.264 kodavimo algoritmo žingsnių, Zig-Zag matricos skaityme. Naudojant [17] metodą yra pakeičiamas Zig-Zag skaitymas, 11 pav. pavaizduotas standartinis skaitymo modelis. Jeigu skaitymo metodas nėra žinomas, baitai eilutėje bus sudėlioti nežinoma tvarka, dekodavimo įrenginys nežinodamas šios tvarkos negalės teisingai surinkti matricos tolesniam vaizdo dekodavimui. 11 pav. taip pat pateikta standartinė MB matrica atlikus DCT kodavimą, kaip matome naudojant standartinį Zig-Zag metodą, nenuliniai koeficiantai sudedami eilutės pradžioje, todėl taikant Run Length Encoding metodą, duomenys yra gerai suspaudžiami, panaudojus kitokį, sumaišyta metodą blogėja suspaudimo laipsnis.



$$\begin{bmatrix} 512 & 66 & -10 & 0 & 0 & 0 & 0 & 0 \\ -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

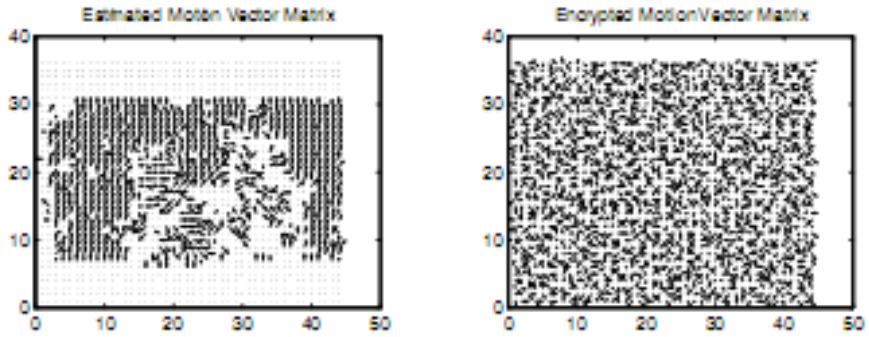
11 pav. Zig-Zag skaitymas, ir standartinė MB matrica po DCT kodavimo.[17]

Taikant šį apsaugos metodą gali ženkliai pablogėti suspaudimo laipsnis. Autorių teigimu Zig-Zag šifravimo metodas užtikrina aukštą apsaugos lygį. Norint pakeisti nuskaitymo kelią reikia dalinai perkoduoti visą vaizdo srautą.

2.6.4 Poslinkio vektorių šifravimas

Vaizdas susideda iš kadrų, kadrai iš I, P ir B tipo MB, P arba B tipo makro blokai turi poslinkio vektorių, kurie nusako kuri kadro dalis turėtų būti "nukopijuota" į atitinkamą MB. Vektorius gali rodyti į tą patį arba kitą kadrą. MVEA(Motion Vectro Encryption Algorythm) apsauga remiasi tuo, kad apie 70% makro blokų yra P arba B tipo, ir tai sudaro tik apie 30% srauto. Poslinkio vektorių kodavimo algoritmas, susideda iš dviejų lygių

1. Užšifruojami poslinkio vektoriai XOR algoritmu
2. Pasinaudojant kitų šifravimo algoritmu sumaišomi poslinkio vektoriai[10].



12 pav. Judesio vektorių kodavimas.[10]

2.6.5 DCT koeficientų šifravimas

VEA algoritmas pasiūlytas [7] straipsnyje, suskirsto visus DCT koeficientus į 3 saugumo lygius, kiekvienam lygmenyje yra skirtingas kiekis koeficientų

- Pirmame įtraukti 0-4
- Antrame 5-19
- Likę 44 koeficientai yra trečiame lygmenyje.

Koduojami tik tie lygiai kuriuos reikia, norimam saugumo lygiui pasiekti, 11 pav. matome kad koeficientai išsibarstę dažniausiai pirmame matricos ketvirtyje, t.y VEA metodo pirmas ir antras lygmuo. Pagal pateiktą metodą, 1 lygio šifravimas suteikia žemiausią apsaugą, 1+2 aukštesnis 1+2+3 saugiausias šifravimas. Toks metodas suspaudimo neturėtų įtakoti stipriai, nes 0 koeficientai nėra šifruojami.

Paprastesnis metodas pasiūlytas [9] straipsnyje, koduojami yra tik DCT matricos koeficientų ženklai. Koduojant tik ženklus, yra galima pakeitimo ataka, nes pasirinkimas koeficiento ženklui nėra didelis + arba -. [11] straipsnyje pasiūlyta koduoti pokyčio vektoriaus kryptis, kadangi dauguma MB yra P arba B tipo, teisingas jų atstatymas nebus įmanomas, ir matysis tik I tipo MB.

2.6.6 Dalinis šifravimas

Metodas nėra sudėtingas tačiau veiksmingas, H.261 formato dalinis šifravimas aprašytas pasirenkant pagal kadro tipą, I, P arba B[5], nustatyta, kad užtenka užkoduoti 144 vaizdo baitus norint pasiekti gerą saugumo lygį, kai dekoduojama JPEG, ir 16 baitų su H.261. Tačiau, MPEG-4 formate, kiekvienas kadras yra suskirstytas į I, P ir B makro blokus(MB), kiekvieno tipo makro blokų kodavimas iškraipo vaizdą[8]. Koduoti vien I tipo makro blokus ir neatsižvelgiant į P ir B nėra saugu, nes dalinai vaizdo kadrus galima atkurti iš judesio vektorių. Koduojant dalį visų makro blokų tipų pasiekiamas aukštas saugumas, šifravimui panaudotas DES algoritmas[8].

2.7 Šifravimo algoritmai

Vaizdo transliacijoms apsaugoti naudojami šifravimo algortimai, skirtingi algoritmai užtikrina skirtingą apsaugos lygį, taip pat reikalauja skirtingo resursų kiekio užšifruoti ir iššifruoti transliacijai. Trumpai aprašyti darbe naudojami ir panašiuose darbuose naudoti algoritmai.

2.7.1 AES šifravimo algoritmas

AES šifravimo algoritmas atitinka FIPS standartus ir laikomas saugiu informacijai šifruoti[31]. Informacija šifruojama simetriniu raktu. Informaciją skaldant į 128 bitų ilgio blokus. AES algoritmas veikia šifruodamas sudalintą informacijos bloką į 4x4 matricą, su matrica atliekamos sukeitimo, perstūmimo ir XOR su išplėstu šifro raktu operacijos. Norint pasiekti gerą saugumo lygį operacijas reikia pakartotinai įvykdyti 10 kartu su 128 bitų raktu, 12 su 192, ir 14 su 256 bitų ilgio raktu.

2.8 Vaizdo kokybės vertinimas

Siekiant užtikrinti gerą vaizdo apsaugos kokybę, būtina įvertinti ar šifruotas srautas yra netinkamas peržiūrai ir kadrai yra neatkuriami srauto neiššifravus. Siūlomo metodo tinkamumą šifruoti transliacijas vertinsime PSNR ir DSSIM metodais.

2.8.1 Peak signal-to-noise ratio vertinimas

PSNR(Peak signal-to-noise ratio) yra naudojamas ir vaizdo kokybės vertinimui, bendru atveju šis metodas nusako santykį tarp maksimalaus signalo ir triuksmų kurie įtakoja signalą. Šis metodas naudojamas įvertinti kokį poveikį turi vaizdo kodavimo algoritmai, kuo didesnis PSNR tuo vaizdo pakitimas mažesni. PSNR apskaičiuojamas[32].

Suskaičiuojamas mažiausia šaknies paklaida, kur max bespalvis paveikslas, I ir K paveikslai

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - K(i,j) \right]^2$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

kur MAX_i yra didžiausia galima taško reikšmė, kai taškas išreikštas 8bit spalva.

Suspaudimo algoritmų PSNR svyruoja nuo 30 iki 50 dB. PSNR nėra pats geriausias metodas vertinti vaizdo kokybę, todėl gali būti naudojami pažangesni vertinimo metodai. Šio metodo trūkumas yra tai, kad jis vertina skirtumą tarp taškų, o ne bendrą paveikslo struktūra.

2.8.2 Struktūrinis panašumas

SSIM(structurial similarity) indeksas nusako panašumą tarp dviejų paveikslų[33]. Tarp paveikslo x ir y esant dydžiui NxN SSIM skaičiuojamas sekančiai:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

kur:

μ_x - x paveikslo vidurkis, μ_y - y paveikslo vidurkis

σ_x^2 - x pasiskirstymas, σ_y^2 - y pasiskirstymas

σ_{xy} - x ir y nepanašumas

$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ - kintamieji vardiklio stabilizavimui

$L = 2^{bits_{per_{pixel}}} - 1$ taško bitų atvaizdavimo kiekis

$k_1 = 0.01, k_2 = 0.03$ numatyta reikšmė

Struktūrinis nepanašumas(DSSIM) išreiškiamas per struktūrinį panašumą.

$$DSSIM(x,y) = \frac{1 - SSIM(x,y)}{2}$$

2.9 Analizės išvados

Tiesioginėse transliacijose naudojami RTP, SRTP, MPEG-2 TS ar tiesiog UDP protokolai. UDP protokolas neužtikrina paketų pristatymo ir nepakartoja prarastų paketų persiuntimo. RTP/RTCP ir SRTP/SRTCP protokolai veikiantys naudojantis UDP paketais turi pristatymo kokybes vertinimo metodus.

MPEG-2 vaizdo transliacijos srautas perduodamas UDP paketais. MPEG-2 numato transliacijos kokybės klientui informacijos persiuntimą atgal į serverį. Standarte numatytas atsiradusių klaidų taisymo mechanizmas labai mažo patikimumo linijose.

Atliekant vaizdo kodavimą H.264/AVC formatu yra atsižvelgiama į tai, kad keli paketai gali būti prarasti, ar nepataisomai sugadinti, dėl to užkodavimo žingsniuose naudojamas perteklinis kodavimas. Siekiant sumažinti prarandamos informacijos kiekį MPEG-2 standartas numato tik 188B

dydžio paketus, tokio paketo praradimas nesukeltų visos transliacijos nutrūkimo, yra suderinamas su plačiai paplitusiais ATM tinklais, palydoviniais transliacijos kanalais.

Išanalizavus H.264/AVC šifravimo metodus sudaryta lentelė, kurioje pateikiami pagrindiniai principai kurie būdingi apsaugos algoritmams.

3 lentelė. šifravimo algoritmų palyginimas

Metodas	Nepriklauso nuo kodavimo algoritmo	Nekeičia kodavimo algoritmo	Neįtakoja vaizdo suspaudimo	Kintamas saugumo užtikrinimas	Nedidelis šifruojamos informacijos kiekis	Naudojimas netiesioginėje trans.
Dalinis šifravimas	-	+	+	+	-	-
Zig-Zag	-	-	-	-	+	+
Vektorių	-	-	-	-	+	+
DCT koef.	-	-	-	+	+	+
Pasirenkamasis	-	+	+	+	-	-
Dalinis MPEG-2 šifravimas	+	+	+	+	-	-

3 lentelėje pateiktas trumpas apžvelgtų šifravimo metodų palyginimas.

- Nepriklauso nuo kodavimo algoritmo – Nepriklauso nuo vaizdo kodavimo algoritmo.
- Nekeičia kodavimo algoritmo – Vaizdo kodavimo algoritmas nėra keičiamas, ar kitaip modifikuojamas(žr 2.4 poskyrį).
- Neįtakoja vaizdo suspaudimo – Keičiant kodavimo algoritmą gali būti pablogėti vaizdo suspaudimo lygis, kodavimo algoritmo klaidų atstatymo metodai gali nebeveikti.
- Kintamas saugumo užtikrinimas – Keičiant duomenų šifravimo vietą ir kiekį gali būti pasiektas reikiamas saugumo lygmuo.
- Nedidelis šifruojamos informacijos kiekis – Šifruojamos informacijos kiekis gali išaukti šifruojant nekritinius duomenis vaizdo atgaminimui.
- Naudojimas netiesioginėje trans – Nesudėtingas panaudojimas ne tiesioginėse transliacijose, vaizdo medžiagos parsisiuntimas.

Išanalizuoti Zig-Zag, vektorių ar keičiant DCT koef. metodai išnaudoja H.264/AVC vaizdo kodavimo sritis, kuriose padaryti pakeitimai, duomenų užšifravimas, turi didelę įtaką sėkmingam vaizdo atkūrimui. Naudojant šifravimą kodavimo lygmenyje gali būti pasiektas aukštas efektyvumo

lygmuo, tačiau, tai labai sunkiai suderinama su spartintuvais. Norint pakeisti šifravimo algoritmą ar slaptą raktą, reikia perkoduoti bent dalį vaizdo failo(žr. 9 pav. H.264/AVC kodavimo žingsniai).

Siūlomas dalinis MPEG-2 transliacijos srauto šifravimo metodas duomenis šifruoja neatsižvelgdamas į jų paskirtį (vektorius ar I tipo makro blokas), taip išvengiama skaičiavimų reikalingų surasti I makro blokus ar vektorius. Norint pakeisti šifravimo algoritmą ar slaptą raktą, reikia peršifruoti tik srautą, kuris gali būti paruoštas transliacijai nešifruotas iš anksto, taip sutaupoma serverio resursų.

3 DALINIS MPEG-2 TRANSPORTO SRAUTO ŠIFRAVIMO METODAS

Išanalizavus plačiausiai naudojamus vaizdo perdavimo metodus ir protokolus nustatyta, kad MPEG-2 transliavimo srautas susidedantis iš 188B(13 pav. Transport Stream paketas) ilgio paketų yra gera vieta šifruoti duomenims. Paketus galima sumaišyti ar šifruoti, reikia atsižvelgti į tai, kad H.264/AVC suspaudimo formate numatytas klaidingų kadrų taisymas, dėl to per mažas šifruojamų duomenų kiekis gali neužtikrinti saugumo. Televizijos transliacijoms žiūrėti naudojami STB, mobilieji įrenginiai, kurie nėra pakankamai spartūs dešifruoti sudėtingais algoritmais šifruotą didelį duomenų srautą. Siekiant apsaugoti vaizdo transliaciją ir sumažinti resursų poreikį galima:

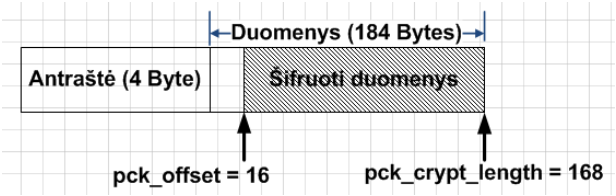
- Šifruoti dalį duomenų.
- Nešifruoti lengvai nuspėjamų duomenų.
- Nešifruoti transliacijos sinchronizacijos duomenų.

MPEG-2 TS standarte vaizdo srautas yra suskaidytas į 188B dydžio paketus, kuriu surinkimo tvarka privalo būti teisinga, kitaip vaizdo dekodavimas nebus įmanomas, iš to seka, kad sumaišius paketus, ar dalį užšifravus, vaizdo atkurti nepavyks, nežinant algoritmo ar šifro raktų. Šis metodas taip pat būtų suderinamas su aparatiniu vaizdo dekodavimu, nes duomenys jau būtų dešifruoti prieš juos perduodant dekodavimo įrenginiui. Šifruojant duomenis pasirinktoje vietoje nereikalingas vaizdo perkodavimas, norint jį užšifruoti skirtingais metodais ar raktais. Pvz, naudojamas transliacijos keletui klientų metodas, visi klientai suskirstyti į grupes siekiant šiek tiek sumažinti serverio apkrovas, kiekvienai grupei transliuojamas vaizdas koduojamas skirtingais šifro raktais.

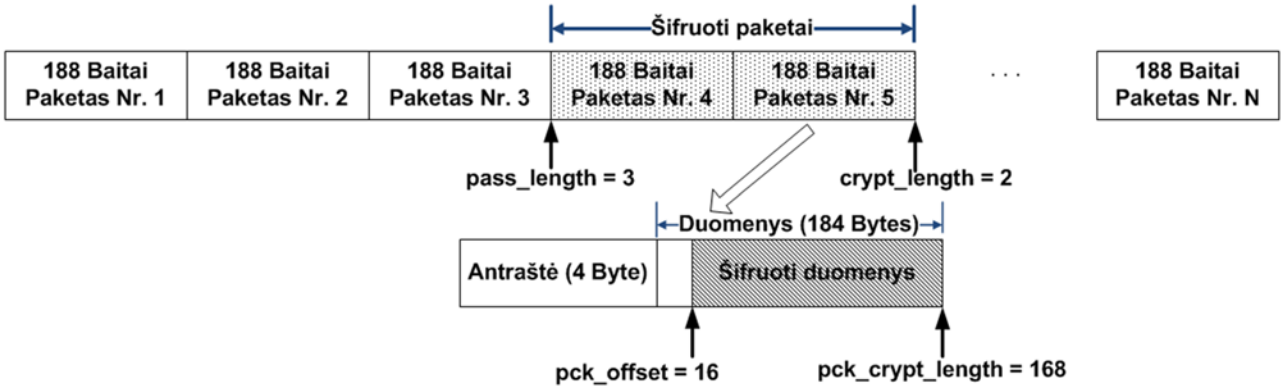
3.1 Dalinio MPEG-2 transporto srauto šifravimo metodo aprašymas

Transliacijos srauto siūlomas apsaugos metodas šifruoja dalį 188B ilgio MPEG-2 transporto srauto paketo(žr. 13 pav.) Paketas sudarytas iš 4B antraštės ir 184B duomenų, antraštės ir duomenų ilgiai gali skirtis, jei naudojamos papildomos vėliavėlės antraštėje.

Standarto suderinamumui, paketo antraščių nešifruosime, bus šifruojama tik dalis duomenų(žr. 14 pav.). Eksperimento metu bus nustatyta kokią dalį duomenų reikia užšifruoti norint apsaugoti vaizdo transliaciją. Kokybės vertinimui naudosime PSNR ir DSSIM kadrų palyginimo metodą.



13 pav. Dalinai šifruoti transporto paketo duomenys.[34]



14 pav. Siūlomo metodo šifravimo principinė schema.[34]

Eksperimentai bus atliekami keičiant *pass_length*, *crypt_length* sekos parametrus, ir paketo šifravimo parametrus – *pck_offset* ir *pck_crypt_length*.

3.1.1 Transliacija panaudojant dalinį MPEG-2 šifravimo metodą

Vaizdo transliacijos principinis modelis pavaizduotas 15 pav. bendrais bruožais aprašo vaizdo pristatymą ir žingsnius reikalingus transliacijai vykdyti. Klientas siunčia užklausą transliacijos serveriui, serveris inicijuoja vartotojo užklausos vykdymą.

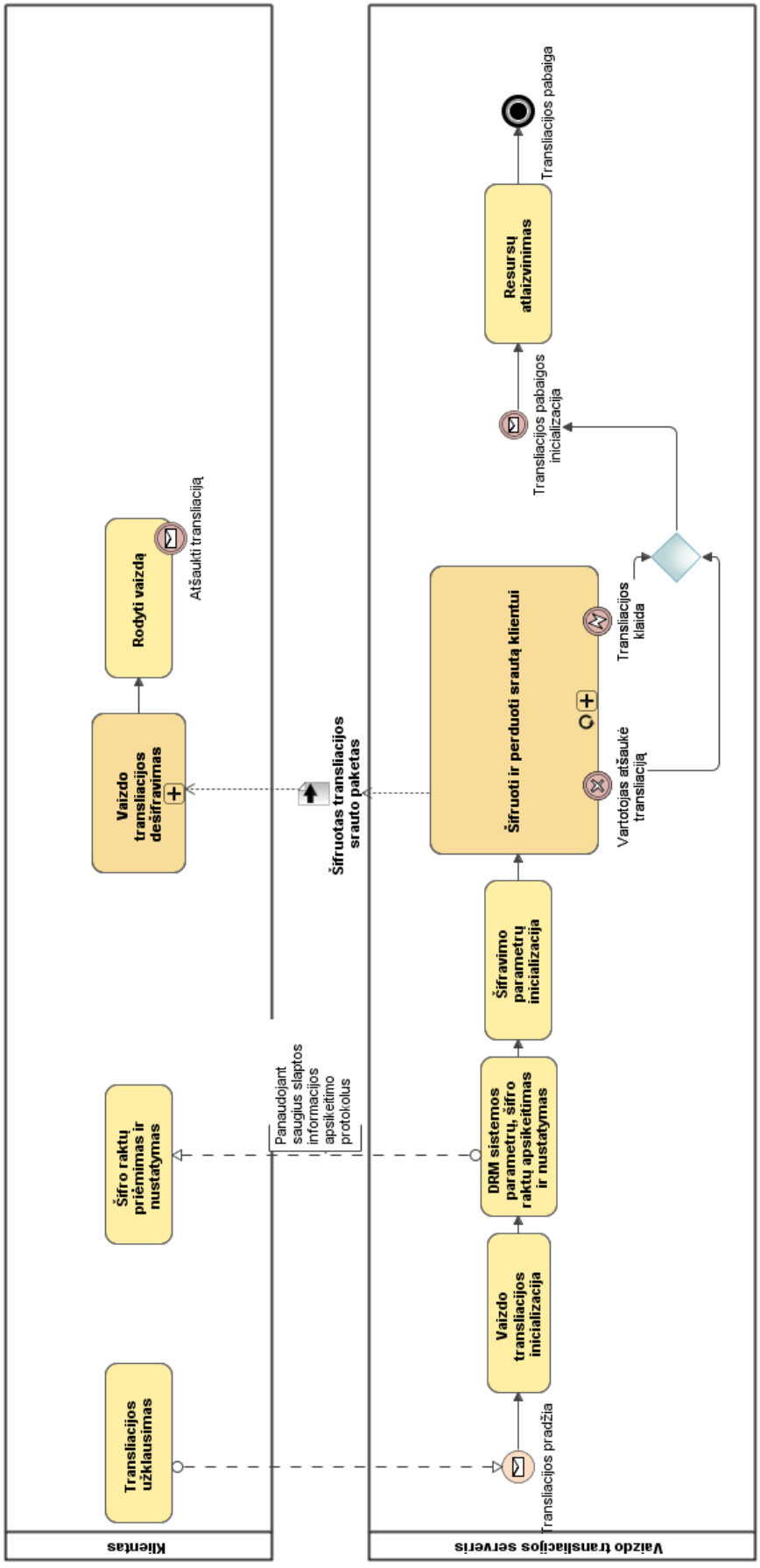
DRM sistemos parametrų, šifro raktų apsikeitimo ir nustatymo žingsnyje vykdomas sesijos raktų sukūrimas ar esamų panaudojimas, priklausomai nuo konkrečios sistemos realizacijos, toliau šifro raktai siunčiami klientui. Visa šifravimo informacija siunčiama tik saugiu kanalu. Šiame žingsnyje taip pat gali būti tikrinamos vartotojo teisės į norimą transliaciją.

Vaizdo transliacijos šifravimo ir perdavimas klientui(serveryje) ir vaizdo transliacijos dešifravimas(klientas) žingsniuose yra naudojamas siūlomas vaizdo transliacijos apsaugos metodas. Smulkesnis žingsnio išskaidymas pateiktas 16 pav.

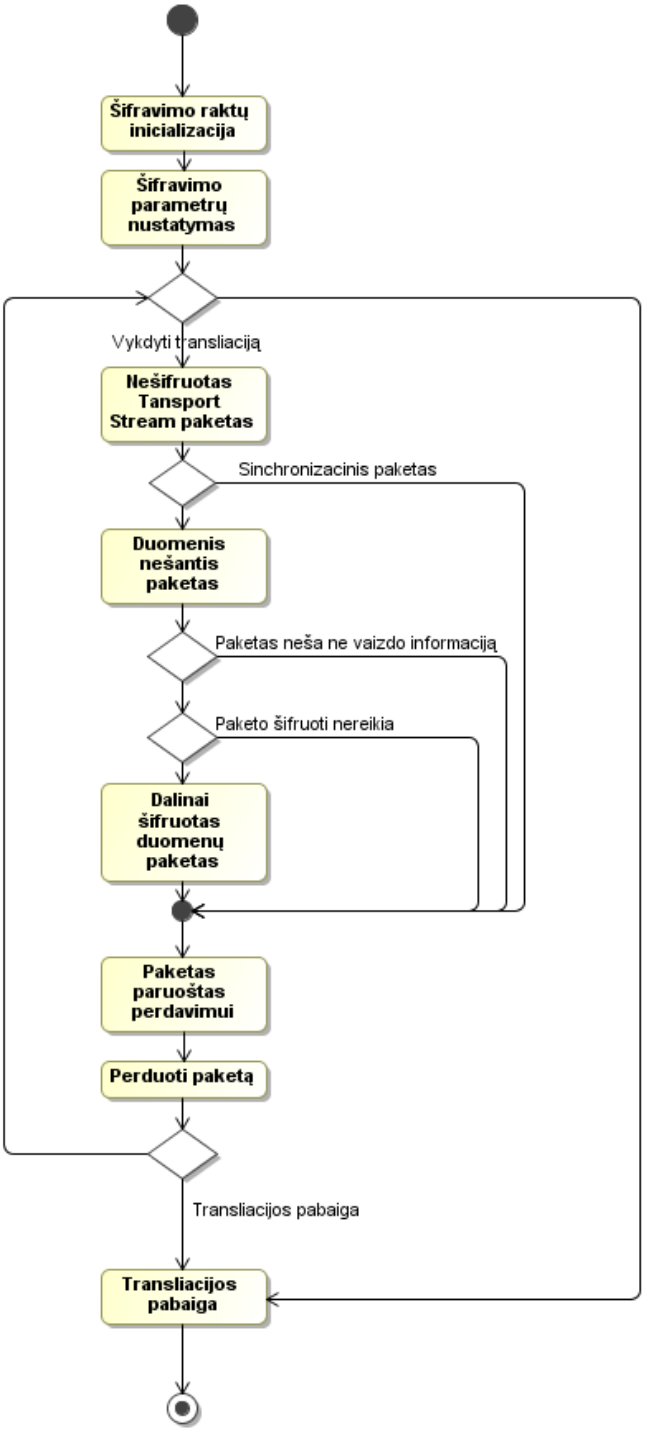
Srautas perduodamas klientui tol, kol jis nenutraukia transliacijos. Kiekvienam klientui vaizdas gali būti šifruotas skirtingais arba tai pačiais šifro raktais ir parametrais, priklausomai nuo inicializacijos parametrų, tai apsprendžiama konkrečiam taikymo pavyzdyje, ir šifravimo algoritmui įtakos neturi.

Klientas gavęs transliacijos paketus juos dešifruoja, 188B dešifruoti paketai yra apjungiami į MPEG-2 standarto srautą(žr. 2.5 poskyrį) kuris yra perduodamas H.264/AVC vaizdo dekodavimo įrenginiui.

Transliacijai pasibaigus ar ją nutraukus sunaikinami sesijos šifro raktai ir atlaisvinami serverio resursai kitiems klientams.



15 pav. Vaizdo transliacijos principinis modelis(BPMN notacija)



16 pav. Vaizdo transliacijos paketo atrinkimo šifravimui modelis(UML notacija)

3.1.2 Užšifravimo algoritmas

Vaizdo transliacijos paketo atrinkimo algoritmas pavaizduotas paveiksle. Algoritmas susideda iš tokių žingsnių:

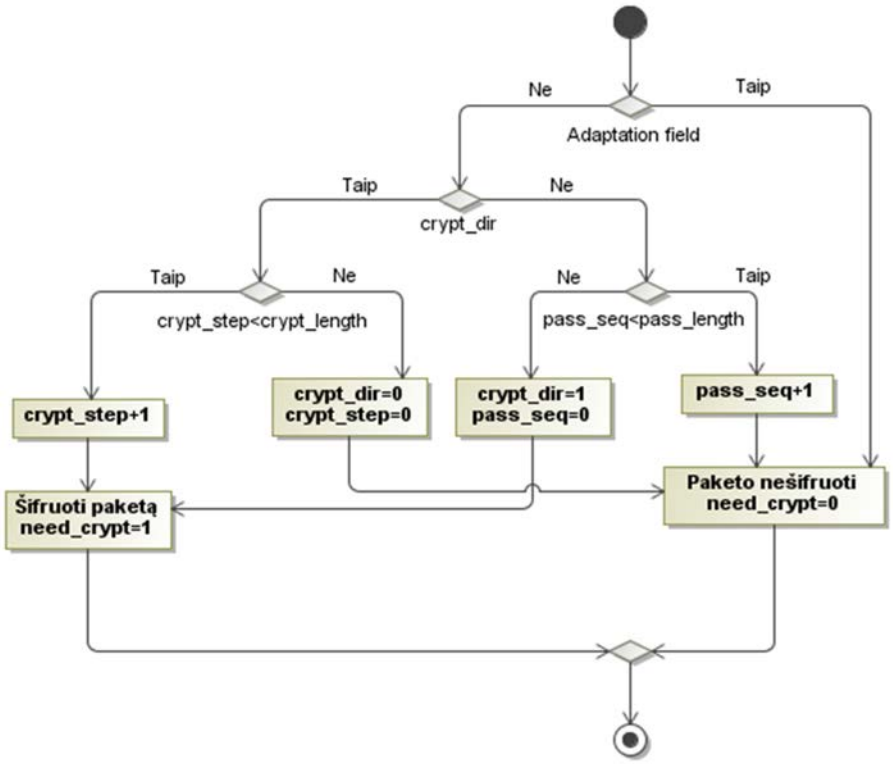
1. Transliacijos pradžioje yra sukuriami saugūs šifravimo raktai AES algoritmui. Raktai klientui pasiekia per DRM sistemą.
2. Nustatomi pradiniai parametrai, srauto sekos šifravimui naudojami 2 parametrai(žr. 14 pav.)

- pass_length – nešifruojamų paketų sekos ilgis, po kurio seka crypt_length.
- crypt_length – šifruojamų paketų sekos ilgis.

Paketo duomenų šifravimui naudojami kiti 2 kintamieji, nusakantys šifruojamą dalį:

  - pck_offset – baitas nuo kuriuo reikės šifruoti duomenis pakete.
  - pck_crypt_length – baitų kiekis kuris turi būti užšifruotas. Turi būti 16 kartotonis naudojant AES šifravimą, kad šifruoti duomenys užimtų tiek pat vietos kaip ir nešifruoti.

3. Jei paketas turi nustatytą adaptation fields jis nėra šifruojamas.
4. Tikrinamos pass_length ir crypt_length reikšmes, jei paketą reikia šifruoti, užšifruojama jo dalis priklausomai nuo pck_offset ir pck_crypt_length parametrų. Šifruotam paketui nustatoma transporto srauto paketo scr vėliavėlė į 1, tai reiškia kad paketas šifruotas.
5. Tikrinamas sekantis srauto paketas(3 punktas).



17 pav. Duomenų paketo atrinkimo šifravimui algoritmo schema

Paketo išrinkimo algoritme naudojami laikini išrinkimo kintamieji:

- *crypt_dir* – nurodo kuriai sekai priklausys einamasis paketas, šifravimo ar praleidimo.
- *crypt_step* – sekoje šifruotų paketų skaičius.
- *pass_step* – nešifruotų paketų skaičius.

- *need_crypt* – vėliavėlė nurodanti, kad paketą reikės šifruoti.

Išrinkimo algoritmui naudojami kintamieji *crypt_length*, *pass_length* patikrinami prieš pradedant transliacijos šifravimo algoritmą. Algoritme yra 3 kraštutiniai atvejai:

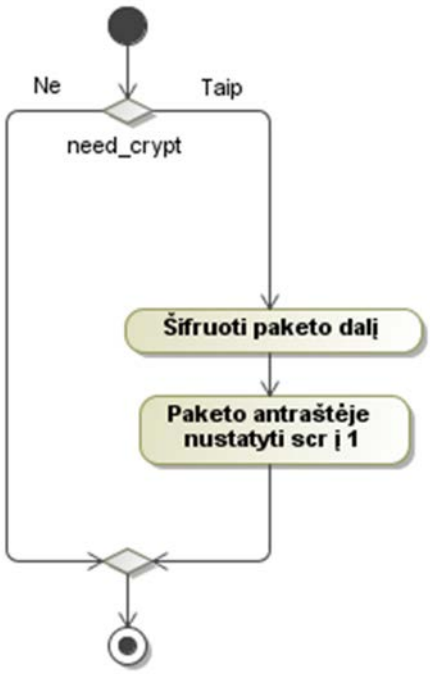
- Jei *pass_length*=0 *crypt_length*>0, bus užšifruoti visi paketai neturintys papildomos informacijos antraštėje.
- Jei *pass_length*>0 *crypt_length*>0, paketai bus šifruojami intervalais.
- Jei *pass_length*>0 *crypt_length*=0, paketai visai nebus šifruojami, taip galima įvertinti išrinkimo algoritmo greitį.

Algoritmas veiks tik tada, jei *crypt_length* arba *pass_length* bus daugiau už nulį.

Paketo šifravimo algoritmas(1817 pav.) nėra sudėtingas. Paketo išrinkimo algoritmas nustato *need_crypt* vėliavėlę jei paketas turi būti užšifruotas. Paketo dalies šifravimas vyksta AES algoritmu(žr. 2.7.1).

Paketo antraštėje esanti *scr* vėliavėlė nustatoma į 1, taip iššifruojant duomenis nereikalingas paketų išrinkimo algoritmas.

Algoritmas veikia tik tada, jeigu *pck_crypt_length* yra 16 kartotinis, ir *pck_offset*+*pck_crypt_length*<=180, t.y. Pakanka duomenų ilgio užšifravimui. Tikrinimai atliekami pradedant transliacijos šifravimo algoritmą.



18 pav. Paketo šifravimo algoritmas

Atliekant duomenų šifravimą užkoduotų duomenų lygmenyje t.y. perdavimo lygmenyje, pasiekiamas lankstesnis sistemos pritaikymas. Perdavimo lygmenyje šifruojant vaizdo srautą, vaizdas gali būti koduotas skirtingais kodavimo metodais, tai įtakos šifravimui neturi.

Siūlomas apsaugos metodas duomenis šifruoja perdavimo lygmenyje, MPEG-2 transporto srauto paketuose. Šifruojamas 188B ilgio paketas pagal standartą jau turi vėliavėlę žyminčia kad jis yra užšifruotas(žr 2.4.5), ji bus naudojama pagal paskirtį, žymėti šifruotiems paketams.

Duomenys bus šifruojami AES metodu kuris užtikrina reikiamą saugumo lygį ir iššifravimas nereikalauja labai didelių skaičiavimo resursų.

3.1.3 Iššifravimo algoritmas

Iššifravimo algoritmas yra daug paprastesnis nei užšifravimas, norint srautą iššifruoti pakanka žinoti slaptą raktą, *pck_offset* ir *pck_crypt_length* parametrus. Iššifravimas atliekamas sekančiais žingsniais:

1. Patikrinama ar paketo SCR(užšifruota) vėliavėlė nustatyta.
2. Jei SCR vėliavėlė nustatyta, iššifruoti duota dalį paketo duomenų.

19 pav. Paketo iššifravimo algoritmas

4 MPEG-2 TRANSLIACIJOS SRAUTO DALINIO ŠIFRAVIMO GREITAVEIKOS IR PATIKIMUMO NUSTATYMAS

Siūlomo dalinio MPEG-2 transporto srauto šifravimo metodui ištirti bus sukurta bandomoji programa, atliekanti šifravimą/iššifravimą transliacijos srautui išsaugotam faile.

Kadangi skirtingos tematikos vaizdas koduojamas skirtingai(žr. 2.5), ir transliacijoje bus skirtingas svarbios informacijos kiekis iš kurios bus galima atkurti vaizdą. Nustatyti pakankamą kiekį informacijos, kuris turi būti užšifruotas siekiant gauti norimą saugumo lygį.

Vaizdo kokybė vertinama PSNR(žr. 2.8.1) ir DSSIM(2.8.2 Struktūrinis panašumas) metodais. Vaizdo kodavimui leidžiamos PSNR reikšmės yra nuo 30 iki 50 decibelų, kur didesnis PSNR reiškia geresnę vaizdo kokybę. Jei PSNR yra mažiau nei 15 vaizdas yra blogos kokybės, kadre esančios informacijos išskirti negalima, matoma tik vaizdo šiukšlės. PSNR vertei esant 15 vaizdo kadrai yra neatkuriamai neteisingi. DSSIM nusako vaizdo nepanašumą, kuo didesnė reikšmė, tuo vaizdai daugiau skiriasi, ir esant 0 – vaizdai identiški.

Eksperimentai bus atliekami su dviejų tipų vaizdo įrašais.

- Statinis vaizdas, užkoduotas vienas paveiksliukas į 2min vaizdo srautą.
- Dinaminis vaizdas, vaizdas kuriame yra daug judančių objektų pvz. raibuliuojantis vanduo, žmonių minia.

Eksperimento metu siekiama nustatyti:

1. Šifravimo parametrus, užtikrinančius saugumą, statiniam ir dinaminiam vaizdui šifruoti.
2. Nustatyti šifruojamų ir nešifruojamų paketų kiekį paruoštoje transliacijoje.
3. Pamatuoti šifravimui ir iššifravimui reikalingą laiką.
4. Suskaičiuoti PSNR ir DSSIM reikšmes vaizdo kadrui.

4.1 Eksperimento atlikimo aplinka

Siekiant, kad eksperimentui neturėtų šalutinės įtakos tokie veiksniai kaip:

- Operacinės sistemos vartotojo sąsaja.
- Operacinės sistemos pašalinių procesų veikla.
- Vaizdui koduoti ir dekoduoti naudojamų programinių paketų įtaka realiu laiku.

Eksperimentas bus atliekamas ant nedidelius skaičiavimo resursus turinčio įrenginio. Vaizdas užkoduotas H.264/AVC formatu. MPEG-2 transporto srautas išsaugotas faile.

4.1.1.1 Techninė įranga

Eksperimentas atliekamas naudojant MiniPC kompiuterį su Ubuntu 11.10 Linux pagrindo distribucija, be grafinės vartotojo sąsajos.

MiniPC kompiuterio techninės charakteristikos:

- 500MHz CPU
- 1 GB DDR2, 667MHz RAM
- 40GB HDD

4.1.1.2 Programinė įranga

Eksperimentui atlikti naudojama programinė įranga:

- Ubuntu 11.10 operacinė sistema, be grafinės sąsajos.
- pnmpsnr – programinis paketas skaičiuoti psnr reikšmę tarp dviejų paveikslėlių. Naudojamas palyginti kadrą tarp originalaus ir šifruoto, originalaus ir dešifruoto.
- perf stat – paketas skirtas programos veikimo ir sunaudotų resursų kiekiui stebėti. Naudojamas pamatuoti laikui kurio reikia norint užšifruoti ir iššifruoti visą transliacijos srauto failą.
- ffmpeg – vaizdo kodavimo įrankių rinkinys. Naudojamas transliacijos failo paruošimui, vaizdo kodavimui ir dekodavimui. Kadro ištraukimui iš transliacijos srauto failo palyginimui.
- qmake – c kompiliatorius testiniai programai sukompiliuoti.
- libcrypto++ - šifravimo algoritmų rinkinys, naudojamas paketo duomenims šifruoti.
- putty – SSH prieigos klientas, naudojamas komunikavimui su MiniPC.
- dssim – vazidų lyginimas DSSIM metodu.

4.2 Eksperimento atlikimo eiga

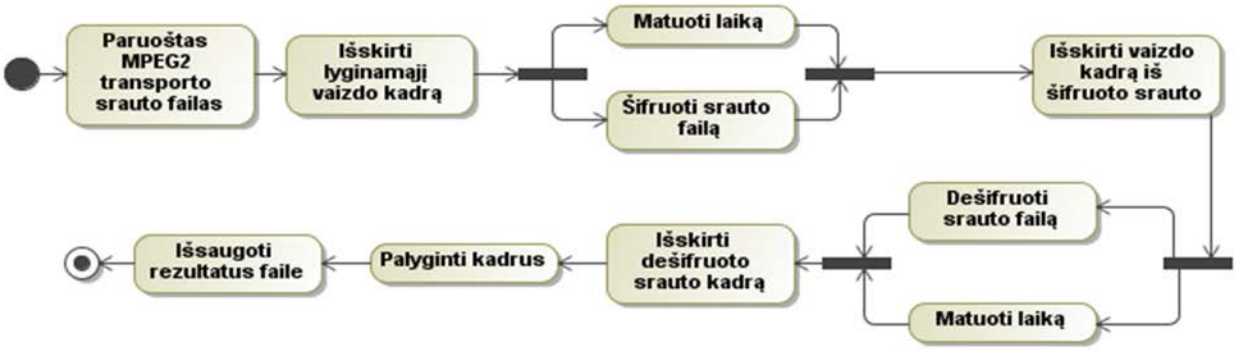
Eksperimentui atlikti aplinka turi būti paruošta, operacinė sistema turi būti Linux pagrindo. Kompiuteryje turi būti instaliuoti šie programiniai paketai:

- qmake
- libcrypto++
- perf stat
- pnmpsnr
- ffmpeg
- OpenCV ir DSSIM

Eksperimento rezultatų surinkimo žingsnių seka pavaizduota 20 pav.

- Iš paruošto MPEG-2 transporto srauto failo išimamas kadras PSNR ir DSSIM lyginimui.

- Naudojantis perf stat matuojamas srauto failo dalinio šifravimo laikas.
- Iš šifruoto srauto failo išimamas kadras PSNR ir DSSIM lyginimui.
- Matuojamas šifruoto srauto failo dešifravimo laikas.
- Gauti rezultatai surašomi faile.



20 pav. Eksperimento rezultatų surinkimo veiksmų seka

Siūlomas dalinis MPEG-2 transliacijos srauto šifravimas patikritas atliekant testus. Eksperimentai atlikti Ubuntu 10.10 operacinėje sistemoje, miniPC kompiuteryje su Intel atom 500MHz procesoriumi. Paketams šifruoti ir dešifruoti parašyta metodą realizuojanti bandomoji programos versija. Šifravimo ir dešifravimo laikai matuoti naudojantis *perf* įrankiu. PSNR skaičiavimamas naudojamas *ffmpeg* programinis paketas.

Apsaugos lygio įvertinimui atlikti matavimai, siekiant nustayti ribą, kai apsauga nebeužtikrinama. Parinkti tokie bandymų parametrai:

- *pass_length* – 0, 1, 2, 3, 6, 9, 12, 15, 19, 24, 32, 64
- *crypt_length* – 1, 2, 3, 6, 9, 12
- *pck_crypt_length* – 16, 32, 64, 128, 176(skaičiuojant nuo paketo pabaigos)

Vertinamo vaizdo kokybė yra bloga, jei PSNR reikšmė neviršija 15dB, DSSIM atveju daugiau nei 0,5. Atliekant eksperimentą šifravimas ir iššifravimas atlikti 5 kartus, laiko reikšmė yra matavimų vidurkis. Užšifravimo ir iššifravimo laikai eksperimento metu nesiskyre daugiau kaip 10%. Visose eksperimentų lentelėse pateikti sutrumpinti rezultatai.

Eksperimentai atliekami su vaizdo įrašais skirtais bandyti vaizdo suspaudimo formatus, filtrus ir pan. Vaizdai pasižymi įvairiomis savybėmis kurios neturėtų sukelti sutrikimų ar nenumatytų rezultatų algoritmams.

4.3 Eksperimento rezultatai

Siekiant geriau įvertinti apsaugą kurią užtikrina siūlomas metodas atlikti bandymai su kelių tipų vaizdo įrašais. Statiniais ir dinaminiais vaizdais, statinis vaizdas buvo sugeneruotas pasinaudojant paveikslėlį, dinaminių vaizdų tyrimui naudoti bandomieji vaizdo įrašai.

4.3.1 Dinaminio vaizdo vertinimas

Eksperimentai atlikti naudojant vaizdo įrašus kuriuose yra objektų dinamikos, judesio, vaizdų kaita. Bandymai atliekami su keletų vaizdo įrašų, siekiant geriau nustatyti siūlomo metodo apsaugą vaizdo transliacijai.

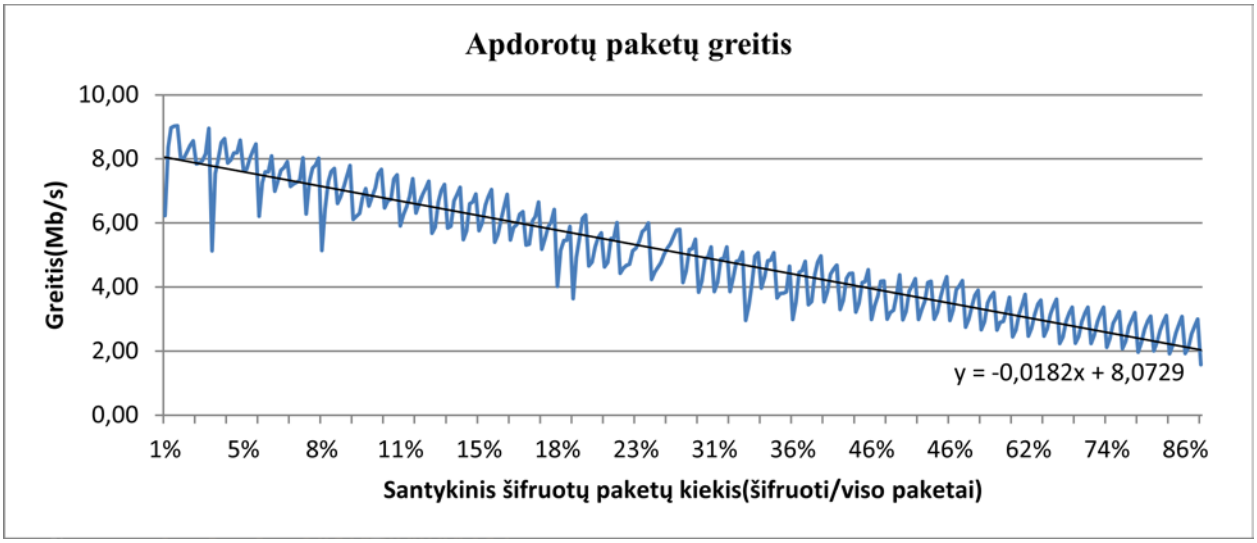
4.3.1.1 Akiyo vaizdo įrašas

4 lentelėje pateiktas laiko ir kokybės vertinimas dinaminiam vaizdui. Atlikus eksperimentus nustatėme, kad šifruojant žingsniu 32-1(*pass_length*=32-*crypt_length*=1), kas 32 paketą, ir pakete 176 paskutinius baitus, vaizdo kokybė yra labai prasta, ir neatstatoma. Lyginant šifravimo laiką kurio reikia užšifruoti 32-1 žingsniu su pilnu šifravimu gauname, kad laiko sąnaudos 5 kartus mažesnės. Šifruojant 24-1 žingsniu gauname šiek tiek didesnę apsaugą, nes daugiau duomenų yra šifruojama, šifravimo laikas nežymiai išauga, bet PSNR yra beveik lygus pilnai šifruotiems paketams.

4 lentelė. Akyio vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,93	0,38	11,3	1,17	2,78	0,42	11,4	1,15	2,52	0,60	11,4	1,10	**1,76**
3-1	0,23	0,18	11,1	1,33	6,00	0,18	11,3	1,18	5,73	0,23	11,4	1,22	4,59
3-2	0,37	0,23	11	1,23	4,68	0,23	12,2	1,27	4,55	0,30	12,1	1,12	3,54
3-9	0,70	0,31	11,3	1,18	3,37	0,33	11,3	1,21	3,18	0,47	11,4	1,11	2,24
3-12	0,74	0,33	11,4	1,12	3,24	0,35	11,3	1,18	3,06	0,50	11,4	1,10	2,12
6-1	0,13	0,16	11,1	1,32	6,60	0,15	11,7	1,29	6,90	0,19	10,7	1,28	5,48
6-2	0,23	0,18	11,7	1,27	5,81	0,20	11,1	1,29	5,41	0,24	11,4	1,13	4,43
6-9	0,55	0,28	10,9	1,21	3,83	0,29	10,9	1,20	3,70	0,40	11,6	1,16	2,66
6-12	0,62	0,29	10,4	1,37	3,59	0,31	10,8	1,17	3,47	0,43	10,8	1,31	2,46
12-1	0,07	0,13	11,3	1,06	8,03	0,14	12,2	1,09	7,34	0,15	12,3	1,34	7,14

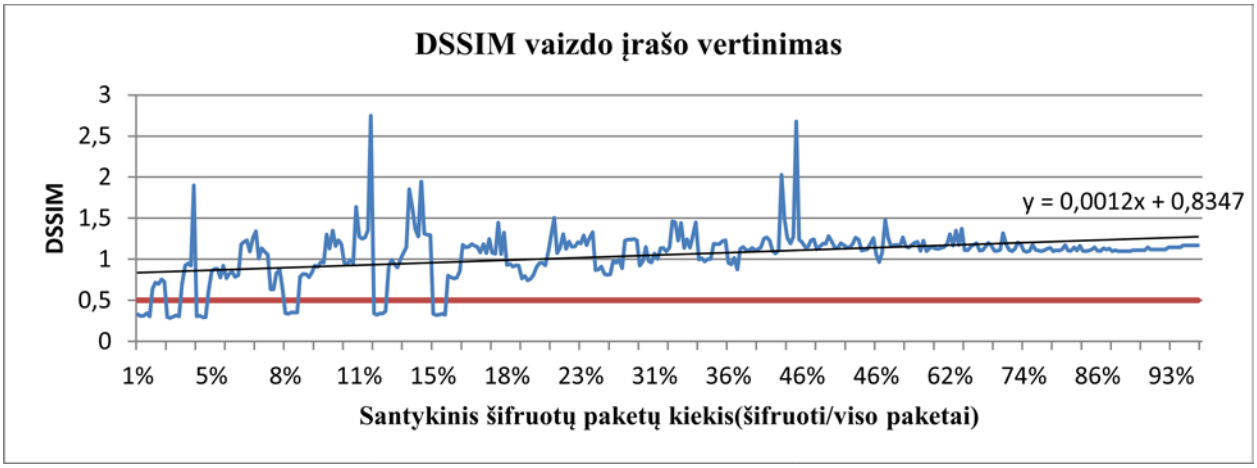
12-2	0,13	0,15	10,4	1,37	7,11	0,15	9,9	1,64	6,88	0,18	11	1,08	5,84
12-9	0,40	0,24	9,63	1,50	4,43	0,24	7,3	2,03	4,42	0,32	10,9	1,10	3,30
12-12	0,46	0,25	10,5	1,26	4,20	0,26	9,9	1,48	4,03	0,36	11	1,05	2,96
24-1	0,04	0,21	12,5	0,69	5,13	0,12	13	1,90	8,64	0,13	12,3	0,95	7,97
24-2	0,07	0,13	13,7	0,65	8,02	0,14	12,4	0,88	7,80	0,17	13,9	0,63	6,28
24-9	0,25	0,18	12,3	1,00	5,78	0,19	12,3	0,96	5,59	0,24	12,4	0,88	4,47
24-12	0,31	0,21	12,3	1,01	5,07	0,22	12,3	1,00	4,84	0,27	12,4	1,00	3,97
32-1	0,03	0,13	12,1	0,70	8,20	0,13	12,9	0,64	8,02	0,13	11,9	0,76	**8,42**
32-2	0,06	0,14	11,9	0,85	7,59	0,17	13	0,77	6,21	0,15	11,9	0,82	7,26
32-9	0,20	0,17	12	0,81	6,25	0,17	13,1	0,76	6,14	0,21	12,1	0,80	4,93
32-12	0,25	0,18	11,8	0,89	5,80	0,20	13	0,82	5,21	0,25	11,9	0,87	4,24
64-1	0,01	0,12	13,3	0,23	9,04	0,12	12,9	0,35	9,03	0,17	13,3	0,25	6,23
64-2	0,03	0,12	13,3	0,31	8,95	0,13	12,9	0,32	8,16	0,13	13,4	0,29	7,88
64-9	0,11	0,15	13,2	0,34	6,89	0,14	12,9	0,37	7,30	0,17	13,3	0,34	6,31
64-12	0,15	0,15	13,3	0,33	7,04	0,16	13	0,34	6,82	0,18	13,3	0,34	5,77



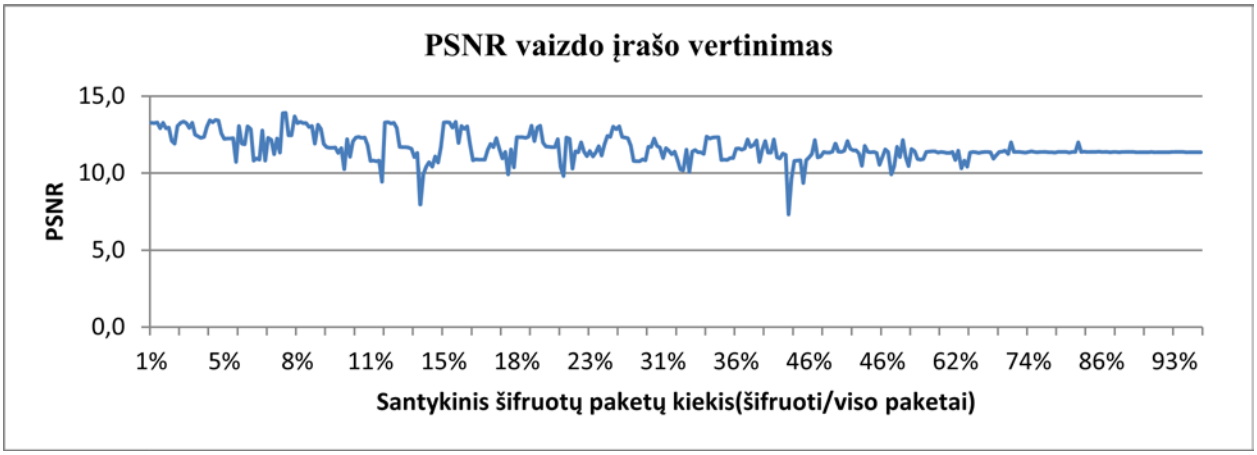
21 pav. Akiyo vaizdo įrašo apdorojamų duomenų greitis.

Pagal 4 lentelės duomenis nubraižytas apdorotų paketų greičio priklausomybės nuo šifruojamų paketų dažnumo grafikas, pateiktas 21 paveiksle. Grafike aiškiai matomas apdorotų paketų greičio augimas mažinant šifruojamų paketų kiekį. Nors ir rečiau šifruojant vaizdo srauto duomenis, atkuriami kadrai yra blogos kokybės, 22 pav. matome DSSIM funkciją nuo šifruojamų paketų dažnumo, funkcija yra beveik tiesinė kai praleidžiamų paketų kiekis neviršija 64.

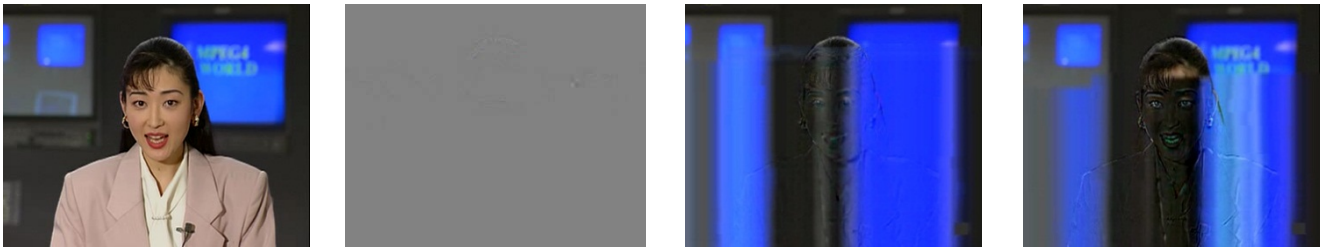
23 pav. pateiktas PSNR grafikas pagal 4 lentelės duomenis. PSNR funkcija, kaip ir DSSIM, išlieka beveik tiesi, nepriklausomai nuo šifruojamų duomenų kiekio. 24 pav. pateikti šifruoto ir originalaus vaizdo kadrai, naudojant skirtingus šifravimo parametrus. Praleidžiant daugiau kaip 64 paketus nešifruotus, gerai atstatoma daugiau nei pusė kadro.



22 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.



23 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.



Nešifruotas kadras | Šifruota 16B kiekvienam pakete | Kas 32 paketas, 176B pakete | Kas 64 paketas, 176B pakete

24 pav. Šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Eksperimento metu nustatyta, kad naudojant siūlomą metodą šifravimo laikas sutrumpėja nuo 4,5s iki 0,5s, ir iššifravimas užtrunka nuo 4,2s iki 0,5 atitinkamai. Lyginant šifruojamų paketų kiekį pastebime, kad šifruojant siūlomu metodu iš 46323 paketų užšifruoti reikia 1814 paketus, t.y apie 4% paketų.

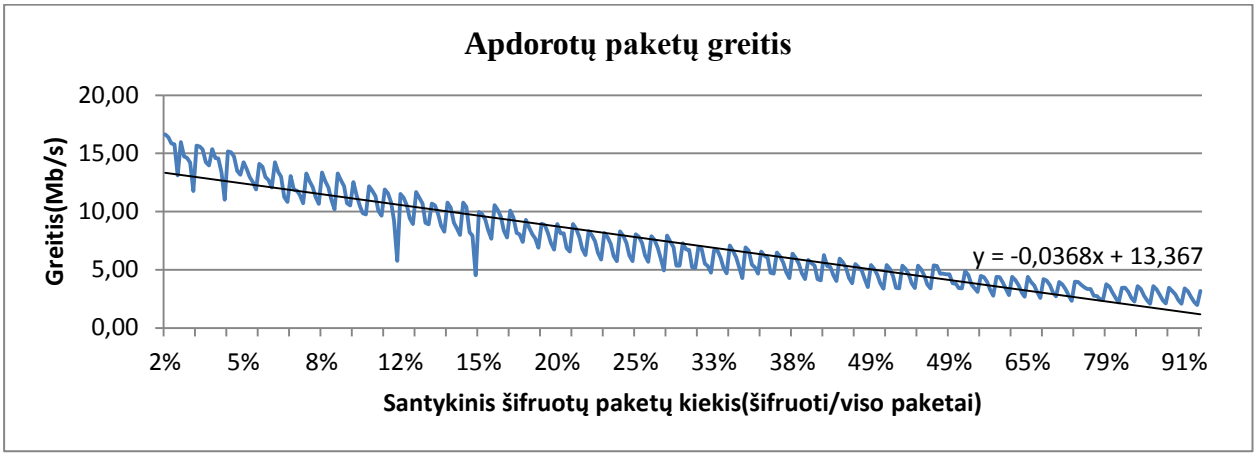
4.3.1.2 Foreman vaizdo įrašas

Foreman vaizdo įrašo eksperimento metu nustatyta, kad šifruojant kas 64 pakete paskutinius 16 baitų pasiekiamas geras saugumo lygis, ir vaizdas nebeatkuriamas. Šifruojant kas 64 paketo 16 baitų, užšifruojama apie 2% vaizdo srauto, šifravimo greitis padidėja 8,7 karto Šifruojant kas 32 paketą, 16 baitų pakete šifravimo greitis padidėja 8,5 karto.

5 lentelė. Foreman vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

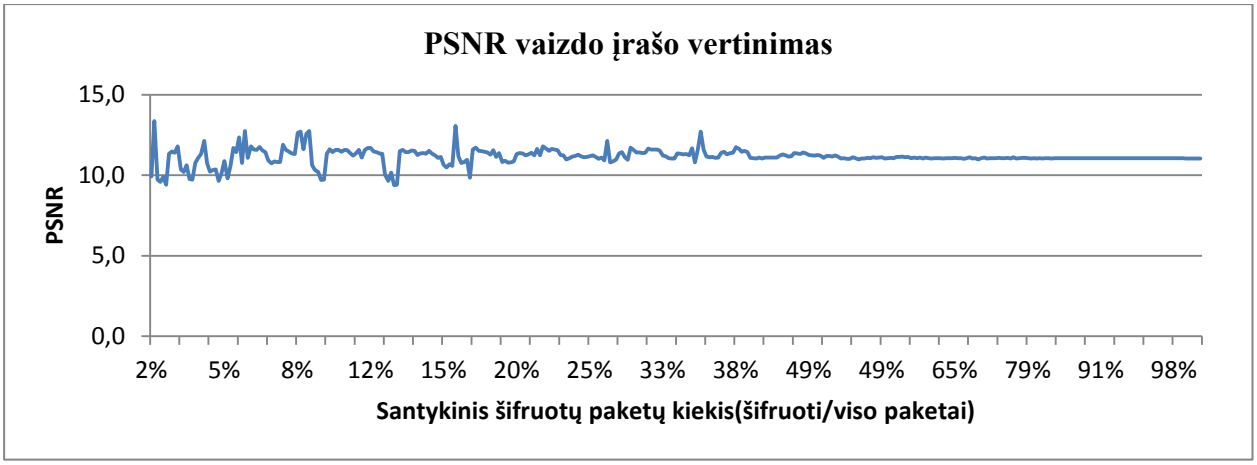
Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,98	1,45	11,1	1,11	3,11	1,56	11,1	1,11	2,89	2,41	11,0	1,13	**1,87**
3-1	0,25	0,56	11,2	1,22	8,04	0,58	11,2	1,28	7,82	0,79	11,1	1,14	5,68
3-2	0,39	0,81	11,1	1,13	5,58	0,77	11,1	1,15	5,86	1,10	11	1,12	4,09
3-9	0,74	1,13	11,1	1,23	3,98	1,30	11,1	1,13	3,47	1,85	11,1	1,11	2,44
3-12	0,79	1,20	11,1	1,12	3,76	1,27	11	1,13	3,55	2,06	11,1	1,11	2,19
6-1	0,14	0,42	11,5	1,42	10,7	0,43	11,3	1,41	10,3	0,57	11,1	1,22	7,93
6-2	0,25	0,55	11,3	1,32	8,16	0,58	11,2	1,20	7,79	0,79	11,2	1,15	5,73
6-9	0,59	1,03	11,1	1,19	4,37	1,03	11,1	1,17	4,40	1,61	11	1,12	2,81
6-12	0,65	1,03	11,1	1,14	4,38	1,14	11,1	1,12	3,96	1,74	11,1	1,12	2,59
12-1	0,08	0,34	11,9	1,63	13,2	0,36	11,6	1,63	12,6	0,42	11,3	1,46	10,6
12-2	0,14	0,42	11,5	1,54	10,7	0,44	11,2	1,42	10,3	0,56	11,3	1,19	7,99
12-9	0,42	0,76	11,2	1,27	5,96	0,80	11,3	1,19	5,66	1,17	11,2	1,18	3,85
12-12	0,49	0,83	11,3	1,24	5,41	0,89	11,2	1,17	5,07	1,34	11,2	1,14	3,38
24-1	0,04	0,31	11,3	1,61	14,5	0,29	10,8	1,56	15,3	0,34	12,1	1,38	13,3
24-2	0,08	0,35	10,9	1,44	13,0	0,38	10,9	1,47	11,8	0,42	10,8	1,34	10,7
24-9	0,27	0,57	11,4	1,37	7,94	0,61	11,4	1,27	7,43	0,85	11,7	1,20	5,34
24-12	0,33	0,65	11,7	1,27	6,93	0,68	10,8	1,49	6,64	0,97	11,6	1,27	4,67
32-1	0,03	0,28	9,4	1,72	**15,9**	0,31	11,3	1,63	14,7	0,38	11,8	1,45	11,7

32-2	0,06	0,32	12,3	1,58	14,0	0,33	10,8	1,54	13,8	0,37	11,8	1,56	12,0
32-9	0,22	0,51	11,4	1,40	8,93	0,53	11,2	1,35	8,53	0,72	11,8	1,32	6,26
32-12	0,27	0,57	10,9	1,34	7,89	0,60	12,1	1,27	7,55	0,91	11	1,35	4,94
64-1	0,02	0,28	13,4	1,11	**16,3**	0,27	9,93	1,61	16,6	0,28	9,72	1,29	15,8
64-2	0,03	0,29	10,4	1,46	15,6	0,29	10,6	1,39	15,3	0,32	9,76	1,28	14,2
64-9	0,12	0,39	10	1,59	11,6	0,40	9,64	1,53	11,1	0,51	9,4	1,43	8,90
64-12	0,16	0,43	11,2	1,20	10,5	0,44	10,8	1,18	10,1	0,58	9,85	1,22	7,77



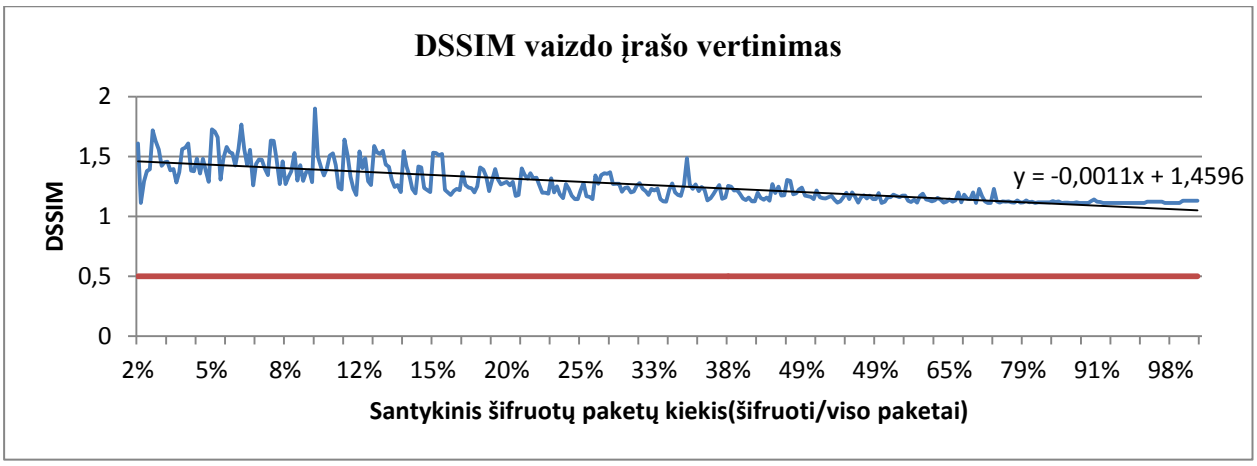
25 pav. Foreman vaizdo įrašo apdorojamų duomenų greitis.

25 pav. Pateikta foreman vaizdo įrašo greičio priklausomybė nuo šifruojamų duomenų kiekio, šifruojant 2% duomenų greitaveika padidėja 8,7 karto, nuo 1,87s iki 15,9s tam pačiam duomenų kiekiui užšifruoti.

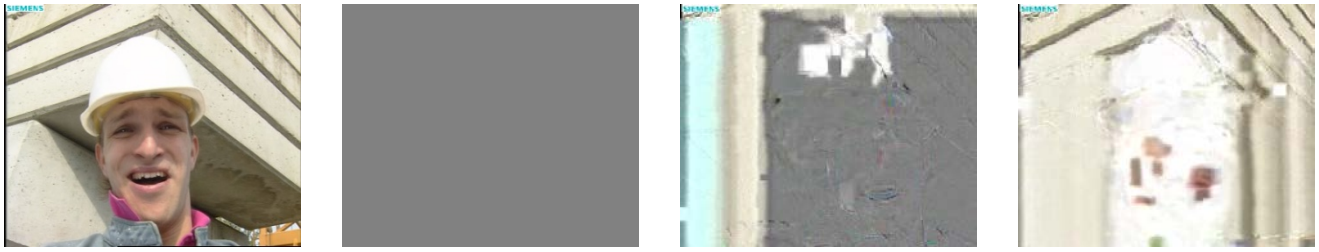


26 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.

Vertinant foreman vaizdo įrašą PSNR metodu, visų bandymų metu PSNR reikšmė atitiko keliamus saugumo reikalavimus, t.y neviršijo 15 dB. Vertinimas DSSIM metodu patvirtina PSNR matavimų išvadas.



27 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.



Nešifruotas kadras | Šifruota 16B kiekvienam pakete | Kas 32 paketas, 176B pakete | Kas 64 paketas, 176B pakete

28 pav. Foreman šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Eksperimento metu nustatyta, kad Foreman vaizdo įrašo apsaugai užšifruoti pakanka mažą dalį duomenų, vaizdas nėra atstatomas, 26 pav., 27 pav. matyti, kad PSNR kokybės reikšmės nevišija 15, o DSSIM ne mažiau nei 0,5. 28 pav. pateikti dekoduoti vaizdo kadrai patvirtina vaizdo neatkuriamumą.

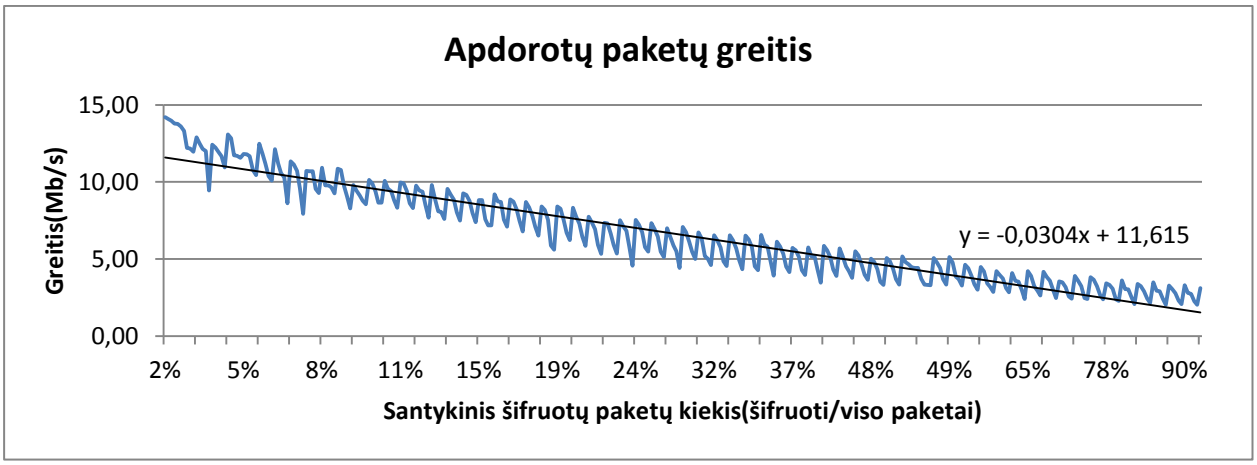
4.3.1.3 Container vaizdo įrašas

Container vaizdo įrašas pasižymi dinaminiu fonu(raibuliuojantis vanduo) ir lėtai judančių laivu su daug smulkių dalių. 6 lentelėje pateikti eksperimentų rezultatai.

6 lentelė. Container vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

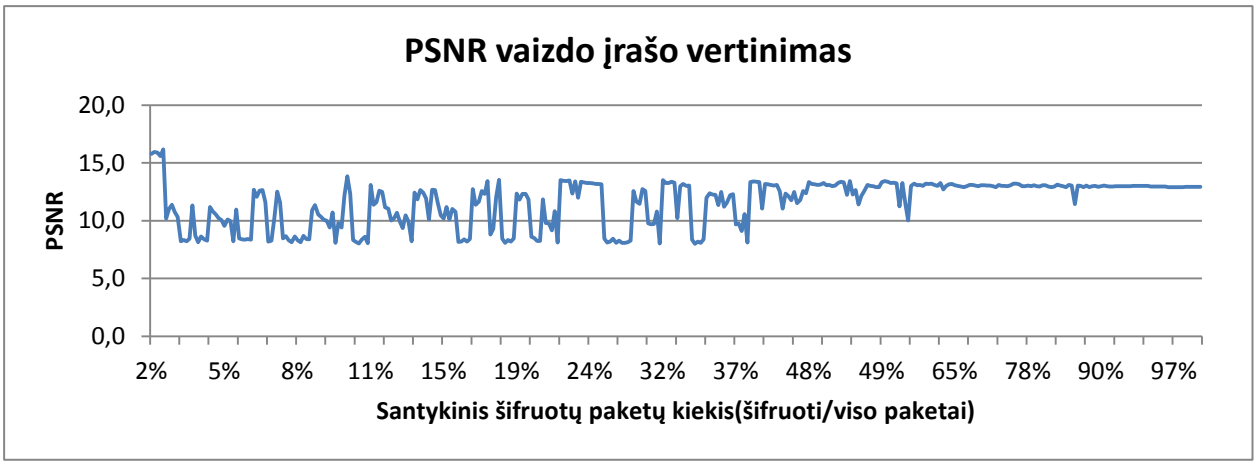
Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,97	0,87	13,0	1,00	3,09	0,93	13,0	1,00	2,87	1,45	12,9	1,00	**1,85**
3-1	0,24	0,36	13,3	1,03	7,54	0,37	13,2	1,03	7,23	0,49	13,2	1,02	5,47
3-2	0,39	0,46	13,2	1,06	5,85	0,48	13,1	1,03	5,57	0,69	13,1	1,02	3,91

3-9	0,73	0,70	13,2	1,03	3,82	0,73	13,2	1,05	3,65	1,12	13	1,01	2,39
3-12	0,78	0,81	13	1,00	3,31	0,78	13	1,01	3,43	1,18	13	0,99	2,27
6-1	0,14	0,29	12,7	1,08	9,17	0,29	10,1	1,10	9,27	0,36	10,5	1,06	7,40
6-2	0,24	0,36	13,4	1,04	7,52	0,38	12	1,12	7,13	0,59	13,3	1,04	4,57
6-9	0,58	0,68	13,2	1,03	3,96	0,63	13,2	1,03	4,22	0,95	13	1,00	2,83
6-12	0,65	0,66	13,3	1,03	4,08	0,75	12,7	1,03	3,59	1,11	13,2	1,03	2,41
12-1	0,07	0,24	8,18	1,39	11,3	0,25	10,2	1,21	10,7	0,34	11,6	1,31	7,93
12-2	0,14	0,28	12,4	1,17	9,55	0,29	11,8	1,11	9,18	0,36	11,9	1,05	7,49
12-9	0,42	0,47	12,5	1,05	5,70	0,52	11	1,09	5,14	0,71	11,8	1,05	3,78
12-12	0,48	0,60	12,2	1,07	4,45	0,56	13,4	1,02	4,80	0,80	11,4	1,10	3,33
24-1	0,04	0,22	8,64	1,25	**12**	0,22	8,15	1,18	12,3	0,24	8,28	1,10	10,9
24-2	0,07	0,25	8,48	1,21	10,7	0,25	8,29	1,23	10,7	0,28	8,14	1,11	9,55
24-9	0,26	0,37	8,44	1,29	7,33	0,39	8,11	1,14	6,94	0,52	8,09	1,09	5,13
24-12	0,32	0,41	8,35	1,22	6,53	0,43	8	1,12	6,26	0,59	8,09	1,08	4,52
32-1	0,03	0,21	8,21	1,09	**12,9**	0,28	11,3	1,11	9,45	0,22	8,45	1,16	**12**
32-2	0,06	0,23	8,39	1,17	11,8	0,21	8,47	1,21	12,5	0,27	8,35	1,11	10,1
32-9	0,21	0,32	8,61	1,29	8,33	0,35	8,49	1,21	7,73	0,46	11,8	4,04	5,85
32-12	0,26	0,46	8,09	1,08	5,87	0,38	8,28	1,16	6,99	0,61	8,28	1,12	4,43
64-1	0,02	0,19	15,8	0,70	14,2	0,19	16,2	0,75	13,8	0,19	15,9	0,70	14
64-2	0,03	0,20	11	0,87	13,3	0,22	11,4	0,91	12,2	0,22	10,7	0,94	12,2
64-9	0,12	0,28	11	0,97	9,45	0,27	11,2	0,98	9,76	0,35	10,7	0,92	7,68
64-12	0,15	0,37	11	0,86	7,18	0,30	11,2	1,00	8,82	0,37	10,8	0,86	7,17

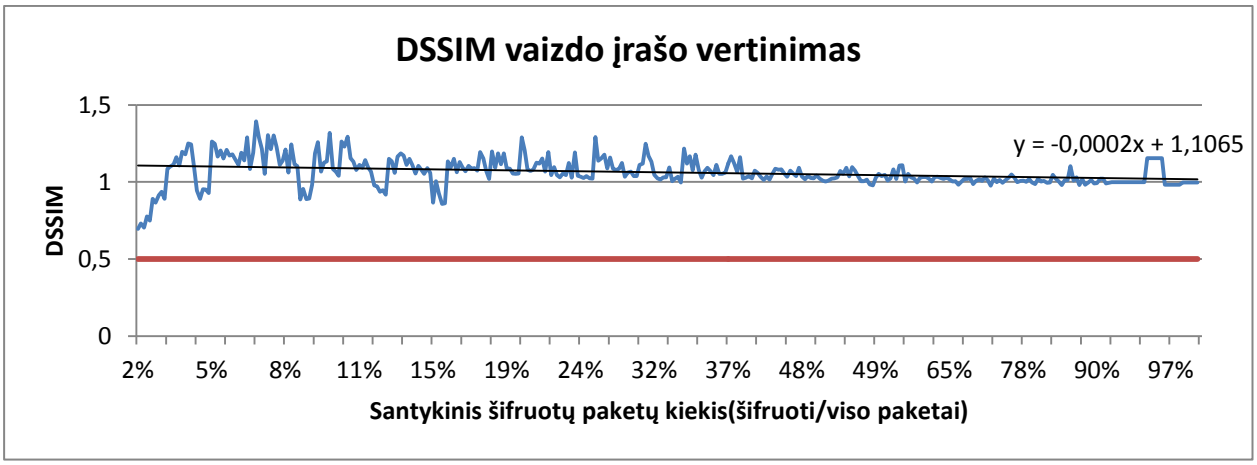


29 pav. Container vaizdo įrašo apdorojamų duomenų greitis.

Pagal 6 lentelėje esamus eksperimento rezultatus sudarytas apdorotų paketų greitaveikos grafikas pateiktas 29 pav. Naudojant pilną duomenų šifravimą įrašo šifravimo greitaveika 1,85Mb/s, naudojant dalinį MPEG-2 srauto šifravimo metodą greitaveika padidėja 7 kartus, iki 12,9Mb/s. Šifruojant kas 32 paketą užšifruojama 3% duomenų.



30 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.



31 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.

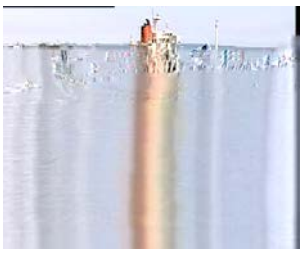
30 pav. Container kokybės vertinimo grafike pateiktas vaizdo įrašo užšifravimo patikimumas, apsauga užtikrinama šifruojant kas 32 paketą, 16 ar daugiau baitų pakete, tai sudaro 3% šifruotų paketų. Šifurojant kas 64 paketą PSNR reikšmės viršija 15 dB, taip pat 31 pav. pateikto DSSIM vertinimo veikšė yra apie 0,7.

32 pav. pateikti kadrai rodo, kad šifruojant ir kas 64 paketą didelė dalis kadro įra blogos kokybės, šio kadro PSNR yra 15,9 dB ir DSSIM vertė apie 0,7.

Nešifruotas kadras

Šifruota 16B kiekvienam pakete

Kas 32 paketas, 176B pakete



Kas 64 paketas, 176B pakete

32 pav. Container šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Analizuojant 6 lentelės eksperimentų rezultatus nustatyta, kad container vaizdo įrašas tinkamai apsaugas šifruojant kas 32 kadrą 16 ir daugiau baitų pakete, tai sudaro 3% šifruojamų paketų, greitaveika padidėja 7 kartus nuo 1,85 Mb/s iki 12,9 Mb/s.

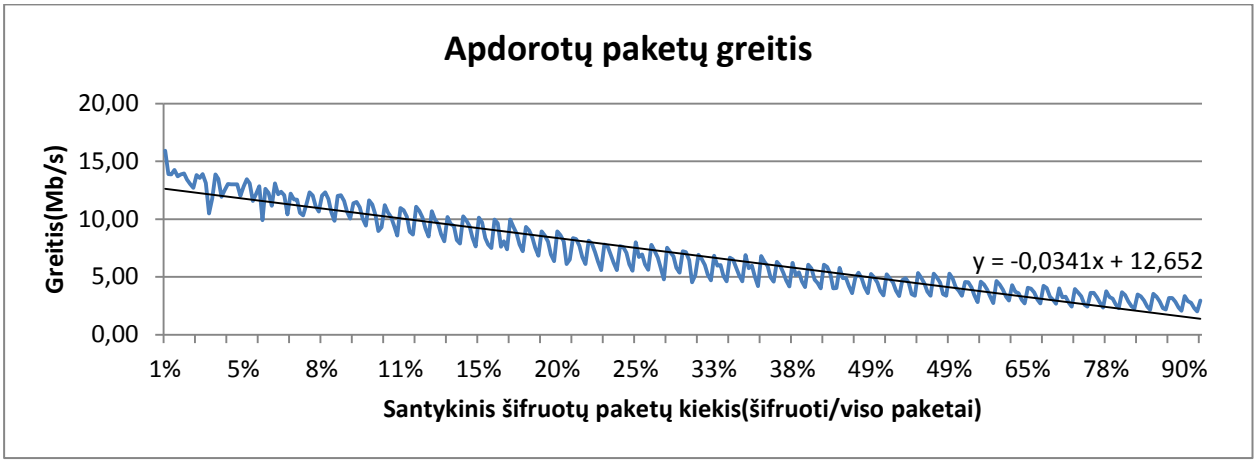
4.3.1.4 Hall monitor vaizdo įrašas

Hall monitor vaizdo įrašo foną sudaro biuro vaizsas, su keliais smulkesniais fragmentais, dinaminė dalis apima vaikštančius žmones.

7 lentelė. Hall monitor vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,98	1,19	13,2	1,15	3,07	1,27	13,2	1,11	2,88	1,96	13,2	1,10	**1,86**
3-1	0,24	0,48	13,1	1,31	7,65	0,48	13,1	1,30	7,57	0,66	13,2	1,19	5,52
3-2	0,39	0,60	13,2	1,22	6,06	0,64	13	1,20	5,72	0,91	13,2	1,12	4,00
3-9	0,73	0,91	13,2	1,29	4,00	1,13	13,2	1,14	3,22	1,50	13,2	1,08	2,44
3-12	0,78	0,97	13,1	1,17	3,76	1,12	13,2	1,16	3,25	1,59	13,2	1,09	2,30
6-1	0,14	0,36	13,1	1,60	10,2	0,37	13,1	1,49	9,87	0,48	13,2	1,38	7,65
6-2	0,25	0,46	13,1	1,49	8,01	0,54	13,1	1,22	6,73	0,65	13,1	1,22	5,61
6-9	0,59	0,78	13,3	1,20	4,65	0,84	13,1	1,19	4,33	1,24	13,2	1,15	2,94
6-12	0,65	0,90	13,1	1,22	4,07	0,92	13,2	1,37	3,98	1,35	13,2	1,10	2,69
12-1	0,08	0,30	13,1	1,65	12,2	0,31	12	2,28	11,7	0,35	13,1	1,22	10,3
12-2	0,14	0,36	13	1,59	10,2	0,38	13,1	1,48	9,58	0,46	13,3	1,20	7,88
12-9	0,42	0,63	13,1	1,18	5,77	0,75	13,1	1,22	4,90	1,02	13,2	1,15	3,58
12-12	0,49	0,69	13,2	1,27	5,28	0,74	13,2	1,21	4,94	1,08	13,2	1,16	3,38

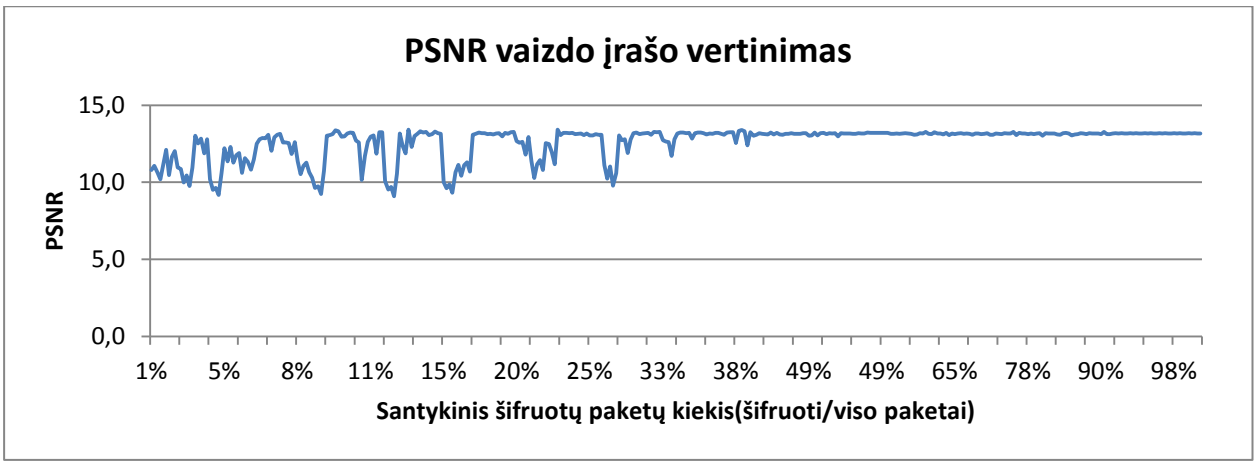
24-1	0,04	0,31	13	2,10	11,8	0,26	12,5	2,16	13,9	0,29	12,8	1,64	**12,5**
24-2	0,08	0,32	12,6	1,86	11,3	0,30	12,6	1,84	12,3	0,34	12,6	1,34	10,7
24-9	0,27	0,49	13	1,25	7,51	0,51	12,7	1,30	7,13	0,68	12,8	1,13	5,38
24-12	0,33	0,53	12,7	1,64	6,83	0,61	12,6	1,41	5,98	0,79	12,8	1,20	4,61
32-1	0,03	0,26	12,1	1,50	**13,9**	0,26	10,5	1,69	14	0,29	11	1,36	**12,7**
32-2	0,06	0,28	11,9	2,81	12,8	0,37	10,6	1,95	9,91	0,33	10,8	1,40	11,2
32-9	0,21	0,44	11,4	1,38	8,35	0,44	10,3	1,49	8,27	0,60	10,8	1,33	6,12
32-12	0,27	0,47	11,1	1,67	7,79	0,50	10,2	1,57	7,26	0,76	10,6	1,36	4,78
64-1	0,01	0,23	10,8	1,25	**15,9**	0,26	11,1	1,49	13,9	0,27	11,2	0,96	**13,7**
64-2	0,03	0,26	10,9	0,99	13,8	0,27	9,99	1,12	13,6	0,35	11	0,96	10,5
64-9	0,12	0,33	10,1	1,18	11,1	0,34	9,53	1,31	10,7	0,43	10,5	1,15	8,49
64-12	0,15	0,36	10	1,27	10,1	0,38	9,61	1,34	9,73	0,49	10,7	1,19	7,51



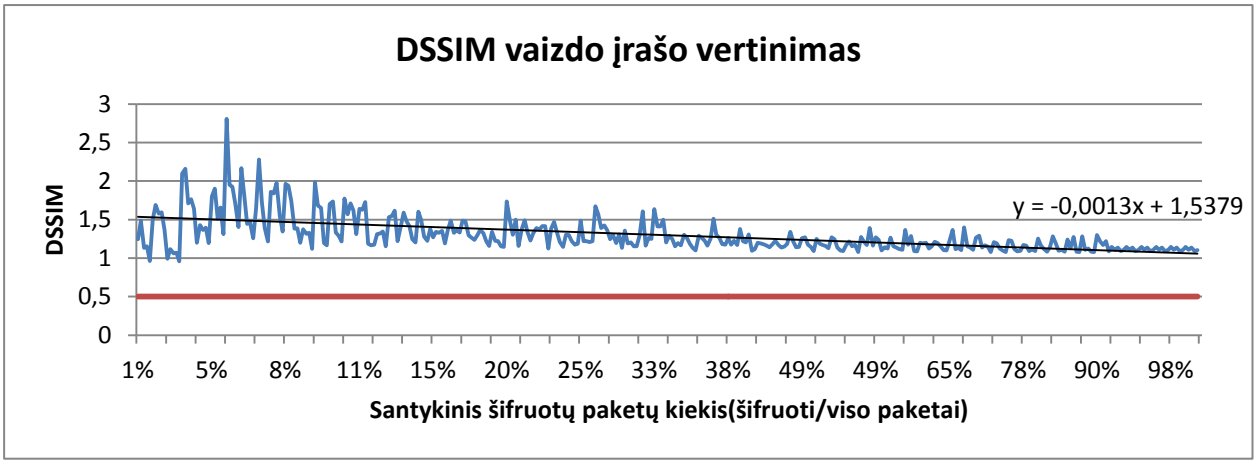
33 pav. Container vaizdo įrašo apdorojamų duomenų greitis.

Naudojant dalinį MPEG-2 srauto šifravimo metodą, Hall monitor vaizdo įrašo paketų praleidumo greitaveika išaugo 8,5 karto, nuo 1,87Mb/s iki 15,9Mb/s(šifruojant kas 64 paketo 16 baitų). Šifruojant pilnai kas 64 paketą pasiekiamas 13,7Mb/s greitis. Šifruojant kas 64 paketą užšifruojama tik 1% duomenų. Atlikus eksperimentą nustatyta, kad šifruojant kas 64-tą paketą vis dar pasiekiamas geras apsaugos lygis, tai rodo PSNR ir DSSIM reikšmės 7 lentelėje,

- Šifruojant kas 64-tą paketą 16 baitų pakete – PSNR – 10,8 ir DSSIM – 1,25
- Šifruojant kas 64-tą paketą 176 baitus pakete – PSNR – 11,2 ir DSSIM – 0,96.



34 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.



35 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.

Vertinant gautus Hall monitor vaizdo įrašo eksperimento rezultatus 34 pav. ir 35 pav. galima teigti kad įrašas apsaugotas tinkamai šifruojant kas 64 paketą bent 16 baitų pakete, šifruojant šiuo žingsniu PSNR yra 10,6 dB o DSSIM 1,25.

Nešifruotas kadras

Šifruota 16B kiekvienam pakete

Kas 32 paketas, 176B pakete

Kas 64 paketas, 176B pakete

36 pav. Hall monitor šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Hall monitor vaizdo įrašui taikant dalinį MPEG-2 srauto šifravimą pasiekiamas geras apsaugos lygis šifruojant kas 64-to paketo bent 16 baitų, tai sudaro 1% šifruojamų paketų, ir yra 8,5 karto sparčiau nei šifruojant visą srautą.

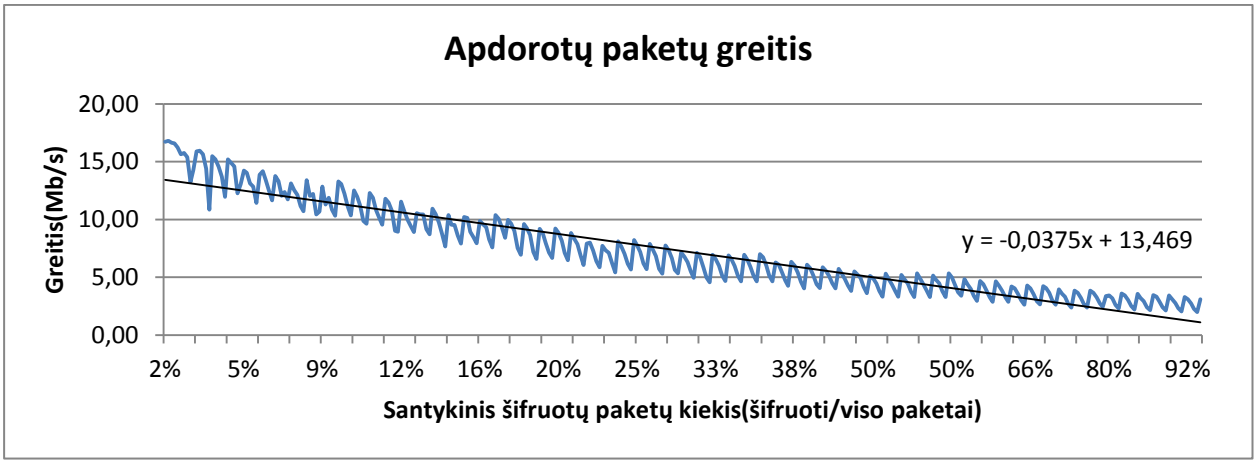
4.3.1.5 Mobile vaizdo įrašas

Atliekant Mobile vaizdo įrašo apsaugos vertinimo kokybę šifruojant daliniu MPEG-2 apsaugos metodu, pastebėta, kad šifruojant net ir labai retai ir mažai duomenų pakete, užtikrinama aukšta apsauga, kadrai visiškai nebeatkuriami.

8 lentelė. Mobile vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

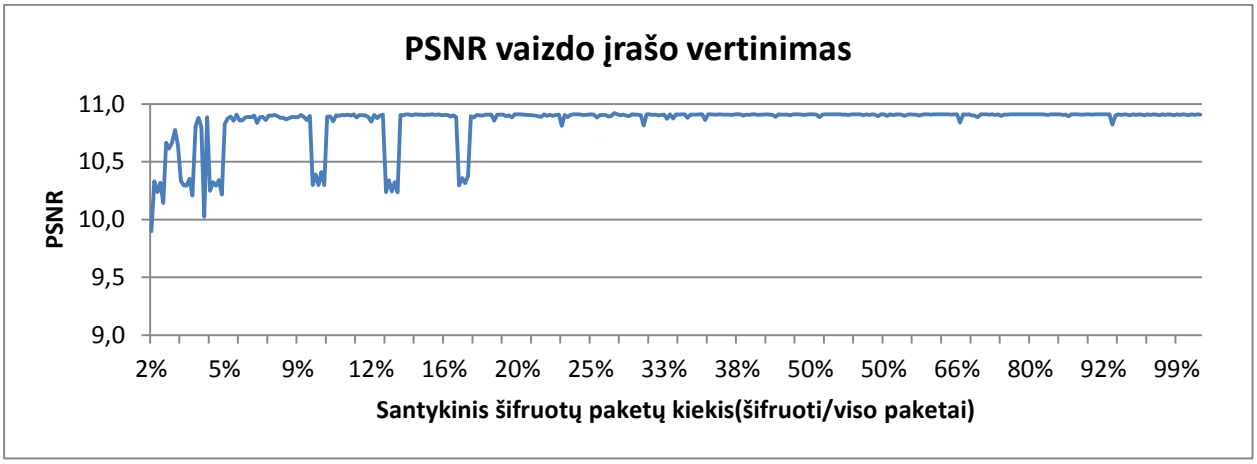
Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,99	4,01	10,9	7,31	3,11	4,37	10,9	7,29	2,85	6,73	10,9	7,30	**1,86**
3-1	0,25	1,52	10,9	7,41	8,21	1,60	10,9	7,29	7,78	2,19	10,9	7,42	5,71
3-2	0,40	2,05	10,9	7,31	6,08	2,16	10,9	7,29	5,78	3,08	10,9	7,31	4,05
3-9	0,75	3,24	10,9	7,29	3,85	3,41	10,9	7,32	3,66	5,25	10,9	7,30	2,38
3-12	0,80	3,72	10,9	7,29	3,36	3,66	10,9	7,29	3,41	5,46	10,9	7,29	2,29
6-1	0,14	1,20	10,9	7,29	10,4	1,31	10,9	7,39	9,55	1,58	10,9	7,37	7,92
6-2	0,25	1,54	10,9	7,29	8,10	1,63	10,9	7,29	7,66	2,20	10,9	7,33	5,66
6-9	0,60	2,68	10,9	7,29	4,66	2,93	10,9	7,29	4,26	4,33	10,9	7,43	2,88
6-12	0,66	2,95	10,9	7,29	4,23	3,08	10,9	7,34	4,05	4,75	10,9	7,42	2,63
12-1	0,08	0,95	10,9	7,58	13,1	0,99	10,9	7,54	12,6	1,17	10,9	7,89	10,7
12-2	0,14	1,22	10,9	7,40	10,2	1,23	10,9	7,29	10,1	1,57	10,9	7,57	7,94
12-9	0,43	2,19	10,9	7,30	5,71	2,34	10,9	7,41	5,34	3,27	10,9	7,61	3,82
12-12	0,50	2,41	10,9	7,29	5,19	2,60	10,9	7,29	4,81	3,81	10,9	7,61	3,27
24-1	0,04	0,81	10,8	8,07	**15,5**	0,82	10,9	8,08	15,2	1,04	10,9	7,70	12
24-2	0,08	0,93	10,9	7,72	13,4	1,04	10,9	8,01	12,1	1,17	10,9	7,73	10,7
24-9	0,27	1,61	10,9	7,29	7,75	1,70	10,9	7,57	7,33	2,34	10,9	7,65	5,34
24-12	0,33	1,80	10,9	7,37	6,95	1,90	10,9	7,29	6,56	2,68	10,9	7,78	4,67
32-1	0,03	0,80	10,7	7,79	**15,7**	0,79	10,6	7,55	15,8	0,87	10,6	7,22	**14,3**
32-2	0,06	0,90	10,9	7,33	13,9	0,88	10,9	7,42	14,2	1,07	10,9	7,31	11,7
32-9	0,22	1,41	10,9	7,46	8,82	1,50	10,9	7,59	8,34	2,06	10,9	7,68	6,06
32-12	0,27	1,58	10,9	7,46	7,89	1,70	10,9	7,43	7,36	2,35	10,9	7,48	5,32
64-1	0,02	0,75	9,9	8,99	**16,7**	0,74	10,3	8,52	16,8	0,77	10,1	8,91	**16,2**
64-2	0,03	0,79	10,3	7,48	15,8	0,78	10,3	7,86	15,9	1,15	10,2	7,30	10,8

64-9	0,12	1,18	10,2	7,15	10,5	1,20	10,3	7,05	10,4	1,43	10,2	6,99	8,72
64-12	0,16	1,20	10,3	6,81	10,3	1,24	10,4	6,91	10,0	1,63	10,3	6,71	7,68



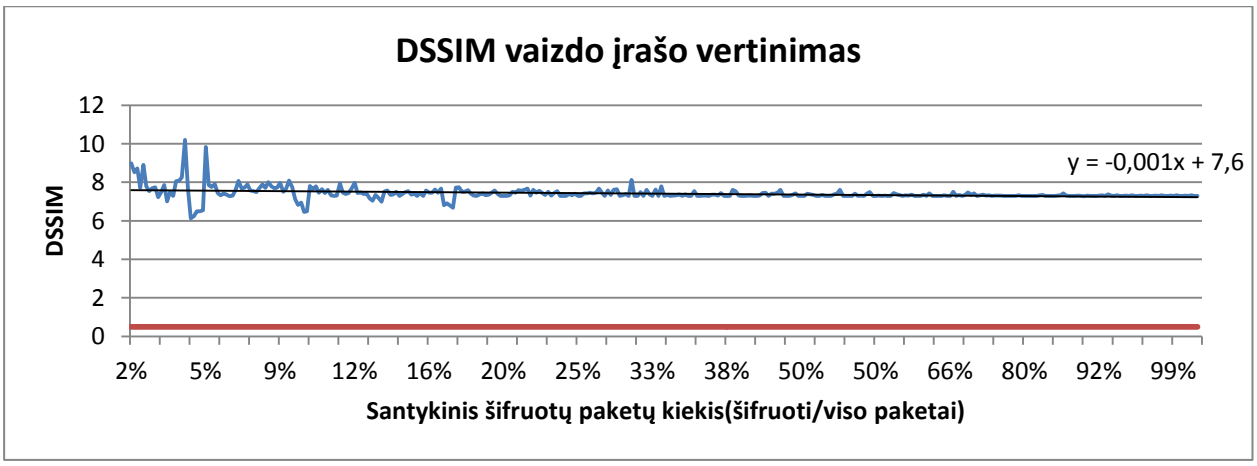
37 pav. Mobile vaizdo įrašo apdorojamų duomenų greitis.

Mobile eksperimento metu nustatytas didžiausias greičio pasikeitimas naudojant siūloma dalinį MPEG-2 šifravimą lyginant su pilnu šifravimu. Šifravimo greitaveika padidėja 9 kartus, nuo 1,86Mb/s iki 16,7Mb/s šifruojant kas 64 paketą, 16 baitų pakete. Taikant dalinio šifravimo metodą vaizdo įrašas išlieka apsaugotas, PSNR ir DSSIM reikšmės yra atitinkamai 9,9dB ir 8,99. 37 pav. greitaveika tolygiai didėja mažinant šifruojamų paketų kiekį.

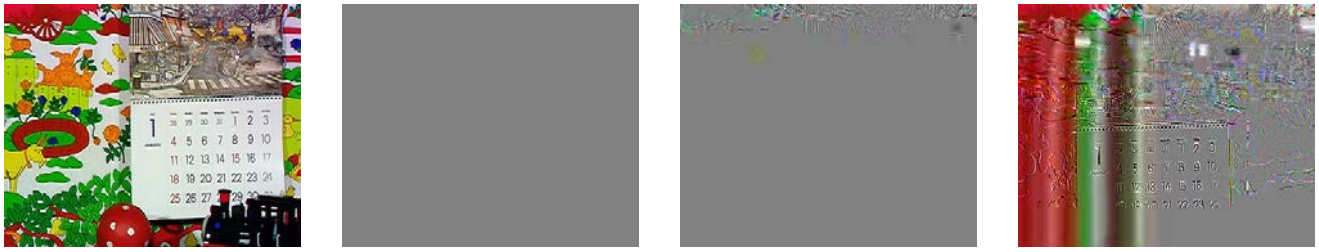


38 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.

38 pav. ir 39 pav. pateikti vaizdo kokybės vertinimo grafikai. PSNR kreivė išlieka pastovi iki 20% šifruojamų duomenų kiekio. Visų matavimų kokybės reikšmės PSNR metodu išlieka žemos, mažiau už 15, vertinant DSSIM metodu reikšmės yra labai aukštos, daugiau už 0,5. 40 pav. pateikti vaizdo kadrai patvirtina gerą apsaugos lygį.



39 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.



Nešifruotas kadras | Šifruota 16B kiekvienam pakete | Kas 32 paketas, 176B pakete | Kas 64 paketas, 176B pakete

40 pav. Mobile šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Taikant dalinio MPEG-2 šifravimo metodą, mobile vaizdo įrašo šifravimo greitaveika padidėja 9 kartus lyginant su pilnu šifravimu, nuo 1,86Mb/s iki 16,7Mb/s. Šifruojant kas 64 paketo 16 baitų užtikrinamas geras apsaugos lygis, vaizdo įrašo PSNR yra apie 9,9dB o DSSIM vertė 8,99.

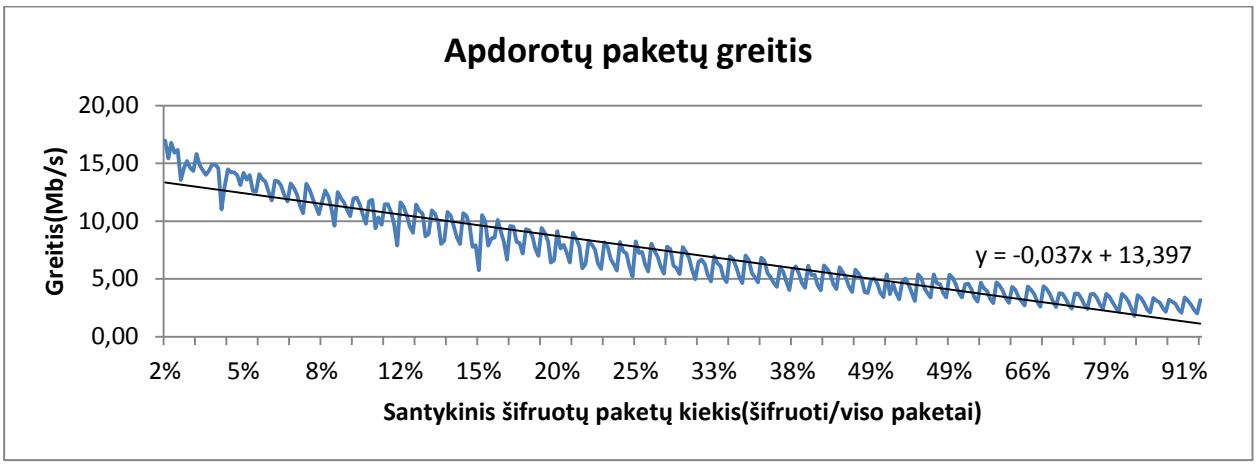
4.3.1.6 Waterfall vaizdo įrašas

Waterfall vaizdo įrašas pasižymi didele spalvų gausa, dideliu mažų objektų kiekiu ir krintančiu vandeniu. Tai įtakoja vaizdo kodavimo algoritmų suspaudimo lygį, atsirandančių I kadrų blokų kiekį. Dėl atsirandančio didelio kiekio I blokų, gali tekti šifruoti didesnę dallį vaizdo įrašo.

9 lentelė. Waterfall vaizdo įrašo šifravimo greitaveika ir šifravimo patikimumo vertinimas.

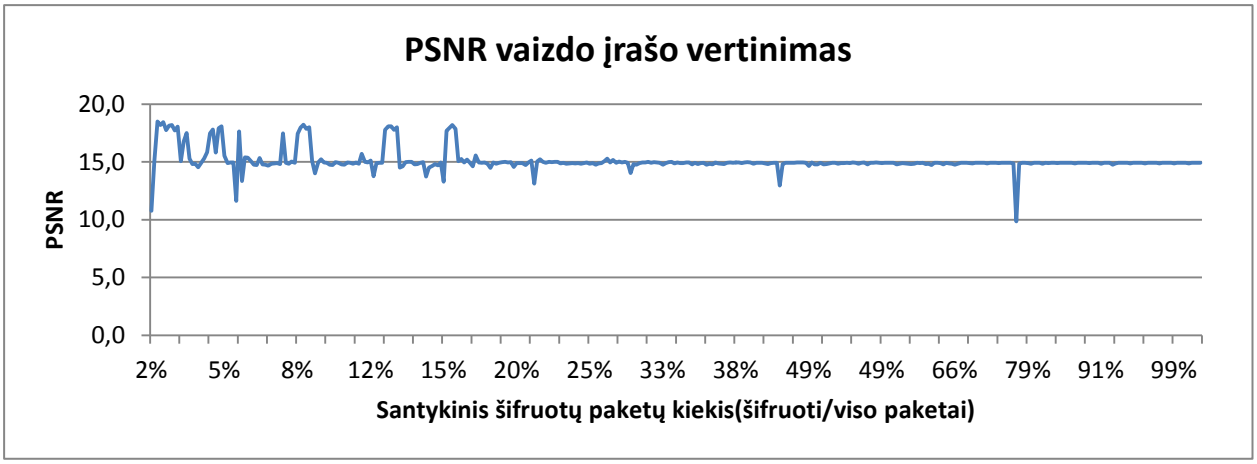
Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,99	2,53	14,9	2,99	1,83	1,66	11,4	3,03	2,80	2,53	11,4	2,99	**1,83**
3-1	0,25	0,82	14,9	3,06	5,64	0,64	11,3	3,25	7,25	0,82	11,4	3,06	5,64

3-2	0,39	1,16	14,9	3,03	4,00	0,87	12,2	3,19	5,35	1,16	12,1	3,03	4,00
3-9	0,74	1,91	14,9	3,02	2,42	1,24	11,3	3,10	3,72	1,91	11,4	3,02	2,42
3-12	0,79	2,14	14,9	3,04	2,16	1,34	11,3	3,12	3,45	2,14	11,4	3,04	2,16
6-1	0,14	0,59	14,9	3,04	7,90	0,44	11,7	3,27	10,4	0,59	10,7	3,04	7,90
6-2	0,25	0,89	14,9	3,03	5,21	0,63	11,1	3,28	7,40	0,89	11,4	3,03	5,21
6-9	0,59	1,59	14,9	3,07	2,91	1,11	10,9	3,22	4,16	1,59	11,6	3,07	2,91
6-12	0,66	1,72	14,8	3,09	2,69	1,14	10,8	3,22	4,06	1,72	10,8	3,09	2,69
12-1	0,08	0,43	14,8	3,24	10,6	0,36	12,2	3,19	12,8	0,43	12,3	3,24	10,6
12-2	0,14	0,58	13,7	3,27	8,02	0,44	9,9	3,02	10,4	0,58	11	3,27	8,02
12-9	0,42	1,19	14,9	3,03	3,88	0,83	7,32	3,07	5,55	1,19	10,9	3,03	3,88
12-12	0,49	1,36	14,9	3,04	3,40	0,91	9,92	3,12	5,10	1,36	11	3,04	3,40
24-1	0,04	0,35	15,8	2,42	13,0	0,31	13	3,18	14,8	0,35	12,3	2,42	13,0
24-2	0,08	0,44	14,9	2,35	10,6	0,36	12,4	2,83	12,7	0,44	13,9	2,35	10,6
24-9	0,27	0,85	14	2,31	5,43	0,61	12,3	3,10	7,59	0,85	12,4	15,3	5,43
24-12	0,33	0,98	14,9	2,72	4,72	0,73	12,3	3,08	6,37	0,98	12,4	2,72	4,72
32-1	0,03	0,32	14,8	2,44	**14,3**	0,31	12,9	2,28	14,8	0,32	11,9	2,44	**14,3**
32-2	0,06	0,39	15,1	2,21	11,7	0,34	13	2,33	13,6	0,39	11,9	2,22	11,7
32-9	0,22	0,74	15	2,29	6,27	0,55	13,1	3,23	8,46	0,74	12,1	2,29	6,27
32-12	0,27	0,85	14,9	2,44	5,45	0,62	13	2,33	7,42	0,85	11,9	2,44	5,45
64-1	0,02	0,29	18,5	1,30	16,1	0,30	12,9	3,49	15,4	0,29	13,3	0,25	16,1
64-2	0,03	0,32	18,1	1,30	14,3	0,32	12,9	1,57	14,6	0,32	13,4	0,29	14,3
64-9	0,12	0,52	18	1,51	8,89	0,42	12,9	1,56	10,9	0,52	13,3	0,34	8,89
64-12	0,16	0,60	17,9	1,59	7,69	0,46	13	1,61	9,98	0,60	13,3	0,34	7,69

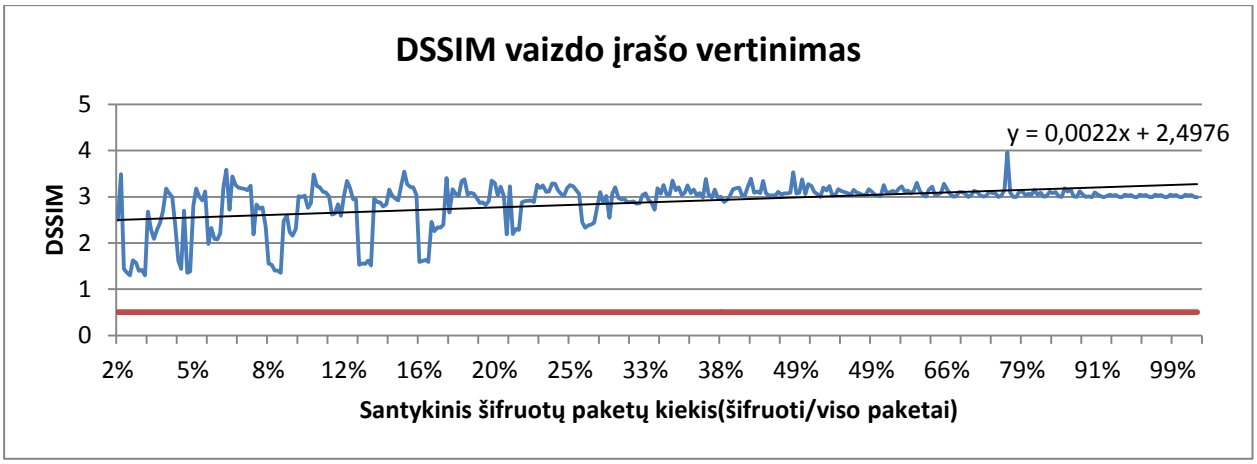


41 pav. Container vaizdo įrašo apdorojamų duomenų greitis.

Waterfall vaizdo įrašo apsaugos greitaveika išauga 7,8 karto lyginant su pilnu šifravimu, pilnai šifruojant kas 32 paketą. Šifruojant kas 32 paketą greitis buvo vienodas, tiek šifruojant pilnai paketą, tiek šifruojant tik 16 baitų (9 lentelė).



42 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.



43 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.

42 pav. pateikta PSNR vertinimo kokybė šifruotam vaizdui siūlomu daliniu šifravimo metodu, visi rezultatai svyruoja apie 15db, pasiekus 17% šifurojamų duomenų vertės išauga virš 15dB, tai rodo nepakankamą apsaugos lygį. 43 pav. pateiktas DSSIM vaizdo vertinimo grafike visos reikšmės išsidėsčiusios virš 0,5, tai reikštų pakankamą apsaugos lygį. 44 pav. pateikti vaizdo kadrai taip pat liudija apie gerą apsaugos lygį watterfall vaizdo įrašą apsaugant daliniu MPEG-2 srauto šifravimo algoritmu.

Nešifruotas kadras

Šifruota 16B kiekvienam pakete

Kas 32 paketas, 176B pakete

Kas 64 paketas, 176B pakete

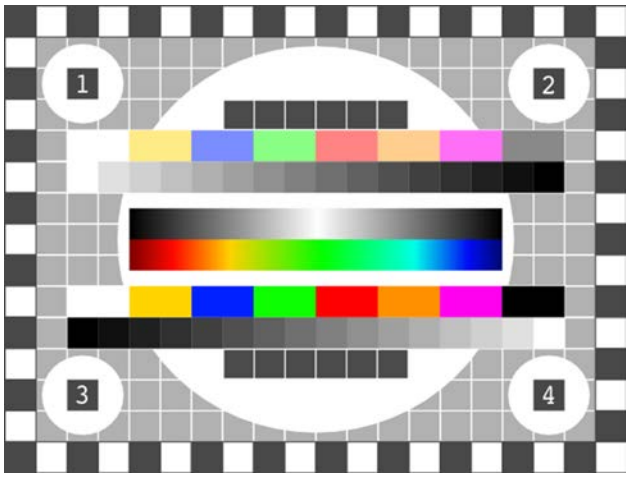
44 pav. Waterfall šifruoto vaizdo kadrai užšifruoti naudojant skirtingus algoritmo parametrus.

Waterfall vaizdo įrašas tinkamai apsaugomas pilnai šifruojant kas 32 paketą, naudojant dalinį MPEG-2 apsaugos algoritmą. Paketų apdorojimo greitaveika padidėja 7,8 karto, užšifruojant 3% paketų. Šifruojant 3% paketų vaizdas nebeatkuriamas, PSNR vertinimas rodo kad kadrai turėtų būti bent dalinai matomi, tačiau DSSIM ir kadrų peržiūrėjimas patvirtina, kad apsauga tinkama.

4.3.2 Statinio vaizdo vertinimas

Dėl skirtingo suspaudimo laipsnio koduojant h.264/AVC formatu, atlikta eksperimentai ir su labai statiniais vaizdais. Transliacijos srautas sugeneruotas panaudojus vieną sintetinį paveikslėlį(45 pav.).

Spaudžiant statinį vaizdą, sraute yra daug P ir B tipo makro blokų, jų apimtis nėra didelė, todėl gali nepakakti šifravimo intervalo dažnumo.

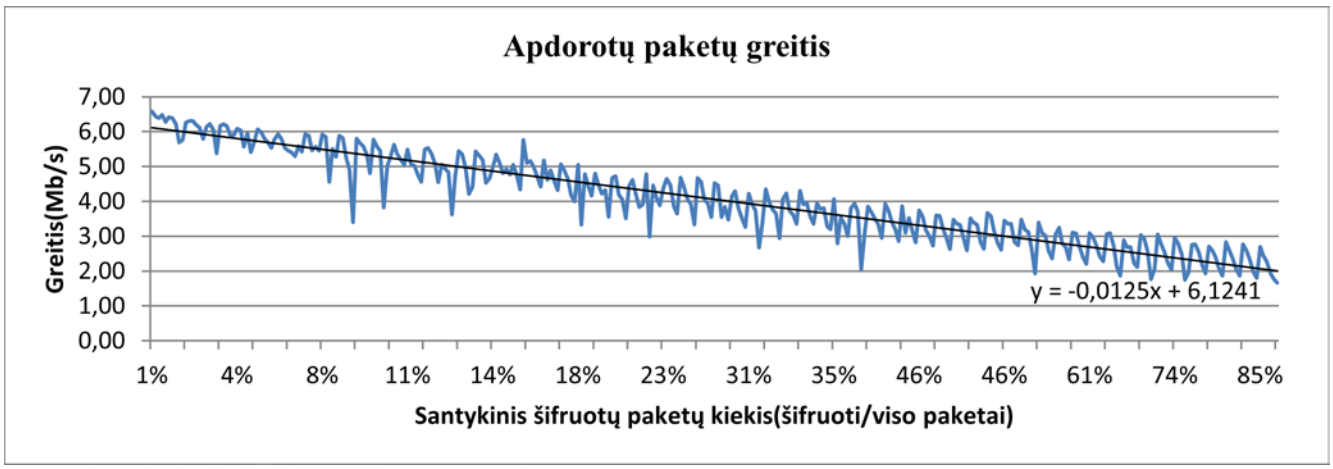


45 pav. Statinio vaizdo transliacijai generuoti naudotas paveikslas

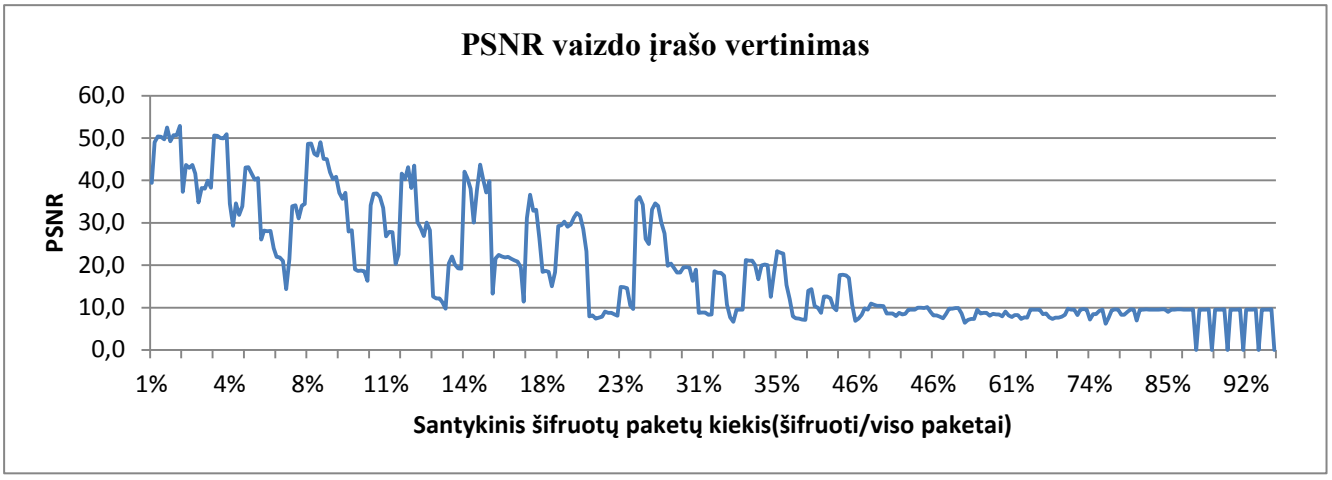
H.264/AVC vaizdo suspaudimo standartas nėra skirtas spausti sintetiniams vaizdams, todėl eksperimento rezultatai skirsis nuo prieš tai aptartų.

10 lentelė. Šifravimo greitaveika ir kokybė statiniams vaizdams

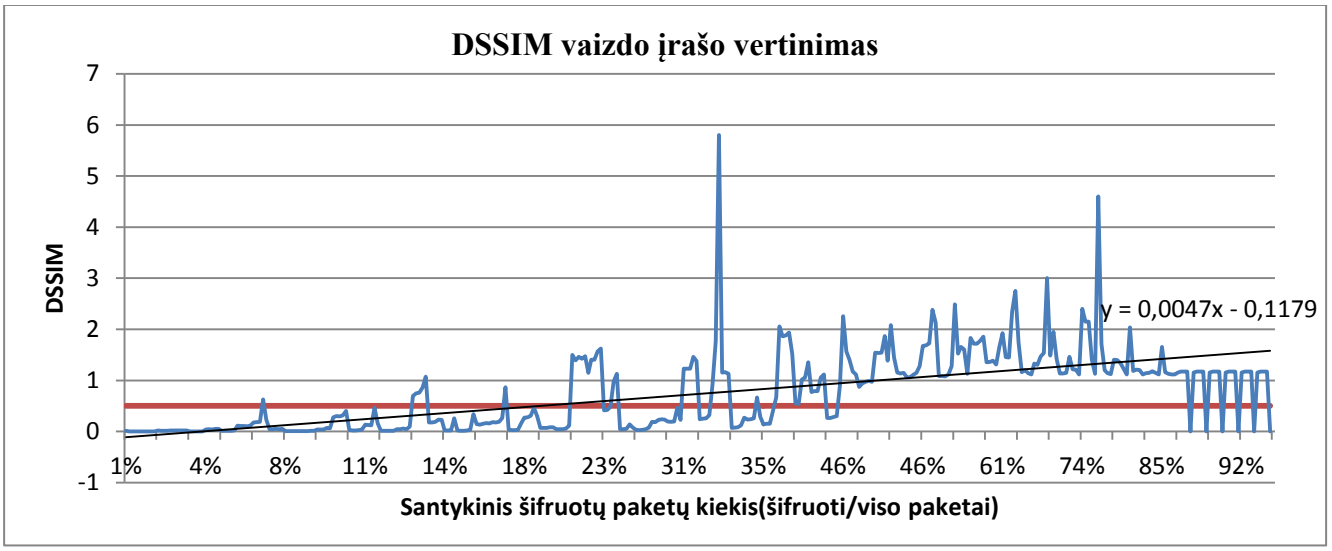
Paketų šifravimas		**Šifravimas paketo viduje: *pck_crypt_length***											
		16 byte				**32 byte**				**176 byte**			
Praleisti - šifruoti	**Šifravimo dažnis**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**	**Šifravimo laikas(s)**	**PSNR**	**DSSIM**	**Greitis (Mb/s)**
0-1	0,92	0,28	9,5	1,15	2,36	0,30	9,51	1,17	2,14	0,40	9,54	0,00	**1,63**
3-1	0,22	0,15	7,95	1,50	4,44	0,15	7,42	1,46	4,17	0,16	7,87	1,47	3,91
3-2	0,37	0,17	7,97	2,05	3,80	0,17	7,4	1,88	3,72	0,21	7,07	1,52	3,12
3-9	0,69	0,22	8,45	1,33	2,89	0,24	7,71	1,47	2,69	0,31	7,59	3,00	2,11
3-12	0,74	0,22	7,18	2,40	2,96	0,26	8,48	2,15	2,48	0,33	9,54	1,13	1,95
6-1	0,13	0,12	12,6	0,70	**5,44**	0,13	12,2	0,76	4,99	0,15	9,75	1,07	**4,42**
6-2	0,23	0,14	9,07	1,15	4,78	0,14	8,79	1,40	4,46	0,17	8,1	1,62	3,89
6-9	0,55	0,21	8,57	1,83	3,05	0,23	8,73	1,72	2,79	0,28	8,5	1,85	2,34
6-12	0,61	0,21	8,41	1,36	3,10	0,24	7,94	1,39	2,72	0,29	8,13	1,66	2,21
12-1	0,07	0,12	22	0,17	5,45	0,12	21	0,19	5,30	0,12	21,4	0,23	5,42
12-2	0,13	0,12	20,4	0,18	5,43	0,12	20,2	0,19	5,19	0,14	19,2	0,23	4,65
12-9	0,39	0,16	12,6	0,77	3,92	0,19	12,3	0,79	3,40	0,23	9,39	1,12	2,86
12-12	0,46	0,18	10,9	0,87	3,59	0,20	10,4	0,97	3,20	0,25	10,4	0,97	2,63
24-1	0,04	0,11	34,8	0,02	5,79	0,10	38	0,02	6,22	0,12	38,3	0,02	5,38
24-2	0,07	0,11	33,9	0,04	5,93	0,12	31	0,05	5,46	0,12	34,4	0,05	5,45
24-9	0,25	0,14	35,2	0,05	4,68	0,16	34,3	0,05	4,08	0,19	25	0,08	3,33
24-12	0,31	0,15	21,2	0,07	4,30	0,16	21,1	0,08	3,95	0,19	16,7	0,27	3,36
32-1	0,03	0,10	37,3	0,02	6,26	0,10	43	0,01	6,31	0,11	41,6	0,02	6,11
32-2	0,05	0,11	43	0,01	5,69	0,11	41,6	0,01	5,98	0,11	40,6	0,02	5,69
32-9	0,20	0,13	29,2	0,07	4,80	0,15	30,3	0,07	4,22	0,18	29,6	0,08	3,56
32-12	0,25	0,14	33,2	0,03	4,67	0,16	34	0,03	4,05	0,18	27,5	0,08	3,55
64-1	0,01	0,10	39,4	0,01	6,58	0,10	50,4	0,00	6,39	0,10	49,7	0,00	6,28
64-2	0,03	0,10	52,5	0,00	6,42	0,10	50,7	0,00	6,21	0,11	52,9	0,00	5,76
64-9	0,12	0,12	41,6	0,01	5,49	0,12	43,1	0,01	5,37	0,14	43,5	0,01	4,55
64-12	0,14	0,13	42,1	0,02	4,96	0,13	38,1	0,02	5,10	0,15	40,4	0,02	4,35



46 pav. TV test vaizdo įrašo apdorojamų duomenų greitis.



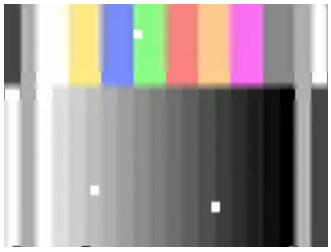
47 pav. Šifruoto vaizdo kokybės vertinimas PSNR metodu.



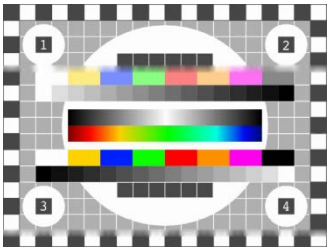
48 pav. Šifruoto vaizdo kokybės vertinimas DSSIM metodu.

Eksperimento metu nustatyta, kad statinis vaizdas atsparesnis atsiradusioms klaidoms sraute, t.y šifruotam duomenų taisymui. Pagal PSNR reikšmes, kurios svyruoja nuo 23 iki 45 dB

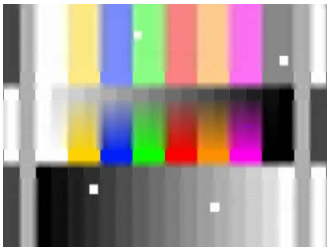
galime daryti išvadas, kad reikia transliacijos srautą šifruoti dažniau. 10 lentelėje, galima išskirti eilutes kur praleidžiamų paketų skaičius *pass_length*=6, PSNR neviršija 15. 6-1 sekoje PSNR yra 12.6, o DSSIM – 0,7, t.y vaizdas tinkamai paslėptas. Vizualinis patikrinimas patvirtina gautus rezultatus(49 pav.). Šifruojant transliacijos srautą seka 6-1 užšifruoti reikia 6478 paketus iš 46323, tai sudaro 14% srauto.



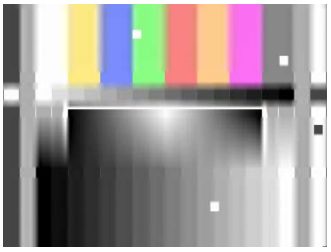
1Statinės transliacijos kadras, šifravimo parametrai *pass_length*=3, *crypt_length*=1, *pck_crypt_length*=16, *pck_offset*=168.



Statinės transliacijos kadras, šifravimo parametrai *pass_length*=24, *crypt_length*=3, *pck_crypt_length*=176, *pck_offset*=8.



Statinės transliacijos kadras, šifravimo parametrai *pass_length*=6, *crypt_length*=3, *pck_crypt_length*=176, *pck_offset*=8.



Statinės transliacijos kadras, šifravimo parametrai *pass_length*=6, crypt_length=1, *pck_crypt_length*=176, *pck_offset*=8.

49 pav. TV test dekoduotos šifruotos transliacijos kadrai

4.4 Eksperimento išvados

Eksperimento atlikimo aplinka pasirinkta kuo panašesnė į transliacijose naudojamus STB, mobiliuosius įrenginius ir kitus mažai resursų turinčius įrenginius. Metodas realizuotas c kalba, daliniam paketų šifravimui panaudota *libcrypto++* šifravimo algoritmų biblioteka. Šifravimo bibliotekoje esanti AES realizacija yra optimizuota, todėl nėra tiesinės priklausomybes nuo šifruojamų duomenų kiekio ir laiko jam užšifruoti. Naudojamas perf stat įrankis laikui matuoti užtikrina tikslų laiko matavimą, nes jis veikia branduolio lygmenyje, veikia kaip kritinis procesas kuriam suteikiami visi reikalingi resursai. Apsaugos kokybei vertinti panaudoti PSNR ir DSSIM kokyb4s vertinimo metodai, PSNR < 15 reiškia kad lyginami kadrai nepanašūs, aukštas triukšmo lygis, lyginimas atliekamas taškas – taškas. DSSIM vertinimo metodas kadrus lygina blokais 8x8 taškai, jei DSSIM > 0,5 vaizdai nepanašūs.

Eksperimentai atlkiti naudojantis priimtais vaizdo įrašais naudojamais vaizdo kodavimo, šifravimo ir kitiems algoritmams lyginti. Vaizdo įrašai pasižymi skirtingomis turinio charakteristikomis, tokiomis kaip daug statiškumo, vientisos spalvos, daug dinamiškai judančių objektų ir pan.

Siūlomas dalinis MPEG-2 dalinis srauto šifravimo algoritmas patikrintas 7 testais, TV bandomasis įrašas sugeneruotas panaudojant vieną sintetinį paveikslėlį.

11 lentelė. Vaizdo įrašų dalinio šifravimo efektyvumas lyginant su pilnu šifravimu.

Vaizdo įrašas	**Šifruojamų paketų seka(praleisti-šifruoti, šifr. pak. dalis)**					**Efektyvumas (dalinis/pilnas)**
	0-1, 100%	**6-1, 14%**	**24-1, 4%**	**32-1, 3%**	**64-1, 2%**	
Akiyo	1,76	5,48	7,97	**8,42**	6,23	4,78
Foreman	1,87	7,93	13,3	11,7	**15,8**	8,45
Container	1,85	7,40	10,9	**12**	14	7,57
Hall monitor	1,86	7,65	12,5	12,7	**13,7**	7,37
Mobile	1,86	7,92	12	14,3	**16,2**	8,70
Waterfall	1,83	7,90	13,00	14,3	18.5	7,81
TV	1,63	**5,44***	5,42	6,11	6,28	3,34

Vaizdo įrašų eksperimentų rezultatų bendram palyginimui sudaryta 11 lentelė, paryškintos reikšmės žymi greitaveikas, prie kurių užtikrinamas pakankamas saugumo lygis vaizdams. Paryškinti laukeliai rodo, kad šifruojama per mažai duomenų, ir apsauga netenkina keliamų reikalavimų.

Eksperimentų metu nustatyta, kad 7 iš 5 atvejų siūlomas dalinis šifravimas yra apie 8 kartus greitesnis nei pilnas, užtikrinant vaizdo įrašo apsaugą. 6 iš 7 vaizdo įrašų pakako užšifruoti iki 3% viso duomenų srauto. TV vaizdo įrašui, sugeneruotas iš paveikslėlio, taikant dalinį šifravimą pasiekta 3,34 didesnė greitaveika nei taikant pilną šifravimą, dalinai šifruojama 16 baitų pakete.

5 IŠVADOS

Labiausiai papiltę tiesioginėse vaizdo transliacijose RTP ir MPEG-2 standartai, naudojami internetinėms, antžeminėms ir satelitinėms transliacijoms perduoti. RTP ir MPEG-2 standartai numato pilną duomenų šifravimą naudojant simetrinius šifravimo algoritmus. Transliacijose vis dar plačiai papiltęs H.263 vaizdo suspaudimo algoritmas, tačiau jį sparčiai keičia pažangesnis H.264, efektyviau suspaudžiantis vaizdo medžiagą, geriau atstatantis nedideles prarastas vaizdo sritis.

Išanalizavus vaizdo šifravimo metodus, pagal šifruojamų duomenų vietą, juos galima suskirstyti į dvi pagrindines grupes: šifravimas vaizdo kodavimo metu, vienoje iš stadijų keičiant, šifruojant duomenis, ir užkoduoto vaizdo šifravimas. Pirmasis metodas keičia standartą, gali įtakoti suspaudimo lygius, šio metodo naudojimas apsunkina vaizdo dekodavimo spartintuvų panaudojimą, nes spartintuvuose nėra numatytas pašalinis šifravimas ar kitoks maišymas. Antrasis metodas yra paprastesnis, šifruojamas jau užkoduotas vaizdas. Vaizdo suspaudimo standartuose, tinkančiuose transliacijoms, yra numatyta klaidų šalinimo ir taisymo metodai, dėl to išauga šifruojamų duomenų kiekis.

Išanalizavus kelis H.264/AVC vaizdo apsaugos metodus naudojamus transliacijose, kurių principas skiriasi, pastebėjome, kad jie turi trūkumų, siekiant kuo mažesnio šifruojamų duomenų kiekio buvo pamirštama apie sudėtingumą juos surandant, ir sudėtingą algoritmo ar raktų pakeitimą. Siūlomas metodas duomenis šifruoja prieš juos perduodant, taip išvengiama papildomų skaičiavimų vaizdo apsaugai. Šifruojama yra MPEG-2 transporto srauto, 188B paketo dalis naudojant AES šifravimo algoritmą. Šifruojant tik dalį paketo pasiekiamas nedidelis šifruojamų duomenų kiekis ir aukštas saugumo lygis.

MPEG-2 dalinio šifravimo metodu realizuotos paprogramės, c kalba, eksperimentams atlikti. Paprogramės matuoja šifruotų ir praleistų paketų kiekius, laikas matuojamas naudojantis perf stat įrankiu.

Eksperimentai atlikti naudojantis bandomaisiais vaizdo failais pasižyminčiais skirtingomis charakteristikomis. Atlikti greitaveikos matavimai, užšifruoto vaizdo kokybės vertinimo PSNR ir DSSIM metodais, siekiant įvertinti siūlomą metodą. Eksperimento mettu nustatyta, kad šifruojant kas 32-to paketo paskutinius 176 baitų, pasiekiamas geras saugumo lygis, ir šifravimo/dešifravimo greitaveika išauga apie 7-8 kartus. Sugeneruotam vaizdo įrašui, reikėjo užkoduoti kas 6-to paketo paskutinius 16 baitų norint jį tinkamai apsaugoti. Naudojant dalinį šifravimą užšifruojama apie 3% duomenų srauto. Apsaugos praeidumas apie 6Mb/s kurio pakanka perduodant H.263 ar H.264 formatu koduotam vaizdui. Siūlomas dalinis MPEG-2 transporto srautas yra tinkamas ribotų skaičiavimų įrenginiams teikti tiesioginių vaizdo tranliacijų paslaugas.

6 LITERATŪRA

[1] Shin-Ho Liu, Han-Yen Yu, Jia-Yen Wu & Jiann-Jone Chen, Jun-Lin Liu and De-Hui Shiue. A Secured Video Streaming System. 2010 International Corerence on System Sience and Engineering, 625-630 psl, 2010.

[2] Yusei Nishimoto, Hiroyuki Imaizumi, Nagahisa Mita. Integrated Digital Rights Management for Mobile TV Using Broadcasting and Communications. IEICE 08 SB 0083, 2008.

[3] Yongdong Wu, Feng Bao. Flexible Accessto Video Streaming. Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, 2568-2571 psl.

[4] Ahmet M. Eskicioglu, Edward J. Delp. AN INTEGRATED APPROACH TO ENCRYPTING SCALABLE VIDEO. Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference, C2002 IEEE, 573-576 psl;

[5] Salah Aly. A Light-Weight Encrypting For Real Time Video Transmission. CTI Symposium Conference, DePaul University, Chicago, USA, November 2003.

[6] NOUR EL DEEN M. KHALIF A, HESHAM N. ELMAHDY. The Impact of Frame Rate on Securing Real Time Transmission of Video over IP Networks. Networking and Media Convergence, 2009. ICNM 2009. International Conference, 57-63 psl.

[7] Ali Saman Tosun, Wu-chi Feng. Efficient Multi-layer Coding and Encryption of MPEG Video Streams. Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference, 119-122 psl.

[8] Gunhee Kim, Dongkyoo Shin, Member, Dongil Shin. Intellectual Property Management on MPEG-4 Video for Hand-Held Device and Mobile Video Streaming Service. Consumer Electronics, IEEE Transactions, 139-143 psl.

[9] Shiguo Lian, Member, Zhongxuan Liu, Zhen Ren, and Haila Wang. Secure Advanced Video Coding Based on Selective Encryption Algorithms. Consumer Electronics, IEEE Transactions, 621-629 psl.

[10] Zheng Liu, Xue Li. Motion Vector Encryption in Multimedia Streaming. Proceedings of the 10th International Multimedia Modelling Conference (MMM'04), Multimedia Modelling Conference, 2004. Proceedings. 10thInternational, 1-8 psl.

[11] Shiguo Lian, Jinsheng Sun, Zhiquan Wang, Yuewei Dai. A Fast Video Encryption Scheme Based-on Chaos. 2004 8th International Conference on Control, Automation, Robotics and Vision Kunming, Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th, 126-131 psl.

[12] Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra. The H.264/AVC Advanced Video Coding Standard:Overview and Introduction to the Fidelity Range Extensions. Presented at the SPIE Conference on Applications of Digital Image Processing XXVIISpecial Session on Advances in the New Emerging Standard: H.264/AVC, August, 2004

[13] John Wiley, Sons Inc. Next generation IPTV services and technologies by Gerard O'Driscoll. 2007, 64-115 ir 118-128 psl.

[14] Cai Mian, Jia Jia, Yan Lei. An H.264 Video Encryption Algorithm Based On Entropy Coding. Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference.

[15] Ali Saman Tosun and Wu-chi Feng. Efficient Multi-layer Coding and Encryption of MPEG Video Streams. Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference, 119-122 psl.

[16] Wail S. Elkilani, Hatem M. Abdul-Kader. Performance of Encryption Techniques for Real Time Video Streaming. Networking and Media Convergence, 2009. ICNM 2009. International Conference, 130-134 psl.

[17] L.Tang. Methods for encrypting and decrypting MPEG video data eficiently. In ACM Multimedia, 1996, 219–229 psl.

[18] Iain Richardson. Overview of H.264 / AVC [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.vcodex.com/h264overview.html

[19] Iain Richardson. H.264/MPEG-4 Part 10 White Paper. [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.vcodex.com/files/h264_overview_orig.pdf

[20] [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.entrust.com/pki.htm

[21] RTSP standartas, [žiūrėta 2011-06-16]. Prieiga per internetą: http://tools.ietf.org/html/rfc2326

[22] SRTP standartas, [žiūrėta 2011-06-16]. Prieiga per internetą: http://tools.ietf.org/html/rfc3711

[23] DRM serverio aprašymas [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.iptvdictionary.com/IPTV_Dictionary_DRM_License_Server_Definition.html

[24] Xiaoyun Liu , Tiejun Huang , Longshe Huo , Luntian Mou. A DRM ARCHITECTURE FOR MANAGEABLE P2P BASED IPTV SYSTEM. Multimedia and Expo, 2007 IEEE International Conference

[25] RTP protokolo specifikacija, [žiūrėta 2011-06-16]. Prieiga per internetą: http://tools.ietf.org/html/rfc3550

[26] RTP protokolo specifikacija, [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.networksorcery.com/enp/protocol/rtcp.htm#Version

[27] SRTP protokolo specifikacija, [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.networksorcery.com/enp/protocol/srtp.htm

[28] SRTCP protokolo specifikacija, [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.networksorcery.com/enp/default1101.htm

[29] VLC grotuvo internetinis puslapis [žiūrėta 2011-06-16]. Prieiga per internetą: http://www.videolan.org/vlc/

[30] YUV Analyzer programos internetinis puslapis [žiūrėta 2011-06-16]. Prieiga per internetą:http://www.sunrayimage.com/

[31] AES šifravimo algoritmas, FIPS 197 [žiūrėta 2011-06-16]. Prieiga per internetą: http://csrc.nist.gov/publications/PubsFIPS.html

[32] PSNR vaizdo kokybes vertinimas [žiūrėta 2011-11-18]. Prieiga per internetą: http://qpsnr.youlink.org/

[33] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing, vol. 13, no. 4, 600-612psl, 2004.

[34] MPEG-2 Transporto srauto aprašymas [žiūrėta 2011-11-22]. Prieiga per internetą: http://www.iptvdictionary.com/iptv_dictionary_MPEG_Transport_Stream_TS_definition.html

7 PRIEDAI

7.1 Efficient MPEG-2 Transport Stream Encryption Method for Low Processing Power Mobile Devices

Straipsnis spausdintas ELECTRONICS AND ELECTRICAL ENGINEERING 2012m Nr. 8(118) žurnale. Kaunas : KTU. ISSN 1392-1215, 81-88 psl.

ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2012. No. 2(118)*
ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
T 120
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Efficient MPEG-2 Transport Stream Encryption Method for Low Processing Power Mobile Devices

V. Simanaitis, A. Liutkevicius, A. Vrubliauskas, E. Kazanavicius
Real Time Computing Systems Centre, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, e-mails: vytautas.simanaitis@stud.ktu.lt, agnius@ifko.ktu.lt, aras@ifko.ktu.lt, ekaza@ifko.ktu.lt
D. Imbrasas
JSC "Elsis TS",
Uosio St. 10, Kaunas, Lithuania, e-mail: darius.imbrasas@elsis.lt

crossref http://dx.doi.org/10.5755/j01.eee.118.2.1180

Introduction

An intellectual property rights protection became a very common problem in recent years. Especially it is important for the digital video content such as IPTV providers, because their business continuity is directly affected if anybody can use that content illegally. Smart-home and smart-environment content provisioning systems, like SNAPAS [1] also must protect their digital content and ensure, that end-users with their low power terminal devices will be able to watch video stream in real-time. Usually content is protected using digital rights management (DRM) systems, whose allow only authorized users to watch protected content. Various topologies and architectures are used for the IPTV streaming with different DRM solutions, leading to various problems when trying to protect content and still deliver good quality services to customers.

MPEG-4 H.264/AVC video streaming format became very popular recently. It has many advantages comparing with older MPEG H.261 encoding format, because it requires less internet bandwidth, has better decoded video quality, many IPTV STB devices and TV have hardware decoding accelerators.

DRM solutions are using video stream data encryption methods. Usually AES method is used, because it is fast and reliable [2, 3]. Other methods like outdated DES or not very safe XOR are still widely used as well. Not all encryption methods have the same reliability level, while most safe methods are more resource and time consuming. This is not a big problem, when video content users have powerful end-user terminal devices, like high-end PC or similar. The problem arises, when users watch video content using low computing power STB (set-top-box) devices or handheld devices like phones, PDA, etc. These devices are restricted enough and are not able to decrypt streams with complex encryption methods in real-time.

This paper presents novel video content streaming encryption method, which is less computational resource demanding and compatible with end-users video stream decoding systems. This method is suitable for low computing power end-user devices (phones, PDA, STB, etc.) and is capable to encrypt various types of encoded streams, like H.264/AVC or older H.263 and H.261. The proposed encryption method ensures the same level of protection like standard MPEG-4 full encryption method, but it is faster and compatible with almost any existing DRM systems.

Video streaming methods

Video streaming can be categorized into three types: unicast, multicast and broadcast. Paid content is protected from illegal use by adapting data encryption algorithms, while algorithm specific information (like public or private keys, codes, etc.) is provided to the end-users via DRM system using safe channels.

Video streams are of two types: real-time video and already prepared (non real-time) video content streaming. Best known example of video streaming service is internet television (IPTV). Video content should be prepared in such way, that end-users would be able to watch this content using their terminal devices, usually STB connected to TV or TV with integrated STB. IPTV STB device receives video stream from the content provider, decrypts it (if stream is protected), decodes it and sends the output to the TV. At this moment H.264/AVC encoding is most efficient and a best quality ensuring standard.

Both real-time and non real-time streaming uses the same standards and streaming methods. But real-time streaming requires more efficient encoding and encryption methods, because all operations are made on-the-fly, ensuring minimal delays.

Not the all real-time streaming protocols can be applied for different streaming (and protection) methods. Protocols are dedicated for unicast, multicast or broadcast with or without encryption. RTP (Real-Time Transport Protocol) together with SRTP (Secure Real-Time Transport Protocol) are used for the real-time video streaming. SRTP and SRTCP (Secure Real-Time Control Protocol) are the RTP and RTCP (Real Time Control Protocol) versions with data encryption added. SRTP ensures full video stream encryption, while SRTCP is used to share encryption keys and encryption algorithm between content provider and end-user. Other widely used real-time streaming protocol is RTSP (Real Time Streaming Protocol). It is not used for content streaming but rather for controlling streaming servers and providing users with such functions like play, stop, pause, etc. At some cases TCP or HTTP protocols can be used for video streaming as well, but they are not very suitable for such purpose.

A high quality video stream must be compressed effectively in order to send it over a low bandwidth networks. More effective H.264 encoding standard supersedes widely used H.263 encoding standard. H.263 encoding requires 4Mb/s network transfer rate in order to transmit 720x576 25 frames stream, while H.264/AVC encoding requires only 1.5-2 Mb/s transfer rate. H.264 is more effective because of new compression algorithms applied. H.264 encoding is based on video stream frame segmentation in so called macro blocks (MB). According to standard, macro blocks size can be 4x4, 8x8, or 16x16 pixels. There are three types of MB: I-slice, P-slice and B-slice. Each MB is encoded using Discrete Cosine Transform (DCT) and the output is matrix, which is transformed into the digital data sequence using Zig-Zag method. Then the sequence is compressed by Run Length Encoding and Huffman method. Decoding is performed in reverse order and does not require additional computations to divide frames into I, B or P type. H.264 encoding loses some quality because of optimizations during compression, e.g. after applying DCT small coefficients become 0, so before frame is displayed, it should be filtered.

H.264 standard is error-resistant, when errors appear during video stream transmission over unreliable IP networks. The main idea is to find error and to conceal it using appropriate method. Usually it is done by searching similarities in nearby macro blocks, e.g. if 16x16 MB is damaged or lost, it can be treated as zero and displayed as black or green region in the frame. Spatial interpolation, motion compensated interpolation, temporal concealment and similar methods can be applied to repair damaged MB. These methods are good enough to recover one or few MB, but in very noisy channels it is a difficult task, because lost packets contain several MB.

Related work

Compressed video stream is not error resilient, because errors propagate to the subsequent frames and macro blocks. Error propagation can be used for the video stream encryption and protection, because encrypted data are seen as corrupted data at the receiving side. Existing video stream encryption methods and techniques can be categorized into several types as described below.

Full Encryption. There are many video content protection and transmission systems using full stream encryption, like [4] or [5], but they require powerful end-user devices and high bandwidth of the network. Usually full encryption is implemented using SRP and SRTCP protocols [6]. The main advantages of full encryption systems and protocols are independence of video stream encoding standard and very high protection level. The main disadvantage is that decryption process is very resource consuming, hence mobile devices like phones or PDA are not capable to perform such task in real-time.

Selective Encryption. Tang in [7] proposed four different levels of stream encryption with different protection levels. First and least secure method is encryption of all headers. In this case decoder does not know the encoding standard, but such protection is very primitive, since real-time streaming is performed using only few well known standards and headers can be simply guessed. Second level of protection is obtained by encryption of all headers and I-frames. Third level includes encrypting of all I-frames and I-macro blocks in P and B frames, but not the headers. This ensures good protection, because even encoding is known, the content itself is encrypted and cannot be recovered. Fourth level is the most secure, because all frames are encrypted.

Zig-Zag Permutation. This method exploits one of the stages of H.264 encoding algorithm, where 8x8 blocks are transformed into 1x64 sequence using Zig-Zag algorithm. If sequence order is unknown, the decoder cannot recover the matrix and complete decoding successfully. Such method ensures low protection level, because once permutation principle is known, algorithm is not secure any longer. Also such encryption leads to worse compression level of data, because after applying Zig-Zag and DCT algorithms, non-zero coefficients are placed in the beginning of the sequence and can be compressed effectively using Run Length Encoding. In case of Zig-Zag permutation, non-zero coefficients are distributed over the sequence and compression is not so effective [8].

Motion Vector Encryption. Video stream contains frames consisting of macro blocks. Each MB can have motion vector, which specifies which part of frame should be copied into the specific MB. Vector points to the same frame or other frame. MVEA (Motion Vector Encryption Algorithm) was proposed in [9]. Motion vector encryption has two stages. In the first stage motion vectors are concealed using XOR algorithm. In the second stage motion vectors are scrambled with random generated numbers.

DCT coefficient encryption. DCT encryption proposed in [10] is based on three levels of protection. For each level the different number of DCT coefficients is encrypted. First level includes 0-4 coefficients, second 5-19 and third includes remaining 44 coefficients. Depending on desired protection degree, several levels can be encrypted. Coefficients are usually distributed in the first quarter of matrix, which corresponds to the first and

second protection level. Encryption of the first level coefficients guarantees worst protection degree, while encryption of all levels gives the best protection. Such method should maintain a good video stream compression level, because 0 coefficients are not encrypted.

More simple method is proposed in [11], where only signs of DCT matrix coefficients are encrypted. But this encryption is not very strong, since coefficient can have only two possible "+" or "-" signs, which can be guessed. Sign encryption also is used in [12], where signs of motion vectors are encrypted.

Partial encryption. Partial encryption is quite simple, but still effective approach. Partial encryption method of MPEG-2 transport stream is proposed in [13], where transport stream packets containing I-frames are encrypted. But encryption of only I type of blocks does not guarantee good protection level, because some portions of video can be restored from motion vectors. Most of partial encryption methods perform encryption not at the transport level, but at the encoding level, e.g. method proposed in [14] encrypts part of all types of macro blocks using DES.

Analyzed encryption methods have several drawbacks. Some of them require high computational power. Others perform encryption at the encoding level, which gives good protection level, but leads to the incompatibilities with decoding accelerators. Encryption at encoding level also requires re-encoding of at least a half of the video file, when encryption algorithm or secret key needs to be changed. Partial encryption methods are more lightweight and suitable for real-time streaming, but existing techniques are based on the analysis and encryption of specific I, P, or B type of frames or macro blocks, which requires additional computational steps.

In this paper we propose a novel partial encryption method, which performs encryption not at the encoding level, but at the transport level (MPEG-2 transport stream). Proposed method encrypts video stream data not taking into account the type of data (is it motion vector, I-type MB or other type of data). This eliminates additional steps and computations to find I-type macro blocks or motion vectors. Another advantage is that in order to change encryption algorithm or secret key, only transport stream must be re-encrypted, while encoded video data can be prepared before and just reused.

Proposed partial encryption method

Proposed method encrypts MPEG-2 transport stream (which is also used for the MPEG-4 video streaming), that consists of 188 Byte length data packets. Encrypting data at this level should take into account the fact, that H.264/AVC encoding and compression standard has corrupted frames recovery feature, hence enough data should be encrypted to guarantee desired protection level.

The main restriction for the real-time IPTV streaming is a low computing power end-user handheld devices. They are not able do decrypt large portions of video streams encrypted using complex encryption methods. Hence only the most important and hardly guessable parts of video stream should be encrypted. Since STB video decoding is performed using HW, encryption method should not interfere with encoding/decoding process, like many other encryption methods do.

The MPEG-2 transport stream consists of 188B packets, which must be collected in correct order. Otherwise, decoding will not be successful. This means, that mixing the order or simply encrypting the packets, will not allow restoring and decoding the stream, if encryption methods or keys are not known. Proposed method is fully compatible with hardware video decoding, because video stream data would be decrypted before sending them to the decoder. Another advantage is that once video stream is encoded (e.g. using H.264 format), it can be encrypted with different encryption algorithms and keys. This is useful for streaming to the different end-user groups, using different encryption keys.

Proposed partial transport stream encryption method encrypts part of the 188B MPEG-2 transport stream packet. Packet consists of 4B header, and 184B data. For the compatibility with standard purpose the header remains unencrypted and only part of the data payload is encrypted. Moreover, in order to reduce encryption and decryption time and requirements for the decryption device, only part of transport packets are encrypted.

Proposed encryption algorithm consists of several steps:

1. Secure encryption keys are created for the AES algorithm and distributed to the end-users through DRM system.
2. Encryption is based on four parameters, which are set before encryption: *pass_length* defines the length of unencrypted packets block; *crypt_length* defines the length of encrypted packets block, which follows each unencrypted packets block defined by *pass_length*; *pck_offset* defines the encryption offset inside MPEG-2 transport packet's payload data; *pck_crypt_length* defines how much bytes should be encrypted starting from *pck_offset* position.
3. Transport packet is checked and only packet without *adaptation fields* is encrypted.
4. *Pass_length* and *crypt_length* values are evaluated and if packet needs to be encrypted, then part of the packet is encrypted according to the *pck_offset* and *pck_crypt_length* (see Fig. 1). Packet's *scr* flag is set, which means that packet is encrypted (scrambled).
5. Next stream packet is processed (return to step 3).

One of the main advantages of proposed algorithm, comparing with other partial encryption algorithms, is that encryption is performed at the transport level. This means that video content can be encoded using different methods, like H.264, H.263 or H.261, etc. The proposed method is compatible with MPEG streaming standards and no additional data or custom flags are used in transport packet for achieving full compatibility with standards. 188 Byte length transport packet already contains flag, indicating that packet is encrypted.

This flag is reused by proposed encryption algorithm by setting the flag for the encrypted packets.

AES encryption algorithm is used for the experimental evaluation of proposed method. This algorithm is lightweight but still very effective [2, 3].

Any other existing encryption algorithms can be used with proposed encryption method as well.

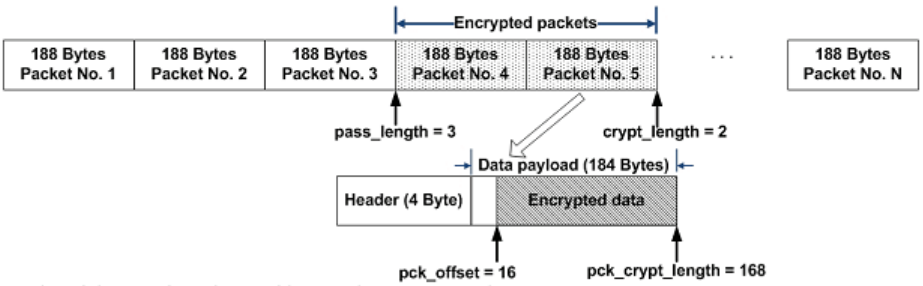


Fig. 1. Proposed partial encryption scheme with example parameters values

Decryption process for the proposed encryption method is rather simple. To decrypt the stream receiver needs to know the secret key, *pck_offset* and *pck_crypt_length* parameters:

1. Received transport packet is inspected for the *scr* flag;
2. If *scr* flag is set, the encrypted part of the packet is identified using *pck_offset* and *pck_crypt_length* parameters and decrypted using secret key;
3. Decrypted packet is sent to decoder;
4. Next stream packet is processed (return to step 1).

Experimental Methodology and Setup

Proposed partial encryption method was evaluated experimentally measuring encryption and decryption times and encrypted video quality. VLC media player (http://www.videolan.org/vlc/) plug-in has been implemented which incorporates proposed encryption algorithm and is able to encrypt and decrypt video data. The plug-in is based on SRTP protocol, where standard encryption is replaced by proposed encryption algorithm.

The main purpose of experimental evaluation was to evaluate the performance of full encryption and proposed protection method at the same time maintaining acceptable protection level.

The encrypted video stream protection level was evaluated measuring video stream quality using peak signal-to-noise ratio (PSNR). Higher PSNR values mean better image quality. For IPTV video streaming at least 30 dB should be reached, while for wireless transmissions 20 dB values are acceptable. If PSNR is below 10 dB, video quality is so low, that end-users see only random noise.

The number of encrypted packets was measured in order to evaluate the percentage of encrypted/decrypted data.

Two experimental video streams were used. One stream contained dynamic video and other stream contained static video (one image shown for several seconds). According to MPEG standard it is obvious, that dynamic videos can be protected with few encrypted packets. Static video needs more encryption, because video can be recovered even from one key frame. We evaluated protection level of proposed method according to the PSNR and visual inspection of the encrypted video.

Intel Atom based end-user low power mobile device was used for the experiments.

VLC plug-in was compiled using *gcc* compiler and integrated with freely available VLC libraries and source code. The plug-in was developed using KDevelop4 with lua language support.

Encryption and decryption times were measured using *perf* tool. Video frames analysis was performed using *pnmpsnr* tool.

Evaluation of Proposed Encryption Method

The evaluation of proposed encryption method was done via series of experiments, with different *pass_length*, *crypt_length*, *pck_offset* and *pck_crypt_length* parameter values. The main purpose was to find the threshold values of the parameters, which guarantee good level of protection with minimum encrypted data.

Table 1 – Table 2 contain encryption and decryption times, encrypted packets numbers, and PSNR values, depending on the *pass_length*, *crypt_length*, *pck_crypt_length, pck_offset* parameters. Experimental evaluation starts with full transport stream packets encryption (*pass_length=0, crypt_length=1*) and ends with *pass_length=24* and *crypt_length=3* values, where group of three packets was encrypted after each 24 unencrypted packets group. At the same time encryption inside the each selected packet was performed with three different combinations of *pck_crypt_length* and *pck_offset* parameters, resulting in 16, 64 or 176 encrypted bytes.

The number of passed and encrypted transport stream packets, encryption and decryption times and PSNR values were found for each combination of *pass_length*, *crypt_length*, *pck_crypt_length, pck_offset* parameters.

In some cases PSNR value was N/A, which means that decoding software was unable successfully decode encrypted stream and simply crashed, which means the highest protection level possible (or like PSNR $\leq$ 0dB).

The time values presented in Table 1 – Table 2 and Fig. 2 are the average of ten values, i.e. each row of the tables corresponds to the ten experimental measurements.

It can be clearly seen, that even when *pass_length* value is high (24 in our experimental case), the PSNR is still very low (about 10 dB). This means, that somebody who is trying illegally watch this stream see only screen with noise. At the same time the number of encrypted packets and corresponding encryption and decryption times are reduced dramatically comparing with full encryption. Comparing with standard full MPEG encryption algorithm, which encrypts all data payload (184 Bytes) of each stream packet, the proposed encryption method with *pass_length=24, crypt_length=1*, *pck_crypt_length=16* and *pck_offset=168* is about 9 times

faster, but still maintaining almost the same protection level. The last column's first three rows in Table 1 are most close to the standard MPEG encryption with difference only 8 bytes (176 against 184). Full encryption takes ~4.5 s, while 24-1 type encryption takes only ~0.5 s, while decryption takes ~4.2 s and ~0.5 s respectively.

We also performed evaluation of encryption for the static video stream. The results are presented in Table 3 – Table 4 and Fig. 3 – Fig. 6. As mentioned before, it is more difficult to encrypt static images at the transport stream level. The reason is that proposed method does not take into account what type of video frame is inside transport stream packet.

The main feature of static videos is that there is one I-frame with reference image, and other frames are of type B or P. Since proposed method does not analyze frames, additional experimental evaluation was performed to find out the method parameters, which guarantee not only low PSNR value, but also ensure such frequency of encrypted packets, that I-frame becomes encrypted anyway.

Table 1. Encryption performance and quality, encrypted packet count and PSNR values for dynamic video

Encryption								
Sequence of encrypted packets: *pass_length-crypt_length*	**Packets**		**Encryption inside the packet:** *pck_crypt_length* (*pck_offset*)					
			16 Bytes (offset 168)		**64 Bytes (offset 120)**		**176 Bytes (offset 8)**	
	Passed	**Encrypted**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**
0-1	960	45363	2.8658	2.74	3.2441	2.71	4.4009	N/A
0-2	960	45363	2.7102	2.74	3.1088	2.71	4.3892	N/A
0-3	960	45363	2.5417	2.74	3.0905	2.71	4.2509	N/A
3-1	35000	11323	0.9774	10.25	1.1299	10.12	1.4309	N/A
3-2	28179	18144	1.2897	2.8	1.4817	10.29	1.9756	N/A
3-3	23602	22721	1.5773	2.76	1.7740	2.8	2.3702	N/A
6-1	39845	6478	0.8468	10.26	0.8123	10.12	1.0061	10.24
6-2	35008	11315	0.9691	10.23	1.0951	10.2	1.5107	10.19
6-3	31193	15130	1.1536	4.64	1.3351	2.87	1.6998	N/A
24-1	44509	1814	0.5126	10.08	0.5474	10.08	0.5976	10.26
24-2	42826	3497	0.5871	2.83	0.6658	2.89	0.7770	N/A
24-3	41289	5034	0.6516	10.09	0.7489	10.07	0.8763	N/A

Table 2. Decryption performance and quality, decrypted packet count and PSNR values for dynamic video

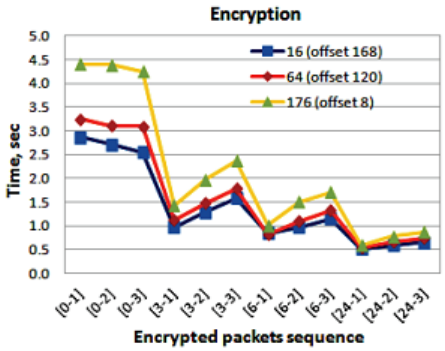
Decryption								
Sequence of encrypted packets: *pass_length-crypt_length*	**Packets**		**Encryption inside the packet:** *pck_crypt_length* (*pck_offset*)					
			16 Bytes (offset 168)		**64 Bytes (offset 120)**		**176 Bytes (offset 8)**	
	Passed	**Decrypted**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**
0-1	960	45363	2.4868	∞	3.0832	∞	4.1748	∞
0-2	960	45363	2.5308	∞	3.0077	∞	4.1784	∞
0-3	960	45363	2.5175	∞	3.0485	∞	4.3418	∞
3-1	35000	11323	0.9406	∞	1.0720	∞	1.3805	∞
3-2	28179	18144	1.2459	∞	1.4632	∞	1.9340	∞
3-3	23602	22721	1.4841	∞	1.7028	∞	2.5169	∞
6-1	39845	6478	0.7268	∞	0.8115	∞	0.9923	∞
6-2	35008	11315	0.9175	∞	1.0683	∞	1.3605	∞
6-3	31193	15130	1.2854	∞	1.3010	∞	1.6937	∞
24-1	44509	1814	0.5111	∞	0.5253	∞	0.5732	∞
24-2	42826	3497	0.5889	∞	0.6380	∞	0.7115	∞
24-3	41289	5034	0.6643	∞	0.7219	∞	0.8292	∞

Table 3. Encryption performance and quality, encrypted packet count and PSNR values for static video

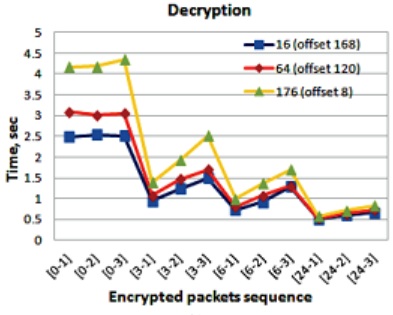
Encryption								
Sequence of encrypted packets: *pass_length-crypt_length*	**Packets**		**Encryption inside the packet:** *pck_crypt_length* (*pck_offset*)					
			16 Bytes (offset 168)		**64 Bytes (offset 120)**		**176 Bytes (offset 8)**	
	Passed	**Encrypted**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**
0-1	960	45363	0.4091	9.53	0.3093	9.51	0.3731	N/A
0-2	960	45363	0.2491	9.53	0.3022	9.51	0.3939	N/A
0-3	960	45363	0.2525	9.53	0.2921	9.51	0.6663	N/A
3-1	35000	11323	0.1345	7.45	0.1455	7.29	0.1652	8.74
3-2	28179	18144	0.1586	6.98	0.1994	6.98	0.2119	6.98
3-3	23602	22721	0.3487	9.06	0.4034	7.94	0.2389	7.42
6-1	39845	6478	0.1188	11.56	0.1240	11.05	0.1539	9.33
6-2	35008	11315	0.1386	8.88	0.1551	7.86	0.1662	7.84
6-3	31193	15130	0.1761	8.32	0.1610	8.32	0.2033	8.32
24-1	44509	1814	0.1020	34.09	0.1039	45.27	0.1125	45.29
24-2	42826	3497	0.1084	27.15	0.1118	27.15	0.1187	26.62
24-3	41289	5034	0.1269	23.43	0.1219	30.46	0.1444	26.17

Table 4. Decryption performance and quality, decrypted packet count and PSNR values for static video

Decryption								
Sequence of encrypted packets: *pass_length-crypt_length*	**Packets**		**Encryption inside the packet:** *pck_crypt_length* (*pck_offset*)					
			16 Bytes (offset 168)		**64 Bytes (offset 120)**		**176 Bytes (offset 8)**	
	Passed	**Decrypted**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**	**Time (s)**	**PSNR (dB)**
0-1	960	45363	0.2487	∞	0.4146	∞	0.3670	∞
0-2	960	45363	0.2669	∞	0.2993	∞	0.3708	∞
0-3	960	45363	0.2479	∞	0.2815	∞	0.4437	∞
3-1	35000	11323	0.1325	∞	0.1411	∞	0.1623	∞
3-2	28179	18144	0.1723	∞	0.1709	∞	0.2209	∞
3-3	23602	22721	0.3727	∞	0.1927	∞	0.2495	∞
6-1	39845	6478	0.1311	∞	0.1263	∞	0.1447	∞
6-2	35008	11315	0.1331	∞	0.1432	∞	0.1829	∞
6-3	31193	15130	0.1492	∞	0.1583	∞	0.2169	∞
24-1	44509	1814	0.1108	∞	0.1023	∞	0.1067	∞
24-2	42826	3497	0.1079	∞	0.1206	∞	0.1167	∞
24-3	41289	5034	0.1156	∞	0.1313	∞	0.1401	∞

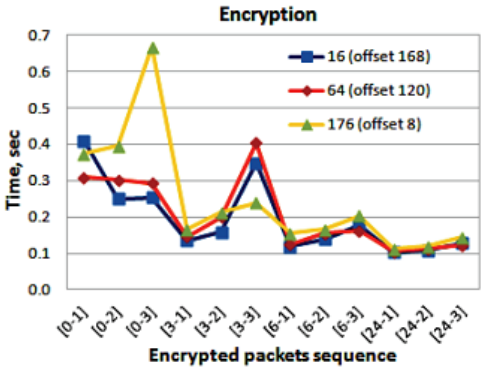


a)

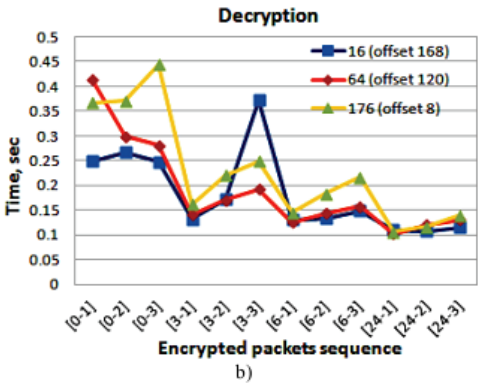


b)

Fig. 2. Encryption and decryption processing times for dynamic video



a)



b)

Fig. 3. Encryption and decryption processing times for static video

We found, that static video needs more frequent packet encryption in order to reduce PSNR values enough. When *pass_length* value is 24, video output is seen as almost unchanged original picture (Fig. 4).

Accordingly PSNR values are very high (~23–45 dB) which also corresponds to acceptable and even very good image quality. PSNR values drop to the acceptable level for protection (~8-11 dB), when *pass_length* ≤ 6.

Visual inspection of such encrypted video shows mainly noise with some color regions, but the original image is unpredictable (Fig. 5 and Fig. 6).

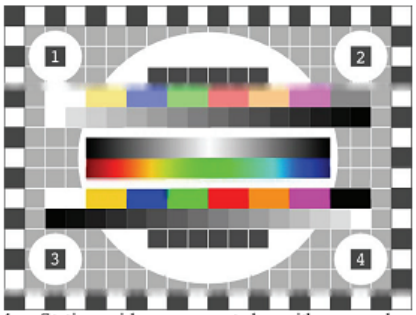


Fig. 4. Static video encrypted with pass_length=24, crypt_length=3, pck_crypt_length=176, pck_offset=8

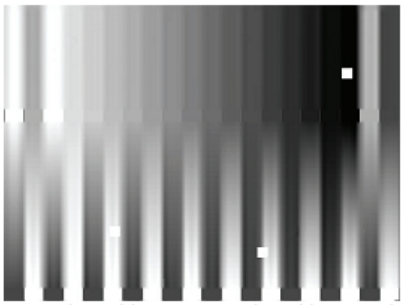


Fig. 5. Static video encrypted with pass_length=6, crypt_length=3, pck_crypt_length=176, pck_offset=8

Fig. 6. Static video encrypted with pass_length=6, crypt_length=3, pck_crypt_length=176, pck_offset=8

Conclusions

In this paper we proposed partial encryption method for MPEG-2 transport stream encryption.

This method is fast and efficient, suitable for end-user's low computing power mobile devices. Only part of transport stream packets is encrypted, encrypting only part of selected packets.

The experimental results show that proposed transport stream partial encryption method ensures good video content protection levels, at the same time substantially reducing encryption and decryption times comparing with standard MPEG full encryption. Method is lightweight and does not require high computing power.

It was found, that for dynamic videos (rapidly changing video frames), it is enough to encrypt 16 Bytes portion of each 24th transport packet, to achieve good protection level (PSNR ~10 dB). For static videos (video frames do not change over time, or changes are minimal) it is enough to encrypt 16 Bytes portion of each 6th transport packet, to achieve good protection level (PSNR ~11 dB). Encryption and decryption times, comparing with full encryption, are reduced from 3 (static video) up to 9 times (dynamic video), making proposed method a good choice for low processing power decryption devices.

References

1. **Liutkevičius A., Vrubliauskas A., Kazanavičius E., Imbrasas D.** Smart home services development, provisioning, and management framework // Information technology and control. – Kaunas: Technologija, 2011. – Vol. 40. – No. 2. – P. 163–169.

2. **Toldinas J., Štuikys V., Ziberkas G., Naunikas D.** Power Awareness Experiment for Crypto Service–Based Algorithms // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 5(101). – P. 57–62.
3. **Toldinas J., Stuikys V., Damasevicius R., Ziberkas G., Banionis M.** Energy Efficiency Comparison with Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 2(108). – P. 11–14.
4. **Shin–Ho L., Han–Yen Y., Jia–Yen W., Jiann–Jone C., Jun–Lin L., De–Hui S.** A Secured Video Streaming System // International Conference on System Science and Engineering, 2010. – P. 625–630.
5. **Nishimoto Y., Imaizumi H., Mita N.** Integrated Digital Rights Management for Mobile TV Using Broadcasting and Communications // 14th Asia–Pacific Conference on Communications (APCC'2008), 2008. – P. 1–5.
6. **Khalifa N. E.–D. M., Elmahdy H. N.** The Impact of Frame Rate on Securing Real Time Transmission of Video over IP Networks // International Conference on Networking and Media Convergence (ICNM'2009). – P. 57–63.
7. **Tang L.** Methods for Encrypting and Decrypting MPEG Video Data Efficiently // Proceedings of the ACM Multimedia, 1996. –P. 219–229.
8. **Qiao L., Nahrstedt K.** Comparison of MPEG Encryption Algorithms // Computers & Graphics. 1998. – Vol. 22. – Iss. 4. – P. 437–448.
9. **Zheng L., Xue L.** Motion Vector Encryption in Multimedia Streaming // Proceedings of the 10th International Multimedia Modelling Conference, 2004. – P. 64–71.
10. **Tosun A. S., Feng W. C.** Efficient Multi–layer Coding and Encryption of MPEG Video Streams // IEEE International Conference on Multimedia and Expo (ICME'2000). – Vol. 1. – P. 119–122.
11. **Shiguo L., Zhongxuan L., Zhen R., Haila W.** Secure Advanced Video Coding Based on Selective Encryption Algorithms // IEEE Transactions on Consumer Electronics, 2006. – Vol. 52. – Iss. 2. – P. 621–629.
12. **Shiguo L., Jinsheng S., Zhiquan W., Yuewei D.** A Fast Video Encryption Scheme Based–on Chaos // 8th International Control, Automation, Robotics and Vision Conference, 2004. – Vol. 1. – P. 126–131.
13. **Jeong–Hyun K., Yeon–Jeong J., Ki–Song Y.** Protection scheme for secure MPEG–2 streaming // IEEE International Conference on Multimedia and Expo, 2004. – Vol. 2. – P. 927–930.
14. **Gunhce K., Dongkyoo S.** Dongil S. Intellectual property management on MPEG–4 video for hand–held device and mobile video streaming service // IEEE Transactions on Consumer Electronics, 2005. – Vol. 51. – Iss. 1. – P. 139–143.

Received 2011 09 12
Accepted after revision 2011 12 05

V. Simanaitis, A. Liutkevicius, A. Vrubliauskas, E. Kazanavicius, D. Imbrasas. Efficient MPEG-2 Transport Stream Encryption Method for Low Processing Power Mobile Devices // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 2(118). – P. 81–88.

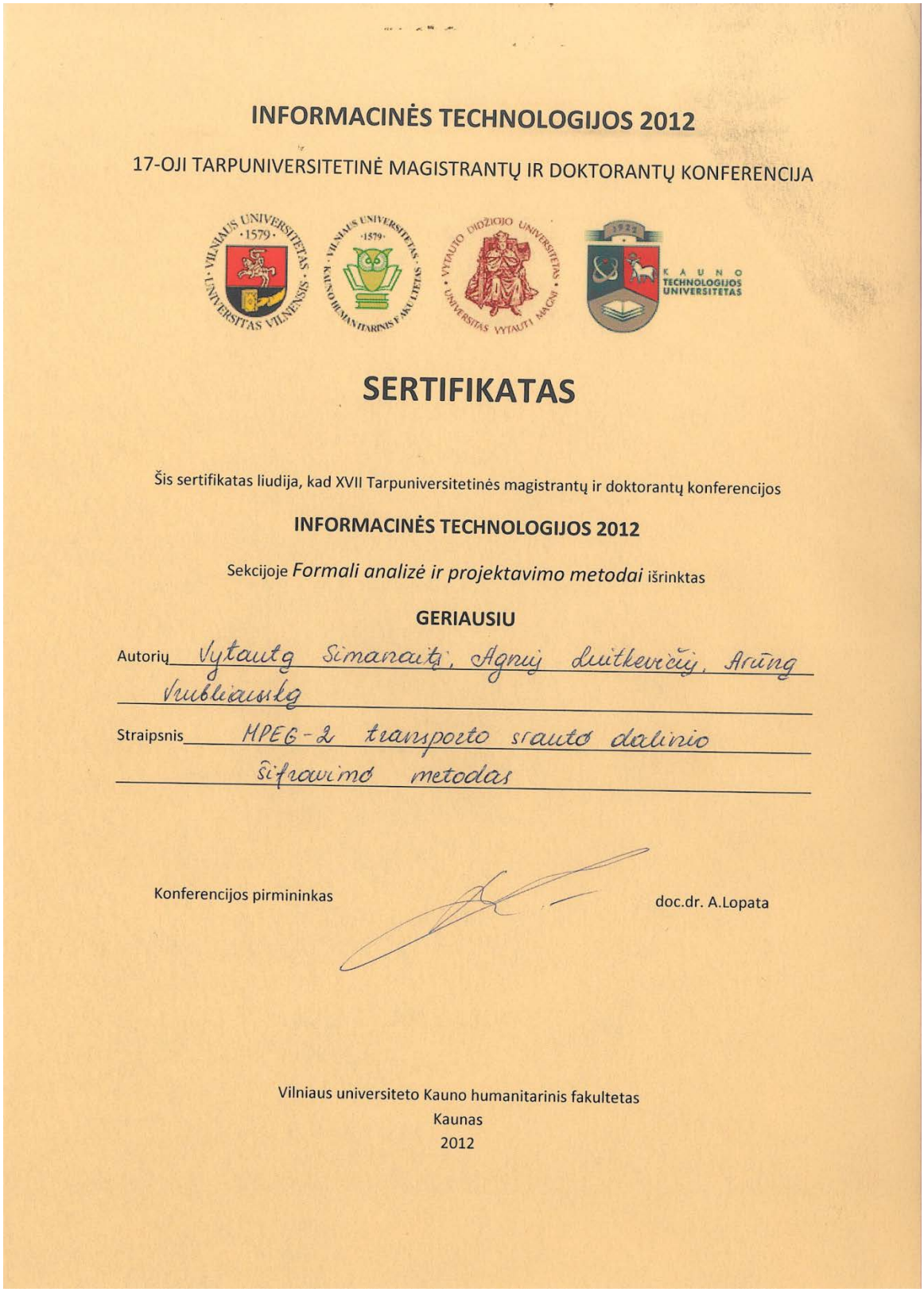
Video stream encryption is the main method for the protection of intellectual property in modern digital rights management systems (DRM). Existing encryption methods ensure very high protection level of encrypted content, but at the same time are very resources demanding, making them hardly suitable for low processing power mobile end-user devices. This paper presents partial MPEG-2 transport stream encryption method, which is suitable for such low power user terminals. The experimental evaluation of proposed method show that encryption and decryption times, comparing with standard full encryption, are reduced from 3 (static video) up to 9 times (dynamic video), making proposed method a good choice for low processing power decryption devices. Ill. 6, bibl. 14, tabl. 4 (in English; abstracts in English and Lithuanian).

V. Simanaitis, A. Liutkevičius, A. Vrubliauskas, E. Kazanavičius, D. Imbrasas. Efektyvus MPEG-2 transporto srauto šifravimo metodas ribotiems mobiliųjų įrenginių ištekliams skaičiuoti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 2(118). – P. 81–88.

Videosrauto šifravimas yra pagrindinis metodas intelektinei nuosavybei užtikrinti šiuolaikinėse skaitmeninių teisių valdymo sistemose (DRM). Esami šifravimo metodai garantuoja labai aukštą apsaugos lygį, bet tuo pat metu reikalauja labai didelių skaičiavimo išteklių ir yra sunkiai pritaikomi ribotų skaičiavimo išteklių mobiliesiems vartotojų įrenginiams. Šiame straipsnyje pristatomas dalinis MPEG-2 transporto srauto šifravimo metodas, kuris yra tinkamas tokiems ribotų resursų terminaliniams vartotojų įrenginiams. Siūlomo metodo eksperimentinio įvertinimo rezultatai rodo, kad šifravimo ir dešifravimo trukmės, palyginti su standartiniu visišku šifravimu, yra sutrumpinamos nuo trijų (esant statiniam vaizdui) iki devynių kartų (esant dinaminiam vaizdui), todėl šis metodas tinka ribotų skaičiavimo išteklių dešifravimo įrenginiams. Il. 6, bibl. 14, lent. 4 (anglų kalba; santraukos anglų ir lietuvių k.).

7.2 Tarpuniversitetinės magistrantų doktorantų konferencijos sertifikatas.

Straipsnis “MPEG-2 transporto srauto dalinio šifravimo metodas” pristatytas Tarpuniversitetinėje magistrantų ir doktorantų konferencijoje 2012-04-20 Kaune, Vilniaus universiteto Kauno humanitariniame fakultete.



INFORMACINĖS TECHNOLOGIJOS 2012

17-OJI TARPUNIVERSITETINĖ MAGISTRANTŲ IR DOKTORANTŲ KONFERENCIJA

SERTIFIKATAS

Šis sertifikatas liudija, kad XVII Tarpuniversitetinės magistrantų ir doktorantų konferencijos

INFORMACINĖS TECHNOLOGIJOS 2012

Sekcijoje *Formali analizė ir projektavimo metodai* išrinktas

GERIAUSIU

Autorių Vytautą Simanaitį, Agnių Liutkevičių, Arūną Vrubliauską

Straipsnis MPEG-2 transporto srauto dalinio šifravimo metodas

Konferencijos pirmininkas doc.dr. A.Lopata

Vilniaus universiteto Kauno humanitarinis fakultetas
Kaunas
2012

7.2.1 MPEG-2 transporto srauto dalinio šifravimo metodas

MPEG-2 TRANSPORTO SRAUTO DALINIO ŠIFRAVIMO METODAS

Vytautas Simanaitis[1], Agnius Liutkevičius[2], Arūnas Vrubliauskas[3], Egidijus Kazanavičius[4]

[1]*Kauno technologijos universitetas, Realaus laiko kompiuterių sistemų centras, Kaunas, Lietuva, vyt.sima@gmail.com*

[2]*Kauno technologijos universitetas, Realaus laiko kompiuterių sistemų centras, Kaunas, Lietuva, agnius@ifko.ktu.lt*

[3]*Kauno technologijos universitetas, Realaus laiko kompiuterių sistemų centras, Kaunas, Lietuva, aras@ifko.ktu.lt*

[4]*Kauno technologijos universitetas, Realaus laiko kompiuterių sistemų centras, Kaunas, Lietuva, ekaza@ifko.ktu.lt*

Santrauka (abstract). Video srauto šifravimas yra pagrindinis intelektinės nuosavybės užtikrinimo metodas šiuolaikinėse skaitmeninių teisių valdymo sistemose (DRM). Egzistuojantys šifravimo metodai užtikrina labai aukštą apsaugos lygį, bet tuo pat metu reikalauja labai didelių skaičiavimo resursų ir yra sunkiai pritaikomi ribotų skaičiavimo resursų mobiliems vartotojų įrenginiams. Šiame straipsnyje pristatomas dalinis MPEG-2 transporto srauto šifravimo metodas, kuris yra tinkamas tokiems ribotų resursų terminaliniams vartotojų įrenginiams. Siūlomo metodo eksperimentinio įvertinimo rezultatai rodo, kad šifravimo ir dešifravimo laikai, lyginant su standartiniu pilnu šifravimu, yra sumažinami apie 5 kartus, todėl šis metodas yra tinkamas ribotų skaičiavimo resursų dešifravimo įrenginiams.

Raktiniai žodžiai: video srauto šifravimas, ribotų resursų terminaliniai įrenginiai, MPEG-2 transporto srautas.

1 Įžanga

Pastaraisiais metais vis didėjanti intelektualinės nuosavybės problema kelia didelį susirūpinimą. Tai ypač svarbu skaitmeninio turinio platintojams, IPTV tiekėjams, nes jų verslas tiesiogiai priklauso nuo mokių klientų. Apsaugotas skaitmeninis turinys dažniausiai teikiamas naudojantis skaitmeninių teisių valdymo (DRM) sistemomis, kurios užtikrina tik autorizuotų vartotojų prieigą prie turinio. Vaizdo srauto apsaugos metodai dažnai naudoja AES šifravimo algoritmą, dėl jo greitaveikos ir patikimumo[2][3], taikomi ir kiti algoritmai, tokie kaip DES, XOR. Chaosu paremtas šifravimas pateiktas [12] straipsnyje Ne visi šifravimo metodai užtikrina vienodą saugumo lygį, o saugiausi metodai dažnai reikalauja didelių skaičiavimo resursų. Tai nėra problema, kai klientas naudoja personalinį kompiuterį ar panašų įrenginį. Problema iškyla, kai vartotojui turinys teikiamas į mažų skaičiavimo pajėgumų įrenginį, pvz. TV priedėlį, telefoną, PDA [14]. Šie įrenginiai nėra pajėgūs iššifruoti didelio duomenų srauto realiu laiku.

Šiame straipsnyje pateikiamas naujas vaizdo transliacijos apsaugos metodas, reikalaujantis mažiau skaičiavimo pajėgumų ir suderinamas su įrenginių vaizdo dekodavimo spartintuvais. Šis metodas buvo pirmą kartą pristatytas [1] publikacijoje, kur buvo įvertintas siūlomo metodo tinkamumas vaizdo transliacijoms, pasižyminčioms labiau statiniais (mažai kintančiais) vaizdais, kuriuos yra sunkiausia apsaugoti. Šiame straipsnyje įvertinamas siūlomo metodo tinkamumas video srauto transliacijoms, kurios sudarytos iš labiau kintančių vaizdų, kas teoriškai leidžia pasiekti dar geresnį apsaugos lygį dėl smarkiai besiskiriančių vaizdo kadrų. Metodas yra tinkamas mažai skaičiavimo resursų turintiems vartotojo įrenginiams (telefonai, PDA, STB ir pan.). Metodas gali būti pritaikytas šifruoti H.264/AVC, H.263, H.261 užkoduotą vaizdą. Siūlomas metodas užtikrina reikalingą saugumo lygį, bet yra spartesnis už pilną šifravimą, ir suderinamas su daugeliu DRM sistemų.

2 Panašūs darbai

Egzistuoja daug vaizdo apsaugos metodų naudojant pilną šifravimą, tokių kaip [4] ar [5], bet jie reikalauja didelių skaičiavimo resursų, ir gero tinklo pralaidumo. Pilnas transliacijos šifravimas dažniausiai remiasi SRTP ir SRTCP protokolais[6]. Pagrindinis privalumas šifruojant transliaciją yra tai, kad vaizdas gali būti koduotas bet kokiu kodavimo algoritmu, taip pat tai užtikrina aukštą saugumo lygį. Tokio metodo trūkumas yra resursai reikalingi vaizdui šifruoti ir iššifruoti. Siūlomi ir išrenkamojo šifravimo algoritmai, pvz. [7], kur užtikrinami 4 apsaugos lygmenys. Tačiau pirmųjų lygmenų apsauga yra nepatikima, o ketvirtajame lygmenyje šifruojami visi duomenys. Apsauga gali būti realizuojama ir viename iš H.264 kodavimo algoritmo žingsnių, Zig-Zag matricos skaityme [8]. Jeigu skaitymo metodas nėra žinomas, baitai eilutėje bus sudėlioti nežinoma tvarka, o dekodavimo įrenginys nežinodamas šios tvarkos negalės teisingai surinkti matricos tolesniam vaizdo dekodavimui. Tačiau taikant šį metodą, blogėja vaizdo suspaudimo laipsnis. Judesio vektorių šifravimo metodai pagrįsti tuo, kad vaizdas susideda iš kadrų, kadrai iš I, P ir B tipo MB (makro blokų), P arba B tipo makro blokai

turi poslinkio vektorių, kurie nusako kuri kadro dalis turėtų būti "nukopijuota" į atitinkamą MB. Poslinkio vektorių kodavimo algoritmas [9] susideda iš dviejų lygių: užšifruojami poslinkio vektoriai XOR algoritmu arba pasinaudojant kitu šifravimo algoritmu poslinkio vektoriai sumaišomi. Dalis apsaugos metodų naudoja DCT koeficientų šifravimą H.264 kodavimo žingsnyje, pvz. [10] pasiūlyta apsauga užtikrina tris saugumo lygius su skirtingais kiekiais šifruojamų koeficientų (1 lygis – 0-4 koeficientai, 2 lygis – 5-19, 3 lygis – likę 44 koeficientai). Naudojant šį metodą suspaudimo laipsnis neturėtų nukentėti, nes 0 koeficientai nėra šifruojami. Paprastesnis metodas pasiūlytas [11], kur šifruojami tik DCT koeficientų ženklai. Toks šifravimas nėra itin patikimas nes galima atspėti ženklą, kai kurie gali būti nepakitę. Dalinis šifravimas yra paprastas metodas pasiūlytas [13]. Šifruojama dalis I-kadro, kai vaizdas yra suspaustas H.263 metodu. Tačiau šifruojant tik I tipo blokus neužtikrinamas saugumas, nes vaizdas gali būti atstatytas iš poslinkio vektorių.

Išanalizuoti apsaugos metodai turi trūkumų, sudėtingas jų taikymas. Kai kurie reikalauja didelių skaičiavimo resursų, sunkei suderinami su egzistuojančiais sprendimais. Kelių metodų apsauga realizuota kodavimo lygmenyje, tai suteikia gera saugumo lygį, tačiau šifro rakto pakeitimas reikalauja perspausti vaizdą. Išanalizuoti metodai aprašo I, P arba B tipo blokų šifravimą, tačiau jų nustatymui reikalingi papildomi skaičiavimai. Šiame straipsnyje siūlome naują šifravimo metodą, kuris šifruoja ne kodavimo lygmenyje, bet transporto (MPEG-2 transporto srautas) lygmenyje. Siūlomas metodas duomenis šifruoja neatsižvelgdamas į jų tipą, todėl nereikalingi skaičiavimai I, P ar B bloko paieškai. Siūlomas metodas šifruoja transliacijos srautą, todėl norint jį retransliuoti pakanka peršifruoti šifruotas dalis.

3 Siūlomas metodas

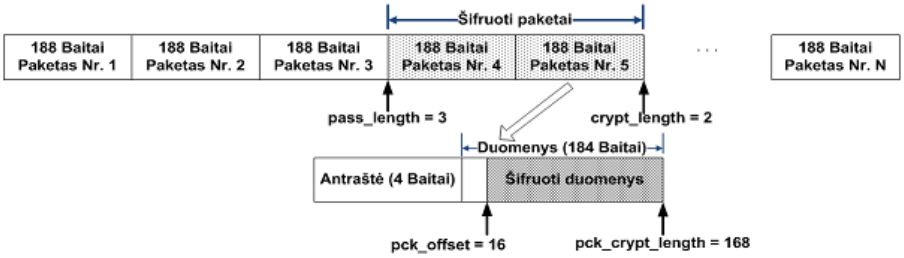
Siūlomas metodas šifruoja dalį MPEG-2 transporto srauto, kuris yra sudarytas iš 188B dydžio paketų. MPEG-2 srauto paketų surinkimo tvarka privalo būti teisinga, kitaip vaizdo dekodavimas nebus įmanomas, iš to seka, kad sumaišius paketus, ar dalį užšifravus, vaizdo atkurti nepavyks, nežinant algoritmo ar šifro raktų. Šis metodas taip pat būtų suderinamas su aparatiniu vaizdo dekodavimu, nes duomenys jau būtų dešifruoti prieš juos perduodant dekodavimo įrenginiui. Šifruojant duomenis pasirinktoje vietoje nereikalingas vaizdo perkodavimas, norint jį užšifruoti skirtingais metodais ar raktais. Pvz, naudojant transliacijos keletui klientų metodą (angl. multicast), visi klientai suskirstomi į grupes, siekiant sumažinti serverio apkrovas, o kiekvienai grupei transliuojamas vaizdas šifruojamas skirtingais šifro raktais.

MPEG-2 transporto srauto paketas susideda iš 4B antraštės ir 184B duomenų. Dėl suderinamumo antraštės nėra šifruojamos, siekiant sumažinti šifruojamų duomenų kiekį, bus šifruojama tik dalis duomenų paketo. Siūlomo metodo žingsniai:

1. Transliacijos pradžioje yra sukuriami saugūs šifravimo raktai AES algoritmui. Raktai klientui perduodami per DRM sistemą.
2. Nustatomi pradiniai, srauto sekos šifravimui naudojami 2 parametrai (žr. 1 pav.)
   - *pass_length* – nešifruojamų paketų sekos ilgis, po kurio seka *crypt_length*.
   - *crypt_length* – šifruojamų paketų sekos ilgis.

   Paketo duomenų šifravimui naudojami kiti 2 kintamieji, nusakantys šifruojamą dalį:

   - *pck_offset* – baitas nuo kuriuo reikės šifruoti duomenis pakete.
   - *pck_crypt_length* – baitų kiekis kuris turi būti užšifruotas. Turi būti 16 kartotinis naudojant AES šifravimą, kad šifruoti duomenys užimtų tiek pat vietos kaip ir nešifruoti.



1 pav. Siūlomo apsaugos metodo schema su pavyzdiniais parametrais

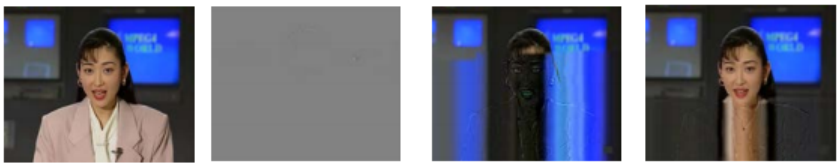
3. Jei paketas turi nustatytą „adaptation fields“ jis nėra šifruojamas.
4. Tikrinamos *pass_length* ir *crypt_length* reikšmes, jei paketą reikia šifruoti, užšifruojama jo dalis priklausomai nuo *pck_offset* ir *pck_crypt_length* parametrų. Šifruotam paketui transporto srauto paketo antraštėje nustatoma *scr* vėliavėlė į 1, tai reiškia kad paketas šifruotas.
5. Tikrinamas sekantis srauto paketas (3 punktas).

4 Eksperimento atlikimo metodika

Siūlomas dalinis MPEG-2 transliacijos srauto šifravimas įvertintas atliekant testus. Eksperimentai atlikti Ubuntu 10.10 operacinėje sistemoje, FitPC2 kompiuteryje su Intel Atom 500MHz procesoriumi. Paketams šifruoti ir dešifruoti parašyta metodą realizuojanti bandomoji programos versija. Šifruoto vaizdo kokybė vertinta skaičiuojant PSNR ir DSSIM. PSNR nusako santykį tarp gero ir blogo vaizdo lyginant taškas į tašką. DSSIM – struktūrinis nepanašumas, vaizdo kadrai yra lyginami sritimis 8x8 taškų. Šifravimo ir dešifravimo laikai matuoti naudojantis *perf* įrankiu. PSNR skaičiavimams naudojamas *ffmpeg* programinis paketas.

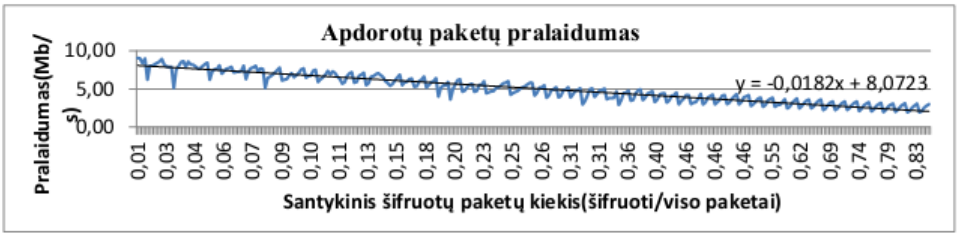
Apsaugos lygio įvertinimui atlikti matavimai, siekiant nustatyti ribą, kai apsauga nebeužtikrinama. Parinkti tokie bandymų parametrai: *pass_length* – 0, 1, 2, 3, 6, 9, 12, 15, 19, 24, 32, 64; *crypt_length* – 1, 2, 3, 6, 9, 12; *pck_crypt_length* – 16, 32, 64, 128, 176 (skaičiuojant nuo paketo pabaigos). Vertinamo vaizdo kokybė yra bloga (gerai apsaugota), jei PSNR reikšmė neviršija 15dB, arba DSSIM yra daugiau nei 0,5. Atliekant eksperimentą šifravimas ir iššifravimas atlikti 5 kartus, laiko reikšmė yra matavimų vidurkis. Užšifravimo ir iššifravimo laikai eksperimento metu nesiskyrė daugiau kaip 10%.

Nustatyta, kad šifravimo eigoje, praleidžiant 64 paketus ir pilnai šifruojant 12, reikiamas saugumo lygmuo nepasiekiamas, t.y. DSSIM reikšmės yra mažiau nei 0,5 (žr. 2 pav). Praleidžiant 32 ir pilnai šifruojant 1 paketą, matomi iškraipyti kontūrai. Vaizdo srautą pilnai šifruojant gaunama didžiausia apsauga, vaizdas visiškai nebeatkuriamas. Pilnai šifruojant vaizdo sraute kas 32 paketą, pasiekiamas 8,4Mb/s pralaidumas, t.y 5 kartus greičiau lyginant su pilno šifravimo pralaidumu kuris yra 1,7Mb/s.

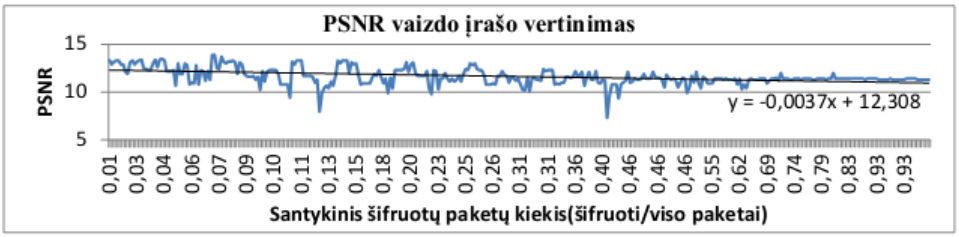


Nešifruotas kadras | Pilnas šifravimas | Praleisti 32, šifruotas 1 paketas (176B) | Praleisti 64, šifruoti 12 paketų (176B)

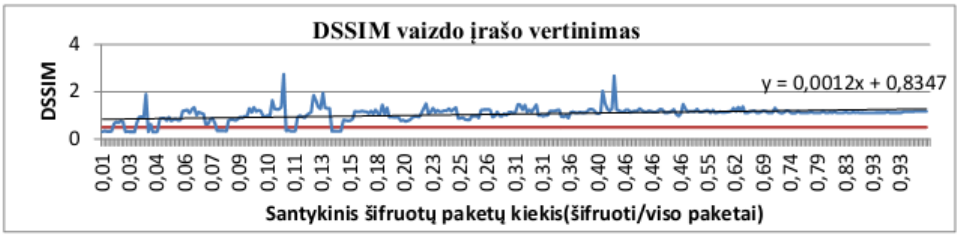
2 pav. Kadrai iš eksperimentui naudoto vaizdo įrašo



3 pav. Akiyo vaizdo įrašo apdorotų duomenų pralaidumo priklausomybė nuo šifruojamų duomenų kiekio



4 pav. Vaizdo įrašo kokybės kitimas pagal PSNR



5 pav. Vaizdo įrašo kokybės kitimas pagal DSSIM

3 grafike pateiktas duomenų pralaidumas transliaciją šifruojant siūlomu metodu. Taikant šį metodą pralaidumas padidėja priklausomai nuo norimo apsaugos lygio. 4 grafike pateiktas vaizdo vertinimas naudojant PSNR kadrų lyginimo metodą, atkuriamo vaizdo kokybė yra prasta, šifruojant net ir nedidelį duomenų kiekį.

Lyginant šifruotus ir orginalius kadrus DSSIM metodu, pastebime kelis taškus kurie nebeatitinka keliamo saugumo, kadrai yra bent dalinai atkurti.

Iš 4 ir 5 grafikų rezultatų nustatėme, kad jei visuose paketuose šifruosime po 16 iš 184B pasieksime adekvačią vizualinę vaizdo apsaugą. Šifruojant ne visus paketus, apsauga užtikrinama iki tol kol praleidžiamų paketų kiekis neviršija 32.

5 Išvados

Šiame straipsnyje pristatytas dalinis MPEG-2 srauto šifravimo algoritmas yra greitas ir efektyvus. Eksperimento metu nustatyta, kad sukurtas šifravimo metodas apie 5 kartus greitesnis už pilną MPEG-2 srauto šifravimą. Šifruojant kas 32 paketą sraute, pasiekiamas adekvatus vizualinis apsaugos lygis, kaip ir naudojant pilną šifravimą. Siekiant didesnio patikimumo apsaugant turinį galima šifruoti ir dažniau, kas 24 paketą, tai būtų 4,5 karto sparčiau nei pilnas šifravimas.

Literatūros sąrašas

[1] **Simanaitis, V., Liutkevičius, A., Vrubliauskas, A., Kazanavičius, E., Imbrasas, D**. Efficient MPEG-2 transport stream encryption method for low processing power mobile devices // Electronics and Electrical Engineering = Электроника и электротехника = Elektronika ir elektrotechnika. Kaunas : Technologija. ISSN 1392-1215. 2012, nr. 2(118).

[2] **Toldinas J., Štuikys V., Ziberkas G., Naunikas D**. Power Awareness Experiment for Crypto Service–Based Algorithms. Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 5(101). – P. 57–62.

[3] **Toldinas J., Stuikys V., Damasevicius R., Ziberkas G., Banionis M**. Energy Efficiency Comparison with Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices. Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 2(108). – P. 11–14.

[4] **Shin-Ho L., Han-Yen Y., Jia-Yen W., Jiann-Jone C., Jun-Lin L., De-Hui S**. A Secured Video Streaming System. International Conference on System Science and Engineering, 2010. –P. 625–630.

[5] **Nishimoto, Y., Imaizumi, H., Mita, N**. Integrated Digital Rights Management for Mobile TV Using Broadcasting and Communications. 14th Asia-Pacific Conference on Communications, APCC 2008. –P. 1–5.

[6] **Khalifa, N.E.-D.M., Elmahdy, H.N**. The Impact of Frame Rate on Securing Real Time Transmission of Video over IP Networks. International Conference on Networking and Media Convergence, ICNM 2009. –P. 57–63.

[7] **Tang L**. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. Proceedings of the ACM Multimedia, 1996. –P. 219–229.

[8] **Qiao L., Nahrstedt K**. Comparison of MPEG Encryption Algorithms. Computers & Graphics. 1998. –Vol. 22. – Issue 4. –P. 437–448.

[9] **Zheng L., Xue L**. Motion Vector Encryption in Multimedia Streaming. Proceedings of the 10th International Multimedia Modelling Conference, 2004. –P. 64-71.

[10] **Tosun, A.S., Feng, W.C**. Efficient Multi-layer Coding and Encryption of MPEG Video Streams. IEEE International Conference on Multimedia and Expo, ICME 2000. –Vol. 1. –P. 119–122.

[11] **Shiguo L., Zhongxuan L., Zhen R., Haila W**. Secure Advanced Video Coding Based on Selective Encryption Algorithms. IEEE Transactions on Consumer Electronics, 2006. –Vol. 52. –Issue 2. –P. 621–629.

[12] **Shiguo L., Jinsheng S., Zhiquan W., Yuewei D**. A Fast Video Encryption Scheme Based-on Chaos. 8th International Control, Automation, Robotics and Vision Conference, 2004. –Vol. 1. –P. 126–131.

[13] **Jeong-Hyun K., Yeon-Jeong J., Ki-Song Y**. Protection scheme for secure MPEG-2 streaming. IEEE International Conference on Multimedia and Expo, 2004. –Vol. 2. –P. 927–930.

[14] **Gunhce K., Dongkyoo S., Dongil S**. Intellectual property management on MPEG-4 video for hand-held device and mobile video streaming service. IEEE Transactions on Consumer Electronics, 2005. –Vol. 51. –Issue 1. –P. 139–143.

Partial Encryption Method for MPEG-2 Transport Stream

Video stream encryption is the main method for the protection of intellectual property in modern digital rights management systems (DRM). Existing encryption methods ensure very high protection level of encrypted content, but at the same time are very resources demanding, making them hardly suitable for low processing power mobile end-user devices. This paper presents partial MPEG-2 transport stream encryption method, which is suitable for such low power user terminals. The experimental evaluation of proposed method show that encryption and decryption times, comparing with standard full encryption, are reduced about 5 times, making proposed method a good choice for low processing power decryption devices.