KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Turinio apdorojimo sistemos (T120B161)

2 Laboratorinis darbas. Transliuojamo turinio apsaugos technologijos.

Atliko:

IFF-4/1 gr. studentas

Mangirdas Kazlauskas

2017 m. kovo 27 d.

Priėmė:

Lekt. V. Jukavičius

KAUNAS 2017

Turinys

[Įvadas 3](#_Toc478387438)

[Apie DRM 3](#_Toc478387439)

[Vaizdo įrašo paruošimas adaptyviam transliavimui su DRM 3](#_Toc478387440)

[Sistemos, skirtos turiniui atvaizduoti, kūrimas 5](#_Toc478387441)

[Testavimas 7](#_Toc478387442)

[Išvados 8](#_Toc478387443)

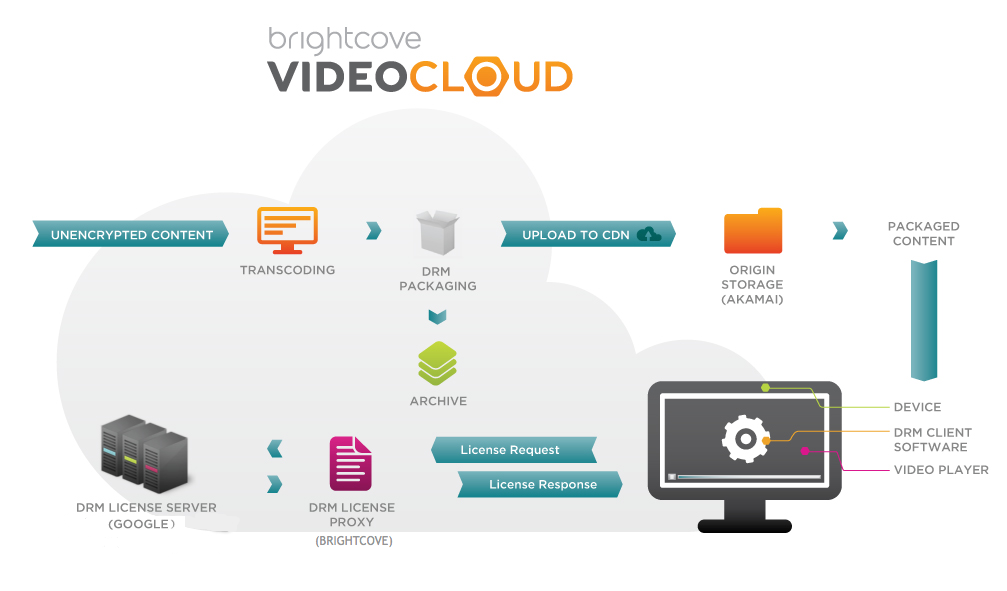
# Įvadas

Laboratorinio darbo tikslas – susipažinti su naujausiomis turinio apsaugos technologijomis ir realizuoti saugią turinio transliavimo sistemą naudojant internete surastomis priemonėmis.

# Apie DRM

DRM (angl. Digital Rights Management) – priemonė, skirta apsaugoti skaitmeninį turinį (pvz., vaizdo įrašus, kompiuterinius žaidimus) nuo neteisėto naudojimo ir platinimo.

Supaprastinta DRM schema pateikta pirmame paveikslėlyje:



pav. DRM veikimo schema (http://img.brightcove.com/node10073-drm-widevine.png)

Veikimo principas yra toks: pirmiausia skaitmeninis turinys, pavyzdžiui, vaizdo įrašas yra perkoduojamas į adaptyvaus transliavimo formatą bei patalpinamas serveryje. Nuo paprasto adaptyvaus turinio transliavimo skirtumas yra toks, kad perkoduojant vaizdo įrašą, jis kartu yra užšifruojamas naudojant raktus. Kai turinys yra užšifruotas bei perkoduotas (paruoštas peržiūrai), jis yra patalpinamas CDN tinkle, iš kurio vartotojas gali turinį pasiekti/peržiūrėti. Naudotojas savo turinio peržiūros įrenginyje (kompiuteryje, vaizdo grotuve, telefone) turi DRM leistuvą, kuris leidžia vartotojui peržiūrėti serveryje patalpintą turinį. Kai naudotojas nori paleisti ar peržiūrėti turinį, kliento pusėje esanti DRM programinį įrangą siunčia užklausą į licencijos serverį, kur patikrinama, ar naudotojas turi priėjimą prie turinio, kurį nori peržiūrėti. Jei gaunamas teigiamas atsakymas, DRM grotuvas atšifruoja užkoduotą turinį ir naudotojas gali jį peržiūrėti.

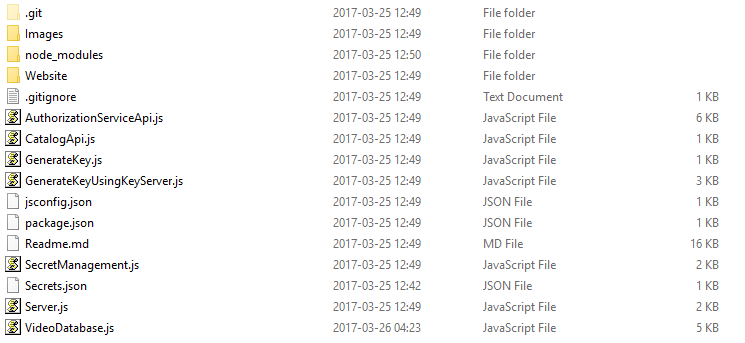
Internete galima rasti kelias DRM pagrindu sukurtas technologijas, tačiau šiam laboratoriniam darbui buvo pasirinkta naudoti Widevine. Šis sprendimas priimtas dėl kelių priežasčių:

1. Šią technologiją palaiko daug (virš 2 milijardų) įrenginių, svarbiausia, ją palaiko populiariausios interneto naršyklės, todėl ši technologija yra tinkama laboratoriniam darbui atlikti;
2. Kiek teko ieškoti internete, visi surasti DRM technologijai reikalingi licencijų serveriai yra mokami, rasta viena įmonė – Anixom, kuri testavimo reikmėms suteikia nemokamą prieigą prie licencijos serverių, o pačios įmonės programinė įranga taip pat remiasi Widevine technologija.

# Vaizdo įrašo paruošimas adaptyviam transliavimui su DRM

Vaizdo įrašo paruošimas, užkodavimas buvo atliktas remiantis pačios Anixom įmonės suteikiamu gidu: <https://github.com/Axinom/drm-quick-start>

Iš šio gido/repozitorijos parsisiunčiami failai, reikalingi failai, skirti vaizdo įrašo užkodavimui bei adaptyvaus turinio transliavimui, prieš tai turinį atšifruojant. Tam, kad parsisiųsti failai veiktų, pirmiausia kompiuteryje reikia įdiegti Node.js serverį, kuris parsiunčia visus reikalingus failus veikimui (darbiniame kataloge reikia įvykdyti konsolės komandą npm install). Darbinių failų struktūra pateikta antrame paveikslėlyje:



pav. Darbinių failų struktūra

Svarbiausi failai yra šie:

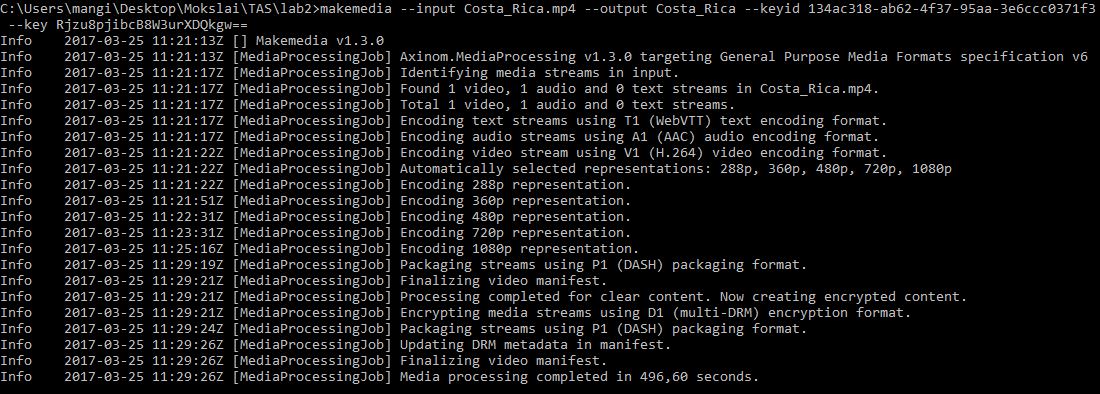
1. Server.js – sukuria HTTP serverį, kuris paleidžia svetainę bei autorizacijos servisą;
2. VideoDatabase.js – aprašo vaizdo įrašų sąrašą, kurį gali matyti naudotojas;
3. CatalogApi.js – leidžia puslapiui surinkti vaizdo įrašų failus, kurie gali būti atvaizduojami;
4. AuthorizationServiceApi.js – implementuoja autorizacijos API, kuris siunčia autorizacijos užklausas į licencijos serverį, iš kurio gauna žetonus, reikalingus rodomam turiniui atkoduoti;
5. GenerateKey.js – sugeneruoja atsitiktinį raktą, naudojama failo užkodavimui;
6. Secrets.json – šiame faile patalpinami iš Anixom įmonės (pasirašius sutartį) gauti raktai, reikalingi komunikavimui su licencijos serveriu.

Pirmiausia paruošime failą adaptyviam transliavimui. Pirmiausia reikia sugeneruoti raktus, reikalingus failo užkodavimui (3 pav.):

C:\Users\mangi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\NodeGenerateKey.png

pav. Sugeneruoti raktai

Tada naudojant Makemedia programinį įrangą (gauta iš įmonės Anixom kartu su raktais prisijungimui prie licencijos serverio) failas yra užkoduojamas ir kartu paruošiamas adaptyviam transliavimui - paruošiami skirtingų kokybių vaizdo įrašai, suskaidyti į segmentus (4 pav.):

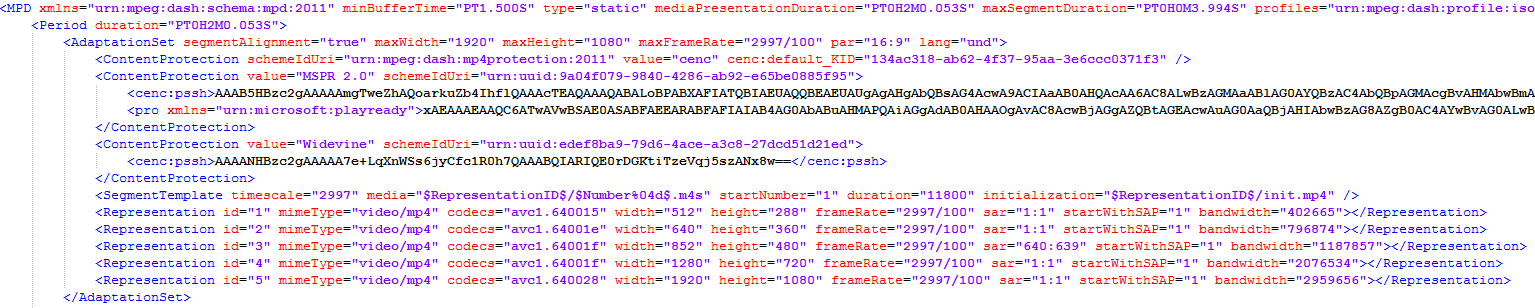


pav. Failo paruošimas

Komandos makemedia parametrų paaiškinimas:

* --input - vaizdo įrašas, kurį reikia perkoduoti;
* --output – vieta (katalogo pavadinimas), kurioje bus patalpinti perkoduoti failai;
* --keyid – anksčiau sugeneruoto rakto identifikacinis numeris;
* --key – anksčiau sugeneruotas raktas;

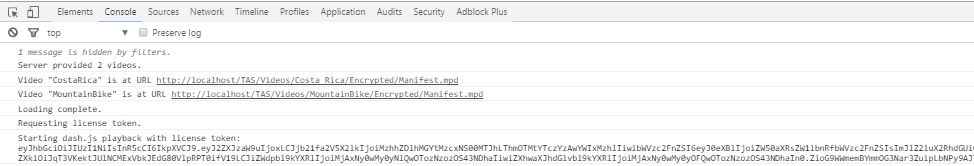
Įvykdžius šią komandą, gautas katalogas, kuriame patalpinti skirtingų kokybių vaizdo įrašo failai bei manifestinis (MPD) failas (5 pav.):



pav. Sugeneruotas MPD failas

Kaip matome, šis MPD failas skiriasi nuo to manifestinio failo, kuris buvo gautas pirmajame laboratoriniame darbe, t.y., šiame faile papildomai atsirado elementai <ContentProtection>, kurie apibrėžia, kad vaizdo įrašas yra užšifruotas.

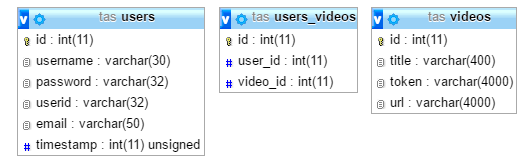
Kai failas paruoštas, Node.js pagalba paleidžiamas failas Server.js, kuris paleidžia testinę svetainę, bet kartu ir sugeneruoja žetonus vaizdo įrašams, reikalingus turiniui atšifruoti (6 pav.):



pav. Konsolės lange išvedama informacija apie gautą žetoną, reikalingą vaizdo įrašui atšifruoti

# Sistemos, skirtos turiniui atvaizduoti, kūrimas

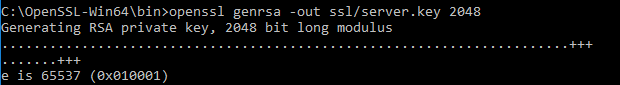
Sistema yra sukurta tokiu principu: naudotojai yra autentifikuojami prisijungdami prie sistemos. Duomenų bazėje saugoma tokia informacija (7 pav.):



pav. Duomenų bazės lentelės

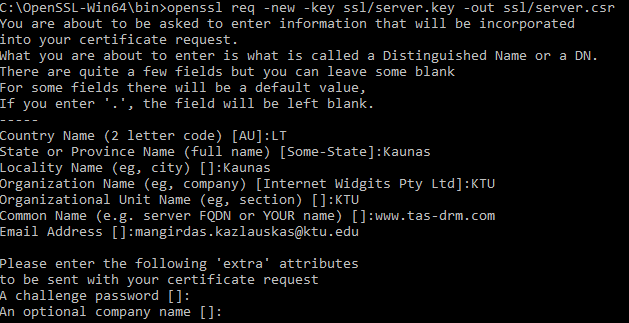
Lentelėje „users“ yra saugoma informacija apie sistemos naudotojus (jų el. paštas, prisijungimo vardas, slaptažodis, sesijos identifikacinis numeris bei laikas, kada jis paskutinį kartą prisijungė prie sistemos). Lentelėje „videos“ yra saugoma informacija apie kiekvieną sistemoje esantį vaizdo įrašą: jo pavadinimas, žetonas, reikalingas turiniui atšifruoti (pagal jį siunčiamos užklausos į licencijos serverį) bei url adresas, kur yra patalpinti vaizdo įrašo manifestinis (MPD) failas. Lentelė „users\_videos“ yra skirta susieti naudotojus ir vaizdo įrašus, tai yra, koks vartotojas kokį vaizdo įrašą gali peržiūrėti sistemoje. Sistema kuriama lokaliame serveryje, sistemos url adresas: [www.tas-drm.com](http://www.tas-drm.com).

Saugios vartotojo ir sistemos sąsajos realizavimui taip pat sukurtas SSL sertifikatas, leidžiantis prie sistemos jungtis per HTTPS. Sertifikatas nebuvo pasirašytas jokios CA organizacijos, todėl jis nėra patvirtintas, bet kadangi serveris yra lokalus, tai laboratorinio darbo atlikimui užteks ir „self-signed“ sertifikato. SSL sertifikato kūrimui buvo naudojama OpenSSL programinė įranga, leidžianti sukurti „self-signed“ sertifikatą. Pirmiausia sukuriamas RSA privatus raktas (8 pav.):



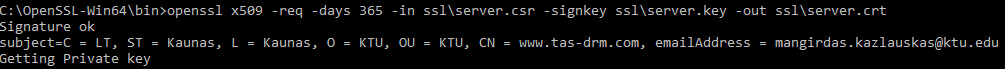
pav. Privataus rakto sukūrimas

Tada sugeneruojamas sertifikatas (9 pav.):



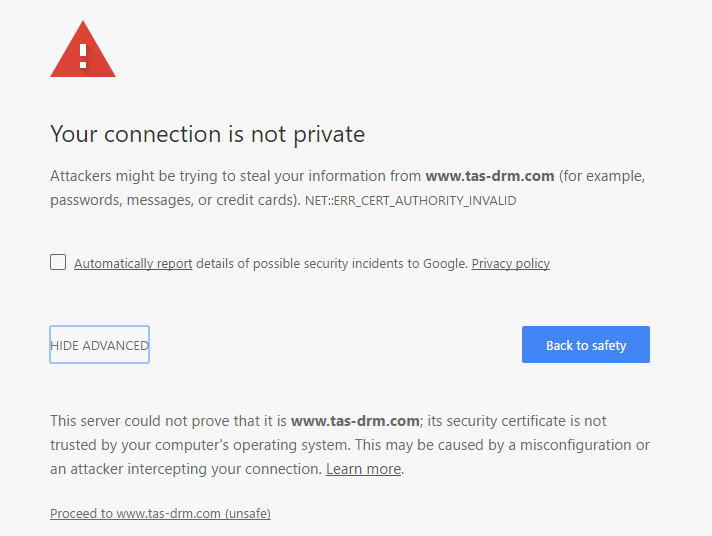
pav. Sertifikato generavimas

Vėliau sertifikatas pasirašomas (10 pav.):



pav. Self-signed sertifikato pasirašymas

Sertifikatas yra įdiegiamas į naudojamą lokalų serverį, o tada serveris yra perkraunamas. Persikrovus serveriui, galima mėginti jungtis prie svetainės jau anksčiau minėtu URL adresu. Tačiau pirmą kartą jungiantis prie svetainės pranešimas perspėja, kad puslapis yra nepatikimas – taip yra dėl to, kad sertifikatas nėra patvirtintas jokios CA organizacijos (11 pav.). Pasirenkame proceed to www.tas-drm.com (unsafe).



pav. Pranešimas, įspėjantis apie galimai pavojingą svetainę

Prisijungus prie svetainės rodoma, kad ji nėra saugi (12 pav.):



pav. Įspėjimas rodo, kad svetainė nėra saugi

Kad to išvengtume, reikia prie naršyklės pridėti mūsų sukurtą ir pasirašytą sertifikatą. Kad tai padarytumėm (Google Chrome naršyklėje) pasirenkame Settings -> Show advanced settings -> HTTPS/SSL -> Manage certificates -> Trusted Root Certification Authorities -> Import.

Pasirenkame mūsu sukurtą sertifikatą, išsaugome ir perkrauname naršyklę. Dabar matome, kad svetainė laikoma saugia (13 pav.):



pav. Naršyklė parodo, kad svetainė yra saugi

Pati sistema yra sukurta naudojant PHP programavimo kalbą, svetainės išvaizdos kūrimui naudotas Bootstrap karkasas, DRM vaizdo įrašams transliuoti naudojamas dash.js (14 pav.) grotuvas.



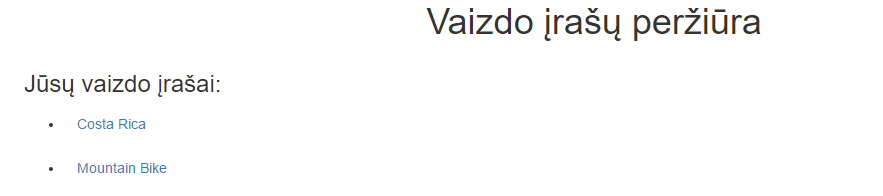
pav. Dash.js grotuvo kodas

Veikimo principas: kiekvieną kartą naudotojui paspaudus ant vaizdo įrašo nuorodos, siunčiama asinchroninė užklausa (AJAX) į serverį, kur iš duomenų bazės grąžinami duomenys apie vaizdo įrašą – nuoroda į vaizdo įrašo MPD failą bei žetonas, reikalingas siųsti užklausas į licencijos serverį. Jei asinchroninė užklausa sėkminga – dash.js grotuvui priskiriamas vaizdo įrašas leidimui bei priskiriamas žetonas siunčiamoms užklausoms į licencijos serverį.

Vėliau paruošti vaizdo įrašų failai yra sukeliami į lokalų serverį ir pradedamas sistemos testavimas.

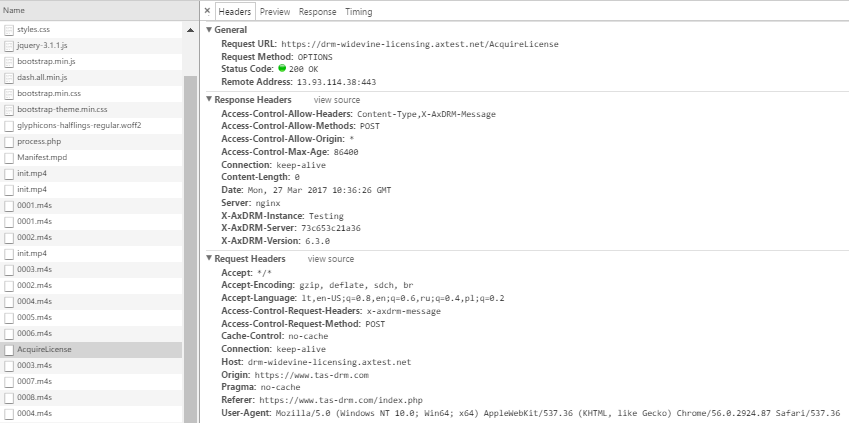
# Testavimas

Sistemos testavimui paruošiami du vaizdo įrašų failai: „Costa Rica“ ir „Mountain Bike“. Jų informacija yra patalpinama duomenų bazėje (failų žetonai ir nuorodos į MPD failus). Taip pat sukuriami 3 skirtingi naudotojai: Lankytojas1, Lankytojas2, VIPLankytojas. Visiems sistemos naudotojams priskirti jiems galimi peržiūrėti vaizdo įrašų failai: Lankytojas1 gali peržiūrėti vaizdo įrašą „Costa Rica“, Lankytojas2 – „Mountain Bike“, o VIPLankytojas – abu vaizdo įrašus. Tarkime, prisijungėme kaip VIPLankytojas. Prisijungęs prie sistemos jis mato tokį vaizdą (15 pav.):



pav. Galimų peržiūrėti vaizdo įrašų sąrašas

Pasirinkus vaizdo įrašą, jis pradedamas rodyti. 16 paveikslėlyje matome, kad vaizdo įrašas yra užkraunamas segmentais, priklausomai nuo vaizdo įrašo kokybės, kuri parenkama pagal naudotojo interneto spartą. Taip pat matome, kad papildomai siunčiamos užklausos į licencijos serverį, kurios tikrina, ar naudotojas turi teisę peržiūrėti vaizdo įrašą. Jei gaunamas teigiamas atsakymas, turinys yra atšifruojamas ir vaizdo įrašas leistuve yra rodomas:



pav. Užklausa į licencijos serverį

# Išvados

Kaip matome iš testavimo rezultatų, pavyko realizuoti saugią sistemą, kuri leidžia vartotojams peržiūrėti jiems priskirtus vaizdo įrašus. Pats vaizdo įrašų transliavimas yra adaptyvus, o vaizdo įrašų segmentai prieš juos rodant yra atkoduojami naudojant vaizdo įrašui sugeneruotą žetoną bei siunčiant užklausas į licencijos serverį. Iš rezultatų bei siunčiamų užklausų matome, kad komunikacija vykdoma sėkmingai, vaizdo įrašas transliuojamas. Iš to galime daryti išvadą, kad sėkmingai realizavome DRM principu sukurtą sistemą, leidžiančią ne tik transliuoti adaptyvų turinį, bet prieš tai jį ir užkoduoti, taip apsaugant turinį nuo nelegalaus jo platinimo.