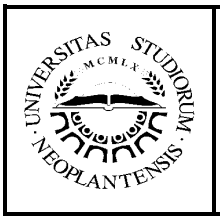


УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: **Миленко Максић**
Број индекса: **РА11-2016**

Тема рада: **Имплементација генератора *Widevine* порука за контролу дозволе за приказ садржаја дигиталне телевизије**

Ментор рада: **Проф. Др Илија Башичевић**

Нови Сад, април, 2021.

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:		
Идентификациони број, **ИБР**:		
Тип документације, **ТД**:		Монографска документација
Тип записа, **ТЗ**:		Текстуални штампани материјал
Врста рада, **ВР**:		Завршни (Bachelor) рад
Аутор, **АУ**:		Миленко Максић
Ментор, **МН**:		Проф. Др Илија Башичевић
Наслов рада, **НР**:		Имплементација генератора *Widevine* порука за контролу дозволе за приказ садржаја дигиталне телевизије
Језик публикације, **ЈП**:		Српски / латиница
Језик извода, **ЈИ**:		Српски
Земља публиковања, **ЗП**:		Република Србија
Уже географско подручје, **УГП**:		Војводина
Година, **ГО**:		2021.
Издавач, **ИЗ**:		Ауторски репринт
Место и адреса, **МА**:		Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)		
Научна област, **НО**:		Електротехника и рачунарство
Научна дисциплина, **НД**:		Рачунарска техника
Предметна одредница/Кqучне речи, **ПО**:		Дигитална телевизија, *CAS*
УДК		
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, **ВН**:		
Извод, **ИЗ**:		У овом раду је приказано једно рјешење имплементације генератора *ECM* порука, односно порука задужених за контролу дозволе за приказ садржаја дигиталне телевизије. Представљен је један пут током ког се садржај дигиталне телевизије најприје скремблује, и као такав штити од корисника који се нису претплатили на исти. Исто тако, приказан је и обрнути процес, дескрембловање, захваљујући ком корисници претплаћени на поменути садржај, могу без проблема да га и гледају.
Датум прихватања теме, **ДП**:		
Датум одбране, **ДО**:		
Чланови комисије, **КО**:	Председник:	Проф. Др Иван Каштелан
	Члан:	Проф. Др Мирослав Поповић
	Члан, ментор:	Проф. Др Илија Башичевић

Потпис ментора

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:			
Identification number, **INO**:			
Document type, **DT**:		Monographic publication	
Type of record, **TR**:		Textual printed material	
Contents code, **CC**:		Bachelor Thesis	
Author, **AU**:		Milenko Maksić	
Mentor, **MN**:		Prof. Dr. Ilija Bašičević	
Title, **TI**:		Implementation of Widevine message generator for control of digital television content display	
Language of text, **LT**:		Serbian	
Language of abstract, **LA**:		Serbian	
Country of publication, **CP**:		Republic of Serbia	
Locality of publication, **LP**:		Vojvodina	
Publication year, **PY**:		2021.	
Publisher, **PB**:		Author's reprint	
Publication place, **PP**:		Novi Sad, Dositeja Obradovica sq. 6	
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)			
Scientific field, **SF**:		Electrical Engineering	
Scientific discipline, **SD**:		Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, **S/KW**:		Digital television, *CAS*	
UC			
Holding data, **HD**:		The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, **N**:			
Abstract, **AB**:		In this paper we present one solution for implementation of ECM message generator, which are messages used for digital television content display control. One pathway during which the digital television content is first scrambled, so that it is protected from non – subscribed users, is shown. Also, the reverse process, descrambling, is shown thanks to which users subscribed to the said content are able to see it without problem.	
Accepted by the Scientific Board on, **ASB**:			
Defended on, **DE**:			
Defended Board, **DB**:	President:	Prof. Dr. Ivan Kaštelan	
	Member:	Prof. Dr. Miroslav Popović	Menthor's sign
	Member, Mentor:	Prof. Dr. Ilija Bašičević	

Захвалност

Велику захвалност за помоћ при изради документације везане за дипломски рад, дугујем свом академском ментору, Проф. Др Илији Башичевићу. Такође, огромна захвалност мом техничком ментору, Небојши Кошутићу, који ми је био на располагању током практичне израде дипломског рада, али и свим колегама из *iWedia* групе. Наравно, велику захвалност дугујем својој породици и пријатељима, на пруженој подршци током израде овог рада, али и током цијелокупног школовања.

САДРЖАЈ

Списак слика

Скраћенице

EMM – Entitlement Management Message, порука о управљању правима приступа

ECM – Entitlement Control Message, порука о контроли права приступа

TS – Transport Stream, преносни ток

PID – Packet ID, идентификатор пакета

ES – Elementary Stream, елементарни ток података

PES – Packetized Elementary Stream, пакетизовани елементарни ток података

PCR – Program Clock Reference, референца програмског сата

PAT – Program Association Table, табела удруживања програма

PMT – Program Map Table, табела са мапом програма

SCS – Simulcrypt Synchronizer, протокол за размјену порука

1. Увод

Телевизија представља комуникациони медијум за слање и пријем покретних слика и звука. Ријеч телевизија настала је од грчке ријечи *tele*(далеко), и латинске *video*(гледати). Дакле, намјена телевизије јесте омогућавање увида у догађаје који нису у нашем непосредном видику.

Реализација телевизије данас је проблем који обухвата разнородне културно-умјетничке, производно-техничке, технолошко-инжењерске, економско-социјалне и правне аспекте.

Рачунарски системи и намјенски уређаји учествују у различитим сегментима телевизије, почев од снимања садржаја, његовог енкодовања, транскодовања, мултиплексирања, модулације, демодулације, демултиплексирања, декодовања и репродукције на екранима уређаја код гледалаца. Ови уређаји, на страни производње и емисионе технике, обухватају информационе системе, дигиталну аудио/видео технику (камере, микрофоне, миксете, алате за уређивање садржаја), транскодере, дигиталне модулаторе. На страни корисника, уређаје чини опрема за пријем и репродукцију: антенски систем, појачавач, дигитални ТВ пријемник.

Суштина области рачунарског инжењерства је да посматра софтвер и хардвер као неодвојиве цијелине. Телевизијски пријемници су управо уређаји од чије хардверске архитектуре кључно зависи функционалност уређаја.

Хардвер и софтвер ТВ пријемника посвећени су имплементацији алгоритама обраде сигнала по посебно дефинисаним стандардима. Такође, аудио и видео декодовање суштински је везано за концепт улазног сигнала у телевизијски пријемник, његове битске

брзине, пакетизације и композиције. Овај улазни сигнал, осим што преноси аудио и видео садржаје, преноси и додатне садржаје који су за њих везани (преводи, телетекст, апликације, програмске шеме и сл.), те кључне метаподатке који су неопходни за разврставање аудио-видеа и осталих садржаја који припадају појединачним сервисима (ТВ програмима), односно метаподатака који идентификују формате садржаја.

Скремблoвање аудио – видео садржаја, односно скремблoвање канала представља планирани прекид сигнала на одређеним ТВ станицама у току емитовања одређених садржаја. Кабловски оператери су дужни да на захтјев Републичке радиодифузне агенције, имаоца права или емитера, телевизије која откупи ексклузивна права за емитовање одређеног спортског догађаја за територију Србије, скремблује стране канале који такође емитују тај догађај. Забране емитовања се осим за мечеве, односе и на друге лиценциране емисије, јер садрже материјал који не смије да се користи ван плаћене територије. Закључак је да нас и закон на неки начин обавезује да се приступ одређеном аудио-видео садржају ограничи, односно да се тај садржај скремблује.

2. Теоријске основе

У овом поглављу су дате теоријске основе на којима је овај рад заснован. Биће ријечи о *broadcast* преносу података, мултиплекс формату, као и уопштено о *Widevine*-у.

2.1 Broadcast

Уређаји за пријем ТВ садржаја разликују се према типу ТВ сигнала који примају, конфигурацији излазних спрега у смислу репродукције садржаја корисницима, као и формату, односно, генерацији стандарда преноса који подржавају. У зависности од карактеристика одређених тим наведеним разликама, одређује се и тип уређаја за пријем ТВ садржаја.

У зависности од типа сигнала, уређаји се могу подијелити на:

1. земаљске(терестричке) пријемнике;
2. сателитске пријемнике;
3. кабловске пријемнике;
4. хибридне пријемнике и
5. ИП-засноване пријемнике.

Земаљски, сателитски и кабловски пријемници спадају у емисионе (eng. *broadcast*) пријемнике, с обзиром на емисиону (једносмјерну) природу испоруке ТВ сигнала.

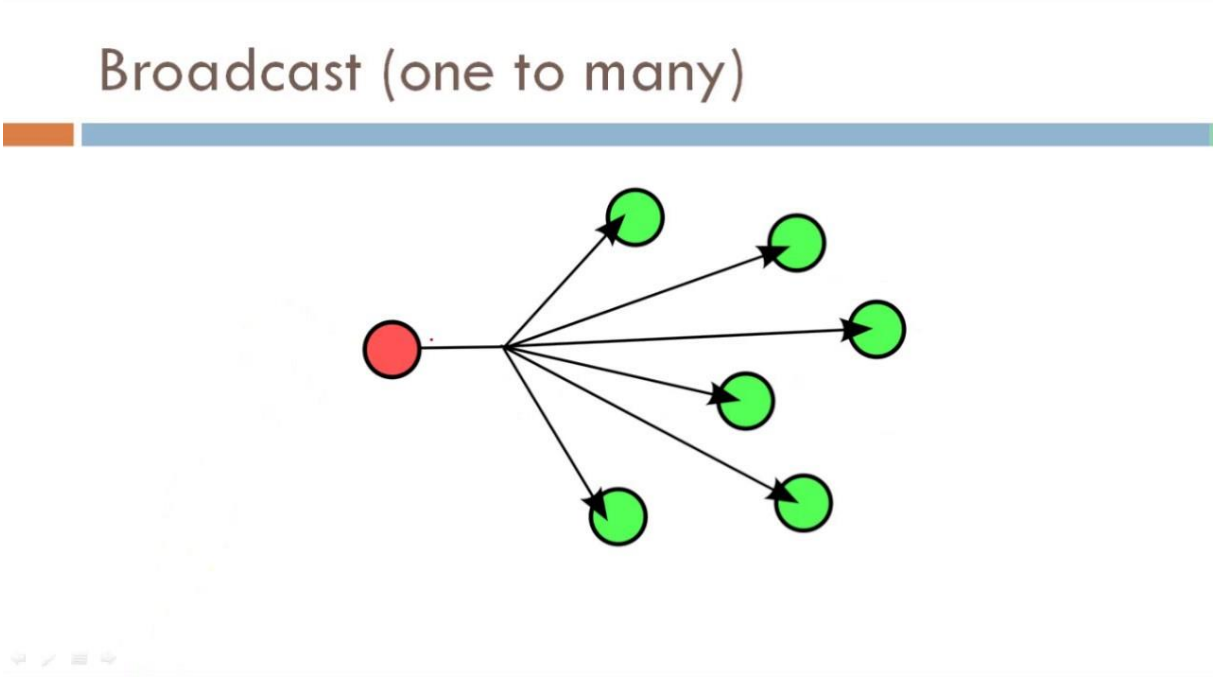
Broadcast пренос се још назива и дифузни пренос. Иста информација (порука) се шаље до свих корисника који користе исти спектар.

Осим тога што се користи у телевизији, као и у радиодифузији, користи се и у мрежама које за циљ имају да обавјесте или алармирају ширу популацију о неком догађају, као на примјер о доласку урагана или торнада.

Називају се још и мреже са неусмјереним емитовањем. Имају један комуникациони канал који користе сви рачунари у мрежи. Кратке поруке, зване пакети, које шаље један рачунар примају сви остали рачунари.

У адресном пољу пакета назначава се коме је он намјењен. По пријему пакета, рачунар провјерава адресно поље, и ако је пакет намјењен њему, прихвата га и обрађује, а ако није, онда га одбацује.

Broadcasting функционише тако што рачунар који шаље пакет уписује у његовом адресном пољу одговарајући код на основу ког сви рачунари у мрежи знају да је пакет њима упућен па га прихватају и обрађују.



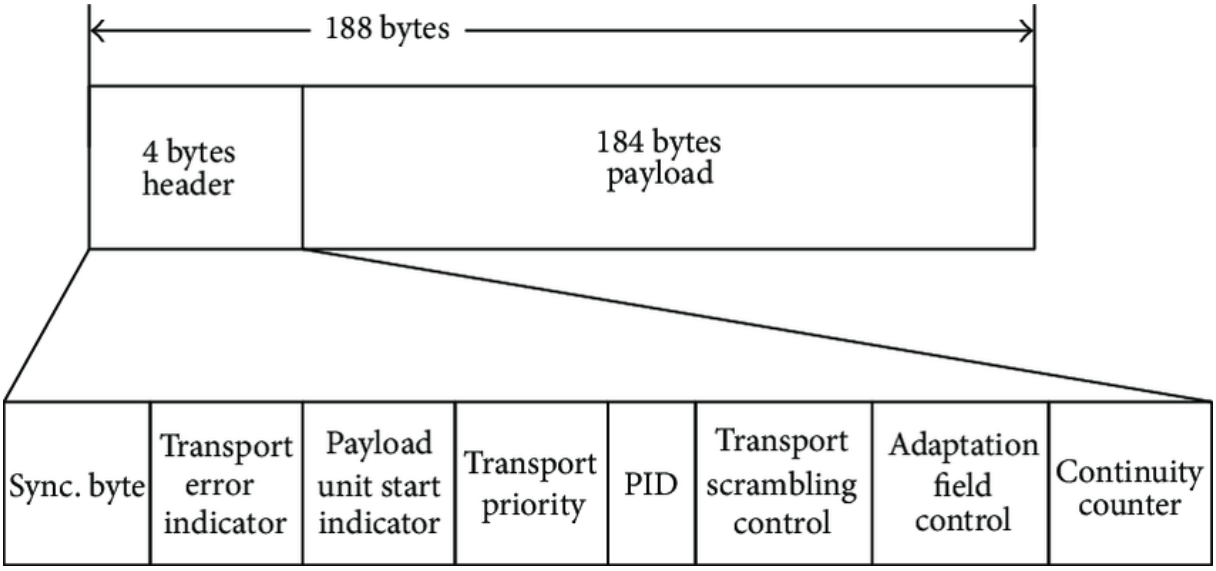
Слика 1: *Broadcast* пренос

2.2 Мултиплекс формат

У телекомуникацијским и рачунарским мрежама, мултиплексирање је метода којом се више аналогних или дигиталних сигнала комбинује у један, преко заједничког медијума за пренос. Циљ је да се ресури, који су релативно оскудни, искористе на што бољи могући начин.

Примјера ради, у телефонској комуникацији, могуће је истовремено обављати више телефонских позива преко једне заједничке телефонске жице. Мултиплексирани сигнал се преноси преко комуникационог канала, попут кабла.

Мултиплексирањем се капацитет комуникационог канала дијели на неколико логичких цијелина, од којих је свака задужена за одређени сигнал поруке, или ток података који се преносе

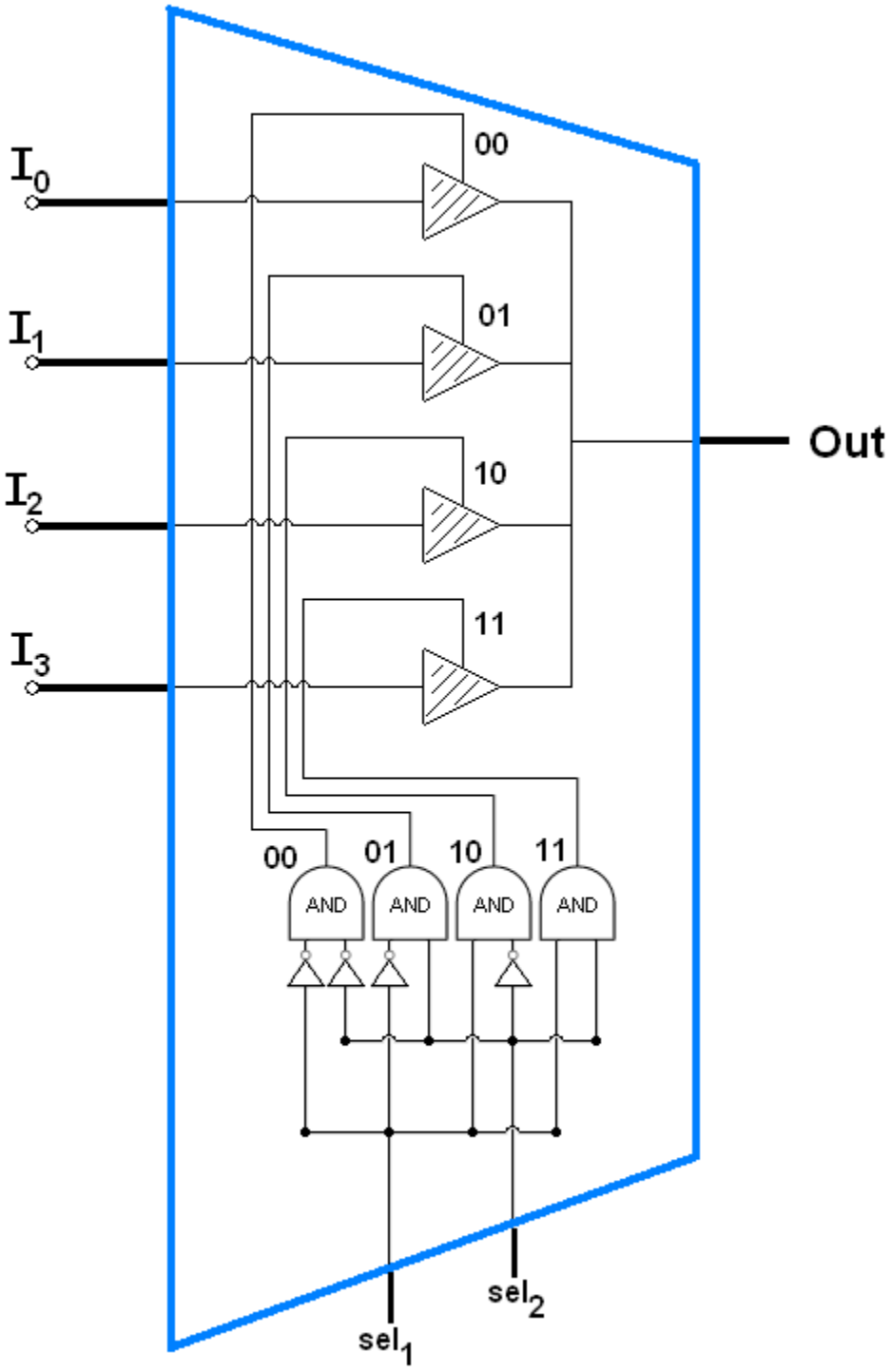


Слика 2: Пакет транспортног тока података

На слици 2, приказан је формат једног пакета који описује транспортни ток података. Јасно се види да је пакет величине 188 бајтова, и то спакованих тако да прва 4 бајта представљају заглавље пакета, а преосталих 184 бајта су намјењена преносу података који се односе на сам аудио – видео садржај. Карактеристично са један овакав пакет је да по правилу почиње са синхронизационим бајтом, чија је вриједност 0x47.

Оваква структура пакета добија се на начин да се потребни подаци који се шаљу провуку кроз компоненту која се назива мултиплексер. Задатак мултиплексера је да на одговарајући начин сложи пакет који се шаље.

На пријемној страни се налази демултиплексер, који ради по обрнутој логици у односу на мултиплексер, односно демултиплексер треба да разложи један пакет на податке које је мултиплексер објединио у пакет за слање.

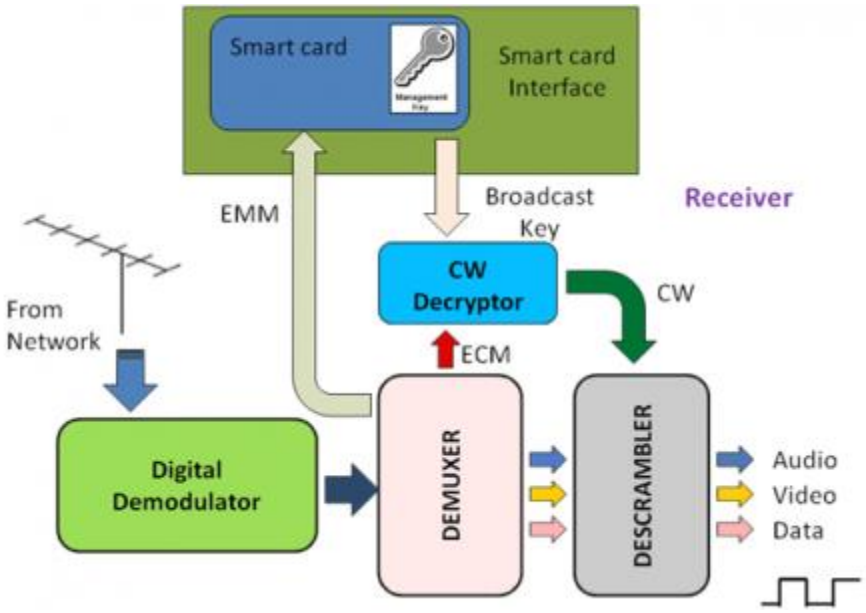


Слика 3: Мултиплексер

2.3 Поступак декрипције аудио – видео садржаја

Подаци који се добијају преко пакета транспортног тока података могу да буду шифровани, односно енкриптовани(заштићени). У том случају, приступ садржају је ограничен.

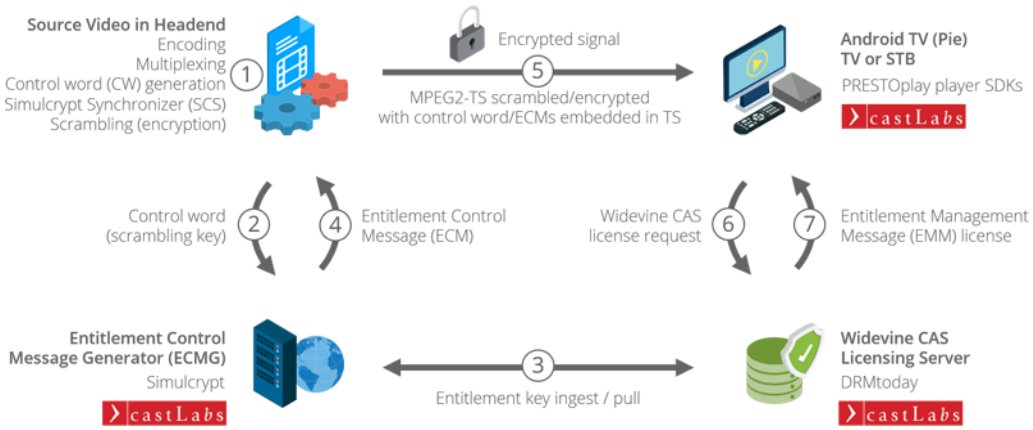
У оквиру сет-топ бокс уређаја, најприје се добавља сервисни кључ садржан у оквиру *EMM* поруке, тако што се исти декриптује коришћењем корисничког кључа који је достављен у облику смарт картице (или ауторизацијом уређаја од стране корисничког сервиса оператера). Даље, сервисни кључ се користи за декриптовање *ECM* поруке да би се издвојила контролна ријеч којом се коначно иницијализује дескремблер. Када је дескремблер иницијализован, могуће је обавити његово дескрембловање, односно његово дешифровање(декрипцију), и на та начин добити приказ аудио – видео садржаја.



Слика 4: Процес дескрембловања садржаја добијеног са антене

2.4 Widevine

Widevine технологија се користи за управљање дигиталним правима, а у употреби је код уређаја широке потрошње из области електронике.
Ова технологија подржава разне шеме шифровања и безбједности хардвера како би сигурно дистрибуирали видео садржај на потрошачке уређаје према правилима која дефинишу власници садржаја.



Слика 5: *Widevine CAS*

Widevine CAS ради само на дигиталним ТВ пријемницима са андроид оперативним системом, и то од верзије *Android 9 "Pie"*. Алгоритми за скремблование аудио – видео садржаја кој су подржани од стране *Widevine CAS* су: *AES -128 CTR, CBC DVB* или *CSAv2*.

Google је 2010. године откупио *Widevine*. *Widevine CAS* је последња технологија развијена од стране *Google*-а, а тиче се заштите при преносу садржаја, тако да је веома брзо своју примјену нашла и у дигиталној телевизији. Традиционално коришћење система са условним приступом, *CAS* системи, за пренос садржаја у дигиталној телевизији, представљали су веома скупу инвестицију. Почетни трошкови били су веома високи, који су се заснивали на ослањању на заштићене технологије, додатне надокнаде потребне за добављање лиценци намјењених уређајима, као и трошкове одржавања

компликоване инфраструктуре. Управо из тог разлога је и настала *Widevine CAS* технологија, која смањује свеукупне трошкове, али и даље гарантује безбједност садржаја који се преноси путем дигиталне телевизије.

2.5 *TSDuck* програмска подршка

TSDuck представља програмску подршку која се користи у дигиталној телевизији. Чини је сет алата који се користе за разне манипулације са *MPEG* преносним токовима. Под поменутим манипулацијама, подразумјевамо тестирање и надгледање преносног тока података. Такође, користи се и за интеграцију нових софтверских компоненти у систем, као и за њихово тестирање, и отклањање евентуалних грешака.

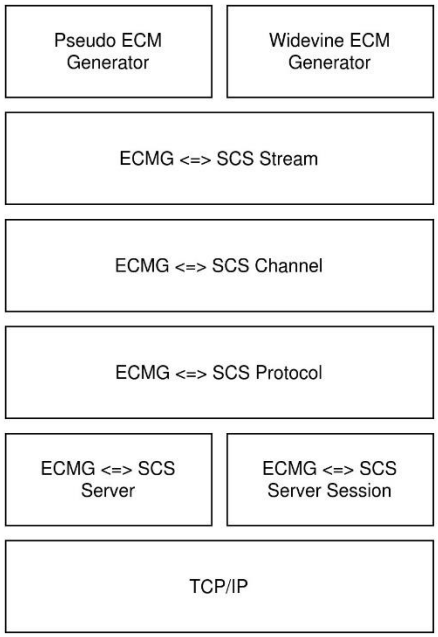
Овај алат је доступан за следеће оперативне системе: *Linux, Window* као и *macOS.* Имплементација алата је рађена у програмском језику *C++*. Архитектура *TSDuck* програмске подршке је модуларног карактера, што значи да се на релативно лак начин, нова компонента која има одређену функционалност, може интегрисати у систем, тестирати, и у складу са потребама корисника, додатно оптимизовати, и прилагођавати крајњој употреби.

3. Концепт решења

У овом поглављу биће описан идејни план по ком се постепено долазило до финалног рјешења.

3.1 Модуларни приказ рјешења

Генератор вајдвајн порука је компонента која се састоји од више модула. Први корак при прављењу концепта рјешења, била је скица архитектуре генератора.



Слика 6: Архитектура *ECM* генератора

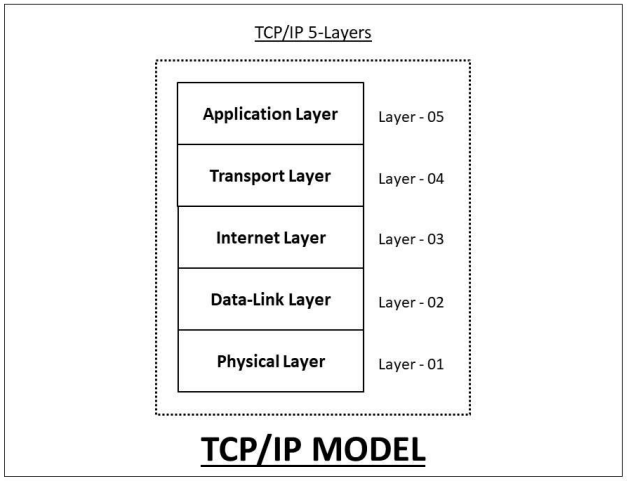
Са слике се види да је архитектура подијељена на 6 нивоа. Идеја је била да се са имплементацијом крене од најнижег нивоа, и да се сваки ниво приказане архитектуре имплементира независно један од другог, али да међусобно буду компатибилни, и да на крају када се обједине представљају једну функционалну цјелину.

Имплементација је могла да крене и од врха архитектуре, али би у том случају тестирање појединачних компоненти било комплексније, и било би много више простора за евентуалне грешке везане за имплементацију и функционалност цјелокупног система. У склопу поглавља која слиједе, детаљно ће бити објашњена наведена архитектура, односно програмски стек, и то од дна ка врху, баш како је и имплементација рађена.

3.2 Сервер и сесија

У комуникацији на релацији генератор-скремблер, генератор представља серверску страну. На основу *TCP/IP* протокола, реализована је серверска страна, која константно ослушкује клијентску страну, и чека захтјев за успостављање везе. Након што од клијента добије захтјев за успостављање везе, сервер га прихвата, и веза је успостављена.

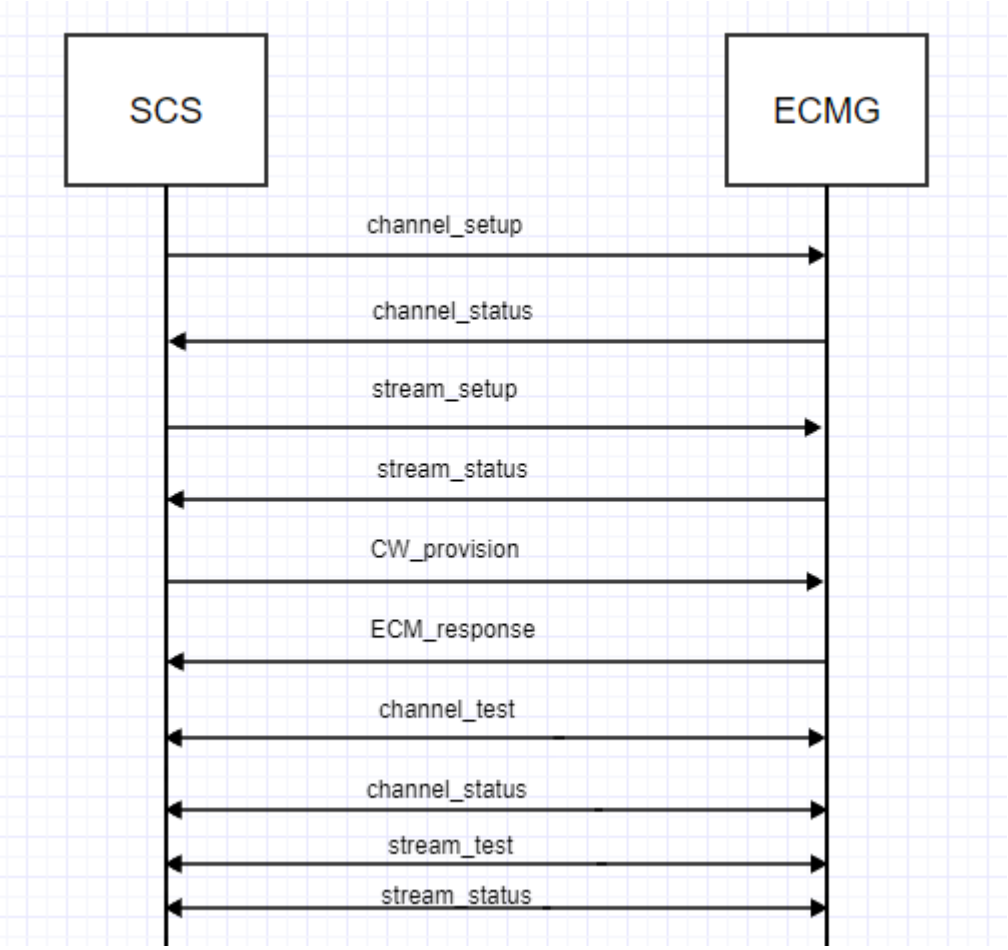
По успостављању везе, потребно је креирати сесију, која ће бити задужена за размјену порука између сервера и клијента током трајања успостављене везе. Када је веза успостављена, у сваком тренутку могуће је зауставити активну сесију, и креирати нову, уколико је то потребно. Серверска страна има константну транспортну адресу 2244, на основу које може бити препозната од стране потенцијалног клијента.



Слика 7: Слојевита *TCP/IP* архитектура

3.3 *ECMG/SCS* протокол

Након успостављене конекције и сесије, потребно је у складу са *ECMG/SCS* протоколом јасно дефинисати шта ће сервер слати као одговор на одговарајућу поруку. С тим у вези, неопходно је било парсирати долазне поруке.



Слика 8: *ECMG/SCS* протокол

На слици је приказан комуникациони протокол на основу ког серверска страна зна шта треба да пошаље као одговор.

Такође, протокол јасно дефинише и формат сваке поруке, односно како она треба да буде спакована, тако да се у складу са тим ради парсирање долазних порука, на серверској страни, и узимају се очекивани параметри, који се даље обрађују. По истом принципу се и склапају поруке које се шаљу као одговор, такође са серверске стране.

Након што је и овај модул успјешно реализован, генератор и скремблер могу да комуницирају.

3.4 *ECMG/SCS* канал и ток података

Да би комуникација била успјешна, најприје треба да се створи комуникациони канал, размјеном одговарајућих порука. Могуће је успоставити само један комуникациони канал по једној *TCP* конекцији.

Друга битна ствар, јесте размјена конфигурационих порука за пренос пакета транспортног тока података. Ако је све то одрађено без грешака, могуће је генерисање *ECM* порука, и њихово слање ка скремблеру.

3.5 *ECM* генератор

Пошто се имплементација рјешења ради у спрези са *TsDuck* програмском подршком, идеја је да се прво направи *ECM* генератор који се може тестирати у оквиру *TsDuck*-а, па у случају да та варијанта коректно функционише, исти тај генератор ће се уз неке мале адаптације прилагодити за *Widevine.*

ECM генератор прави пакет карактеристичан за транспортни ток података, чија је величина 188 бајтова.

У склопу овог пакета налази се индикатор постојања *ECM*-а, а то је поље *table_id,* чија је вриједност у хексадецималном запису 0x80 или 0x81, у зависности од парности контролних ријечи.

Такође, у оквиру пакета се преносе и парна, односно непарна контролна ријеч. Задатак контролних ријечи је да обаве иницијализацију дескремблера, и да га одржавају активним све док клијентској страни стиже енкриптован аудио – видео садржај.

4. Програмско решење

У овом поглављу биће наведени детаљи реализације програмских модула, са описима апликативне спреге у модулима.

4.1 *TCP/IP* сервер

У склопу овог модула, имплементиране су 2 кључне функције: *void TCPstart(int port, void (*onSessionEstablished)())* и void *TCPstop(int port).*

Функција *void TCPstart(int port, void (*onSessionEstablished)())* као параметре прима вриједност порта, и функцију која носи информацију да ли је сесија успјешно успостављена. Дакле, функција има задатак да сервер постави у стање ослушкивања, те да након примљеног захтјева за конекцијом, успостави везу, као и одговарајућу сесију, како би сервер и клијент могли да комуницирају.

Функција *void TCPstop(int port)* за задати порт кроз параметар, гаси жељени сервер, као и цијели систем.

4.2 *ECMG/SCS* протокол

За овај модул можемо рећи да је кључан, из више разлога. У склопу њега, дефинисана су правила на основу којих се размјењују поруку између клијента и сервера. Такође, у склопу овог модула, извршена је имплементација функције која је задужена за парсирање и даље тумачење долазних порука.

Та функција је *int32_t msg_pars(const uint8_t* buff, uint32_t size, struct ecmgp_msg* msg)*. Функција као први параметар прима долазну поруку коју је потрбно парсирати. Други параметар који се прослеђује функцији је дужина долазне поруке изражена у бајтима. Трећи параметар је дефинисана структура поруке, која се састоји од верзије протокола који користимо, типа поруке која се парсира, броја параметара који се преносе, и на крају самих параметара који се преносе том поруком. Параметри су такође организовани кроз структуру података, која се састоји од типа параметра, дужине параметра, као и вриједности параметра. С тим у вези, осим горе поменуте функције за парсирање порука, имплементирана је још једна функција, *uint32_t param_pars(const uint8_t* buff, uint32_t size, struct ecmgp_param* param)*, која се користи за парсирање параметара поруке, и позива се унутар саме *msg_pars()* функције.

Функција *int32_t ecmgp_init(struct ecmgp_client_interface client)*, има сврху да иницијализује протокол, односно да након успјешно успостављене везе између клијента и сервера, створи услове за нормалну комуникацију.

Као параметар прима структуру показивача на функције, а те функције су уствари задужене за обраду сваке поруке појединачно, односно кроз њих је дефинисано које параметре која порука треба да достави, и у зависности од тога ти параметри се чувају, или прослеђују даље на обраду.

У било ком тренутку серверска или клијентска страна може да пошаље захтјев за провјеру стања у тренутно активном комуникационом каналу међу њима. Функција *int32_t ecmgp_channelTest(uint16_t channelId, struct channel_status* status)* је задужена за то. Функција за задати идентификатор комуникационог канала, враћа његов тренутни статус.

Као што могу да провјере стање у комуникационом каналу, на потпуно исти начин клијент и сервер могу провјерити стање тренутно активног тока података преко функције која је имплементирана скоро на исти начин као ова горе поменута. Та функција је *int32_t ecmgp_streamTest(uint16_t channelId, uint16_t streamId, struct stream_status* status)*. Да би вратила тренутни статус активног тока података, потребно јој је додатно прослиједити и идентификатор датог тока података.

4.3 Псеудо *ECM* генератор

Крајњи циљ самог пројектног задатка био је направити генератор *ECM* порука. Као што сам већ поменуо, за потребе тестирање и симулације клијентске стране сам користио *TsDuck* програмску подршку. Из тог разлога, било је потребно направити генератор *ECM* порука који је прилагођен овом виду програмске подршке. Имплементација генератора је одрађена кроз функцију *int32_t gen_ecm_datagram(uint8_t* ecm_datagram, struct ecmgp_msg* msg)*. Функција је реализована тако да као параметре прима парне и непарне кључеве са клијентске стране, и да у складу са њима формира *ECM* датаграм, који се шаље као одговор клијентској страни. На основу њега, клијентска страна може да дескремблује скремблован аудио – видео садржај, и коректно репродукује претходно заштићен садржај.

4.4 *Widevine ECM* генератор

За реализацију *Widevine ECM* генератора сам искористио већ постојећу *wv_cas_ecm.h* библиотеку. У склопу ње су већ имплементиране функције које су ми биле потребне. Мој задатак је био да их само позовем на одговарајући начин, и пошаљем им потребне параметре. Напоменуо бих само да је *wv_cas_ecm.h* библиотека писана у програмском језику C++, а сви модули чији сам ја аутор су писани у програмском језику C, тако да је било потребно додатно адаптирати Make фајлове, како би све могло да се преводи како треба, и коначно да се добије функционална извршна датотека.

wv_cas_ecm.h библиотека припада *media_cas_packager_sdk* пакету. Функције које су биле од значаја *из wv_cas_ecm.h* библиотеке су све функције које се налазе у оквиру *WvCasEcm* класе. Ту спадају следеће функције: *Initialize(), GenerateTsPacket(), GenerateEcm(),* као и *GenerateSingleEcm().*

Нарочито бих истакао функцију *GenerateEcm(),* која обавља генерисање *ECM* датаграма. Функција као параметре прима кључеве, и то парни и непарни, и на основу њих генерише одговарајући *ECM* датаграм. Након тога, позива се функција *GenerateTsPacket(),* која има задатак да генерисан *ECM* датаграм упакује у одговарајући пакет транспортног преносног тока, који као такав може да се шаље клијентској страни.

5. Резултати

У овом поглављу биће описан начин на који је тестирано коначно рјешење, на основу ког је потврђена његова испраавност.

Тестирање коначне имплементације *ECM* генератора, као и његове интеграције у цјелокупни систем, обављено је на 2 начина, и то:

1. Коришћењем *TsDuck* програмске подршке
2. Директним пуштањем садржаја са мреже, уз употребу развојне плоче

При тестирању, првобитно смо користили податке из фајла. Тако организоване податке у транспортни ток, скрембловали смо, односно шифровали, како би ограничили могућност њиховог коришћења.

За потребе скрембловања података, коришћен је софтверски алат *TsDuck*, у који смо интегрисали нашу имплементацију *ECM* генератора, која као што смо горе већ навели представља серверску страну на релацији комуникације *ECM* генератор – скремблер.

Добијени скремблован фајл са подацима, даље је анализиран помоћу алата *tsanalyze*, који се налази у оквиру *TsDuck* програмске подршке.

Поменути алат омогућава провјеру да ли се у оквиру скремблованог фајла са подацима налази *ECM*, генерисан од стране нашег *ECM* генератора.

Анализом смо могли јасно да видимо да се *ECM* коректно додаје у транспортни ток података, као и да се адекватно мијења у времену, односно да је промјена парних и непарних контролних ријечи одрађена на очекиван начин.

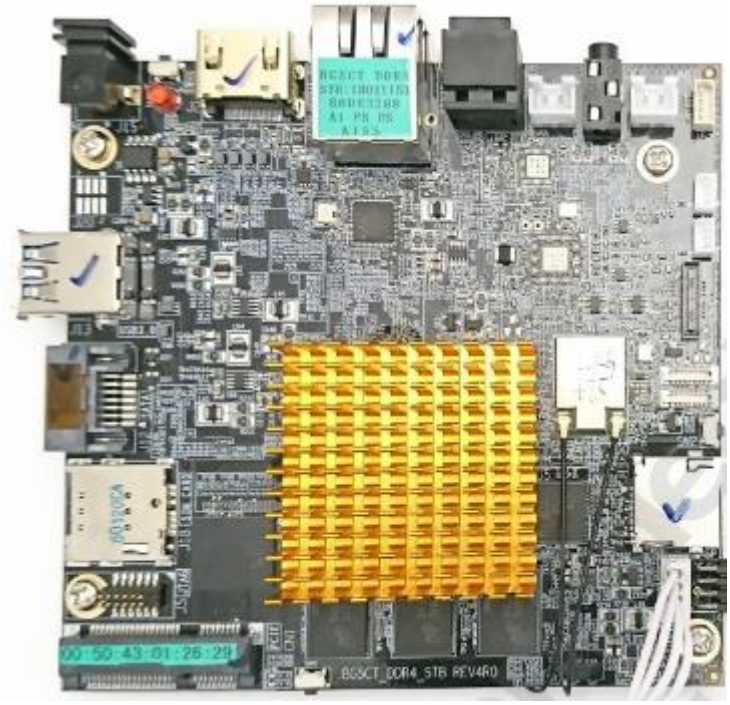
```
ECM (even), TID 128 (0x80), PID 4097 (0x1001)
Short section, total size: 152 bytes
- Section 0:
  0000:  4A D4 01 05 80 66 61 6B 65 5F 6B 65 79 5F 69 64
  0010:  31 2E 2E 2E 2E 1A C3 44 0C 18 47 76 0E 95 8A 4D
  0020:  3D 54 5F 85 EC BA 36 38 95 57 71 0C 27 91 DE 6F
  0030:  2A CF 5F F2 D7 EC 4D 3D 38 57 23 BC 91 85 BA 23
  0040:  41 00 A6 B5 80 65 76 65 6E 5F 69 76 2E 66 61 6B
  0050:  65 5F 6B 65 79 5F 69 64 32 2E 2E 2E 2E F4 C8 25
  0060:  95 48 48 78 25 3C 48 0C 95 BA 9A EB B0 51 10 6F
  0070:  E2 BF EF 21 5A 32 BE 99 CF 19 51 94 9A B3 F9 72
  0080:  0D 53 15 78 9A 16 13 7A 98 F0 D3 73 CF 6F 64 64
  0090:  5F 69 76 2E 2E

ECM (odd), TID 129 (0x81), PID 4097 (0x1001)
Short section, total size: 152 bytes
- Section 0:
  0000:  4A D4 01 05 80 66 61 6B 65 5F 6B 65 79 5F 69 64
  0010:  31 2E 2E 2E 2E F4 C8 25 95 48 48 78 25 3C 48 0C
  0020:  95 BA 9A EB B0 08 56 3E 7B 2F 70 C2 CE 22 C9 D8
  0030:  B1 F6 E9 30 13 B3 F9 72 0D 53 15 78 9A 16 13 7A
  0040:  98 F0 D3 73 CF 65 76 65 6E 5F 69 76 2E 66 61 6B
  0050:  65 5F 6B 65 79 5F 69 64 32 2E 2E 2E 2E 93 0E 73
  0060:  E0 44 CA 10 9E D6 48 99 67 02 B8 7A EB 97 B7 A6
  0070:  29 2B 8C E0 79 18 DF 27 04 48 C7 69 95 CE 31 E4
  0080:  C1 9A B6 79 01 4F A5 09 8E 0A 5B F4 AD 6F 64 64
  0090:  5F 69 76 2E 2E
```

Слика 9: Детекован *ECM* у транспортном току података

Други корак у верификацији рјешења било је пуштање садржаја директно са мреже уз помоћ *STB* развојне плоче *Synaptics BG5CT.*

У овом кораку, успјели смо да пустимо канале као што су РТС, ПРВА и ПИНК.

На овај начин, добили смо коректан садржај који се емитује без икаквих проблема, и на тај начин потврдили да је имплементација нашег генератора одрађена како треба, те да се његова функционалност не доводи у питање.



Слика 10: *STB* развојна плоча *Synaptics BG5CT*

6. Закључак

У оквиру овог рада приказано је једно рјешење имплементације *ECM* генератора. Идеја је била да рјешење буде што генеричније, и да се уз минималне измјене може искористити за стварне индустријске потребе.

На основу овог примјера имплементације *ECM* генератора, на веома лак начин се може одрадити и имплементација генератора *EMM* порука, те на тај начин заокружити једна цјелина везана за поруке од интереса за заштиту транспортног тока података, односно контролисано приказивање аудио – видео садржаја.

Кроз практичну израду рада, задовољени су сви првобитно дефинисани услови, и реализоване све функционалности неопходне за правилно и ваљано понашање коначног система, у који смо интегрисали наш генератор.

Тестирањем је доказано да систем ради како треба, те да се његово понашање не мијења у времену, односно да имамо константан квалитет репродуковања аудио – видео садржаја

7. Литература

[1] https://networkstoragesecurity.wordpress.com/unicast-multicast/, датум приступа: фебруар, 2021.

[2] https://www.wikiwand.com/pl/Multiplekser, датум приступа: фебруар, 2021.

[3] https://www.researchgate.net/figure/Structure-of-the-MPEG-2-Transport-Stream-packet_fig11_261760328, датум приступа: фебруар, 2021, Joskowicz, Jose & Sotelo, Rafael. (2014). A Model for Video Quality Assessment Considering Packet Loss for Broadcast Digital Television Coded in H.264. International Journal of Digital Multimedia Broadcasting. 2014. 10.1155/2014/242531.

[4] http://fatmanscafe.blogspot.com/2011/09/conditional-access-system-cas.html, датум приступа: фебруар, 2021.

[5] https://castlabs.com/news/widevine-cas-to-disrupt-broadcast-industry/, датум приступа: фебруар, 2021.

[6] https://afteracademy.com/blog/what-is-the-tcp-ip-model-and-how-it-works, датум приступа: фебруар, 2021.

[7] Софтвер у дигиталној телевизији 1, аутори: др. Милан Бјелица, др. Никола Теслић, мр. Велибор Михић