EasyChair Preprint

№ 1974

Live Broadcast Data Processing & Security by Software Based Encryption

Anil Kumar Chaurasiya and Prashant Yadav

November 18, 2019

LIVE BROADCAST DATA PROCESSING & SECURITY BY SOFTWARE BASED ENCRYPTION

Anil Kumar Chaurasiya[1*]
Department of Electronics and Communication
NRI Group of Institutions,
Bhopal (M.P.)
*mail2anilec@gmail.com

Prashant Yadav[2]
Department of Electronics and Communication
NRI Group of Institutions,
Bhopal (M.P.)
prashant10srm@gmail.com

***Abstract*— There are various types of digital content protection techniques available in the market. Effectively securing the content is the main challenge as we have various advance tricks & methods available that can easily hack our premium content. So, delivering the content like OTT & Digital TV services to end-user makes advance data security a very interesting topic. This paper focuses on a key concept to use software-based encryption for Broadcast data security. This paper covers complete information about live data processing through broadcast equipment. The main point is securing the content through software-based encryption that will reduce the additional hardware needed to encrypt & decrypt the live content. The simplified approach of securing the content via open-source tools & platform makes the entire work interesting & cost-effective.**

Keywords: CAS , ECM, EMM, CW, OTT, DVB, MPEG, TS, STB, Security, activation, Subscription.

I. Introduction

The first DVB-based digital television services began operation in the mid-1990, hardware-based CAS clients have been used; either embedded in set-top boxes or in the form of smart cards. Because these initial services were predominantly satellite-based – transmitted over an "open" broadcast network that anyone within the footprint could receive - a CAS was required that could store the subscriber identity and their service entitlements within a secure environment. The smart card provided such an environment, and through a process of pirate compromises and subsequent technological advancements and innovations, it continues to be the workhorse of the pay-tv industry more than 20 years after its initial launch. For large pay-tv operations with high subscriber revenue and exclusive content, hardware-based security continues to be the tool of choice for nearly every pay-tv operator on the planet, because it has a proven ability to provide the protection required. Even as new threats like Control Word Sharing have emerged, smart card-based

systems have evolved to effectively combat them. And some next-generation solutions eliminate the gap that caused the vulnerability in the first place. Because of this, it is likely that smart cards will continue to be used well into the future for high-value services.

But complexity & cost involved in the card-based CAS gives the need for the software-based encryption. The basic idea behind this paper to illustrate the procedure that we can use to make the CAS system simple, secure as per the standard.

In the platform side, we can use existing broadcasting system for implementation & testing. Main point that is most important to setup this CAS system software the open source & secured OS (Ubuntu Linux i.e. most widely used) & PHP language that is very simple to understand. Other import packages are MySQL-server, apache-server, redis-server and few required dependencies. Database tables can be easily seen by open-source tool phpmyadmin for display and desktop appearance.

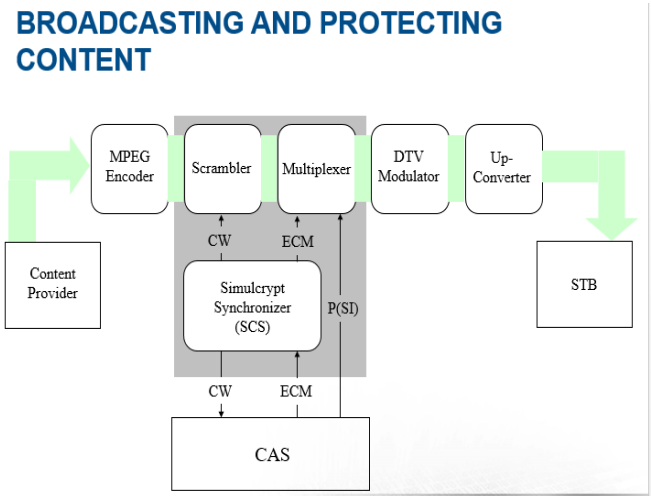

Fig 1 The basic flow of the broadcasting & Encryption

II. Literature Survey:

1. Monitoring Viewer Experience for Digital Terrestrial & Cable tv: This journal mainly focusses on QOE for the Digital TV platform. Monitoring of any product ensures that services are running smooth & error free. This proposes the end to end validation of each components & working model. Hence in order to achieve step by step execution of Live data broadcast & security using

software-based encryption this reference provides the root & basics.

2. Commutative Encryption and Data Hiding in HEVC Video Compression: In this paper, an efficient commutative encryption and data hiding scheme for HEVC videos is proposed. The commutative property allows ciphering a steganographic video without interfering with the embedded signal or to perform steganography on an encrypted video while still allowing perfect decryption. This is the key area where security work needs more attention. This paper gives idea that Video encryption converts video data into an incomprehensible form that protects the confidentiality of the content. Only the authorized user who has the right key can recover video data correctly. The technique enables applications like digital rights management (DRM) and privacy preservation. In this paper, an efficient commutative encryption and data hiding scheme based on HEVC codec is presented, which provides reliability control functionalities.

3. The past, present and future of broadcasting engineering using Nigeria, Canada, USA, Ghana and UK as a case study: This case study journal explains how Broadcasting Engineering involves the dissemination of information. In various places there are different methods of broadcasting with respect to the past, present and future. This paper also describes the past, present, future of broadcasting engineering in Ghana, USA, UK, Nigeria and Canada. It is quite helpful to understand how other countries work on the broadcasting rules & regulation.

4. Analysis of Digital Video Broadcasting – Terrestrial Second Generation (DVB-T2) Based on OFDM System on Transmission Aspect: This paper describes the basics of DVB project. DVB is a common standard that is accepted internationally for digital video broadcasting. DVB systems distribute data using a variety of approaches, satellite (DVB-S), cable (DVB-C), terrestrial (DVB-T) and digital terrestrial for handheld (DVBH). This paper proposes the practical approach to understand the process thoroughly.

5. Video Encryption Algorithm and Key Management using Perfect Shuffle: This paper, in contrast to the conventional system of Pure Random permutation, it proposed a Block Shuffling based video encryption with Faro INOUT Shuffle and rotation, i.e., first image is rotated by an angle then key is generated based Block size using Faro IN OUT perfect shuffle which is a perfect shuffling algorithm which is isomorphic to random permutation. Further we showed that our proposed method provides more scrambling of image then random permutation. Future enhancement to the proposed approach can be done by compressing video. This idea of encryption algorithm gives a complete picture to encrypt the data real time.

6. Real Time Video Encryption for Secure Multimedia Transfer: A Novel Approach: This journal proposes how encryption process & decryption process executes step by step. It explains how encryption improves transmission confidentiality of the video data. The proposed algorithm is suitable for encrypting real time video, as it provides better security in less computational time.

7. Implementation of Reed Solomon Algorithm over Data Hiding in Video Streams: This journal mainly focusses on error correction mechanism while encryption. It explains, the hiding of data in encrypted media is a new topic that has emerged an attention because of the privacy preserving requirements from cloud data management. In codeword substitution-based hiding, an algorithm is used to embed additional data in encrypted H.264/AVC video bit stream, which consists of video encryption, data embedding and data extraction stages. The algorithm can preserve the bit-rate exactly even after encryption and data embedding, and is simple to implement as it is directly performed in the compressed and encrypted domain, i.e., it does not required encrypting or partial decompression of the video stream thus making it ideal for real-time video applications.

III. Proposed Framework & Components

On referring to figure 1, we can see the typical broadcasting & protecting content setup. Let's understand each block:

1. Content provider: This is the raw data that we get from the broadcaster ex Star Network, Zee, and Sun TV, etc.
2. MPEG Encoder: We first need to encode this raw content in desired resolution & bit rate. This can be achieved by Encoder.
3. Scrambler: This is the main block where we scramble the content with the help of control word. This control word generates from SCS (Simulcrypt Synchronizer).
4. CAS: This is the block where I am focusing on this paper, CAS takes control word from SCS & encrypt using some defined algorithm & gives ECM response to the multiplexer.
5. Multiplexer: This block combines the input encoded data in the TS (Transport Stream) packets.
6. Modulator: Multiplexed stream needs modulation to further transmission to the up-conversion process. Upconversion only required in DVB –S, not in DVB-C.
7. Upconverter: If the modulated output is L band then we need upconversion to the required band ex C band & Ku band.
8. Transmission: For DVB-C signal, we don't need uplink & downlink but for DVB- S we have to uplink to the communication satellite & can downlink in the satellite footprint area. We need a power amplifier & Dish antenna to achieve this process.

9. STB: This is the receiver installed at the consumer house. It decodes the signal transmitted from the base station. Generally, Ku band signal used since it is a cheaper medium to receive the signal.

In the STB side, CAK (conditional access kernel) plays an important role in the reception of the signal & authorization of the secure content.

The Major components in the STB side are as below:

1. Front End (Tuner + Demod): Once the signal arrives from the physical transmission media, Tuner tunes to a frequency among the various multiplex of frequencies. DVB-C STB uses QAM DEMOD, which basically converts RF signal (Analog) into a digital bitstream (MPEG TS packets).
2. MPEG Transport Demux (Demux + Descrambler): Demux receives the MPEG TS packets from Demod. It filters out the TS packets based on the Channel (Service) to be viewed. Also, Demux identifies the scrambled packets and forwards them to Descrambler. The packets are then de-scrambled by the De-scrambler and are sent back to Demux. Demux then forwards these packets to the Video decoder and Audio decoders respectively based on the stream type of the packets.
3. Decoders: Video decoder decompresses the Video packets into a sequence of pictures which are displayed on the TV. Audio decoders decompress the Audio packets and are given to the Audio output device.
4. CA Kernel: CA Kernel receives the ECM and EMM packets from Demux. It is then responsible to parse the ECM, EMM packets and extract the Control Word. Control Word is set to the De-scrambler for descrambling the content (stream).
5. Application (TV UI): Application (TV UI) is responsible to take the input from the USER through Remote and pass on the events to the Middleware. The application also takes care of displaying the necessary information to the USER, for ex: displaying EPG, Bmail, Information banner, etc...
6. Middleware: Middleware plays a vital role as it processes all the events sent from the Application. It acts as an interface between the Application and the Device drivers, which converts the key events into actions and notifies the output of the necessary actions to the TV UI.

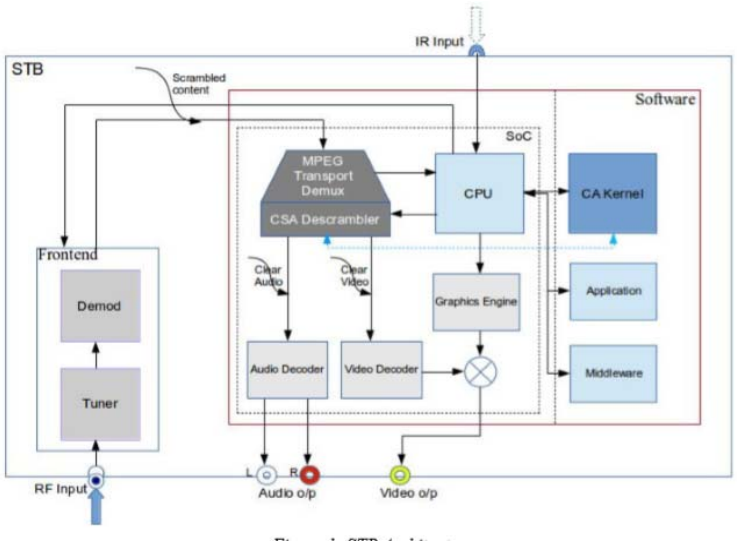

Fig 2: Downlinking & decoding the signal in STB

Here we have covered all the basic components that really required in the setup. Now our main motto is to understand how to encrypt the content. Clear or FTA content can be easily decoded without any cost factor. So, Industry needs to secure the content. For this, we need DRM & CAS. Generally, IPTV & OTT industry use DRM and STB industry use the CAS. To understand CAS first we need a detailed process of encrypting the content in the scrambler side. Please refer below diagram:

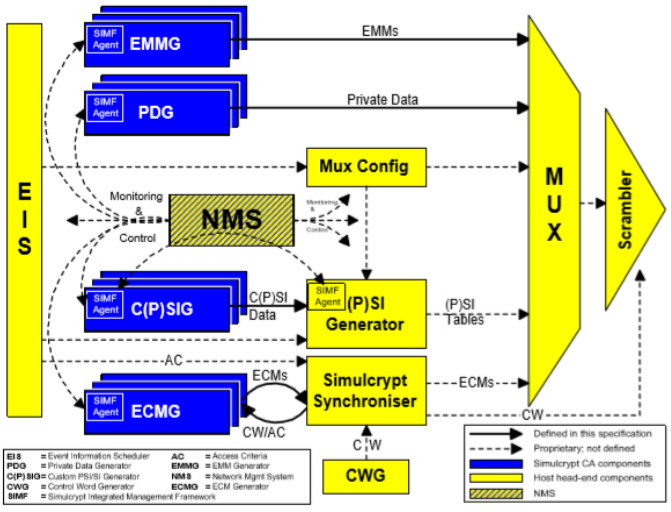

Fig 3: Detailed encryption process

EMMG & ECMG protocol plays an important role here.

IV. CAS & SMS implementation to achieve the Encryption

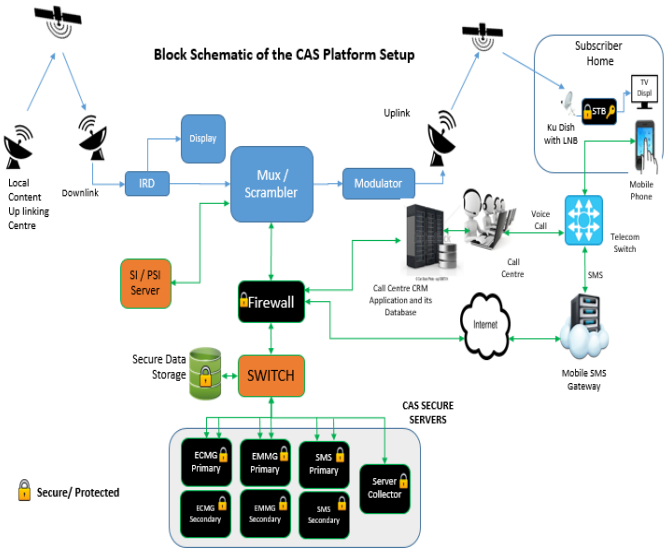

Fig 4. End to end flow of Broadcast system with CAS setup.

SMS & CAS server: These servers come under secure servers. We need to install ECMG, EMMG & SMS servers to run these instances as per the no of subscriber's databases. We can maintain redundancy if required.

SMS Server & uses:

SMS stands for the Subscriber management system. Here we maintain the inventory of the customers & maintain payment details. All the EMM & ECM commands can be sent from the SMS server also. Here are the main commands that can be sent:

1. Activation
2. Package subscription
3. Package De-subscription
4. De-activation
5. B-mail
6. Fingerprinting
7. Service Tuning
8. OTA services
9. Factory reset
10. Alert services
11. A-la-carte based subscription

Above commands can be targeted by individual, group, region & global level. The payload generated by these commands goes to the EMMG server & MUX receives the data through TCP connection.

SMS system can handle various features as per the project requirement as below:

1. User management (Admin, operator, Install & monitor)
2. User Group for a big organization
3. Categories of the various types of channel, same as BAT bouquet association table.
4. Creation of various channel or services & segregation of each channel as per the supplier or as per the requirement.
5. Subscription mechanism like daily, weekly, monthly or yearly basis.
6. PPV (Pay per view) & Package based subscription.
7. Inventory tab with all the required input for maintaining the customer details.
8. Customer & consumer device pairing as per the CAF, Customer application form.
9. Payment & billing mechanism to support Customer + Supplier in the same database.
10. Most important backup & restore mechanism.

Role of SMS is to give the flexibility for the MSO (Multi-service operator) to handle their project in the most effective manner without any security concern. We can analyze the entire data of the project with the help of this tool. Redundancy or mirroring is another way to keep our data replicate in another place in case of any outage/failure.

CAS Server & uses:

CAS server required to deploy the CAS software. SAS consists of ECMG and EMMG along with the configuration files. For Head-end to trigger ECM and EMM commands EMMG and ECMG should be up and running. SAS build is received as a .deb file.

We can use Ubuntu Linux open source OS to give runtime environment. CAS server needs some packages to install as below:

1. sudo apt-get update
2. sudo apt-get dist-upgrade
3. sudo apt-get install mysql-server apache2 php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl php5 libapache2-mod-php5 php5-mcrypt phpmyadmin mcrypt
4. sudo php5enmod mcrypt
5. sudo service apache2 restart
6. sudo apt-get install redis-server
7. sudo apt-get install ssh
8. sudo apt-get install libbsd-dev
9. sudo apt-get -y install openjdk-7-jre qtcreator build-essential libssl-dev libqt5sql5-mysql
10. sudo apt-get install libjpeg62:i386
11. sudo apt-get install libsm6:i386
12. sudo apt-get install libgtk2.0-0 libgdk-pixbuf2.0-0 libfontconfig1 libxrender1 libx11-6 libglib2.0-0 libxft2 libfreetype6 libc6 zlib1g libpng12-0 libstdc++6-4.8-dbg-arm64-cross libgcc1
13. sudo apt-get install libxrender1

14. sudo apt-get install libfontconfig1
15. sudo apt-get install libice6 libsm6 libxt6 libxrender1 libfontconfig1 libcups2 libsm6:i386 libxrender1:i386 libfontconfig:i386 libxtst6:i386
16. sudo apt-get install lib32stdc++6

Note: These are the commands to install the required config files. These dependencies meet the requirement as per the references taken to make the software.

After these packages' installation, we install CAS .deb file. And need to run the ECMG & EMMG instance. CAS server ECMG connection is also TCP based but it is two-way communication. MUX sends control word & CAS server encrypt the CW & sends the ECM response back to MUX that further goes to STB.

Use of CAS server:

- CAS server provides the platform to register the services/channels to get encrypted.
- CAS provides a service key to each service to send the MUX.
- This service key goes to MUX as ECM response.
- Control word provision starts from the MUX & CAS server revert it as ECM response.

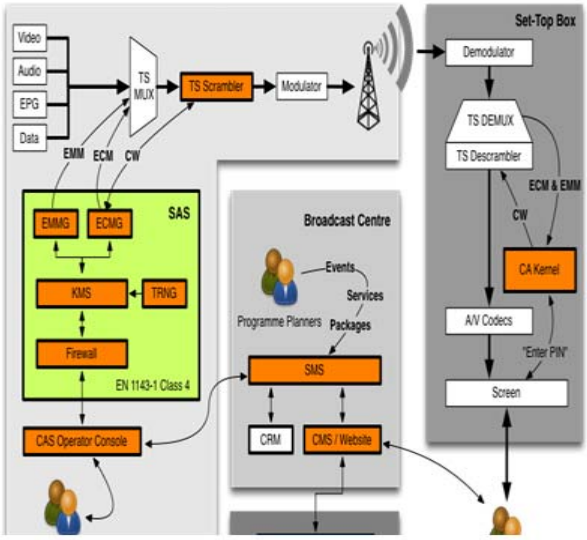

Fig:5 Implementation level Flow

V. Results & Wireshark Analysis:

Wireshark is very useful software to analyze the network protocol. We can analyze the data going from the server to mux i.e. client. After connecting the CAS server & SMS server we can see the results through Wireshark whether connection established or not.

Let's check first the EMMG connection.

EMMG connection analysis:

Please refer to below fig for the connection.

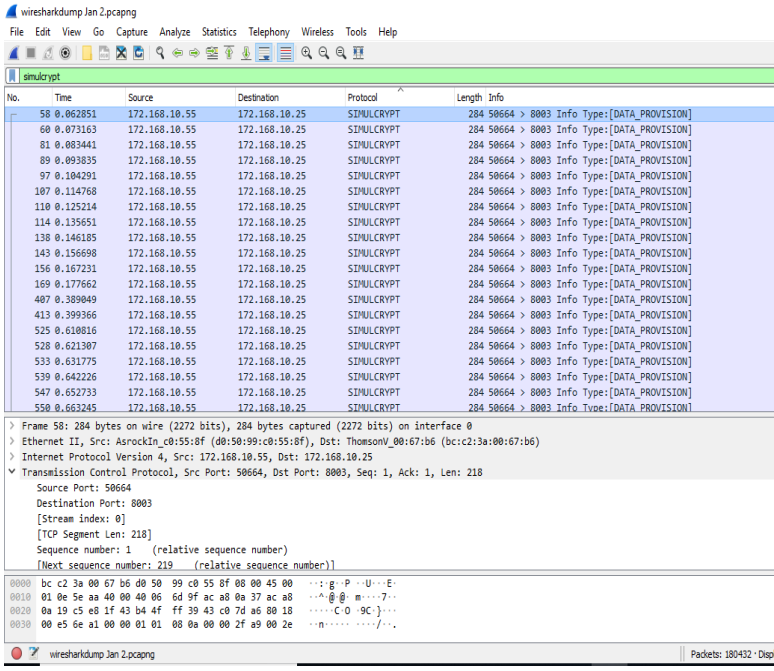

Fig 6 Data provision from the server to MUX. {Here server IP is 172.168.10.55 & MUX IP is 172.168.10.25. The server is sending data in the Port 8003.}

Here data provision means data is going from the EMM server to MUX. Here data is hitting to the port 8003 so that MUX can easily find to process further.

Required parameters are as below:

Client ID

Data ID

Stream ID

EMMG to MUX interface port

Channel ID

These parameters are very helpful in the troubleshooting on the connectivity side. If these parameters are properly reaching to the MUX, then it means the connection is established & working properly.

ECMG connection analysis:

Please refer the below fig to check the connectivity between MUX & server.

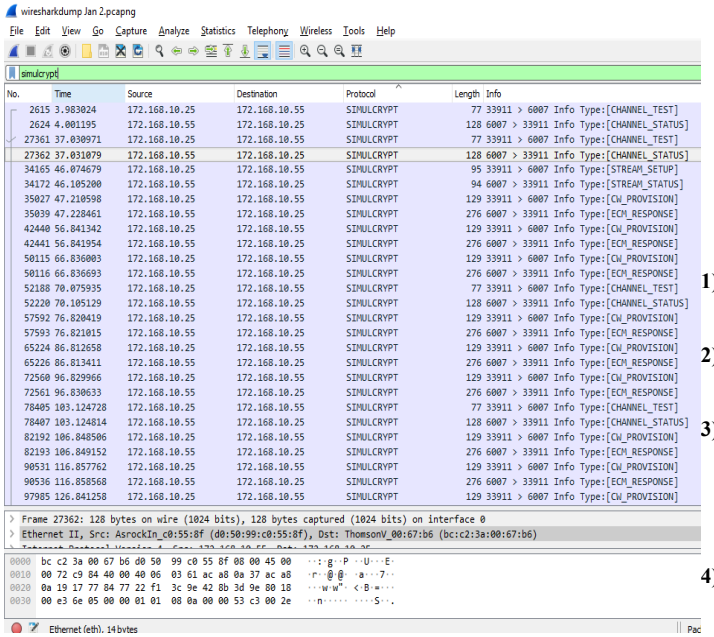

Fig 7 ECMG connection {here server IP is 172.168.10.55 & MUX IP is 172.168.10.25. The server is communication with the Port 6007.}

As per fig,

Channel test request comes from MUX then server replies to MUX channel status.

After positive ACK, Stream setup request comes from MUX then server send Stream status.

If positive ACK, MUX sends CW provision, in return, Server sends ECM response.

Once MUX received ECM response process gets complete. This repeats as per the configuration defined or well suited. In general, every 10 sec. CP request comes from MUX & Server sends ECM response.

VI. Conclusion:

Hence, we have seen the new & simplified approach of live broadcast data processing & security using software-based encryption. We covered all the blocks that been used in the DVB broadcast as per the standard. We have outlined how to check the connectivity using the Wireshark. This paper elaborates all the fundamentals to process the data through satellite as well as cable. We saw how secure connection establishes between servers to MUX & processing of the data takes place. Most important is the approach that takes place in the server side is completely cost-effective since the platform that has been used is open source. As the entire encryption process is dependent on software-based solutions, consumer end also needs to focus basic minimum hardware that is required for processing the content/signal, card-less CAS solution is coming in the picture. These solutions are more reliable & needless dependencies on physical card-based security. Govt of India is also supporting cardless CAS solution under make in India campaign.

VII. References:

1) Ioan Tache “Monitoring Viewer Experience for Digital Terrestrial & Cabel tv” IEEE TRANSACTIONS 2018

2) DAWEN XU “Commutative Encryption and Data Hiding in HEVC Video Compression” IEEE TRANSACTIONS Apr 2019

3) Frances Nwukor1, Alashiri Olaitan2, Jean Akpamu3, Felix Igwe4 “The past, present and future of broadcasting engineering using Nigeria, Canada, USA, Ghana and uk as a case study” International Research Journal of Engineering and Technology (IRJET) Aug 2018

4) Sukamto1, R. Gaguk Pratama Yudha2 “Analysis of Digital “Video Broadcasting - Terrestrial Second Generation (DVB-T2) Based on OFDM System on Transmission Aspect” International Research Journal of Advanced Engineering and Science ISSN (Online): 2455-9024 IRJAES 2018

5) Sayyada Fahmeeda Sultana *, Dr. Shubhangi D C** “Video Encryption Algorithm and Key Management using Perfect Shuffle” Int. Journal of Engineering Research and Application: IJERA TRANSACTIONS JUL 20 17

6) Neha Dilkash1, Anku Gupta2, Arpita Jain3 “Real Time Video Encryption for Secure Multimedia Transfer: A Novel Approach” International Journal of Engineering Science and Computing, April 2018 IJESC TRANSACTIONS

7) Shana Jebin P, Shyni K “Implementation of Reed Solomon Algorithm over Data Hiding in Video Streams” International Journal of Innovative Research in Computer and Communication Engineering Vol. 5, Issue 4, April 2017 IJIRCCE