IMPLEMENTATION OF KRYPT. *OUR VIDEO ENCRYPTION MOBILE APPLICATION*

by

Group 2 Members

Department of Computer Sciences

University of Lagos

500 level

(**Ref:** CSC504/ May 2017)

GROUP 2 MEMBERS

Alagbe Akinwale	120805024
Nenne Nwodo	130805063
Adedara Olanrewaju Samuel	120805006
Omitiran Demilade Michael	120805102
Chukwu Barbara	120805045
Akeju Fatai Adekunle	120805013
Bamgbade Babatunde	120805039
Enyi Queenet Nnenna	120805056
Iribiri Mary	120805068
Akinbode Adedapo	130805071
Obawole Daniel Oluwakorede	120805078
Oluyemo Oyindamola	130805066
Omosowon Afolorunsho	120805103
Ojo John Babatunde	120805090
Maliki Yahya	120805074
Olamide Adeniyi Olaniyan	120805098
Jamiyu Abiola Azeez	120805069
Ayodele Taofeek	120805073
Babalola Seun	120805037
Adeodu Caleb	110805007
Okoye Chiamaka	110805064

ABSTRACT

This report describes Cryptography, Video Encryption and the AES algorithm. The report is divided into four main chapters. The first chapter introduces the Case Study, the second chapter gives a general overview on encryption and video encryption, the third chapter goes deep into the AES algorithm which was used for implementation and the fourth chapter talks about the application which was implemented.

The project within is a Video Encryption Application for Mobile devices. The system is coded with Java. The Analysis, Design and Implementation are further described in the body of the report.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

Most times, information is meant to be classified as being for a person or for a group of people. The increase of the use of the internet and the increase in the creation of *"private"* information are directly proportional. Problems arise by attackers who somehow gain access to the system when attempts are made to alter the privacy of the information.

Information security is extremely vital in today's world, this is because it protects private or confidential information from intruders (attackers). This case study focuses primarily on videos.

1.2 Problem Statement

Smartphone users store messages, videos, photos and other multimedia. The absence of inbuilt encryption for videos has led to inconveniences for users who may choose to protect confidential videos that are just saved plainly in their gallery. The videos are open to people who steal or somehow have access to their phones. This has led to theft of ideas, strategy etc.

1.3 Objectives of the Study

The objectives of the study are:

- Understanding encryption (and decryption) algorithms that can be used for videos.
- Developing a mobile application which helps in the encryption and decryption of videos stored locally on the device.
- High confidentiality and improved security.

1.4 The need to secure data

Regardless of secure passwords, pins and backups, there is still a great need for us to ensure our privacy, protect the data and secure intellectual property. Most times, not much can be done in the case of physical security (theft), but encrypting protects confidential data from unwanted access. Also, it is possible for data in transit to be intercepted; for example, data transmitted through networks, mobile telephones, Bluetooth, etc. Encryption of these data prevents eavesdropping of network traffics by unauthorized users.

CHAPTER TWO
ENCRYPTION

*"Cryptography (from Greek word **kryptós**, which means 'hidden secret') is the practice and study of techniques for secure communication in the presence of third parties called adversaries"* (Wikipedia, 2017).
"Encryption is the process of changing information from one form to another to hide its meaning" (Merriam Webster Online, 2017).
"*Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form"* (Techopedia, 2017).

2.1 Overview of Encryption

"The concept of Encryption can be dated back to the time of the Romans and the Greeks, who sent secret messages by substituting letters that can only be deciphered with a secret key" (Wikipedia, 2017).

Encryption is the process of encoding a message such that it can only be viewed by only those that have access. Encryption is a medium used to prevent original data access to intruders, intrusion is still possible, but the intruders will see the encrypted data and not the original data. This procedure requires using an algorithm to encrypt the **plain text** (original data), the encrypted result is called the **cipher text** and this is what is decrypted in return to get back the original message. A private key is used for encryption and decryption.

2.2 Video Encryption

This is the process of making video files private, either for personal reasons or Digital Rights Management. When encrypting a video file, we have to know that:

- No computer is immune to cyber-attacks and intrusion, however the cost of attacks can be increased by encrypting with larger bit depths.
- An unencrypted version of the video file should be kept in a secure place due to fact that constant evolution of technology may result in obsolete encryption methods, and if not updated, the files will become unreadable in future.

2.3 Video Encryption Methods

Naïve Approach: This method encrypts every byte in the video using traditional algorithms like AES or DES. The video bit stream is considered as text data. This method is very secure as all the bytes are encrypted one by one. However, this method is not suitable for real-time applications. This is the technique adopted for our application.

Pure Scrambling: Permutation is used to shuffle the bytes in each frame. This method is good for applications that use hardware for decryption (the software is usually responsible for decryption). Pure scrambling is susceptible to the **known-plaintext attack,** so it should be carefully used. This is because the attacker can figure out the permutation sequence by comparing the known frames with the cipher text.

Crisscross Permutation: The proposed algorithm first generates a 64 byte permutation list. This list is then quantized into an 8x8 block. This is followed by a simple splitting procedure. The random permutation list is then applied to the split blocks and the result is then encoded. Computational complexity is relatively low and hence the encryption and decryption process is not too complex. Crisscross permutation distorts the DCT coefficients and hence the video compression rate is lowered. This algorithm also cannot withstand the known-plaintext attack.

Choose and Encrypt: In real time applications, it is very impractical to encrypt and decrypt the entire video stream. In choose and encrypt, some selected video frames are encrypted. Using this technique, complexity, encryption overhead and decryption overhead is massively reduced. This algorithm is successful if a proper tradeoff can be maintained between complexity and security.

CHAPTER THREE

ADVANCED ENCRYPTION STANDARD ALGORITHM

The Advanced Encryption standard algorithm (AES) is a subset of the **Rijndael** cipher developed by two cryptographers, Vincent Rijmen and Joan Daemen.

3.1 Overview of the Algorithm

AES is a very popular symmetric block cipher which is based on the substitution-permutation network. AES possesses a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. This algorithm operates on a 4x4 column-major order matric of bytes, called the **state matrix**. E.g. if there are 16 bytes (0 – 15), they would be represented as:

0 4 8 12
1 5 9 13
2 6 10 14
3 7 11 15

The AES algorithm uses a number of **encryption rounds** which does the conversion from plain text to cipher text. The output of round I is round i+1's input. The output of the final round is the encrypted file. 128-bit keys usually have 10 rounds, 192-bit keys have 12 rounds, 256-bit keys have 14 rounds. Each round consists of several processing steps, each containing four similar but different stages.

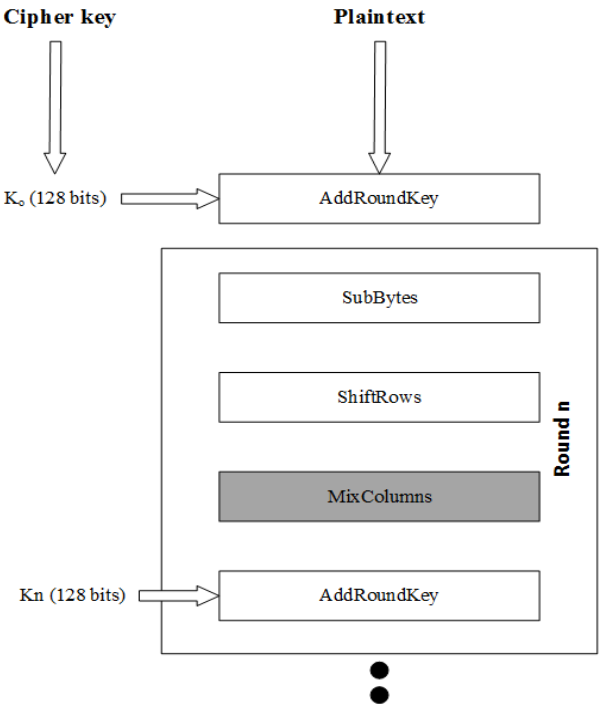
3.2 Encryption Process

The algorithm begins with an **Add Round key** stage, followed by repeated rounds of the four stages, and the final stage (which does not contain the Mix Columns step).

- **SubBytes:** This process reorganizes each byte of the state independently using the ***Rijndael S-Box***. This is done in a non-linear fashion. The S-box is constructed by the composition of two transformations:
    1. Get the multiplicative inverse in Rijndael's finite field
    2. Affine transformation which is documented in the Rijndael documentation.

Pre-calculated forms are used since the S-Box does not depend on any input. Each byte of the state is substituted by the value in the S-Box whose index corresponds to the value in the state. **a(i, j) = S-Box[a(i, j)].** The result is in a 4x4 matrix.

- **ShiftRows:** This step operates on the rows of the state. It shifts the states by a certain offset in a circular manner. For this algorithm, the first row of the state is not altered, the second, third and fourth rows are shifted 1, 2 and 3 bytes to the left respectively. The shift rows inverse (for decryption) performs these shifts to the right.
- **MixColumns:** In this step, the four bytes of each column of the state matrix are combined using an invertible linear transformation. A randomly generated polynomial is arranged in a 4*4 matrix. The same polynomial is used during decryption. Each column of the state matrix is XOR-ed with the corresponding column of the polynomial matrix. The result is updated in the same column. The output matrix is the input to AddRoundKey. This step is not included in the final stage.
- **AddRoundKey:** A round key is generated by performed various operations on the cipher key. This round key is XOR-ed with each byte of the state matrix. For every round a new round key is generated using Rijndael's key scheduling algorithm.



3.3 Equivalent Inverse Cipher

The inverse cipher is the decryption algorithm for AES. In addition, the cipher and the inverse cipher operations must be executed in such a way that they cancel each other. The round keys must also be used in reverse order. The process of decrypting an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order. Since processes in each round are in reverse manner, decryption needs to be implemented separately from encryption, although there are very closely related. The Cipher Text which is formed of 256-bit 4*8 Matrix is the input for the decryption process.

CHAPTER FOUR

KRYPT – OUR VIDEO ENCRYPTION APPLICATION

Krypt is our mobile application that encrypts videos stored locally on the device.

4.1 Functional Requirements

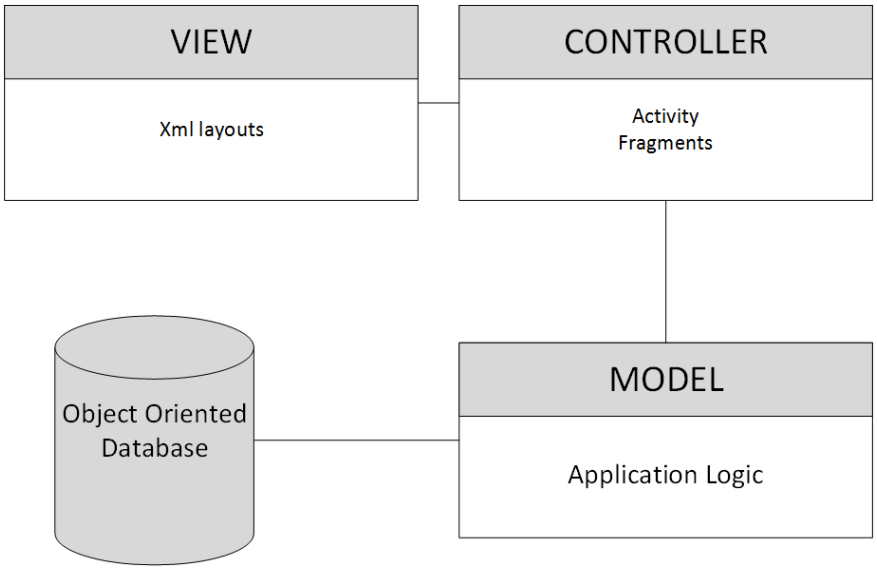
- Authentication.
- The application must allow users to encrypt videos
- The application must allow users to decrypt videos
- The application must be able to play encrypted videos.

4.2 Non Functional Requirements (Quality Metrics)

- **Security:** Krypt must provide protection of information through the mechanisms of passwords incorporated in it. This helps to prevent intruders from decrypting or viewing private videos
- **Multithreading:** Krypt must give room for simultaneous execution of multiple activities (e.g. Encrypting, Decrypting, Previewing)
- Responsiveness and User Friendliness (Adapt to all screen sizes, UI must be rich and easy to understand)
- Krypt must make efficient use of the mobile device's battery. i.e. make use of the minimum amount of energy possible
- **Portability:** Krypt must run efficiently on different Android devices
- **Scalability:** Performance must not degrade of there is an increase in the number of videos

4.3 Framework Model View Controller

The **Framework Model View Controller** in Android was the pattern used for implementation. The Android OS is known as the **framework**, the **Model** contains the application logic and communicates directly with the Data Store which in this case is an Object Oriented Database (Realm). The **Controller** contains the Activities and Fragments of the app and interacts directly with the **Model**. The **Controller** updates the **view**. The **view** is what is shown to the user, which is in form of xml layouts.



4.4 Principal System Components

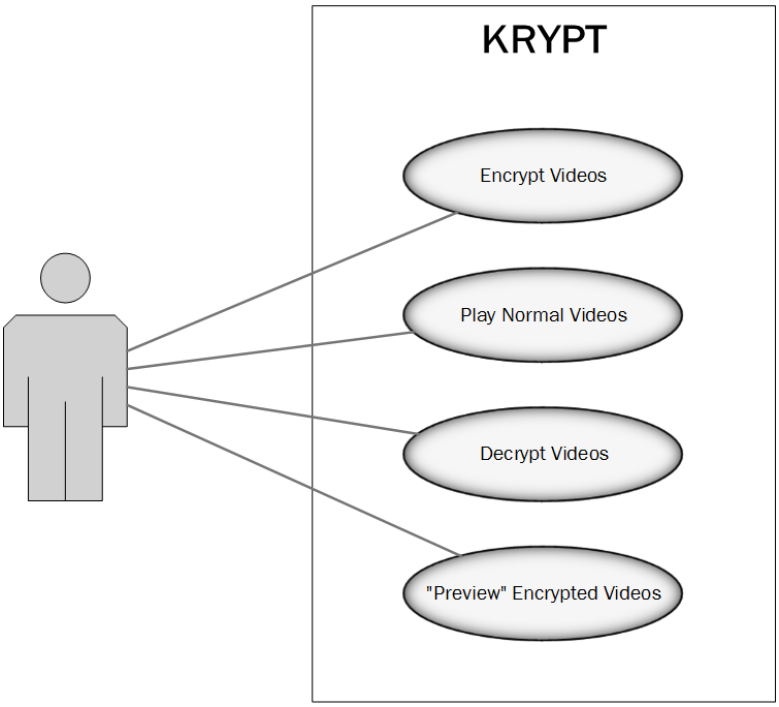
The Principal components of the system can be deduced to be:

- Video Encryption
- Video Decryption
- Play Video In app

4.5 Database Description

An Object Oriented database called **Realm Mobile Database** is used for development. The encrypted videos are saved as objects into the database. They also have primary keys associated with each object. This database is not relational, so it does not deal with tables.

4.6 Use Cases



4.7 How Krypt works

The first time a user installs **Krypt**, the user is asked to sign up and specify a pin. That pin would be the private key known to only the user that would be used to encrypt and decrypt all videos. After installation, anytime a user opens or resumes the app, there is a prompt to enter the pin to continue. This is to promote security within the app since the pin is the private key that only the user should know about.

On successful login to **Krypt**, the user can immediately see two tabs, one showing the user's *videos* from the media library and the other showing the *encrypted videos*. The user can play or **encrypt** a video in the *videos* tab, and **preview** or **decrypt** a video in the *encrypted videos* tab.

The Naïve approach is the video encryption method adopted for this project. On encryption, the video in the file directory's videos folder is encrypted using the AES algorithm. A copy of the original video is then moved to another secure folder for backup. This secure folder backs up all encrypted videos. Equivalently, the original video that has been encrypted (**not the copy)**, is moved to the *encrypted videos tab.* On decryption, the video is decrypted and moved back to the videos folder in the file directory as well as the *videos* tab in **Krypt.** If the user chooses to **preview** an encrypted video, **Krypt** decrypts the video temporarily and then plays it

4.8 System Implementation

Krypt is exclusively a Mobile Application developed using Java for the backend functionalities.

The front end for this project was designed using xml layouts. The Tabs were designed using Adapters and fragments. The system information is housed using an Object Oriented Database (Realm). Icons and Splash screens were also designed in Adobe Photoshop.

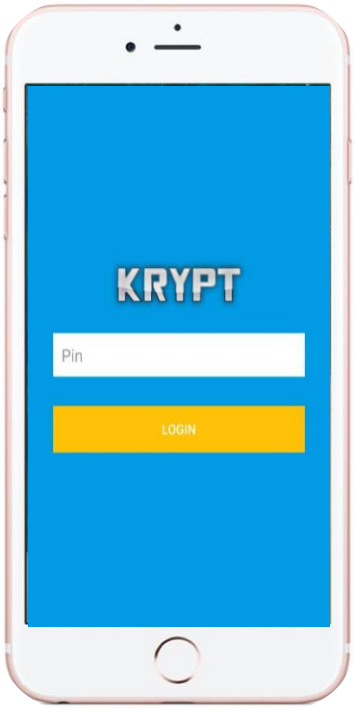
Other exception handlings are implemented using Java.

4.8 Sample Result

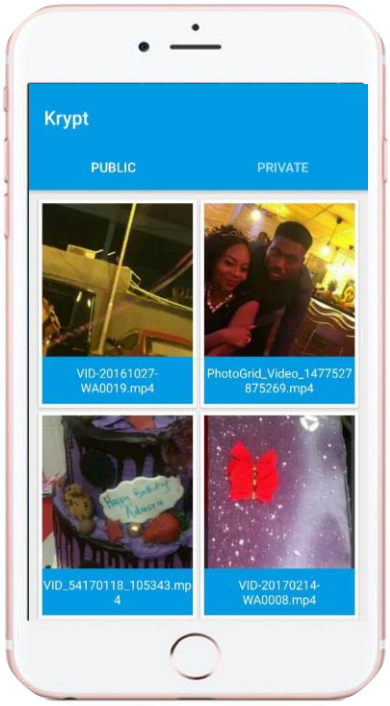
Splash Screen

Sign up page

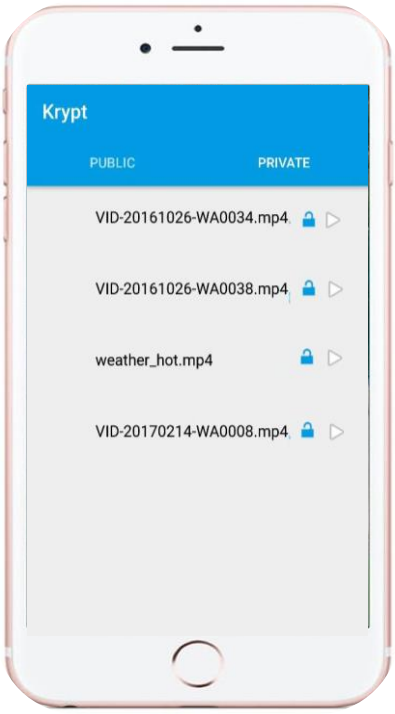
Login page



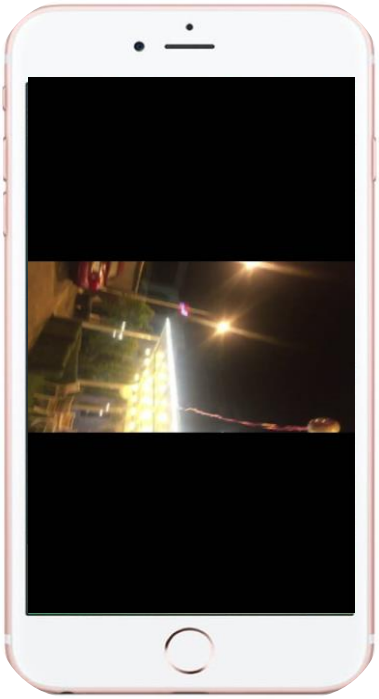
Normal Videos Tab



Encrypted Videos Tab



Playing a Video in the App



CHAPTER FIVE

CONCLUSION

In the present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches. The AES security is assured only if it is correctly implemented and good key management is employed.

Our Video Encryption Application (**Krypt**), was implemented in Java using the AES algorithm. As intruders exist over networks, they also exist physically (e.g. thieves). Krypt takes into account the importance of encryption on local content housed in the media library of the device.

We have got to understand the importance of encryption and why we should always keep important data secure at all time to prevent intruders from **viewing** the data.

BIBLIOGRAPHY

"What is Decryption? - Definition from Techopedia." *Techopedia.com*. N.p., n.d. Web. 01 May 2017.

Tutorialspoint.com. "Advanced Encryption Standard." *Www.tutorialspoint.com*. N.p., n.d. Web. 01 May 2017.

"Advanced Encryption Standard." *Wikipedia*. Wikimedia Foundation, 30 Apr. 2017. Web. 01 May 2017.

"AES and RSA Encryption." *Encryption software to secure cloud files*. N.p., n.d. Web. 01 May 2017.

"Cryptography." *Wikipedia*. Wikimedia Foundation, 27 Apr. 2017. Web. 01 May 2017.

"Encrypt." *Merriam-Webster*. Merriam-Webster, n.d. Web. 01 May 2017.

Jagadev, Aseem. *AES Implementation*. Thesis. National Institute of Technology Rourkela, 2009. N.p.: n.p., n.d. Print.

Moser, Jeff. *A Stick Figure Guide to the Advanced Encryption Standard (AES)*. N.p., n.d. Web. 01 May 2017.