DOCUMENT READING SYSTEM FOR BLIND PEOPLE ("READING EYE")

Final Report for Security Module

Project ID: 19_20-J17

I.N.Kalansooriya
IT16176348

Supervisor – Mrs. Suranjini Silva

Co-Supervisor- Dr. Anuradha Jayakody

Bachelor of Science (Hons) in Information Technology Specialized in Cyber Security

Department of Computing

Sri Lanka Institute of Information Technology

Sri Lanka

5th May 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Signature: Date: 5th of May 2020

Signature of the Supervisor: Date: 5th of May 2020

Signature of the Co-Supervisor: Date: 5th of May 2020

ABSTRACT

Reading eye is a mobile application which is developed for visually impaired individuals to assist with reading materials such as books, documents. This application is differing from others by its ability to perform extraordinary functions such as reading images, tables, equations and graphs. Also, it has unique encryption mechanism to safeguard the client-server connection.

ACKNOWLEDGEMENT

I would like to express my gratitude to supervisor Mr. Suranjini Silva and co-supervisor Dr. Anuradha Jayakody for the guidance and mentoring given throughout this period to make my research a success.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NIST	National Institute of Standards and Technology
AES	Advanced Encryption Standard
DES	Data Encryption Standard
PFS	Perfect Forward Secrecy algorithm
DRA	Double Ratchet Algorithm
OTR	Off the Record algorithm

1. INTRODUCTION

1.1. Background Literature

Security is a main concern of IT systems. Lot of mobile applications are lacking security. There are few mobile applications that use encryption mechanism to secure their data. WhatsApp, Viber, Telegram are some of them. These mobile apps use end-to-end encryption because they are communication applications. Viber use Elliptic-Curve Diffie Hellman key-exchange algorithm for sharing the keys. The encryption algorithm is Salsa20[1]. The key use to encrypt is 128-bit symmetric key. WhatsApp encryption algorithm developed by a company named Open Whisper Systems. They developed an encryption algorithm using Perfect Forward Secrecy algorithm (PFS), the Double Ratchet Algorithm (DRA) and Off the Record algorithm (OTR). Also, WhatsApp use Elliptic-Curve Diffie Hellman key-exchange algorithm for sharing the keys. Telegram application as their own encryption algorithm called MTProto[3] which is developed using 256-bit AES.

1.2. Research Gap

There are many mobile applications for visually impaired people. But they are lacking Vital security mechanisms. Because of that Integrity, confidentiality and availability of these applications may at risk. There are many applications that have encryption as a security mechanism. But characteristic of those applications are very different. Therefore, we cannot apply those security mechanisms to reading eye application. Therefor need of developing security mechanism for the reading eye is essential.

1.3. Research problem

Reading eye is based on machine learning algorithms. When image processing these algorithms need lot of process power. But cloud high performance VMs are very costly. Because of that we couldn't afford high performance VMs for the reading eye.

1.4. Research objectives

1. Securing client-server communication

Using modern technology attackers can intercept client-server communication and they can gain access to critical data. Therefore, securing client-server connection is vital. When using encryption even though attackers intercept the connection hey cannot read data.

2. Light weighting the images

Modern days mobile phones have high definition cameras. The images that took from these cameras at are 6MB in size. Therefore, uploading these images to cloud can become time consuming and inefficient. Therefore, light weighting algorithm is essential.

3. Create cloud policy to enforce the security of the cloud

Even though “Reading Eye” is developing as mobile application, 99% of its process is done by the cloud. Basically, heavy lifting of the system is done by the cloud. Therefore, cloud security is vital fact for the systems availability. There are many security concerns when using the cloud. The cloud policy will address these issues.

2. METHODOLOGY

This section contains methodologies about how to secure client and server side, how to light weight the image and how to secure cloud using cloud policy.

After taking the picture it will be resized and split into image segments and then image segments will be encrypted using AES hybrid encryption algorithm with a 256-bit key and send to the cloud as an image stream (Figure 1).

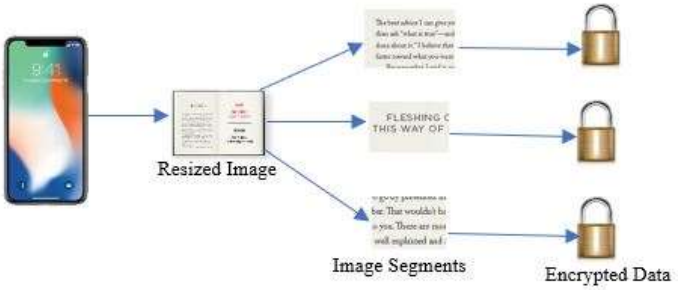

Figure 1 Client side

The cloud will receive the encrypted segments and it will decrypt them and merge them together to get the full image. The key for the encryption will exchange using Diffie–Hellman key exchange method.

Server and client have common two numbers p and g and a secret number. client generate a key using p, g and secret number and then send to the server. Server generate a key using its secret number and p and g and then send to the client. After that server create a secret key using client's key and server's secret number. Client creates secret key using its secret number and server's key. Both secret keys are same. This is called Diffie–Hellman key exchange method. This method is used to exchange key through insecure channels. In this method the key is not sending through insecure channels.

After taking the picture the mobile application creates a connection to the server. And then it will generate a key and send it to the server. The server will do the same. Then both parties create same secret keys. Then the app will resize and split the image into segments. Then the segments will be encrypted using secret key and send to the server.

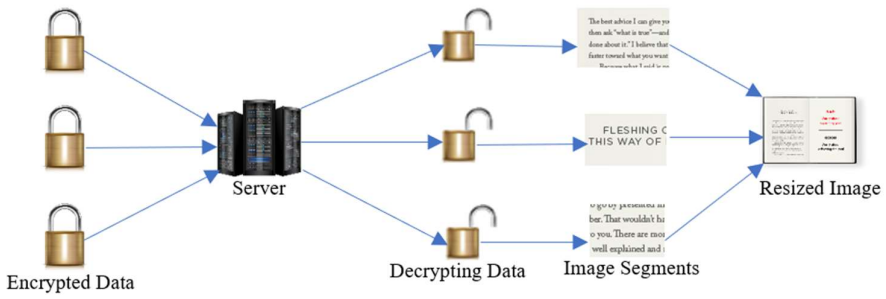

Figure 2 Server side

2.1 Securing client and server communication

Due to advancement of modern technology cyber-attacks to IT systems are rapidly growing in a considerable rate. Hackers can intercept client-server communications very easily. Therefore, need an encryption mechanism to secure data from hackers. The encryption is converting readable and understandable data to unreadable and meaning less data. When hackers intercept encrypted data, they cannot read or understand it. This way we can secure data and secure client server connection.

They are many encryption algorithms. Not every encryption algorithm is secure. Hackers can crack some encryption algorithms. According to NIST (National Institute of Standards and Technology) encryption algorithms must possess vital two functions if they are secure.

- Diffusion
- Confusion

Diffusion is concealing the relationship between the plain text and the ciphertext. Confusion is obscuring the relationship between the key and the ciphertext. When considering the key length of the encryption algorithm, encryptions have higher chance to cracked if they have keys which are having short key length. Hackers can use brute forcing method to crack these encryptions. Therefore, I had to choose an algorithm which has confusion and diffusion, and which has longer key length. Therefore, I choose most famous known to be secure AES algorithm. U.S.A use this algorithm to secure their data.

AES is originally invented by Vincent Rijmen and Joan Daemen. It is a symmetric block cipher. It encrypts data block by block. It can have 256-bit length key. It is known as a successor cipher of famous DES cipher.

When implementing AES on android we must consider about the device's performance. Because AES need more process power. AES has four main functions.

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

In order to optimize the algorithm for mobile applications I had to find out, out of above functions which one need more process power than others.
I use time module in pyhton3 to measure times above functions take. After studying result, I concluded that MixColumns function need more process power than others. It performs complicated arithmetic. It operates column-by-column on the block, treating each column as four-term polynomial [4]. following figure 3 shows how bytes in a column get replaced.Figure 4 shows how the MixColumns function operates.

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

Figure 3 How four bytes in a column are replaced

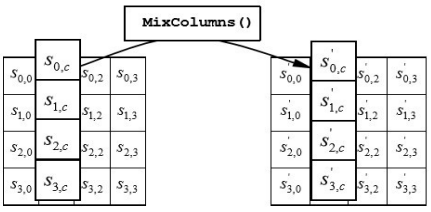

Figure 4 MixColumns() operates on the State column-by-column

In AES, diffusion part is done by the MixColumns function. Therefore, I needed to find efficient diffusion function that will replace the MixColumns function. My basic instinct was to try DES algorithm. Because AES is the successor algorithm of DES.AES and DES has similarities. In DES diffusion function is done by the permutation function. It basically uses permutation table to replace data bytes. DES is an efficient cipher. But it is not secure because of the short key length (56-bit). Therefore, I try to replace MixColumns function with permutation function. But I couldn't use same permutation table because AES has 128-bit size blocks. But DES has only 64-bit block size. Therefore, I created my own permutation table for AES.AES block is 4*4 bytes state (Figure 5). I stored permutation table as an array. Because of that I needed to convert matrix into an array before applying the permutation function.

```
pc1 = [6, 4, 15, 7, 13, 9, 11, 2, 5, 14, 8, 10, 0, 12, 1, 3]
fp  = [12, 14, 7, 15, 1, 8, 0, 3, 10, 5, 11, 6, 13, 4, 9, 2]
```

Figure 5 Permutation table for Hybrid algorithm

After applying the function, the array will be converted back to a 4*4 matrix. After applying these changes and checking encryption and decryption time, I could see the hybrid AES algorithm is much faster than the original algorithm.

2.2 Light weighting algorithm

When sending images to the server we need to speed up the sending process by light weighting the image. This is very vital process when considering the efficiency of the mobile application. After taking the picture it needed to be light weighted. Due to performance of the mobile devices the algorithm must be very simple. Therefore, I used resizing method and image streaming method to light weight the image. After capturing the image, it will be resized into 1000 pixels width and height. This will reduce the mage size significantly. The image will be

split into 200 pixels width and height image segments. Then the image will send to the server like an image stream.

2.3 Cloud Policy

Cloud security is very vital for the overall performance of the system. Because the system will be hosted in the cloud and if something happens to the reading eye server in the cloud, availability of the whole system will be lost. Therefore, I created the cloud policy to enforce the security of the system. Following are the cloud policies.

Password policy

- Password must be at least 10 characters long
- Must include numbers, upper case letters, special characters
- Must not use personal information (birthdays, NIC number, name, organization name, license plate numbers)
- Must not include meaning full words
- Must include numbers in username
- For admins, Password must be changed after every two months.
- For super user, Password must be changed every month.
- For normal users, Password must change every six months.

User policy

- Give minimum privileges to users
- Every activity in the cloud must be logged
- Only super user can create admin accounts.
- Users cannot delete resources in cloud, they can only read.
- Admins cannot delete data. They can only write and read.

Backup policy

- Must backup data every day.
- Must backup sensitive information.
- User log
- Image file etc...
- Must backup user activity log
- Must conduct auditing every month.

Network policy

- Must use VPN to connect to the cloud.
- Every data that transferred through VPN tunnel must be encrypted.
- Encryption must be done using 256-bit length key.

OS security policy

- Must update OS automatically.
- Before update, backup the OS is necessary.
- Virus guard must be UpToDate.

3. TESTING AND RESULTS

Encryption algorithm, light weighting algorithm and client-server image uploading were tested on AsusVivoBookS510U Laptop. Tested VM is Kali Linux 2020 as the cloud with 2GB RAM and 1 core i7 processor with 1.80GHZ speed and the key for encryption and decryption is 256-bit long.

Table 1 Time for encryption

Size of the image (KB)	Average Time elapsed for encryption	
	AES Original (In Seconds)	AES Hybrid (In Seconds)
10	0.24	0.17
18	0.42	0.31
51	1.23	0.93
150	3.90	2.67
550	14.43	10.20
1702	44.47	31.30
4180	111.26	76.96
4892	128.07	89.68

Below graph is the analysis of the above table. It shows that the encryption of the AES hybrid encryption algorithm is more efficient than the original AES algorithm.

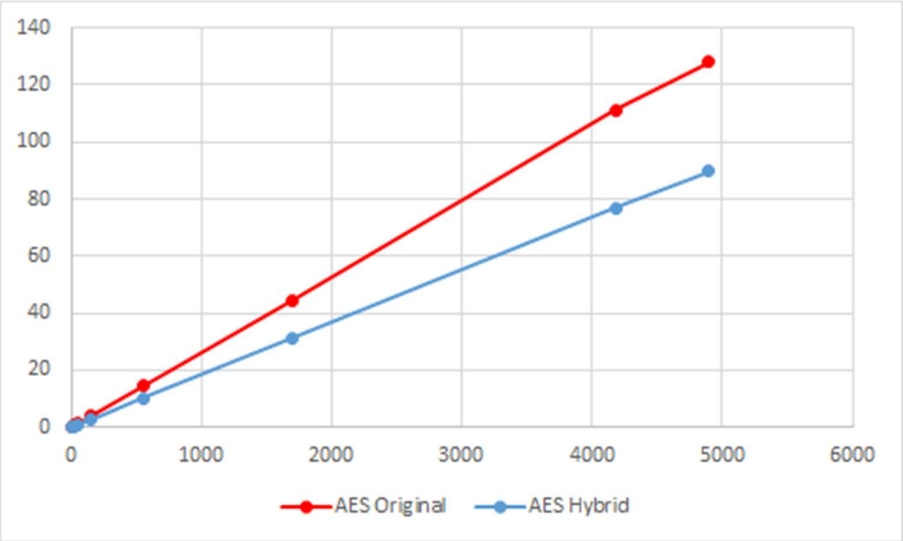

Figure 6 Encryption Speed

Table 2 Time for Decryption

Size of the image (KB)	Average Time elapsed for Decryption	
	AES Original (In Seconds)	AES Hybrid (In Seconds)
10	0.37	0.16
18	0.57	0.31
51	1.84	0.86
150	4.99	2.56
550	18.48	9.47
1702	58.31	30.60
4180	150.25	78.19
4892	174.06	92.23

Below graph is the analysis of the above table. It shows that the decryption of the AES hybrid encryption algorithm is more efficient than the original AES algorithm.

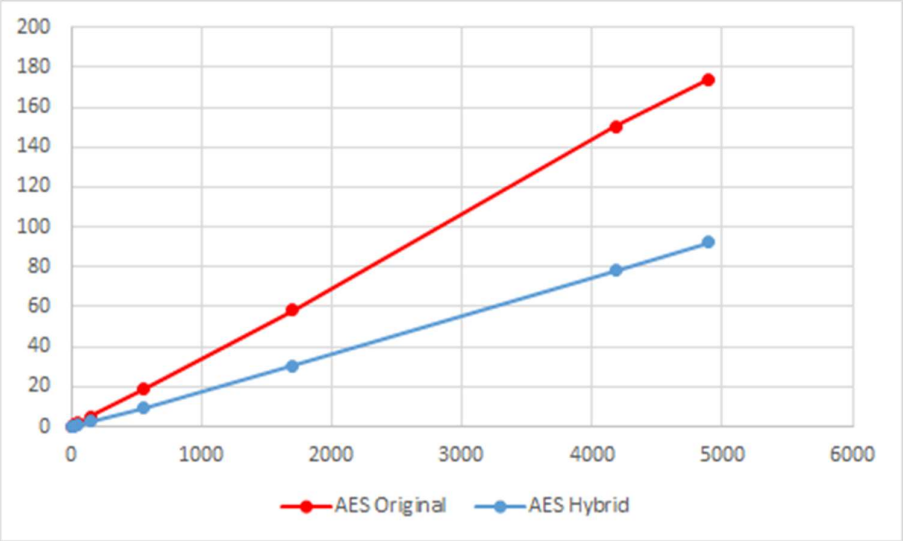

Figure 7 Decryption Speed

Table 3 Average speed for uploading

Size of the image (KB)	Average time elapsed for uploading the image	
	Without Segmentation (In Seconds)	With Segmentation (In Seconds)
1967	7.22	2.78
2890	8.88	3.97
3234	10.99	4.98
3980	11.76	5.32
4544	13.44	7.89
4765	14.05	9
4892	14.28	9.97

Below graph is the analysis of the above table. It will show the time difference of sending the full image to the cloud without splitting it into segments and sending the segments to the cloud after splitting the image into segments. According to the below graph, sending the image segments like a stream is faster than sending the full image at once.

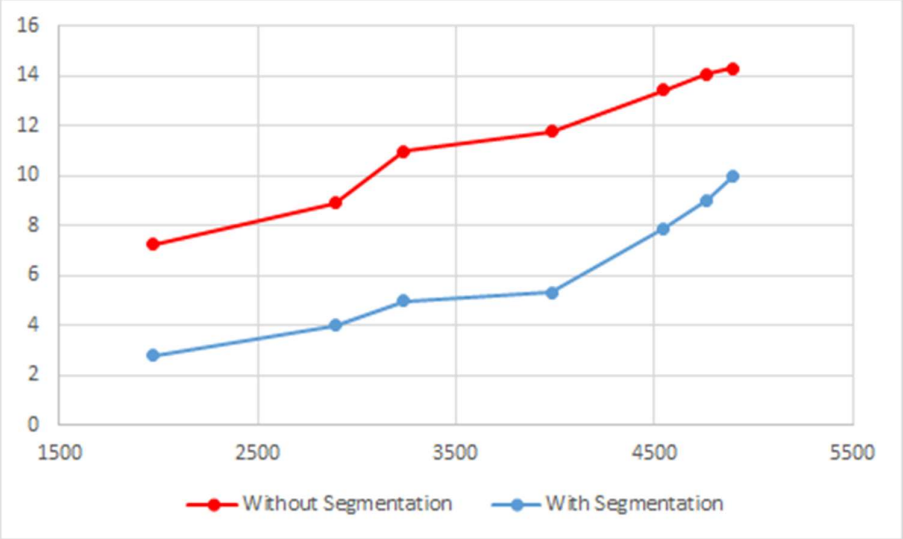

Figure 8 Upload Speed

4. CONCLUSION

Above result shows that AES hybrid encryption is faster than the original and streaming image as segments is faster than sending full image at once. Even though client send full image at once, the server receives it as pieces. Because the maximum amount server can take per once is 40000-bits. Because of that sending full image at once is highly unstable and highly insecure.

5. REFERENCES

[1] “Viber Goes fully encrypted in name of privacy—or whatsapp,IMESSAGE competition ” , Adam Toobin , Available: https://www.inverse.com/article/14466-viber-encryption

[2] “Which Cryptography algorithm is used in WhatsApp end-to-end security?” , Munkeyoto, Available: https://security.stackexchange.com/questions/120238/which-cryptography- algorithm-is-used-in-whatsapp-end-to-end-security

[3] “Telegram Messenger: Security Overview”, INFOESEC , Available: https://www.secjuice.com/telegram-messenger-security-overview/

[4] Csrc.nist.gov. (n.d.). [online] Available at: https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/ fips-197.pdf [Accessed 3 Mar. 2020.