Comparative Study and Analysis of AES and DES Algorithms on Various Multimedia Platforms.

A PROJECT REPORT

Submitted by

TARUSHI JAIN, 18BCE0859
SHAURYA SINGH, 18BCE2017
KARTIKAY GUPTA, 18BCE2199

Course Code: CSE3501
Course Title: Information Security Analysis and Audit

Under the guidance of
Anil Kumar K
Designation, School,
VIT University, Vellore.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November 2020

ACKNOWLEDGEMENT

We want to acknowledge our respected professor Anil Kumar K who helped us in all research and assisted us in all the way he could have done. He has been the backbone by supporting us in all the possible ways and made us this far to achieve our result. We thank our prestigious university for giving us this wonderful opportunity for proving us whom we are and not limiting theories to just books but also in the involvement of various mini projects. We consider this opportunity as a chance to deliver our token of thanks to VIT for providing us immense facilities and equipment required to make our project a great success. We take this time to express all our gratitude towards all the staff members who helped us during the course of the project.

TABLE OF CONTENTS

1. Introduction

1.1 Abstract

Cryptography is a critical apparatus for the assurance of multimedia content. All the Multimedia files are scrambled before being circulated over the web. Because of the Encryption of the file, it is impossible for everyone to get the keys. So the key for the decoding of the substance ought not to be unveiled to any other person other than the substance supplier. Encryption is an approach to shield data from undesirable assaults by transforming it into a shape that can't be perceived by any aggressors. Information encryption for the most part is changing of the information, for example, content, picture, sound, and so forth with the goal that it is indistinguishable, undetectable or invulnerable amid the transmission. So with a specific end goal to recoup the first information the receiver just inverses information encryption known as data decryption. The problem that we are dealt for this project is to know, understand and get acquainted with the important encryption algorithm related to the security of the documents more specifically the ones that are subject to the transfer between 2 separate entities. For the purpose of this project we need to study the AES, DES and RSA algorithm and their nature, the way in which it works, how it implements over different types of files and gives us an output in different time frames with appropriate results. Moreover, we need to study and use the Eclipse tool which helps us execute the encryption algorithm with subtle variations in the file types and providing a Java platform to execute it.

2. Background

Cryptographic techniques play crucial role when users exchange information. When multimedia contents are shared among the users, it faces security threats. Usually multimedia contents take much space. Encryption technique should be time efficient. In this work we consider four encryption techniques: Blowfish, AES, XOR and RSA and four types of media content: text, image, audio and video. Simulation shows that AES is time efficient than others. Comparing between symmetric and asymmetric cryptography, symmetric cryptographic techniques take less time than asymmetric technique.

Cryptography also termed as “secret writing” is a science of concealing information so that only the intended parties can have access to the private information. It protects the privacy and modification of data which may occur due to active and passive attacks in the channel. Cryptography consists of two things – Plain text and Ciphertext .Plain text in the original data which the sender intends to send and Ciphertext is the encrypted format of the plain text. The plain text is converted to the Ciphertext and vice versa with the help of an encryption and decryption algorithm. The encryption- decryption algorithms are mainly classified into two type e.g. symmetric key algorithm and asymmetric key algorithm. In this paper, different encryption algorithms are discussed along with their applications. In traditional end-to-end security solutions, only the end hosts can verify the validity and integrity of the traffic, which leads to several problems. First, they are not effective if the network infrastructure itself is under attack and unable to deliver packets. Second, they are not enough to provide sufficient security by themselves. The large amount of traffic on the current Internet is just unsolicited e-mail or other garbage, and the culprits behind attacks are rarely caught. Therefore, we feel that there is a clear need for security solutions, where the security policies are applied at every hop as the packet travels through the network. If the network infrastructure can verify the validity of the traffic, countermeasures against various attacks could be taken within the network, and not only by the end hosts. This would allow attacks to be stopped quickly and more efficiently, and would increase the chance of catching perpetrators.

Encryption is an approach to shield data from undesirable assaults by transforming it into a shape that can't be perceived by any aggressors. Information encryption for the most part is changing of the information, for example, content, picture, sound, and so forth with the goal that it is indistinguishable, undetectable or invulnerable amid the transmission. So with a specific end goal to recoup the first information the receiver just inverses information encryption known as data decryption.

The encryption process can be described as C = E (P, K) Where, P = Original data E = Encryption Algorithm

K = Encryption Key

C= Cipher message, which is transmitted and can be subject to attack

The decryption procedure can be described as P= D (C, K) Where, C = Cipher message; D= Decryption Algorithm K= Decryption Key; P= Recovered data

2.1 Symmetric key cryptosystems

All the classical cryptosystems that were developed before 1970 are an example of symmetric key cryptosystems. Besides that, most of the cryptosystems developed after 1970 are symmetric [3]. Some of the very popular examples of modern symmetric key include:

1. AES(AdvancedEncryptionStandard)

2. DES(DataEncryptionStandard)

All symmetric keys have a common interest, they depend on a secret shared between communicating parties. The secret can be used as both encryption and decryption key. The disadvantage of symmetric key is that it is not able to handle a large network of communication.

On the other hand, the symmetric key requires a smaller size for the same level of security as public key cryptosystems, thus, making the communication faster and memory required smaller.

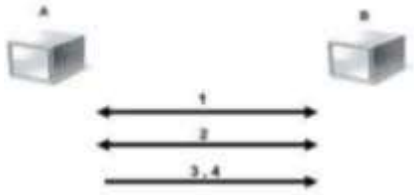


Figure 1. Symmetric Encryption

1. A and B agree on cryptosystems.
2. A and B agree on the key to be used.
3. A encrypts message using the shared key.
4. B decrypts the cipher message using the shared key.

3.Overview and Planning

3.1 Proposed Work

We preprocess the documents to represent them suitable for learning algorithm. Then we will implement an algorithm to learn the training documents to generate the document classifier. In our project, association rule mining algorithm is applied to generate the associative classifier. In testing stage, we put the new documents into the document classifier and get the classified documents. In project implementation, there are three parts: data preprocessing, generate association classifier and validation.

3.2 Working Model

We first of all preprocess the algorithms and the documents to fit accordingly to the algorithms now to put it in the right form we compare the design component of the algorithms to compare and see which algorithm works best. AES and triple DES (TDES OR 3DES) are the most usually utilized block figures. By outline, AES is quicker in term of rounds, i.e., exchanging between the equipment is less complex than exchanging between programming. Notwithstanding, DES encodes information in 64 bit and uses a 56 bit key, because of which it has an estimated probability of 72 quadrillion . Despite the fact that the number is tremendous, because of the computing energy of current technology, it isn't adequate and can be presented to assaults. Along these lines, because of DES not having the capacity to stay aware of the technology progression, it isn't a fitting security. Due to the tremendous utilization of DES, the speediest arrangement was to refresh to 3DES, which is sufficiently secure for current technology. The Rijndael calculation has been supplanted 3DES. The fundamental purpose behind Rijndael to be picked as the cutting

edge AES may be: i. Security ii. Software and Hardware Performance iii. Suitability in restricted-space environments iv. Resistance to power analysis and other implementation attacks

Factors	AES	DES
Key length	128, 192 or 256 bits	56 bits
Cipher type	Symmetric block cipher	Symmetric block cipher
Block size	128, 192, or 256 Bits	64 bits
Cryptanalysis resistance	Strong against differential, linear, interpolation and square attacks.	Vulnerable to differential and linear cryptanalysis; weak substitution tables
Security	Considered secure	Proven inadequate
Possible keys	2128,2192 or 2256	256
Times required checking all possible keys 50 billion Keys per second	For a 128-bit key 5×10^{21} years	For a 56-bit 400 days
Rounds	10,12 and 14 for 128, 192 And 256-bits respectively	16

Table 1 Differences between AES and DES

3.3 Design description

3.3.1 AES

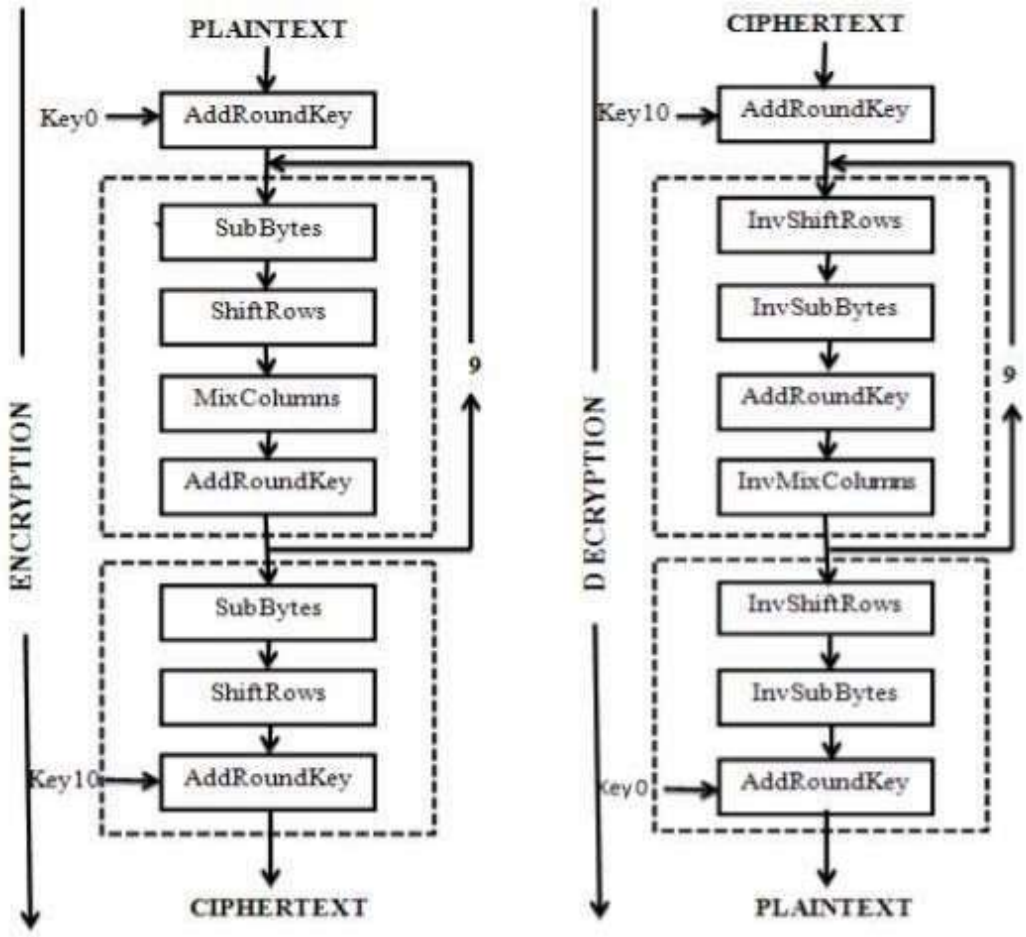


Figure 4. Flowchart of AES Encryption and Decryption

3.3.2 DES

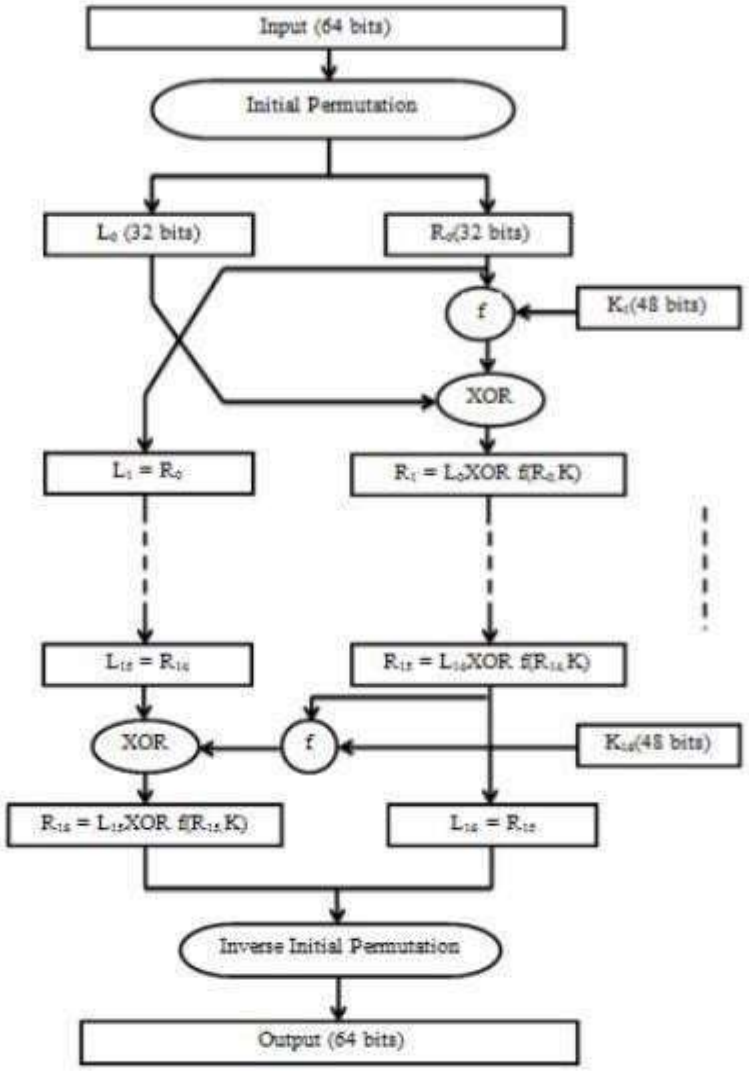


Figure 5. Flowchart of DES Encryption

3.4 Hardware requirements:

A computer with an Intel Core i series CPU and a GPU

3.5 Software requirements:

Windows 10, Java Runtime Environment, JDK, Eclipse

4.Literature Summary and Review

4.1 Literature Survey

[1] Performance Evaluation of Cryptographic Algorithms: AES and DES, Divya Sukhija

Student at JCD College of Engineering, India.

In this paper author discuss about strength and weakness of these algorithms by using various sources and explains about cryptography. Cryptographic algorithms are also known as encryption algorithms. It is a mathematical procedure for performing encryption of data. Through the use of an algorithm, information is made into meaningless cipher text and require the use of a key to transform the data back into its original form. There are a number of encryption algorithms available to encrypt the data. Their strengths depend upon the cryptographic system. Any computer system which involves cryptography is known as cryptographic system, the strength of encryption algorithm heavily relay on the computer system used for the generation of keys

Pros and cons:

Advanced Encryption Standard (AES): • AES is highly efficient, secure and it is not complex. • It needs more processing. • It requires more rounds of communication as compared to DES.

Data Encryption Standard (DES): • DES has been around a long time since1978. and has been studied to death.even now no real weakness have been found. • The most efficient attack is still brute force.The 56 bit key size is the biggest defect. • Hardware implementations of DES are very fast; DES was not designed for software and hence runs relatively slowly.

[2] IITM Journal of Management and ITSOUVENIRNational Conference on Emerging Trends in Information Technology-Cyber Security.

In this paper author discuss about how aes and des works in ATM. This paper based on AES and DES cryptographic algorithm technique, how DES at some place used in ATM and AES is more secure than DES so at everywhere in ATM AES algorithm should be used.

ATM uses secret key ,called the PIN key ,to derive the PIN from the account number in terms of algorithm known as DES.Theresult is natural PIN ,an offset can be added to it and then final PIN which the customer enter. The offset has no cryptographic function, it just used for customer to choose their own PIN

ATM using DES has been breached 24 hours. Advanced encryption standard (AES)is recent and new encryption algorithm.AES support AES with CBC (cipher block chaining) mode to IP security . Usually DES is use to encrypt the ATM transaction but most of time need more secure triple

·DES There are many illegal withdrawals take place from ATM.RossAnderson,a researcher

investigated various cases of illegal withdrawals and exposing errors in bank security. There have many cases in which criminals used fake machines, attached keypads or card readers to real machines, and record customer's PIN and bank account details to access the accounts illegally.

[3] An Advanced Security Analysis by Using Blowfish Algorithm R. Vasantha1, Dr. R. Satya Prasad2 1Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, India 2Associate Professor, Department of CSE, Acharya Nagarjuna University, Guntur, India

In this paper author discuss about security of information in cloud. Cloud enrolling appears to be to a great degree accommodating organization for a few people; every third individual is using cloud in different ways. As a result of its versatility, various individuals are trading their data to cloud. Disseminated registering exhibit a greatly viable application for affiliations. Since affiliations have broad measure of data to store and cloud gives that space to its customer and moreover empowers its customer to get to their data from wherever at whatever point easily. As people are saving their own and basic data to fogs, so it transforms into a vital issue to store that datasecurely. Numerous calculations exist for the information security like DES, AES, and Triple DES. These are symmetric key calculations in which a solitary key is utilized for encryption and decoding.

Each message comprises of various hash esteem, yet the hashing has one downside i.e. once the information is scrambled, it can't be unscrambled. This confinement of hashing was evacuated by symmetric and awry calculations. Symmetric calculation is otherwise called "Mystery Key Encryption Algorithm" in symmetric key calculation, just a single key is utilized for encryption and unscrambling i.e. private key, where as in hilter kilter calculation both open and private keys are utilized for encryption and decoding, uneven calculation is otherwise called "Open Key Encryption Algorithm". AES works quick on both programming and equipment.

[4] Simulation of Image Encryption using AES Algorithm. P.Karthigaikumar Asst.Professor (SG) Department of Electronics and Communication Karunya University,Coimbatore.

In this paper comparative study of these existing techniques has been presented also present types of images and different techniques of image processing with steps used to process an image. Security in transmission of computerized pictures has its significance in today's picture interchanges, because of the expanding utilization of pictures in modern process, it is fundamental to shield the private picture information from unapproved get to, Image security has turned into a basic issue. The troubles in guaranteeing people security turn out to be progressively testing. Different techniques have been explored and created to secure information and individual protection. Encryption is likely the clearest one. With a specific end goal to shield significant data from undesirable readers, picture encryption is basic. In these section different techniques for image processing to provide secure image processing proposed by various researchers has been reviewed.

DES Pros and Cons:

It was a secure algorithm till the 1970's. It was based on the hardware implementation, hence it runs fast.

It was easily cracked by the Brute Force attack and described as a weak algorithm in terms of security. Its software implementation cannot be described.

AES Pros and Cons:

It provides the better security as compared to the DES, TDES, IDEA, BLOWFISH, etc. algorithms.

It is free of cost.

It provides the better security as compared to the DES, TDES, IDEA, BLOWFISH, etc. algorithms.

It is free of cost.

[5] "Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System" HoWon Kim, Member, IEEE, and Sunggu Lee, Member, IEEE.

This paper entirely talks about the design and implementation of a crypto processor, a special-purpose microprocessor optimized for the execution of cryptography algorithms. This crypto processor can be used for various security applications such as storage devices, embedded systems, network routers, security gateways using IPSec and SSL protocol, etc. The crypto processor consists of a 32-bit RISC processor block and coprocessor blocks dedicated to the AES, KASUMI, SEED, triple-DES private key crypto algorithms and ECC and RSA public key crypto algorithm. The dedicated coprocessor block permits fast execution of encryption, decryption, and key scheduling operations.

Here the high performance and high flexibility of the crypto processor design makes it applicable to various security applications such as storage devices, embedded systems, network routers, security gateways for IPSec and SSL protocol processing, etc.

It can also be implemented to develop additional high performance public key crypto blocks. Also, to enhance the security of the crypto processor, it can be devised to side channel attack resistant techniques in the private and public key crypto blocks.

[6] Comparative Analysis of AES and RC4 Algorithms for Better Utilization Nidhi Singhal1 , J.P.S.Raina2 Department of Electronics & Communication, BBSB engineering college, Fatehgarh Sahib, Punjab,India.

In this paper they compared the AES algorithm with different modes of operation (block cipher) and RC4 algorithm (stream cipher) in terms of CPU time, encryption time, memory utilization and throughput at different settings like variable key size and variable data packet size. And we can conclude from this paper that it is better to use symmetric crypto algorithms rather than asymmetric ones for better efficiency as they are quick and fast to response while getting executed in a system.

Results:

- Here they compared AES and RC4 algorithms for encryption time over different packet sizes and it is resulted that RC4 took less time for encryption beating AES in efficiency. Also, when compared the decryption time, RC4 takes less time with respect to AES and when tested against varying key sizes from 128 bits to 192 bits to 256 bits, encryption time for RC4 is almost constant and is less then AES. Hence it consumes less power w.r.t AES. But for different modes of AES, encryption time increases as key size increases and the results hold same with the decryption one.
- The second result was about the Throughput for AES and RC4 with different key size. The result shows the superiority of RC4 over AES. With different key sizes RC4 gives almost the same result. But for different modes of AES, throughput decreases as key size increases because of more usage of computational power and encryption characteristics. Thus, RC4 is fast in nature and consume less power w.r.t its counterparts. Better results were obtaining in decryption w.r.t encryption.
- In the result we talk about the memory utilization for AES and RC4 with different file size. As per the result AES consume more memory w.r.t.RC4 because of its characteristics. And as the file size increases memory size is drastically increased in AES means for extra-large files, we need a system with good memory and more CPU.

[7] An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists AJ Elbirt1, W Yip1, B Chetwynd2, C Paar1 ECE Department, Worcester Polytechnic Institute 100 Institute Road Worcester, MA 01609, USA.

This paper explores about the hardware implementations of encryption algorithms as they provide cryptographic algorithm agility, physical security, and potentially much higher performance than software solutions. This contribution investigates the significance of FPGA implementations of the Advanced Encryption Standard candidate algorithms. Multiple architectural implementation options are explored for each algorithm.

- For each of the AES finalists, an implementation analysis for each architecture option when optimized for both area and speed was performed to determine the suitability for hardware implementation of each finalist.
- Upon comparison, it was determined that the Serpent algorithm yielded the best throughput results in both modes of operation.
- When evaluating throughput per slice, the Serpent algorithm was also found to yield the best results when operating in non-feedback mode while the Rijndael algorithm was found to yield the best results when operating in feedback mode.
- The Serpent algorithm is clearly the best choice when throughput is the key evaluation characteristic regardless of the mode of operation.

[8] A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security by Gurpreet Singh M.Tech Research Scholar, Department of Computer Science and Engineering Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India.

Encryption is the process of scrambling a message so that only the intended recipient can read it. Encryption can provide a means of securing information. As more and more information is stored on computers or communicated via computers, the need to insure that this information is invulnerable to snooping and/or tampering becomes more relevant. With the fast progression of digital data exchange in electronic way, Information Security is becoming much more important in data storage and transmission. Information Confidentiality has a prominent significance in the study of ethics, law and most recently in Information Systems. With the evolution of human intelligence, the art of cryptography has become more complex in order to make information more secure. Arrays of Encryption systems are being deployed in the world of Information Systems by various organizations. In this paper, a survey of various Encryption Algorithms is presented.

This paper presents a detailed study of the popular Encryption Algorithms such as RSA, DES, 3DES and AES. The use of internet and network is growing rapidly. So there are more requirements to secure the data transmitted over different networks using different services. To provide the security to the network and data different encryption methods are used. In this paper, a survey on the existing works on the Encryption techniques has been done. To sum up, all the techniques are useful for real-time Encryption. Each technique is unique in its own way, which might be suitable for different applications and has its own pro's and con's. According to research done and literature survey it can be found that AES algorithm is most efficient in terms of speed, time, throughput and avalanche effect. The Security provided by these algorithms can be enhanced further, if more than one algorithm is applied to data. Our future work will explore this concept and a combination of algorithms will be applied either sequentially or parallel, to setup a more secure environment for data storage and retrieval.

[9] Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System HoWon Kim, Member, IEEE, and Sunggu Lee, Member, IEEE.

This paper presents the design and implementation of a crypto processor, a special-purpose microprocessor optimized for the execution of cryptography algorithms. This crypto processor can be used for various security applications such as storage devices, embedded systems, network routers, security gateways using IPSec and SSL protocol, etc. The crypto processor consists of a 32-bit RISC processor block and coprocessor blocks dedicated to the AES, KASUMI, SEED, triple-DES private key crypto algorithms and ECC and RSA public key crypto algorithm. The dedicated coprocessor block permits fast execution of encryption, decryption, and key scheduling operations. The 32-bit RISC processor block can be used to execute various crypto algorithms such as Hash and other application programs such as user authentication and IC card interface. The crypto processor has been designed and implemented using an FPGA, and some parts of crypto algorithms have been fabricated as a single VLSI chip using 0.5μm CMOS technology. To test and demonstrate the capabilities of this chip, a custom boardproviding real-time data security for a data storage device has been developed.

For future work, they plan to develop additional high performance public key crypto blocks. Also, to enhance the security of our crypto processor, we will devise side channel attack resistant techniques in the private and public key crypto blocks.

5. System Implementation

5.1 Description of the code/program

In this program we run 2 different algorithm namely AES and DES then we find the file that can be any file type example PDF, doc, txt, xlsx, etc then we calculate the time of finding the file by different algorithms and thus conclude which is the best algorithm.

5.2 Source Code

```
package cyber; import java.io.File; import java.io.FileInputStream; import
java.io.FileOutputStream; import java.security.Key; import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec; public class cryptog { private static long
startTime = System.currentTimeMillis(); static void fileProcessor(int cipherMode
,String key,File inputFile,File outputFile){ try {
Key secretKey = new SecretKeySpec(key.getBytes(),"AES");
Cipher cipher = Cipher.getInstance("AES"); cipher.init(cipherMode,
secretKey);
FileInputStream inputStream = new FileInputStream(inputFile); byte[]
inputBytes = new byte[(int) inputFile.length()]; inputStream.read(inputBytes);
byte[] outputBytes = cipher.doFinal(inputBytes);
FileOutputStream outputStream = new FileOutputStream(outputFile);
outputStream.write(outputBytes); inputStream.close(); outputStream.close();
} catch (Exception e) { e.printStackTrace();
}
}

public static void main(String[] args) {
String key = "This is a secret";
```

```
File inputFile = new File("C:\\Users\\Resham B\\Documents\\DSC_0258.jpg");

File encryptedFile = new File("text.encrypted"); File decryptedFile = new File(

"decrypted-text.jpg"); try { cryptog.fileProcessor(Cipher.ENCRYPT_MODE,key,

inputFile,encryptedFile); cryptog.fileProcessor(Cipher.DECRYPT_MODE,key,

encryptedFile,decryptedFile);

System.out.println("Success");

} catch (Exception ex) {

System.out.println(ex.getMessage()); ex.printStackTrace();

}

long endTime = System.currentTimeMillis();

System.out.println("It took " + (endTime - startTime) + " milliseconds");

}

}
```

5.3 Snapshot/Execution of the Program

```
 1 e crypto;
 2* java.io.File;
 8  class cryptog {
 9=ivate static long
10 artTime = System.currentTimeMillis();
11=atic void fileProcessor(int cipherMode ,String key,File inputFile,File outputFile)
12
13   try {
14           Key secretKey = new SecretKeySpec(key.getBytes(),"DES");
15           Cipher cipher = Cipher.getInstance("DES");
16           cipher.init(cipherMode,secretKey);
17           FileInputStream inputStream = new FileInputStream(inputFile);
18           byte[]
19           inputBytes = new byte[(int) inputFile.length()];
20           inputStream.read(inputBytes);
21           byte[] outputBytes = cipher.doFinal(inputBytes);
22           FileOutputStream outputStream = new FileOutputStream(outputFile);
23           outputStream.write(outputBytes);
24           inputStream.close();
25           outputStream.close();
26   } catch (Exception e)
27   {
28       e.printStackTrace();
29   }
```

Problems Javadoc Declaration Console Coverage

```
<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 7:09:00 pm – 7:09:01 pm)
java.security.InvalidKeyException: Wrong key size
        at java.base/com.sun.crypto.provider.DESCrypt.init(DESCrypt.java:536)
        at java.base/com.sun.crypto.provider.ElectronicCodeBook.init(ElectronicCodeBook.java:97)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:593)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:469)
        at java.base/com.sun.crypto.provider.DESCipher.engineInit(DESCipher.java:186)
        at java.base/javax.crypto.Cipher.implInit(Cipher.java:869)
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:931)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 201 milliseconds
```

5.4 Results

5.4.1 Statistics of the run time execution of AES:

1. **XLSX file**

Runtime 93ms

Used memory is bytes: 1675464

Used memory is megabytes: 1

Code:

```
11⊖    static void fileProcessor(int cipherMode ,String key,File inputFile,File outputFile)
12     {
13         try {
14                 Key secretKey = new SecretKeySpec(key.getBytes(),"AES");
15                 Cipher cipher = Cipher.getInstance("AES");
16                 cipher.init(cipherMode,secretKey);
17                 FileInputStream inputStream = new FileInputStream(inputFile);
18                 byte[]
19                 inputBytes = new byte[(int) inputFile.length()];
20                 inputStream.read(inputBytes);
21                 byte[] outputBytes = cipher.doFinal(inputBytes);
22                 FileOutputStream outputStream = new FileOutputStream(outputFile);
23                 outputStream.write(outputBytes);
24                 inputStream.close();
25                 outputStream.close();
26         } catch (Exception e)
27         {
28             e.printStackTrace();
29         }
30     }
31⊖public static void main(String[] args) {
32     String key = "This is a secret";
33     File inputFile = new File("C:\\Users\\Dell\\Desktop\\5th SEM TOTAL.xlsx");
34     File encryptedFile = new File("text.encrypted");
```

```
Problems  Javadoc  Declaration  Console  Coverage
<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe  (26-Oct-2020, 9:22:02 pm – 9:22:03 pm)
Success
It took 93 milliseconds
```

Output:

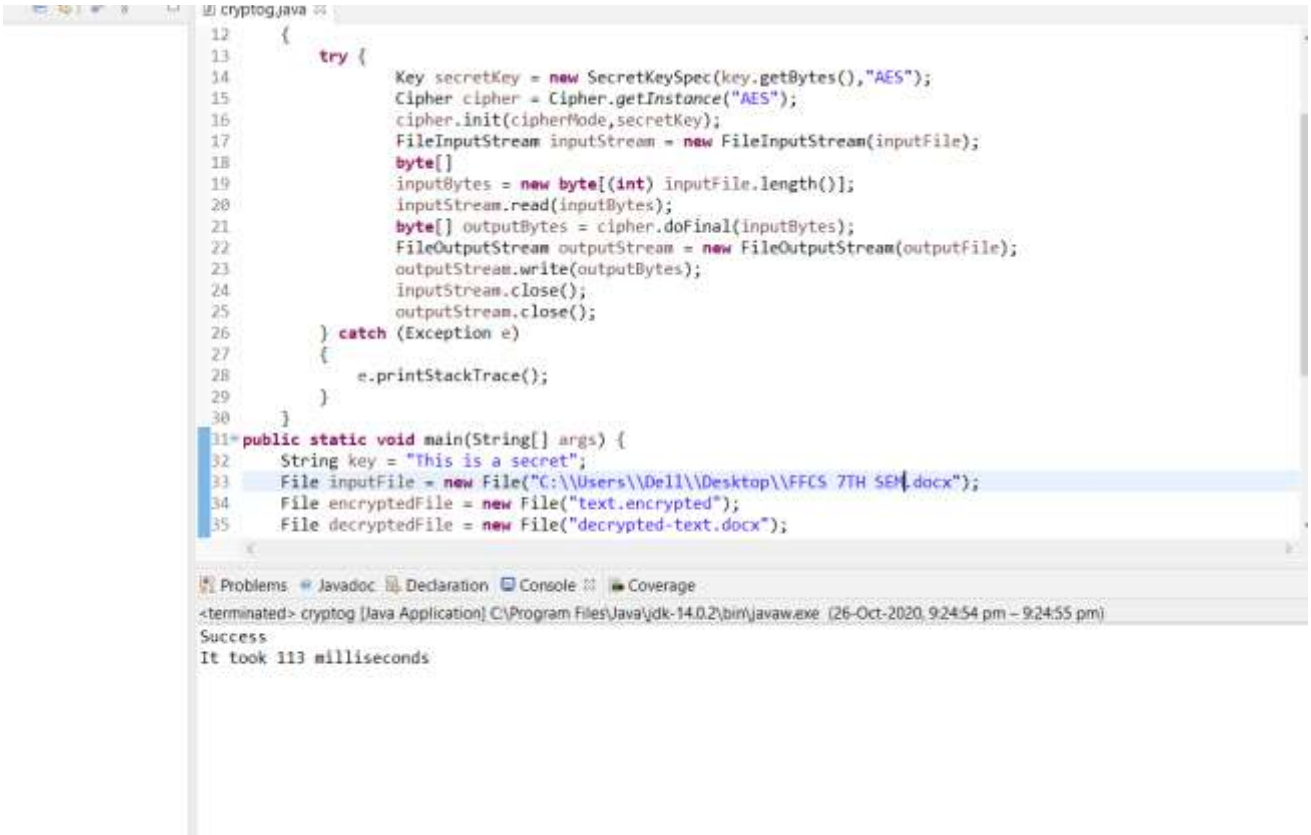
2. **Word file**

Runtime 113ms

Used memory is bytes: 1357464

Used memory is megabytes: 1

Code:



```
    {
        try {
                Key secretKey = new SecretKeySpec(key.getBytes(),"AES");
                Cipher cipher = Cipher.getInstance("AES");
                cipher.init(cipherMode,secretKey);
                FileInputStream inputStream = new FileInputStream(inputFile);
                byte[]
                inputBytes = new byte[(int) inputFile.length()];
                inputStream.read(inputBytes);
                byte[] outputBytes = cipher.doFinal(inputBytes);
                FileOutputStream outputStream = new FileOutputStream(outputFile);
                outputStream.write(outputBytes);
                inputStream.close();
                outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\FFCS 7TH SEM.docx");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.docx");
```

Problems Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:24:54 pm – 9:24:55 pm)

```
Success
It took 113 milliseconds
```

Output:

3. Pdf file

RUNTIME 107ms

Used memory is bytes: 1310392

Used memory is megabytes: 1

Code:

Output:

4. **Text file**

RUNTIME 107ms

Used memory is bytes: 1352608

Used memory is megabytes: 1

Code:

cryptog.java

```
        try {
                Key secretKey = new SecretKeySpec(key.getBytes(),"AES");
                Cipher cipher = Cipher.getInstance("AES");
                cipher.init(cipherMode,secretKey);
                FileInputStream inputStream = new FileInputStream(inputFile);
                byte[]
                inputBytes = new byte[(int) inputFile.length()];
                inputStream.read(inputBytes);
                byte[] outputBytes = cipher.doFinal(inputBytes);
                FileOutputStream outputStream = new FileOutputStream(outputFile);
                outputStream.write(outputBytes);
                inputStream.close();
                outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\slonline compiler code.txt");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.txt");
    try {
```

Problems @ Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:13:12 pm – 9:13:14 pm)

```
Success
It took 107 milliseconds
```

Output:

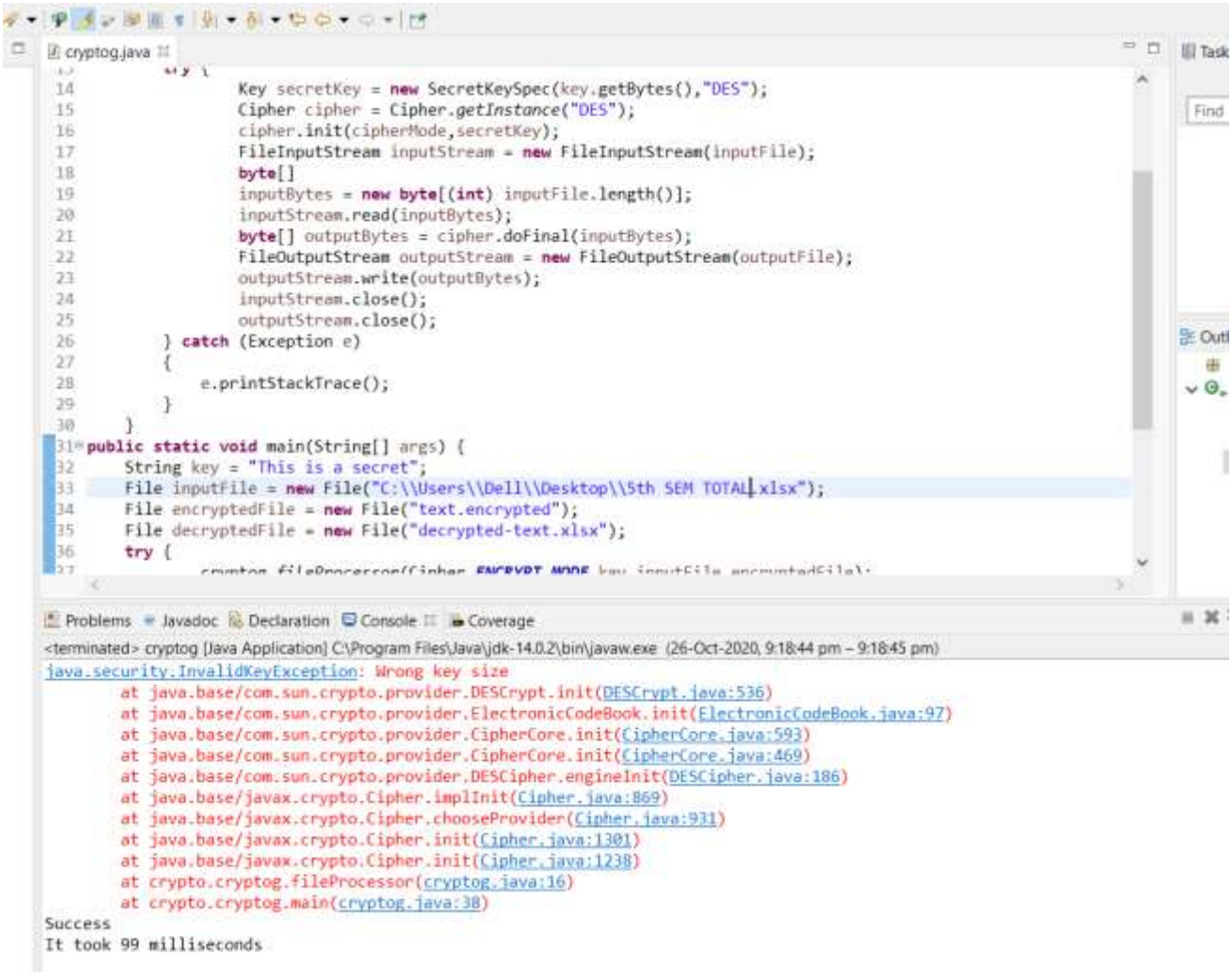
5.4.2 Statistics of the run time execution of DES:

1. XLSX file

Runtime 99ms

Used memory is bytes: 1895464

Used memory is megabytes: 3



Output:

2. Word file

Runtime 127ms

Used memory is bytes: 2337465

Used memory is megabytes: 3

Code:

```
    {
        try {
                Key secretKey = new SecretKeySpec(key.getBytes(),"DES");
                Cipher cipher = Cipher.getInstance("DES");
                cipher.init(cipherMode,secretKey);
                FileInputStream inputStream = new FileInputStream(inputFile);
                byte[]
                inputBytes = new byte[(int) inputFile.length()];
                inputStream.read(inputBytes);
                byte[] outputBytes = cipher.doFinal(inputBytes);
                FileOutputStream outputStream = new FileOutputStream(outputFile);
                outputStream.write(outputBytes);
                inputStream.close();
                outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\FFCS 7TH SEM.docx");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.docx");
```

Problems Javadoc Declaration Console Coverage

```
<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe  (26-Oct-2020, 9:27:02 pm – 9:27:02 pm)
java.security.InvalidKeyException: Wrong key size
        at java.base/com.sun.crypto.provider.DESCrypt.init(DESCrypt.java:536)
        at java.base/com.sun.crypto.provider.ElectronicCodeBook.init(ElectronicCodeBook.java:97)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:593)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:469)
        at java.base/com.sun.crypto.provider.DESCipher.engineInit(DESCipher.java:186)
        at java.base/javax.crypto.Cipher.implInit(Cipher.java:869)
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:931)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 127 milliseconds
```

Output:

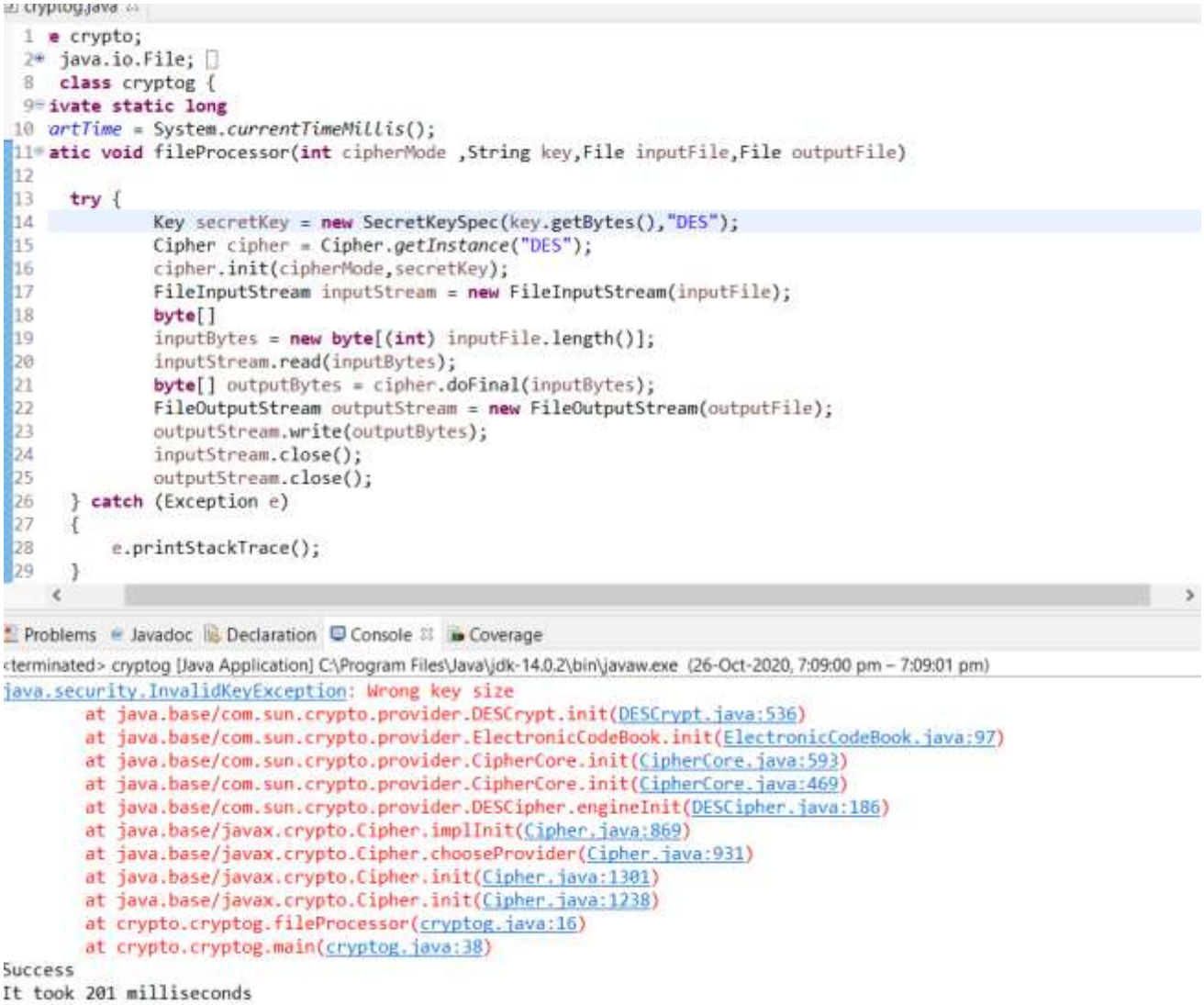
3. Pdf file

RUNTIME 201 ms

Used memory is bytes: 2336292

Used memory is megabytes: 3

Code:



```
e crypto;
java.io.File;
class cryptog {
ivate static long
artTime = System.currentTimeMillis();
atic void fileProcessor(int cipherMode ,String key,File inputFile,File outputFile)

  try {
          Key secretKey = new SecretKeySpec(key.getBytes(),"DES");
          Cipher cipher = Cipher.getInstance("DES");
          cipher.init(cipherMode,secretKey);
          FileInputStream inputStream = new FileInputStream(inputFile);
          byte[]
          inputBytes = new byte[(int) inputFile.length()];
          inputStream.read(inputBytes);
          byte[] outputBytes = cipher.doFinal(inputBytes);
          FileOutputStream outputStream = new FileOutputStream(outputFile);
          outputStream.write(outputBytes);
          inputStream.close();
          outputStream.close();
  } catch (Exception e)
  {
      e.printStackTrace();
  }
```

Problems Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 7:09:00 pm – 7:09:01 pm)

```
java.security.InvalidKeyException: Wrong key size
        at java.base/com.sun.crypto.provider.DESCrypt.init(DESCrypt.java:536)
        at java.base/com.sun.crypto.provider.ElectronicCodeBook.init(ElectronicCodeBook.java:97)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:593)
        at java.base/com.sun.crypto.provider.CipherCore.init(CipherCore.java:469)
        at java.base/com.sun.crypto.provider.DESCipher.engineInit(DESCipher.java:186)
        at java.base/javax.crypto.Cipher.implInit(Cipher.java:869)
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:931)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 201 milliseconds
```

Encrypted file

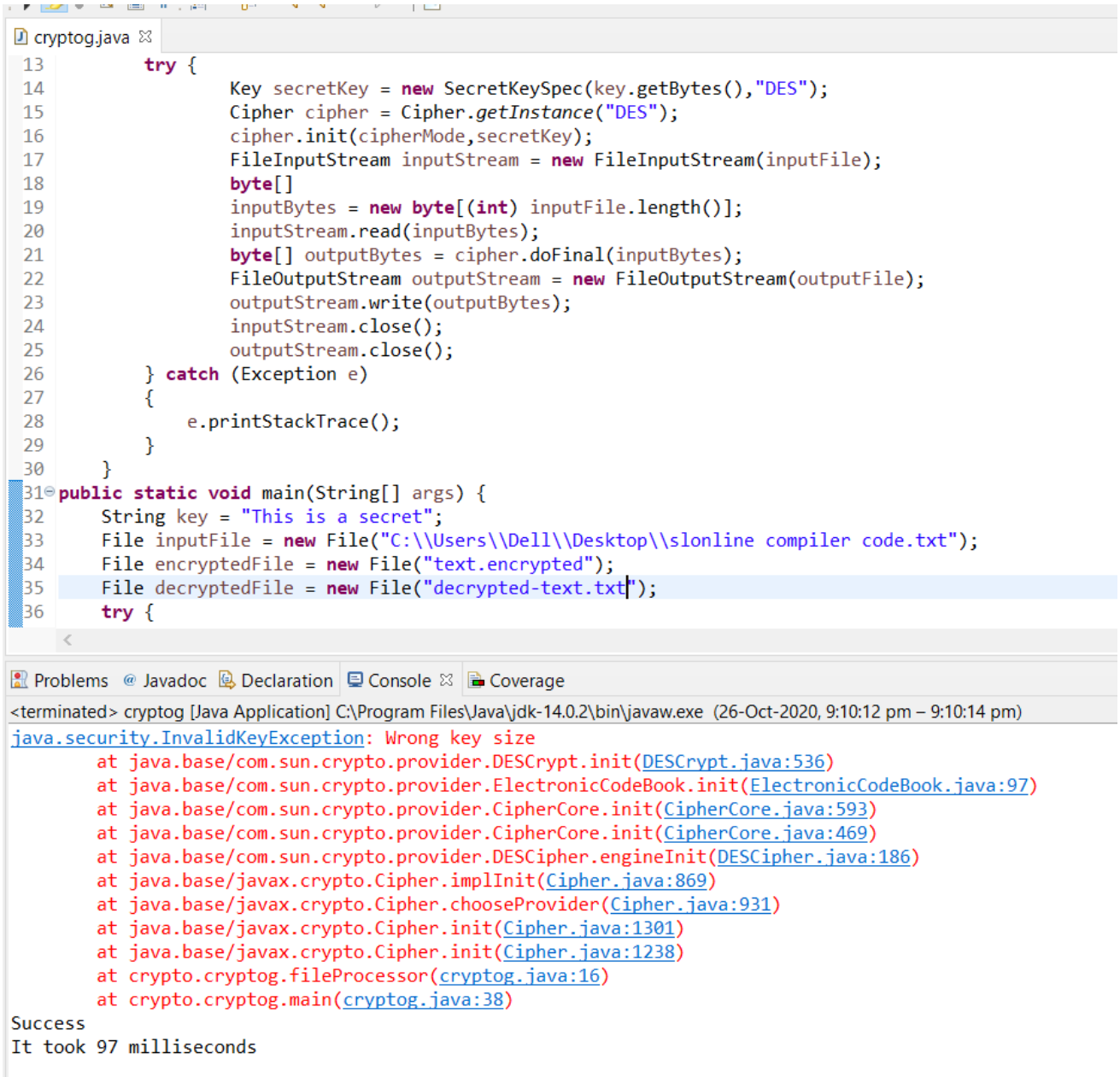
4. **Text file**

RUNTIME 97ms

Used memory is bytes: 1352574

Used memory is megabytes: 3

Code:



Output:

5.4.3 Statistics of the run time execution of RSA:

1. **Excel file**

RUNTIME 108ms

Used memory is bytes: 1352574

Used memory is megabytes: 3

Code:

```
    {
        try {
                Key secretKey = new SecretKeySpec(key.getBytes(),"RSA");
                Cipher cipher = Cipher.getInstance("RSA");
                cipher.init(cipherMode,secretKey);
                FileInputStream inputStream = new FileInputStream(inputFile);
                byte[]
                inputBytes = new byte[(int) inputFile.length()];
                inputStream.read(inputBytes);
                byte[] outputBytes = cipher.doFinal(inputBytes);
                FileOutputStream outputStream = new FileOutputStream(outputFile);
                outputStream.write(outputBytes);
                inputStream.close();
                outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\5th SEM TOTAL.xlsx");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.xlsx");
```

Problems Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:36:51 pm – 9:36:51 pm)

```
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:37)
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 108 milliseconds
```

Output:

2. **PDF file**

RUNTIME 105ms

Used memory is bytes: 1352574

Used memory is megabytes: 3

Code:

```
ry {
        Key secretKey = new SecretKeySpec(key.getBytes(),"RSA");
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(cipherMode,secretKey);
        FileInputStream inputStream = new FileInputStream(inputFile);
        byte[]
        inputBytes = new byte[(int) inputFile.length()];
        inputStream.read(inputBytes);
        byte[] outputBytes = cipher.doFinal(inputBytes);
        FileOutputStream outputStream = new FileOutputStream(outputFile);
        outputStream.write(outputBytes);
        inputStream.close();
        outputStream.close();
 catch (Exception e)

   e.printStackTrace();


atic void main(String[] args) {
g key = "This is a secret";
inputFile = new File("C:\\Users\\Dell\\Desktop\\CSE3002_INTERNET-AND-WEB-PROGRAMMING_ETH_1.1_47_CSE3002.pdf");
encryptedFile = new File("text.encrypted");
decryptedFile = new File("decrypted-text.pdf");
```

Problems Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:38:40 pm – 9:38:41 pm)

```
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:37)
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 105 milliseconds
```

Output:

3. **Text file**

RUNTIME 98ms

Used memory is bytes: 1352574

Used memory is megabytes: 3

Code:

```
            Key secretKey = new SecretKeySpec(key.getBytes(),"RSA");
            Cipher cipher = Cipher.getInstance("RSA");
            cipher.init(cipherMode,secretKey);
            FileInputStream inputStream = new FileInputStream(inputFile);
            byte[]
            inputBytes = new byte[(int) inputFile.length()];
            inputStream.read(inputBytes);
            byte[] outputBytes = cipher.doFinal(inputBytes);
            FileOutputStream outputStream = new FileOutputStream(outputFile);
            outputStream.write(outputBytes);
            inputStream.close();
            outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\sIonline compiler code.txt");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.txt");
    try {
            cryptog.fileProcessor(Cipher.ENCRYPT_MODE,key,inputFile,encryptedFile);
```

Problems Javadoc Declaration Console Coverage

<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:40:15 pm – 9:40:16 pm)

```
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:37)
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 98 milliseconds
```

Output:

4. **Word file**

RUNTIME 116ms

Used memory is bytes: 1352574

Used memory is megabytes: 3

Code:

```
    {
        try {
                Key secretKey = new SecretKeySpec(key.getBytes(),"RSA");
                Cipher cipher = Cipher.getInstance("RSA");
                cipher.init(cipherMode,secretKey);
                FileInputStream inputStream = new FileInputStream(inputFile);
                byte[]
                inputBytes = new byte[(int) inputFile.length()];
                inputStream.read(inputBytes);
                byte[] outputBytes = cipher.doFinal(inputBytes);
                FileOutputStream outputStream = new FileOutputStream(outputFile);
                outputStream.write(outputBytes);
                inputStream.close();
                outputStream.close();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
public static void main(String[] args) {
    String key = "This is a secret";
    File inputFile = new File("C:\\Users\\Dell\\Desktop\\FFCS 7TH SEM.docx");
    File encryptedFile = new File("text.encrypted");
    File decryptedFile = new File("decrypted-text.docx");
```

Problems Javadoc Declaration Console Coverage

```
<terminated> cryptog [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (26-Oct-2020, 9:34:11 pm – 9:34:12 pm)
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:37)
java.security.InvalidKeyException: No installed provider supports this key: javax.crypto.spec.SecretKeySpec
        at java.base/javax.crypto.Cipher.chooseProvider(Cipher.java:960)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1301)
        at java.base/javax.crypto.Cipher.init(Cipher.java:1238)
        at crypto.cryptog.fileProcessor(cryptog.java:16)
        at crypto.cryptog.main(cryptog.java:38)
Success
It took 116 milliseconds
```

Output:

6. Conclusion and Future Work

There are different cryptography strategies and sub-procedures. A few systems perform incomplete encryption while others perform full encryption. A portion of the strategies complete image compressions however others don't. Contingent upon the kind of application, speed, bandwidth, privacy, security and legitimacy degree, one may choose a specific sort of encryption strategy. Every single strategy has its own benefits and faults. One must think in choosing an appropriate figure since now cryptanalysis methods investigate is under core interest. Once a figure is created, one must do different security investigations specified in the project. There is enormous potential in this field for future research and arrangement. With a probability of growing such encryption plans for movement remuneration and execution on installed gadget models.

Our future work will center on looking at and breaking down existing cryptographic algorithmic blocks of encryption techniques like AES, DES and RSA. It will incorporate investigations on image and sound information and focus will be to enhance encryption time and decryption time.

7. References

[1] Borka Jerman-Blazic, Tomaz Klobucar,. Advanced Communications and Multimedia Security. New York: Springer Science+Business Media New York. 2002

[2]Ching-Yung Lin,. Topics in Signal Processing -- Multimedia Security Systems. 2006

[3]Emanuil Rednic; Andrei Toma, n.d. Software Analysis. SECURITY MANAGEMENT IN A MULTIMEDIA SYSTEM, 4(2), pp. 237-247.

[4] Saha Arunabh n.d. Overview of Multimedia Security,

[5] Amit Pande; Prasant Mohapatra; Joseph Zambreno, n.d. Securing Multimedia Content using Joint Compression and Encryption,

[6] Sonal Guleria, Sonia Vatta, TO ENHANCE MULTIMEDIA SECURITY IN CLOUD COMPUTING ENVIRONMENT USING CROSSBREED ALGORITHM, 2(6), pp. 562-568. 2013.

[7] A.A.Zaidan, B.B.Zaidan, Anas Majeed, "High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm", World Academy of Science Engineering and Technology (WASET), Vol.54, ISSN: 2070-3724, P.P 468-479.

[8] A.A.Zaidan, B.B.Zaidan, “Novel Approach for High Secure Data Hidden in MPEG Video Using Public Key Infrastructure”, International Journal of Computer and Network Security, Vol.1, No.1, ISSN: 1985-1553, P.P 71-76, 2009.

[9] A.W.Naji, A.A.Zaidan, B.B.Zaidan, Shihab A, Othman O. Khalifa, " Novel Approach of Hidden Data in the (Unused Area 2 within EXE File) Using Computation Between Cryptography and Steganography ", International Journal of Computer Science and Network Security (IJCSNS) , Vol.9, No.5 , ISSN : 1738- 7906, pp. 294-300.

[10] Anas Majed Hamid, Miss Laiha Mat Kiah, Hayan .T. Madhloom, B.B Zaidan, A.A Zaidan," Novel Approach for High Secure and High Rate Data Hidden in the Image Using Image Texture Analysis", International Journal of Engineering and Technology (IJET) , Published by: Engg Journals Publications, ISSN:0975-4042, Vol.1,NO.2,P .P 63-69.