

Big Data Security on Hadoop Open Source Frame for Healthcare Data Management using One-Time-Pad Encryption Algorithm

Alexander Agunbiade

Department of Computer science, University of Hertfordshire, United Kingdom

Corresponding author: agunbiadealex@gmail.com

doi: <https://doi.org/10.37745/ejcsit.2013/vol12n16882>

Published March 6, 2024

Citation: Agunbiade A. (2024) Big Data Security on Hadoop Open Source Frame for Healthcare Data Management using One-Time-Pad Encryption Algorithm, European Journal of Computer Science and Information Technology, 12 (1), 68-82

ABSTRACT: *The study elicited knowledge about the factors associated with one-time pad encryption/decryption with big data in healthcare; formulate an assembled algorithms model for one-time pad encryption; design and implement the system and evaluating the system performance with the view implementing big data security on Hadoop open-source framework for healthcare data. Literature was sourced to investigate the factors associated with healthcare security attacks and various consequences of breach of data. An assembled algorithm model was formulated using mathematical theory of one-time pad encryption and a model was designed using Universal Modelling Language (UML) and implemented using python programming language, Distributed File System of Hadoop, Yet Another Resource Negotiator called YARN; encryption time and decryption time was adopted for the performance metrics deployed for the evaluation of the developed system. The result showed that as the size of the files increased, the encryption/decryption time keeps increasing as well. While carryout the algorithm evaluation, two different values (file sizes) were used for testing on the Hadoop framework. Securing the healthcare (Ebola) big-data, it was observed that OTP encryption/decryption performed better compared to AES encryption/decryption in term of computational processing time of the healthcare big-data considered. Considering before/after downloading, it was observed that there was need for authentication for another level of security towards securing healthcare records on HDFS. The study concluded that, big data analytics on Hadoop is ideal for today's big healthcare data and also that One Time Pad encryption algorithm is sufficient to provide needed big healthcare data security.*

KEYWORDS: Big data, data security, Hadoop, encryption, algorithm, data vulnerabilities

INTRODUCTION

Big data, in recent times has become a trendy terminology in the world of computing. It has been described as a gigantic repository of complex and extensive structured and unstructured data sets that are swiftly generated and communicated from a wide selection of sources

(Uthayasankar Sivarajah et al., 2017). Owing to its heterogeneous sources and its nature, it is facing serious security challenges that are threatening to underscore its enormous benefits. Data has been described as the oil of today's digitized economy (Toonders, 2014). Data has become an indispensable commodity for the day-to-day running of many business activities. Virtually all areas of human endeavours, public, private and commercial life have embarked on the digitization and datafication technology train (Yusuf, 2019).

Big Data has found and has continued to receive wide acceptability be it finance and business, education, governance and military, agriculture, healthcare and so on but there are deep and broad concerns about the way in which data is gathered and subsequently put to use. Organizations gather and collate diverse background and sensitive data about their clientele. These data are subsequently subjected to some analytical processes for the purpose of extracting relevant information in order to arrive at more insightful decisions for better performance and competitive advantage.

The healthcare industry is not left out of this big data technology transformation. The foremost entities of the health industry are the health specialists (medical doctor or nurses and other hospital assistants), medical services outlets for effective administration of healthcare deliverables, and other healthy living financial establishments (health insurance) carrying out supportive roles for the professionals and service providers (Dash et al., 2019). Data being used in the healthcare system come from the afore-mentioned components ranging from patients' medical records, medical examination results, general hospital records, internet of things enabled devices (Abouelmehdi et al., 2018). Big data and healthcare holds significant importance to the wide-ranging patient care and treatment processes, biomedical research and education, general workflow management, tangible decline in the cost of healthcare services and satisfactory progression in well-being (Belle Ashwin et al., 2015).

These benefits notwithstanding, there has been growing concerns on the extent to which personal data can be used without infringing on the security and privacy of those concerned. Addressing the security issues being encountered in the use of big data should go a long way in promoting its acceptability. A major security concern with big data analytics has to do with a massive aggregate of critical data of persons collected and analysed by corporate entities often times competing organisations for varying purposes. Consequently, every proactive organization should put in place a security mechanism to prevent irreparable security incidence. Big data phenomenon has come to stay and the future holds so much in stock for it therefore the issue of its security must be strategically addressed. In order to achieve this, it is important that identified gaps within the framework be filled and this calls for further future researches. This research work survey security challenges of big data in endpoint vulnerabilities, access controls and authentication as applied to the healthcare industry and also explore ways in which they may be alleviated.

LITERATURE REVIEW

Encryption is a form of technology and it involves the process of transforming data or information and messages in readable and meaningful forms known as plaintext into a form

that is not readable and often times meaningless to unintended recipient. The non-readable form is referred to as ciphertext. Its implementation involves the use of algorithms that help to encrypt and decrypt data (Olufohunsi, 2019). Data encryption technologies are being employed for the purpose of protecting and preserving healthcare data from unauthorized access. Basically there are two essential types of encryption. These are: symmetric encryption and asymmetric encryption.

The use of symmetric encryption approach involves the use of a single symmetric key for the purpose of encrypting and decrypting data. The encryption key will be shared with only authorized entities. These are the people that will be able to decrypt the messages. Symmetric encryption has been found to be faster in execution. It is widely used where and when processing speed of transactions is vital. It is less resource-dependent when compared with the asymmetric encryption. Symmetric encryption is considered as less secure (Harshit, 2020). Asymmetric encryption is known as public key cryptography. The implementation of this encryption method involves the use of two separate individual keys in which one of the keys is designated as the public key which is shared with everyone and the other key known as private key. The private key is known only by the entity who generated the key. The process involves the use of the public key to encrypt the data while the private key is used to decrypt the data (Lydia et al. 2018).

Worthy of note is that the encryption method is designed to offer improved security than symmetric encryption. The use of asymmetric method can be considered excessive in some instances, it can slow down the processing of some transactions, machines and networks. The application of asymmetric method in healthcare record management will involve the use of the two the two secret keys; private and public keys. The public key is used to encrypt the information (original message) from the source while the private key will be required to decrypt the encoded message at the other end (Lydia et al. 2018).

Data encryption technology has been identified as one of the popular system for the preservation of data security and privacy medical records inclusive. At the moment, there has not been an established encryption technique specifically for healthcare data security notwithstanding the afore-mentioned encryption techniques have gained reasonable level of applicability for encryption purposes. The likes of Rijndael, Twofish and Serpent have been found to be most applicable. The highlighted symmetric based encryption standards are implemented using the 128 to 256 bits key system Naemabadi et al. (2018).

Implementing these encryption methods usually involve series of intricate procedures before encryption can take successfully take place. The procedures also entail the implementation of block cipher design which has been found to be error prone. While data stream is being transmitted, propensity to error is considerably increased. Naemabadi et al. (2018).

Theoretically, one-time pad as an encryption technique has been described as the most secured system that demands the use of a one-time pre-shared key for data encryption (Kaicheng 2003). It is said to be undecipherable as long as certain conditions are met. The encryption scheme entails the generation of a random sequence of secret keys (one-time pad) that are used once in

combination with a plaintext and the needed encryption is thereafter implemented. The receiver at the other end is required to use the same key combination to decrypt the now encrypted text (ciphertext). Any third parties can decode the plaintext if and only if access is granted to the randomly generated keys, where there is no such access the ciphertext remains decrypted. Moreover, the key is meant to be used once (never reused), must be according to the specified length and randomly generated. Communication is an integral aspect of human endeavour and it involves the exchange or transmission of information from a sender to receiver. Consequently, ensuring effectiveness in communication requires putting in place measures for securing the messages that are essential components of communication. Some forms of communication are classified as being sensitive and has to be kept from third unintended parties. As far back as over 2000 years ago, efforts have been geared towards securing and ensuring sensitive data. These efforts have produced little or no relevance to modern cryptography. A formal definition of what the goals of secure communication was not obtainable until the 1940s. The first modern definition was pushed forward in 1980 while the second was presented in 1990. One-time pad encryption is the only cryptographic technique developed in 1900 that has evolved with time and as remained relevant in today's encryption system (Rosulek, 2017).

Advanced Encryption Standard algorithm (AES) is a symmetrical cipher algorithm that is the encryption and decryption process of the algorithm involves the use of single (same) key. Its implementation involves the transformation of plain text into ciphertext of blocks of 128 bits and interchanging rounds of swapping and permutation blocks are carried out on the text. The keys used to achieve are in 128, 192 or 256 bits. The key size defines how strong the encryption will be (Anusheh, 2021). AES was invented by the U.S. National Institute of Standards and Technology (NIST) in 2001 as a replacement for the data encryption standard (DES) that was in use till then but implemented with small size key of 56 bits which made it slow and less secure. Owing to its speed and security, it is widely used for electronic data encryption.

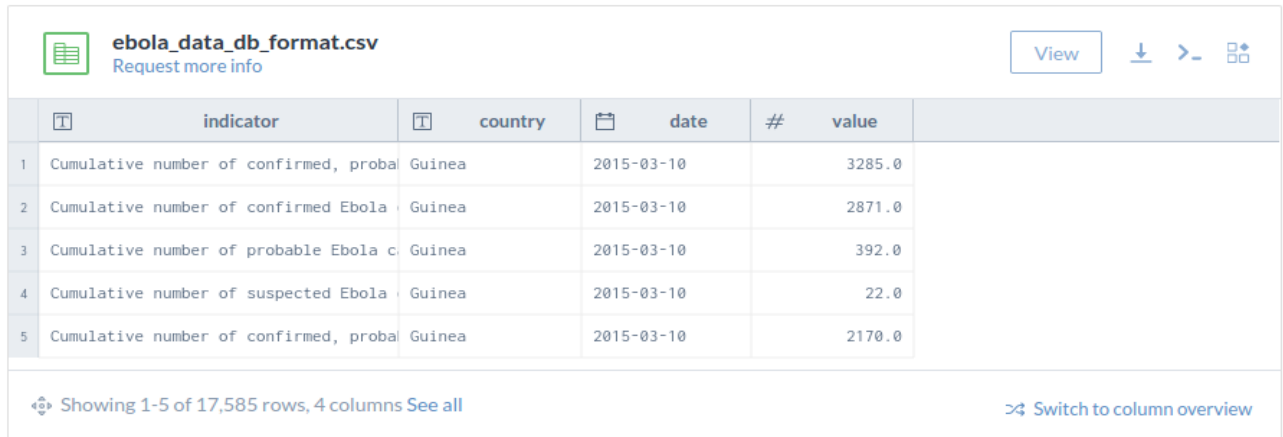
RESEARCH METHODOLOGY

Literatures were sourced to elicit knowledge associated with cryptosystem in healthcare datasets and data bridge. The study resolved to the use of a dummy dataset to test the designed model. The dummy datasets were from surveys on Ebola outbreaks in some selected Countries in Africa, Guinea, Liberia, Nigeria, Sierra Leone, Mali, and Senegal. Also, these data are official materials released by the ministries of health of these countries. The figures given will continue to experience changes due to constant reviewing and analysis and accessibility to outcomes from the laboratory (United Nation, 2014).

The datasets contained the total of 17,585 row (Cumulative number) and 4 columns (Indicator, Country, Date and Value) as presented in Table 1. As presented in Table 1, the dataset comes downloaded as structured digitized Ebola Data and Statistics record management with four variables such as indicator, country, date and value. Patient No. (Patient Number), Hospital No. (Hospital Number), Unit, Sex, Age and Country. For storage over Hadoop technology, it was converted to unstructured dataset.

With this, literatures were sourced to elicit knowledge associated with cryptography in healthcare services. After the literature review and subsequent knowledge assessment were completed, data were analysed in view of one-time cipher encryption for data processing.

Fig 1 Sample of dummy data collected



	indicator	country	date	#	value
1	Cumulative number of confirmed, proba	Guinea	2015-03-10		3285.0
2	Cumulative number of confirmed Ebola	Guinea	2015-03-10		2871.0
3	Cumulative number of probable Ebola c	Guinea	2015-03-10		392.0
4	Cumulative number of suspected Ebola	Guinea	2015-03-10		22.0
5	Cumulative number of confirmed, proba	Guinea	2015-03-10		2170.0

Showing 1-5 of 17,585 rows, 4 columns See all [Switch to column overview](#)

Model formulation

Besides the analysis carried out, algorithm formulated was model using mathematical theory of One-Time Pad (OTP) encryption. Considering this, a cipher is made up of two algorithms in other word, encryption and decryption algorithms.

Moreover, the only satisfying requirement is that this algorithm is reliable, i.e. satisfy the correctness property. The algorithm is often the randomizing algorithm. i.e., when encrypting messages, algorithm E is going to generate random bit for himself and it's going to use those random bit to actual encrypt those message that are given to it while on the other hand, algorithm D is always deterministic i.e. given the key (k) the cipher text output is always the same and does not depend on any randomness that is use by the algorithm.

Considering this study for the purpose of encryption of data supplied over Hadoop technology OTP was adopted as design by (Vernam, 2021) as the most secured cipher among all other cipher. Now why is One-Time Pad (OTP) secure? From information theory of security by (Shannon, 1949), the basic idea was that ciphertext (CTxt) should reveal no info about Plain Text (PTxt) as discussed in the evaluation of this study.

Model Design of One-Time Pad Encryption

One-Time Pad encryption over Hadoop technology was designed using Universal Modeling Language (UML) as shown below. Here the physician enters the dataset as required and encrypts it – automated using python programing language as illustrated bellow. Here, these functions are the main event carried out to encrypt and decrypt medical records needed for storage into HDFS for further action before the system asks if dataset has been encrypted otherwise re-encrypts it.

key

```

lblkey = Label(text = "Key:")
lblkey.place(x = offsetl + 0,y = offsett + 0)
fieldkey = Text(height = 4, width = 40)
fieldkey.place(x = offsetl + 0,y = offsett + 25)

# input
lblinp = Label(text = "Input:")
lblinp.place(x = offsetl + 0,y = offsett + 100)
fieldinp = Text(height = 10, width = 40)
fieldinp.place(x = offsetl + 0,y = offsett + 125)

# encrypt decrypt buttons
lblsel = Label(text = "Select a mode:")
lblsel.place(x = offsetl + 0,y = offsett + 300)
var1 = IntVar()
radioncr = Radiobutton(text = "Encrypt", variable = var1, value = 1,)
radioncr.place(x = offsetl + 0,y = offsett + 325)
radiodcr = Radiobutton(text = "Decode", variable = var1, value = 2,)
radiodcr.place(x = offsetl + 0,y = offsett + 350)
radioncr.select()

# output
lblout = Label(text = "Output:")
lblout.place(x = offsetl + 0,y = offsett + 400)
fieldout = Text(width = 40, height = 10)
fieldout.place(x = offsetl + 0,y = offsett + 425)

```

But if yes, the dataset enters the storage medium, in other words Hadoop Distributed Files System (HDFS). HDFS is a java based distributed file system that allows you to store large data across multiple nodes in a Hadoop cluster. So, for the purpose of this study I considered data of about $\log(X*Y) \geq 12$ and not limited to the physical boundaries of each individual machine.

One of the advantages of distributed and parallel computation is that while taking 43 minutes to process 1 TB file on a single machine. The question how much time will take to process the same 1 TB file when you have 10 machines in a Hadoop cluster with similar configuration – 43 minutes or 4.3 minutes? 4.3 minutes, right! What happened here? Each of the nodes is working with a part of the 1 TB file in parallel. Therefore, the work, which was taking 43 minutes before, gets finished in just 4.3 minutes now as the work got divided over ten machines and scalable. It comprises of name node, data node and secondary node as discussed in the literature review. The dataset received are then processed by providing computational resources for application executions. After the executions, the input dataset is split and mapped and reduced at user's request i.e., for a client's requests for a MapReduce procedure to be executed, there is the need to first locate the input file then read. The input file contains the raw data to

be processed. There are no specifications for input file format notwithstanding the raw data will have to be pre-processed into an acceptable format that the map can readily process.

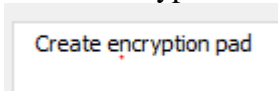
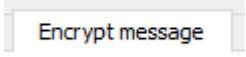
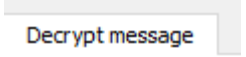
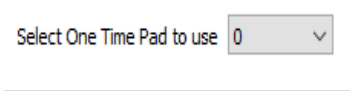
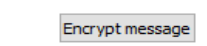
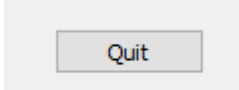
The map and reduce components of the Hadoop framework need to team up to process the data, output from independent mappers is transmitted to the reducers. This functionality within the framework helps to break the big data into smaller pieces for enhanced management. The data thereafter undergoes parallel processing within the distributed cluster. The output of this process is available for users or as input for further processing. These operations are performed in a very robust and reliable organization (Sivaraman and Manickachezian, 2014). And finally, the result received from the output is decrypted using OTP key as presented in the python code above.

System Implementation

The implementation involved the following:

Encryption:

Table 1: The GUI Features of the encryption stage

NO	ICONS	FUNCTIONS
1.		This menu create the one time pad into the storage folder of the system.
2.		Presents a text box for loading the plain text for decrypting.
3.		This tab presents the text box for loading the cipher text for decrypting
4.		This list the one time pads by numbers as stored in the storage folder of the system. Here you make a selection of the pad to be used for encryption.
5.		By clicking on the encrypt button, the plain text was converted to cipher text.
6.		The quit button closes the window.

```
def on_encryptclick(self):
    print('Button encrypt clicked')
    PadSel = self.padCombo_1.currentText()
    with open("Storage/Pad%s.txt" % PadSel, 'r') as f:
        keypad = f.read()
```

```
cipher = onetimepad.encrypt(self.textEditIN.toPlainText(), keypad)
print("Cipher text is ", cipher)
self.textEditOUT.setText(cipher)
self.textEditOUT.repaint()
pyperclip.copy(cipher)
```

The code above performed the encryption of the file. The encryption clicked button was declared. The PadSel variable allows the file to read a pad stored in the storage folder of the system. The pad selected was attached to the encrypted file which will be used for decryption purpose. The cipher text was generated by the keypad selected

```
cipher = onetimepad.encrypt(self.textEditIN.toPlainText(), keypad)
```

1) Transferring the file to the Hadoop environment using these lines of commands:

- `hadoop fs -mkdir /ebola_dir`
- `hadoop fs -mkdir /ebola_data_db_format_dir`
- `hadoop fs -put C:/encrypted_file.txt /ebola_data_db_format_dir`
- `hadoop fs -ls /ebola_data_db_format_dir/*`

The command lines itemized above created a directory on the Hadoop platform called `ebola_dir` and `ebola_data_db_format.dir`. The encrypted file “`encrypted_file.txt`” was loaded into the Hadoop platform using the line command;

```
hadoop fs -put C:/encrypted_file.txt /ebola_data_db_format_dir
```

encrypted file was saved in the directory folder created.

2) Transferring the key to decrypt through desired private communication channel

The key was transferred using private key symmetric method. The same key used for encryption was the same key used for decryption.

Decryption:

1) Accessing the same Hadoop environment
 Figure 2. Accessing Hadoop Environment.

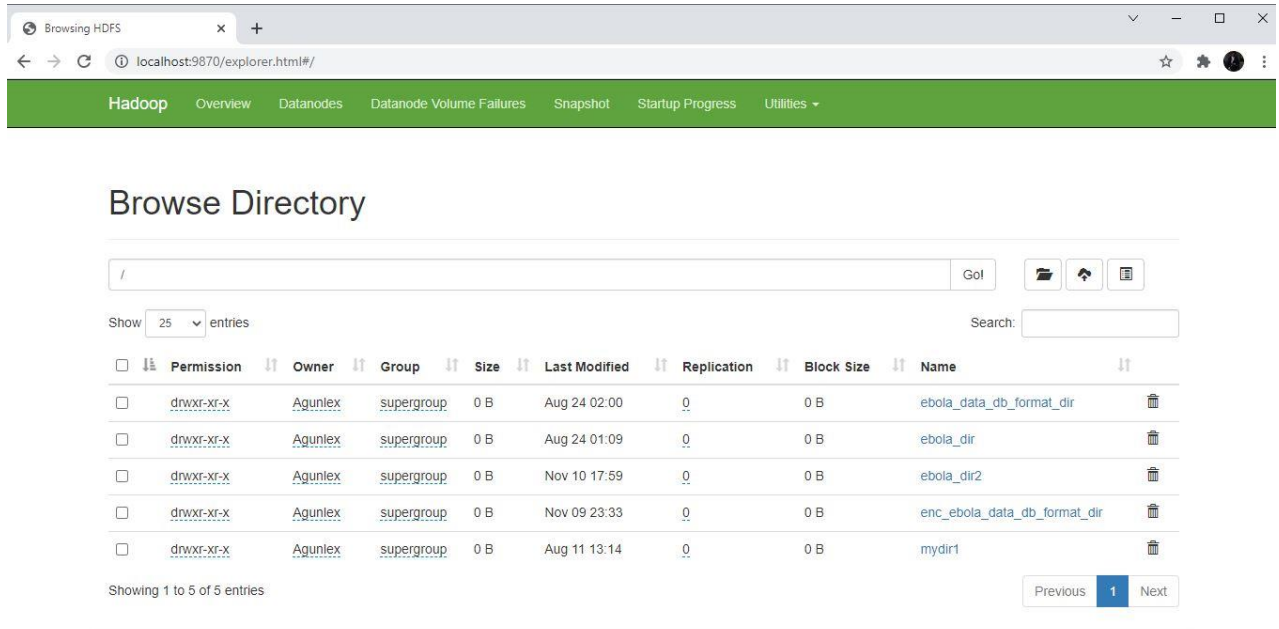


Figure 2 explained how the Hadoop Resources was started from the command prompt. From the browser, <http://localhost:9870/> was typed and this brought out the Hadoop environment. By clicking on the utilities and selecting the directory of the files, it displayed the folders created on the Hadoop platform as shown in figure 4d.

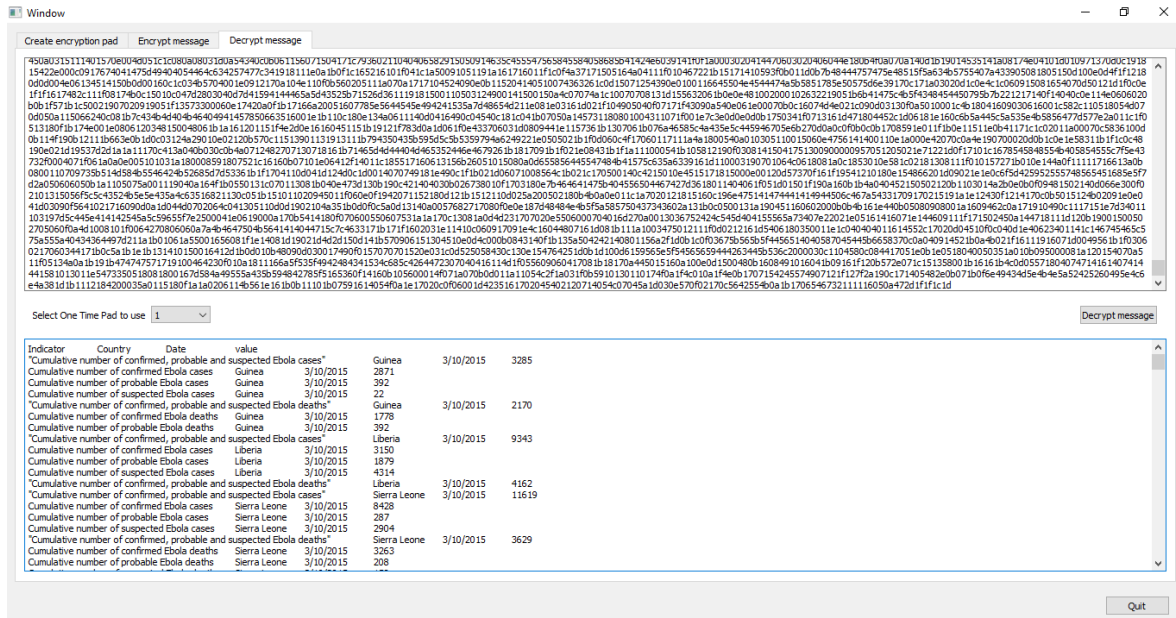
2) Downloading the encrypted file (ciphertext) from Hadoop
 By clicking on the directory containing the encrypted file on the Hadoop platform. The encrypted file was downloaded and ready for decrypting.



Figure 3. Downloading the Encrypted file

3) Loading the ciphertext into the OTP GUI environment

Figure 4. Loading Ciphertext into OTP



```
def on_decryptclick(self):
    print('Button decrypt clicked')
    PadSel = self.padCombo_2.currentText()
    with open("Storage/Pad%s.txt" % PadSel, 'r') as f:
        keypad = f.read()
    msg = onetimepad.decrypt(self.textEditIN_2.toPlainText(), keypad)
    print("Plain text is ", msg)
    self.textEditOUT_2.setText(msg)
    self.textEditOUT_2.repaint()
    pyperclip.copy(msg)
```

This code session performs decryption of the text. The onetimepad generated in the storage folder was read. The PadSel allows you to select a pad from the storage folder which was used for decrypting. By clicking the decrypt button, the plain text message was generated.

4) Carry out decryption using the private key.

System Evaluation

The system evaluation was carried out to compare the developed system's performance. The evaluation involved carrying out a performance analysis on the modelled OTP and AES algorithms using different file sizes.

The evaluation parameters that were used to determine the performance of the algorithms were:

- Encryption time
- Decryption time
- Throughput of encryption

- Throughput of decryption

The encryption time is defined as the time taken for an encryption algorithm to generate a cipher text from a plaintext. This time is also used to determine the throughput of an encryption system. It is a measure of the speed of encryption (Elminaam et al. 2009).

In the figures 5, 6 and 7, the blue line represented the OTP while the red line represented the AES algorithms respectively. Figure 5 showed differences in the output of the file sizes after encryption. This is also presented in table 5. Figure 6 represents the output of the time taken by each algorithm to upload and encrypt the file sizes. There was noticeable difference in the time taken. Figure 7 represent the output of the time taken to download and decrypt the files by the algorithms. The computational time for the downloading and decrypting also varied.

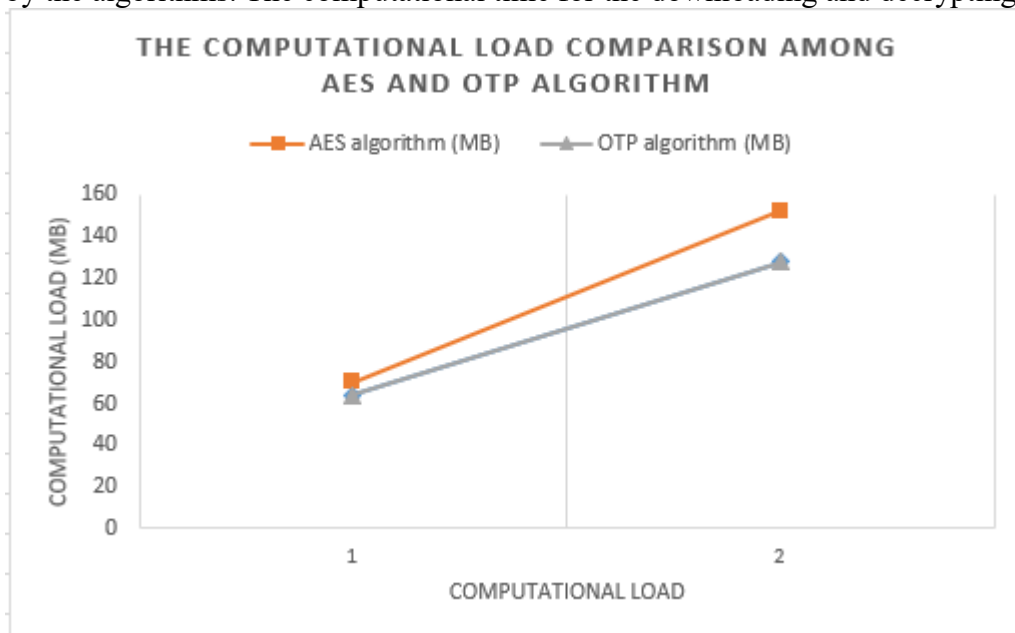


Figure 5: Showing the Computational Outputs of the two algorithms after encryption

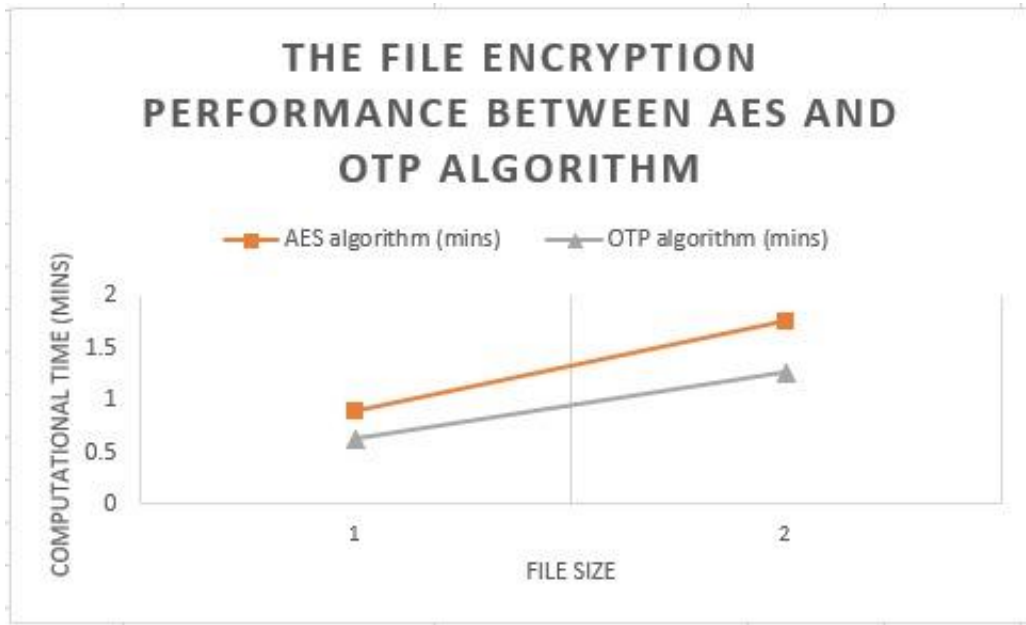


Figure 6: Showing the computational time taken by the two algorithms to encrypt the uploaded plaintext

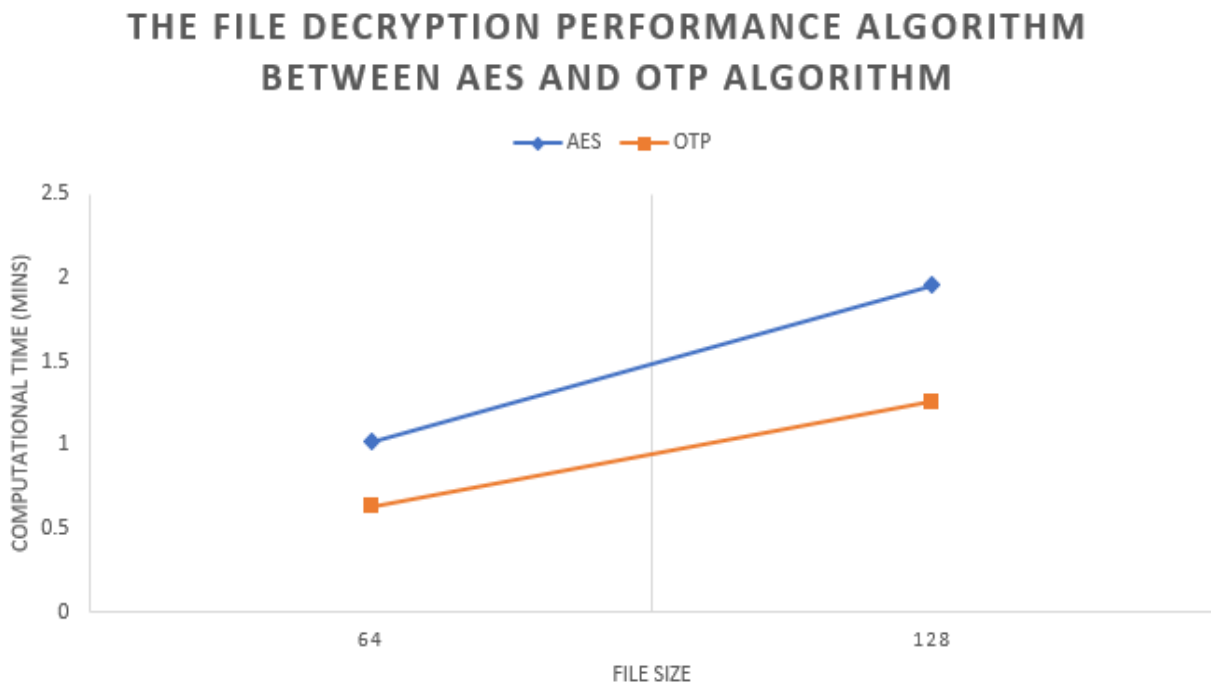
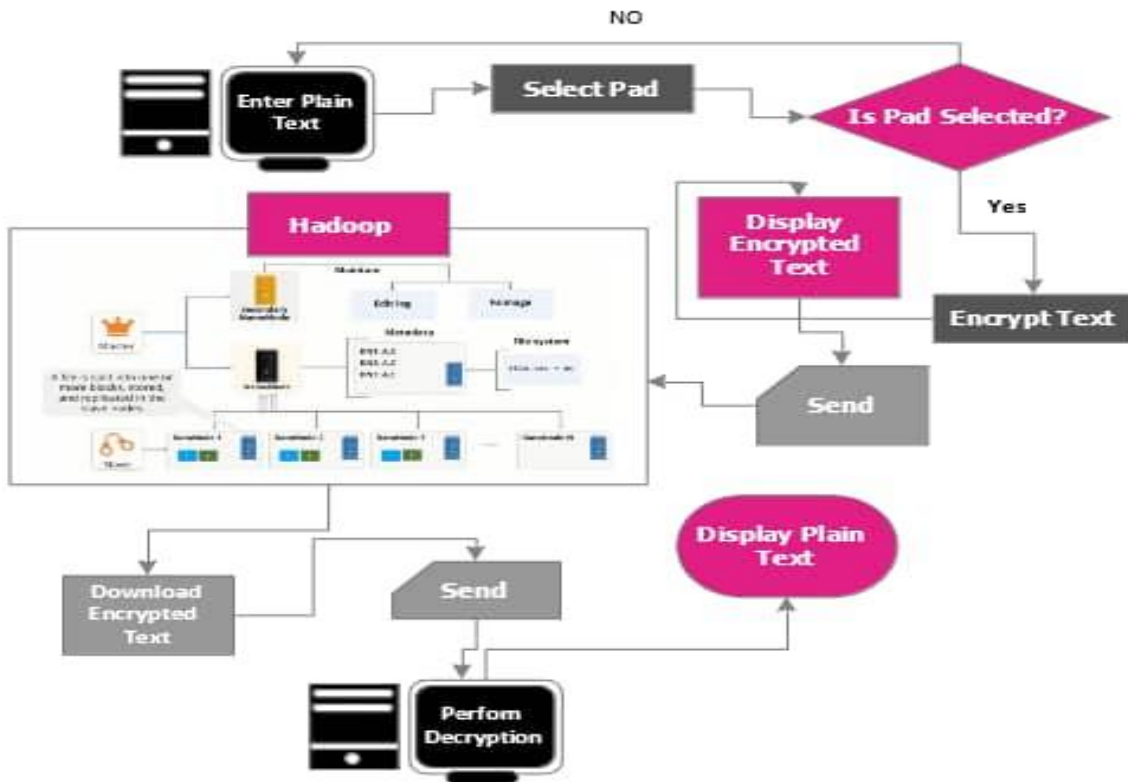


Figure 7: Showing the computational time taken by the two algorithms to decrypt the ciphertext

Figure 8 - Diagrammatic representation of system architecture



RESULT AND DISCUSSION

From the above line chart, it is obvious that the increase in both encryption and decryption time was because of an increase in the data size that was used. Different encrypted Data sizes were used for testing purpose while uploading into the Hadoop framework. Securing the sample healthcare (Ebola) bigdata was achieved by subjecting them to OTP and AES encryption/decryption algorithms.

CONCLUSION

From the above graph, the OTP method of encryption takes less time to do the encryption and decryption. For secure authentication, one-time password was generated via the OTP algorithm and encrypted using the python programming language and then sent to HDFS within the Hadoop framework.

Also, from the graph, I observed that doing this had no effect on the system. From the number of experiments, OTP generation and encryption took less time compared to AES encryption and decryption algorithm as presented in the graphs. From the adopted approach, the

authentication technique introduced before downloading the encrypted message was necessary as a consideration for additional height of security.

In this study, I have presented a computational model for big healthcare data security on the Hadoop framework using the One Time Pad encryption algorithm. From the results obtained, I could deduce that One Time Pad algorithm offers reliable security for healthcare data if properly implemented. The technique can be optimized for real-time data encryption. It is known to be the only unbreakable cipher algorithm till today. From literature reviewed and the evaluation carried out, I found out that OTP offers a more concise solution because of its ease of implementation with fewer codes and complexity when compared with some other encryption algorithms.

In carrying out the study, relevant and recent literatures were reviewed for the purpose of gaining better insight and grasp of the technologies involved in the study i.e., big data analytics, Hadoop, and encryption. Thereafter, the Universal Model Language (UML) was used to model the chosen algorithm. Afterward, the model was implemented using python programming language. Also, appropriate software and hardware requirements for the implementation were sourced and this helped facilitate the deployment and testing of the modelled encryption system.

This study became necessary because of the increasing applicability of big data in the healthcare domain (such as in the areas of personal health management, enhanced clinical care, knowledge discovery, innovative health research) and the noticeable threats and concerns that constitute impediments to maximizing its potentials. Security of big healthcare data is a major issue in this regard. Healthcare data have always been sensitive and are getting more sensitive day by day.

REFERENCES

- Abouelmehdi, K., Beni-Hessane, A. & Khaloufi, H. (2018). Big healthcare data: preserving security and privacy. *J Big Data* 5, 1. <https://doi.org/10.1186/s40537-017-0110-7>
- Anusheh, Z. (2021). *What is AES algorithm*. Available at <https://www.educative.io/edpresso/what-is-the-aes-algorithm> (Accessed: 19 August 2021)
- Belle Ashwin et al., (2015). Big Data Analytics in Healthcare. *BioMed Research International* Vol. 2015:370194. Doi:10.1155/2015/370194.
- Dash, S., Shakyawar, S. & Sharma, M. (2019). Big data in healthcare: management, analysis and future prospects. *J Big Data* 6, 54. Available at: <https://doi.org/10.1186/s40537-019-0217-0>
- Elminaam, D. S., Kader, H. M., & Hadhoud, M. M. (2009). Performance Evaluation of Symmetric Encryption Algorithms. *Communications of the IBIMA*, 8, 58-65.
- Harshit (2020). *What is cryptography?* Available at: <https://www.technoarchsoftwares.com/blog/what-is-cryptography/> [Accessed 1st December 2015].

- Kaicheng, L., (2003). Computer cryptography: Computer network data privacy and security. Tsinghua University Press, 2003.
- Lydia, L., & Harika, P., & Poorna, C., SaiTejaswi, G., & Karthik, P. (2018). Processing and Securing Healthcare Datasets through Hadoop and Implementing Cryptography Technique. *International Journal of Pure and Applied Mathematics* Volume 118 No. 7 2018, 333-339. (Accessed 7th November, 2021)
- Mike Rosulek (2017). The Joy of Cryptography. Available at: <https://open.oregonstate.edu/cryptographyOEfirst/chapter-1-one-time-pad>. (Accessed: 01 December 2021).
- Naeemabadi, Mreza & Ordoubadi, B.s & Dehnayi, Alireza & Bahaadinbeigy, Kambiz. (2015). Comparism of serpent, Twofish and Rijindael encryption algorithms in tele-ophthalmology system. *Advances in Natural and Applied Sciences*. 9. 137-149
- Olufohunsi, T. (2019). Data Encryption. Available at: https://www.researchgate.net/publication/337889039_data_encryption_olufohunsi_T.
- Shannon, C. E. (1949) "Communication theory of secrecy systems," *The Bell System technical journal*, 28(4), pp. 656–715.
- Sivaraman, E. and Manickachezian, R. (2014) "High performance and fault tolerant distributed file system for big data storage and processing using Hadoop," in 2014 International Conference on Intelligent Computing Applications. IEEE, pp. 32–36.
- Toonders, J (2014) Data is the new oil of the digital economy. *Wired*. Available at: <https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/>
- Vernam Cipher in Cryptography (2021). Available at: <https://www.geeksforgeeks.org/vernamp-cipher-in-cryptography> (Accessed: 3 August 2021)
- Yusuf, P (2019) The Hadoop Security in Big Data: A Technological Viewpoint and Analysis. *International Journal of Scientific Research in Computer Science and Engineering*. Available at: <https://hal.archives-ouvertes.fr/hal-03226895>