Implementation of a File Encryption Software "Hyde" using RIJNDAEL Algorithm (AES)

Article *in* International Journal of Computer Science and Information Security, · April 2022

DOI: 10.5281/zenodo.7129308

3 authors, including:

Itunuoluwa Isewon
Covenant University Ota Ogun State, Nigeria
63 PUBLICATIONS **912** CITATIONS

SEE PROFILE

Jelili Olanrewaju Oyelade
Covenant University Ota Ogun State, Nigeria
71 PUBLICATIONS **1,260** CITATIONS

SEE PROFILE

https://doi.org/10.5281/zenodo.7129308

Implementation of a File Encryption Software "Hyde" using RIJNDAEL Algorithm (AES)

Itunuoluwa Isewon
Department of Computer and Information Sciences
Covenant University
Ota, Nigeria.
itunu.isewon@covenantuniversity.edu.ng

Oluwasola Adare
Department of Computer and Information Sciences
Covenant University
Ota, Nigeria.

Jelili Oyelade
Department of Computer and Information Sciences
Covenant University
Ota, Nigeria.
ola.oyelade@covenantuniversity.edu.ng

ABSTRACT
A File Encryption software is an application tool that helps to protect sensitive data from unauthorized users by encrypting or encoding the data or message into a form generally referred to as a cipher that has no relevance to the unauthorized use of that data or information. This study aims to proffer solutions to the problems of data security, whereby unauthorized users or attackers as the case may be, gain access to their data and information during transmission via various platforms. Also, users would like to embed messages within files such as image stenography and text cryptography. The study was implemented using UML tools for modeling the designs, an object-oriented programming language (Java) for the development, and Adobe Photoshop was used for the UI/UX in order to enhance the user interaction with the system.

General Terms
RIJNDAEL Algorithm, File Encryption, Data Security.

Keywords
RIJNDAEL Algorithm, File Encryption, Data Security.

1. INTRODUCTION
Encryption can simply be described as the protection of data from unauthorized users. These unauthorized users are the ones who do not possess the password or access privileges to view a file or document. For example, a user who wants to make a purchase on an E-commerce website and on the payment platform of this website he would like to make use of his credit card to make the payment for this purchase. Your computer device encrypts the data so that unauthorized personnel do not gain access to your personal data during the transfer. Another case scenario is this; you have a file or document on your computer device you would want to be only accessible by you the creator of that file or document [1].

Encryption can also be described as the transformation of electronic information into a different frame, named a cipher text, this cipher text cannot be comprehended by anybody effortlessly asides from the parties authorized to do so.

Encryption has the primary purpose of providing the privacy of advanced information that are stored on various PC frameworks or perhaps transmitted through the use of the Internet or through any more PC networks [1]. It is assumed by modern encryption algorithms that a fundamental part in the security confirmation of Information Technology frameworks and also communications as they provide confidentiality, and also the listed below key components of security:

- Authentication: this implies that the origin from which the message was sent can be confirmed.
- Integrity: this provides proof that substance of the initial message has not been altered since the send request was initiated.
- Non-repudiation: this means that the sender of a message cannot refute initiating a send request for a message.

Cryptography can be described as the method for hiding and transmitting data in a particular frame so that the intended recipients can read and process the data [1].

Cryptography is a field of study, that is essentially identified with the disciplines of both Cryptology and Cryptanalysis. Cryptography constitutes methods, for illustration purposes let us sample these scenarios; blending words with pictures, microdots, and various other approaches to conceal data. Be that as it may, in the present-day PC-driven world, Cryptography as a concept is often connected with the scrambling of plain text into cipher text – this is a procedure referred to as encryption - then back once more. People who carry out this practice are often referred to as Cryptographers [2].

A computer file is an asset for storing data, which is accessible to a PC program and is normally in view or some likeness thereof of durable storage. A file is "durable", that is, it stays accessible for other different projects to use after the program that created it has completed the process of executing. Computer files can be considered as the data innovation partner of paper documents which generally are kept in office and library files, and this is the wellspring of the term [3].

A file format can be described as the structure of a file regarding the way the information inside the record is sorted out. A program that utilizes the information within a file must have the capacity to perceive and perhaps gain access information inside the record. For instance, the program that we regard as a Web browser can process and output a file in the HTML record, organized with the goal that it displays as a Web page, however it can't show a record with a configuration that is intended for Microsoft's Excel program. A particular file format is regularly shown as a component of the name of the file with the use of a file name expansion. Routinely, the expansion is isolated by a period (a dot) from the file name and contains 3 or 4 letters that recognize the configuration. A program that utilizes or perceives a particular file format could conceivably mind whether the record possesses the appropriate

expansion name provided it can really inspect the bits contained in the file to determine whether the format (arrangement) is one it perceives [3].

There are the same number of various document designs as there are distinctive projects to handle the records. A couple of the more normal document organizations are:

- Word archives (.doc)
- Content of web pages (.htm or .html)
- Pictures obtained from web pages (.gif and .jpg)
- Adobe Postscript records (.ps)
- Executable projects (.exe)
- Adobe Acrobat records (.pdf)
- Multimedia records (.mp3 and others)

Encryption algorithms are ordinarily utilized as a part of computer communications, including FTP transfers. Typically, they are utilized to give secure exchanges. On the off chance that an algorithm is utilized as a part of an exchange, the file is initially interpreted into an apparently insignificant cipher text and after that moved in this design; the recipient PC utilizes a key to make an interpretation of the figure into its unique frame. So, if the message or file is captured before it achieves the accepting PC it is in an unusable (or scrambled) frame [2].

Here are some commonly used algorithms:

- DES or Triple-DES algorithms
- Blowfish algorithm
- AES algorithm
- Two-fish algorithm
- IDEA algorithm
- MD5 algorithm
- SHA 1 algorithm
- HMAC algorithm
- RSA algorithm

In this work, we implemented a cryptosystem for the encryption/decryption of plaintext as well as files of different file formats. This was done by combining both symmetric and asymmetric key cryptography as well as salting and padding to the data being encrypted/decrypted for adequate confidentiality.

This paper is organized as follows: in the next Section, we give a review of the Rijndael Algorithm and how it works for Encryption and Decryption. In section 3, we discussed briefly the results and discussion and we conclude the paper in Section 4.

2. RIJNDAEL ALGORITHM

2.1 Advanced Encryption Standard

On November 26, 2001, the Federal Information Processing Standards Publication 197 announced a standardized form of the Rijndael algorithm as the new standard for encryption. This standard was called Advanced Encryption Standard (AES) and is currently still the standard for encryption [4].

AES algorithm is a symmetric encryption algorithm, meaning that both parties should have the information regarding the components used in the process [5]–[7]. Figure 1 shows a flowchart diagram that describes the flow using the Rijndael Algorithm.

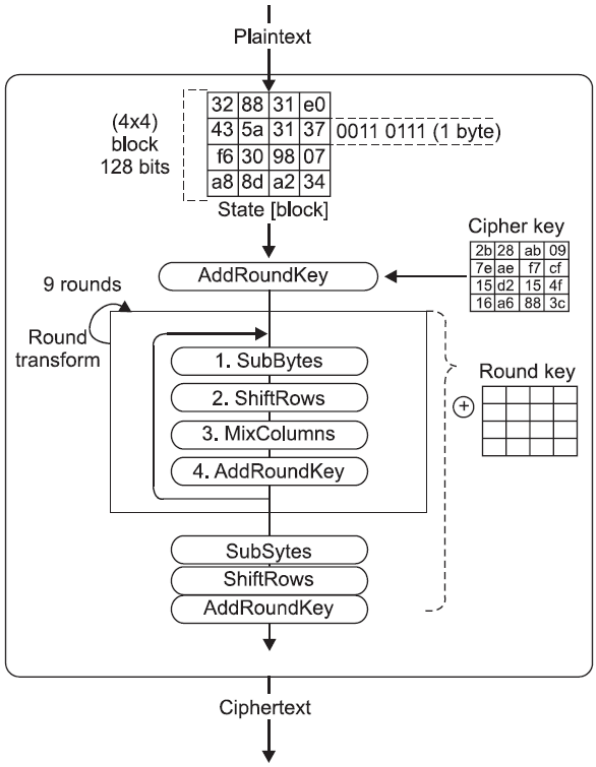

Fig 1: The flowchart of Rijndael Algorithm (AES) [8]

A study conducted by [9] which integrated AES into Short Message Service (SMS) application shows that the process of encryption and decryption did not hinder the communication between two parties when the encryption technique is used [10].

Oh et al [8] developed a Selective Encryption Algorithm (SEA) based on AES technique for Medical Information. SEA modified the AES because AES are considered not suitable for visual data such as digital image because of long computation process. SEA reduced the number of rounds and implementation time using Huffman coding [11]. This illustrates that the AES algorithm can be improved upon as demanded by system requirements.

2.2 Major components of the Rijndael Algorithm

BLOCK AND KEY: The block size and key size must be determined before the algorithm can be applied to the data. AES allows for block sizes of 128, 168, 192, 224, and 256 bits. The key sizes allowed by the AES are: 128, 192, and 256 bits [12], [13]. The standard encryption uses AES-128 where both the block and key size are 128 bits. The block size is usually denoted as Nb while the key size is usually denoted as Nk. Nb refers to the number of columns in the block where each row in the column consists of four cells of 8 bytes each for AES-128 [13], [14]. In Figure 2 below, you find an example of the AES-128 block cipher.

Block

	0	1	2	3
0	T		a	s
1	h	i		t
2	i	s	t	.
3	s		e	.

Fig 2: AES-128 Block Example

ROUNDS: The Rijndael algorithm, at the basic level, utilizes a number of rounds which would be used to transform the data for each block. The number of rounds used is 6 + the maximum of Nb and Nk. The initial block (state) is added to an expanded key derived from the initial cipher key. Then the round processing occurs consisting of operations of the S-box, shifts, and a MixColumn. The result state is then added to the next expanded key. This is done for all ten rounds, with the exception of the MixColumn operation of the final round. The final result is the encrypted cipher block [14].

KEY EXPANSION: The original cipher key needs to be expanded from 16 bytes to 16 * (r + 1) bytes. A round key is required immediately after each round and also required before the first round. Each round key is required to be 16 bytes and this is so because the block size is 16 bytes.

Hence, the cipher key needs to be expanded from 16 bytes to 16*(r + 1) bytes or 176 bytes. The expanded key is then broken up into round keys. Round keys are added to the current state after each round and before the first round [4].

S-BOX: The first step to be performed to a round is to carry-out a byte by byte substitution using a lookup table referred to as an S-box. An S-box can be simply described as a one to one mapping for all byte values from 0 to 255. The S-box is used to change the original plain text which is stored in bytes to cipher text. The S-box is shown below in Figure 3. All values contained in the S-box are represented in hexadecimal notation.

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	e5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	e7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fe	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	60	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0e	13	ee	5f	97	44	17	e4	a7	7e	3d	64	5d	19	73
9	60	81	4f	6e	22	2a	90	88	46	ee	b8	14	6e	5e	05	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6e	56	f4	ea	65	7a	be	08
c	ba	78	25	2e	1e	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	cf	b0	54	bb	16

x

Fig 3: The S-Box which is a matrix representing one-to-one mapping of all byte values from 0 to 255 [8]

The diagram below shows the structure of the block after the substitution operation has been performed. This is as seen in Figure 4.

	0	1	2	3
0	0x20	0xB7	0xEF	0x8F
1	0xF9	0xB7	0x92	0x45
2	0x92	0x31	0xF9	0x8F
3	0x31	0x8F	0xB7	0x4D

Fig 4: showing block after substitution

SHIFTS: The next step in the round is to shift the rows of the state. The rows are shifted x number of bytes to the left where x is the row number. This means row 0 will not be shifted, row 1 will be shifted 1 byte to the left, row 2 will be shifted 2 bytes to the left, and row 3 will be shifted 3 bytes to the left. The resulting order is as seen below in Figure 2.5.

	0	1	2	3
0	0x20	0xB7	0xEF	0x8F
1	0xF9	0xB7	0x92	0x45
2	0x92	0x31	0xF9	0x8F
3	0x31	0x8F	0xB7	0x4D

Fig 5: showing block after shifting

MIXCOLUMN: After applying the S-box and shifts to the state the operation of a MixColumn is used. The MixColumn lookup table takes a byte and transforms it into four bytes. The MixColumn table is generated by the following algorithm. Each element in the table consists of four bytes usually represented as hexadecimal values. The second and third bytes are always same as the input byte.

DECRYPTION: Decryption is simple after understanding the encryption process. It is basically just the inverse. The algorithm was designed for all the steps to be invertible so decryption is basically like doing everything backwards. Therefore, for decryption starts at the last round and the last round key. When processing, each round does the process backwards. So, the round key is added first to the last round. Addition is its own inverse, which is nice. Then the MixColumn step is applied. The MixColumn step is applied to all rounds except the last one. Also, the inverse MixColumn table is used [14]. This table is generated with another matrix similar to the way the MixColumn table was generated. The difference is that there are no short cuts to generate the table. Therefore, the matrix multiplication needs to be performed in the field GF.

3. RESULTS AND DISCUSSION

Here, we evaluate the system by making a practical use of the application by encrypting a file and decrypting the same file.

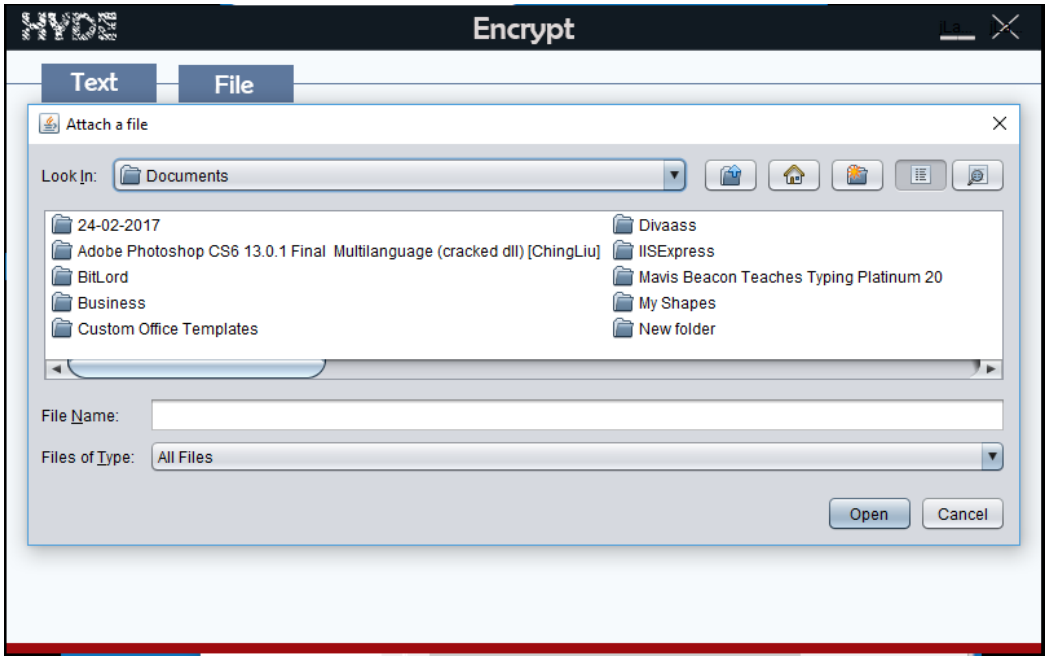

Figure 6: Selected File to be Encrypted. The application provides a dialog box that enables users to select the file to be encrypted. The encryption is not affected by the size or type of file.

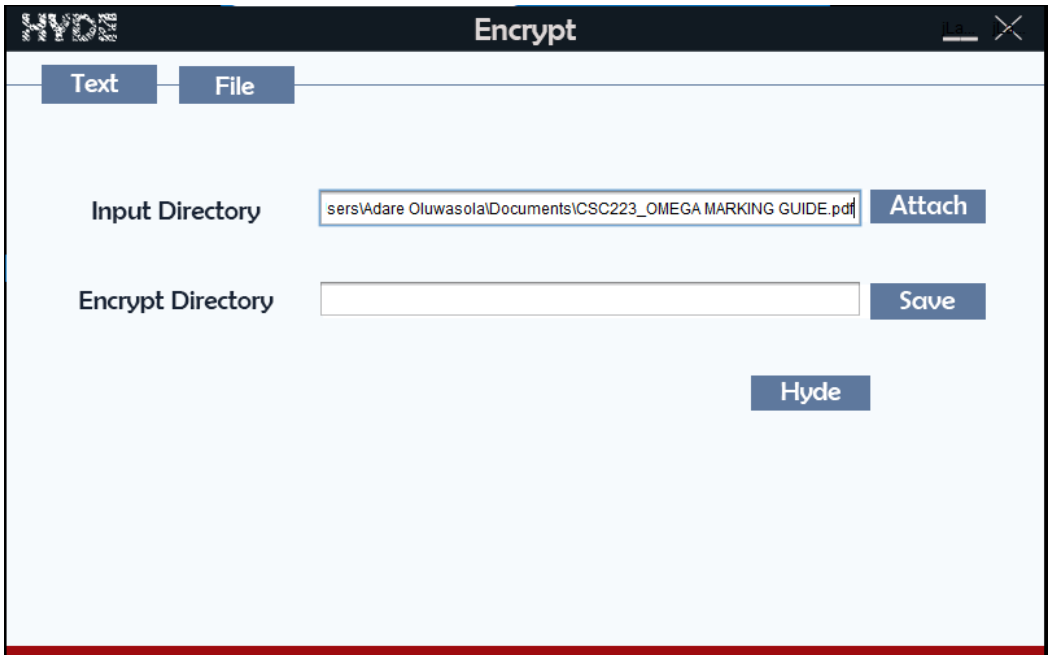

Figure 7: showing the path of the selected file to be encrypted. The interface enables user to specify the file to be encrypted and storage location. The current location of the input file may as well be same for the encrypted (output) file, however, the encrypt directory field must be completed.

3.1 File Encryption Process

Select File to Be Encrypted: This step involves the user using the input dialog box to select the file to be encrypted. This can be found below in Figure 6 and Figure 7 respectfully.

Select File Name: The user input the name with which the encrypted file would be saved as. This can be seen below in Figure 8.

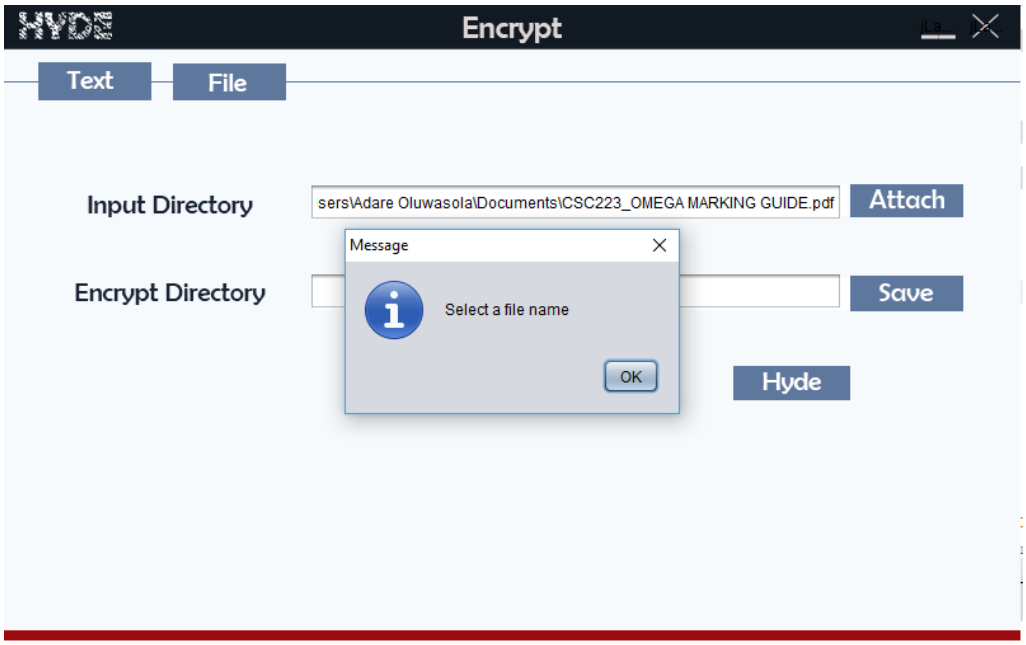

Fig 8: showing the select name prompt. The interface also validates the user input to ensure they are properly filled.

File Encrypted: The user attaches the file to be encrypted and specifies the name of the output file. The software notifies the user upon successful encryption of the file. This can be found below in Figure 9.

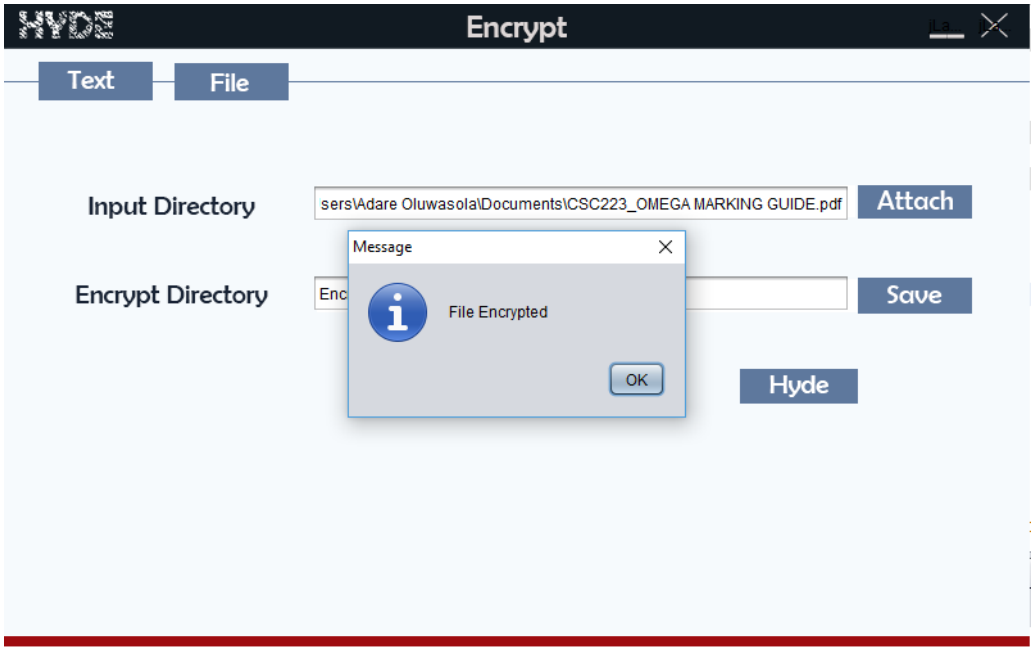

Figure 9: showing the file encrypted prompt. A notification indicates the successful encryption of the input file.

File in Default Folder: Figure 10 showing the user viewing the encrypted file in the default folder.

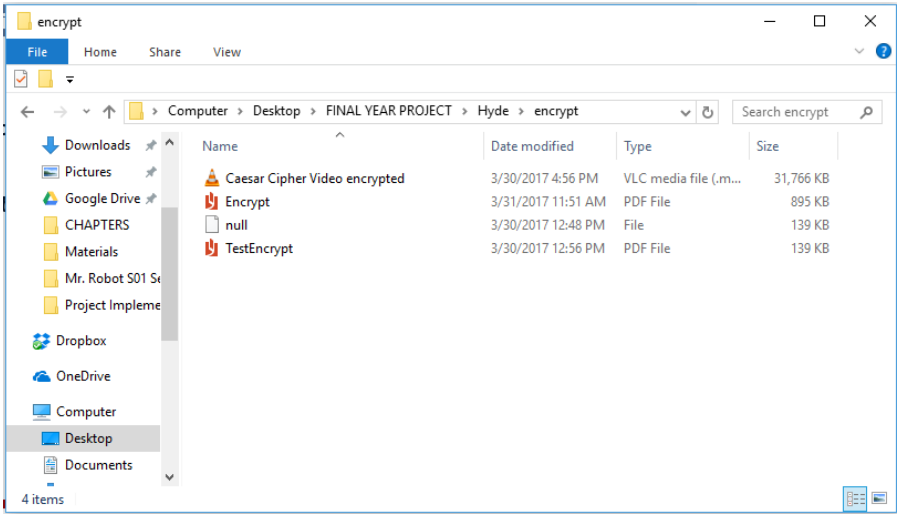

Figure 10: The encrypted file in the default folder. The encrypted file can be accessed from the default or specified output folder.

Encrypted File Being Accessed: Figure 11 shows the user trying to access the encrypted file.

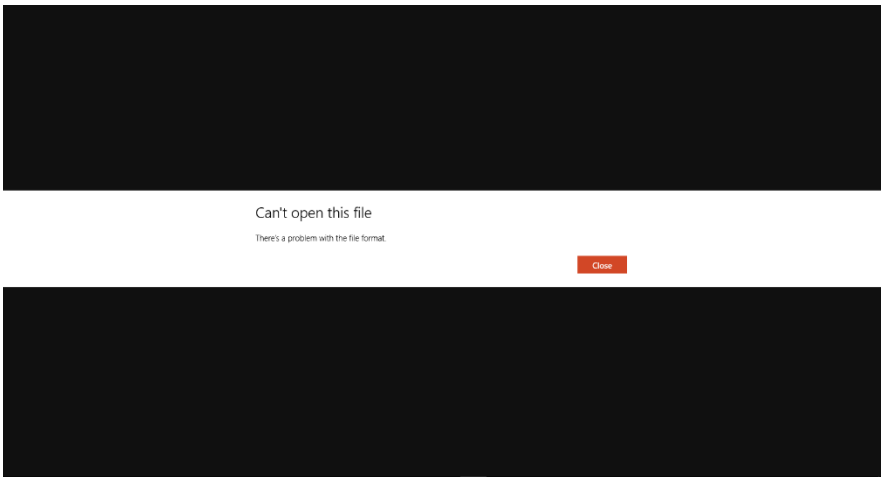

Figure 11: showing the encrypted file being accessed. The content of the encrypted file cannot be accessed until it is decrypted.

3.2 File Decryption Process

Select File to Be Decrypted

This step involves the user using the input dialog box to select the file to be decrypted. This can be found below in Figure 12.

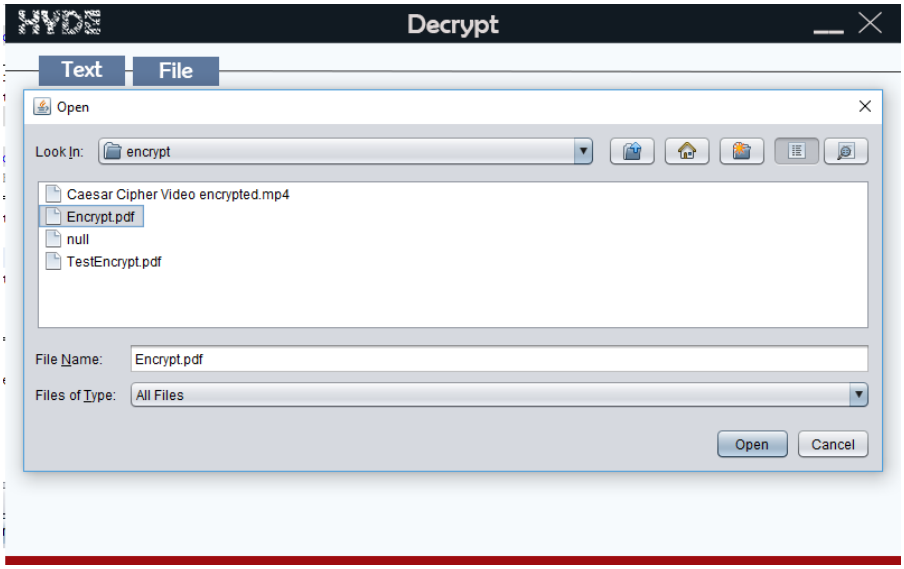

Figure 12: Selected File to be Decrypted. The attach button on the decrypted form in Figure 13 opens this interface to allow the user to select the file to be decrypted.

Select File Name: The user input the name with which the decrypted file would be saved as. This can be seen below in Figure 13.

Figure 13: Interface to select input file and specify name of the file. The interface enables user to specify the name of the file to be decrypted. The interface validates user input to ensure they are filled.

File Decrypted: The user attaches the file to be decrypted and specifies the name of the output file. The software notifies the user upon successful decryption of the file. This can be found below in Figure 14.

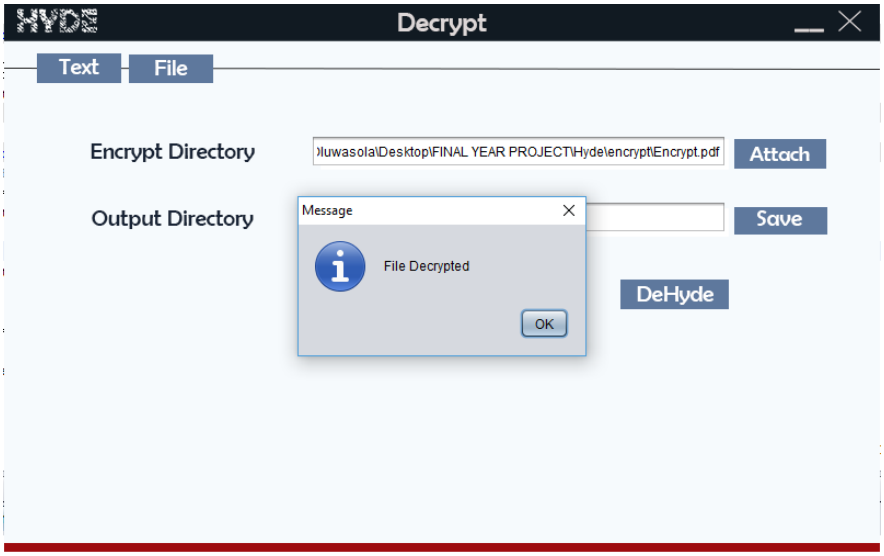

Figure 14: Successful File Decrypted Notification. A notification indicates the successful decryption of the encrypted (input) file. The user is subsequently notified to check the decrypted file in the output folder.

File in Default Folder: Figure 15 shows the user viewing the decrypted file in the default folder

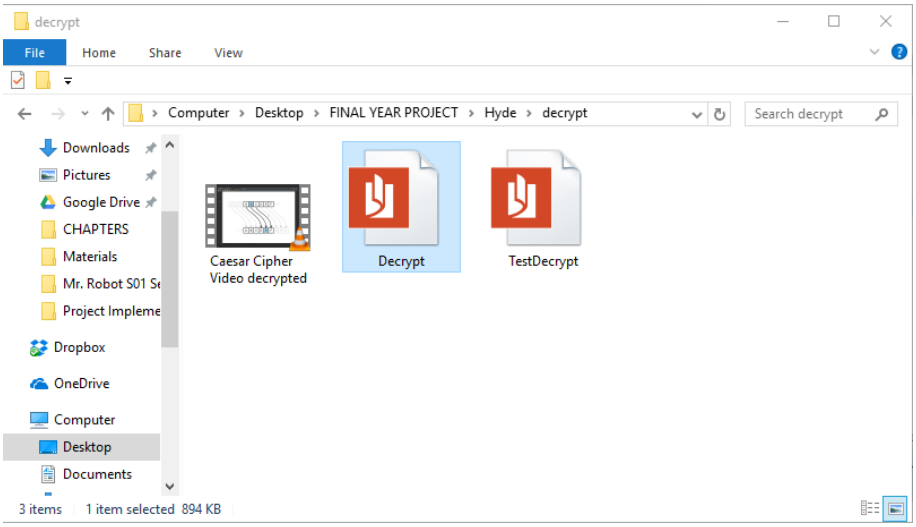

Figure 15: showing the decrypted file in the default folder. The decrypted file can be accessed from the default or specified output folder.

Decrypted File Being Accessed: Figure 16 shows the user accessing the decrypted file

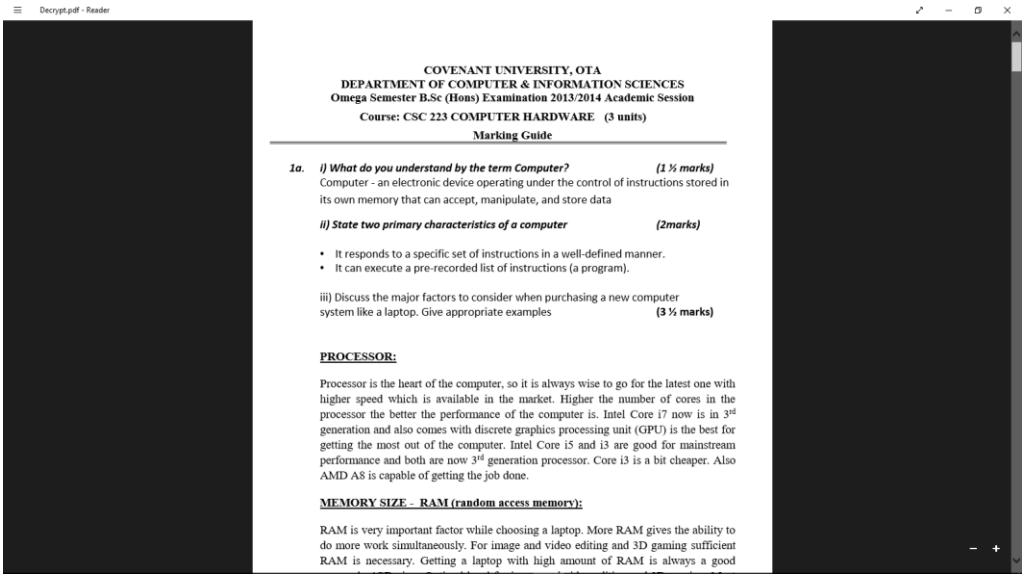

Figure 16: showing the decrypted file being accessed. The file is the output of the decryption process.

3.3 Text Encryption Process

Input Plain Text: Figure 3.15 showing the user inputting a plain text into the corresponding Text Area.

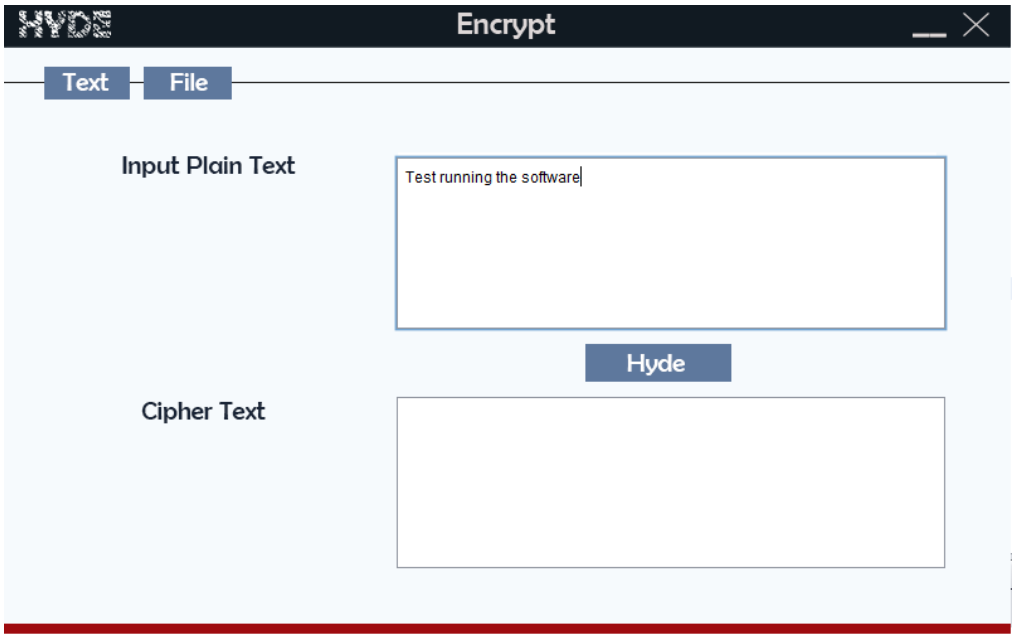

Figure 17: showing the plain text being inputted. Storage of important texts like passwords might need to be encrypted before storing. The form contains two fields, the input plain text field and the cipher text field which is the output field upon clicking the "Hyde" button.

Generate Key Value: Figure 18 shows the key value upon the mouse clicked event initiated by the user.

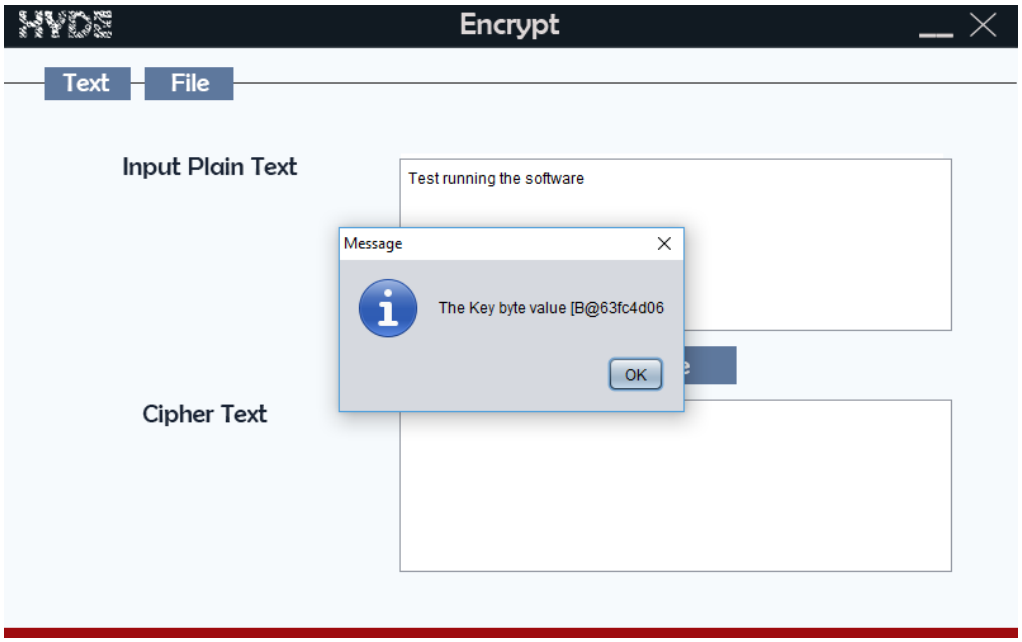

Figure 18: showing the key value generated. Although the encryption key shown is not useful to the user, at the same time do not pose a threat to the system. It also notifies the user of the successful encryption of the plain text.

Cipher Text Generated: Figure 19 showing the cipher text generated by the Hyde.

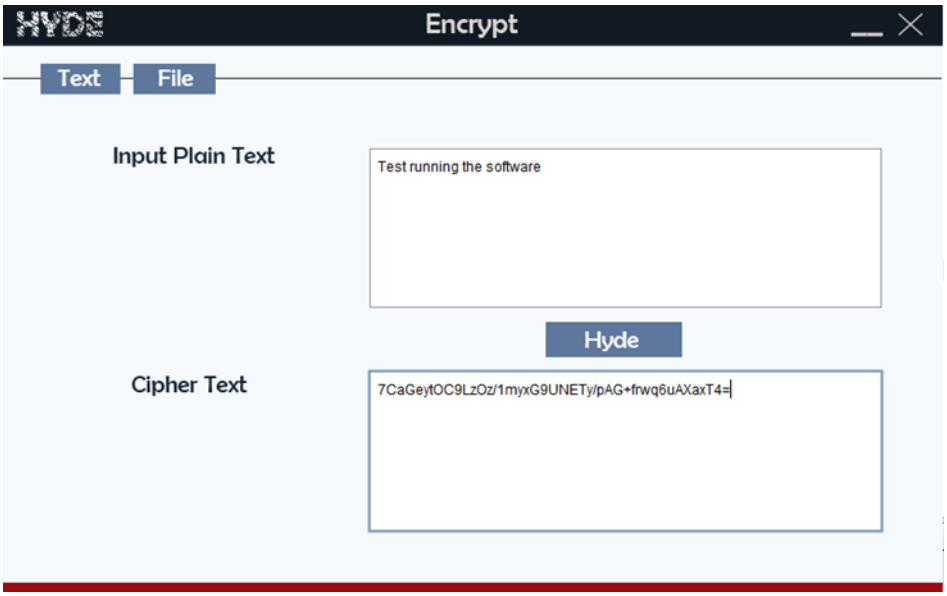

Figure 19: showing the cipher text generated. When the "Hyde" button is clicked, it validates the input field. The cipher text (output) is displayed in the cipher text field. The output can be stored or used for further processing.

3.4 Text Decryption Process

Input Cipher Text: Figure 20 shows the user inputting a cipher text into the corresponding Text Area.

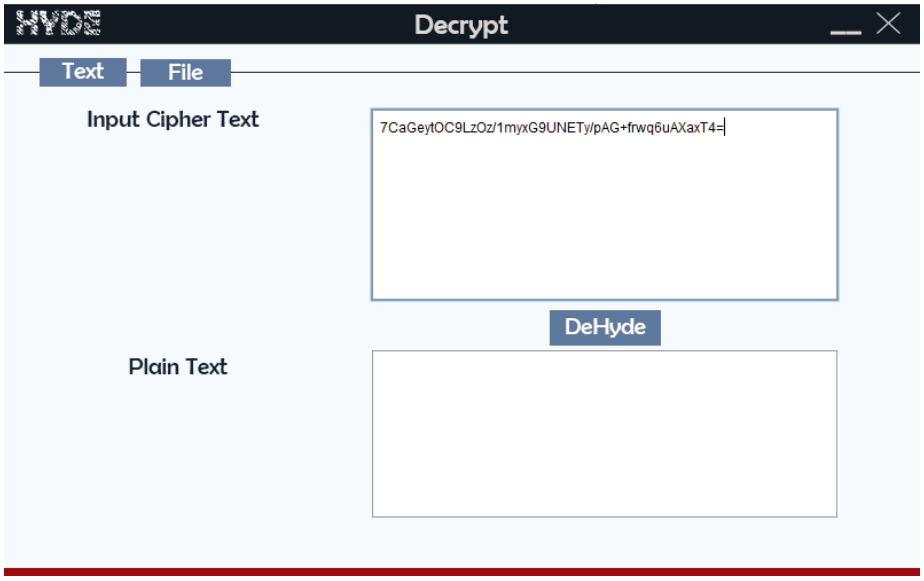

Figure 20: showing the cipher text being inputted. The decrypt form which provides an interface for text decryption has an input field where the cipher text is inputted and the plain text field that displayed the output of the decryption process.

Generate Key Value: Figure 21 showing the key value upon the mouse clicked event initiated by the user.

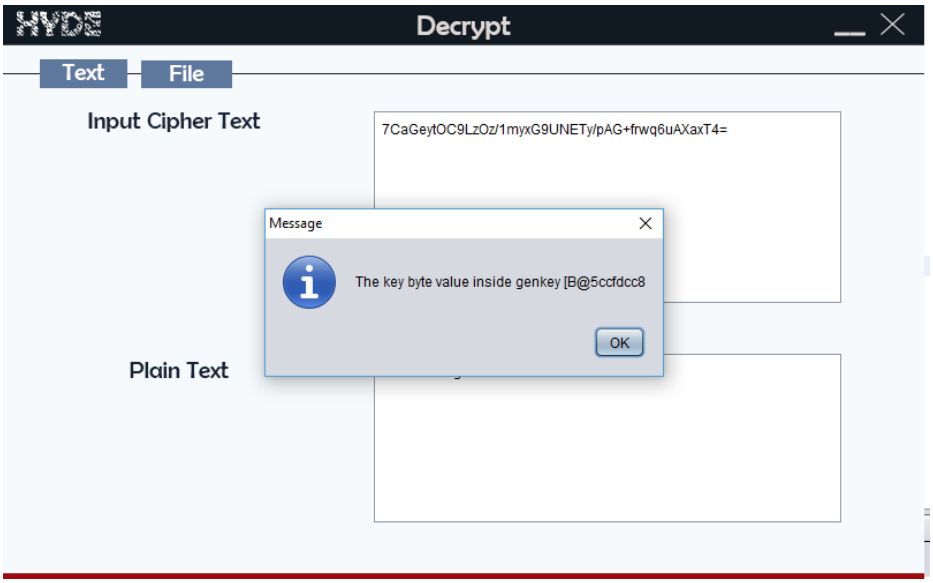

Figure 21: showing the key value generated. The key generated during the decryption process is displayed which also notifies the user of the successful completion of the encryption process.

Plain Text Generated: Figure 22 shows the plain texts generated by the Hyde

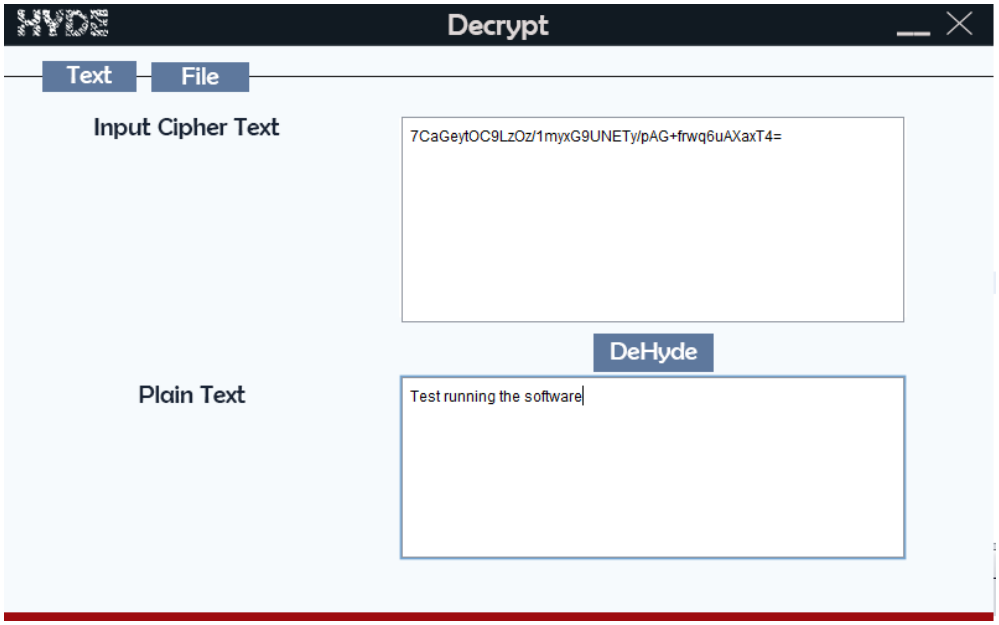

Figure 22: showing the plain text generated. On click of the "DeHyde" button it validates the form and the plain text (output) is displayed in the plain text field. The output can be stored or used for further processing.

4. CONCLUSION

We are in a time where technology is growing at a fast rate so also the amount of data produced is also on the increase. Hence, there is need to ensure that data transmitted is secured.

The importance of secured communication in all sectors today cannot be overemphasized because of the fact that all industries are going digital and there is the need to secure this data.

The development of the Hyde system has helped address this issue. The Hyde system is a tool that provides data integrity services for vital or personal files and texts. It provides additional layer of security for sensitive documents. If the security of a computer system is breached, physically or remotely, a sophisticated encrypted file or text will prevent unauthorized access thereby deterring subsequent attacks.

Future scope of this idea is to develop an integrated encryption tool as a component or service within the operating system of a computer thereby providing a seamless user experience that makes it easier for users to utilize the service. The future development will attempt to provide a service which integrates an option to the short-cut menu of a file object in order to encrypt a plain text/file or decrypt an encrypted file.

5. REFERENCES

[1] V. Agrawal, S. Agrawal, and R. Deshmukh, "Analysis and review of encryption and decryption for secure communication," *Int. J. Sci. Eng. Res.*, vol. 2, no. 2, 2014.

[2] A. Mathur, "A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 9, p. 1650, 2012.

[3] Christensson, "File Definition," 2007. .

[4] N.-F. Standard, "Announcing the advanced encryption standard (AES)," *Fed. Inf. Process. Stand. Publ.*, vol. 197, pp. 1–51, 2001.

[5] R. Padate and A. Patel, "Encryption and decryption of text using AES algorithm," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 5, pp. 54–59, 2014.

[6] D. Selent, "Advanced encryption standard," *Rivier Acad. J.*, vol. 6, no. 2, pp. 1–14, 2010.

[7] M. Pitchaiah and P. Daniel, "Implementation of advanced encryption standard algorithm," 2012.

[8] J.-Y. Oh, D.-I. Yang, and K.-H. Chon, "A selective encryption algorithm based on AES for medical information," *Healthc. Inform. Res.*, vol. 16, no. 1, pp. 22–29, 2010.

[9] R. Rayarikar, S. Upadhyay, and P. Pimpale, "SMS encryption using AES algorithm on android," *Int. J. Comput. Appl.*, vol. 50, no. 19, pp. 12–17, 2012.

[10] M. B. Renardi, N. C. Basjaruddin, and E. Rakhman, "Securing electronic medical record in Near Field Communication using Advanced Encryption Standard (AES)," *Technol. Heal. Care*, no. Preprint, pp. 1–6, 2018.

[11] R. Doomun, J. Doma, and S. Tengur, "AES-CBC software execution optimization," in *Information Technology, 2008. ITSim 2008. International Symposium on*, 2008, vol. 1, pp. 1–8.

[12] C. Kaufman, "Radia perlman, and Mike Speciner," *Netw. Secur. Priv. Commun. a Public World. Pearson Educ.*, 2002.

[13] N. Sklavos and O. Koufopavlou, "Architectures and VLSI implementations of the AES-proposal Rijndael," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1454–1459, 2002.

[14] J. Daemen and V. Rijmen, "AES Proposal: Rijndael. AES Algorithm Submission, September 3, 1999," *URL http//www. nist. gov/CryptoToolKit*, 1999.