# Encryption And Decryption Of Text File Using LabVIEW

Balaji Tata, Surya Prasada Rao Borra, Geetha Devi Appari

PVP Siddhartha Institute of Technology

***ABSTRACT*** **-**Data encryption is employed to provide security to confidential data which thereby denies any unauthorized access. As information travels over the Internet, it is necessary to scrutinize the access from unauthorized organizations or individuals. Due to this, the data is encrypted to reduce data loss and theft. Few common items that are encrypted include text files, images, e-mail messages, user data and directories. The recipient of decryption receives a prompt or window in which a password can be entered to access the encrypted data. For decryption, the system extracts and converts the garbled data and transforms it into words and images that are easily understandable not only by a reader but also by a system. Decryption can be done manually or automatically. It may also be performed with a set of keys or passwords.

In this paper, an efficient design is implemented to encrypt any text file and decrypt it using Fast Fourier Transform and Inverse Fast Fourier Transform algorithms respectively. The design is implemented in LabVIEW software. The basic operation in the encryptor module includes taking the input as text file and converting the data into ASCII values and applying Fast Fourier Transform. To deny unauthorized access, security key is added to the data and obtained output is copied into a file for transmission. In the decryptor module the received data is taken as input and decoded with the security key and Inverse Fast Fourier Transform is applied. The resultant ASCII values are converted into their corresponding string format.

***KEYWORDS:*** *Encryption, Decryption, Message, ASCII code, Fast Fourier Transform, Inverse Fast Fourier Transform.*

## INTRODUCTION

Encryption has long been used by militaries and governments to facilitate secret communication. It is now commonly used in protecting information within many kinds of civilian systems. For example, the Computer Security Institute reported that in 2007, 71% of companies surveyed utilized encryption for some of their data in transit, and 53% utilized encryption for some of their data in storage[1]. Encryption can be used to protect data "at rest", such as files on computers and storage devices (e.g. USB flash drives). In recent years there have been numerous reports of confidential data such as customers' personal records being exposed through loss or theft of laptops or backup drives. Encrypting such files at rest helps protect them should physical security measures fail [2]. Digital rights management systems which prevent unauthorized use or reproduction of copyrighted material and protect software against reverse engineering are another somewhat different example of using encryption on data at rest.

Encryption is also used to protect data in transit, for example data being transferred via networks (e.g. the Internet, e-commerce), mobile telephones, wireless microphones, wireless intercom systems, Bluetooth devices and bank automatic teller machines. There have been numerous reports of data in transit being intercepted in recent years[3]. Encrypting data in transit also helps to secure it as it is often difficult to physically secure all access to networks. Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a message authentication code (MAC) or a digital signature.

Standards and cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem. A single slip-up in system design or execution can allow successful attacks[4]. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption.

**Table 1. ASCII Codes**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NUL | 16 | DLE | 32 | SPC | 48 | 0 | 64 | @ | 80 | P | 96 | ` | 112 | p |
| 1 | SOH | 17 | DC1 | 33 | ! | 49 | 1 | 65 | A | 81 | Q | 97 | A | 113 | q |
| 2 | STX | 18 | DC2 | 34 | “ | 50 | 2 | 66 | B | 82 | R | 98 | B | 114 | r |
| 3 | ETX | 19 | DC3 | 35 | # | 51 | 3 | 67 | C | 83 | S | 99 | C | 115 | s |
| 4 | EOT | 20 | DC4 | 36 | $ | 52 | 4 | 68 | D | 84 | T | 100 | D | 116 | t |
| 5 | ENQ | 21 | NAK | 37 | % | 53 | 5 | 69 | E | 85 | U | 101 | E | 117 | u |
| 6 | ACK | 22 | SYN | 38 | & | 54 | 6 | 70 | F | 86 | V | 102 | F | 118 | v |
| 7 | BEL | 23 | ETB | 39 | ‘ | 55 | 7 | 71 | G | 87 | W | 103 | G | 119 | w |
| 8 | BS | 24 | CAN | 40 | ( | 56 | 8 | 72 | H | 88 | X | 104 | H | 120 | x |
| 9 | HT | 25 | EM | 41 | ) | 57 | 9 | 73 | I | 89 | Y | 105 | I | 121 | y |
| 10 | LF | 26 | SUB | 42 | * | 58 | : | 74 | J | 90 | Z | 106 | J | 122 | z |
| 11 | VT | 27 | ESC | 43 | + | 59 | ; | 75 | K | 91 | [ | 107 | K | 123 | { |
| 12 | FF | 28 | FS | 44 | , | 60 | < | 76 | L | 92 | \ | 108 | L | 124 | \| |
| 13 | CR | 29 | GS | 45 | - | 61 | = | 77 | M | 93 | ] | 109 | M | 125 | } |
| 14 | SO | 30 | RS | 46 | . | 62 | > | 78 | N | 94 | ^ | 110 | N | 126 | ~ |
| 15 | SI | 31 | US | 47 | / | 63 | ? | 79 | O | 95 | _ | 111 | O | 127 | DEL |

**ENCRYPTION**

Data encryption translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it. Encrypted data is commonly referred to as cipher text, while unencrypted data is called plaintext. Currently, encryption is one of the most popular and effective data security methods used by organizations. Two main types of data encryption exist – asymmetric encryption, also known as public-key encryption, and symmetric encryption[5].

The purpose of data encryption is to protect digital data confidentiality as it is stored on computer systems and transmitted using the internet or other computer networks. The outdated data encryption standard (DES) has been replaced by modern encryption algorithms that play a critical role in the security of IT systems and communications.

These algorithms provide confidentiality and drive key security initiatives including authentication, integrity, and non-repudiation. Authentication allows for the verification of a message’s origin, and integrity provides proof that a message’s contents have not changed since it was sent. Additionally, non-repudiation ensures that a message sender cannot deny sending the message.

Data, or plaintext, is encrypted with an encryption algorithm and an encryption key. The process results in cipher text, which only can be viewed in its original form if it is decrypted with the correct key [6]. Symmetric-key ciphers use the same secret key for encrypting and decrypting a message or file. While symmetric-key encryption is much faster than asymmetric encryption, the sender must exchange the encryption key with the recipient before he can decrypt it. As companies find themselves needing to securely distribute and manage huge quantities of keys, most data encryption services have adapted and use an asymmetric algorithm to exchange the secret key after using a symmetric algorithm to encrypt data.

On the other hand, asymmetric cryptography, sometimes referred to as public-key cryptography, uses two different keys, one public and one private. The public key, as it is named, may be shared with everyone, but the private key must be protected. The Rivest-Sharmir-Adleman (RSA) algorithm is a cryptosystem for public-key encryption that is widely used to secure sensitive data, especially when it is sent over an insecure network like the internet[7]. The RSA algorithm’s popularity comes from the fact that both the public and private keys can encrypt a message to assure the confidentiality, integrity, authenticity, and non-reputability of electronic communications and data through the use of digital signatures.

The most basic method of attack on encryption today is brute force, or trying random keys until the right one is found. Of course, the length of the key determines the possible number of keys and affects the plausibility of this type of attack. Alternative methods of breaking a cipher include side-channel attacks and cryptanalysis. Side-channel attacks go after the implementation of the cipher, rather than the actual cipher itself. These attacks tend to succeed if there is an error in system design or execution[8]. Likewise, cryptanalysis means finding a weakness in the cipher and exploiting it.

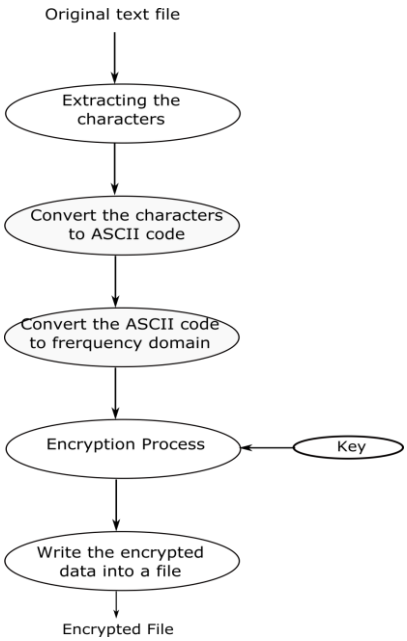


**Fig.1. System flow of Encryption**

**DECRYPTION**

The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password. In other words, Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. Some companies also encrypt data for general protection of company data and trade secrets[9].

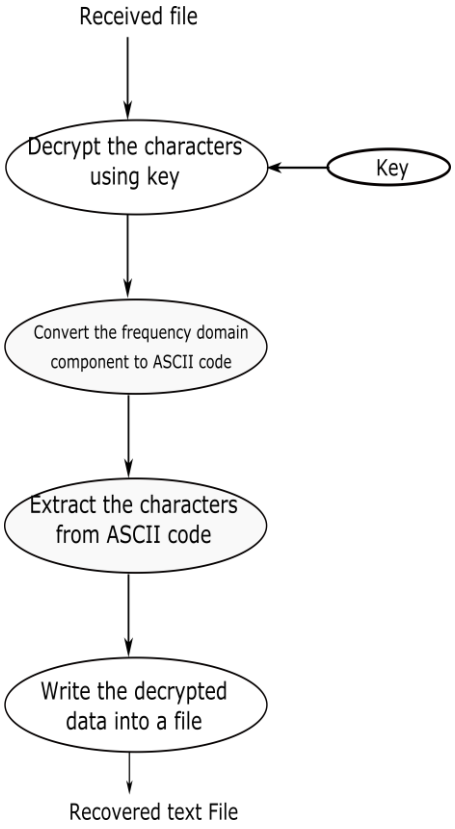


**Fig.2. System flow of Decryption**

**ALGORITHM FOR ENCRYPTION**

- The Encryptor module provides the facility of encrypting a text file, i.e. converting it into a non-readable form.
- The text file that has to be encrypted is first created.
- The location of this file is provided. Each character of the data is extracted and is further converted into ASCII codes.
- The codes are subjected to FFT algorithm for further encryption.
- A series of complex numbers are produced as the result.
- The output obtained after carrying out the FFT algorithm is the required encrypted output.
- The encrypted output is saved and stored in a new file location

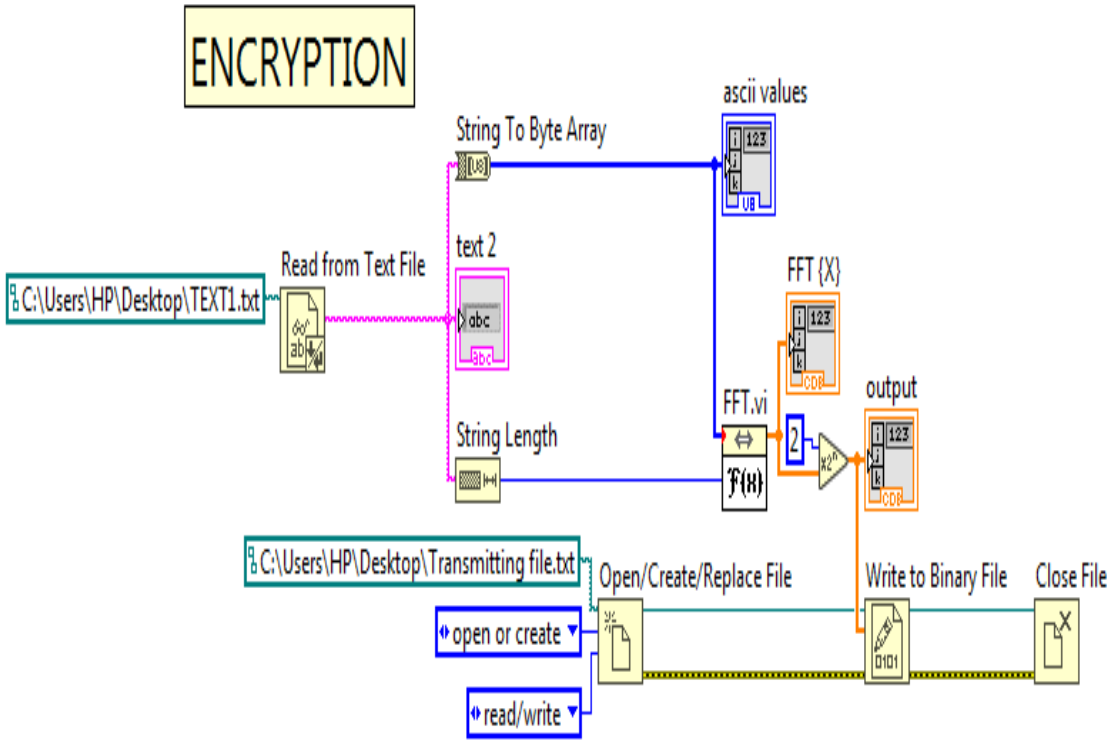


**Fig.3. LabView Block Diagram for Encryption**

**ALGORITHM FOR DECRYPTION**

- The Decryptor module is the other vital part of the application where the encrypted text files are received from encryptor.
- This module designed for the application avail the cryptographer with user friendly and security options.
- Initially, the user needs to provide the path or file location of the data that has been encrypted to so as to convert it into readable form or retrieve it back to its original form.
- The next step is to extract all the coded characters from the required file.
- The file is then made to undergo IFFT algorithm.
- The output of this is the required decrypted output data.
- This is then displayed, saved and stored in another file location.

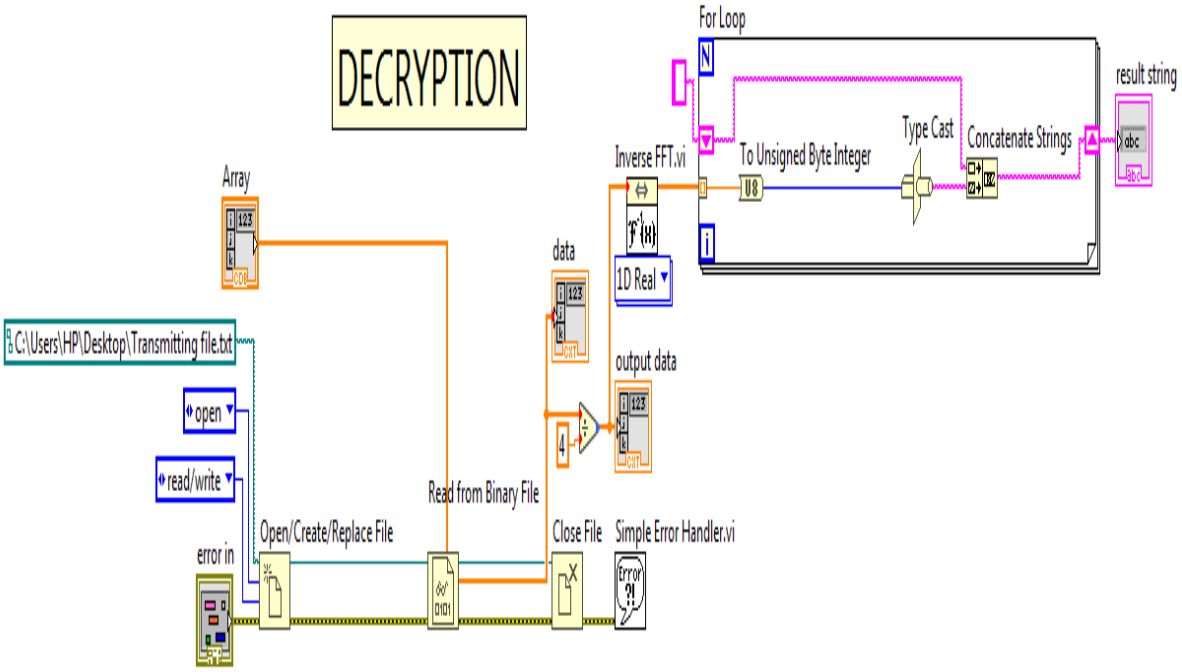


**Fig.4. LabView Block Diagram for Decryption**

**Encryption**

The text file that has to be encrypted is given as the input for encryptor module. Encryption is performed using FFT algorithm and in order to provide security to our confidential data, security key is added to it. The figures shown below indicates the input text file and output of encryption process.

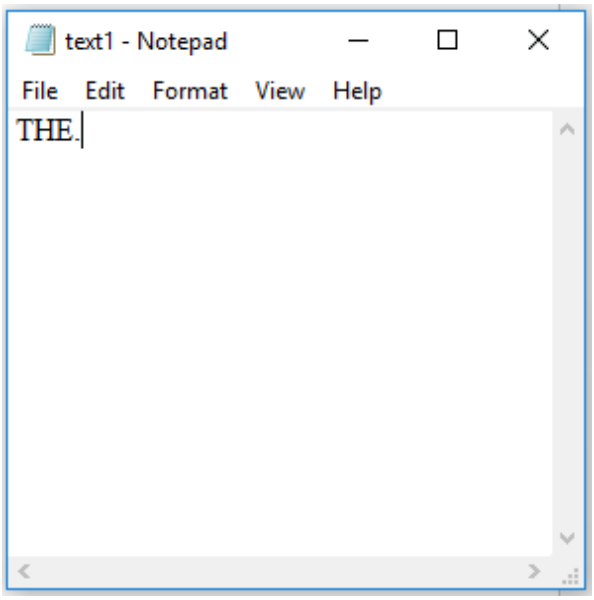


**Fig.5. Text file Input**

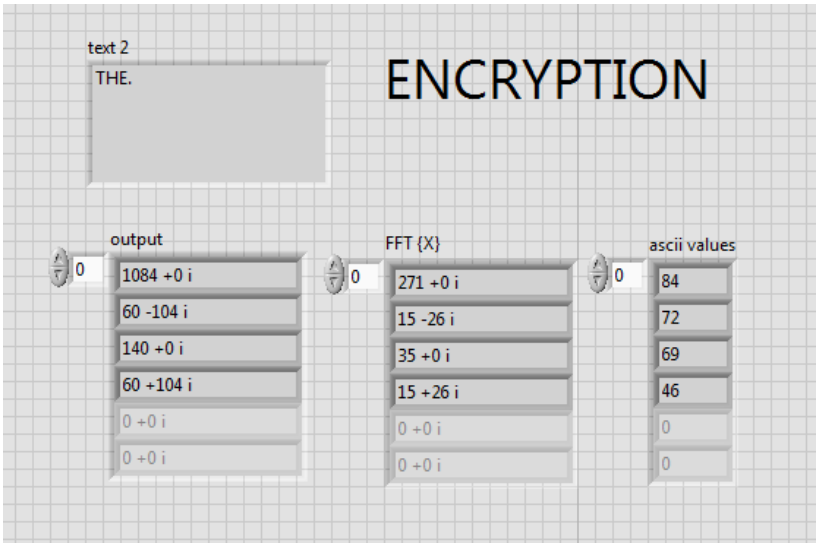


**Fig.6. LabView Front panel of Encrypted output**

**Decryption**

The received encrypted file is given as the input for decryptor module and the data is decoded using security key. Decryption is performed using IFFT algorithm. The ASCII values are converted into its corresponding string format.

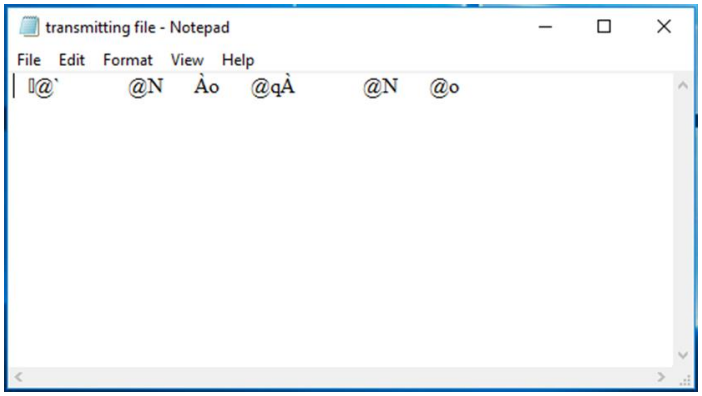


**Fig.7. Received file**

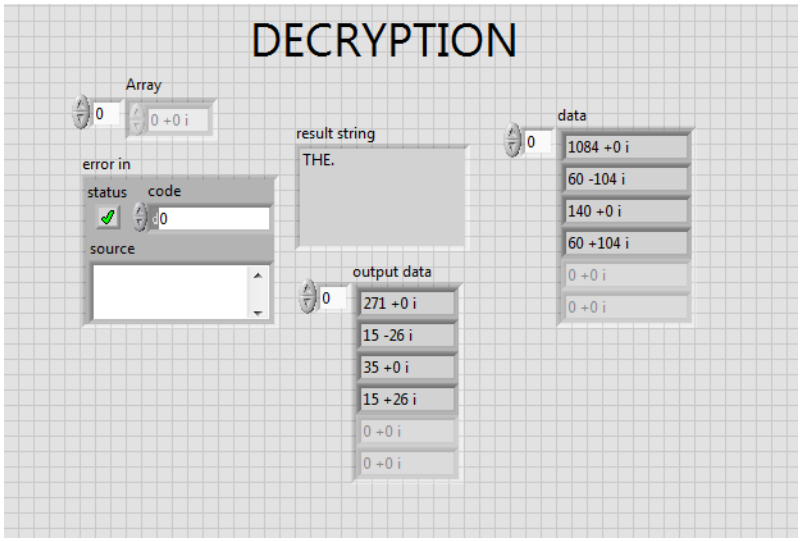


**Fig.8. LabView front panel of Decrypted output**

## CONCLUSION

The original data that is transmitted through encryption is same as the retrieved data after the decryption process. As information security is one of the growing concern in various fields like nation security, defense, banking and protection of various confidential data. The application designed provides a very high security in transmission of a text file.

This application can be used as an effective technique in programming scenarios to secure vital codes and secure transmission in research departments. The application possesses immense scope for further development. The development relies on the factors providing transmission security. The application can be further designed for word documents and other types of text files (pdf, word document, etc.).

## REFERENCES:

**[1]** Quist-Aphetsi Kester,"Image Encryption based on the RGB PIXEL Transposition and Shuffling",IJCNIS, vol.5, no.7, pp.43-50,2013.

[2] Kester, Q. A., & Danquah, P. (2012, October). A novel cryptographic key technique. In Adaptive Science & Technology (ICAST), 2012 IEEE 4th International Conference on (pp. 70-73). IEEE.

[3] Symmetric key Cryptography using two-way updated – Generalized Vernam Cipher method: TTSJA algorithm, International Journal of Computer Applications(IJCA, USA), Vol 42, No.1, March, Pg: 34 -39( 2012)

[4] Satyaki Roy , Shalabh Agarwal, Asoke Nath, Navajit Maitra and Joyshree Nath," Ultra encryption algorithm (UEA): Bit level symmetric key cryptosystem with randomized bits and feedback mechanism", International Journal of Computer Applications (0975 – 8887) Volume 49– No.5, July 2012.

[5] sruthi s, athira vijay, shejo jose, athira v, Encryption & Decryption of Text file and Audio using LabVIEW, "2017 International Conference on Networks & Advances in Computational Technologies (NetACT) |20-22 July 2017| Trivandrum" pp. 462-466.

[6] Ultra Encryption Standard(UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method, Satyaki Roy, Navajit Maitra, Joyshree Nath,Shalabh Agarwal and Asoke Nath, Proceedings of IEEE sponsored National Conference on Recent Advances in Communication, Control and Computing Technology-RACCCT 2012, 29-30 March held at Surat, Page 81-88(2012).

[7] Gang Hu, "Study of file encryption and decryption system using security key," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 2010, pp. V7-121-V7-124.

[8] Zhan Lei and Zhao Jing, "A hardware encryption and decryption system design," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 2010, pp. V5-75-V5-77.

[9] S. Wang and G. Liu, "File Encryption and Decryption System Based on RSA Algorithm," 2011 International Conference on Computational and Information Sciences, Chengdu, China, 2011, pp. 797-800