THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Data Solution

Minal Savant
B.E., Information Technology Department
Vidyavardhini's College of Engineering & Technology, Vasai, Maharashtra, India
Jyotsna Devale
B.E., Information Technology Department
Vidyavardhini's College of Engineering & Technology, Vasai, Maharashtra, India
Urvi Modi
B.E., Information Technology Department
Vidyavardhini's College of Engineering & Technology, Vasai, Maharashtra, India
Sharon Coelho
Professor, Information Technology Department
Vidyavardhini's College of Engineering & Technology, Vasai, Maharashtra, India

Abstract:
Today several application produce huge amount of data, making them available for post-processing mechanism to access data & high performance computing to obtain the result in an acceptable time wrapping the application as web services allows interoperability with other tools and in particular with grid computing environment exploiting large set of resources to the standard interface to support the so called "data intensive" that handles large amount of data. In order to aid compression in order to transfer one big file from one end to another through any media like internet or small storage so, proposed system presents a compression algorithm for dynamic data transfer. Splitting which is used to split the user specifying file according to the user specifying size. Generally for security purposes, breaking the password in two & storing them in two places or transfer them as two will make password exchange more secure. Hence the proposed splitting technique will enable 'secret splitting' for places where security concern is the prime need.

Keywords: *Compression, decompression, split, merge, encryption, decryption*

1. Introduction

Our System provides the features such as splitting, merging, compression and extraction of various types of file formats such as text file, audio file, video file, etc. It also provides security in the form of encryption and decryption. The most important aspect of the system is that, it provides an option which allows the user to perform the specific operation on the specified file without affecting the original file. Thus, in case of any error or mistake while performing the ordered function will not result into loss of data. The software data solution will have data compression –decompression & data splitting & merging features/ function in one single application with improved efficiency and increased precision.

Data compression or source coding is the process of encoding information using fewer bits (or other information-bearing units) than an un-encoded representation would use, through use of specific encoding schemes. Reconversion of compressed data into its original (or closely to original) form so that it can be heard, read, and/or seen as normal.[1]

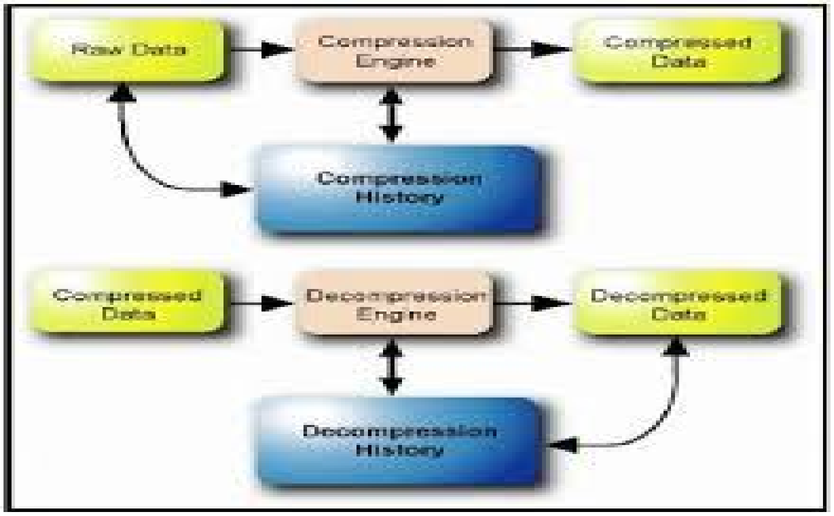


Figure 1: Compression Decompression Process

Data splitting is an approach of protecting sensitive data from unauthorized access by encrypting the data and storing different portions of a file on different servers. When split data is accessed, the parts are retrieved, combined and decrypted. Data splitting can be made even more effective by periodically retrieving and recombining the parts, and then splitting the data in a different way among different servers, and using a different encryption key. Thus, even if a hacker makes progress towards obtaining split data, chances are that the data will have been reorganized before the hacker manages to obtain all the necessary components. By rearranging the data often enough, a network administrator can stay ahead of even the most adept hacker. Merging combining the split data to form the original data or file is known as Merging of data.[2]

Encryption is the process of transforming information using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information contexts, the word encryption also implicitly refers to the reverse process. Decryption is the process of decoding the data that has been encrypted into a secret format.[3]

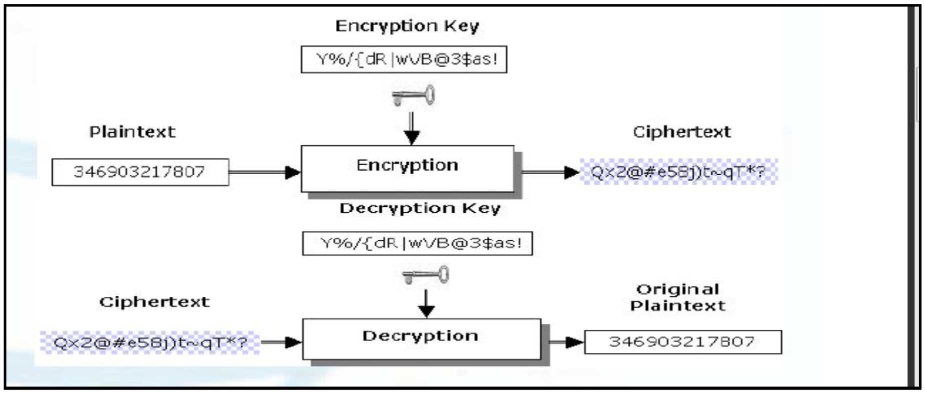


Figure 2: Encryption/Decryption Process

2. Literature Survey

Various compressive investigation on the existing compression-decompression , encryption- decryption and split-merge scheme have been accomplish, it has been discerned that none of recent software schemes cannot provide all the features in one application. With this outcome paper proposes various schemes like compression-decompression, compression-decompression and split-merge which overcome all existing software schemes. Literature review reveals all the studies that are done in past. Some technique are discussed as follow

2.1. Compression-Decompression

2.1.1. Huffman Coding Algorithm

A Huffman Coding is more sophisticated and efficient lossless data compression technique. In Huffman Coding the characters in a data file are converted to binary code. And in this technique the most common characters in the file have the shortest binary codes, and the least common have the longest binary code [4]

2.1.2. Arithmetic Coding Algorithm

Arithmetic encoding is the most powerful compression techniques. This converts the entire input data into a single floating point number. A floating point number is similar to a number with a decimal point, like 4.5 instead of 41/2. However, in arithmetic coding we are not dealing with decimal number so we call it a floating point instead of decimal point [5].

2.2. Split-Merges

The split-apply-combine strategy is similar to the map-reduce strategy for processing large data, recently popularized by Google. In map-reduce, the map step corresponds to split and apply, and reduce corresponds to combine, although the types of reductions are much richer than those performed for data analysis. Map-reduce is designed for a highly parallel environment, where work is done by hundreds or thousands of independent computers, and for a wider range of data processing needs than just data analysis.[6]

2.3. Encryption-decryption

In network security, cryptography has a long history by provides a way to store sensitive information or transmit it across insecure networks (i.e. the Internet) so that it cannot be read by anyone except the intended recipient, where the cryptosystem is a set of algorithms combined with keys to convert the original message (Plain-text) to encrypted message (Cipher-text) and convert it back in the intended recipient side to the original message (Plain-text) [7]. The first model proposed by Shannon on the cryptosystem is shown in figure [8].

3. Proposed system

In our approach we combines all advantages of compression- decompression, splitting-merging and encryption-decryption. The unique feature of this application is its simplicity, to reduce time in downloading different software, and to send error free data over web we using encryption & decryption of data so that both sender and receiver get original data at both the end. Our system provides all this operation in single application with better efficiency.

3.1. System Architecture

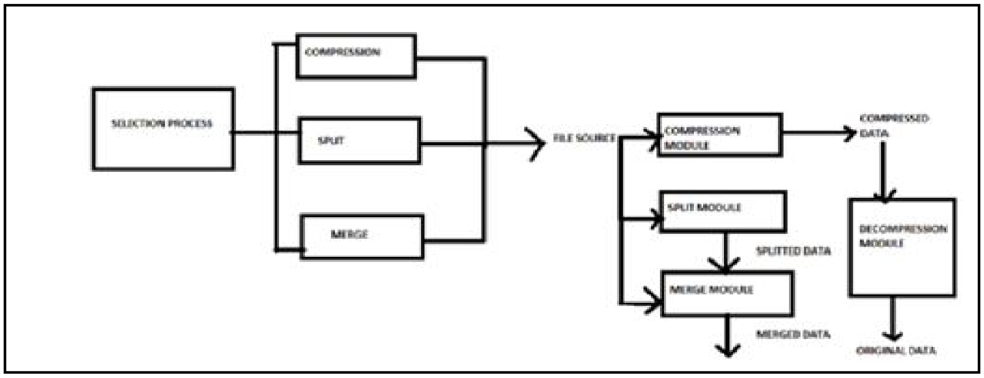


Figure 3: System Architecture

In our approach, we allow the user to select file first, selection process in which user need to select type of operation he/she wishes to perform. After selection user need to give file’s path and destination path where output will be stored. Finally it will give output on user’s destination path.

3.2 Proposed Algorithm

3.2.1 Compression-Decompression Algorithm [9]

Using this algorithm instead of reducing five bit we try to reduce up to four bits

1. If file is to be compress the Pick up a plain text or file which contain the text.
    - For example: - “abcdefgh”
2. Convert each character with its ASCII code
    - |97|98|99|100|101|102|103|104|
3. For conversion to 5-bit, let’s assign new values to the characters.
    - For e.g | a = 1 | b=2 | c=3 | d=4 | e=5 | f=6 | g=7 | h=8 |
4. Obtain binary code of each decimal number.
    - 00000001|00000010|00000011|00000100|00000101|00000110|00000111|00000100|
5. Remove MSB 4 bits from string which all contains 0000.
6. 0001|0010|0011|0100|0101|0110|0111|0100|
7. Rearrange bits in an array of bytes as follows:
    - 00010010|00110100|01010110|01110100|

3.2.2. Split & Merge Algorithm

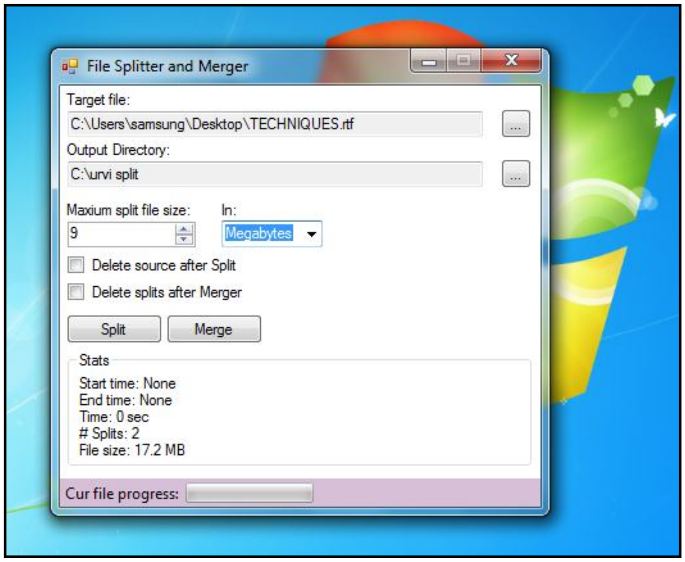
```
•  If file is to be split go to step 2 else merge the fragments of the file and go to step 10.
•  Input  srcpath, destnpath, sof ,buffersize
•  Initialize Status
•  size= size of source file
     ➢ Print size
•  Check buffersize and update status
•  If size>sof  go to step 6  else print file cannot be split and goto step 10
•  Split into fragment=sof
•  size=size-sof
•  If size<sof  goto step 6
•  Merge file,
     ➢ Check buffer size
     ➢ Update Status
     ➢ Take one file from buffer and merge them uptil buffersize is equal to original file size.
•  End
```

3.2.3. Encryption and Decryption Algorithm

```
•  If want to encrypt go to step2 else to step 5
•  Encrypt a byte array into a byte array using a key and an IV
•  Create a MemoryStream that is going to accept the encrypted bytes
•  Create CryptoStream
•  Returns an encrypted string using Rijndael (128,192,256 Bits).
•  Decrypt file,
     ➢ Decrypt a byte array into a byte array using a key and an IV
     ➢ Return Decrypted string.
•  End
```

4. Implementation and Results

4.1. Split and Merge



Target file: C:\Users\samsung\Desktop\TECHNIQUES.rtf

Output Directory: C:\urvi split

Maxium split file size: 9 In: Megabytes

□ Delete source after Split

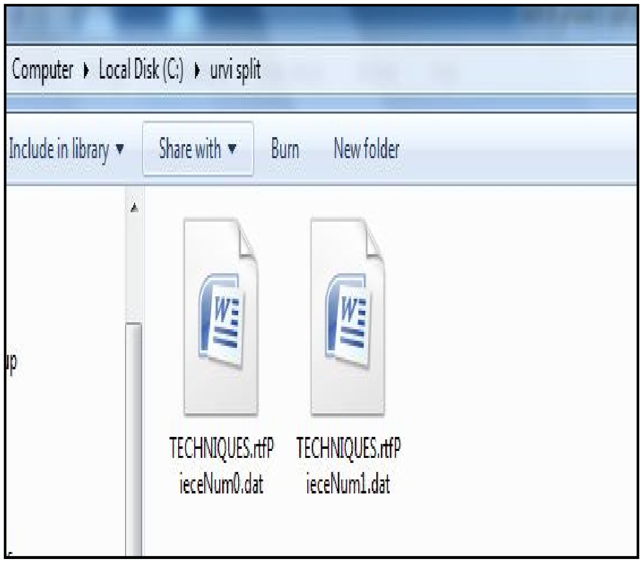
□ Delete splits after Merger

Stats

Start time: None
End time: None
Time: 0 sec
Splits: 2
File size: 17.2 MB

Cur file progress:

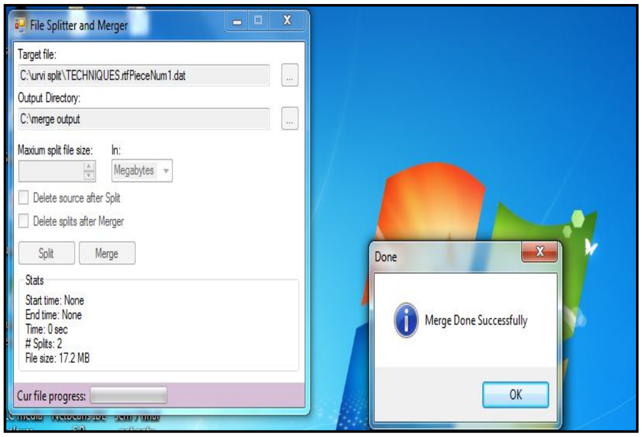
Figure 4



TECHNIQUES.rtfPieceNum0.dat

TECHNIQUES.rtfPieceNum1.dat

Figure 5



Target file: C:\urvi split\TECHNIQUES.rtfPieceNum1.dat

Output Directory: C:\merge output

Maxium split file size:

In: Megabytes

□ Delete source after Split

□ Delete splits after Merger

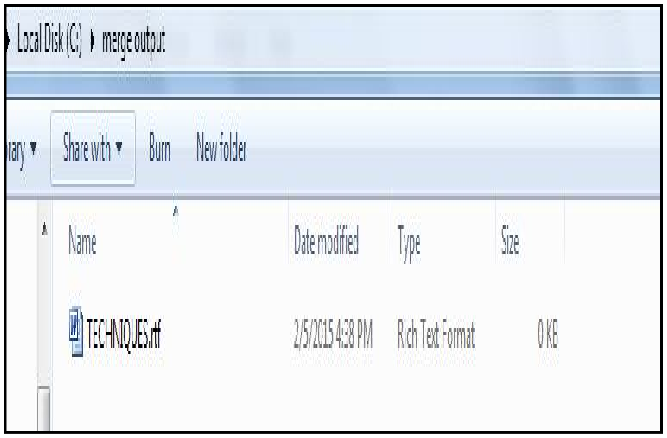
Stats

Start time: None
End time: None
Time: 0 sec
Splits: 2
File size: 17.2 MB

Cur file progress:

Merge Done Successfully

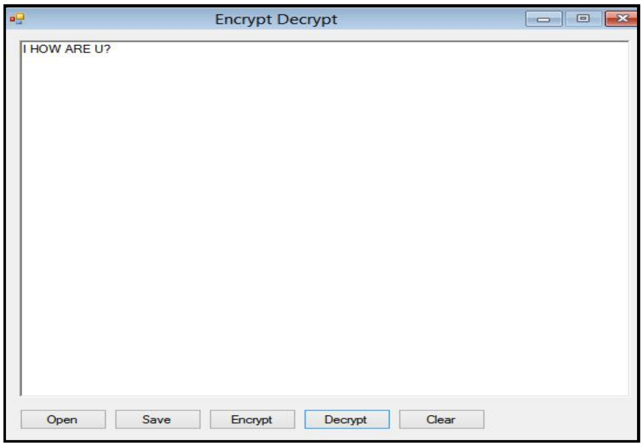
Figure 6



Name	Date modified	Type	Size
TECHNIQUES.rtf	2/5/2015 4:38 PM	Rich Text Format	0 KB

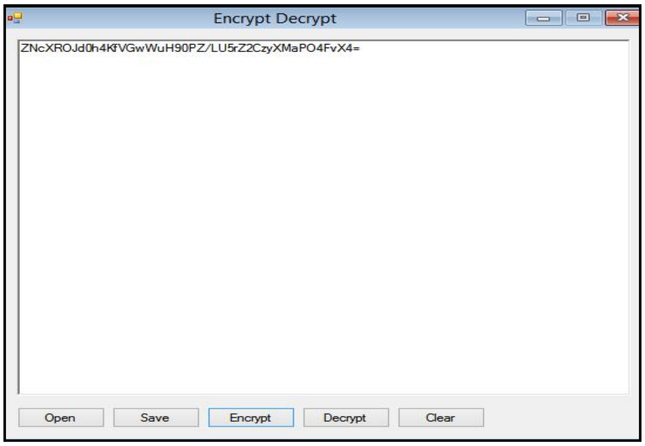
Figure 7

4.2. Encryption & Decryption



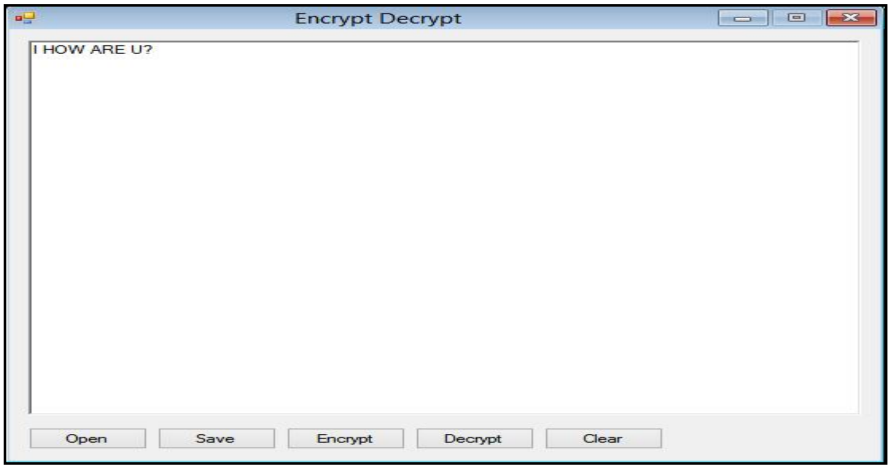
I HOW ARE U?

Figure 8: Before encryption



ZNcXROJd0h4KfVGwWuH90PZ/LU5rZ2CzyXMaPO4FvX4=

Figure 9: After encryption



I HOW ARE U?

Figure 10: After decryption

4.3. Compression-Decompression

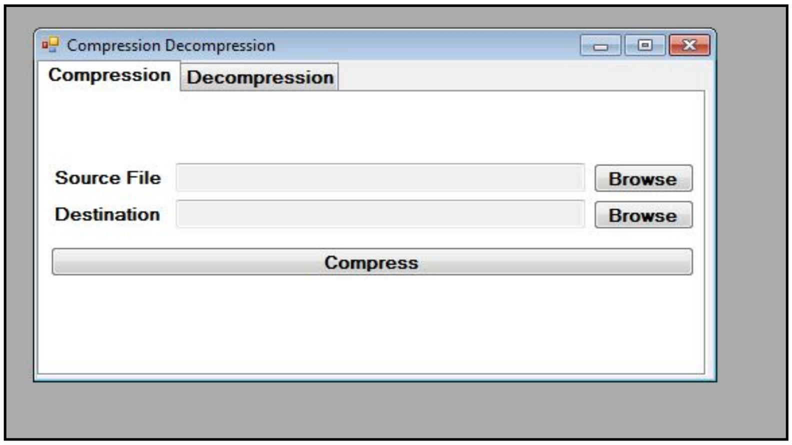


Figure 11

5. Conclusion

Our system provides combination of different features of data which help user to manage data more efficiently. The algorithm of compression and decompression we have used is the technique of ’saving bits’ is employed in this algorithm .We are reducing bits up to 4 bytes. Split and merge algorithm is used for splitting and merging the files according to user specified data size. We have used AES algorithm for encryption & decryption. Existing system provides the compression ratio in between the range of 60 to 75 percent and our system compression ratio in the range between 90 to 95 percent

We had searched lots of topics, finally we got an idea ‘Data Solution’ to combine three different techniques in one system. For this topic we have searched lots of papers. While implementing compression module the main challenge was to provide good compression ratio and while implementing split-merge module the challenge was to open split file, after lots of efforts we ultimately open .txt file.

In future we will try to achieve 100 percent compression ratio to open various file format like pdf, rtf, etc. after splitting.

6. Acknowledgement

It is nearly impossible to thank everyone who ever helps us in the process of taking of ‘Data Solution’ to live running model. This paper is a direct result of all people who directly or indirectly helped us realizing our goal. We are heartily thankful to our guide Prof. Sharon Coelho for noble guidance and moral support. Her suggestion timely advised, inspired us toward sustain for our project work.

7. References

i. S.Gavaskar, Dr.E.Ramaraj, R.Surendiran, “A Compressed Anti IP Spoofing Mechanism Using Cryptography ,“ IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.11, November 2012

ii. searchsecurity.techtarget.com

iii. International Journal of Scientific Engineering and Research (IJSER) www.ijser.in ISSN (Online): 2347-3878 Volume 2 Issue 2, February 2014

iv. Ken Huffman. Profile: David A. Huffman, Scientific American, September 1991, pp. 54–58

v. Blelloch, E., 2002. Introduction to Data Compression, Computer Science Department, Carnegie Mellon University.

vi. Journal of Statistical Software April 2011, Volume 40, Issue 1. http://www.jstatsoft.org

vii. P. Zimmerman, "An Introduction to Cryptography", Doubleday & Company, Inc., United State of America, USA, 1999.

viii. C. Shannon, “Communication Theory of Secrecy Systems”, Bell Systems Technical Journal, MD Computing, vol. 15, pp. 57-64, 1998.

ix. International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X Volume 3, Issue 3, March 2013