# Compression and Encryption: An Integrated Approach

**Dr. Mukesh Sharma[1], Smiley Gandhi[2]**

[1]Associate Professor, T.T.T. & S, Bhiwani, India

[2]M.Tech Scholar, T.T.T.&S,Bhiwani, India

E-mail: smilegandhi@gmail.com

## Abstract

*This paper considers different aspects related to two different but not opposite direction-Data Compression and Cryptography. In this paper a brief review of both of these branches-compression and encryption, need of data compression and data encryption has been discussed. Also, the need of integrating these two branches has been addressed. There occurs a problem of the order of applying these two processes i.e.; Compression should be applied first before Encryption or Encryption should be applied first before compression. This paper discusses this issue of order also of these two processes.*

## 1. A BRIEF REVIEW

### 1.1 Cryptography

Cryptography is the field of technologies for making the information secure. It tries to secure, encrypt the information in such a way that a third party who has access to the hidden, encrypted, data cannot reconstruct, decrypt, the original information. In more practical terms, the encryption methods apply a certain function, algorithm or a routine to information so that it's no longer accessible as it was original. With the right key, that was determined before encrypting the data and the corresponding algorithm, function or routine for decrypting the information, the original information can be recovered. Cryptography was first developed many centuries ago. It was the work of specialists to create encryption algorithms for the military purposes mainly. Nowadays it's being used all around us; in ATM cards, on ecommerce websites, in game consoles, for the distribution of copyrighted music and film and many more applications. This is all possible due to the increase of the use of computer and readily available large amounts of computing power. Encryption is the technique of the translation of data into a secret code. It is the most effective way to achieve data security. In this process data is converted into a form, called a cipher text that cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood. Encryption changes data so that it is meaningless to anyone who does not have a key to unscramble it. For example, 'HelloWorld' might be changed to "2kdi678674234". After you encrypt data, only you and the people you choose can decrypt the information to make it readable again.

### Types of Cryptography

There are two main types of cryptography:

- Secret key cryptography
- Public key cryptography

**Secret key cryptography** is also known as *symmetric key* cryptography. With this type of cryptography, both the sender and the receiver know the same secret code, called the key. Messages are encrypted by the sender using the key and decrypted by the receiver using the same key. This method works well if you are communicating with only a limited number of people, but it becomes impractical to exchange secret keys with large numbers of people. In addition, there is also the problem of how you communicate the secret key securely.

**Public key cryptography**, also called *asymmetric encryption*, uses a pair of keys for encryption and decryption. With public key cryptography, keys work in pairs of matched public and private keys. The public key can be freely distributed without compromising the private key, which must be kept secret by its owner. Because these keys work only as a pair, encryption initiated with the public key can be decrypted only with the corresponding private key. The major advantage asymmetric encryption offers over symmetric key cryptography is that senders and receivers do not have to communicate keys up front. Provided the private key is kept secret, confidential communication is possible using the public keys.

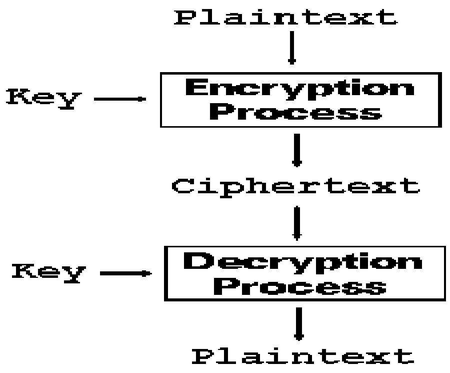

Fig. Encryption and Decryption processes (Image from decodesystems.com)

### 1.2 Compression

In simple words, Data Compression refers to the technique of storing data in a format that requires less space than it takes usual. Compression is the reduction in size of data in order to save space and transmission time. For data transmission, compression can be performed on the data content. Data Content compression can be as simple as removing all extra space characters, inserting a single repeating character to indicate a sequence of repeated characters, and substituting or replacing smaller bit strings for frequently occurring characters. This kind of compression can reduce a text file to 50% of its original size. Compression is performed by a program that uses a formula or algorithm to determine how to compress or decompress data. Data compression is particularly useful in communication systems for the transmission of data because it enables devices to transmit or store the same amount of data in fewer bits. There are a variety of data compression techniques, but only a few have been standardized. The CCITT has defined a standard data compression technique for transmitting faxes (Group 3 standard) and a compression standard for data communications through modems (CCITT V.42*bis*). In addition, there are file compression formats, such as ARC and ZIP. Graphic image file formats are usually designed to compress information as much as possible (since these are very large files). Graphic image compression can be either lossy (some information is lost) or lossless (all information can be restored). Data compression is also widely used in backup utilities, spreadsheet applications, and database management systems. Certain types of data, such as bit-mapped graphics, can be compressed to a small fraction of their normal size.

Compression can be categorized in two broad ways:

### Lossless Compression

Lossless data compression makes use of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. In these type of compression techniques data is compressed and can be (uncompressed) without loss of detail of data or information. These are referred to as bit-preserving or reversible compression systems also. Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the UNIX tool gzip. Lossless compression is used when it is important that the original and the decompressed data be identical. Typical examples are executable programs and source code. Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly. Lossless compression is possible because most real-world data has statistical redundancy. For example, in English text, the letter 'a' is much more common than the letter 'q', and the probability that the letter 'q' will be followed by the letter 'z' is very small. Lossless compression schemes are reversible so that the original data can be reconstructed.

Lossless compression methods may be categorized according to the type of data they are designed to compress. Some main types of targets for compression algorithms are text, images, and sound. In principle, any general lossless compression algorithm (general means that it can handle all binary input) can be used on any type of data, many are unable to achieve significant compression on data that is not of the form that they are designed to deal with. Sound data or audio data, for instance, cannot be compressed well with conventional text compression algorithms.

Most lossless compression programs use two different kinds of algorithms: one which generates a statistical model for the input data, and another which maps the input data to bit strings using this model in such a way that "probable" (e.g. frequently encountered) data will produce shorter output than "improbable" data. Often, only the former algorithm is named, while the second is unspecified.

Statistical modeling algorithms for text (or text-like binary data such as executables) include:

- Burrows-Wheeler transform (BWT; block sorting preprocessing that makes compression more efficient)
- LZ77
- LZW
- PPM

Encoding algorithms to produce bit sequences are:

- Huffman coding
- Arithmetic coding

### Lossy Compression

Lossy data compression does not allow the exact original data to be restored from the compressed data. Video and audio compression techniques are most suited to this form of compression. Lossy compression techniques use source encoding techniques that may involve transform encoding, differential encoding or vector quantization like TIFF and MNG file formats may use either lossless or lossy methods. A lossy data compression method is one where compressing data and then decompressing it retrieves data that may be different from the original, but is "close enough to the original" to be useful in some ways. Lossy data compression is used frequently on the Internet and especially in streaming media and telephony applications. Most lossy data compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively or increasingly loss in the quality. This is in contrast with lossless data compression.

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application. Lossy methods are most often used for compressing sound, images or videos. The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) of lossy video are nearly always far superior to those of the audio and still-image equivalents. Audio can be compressed at 10:1 with no noticeable loss of quality; video can be compressed immensely with little visible quality loss, e.g. 300:1. Lossily compressed still images are often compressed to 1/10th their original size, as with audio, but the quality loss is more noticeable, especially on closer inspection. Lossy data compression algorithms introduce relatively minor differences and represent the picture, video, or audio using fewer bits. Lossy schemes accept some loss of data in order to achieve higher compression.

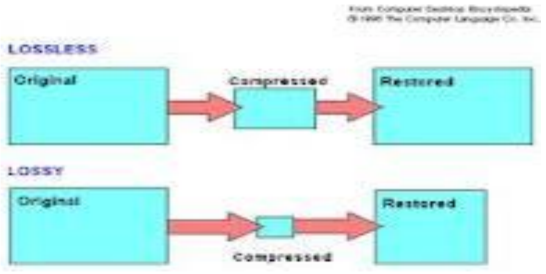

Fig2. Lossless and Lossy Compression techniques (Image form zdnet.com)

## 2. NEED OF ENCRYPTION

Cryptography is needed in order to prevent the interception of data and if in some case, data gets intercepted it should become unreadable for the interceptors or intruders. There are literally thousands of ways to intercept data, but I'll list some common ways.

- **Internet Connections**
  is probably the most dangerous place for your data as concerned with privacy. If you don't use an encrypted connection with the server, it is possible that anybody can access your total communication.
- **Electronic mail transfer**
  has much the same things to deal with as for internet.
- **Wi-Fi**
  It is extremely dangerous. This is because

the information is just put straight into the air for anybody to receive. Can you imagine what happens if this is unencrypted, or encrypted with some weak encryption scheme? Your whole data will be in hands of interceptors.

- **SMSs**
  is also as weak as the whole rest of the internet.
- **External hard drives**
  these have the same issues as USB-sticks except for that less of them are out there who actually try and protect your data
- **USB peripheral devices**
  might get stolen or lost. Even encrypted and supposedly safe USB sticks might very well turn out to be very insecure after all. Copying data to these devices is not the problem but losing them is a problem. Because they are much smaller than laptops or PDAs, they are easier to lose in hotels, taxis, airplanes, restaurants, and other locations frequented by business travelers.
- **Desktop PCs**
  get stolen. But a bigger risk might be that other people use them as well. If you had protected your account with password, even then it is not safe 100%. If someone has full access to the hardware, then shoud be a need of security measure.
- **Laptops**
  are the same as PC's except for that they are stolen much easier, more often, get in the range of different people more easily and that customs have the right to search them if you travel abroad. Mobile workers are out of the office so often; they have to use laptop computers, personal digital assistants (PDAs), and portable memory devices to exchange and transport business-critical data. In many cases, the security of this data hinges on the physical safety of the devices. Simply put, when mobile devices are lost, so is the data.

Maintaining privacy in our personal communications is something everyone wants to achieve. Encryption is a means to achieve that privacy.

There are many benefits we can get after encrypting our data and information:

- Encryption can provide a means of securing information. As more and more information is stored on computers, transmitted or communicated via computers, the need to ensure that this information is invulnerable to snooping and/or tampering becomes more relevant. In the storage of personal information (i.e. medical records, tax records, credit history, employment history, etc.) we want, need or expect privacy.
- Encryption is seen by many people as a necessary step for commerce on the internet to succeed. Without confidence that net transactions are secure, people are unwilling to trust a site. Encryption can give consumers the confidence they need to do business on the internet.
- Encryption can also provide a means of "message authentication". This scheme proves that the sender was the true originator of the message, and that the message has not been subsequently altered by anyone else. This prevents forgery of that signed message, and prevents the sender from denying the sending of that message.
- E-mail is certainly not secure. While you may believe that the use of a password makes your business private, you should be aware that sending information without encryption makes your message totally open to interception by anyone along the way.
- But there are many common situations, where users have a legitimate need for security both to protect that information such as consumers placing orders with credit cards via the Internet, journalists protecting their sources, therapists protecting client files, businesses communicating trade secrets to foreign branches, ATM transactions, political dissenters, or whistle-blowers -- all are examples of where and why encryption may be needed for e-mail or data files, and why it might be necessary to create a secure environment through its use.

## 3. NEED OF DATA COMPRESSION

As we know, the amount of information stored, transmitted, and handled by computers has been growing exponentially over the last decades. Two recent developments have particularly contributed to this effect. One development is the breakthrough of multi-media systems. The time when computers handled only numbers and text has gone and has been replaced by an new era of sound, images, movies and virtual reality. Another development is the increased availability of the Internet, which has made this information available to a large body of users. The performance of CPU's, disks, and transmission channels has grown tremendously. With the help of compression techniques, we can save storage, cpu-time and transmission time. Most of the information we use is highly correlated. In other works, it contains redundancy. Thus it seems possible to use compression without losing information. The major requirement from the compression techniques is that one can quickly switch between the original and compressed data. The other benefits which can be obtained by data compression are:

- At any given time, the ability of the Internet to transfer data is fixed.
- Thus, if data can effectively be compressed wherever possible, significant improvements of data throughput can be achieved.
- In some instances, file sizes can be reduced by up to 60-70 %.
- At the same time, many systems cannot accommodate purely binary data, so encoding schemes are also employed which reduce data compression effectiveness.
- Many files can be combined into one compressed document making sending easier, provided combined file size is not huge.

## 4. INTEGRATION OF DATA COMPRESSION AND CRYPTOGRAPHY: ANOTHER WAY TO INCREASE THE INFORMATION SECURITY

In this progressing world of computers, there is a large need of the security techniques for the data and to be able to store and transmit the huge amounts of data in a secure way. So, it can be achieved if we integrate the two different approaches discussed above in this paper to achieve compressed and secured (encrypted) data. Data compression removes redundant character strings in a file. This means that the compressed file has a more uniform distribution of characters. In addition to providing shorter plaintext and cipher text, which reduces the amount of time needed to encrypt, decrypt and transmit a file, the reduced redundancy in the plaintext can potentially hinder certain cryptanalytic attacks.

[3]The main purpose of data compression is to reduce the memory space or transmission time, while that of cryptography is to keep the security of the data. So far, these two technologies have been studied separately. This is because any data can be compressed if necessary, and then encrypted. The problem is that, with the rapid progress in computing technology, the encrypted data will not be secure any more after a few years. Once the data are decrypted, all secret will be leaked. Another problem is related to large size of data especially multimedia data. The revolution of multimedia and hyper media has been a driving force behind fast and secured data transmission techniques. In general, video data takes more time for encryption, because of its large size. Since the size of video data is huge in volume, it needs to be compressed and encrypted to avoid security threats and delay. Compression algorithms reduce the redundancy in data representation to decrease the storage required for that data. Data compression offers an attractive approach to reducing communication costs by using available bandwidth effectively. Over the last decade there has been an unprecedented explosion in the amount of digital data transmitted via the Internet, representing text, images, video, sound, computer programs, etc. With this trend expected to continue, it makes sense to pursue research on developing algorithms that can most effectively use available network bandwidth by maximally compressing data. It is also important to consider the security aspects of the data being transmitted while compressing it, as most of the text data transmitted over the Internet is very much vulnerable to a large number of attacks.

To solve this problem, in this paper we can think of a new way to increase the information security through integration of data compression and cryptography. Thus, even if the encrypted data are deciphered by some malicious persons, it is hard for them to reconstruct the original data without the transformation function, which is encrypted and kept by the user as a secret key. To encrypt the compressed data and the transformation function, we can use any existing techniques, as long as they are relatively secure.

## 5. ENCRYPTION BEFORE COMPRESSION OR COMPRESSION BEFORE ENCRYPTION

### 5.1Compression before Encryption

It has long been appreciated that there are advantages to eliminating regularities in the plaintext before encrypting.Compression should be done first before encryption because of the following reasons:

- **Compressing it last won't reduce the file size much.** Good encryption should make any input data (especially redundant data) appear random. But compression works by removing redundancy, and doesn't work well on random data.
- **Compressing it should decrease the effectiveness of some attacks.** Compression works by reducing the redundancy in the data. A common cryptanalysis method is frequency analysis, which relies on finding repeated data. Compressing it should reduce its effectiveness.
- **Brute force attacks will take longer.** Brute force attacks work by trying various keys and decrypting the data and checking if the output data makes any sense. By compressing it first, an attacker must decrypt the data and then decompress it before seeing if the output data makes any sense. This takes much longer, and if an attacker doesn't know you're compressing the data at all, they might never break the encryption.
- **The opponents get less cyphertext to analyze**. The less data the enemy (intruders) has to analyze, the fewer clues the have about the internal state of your cipher, and thus it's key.
- **What they do get has a corresponding plaintext with fewer redundancies and regularities.** It hinders cryptanalytic attacks.

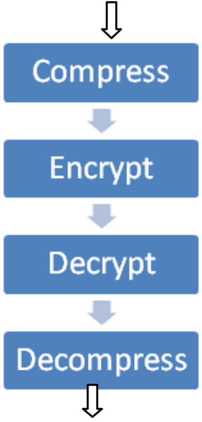

Fig. Compression before Encryption

By contrast, compressing a file after encryption is inefficient. The cipher text produced by a good encryption algorithm should have an almost statistically uniform distribution of characters. As a consequence, a compression algorithm should be unable to find redundant patterns in such text and there will be little, if any, data compression. In fact, if a data compression algorithm is able to significantly compress encrypted text, then this indicates a high level of redundancy in the cipher text which, in turn, is evidence of poor encryption.

### 5.2 Encryption before Compression

Encryption can also be done before compression in some situations such as when it is desired to transmit redundant data over an insecure and bandwidth-constrained channel, it is customary to first compress the data and then encrypt it.[1]Traditionally in communication systems, data from a source is first compressed and then encrypted before it is transmitted over a channel to the receiver. While in many cases this approach is benefitting, there exist scenarios where there is a need to reverse the order in which data encryption and compression are performed. Consider for instance a network of low cost sensor nodes that transmit sensitive information over the internet to a recipient. The sensor nodes need to encrypt data to hide it from potential eavesdroppers, but they do not necessarily want to compress it as that would require additional hardware and thus higher implementation cost. On the other hand, the network operator that is responsible for transferring the data to the recipient wants to compress the data to maximize the utilization of its resources. It is important to note that the network operator is not trusted and hence does not have access to the key used for encryption and decryption of data. If it had the key, it could simply decrypt data, compress and encrypt again.

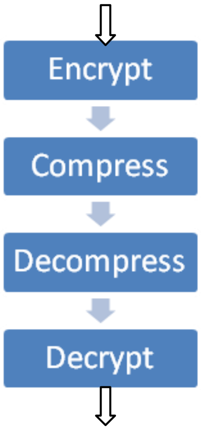

Fig. Encryption before Compression

## 6. CONCLUSION

In this paper, it has been discussed that by combining two different but related approaches-compressions and encryption, more information security can be achieved and it becomes efficient to transmit data and to store data more efficiently. But In this case one more problem of determining the order of these two approaches i.e, whether encryption should be applied before compression or compression should be applied before encryption. In 70% cases, it is more efficient to apply compression before encryption but in some particular situations and for special purposes, encryption can also be applied before compression.

## 7. REFERENCES

[1] Demijan Klinc_, Carmit Hazayy, Ashish Jagmohan, Hugo Krawczyk and Tal Rabin ,"On Compression of Data Encrypted with Block Ciphers", http://users.ece.gatech.edu/~demi/docs/dcc09paper.pdf

[2] W. Mao, Modern Cryptography: Theory and Practice. Prentice Hall, 2003.

[3] Chuanfeng Lv, Qiangfu Zhao,"Integration of Data Compression and Cryptography: Another Way to Increase the Information Security". AINA Workshops (2) 2007: 543-547

[4] Sien, O.B, Samsudin, A., Budiarto, R." A new image-database encryption based on a hybrid approach of data-at-rest and data-in-motion encryption protocol",Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference

[5] M. Johnson, D. Wagner, and K. Ramchandran, \On compressing encrypted data without the encryption key," in Proc. of the Theory of Crypto. Conf., Cambridge, MA, Feb.2004.

[6] William Stallings. *Advanced Encryption Standard*. Chapter *5* in Cryptography and Network Security: Principles and Practices. International Third Edition. Prentice Hall: United States of America. (2003)

[7] Application Security, Inc. *Encryption of Data at Rest - Database Encryption*. White Paper. (2002)

[8] Lala Krikor, Sami Baba, Thawar Arif, Zyad Shaaban," Image Encryption Using DCT and Stream Cipher", European Journal of Scientific Research, ISSN 1450-216X Vol.32 No.1 (2009), pp.47-57