Asian Journal of Information Technology 5 (12): 1306-1309, 2006

A Trigonometric Approach for Simultaneous Compression and Encryption

[1]A. Anasuya Threse Innocent, [2]V. Kavitha and [3]K.S. Easwarakumar
[1]School of Computing Sciences, Vellore Institute of Technology-Deemed University, Vellore, Tamilnadu, India
[2]Department of IT, Noorul Islam College of Engineering, Kumaracoil, Tamilnadu, India
[3]Department of CSE, Anna University, Chennai, Tamilnadu, India

Abstract: The increasing network transaction of sensitive information needs an efficient and secured way of information transmission. In today's era of Information Technology, there is a strong need for transmitting heavy volumes of information within the available bandwidth. This can be achieved by means of compression techniques thereby the efficiency of data transmission is increased. Data security has to be incorporated for secure transaction of sensitive data. Cryptographic technique ensures privacy by keeping the information hidden from anyone to whom it is not intended. The present scenario does encryption and compression as individual process to achieve secure and effective transmission. Pre-analysis proves that performing compression and then encryption is the best. To overcome the individual compression and encryption process and to minimize the time of execution, this trigonometry approach does simultaneous compression and encryption.

Key words: Cryptography, compression, encryption, trigonometry

INTRODUCTION

The growth of high-speed networks has explored means of new business, science, entertainment and social opportunities. This has been achieved through Client-Server Technology. Compression and Encryption places a vital role in client-server technology.

Effective transmission is done when the sensitive data is compressed and encrypted before transmission. Chung-E Wang[1] has proposed methods for simultaneous compression and encryption in Arithmetic Huffman coding algorithm. This study describes a new algorithm that intertwines both compression and encryption in itself. The basic idea behind, the trigonometry leads to the birth of the new algorithm that performs simultaneous compression and encryption.

Consider an asymmetrical triangle shown in Fig. 1,

Where,
a, b, c : sides of the triangle
A°, B°, C°: angles opposite to the sides a, b, c in degrees.

Block of data is assumed to be sides (*a* and *b*) of a triangle and the secret key is used as the angle C° between these two sides to find the third side, the ciphertext *c*. This angle C° is the secret key that has to be shared between source and destination through a secured channel.

Also for an asymmetrical triangle the below relation holds,

$$\frac{a}{\sin A^{\circ}} = \frac{b}{\sin B^{\circ}} = \frac{c}{\sin C^{\circ}}$$

In an asymmetrical triangle, if two sides *a* and *b* and the angle between them C° are known, it is possible to calculate the third side *c* as,

$$c = \sqrt{a^2 + b^2 - 2ab\cos C^{\circ}}$$

The angle A° is calculated using the formula $A^{\circ} = \sin^{-1}\left[\frac{a \sin C^{\circ}}{c}\right]$ and is sent along with the ciphertext to the receiver that helps to find the sides, *a* and *b* from the ciphertext.

Side *a* is calculated as $a = \frac{c \sin A^{\circ}}{\sin C^{\circ}}$. If two angles and one side is known it is possible to find the other two sides. Knowing angles A° and C° the third angle B° can be calculated as $B^{\circ} = 180^{\circ} - (A^{\circ} + C^{\circ})$ and the other side *b* can be calculated from the ciphertext as, $b = \frac{c \sin B^{\circ}}{\sin C^{\circ}}$.

Corresponding Author: A. Anasuya Threse Innocent, School of Computing Sciences, Vellore Institute of Technology-Deemed University, Vellore, Tamilnadu, India

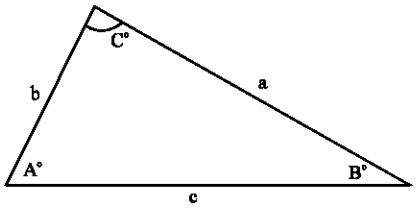

Fig. 1: Asymmetrical triangle that does simultaneous compression and encryption

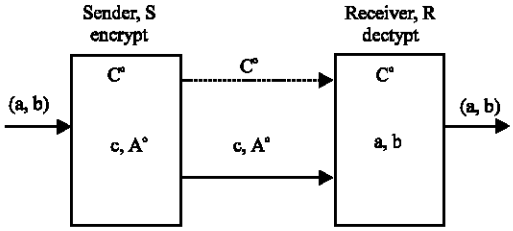

Fig. 2: Architecture of Simultaneous Compression and Encryption

The architecture of Simultaneous Compression and Encryption is shown in Fig. 2. The angle C° is conveyed to the other end by secured channel.

Key generation: Fix C° in the range 0° to 180° (both 0° and 180° exclusive) Private key : C°.

Encryption: Plaintext : (a, b).

Consider a and b as its two sides of a triangle with C° as its angle between them, Calculate its third side $c = \sqrt{a^2 + b^2 - 2ab\cos C^{\circ}}$ and angle between b and c as $A^{\circ} = \sin^{-1}\left[\frac{a \sin C^{\circ}}{c}\right]$ or angle between a and c as $B^{\circ} = \sin^{-1}\left[\frac{b \sin C^{\circ}}{c}\right]$.

Ciphertext : (c, A°) or (c, B°).

Decryption: Ciphertext: (c, A°) or (c, B°).

If A° is calculated and sent then, calculate a as, $a = \frac{c \sin A^{\circ}}{\sin C^{\circ}}$ or $b = \frac{c \sin B^{\circ}}{\sin C^{\circ}}$.

Calculate B° as,

$B^{\circ} = 180^{\circ} - (A^{\circ} + C^{\circ})$ or $A^{\circ} = 180^{\circ} - (B^{\circ} + C^{\circ})$.

If A° is sent then calculate b as, $b = \frac{c \sin B^{\circ}}{\sin C^{\circ}}$ or $a = \frac{c \sin A^{\circ}}{\sin C^{\circ}}$.

Deciphered plain text: (a, b)

Analysis: The complexity of this algorithm to the Brute Force attacker increases exponentially with the increase of n, where n is the number of decimal digits after the decimal point in the key. The complexity in terms of the length of the key is $O(10^n)$. The key space available for the sender to perform encryption is ($179 \times 10^n - 1$). But the key space available for the Brute Force attacker is ($180\text{-}A\text{-}1$) $\times 10^n - 1$. To increase the security, the value of n has to be increased. To ensure the security, the value of a and b should be chosen to real.

Analysis was done on Pentium IV with compression algorithms such as Huffman, LZSS, LZW and Fractal [2-5] and encryption algorithms such as AES and DES[6,7] for text and image. The analysis proved that the execution time consumption of the proposed algorithm is very less when compared to that of the existing compression and encryption algorithms when applied as separate process. The compression ratio is also fair when compared to that of the existing algorithms. The algorithm thus does an efficient compression and encryption that is required for network transaction.

The time analysis results for performing encryption after compression and with new algorithm for redundant input data is shown in Fig. 3. The Fig. 4 shows the results for non-redundant input data.

The above analysis clearly states that the new algorithm consumes minimal execution or processing time for compressing and encrypting the text data.

Figure 5 depicts the analysis of existing and new algorithm for image data in the transmitter side.

From these analyses, it is clear that, the new algorithm can minimize the execution time to a great extent in the transmitter side than the existing ones. Similarly, the analysis for the time factor is carried in the receiver side. The time required to retrieve original data is even much less compared to that one in transmitter side. The results of the analysis are shown in the following figures: Fig. 6. and 7. Figure 6 shows the results for the original redundant text input data and the Fig. 7 depicts that for non-redundant data.

When lossy compression is applied over images and then encrypted before transmission, it is not possible to get back the original image. Hence only lossless compression algorithm is recommended for sensitive images. The new algorithm overcomes this problem. This trigonometric approach for simultaneous compression and encryption works well for images.

The new algorithm achieves data compression almost up to 37.5 %. From the related work for every 8 bytes of input data, the ciphertext obtained is 5 bytes: 4 bytes are occupied by cipher and the last one

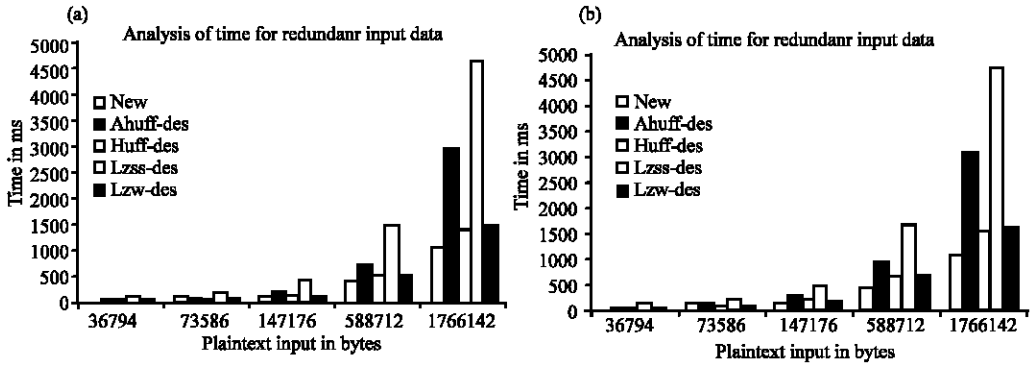

Fig. 3: Time analysis of new and existing algorithms for redundant text data

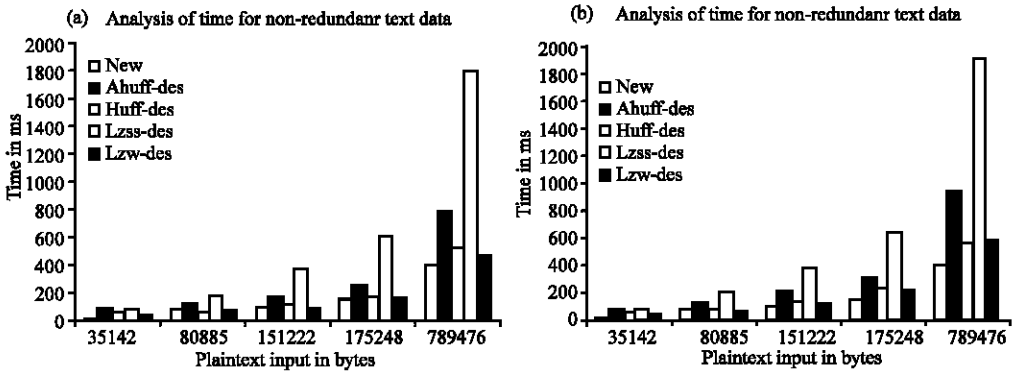

Fig. 4: Time analysis of new and existing algorithms for non-redundant text data

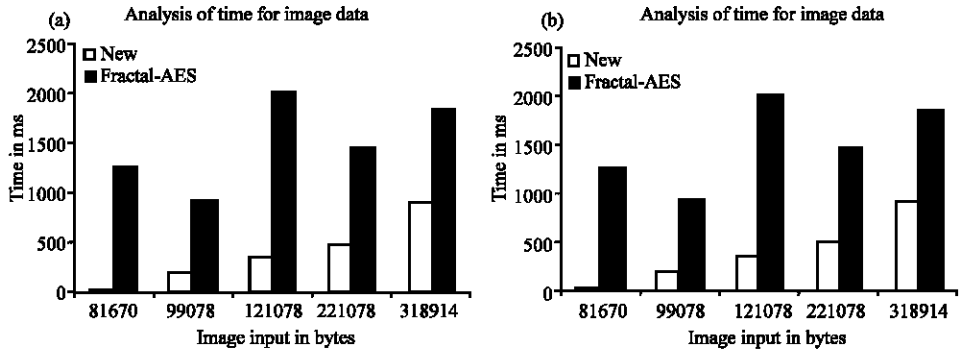

Fig. 5: Time analysis of New and existing algorithms for image data

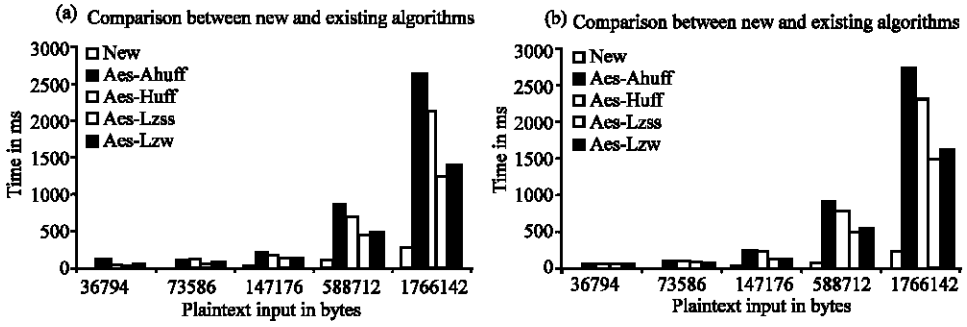

Fig. 6: Time analysis of New and existing algorithms in the receiver side for redundant text data

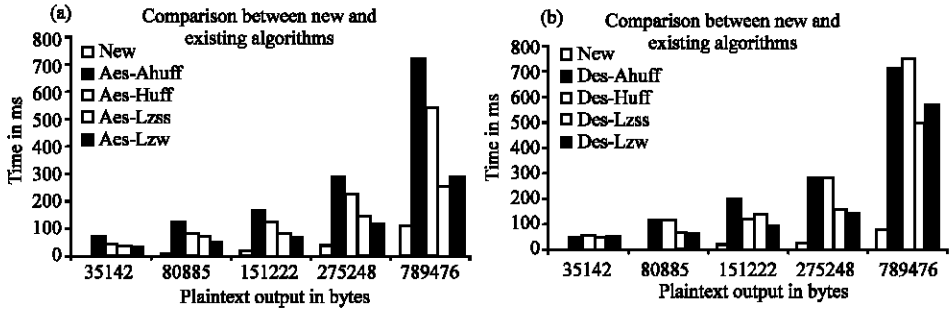

Fig. 7: Time analysis of New and existing algorithms in the receiver side for non-redundant text data

byte to hold the additional information. So theoretically size is being reduced to {[(8-(4+1))/8] × 100} i.e., 37.5 %.

The new algorithm takes less execution time than the existing compression and encryption algorithms when applied as separate process. The algorithm does compression and encryption simultaneously and compression ratio is also fair when compared to that of existing algorithm. The new algorithm thus does an efficient compression and encryption and keeps in pace with IT Industry that is required for network transaction.

REFERENCES

1. Chung-E Wang, 2003. Department of Computer Science, California State University, Sacramento, Simultaneous Data Compression and Encryption, Proceedings of the International Conference on Security and Management, SAM '03, Las Vegas, Nevada, USA, CSREA Press 2003, ISBN 1-932415-17-3, pp: 558-563.
2. Text Compression, 2003. A chapter from Theory in Programming Practice, unpublished Lecture Notes, by Jayadev Misra, University of Texas at Austin, July 1, 2003 available at http://www.cs.utexas.edu/users/misra/ClassNotes.dir/337.pdf
3. Atul Kahate, 2003. Cryptography and Network Security, Tata McGraw-Hill.
4. David Solomon, 1997. Data Compression: The Complete Reference, Springer.
5. Mark Nelson and J.L. Gailly, 1996. The Data Compression Book, 2nd (Edn.), M and T Books, New York, NY.
6. AES, Proposal : Rijndael, a technical paper by John Daemen and Vincent Rijmen., http://csrc.nist.gov/CryptoToolkit/aes/rijndael/
7. Barnsley, M.F. and L.P. Hurd, 1993. Fractal Image Compression, AK Peters Ltd., Wellesley, Massachustts.