# A Discrete Wavelet Transform based Cryptographic algorithm

**Debayan Goswami[1], Naushad Rahman[1], Jayanta Biswas[1], Anshu Koul[1], Rigya Lama Tamang[1], Dr. A. K. Bhattacharjee[2]**

[1] Undergraduate Students, Department of ECE, National Institute of Technology Durgapur India
[2] Professor, Department of ECE, National Institute of Technology Durgapur India

**Abstract**
This paper presents a new and efficient algorithm for cryptographic purpose that considers the representation of the cipher text by using the Discrete Wavelet Transform to find the wavelet decomposition vector containing the approximation and the detail coefficients. The decryption is done by extracting the encrypted data from the wavelet decomposition vector using the inverse Discrete Wavelet Transform algorithm. The encrypted message consists of just the wavelet decomposition vector. The key consists of the code number of the wavelet used and the bookkeeping vector.

***Key words:***
*Cryptography, Algorithm, Discrete Wavelet Transform, Wavelet Decomposition, Signal Processing.*

## 1. Introduction

Cryptography is one of the most important tools that provide data and information security by hiding it. It is usually done through mathematical manipulation of the data with an incomprehensible format for unauthorized users.

In this paper, a cryptographic technique based on Discrete Wavelet Transform is presented.
Section 2 explains the process of encryption and decryption of the text read from the file specified in the code itself. Section 3 describes an implementation of the technique using the code. It takes a real file for the execution of the algorithm. Section 4 shows the results of the algorithm using different file types and shows a graphical study of the technique. Section 5 is an analytic discussion on the technique with the conclusion and future scopes of this work.

## 2. The Scheme

Wavelet transform has become a common tool for analyzing localized variations of power within a time series. By decomposing a time series into time-frequency space, one is able to determine both the dominant modes of variability and how those modes vary in time.
The Discrete Wavelet Transform (DWT) of a signal **x** is calculated by passing it through a series of filters. First the samples of the signal **x** is passed through a low pass filter with the impulse response **g ,** resulting in the convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \quad \#$$

The signal is also decomposed simultaneously using a high pass filter having impulse response **h,** the outputs giving the detail coefficients (from the high pass filter) and the approximation coefficients (from the low pass filter). It is important that the two filters are related to each other and they are known as the quadrature mirror filter. However since half the frequencies of the signal have now been removed, half the samples can be discarded according to the Nyquist rule. The filter outputs are then sub-sampled by two.

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n-k]$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n-k]$$

The decomposition has halved the time resolution since half of each filter output characterizes the signal. However each output has half the frequency band of the input so the frequency resolution has been doubled.

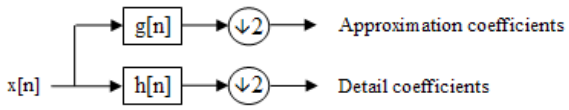



**Figure 1: Block diagram of the filter analysis.**

Manuscript received April 5, 2011
Manuscript revised April 20, 2011

This kind of decomposition can be repeated to further

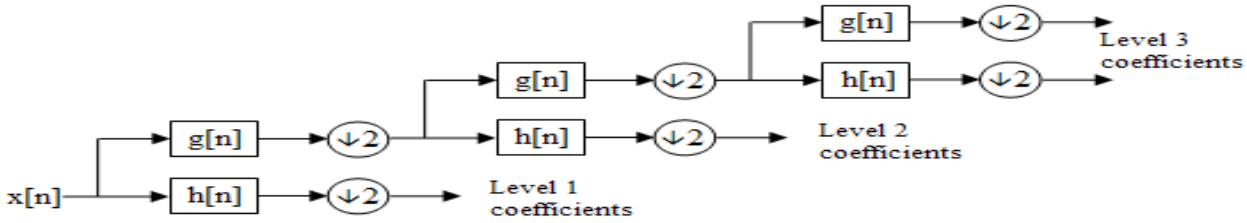



Figure 2: A 3 level filter bank.

increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree (as shown in figure 2) with the nodes representing a subspace with different time frequency localization. This tree is known as a filter bank.

The actual description of the encryption and the decryption of a text message used is presented in this section. Section 2.1 shows the ciphering techniques used for the text, section 2.2 represents the process of key generation, section 2.3 shows the deciphering technique of the cipher text.

## 2.1 Ciphering Technique

Step 1: To encrypt a text message, at first the text is stored in a string variable say ***message***.
Step 2: The length of the message is stored in a numeric variable ***length***.
Step 3: A one dimensional array ***num[]*** which contains the ASCII equivalent numbers of the constituent characters of the message is created.
Step 4: The ***num[]*** is then put to discrete wavelet transform to the maximum allowed level of decomposition (found out using the function "**wmaxlev**" in the Wavelet Toolbox in MATLAB) to get the ***wavelet decomposition vector C[]*** and the ***bookkeeping vector L[]*** as illustrated by figure 3.
Step 5: The wavelet decomposition vector ***C[] can be used as the encrypted array*** to be sent to the destination. The process of key generation is explained in the next section.

## 2.2 Key Generation

After the encrypted array has been generated there is a stratagem needed to reverse the steps and get back the original text. Using only the wavelet decomposition vector it is not possible to get back the original text unless we know the bookkeeping vector and the wavelet used for the decomposition. So the ***key K[]***, is prepared with the code for the wavelet used (See Table 2) as the first element, and the bookkeeping vector concatenated to it. The structure of the key is shown in the figure 1.

Table 1: Key K[] Structure

| | |
|---|---|
| Code number for the wavelet used | Bookkeeping vector, L |

Table 2: Table showing the codes for the different Wavelets.

| Wavelet Name | Code |
|---|---|
| Daubechies2 | 2 |
| Daubechies3 | 3 |
| Daubechies4 | 4 |
| Daubechies5 | 5 |
| Daubechies6 | 6 |
| Daubechies7 | 7 |
| Daubechies8 | 8 |
| Daubechies9 | 9 |
| Daubechies10 | 10 |

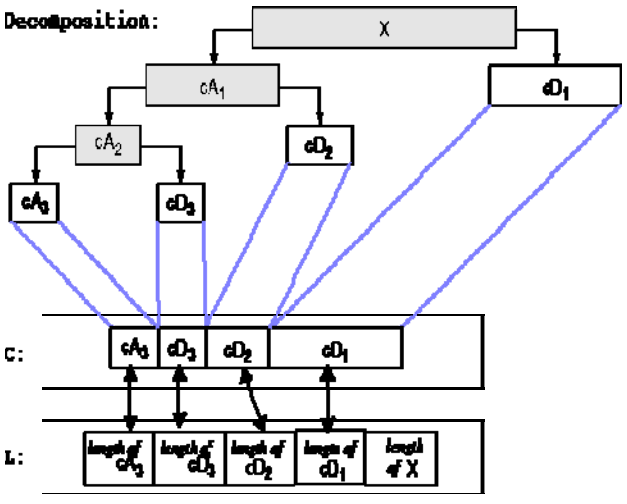



Figure 3: Wavelet decomposition vector C and bookkeeping vector L depiction

### 2.3 Deciphering Technique

Step 1: After getting the encrypted message the receiver will use the key to get back the original message. The first element of the ***key K[]*** is extracted to get the code of the wavelet used in encryption. The proper wavelet is necessary to do the wavelet reconstruction. The rest of the key is taken as the bookkeeping vector ***L[].***.

Step 2: From the encrypted message i.e. the ***C[]*** vector received, and the bookkeeping vector ***L[]*** extracted from the key, the ***num1[]*** vector is retrieved using the wavelet reconstruction techniques and the proper wavelet name.

Step 2: The array thus formed, ***num1[]*** consists of the ASCII value of the characters in order in the original text message. Now the receiver can get back the original message by converting the ASCII values to their corresponding ASCII characters**.**

## 3. Implementation

This section illustrates the encryption and the decryption of a given text in a tabular form showing the concerned matrices. The wavelet used for the decomposition was the **"Daubechies 4"** wavelet.

### 3.1 Process of encryption

Let us consider the plain text "**I will cherish the memories of my college life!**".
The bookkeeping vector ***L[]*** serves as the part of the key.

### 3.2 Process of decryption

After receiving the cipher text in the form of the wavelet decomposition vector ***C[]***, the name of the wavelet used and the key, the decryption process is carried out.
The encryption and the decryption process is shown in the Table 4.
The decrypted message received at the end is the same as the original text.

## 4. Results

The algorithm described works perfectly for all the text formats viz. - **.txt, .doc, .rtf, .bmp, .sys, .exe** files.

A comparative study of the execution times for texts varying in length during encryption and decryption is shown in the Table 3.

A graph showing the relationship between the encrypted time and the text length is demonstrated in the figure 4.

Table 3: Encryption and decryption times

| No. of letters in the text including blank spaces and special characters | Encryption time in seconds | Decryption time in seconds |
|---|---|---|
| **1** | 0.234553 | 0.047926 |
| **8** | 0.22786 | 0.048146 |
| **16** | 0.227138 | 0.047492 |
| **32** | 0.231861 | 0.047455 |
| **40** | 0.221108 | 0.047306 |
| **48** | 0.223747 | 0.046772 |
| **56** | 0.220276 | 0.048992 |
| **64** | 0.229059 | 0.046859 |
| **72** | 0.223548 | 0.049046 |
| **80** | 0.229516 | 0.048016 |
| **88** | 0.223163 | 0.047131 |
| **96** | 0.23021 | 0.047604 |
| **100** | 0.225688 | 0.048374 |
| **500** | 0.231895 | 0.047176 |
| **1000** | 0.234043 | 0.047323 |

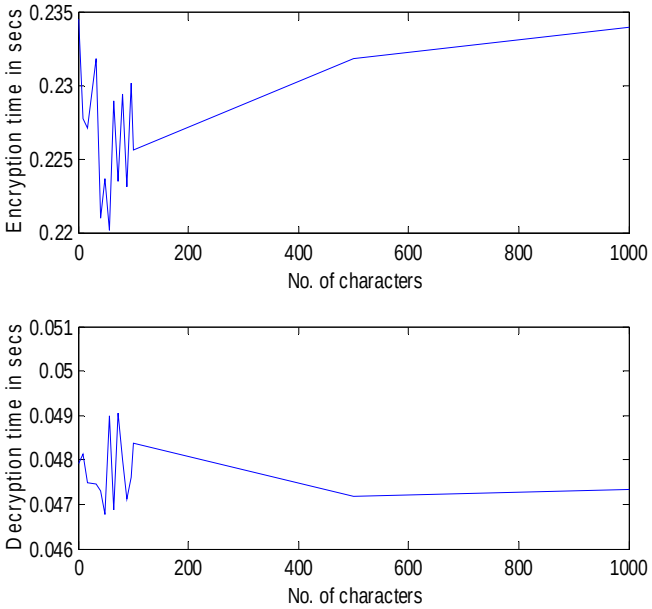



Figure 4: Relationship between execution times and text lengths

Table 4: Encryption and Decryption process

| Before Encryption num[] | Encrypted message C[] | Key K[] | After Decryption num1[] |
|---|---|---|---|
| 73 | 188.3808 | 4 | 73 |
| 32 | 129.4062 | 17 | 32 |
| 119 | 234.3993 | 17 | 119 |
| 105 | 149.1718 | 27 | 105 |
| 108 | 176.1723 | 47 | 108 |
| 108 | 164.3086 | | 108 |
| 32 | 215.9573 | | 32 |
| 99 | 188.9789 | | 99 |
| 104 | 164.1218 | | 104 |
| 101 | 217.8136 | | 101 |
| 114 | 192.8504 | | 114 |
| 105 | 165.8727 | | 105 |
| 115 | 175.6822 | | 115 |
| 104 | 205.0096 | | 104 |
| 32 | 189.1378 | | 32 |
| 116 | 142.6172 | | 116 |
| 104 | 121.3030 | | 104 |
| 101 | -3.858749 | | 101 |
| 32 | -7.35 | | 32 |
| 109 | 53.03642 | | 109 |
| 101 | -0.596609 | | 101 |
| 109 | 23.66805 | | 109 |
| 111 | 43.35265 | | 111 |
| 114 | 8.599247 | | 114 |
| 105 | -8.71093 | | 105 |
| 101 | -9.090203 | | 101 |
| 115 | 46.28193 | | 115 |
| 32 | 27.16337 | | 32 |
| 111 | 2.205919 | | 111 |
| 102 | 52.74635 | | 102 |
| 32 | -16.23308 | | 32 |
| 109 | 5.903183 | | 109 |
| 121 | -56.13243 | | 121 |
| 32 | 5.124079 | | 32 |
| 99 | 18.66153 | | 99 |
| 111 | 48.260732 | | 111 |
| 108 | -11.75297 | | 108 |
| 108 | -49.60677 | | 108 |
| 101 | 6.5400294 | | 101 |
| 103 | 5.481055 | | 103 |
| 101 | 7.3189174 | | 101 |
| 32 | -53.94721 | | 32 |
| 108 | -4.664206 | | 108 |
| 105 | -50.82093 | | 105 |
| 102 | -4.628689 | | 102 |
| 101 | -1.557061 | | 101 |
| 33 | -3.657466 | | 33 |
| | 28.268788 | | |
| | 49.647 | | |
| | -64.59417 | | |
| | 29.692847 | | |
| | 38.466992 | | |
| | -14.68198 | | |
| | -1.207162 | | |
| | 15.851806 | | |
| | 47.682894 | | |
| | -14.97573 | | |
| | -29.7389 | | |
| | 44.397538 | | |
| | -14.21032 | | |
| | -50.70881 | | |

## 5. Analysis and Conclusion

The advantage of this type of algorithm is that the encryption time remains almost constant even for long strings of messages and the decryption time is very small in the millisecond range. There is no distortion of data seen as the text string keeps on getting longer.

The method described here could be useful in a lot of commercial applications whereby large databases can be rendered illegible to unauthorized users. This area of using signal processing in cryptography has larger scopes where other mathematical techniques of signal analysis can be used to encrypt and decrypt messages.

## References

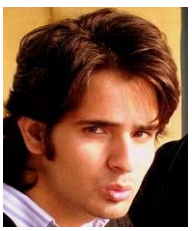
[1] I. Daubechies, Ten lectures on wavelets, (SIAM, Philadelphia,1992).
[2] S. Mallat, A Wavelet Tour of Signal Processing, (AcademicPress, 1999).
[3] Sachin P. Nanavati, Prasanta K. Panigrahi, “Wavelet Transform- A new mathematical microscope”, (Resonance, March 2004)
[4] Jatan K. Modi, Sachin P. Nanavati, Amit S. Phadke, Prasanta K. Panigrahi, “Wavelet Transforms- Application to Data Analysis – 1”,(Resonance, November 2004)

[5] Dutta S. and Mandal J. K., "A Space-Efficient Universal Encoder for Secured Transmission", International Conference on Modelling and Simulation (MS) 2000 – Egypt, Cairo, April 11-14, 2000.
[6] J. K. Mandal, S. Dutta, "A 256-bit recursive pair parity encoder for encryption", Advances D-2004, Vol. 9, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (ASME, France), pp-1-14.
[7] William Stallings, Cryptography and Network security: Principles and practice (Second Edition), Pearson Education Asia, Sixth Indian Reprint 2002.
[8] Atul Kahate (Manager, i-flex solution limited, Pune, India), Cryptography and Network security, Tata McGraw-Hill Publishing Company Limited.

**Debayan Goswami** is a final year B.Tech. student in the department of Electronics and Communication Engineering, National Institute of Technology(NIT) Durgapur, West Bengal, India. His current areas of interest are Signal Processing, Data Analysis and Cryptography.

**Naushad Rahman** is a final year B.Tech. student in the department of Electronics and Communication Engineering, National Institute of Technology(NIT) Durgapur, West Bengal, India. His current areas of interest are Image Processing and Robotics.

**Jayanta Biswas** is a final year B.Tech. student in the department of Electronics and Communication Engineering, National Institute of Technology(NIT) Durgapur, West Bengal, India.



**Anshu Koul** is a final year B.Tech. student in the department of Electronics and Communication Engineering, National Institute of Technology(NIT) Durgapur, West Bengal, India**.**

**Rigya Lama Tamang** is a final year B.Tech. student in the department of Electronics and Communication Engineering, National Institute of Technology(NIT) Durgapur, West Bengal, India.

**Prof. (Dr) A. K. Bhattacharjee** did his Ph.D. in Engineering from Jadavpur University, Kolkata, India in 1989. Presently he is associated with Department of ECE in NIT, Durgapur, INDIA. The senior academicians, having been involved in teaching and research for last 20 years, his current areas of research are Microstrip, Antenna and Cryptography.