# Digital Fortress
# [New Standard for Encryption]

*Manojkumar Parmar*

Institute of Technology,
Nirma University, Ahmedabad
parmarmanojkumar@gmail.com

## Abstract

Digital Fortress is proposed cryptosystem to fulfill the requirement of modern communication system which demands low computation power, faster execution & immunity towards attack. Proposed algorithm is built on the base of Vernam's One Time Pad with the help of Rotating Key Function, Permuted XORing, etc. This algorithm has blend of non-linearity & linearity. Rotating Key Function is based on modulo operator along with algebraic equation to generate the randomize key having the length same as data from finite small two user keys. Permuted XORer performs the operation on Plaintext & randomize Key to generate Ciphertext. It employs first Rotating Permutation then Modified XORing & at the end Rotating Odd Shifter, operation performed in this suggested by name it self. This algorithm employs all the function in primitive format for analysis purpose. This encrypted data is dump in to image with the help of Bit Distribution Function so data is hided so it can't be recognize as encrypted data & give algorithm extra advantage that even encrypted data can't be extracted. All in all, it has ability to resist all kind of existing attack & make it more immune the attack. To deploy this algorithm in commercial field certain recursivity is included at cost of little computing power as employed in most of the encryption standard. In nutshell, this algorithm has ability to open new era in field of cryptosystems having perfect secrecy with finite length of key which was day-dream in past but today it exist with name of Digital Fortress.

## 1. Need of Encryption

Today's world becomes the world of information by transferring enormous amount of data over the communication channel which may be subjected to interception in unauthorized manner to get advantage of it [1]. Introduction of wireless communication channel increases the threat of interception because data is available to everybody in the path of that channel. Privacy is another major issue with which any system has to deal to get QOS (Quality of Service) in order to make communication immune towards intruders [2]. To avoid this type of conflicts in communication of secure data, encryption of data is major concern in system with the utmost priority. Symbol languages developed to communicate in early development of civilizations were the first noticed encryption so certain community can utilize the information where other didn't understand this like Egyptian Chants & etc. History of encryption is full of invention due to need of communication.

## 2. Basics of Encryption

Symbol languages are kind of substitution encryption system in which one can put certain symbol for certain character & by reversing the process original information can be extracted. In world of encryption, original data is referred as Plaintext, encrypted version of Plaintext is known as ciphertext & encryption system employs a Key for conversion [3].

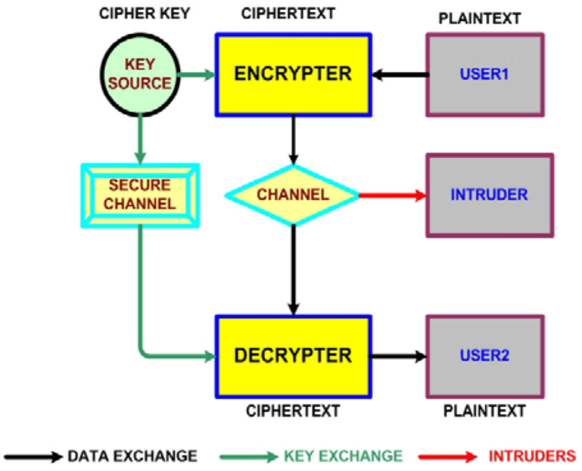

*Figure 1:* Block diagram of encryption system

From technical point of view, first encryption system is introduced by Caesar which is shifted encryption system & one of the basic systems [3]. In 1917, Gilbert Vernam introduced OTP (One Time Pad) algorithm which was based on xor padding & having the property of perfect secrecy. [4] [5]. After this OTP, Permutation encryption, Substitution encryption, shifting encryption are the basic systems [6]. Some changes in this basic system lead to advance version of encryption systems like Affine encryption, Vigenère encryption, Hill encryption, Stream encryption, Substitution Permutation encryption, etc [7]. These all encryption algorithm employs three basic operation named Permutation, Substitution, Xor (modulo shifting) in various manner to get maximum secrecy for encryption. These all system is symmetric systems which can be invertible to get Plaintext from same key [3]. Asymmetric system needs separate Key for encryption &

separate Key for decryption like in public Key encryption standard. Major focus of author is on data encryption with peer to peer in terms of Key in which decryption is only possible for the person who knows the Key for encryption. Cryptanalysis is done on cipher to get Plaintext without knowing the Key for encryption & employed for attacking on encryption systems [8].

## 3. Previous work & Criticism

Focuses of author are on generation of new encryption standard which posses the property of perfect secrecy & having infinite unicity distances with small Key. Vernam's OTP possess property of perfect secrecy & infinite unicity distances but length of Key is having the size same of Plaintext [5]. This is the major issue in OTP because prior to encryption, Key has to send over secure channel for decryption purpose & practically it was fiasco. Some other issue related to this are like if one Key is used for many encryption then algorithm was subject to failure because intruder can easily find out the difference between two messages & by analyzing them in linear fashion, it is easy to get Key[8]. Therefore this algorithm is practically not implemented for commercial purpose. Author' main aim is to enhance this OTP with some unique function such that length of Key is finite & small in nature.

Random Rotating XOR algorithm is another algorithm which deals with normal XORing of Plaintext & cipher Key [9]. In this, Key is rotated randomly but randomizing the Key is very difficult job as such this algorithm is concern. This algorithm deals with data in form of block & randomizes the Key of large block for XORing purpose. This algorithm is viable solution for small LAN because it needs KDC (Key Distribution Center) which is not feasible for large networks [9]. Therefore some intelligence is requiring inside the algorithm to avoid the problem of randomization of Key. Author' main aim is to provide such intelligence so that the implementation of randomization is inherent & easy.

## 4. Introduction to Digital Fortress

Digital Fortress is proposed algorithm which is modified & enhanced version of Vernam's OTP. In this algorithm, author introduces certain unique function to enhance the performance of Vernam's OTP algorithm with small & finite length key. To modify original algorithm some new type of operation introduce in existing function to get the best performance. This algorithm is divided in to four functions named Segmenter, Unique Shifter (U.S.), Permuted XORer (P.X.) and Bit Distributor (B.D.).

This algorithm is classified as symmetric key algorithm but having the blend of linearity as well as non-linearity. It employs two key encryption structure instead of single key in which one is alphanumeric key & second one is only numeric key.

In this algorithm, first data segment each of 8 byte is generated by Segmenter then manipulation on key takes place by U.S. which is actually Rotating Key Function (R.K.F.) to generate same length of key as of data in same segment size. P.X. performs operation on each segment with the help of some functions like Rotating Permutation (R.P.), Modified XORing (M.X.) and Rotating Odd Shifter (R.O.S). P.X. gives the encrypted data as output which can be dump in to image by using B.D. so data get hided in to image & make algorithm immune towards all attack. Proposed algorithm is viable solution for all type of networks & it is subject to some modification for different kind of networks.

## 5. Algorithm for Encryption

This algorithm is divided in to four main functions along with certain sub functions included in it. To understand this algorithm in proper way all the function must be known.

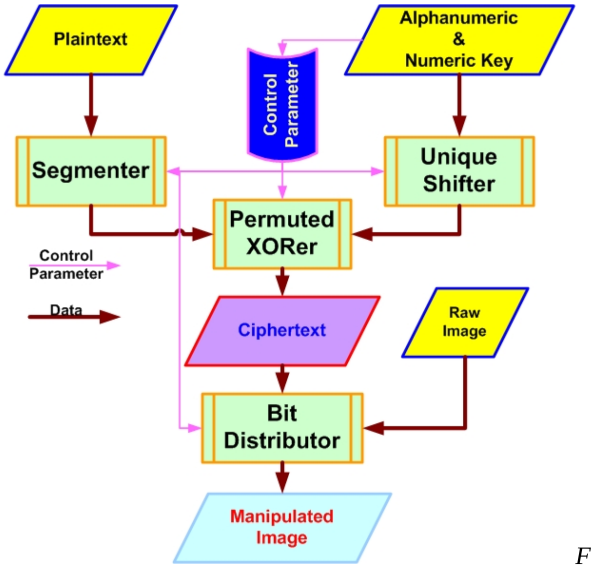

*Figure 2*: Block diagram of Digital Fortress Algorithm

### 5.1 Segmenter

Segmenter take the input as single dimensional array of message where each element in array is character in message & each element represents by a byte in array. Segmenter divides this array in to group of N byte format to generate blocks which can be processed further by functions. Segmenter decide the size of block & in general for basic implementation, the value of N is 8 and for advanced processing N possess the value of $2^n$ where n = 4,5,6,7,…,10. Increment in n will require very high computation power but security is very high. So it is trade off between block size, computation power & security. Author chooses the value of N as 8 for implementation purpose.

### 5.2 Unique Shifter

Unique Shifter takes input as output of Segmenter in forms of block size of N. this is the most crucial function for this algorithm. It is basically a Rotating Key Function which is made-up of modulo & addition/subtraction operation.

This function manipulates the alphanumeric key with the help of numeric key to generate unique key to support the algorithm. Numeric key forms an algebraic equation by providing co-efficient for multiplier & power. First three numbers in numeric key is power co-efficient & they are strictly limited in the range of 0 to 3 and hence denote the degree of equation. Last three digits denote the co-efficient of multiplier for algebraic equation.

R.K.F. use algebraic equation generated by numeric key to produce shifting number to shift the alphanumeric key in bit format. Again this equation is valid for one block only i.e. for 8 byte only. For another block the multiplier co-efficient are changed with rotating this co-efficient with modulo operation.

For example, A,B,C are multiplier co-efficient for block X then for block $X^+$,

$A^+$ = mod (B*C, a); (where a is limiter for modulo operation to lower the computation power)

$B^+$ = A – mod (i*$X^+$, b); (where i is arbitrary value as control parameter & $X^+$ is block number & b is limiter)

$C^+$ = B – mod (j*$X^+$, c) (where j is arbitrary value as control parameter & $X^+$ is block number & c is limiter)

In this manner co-efficient are rotated with some manipulation so value of shifter for each block will be different & hence the key is shifted abruptly to possess the nature of uniqueness. Here alphanumeric key is shifted bitwise & different for the entire symbol range. In this manner unique combination of key is generated from finite length & small key of only 64 bit. This concept satisfies the theory of Perfect Secrecy along with Vernam's OTP [3] [2].

### 5.3 Permuted XORer

This function is made-up of three sub function in which one is primary function & two are secondary function which support the primary function. Here Modified XORing is main function along with Rotating Permutation & Rotating Odd Shifter as secondary functions.

Rotating Permutation takes the input as permutation matrix of 8 element sizes & then this matrix rotated for each block depend up on control parameters. For implementation purpose, author use the linear relation of simple linear shift in either direction only one place. This rotation is circular in nature & by doing so the permutation matrix of 64 elements is generated. If relation is non linear then permutation is also unique in nature for each block.

Rotating Odd Shifter is based on the database of finite set of odd numbers. It takes the input as location number for database & gives the output as odd number. Here the choosing of location number is based on certain relationship which is the part of control parameters. This relationship is either linear or non-linear in nature. Author utilizes simple relationship of one increment in location number for each time function called.

Modified XORing is special kind of XORing adopted from Data Encryption Standard (DES) with certain modification in it [10]. In this first block is taken & permutation is done with supplied permutation matrix from Rotating Permutation. After this data & key is simply XORed with each other to generate the intermediate encrypted data. This intermediate encrypted data is divided in to two equal half each of 4 byte named Right Half & Left Half. According to control parameter one half is chosen & then this is placed in either as right or left part. After this remaining part is taken & according to control parameter mirror image of it is generated & then this image is XORed with chosen part & placed it as remaining part. Combination of two parts is final encrypted version or ciphertext for plaintext.

### 5.4 Bit Distributor

Bit Distributor is optional part of this algorithm. This function takes the input as ciphertext & scrambles it in image according to Bit Distribution function. For this, function takes the color image having 24 bit pixel & 8 bit each for Red, green & Blue plane. For each symbol in ciphertext a pixel is allotted. Function divides the symbol in to part of two & three, after this, this part is scrambled in to the lower nibble of each plane according to parameters. After scrambling the data in to image, it is impossible to detect the change in picture by human visual system.

Integrating all these function in proper manner this algorithm makes the sense for encryption. This algorithm needs certain control parameters which can be generated from system itself & send along with data in scrambled manner.

## 6. Algorithm for Decryption

Algorithm for decryption is not as linear as for the symmetric type cryptosystem. First data is retrieved from image with inverse of B.D. function. Then Segmenter function is employed to generate proper block size. After this, U.S. function generate unique key from two keys. Then inverse P.X. function is employed to generate plaintext. In decryption, one integrator function is required to integrate all this function according to control parameters supplied along with data. This algorithm employs two inverse function, two same functions & a new function from encryption algorithm. So designing of this is easy comparing to encryption algorithm.

## 7. Implementation

This pseudo code is employed for implementation purpose. In this block size is taken as 8 and Rotating Permutation & Rotating Odd Shifter kept linear in fashion. Also bit distribution function is normal which replace last two or three bit from each byte & scramble the data in basic format.

### 7.1 Pseudo Code

```
DIGITAL FORTRESS ($\mathcal{p}, k_1, k_2, \Pi_P, \beta, \mathcal{I}, \mathcal{Cs}$)
y ← $\mathcal{p}$ ; s ← $S$
denote $\mathcal{Cs} = \mathcal{H}_d||\mathcal{M}_r||\mathcal{C}_r||S||y_1||y_2||y_3||y_4||y_5$
denote  y = $Mb_1||Mb_2||Mb_3||\dots||Mb_n$
denote $k_1 = a_1||a_2||a_3||a_4||a_5||a_6||a_7||a_8$
denote $k_2 = x_1||x_2||x_3||x_4||x_5||x_6$
P ← $x_1||x_2||x_3$ ; M ← $x_4||x_5||x_6$
q ← 1
for i ← 1 to n
{if mod(i,8) = 0 then
          {q ← q+1
          $x_4$ ← mod ($\mathbb{x}_5 * \mathbb{x}_6$, $y_1$)
          $x_5$ ← $\mathbb{x}_4$ - mod ($y_2$*q, $y_3$)
          $x_6$ ← $\mathbb{x}_5$ + mod ($y_4$*q, $y_5$) }
$x_1$ ← mod ($x_1$, 4)
$x_2$ ← mod ($x_2$, 4)
$x_3$ ← mod ($x_3$, 4)
n ← mod ($x_4$*i^$x_1$+ $x_5$*i^$x_2$+ $x_6$*i^$x_3$ , 64)
denote $Mb_i = b_1||b_2||b_3||b_4||b_5||b_6||b_7||b_8$
$ak_i$ ← $\underline{\sqcap_{\mathrm{mod}(i,8)}}$ ($\Pi_P$ ( $\prod_n$ ($k_1$) ) )
denote $ak_i = a_1||a_2||a_3||a_4||a_5||a_6||a_7||a_8$
if $\mathcal{H}_d$= 1 then
{if $\mathcal{M}_r$ = 1 then  { $L_i \leftarrow b_8\oplus a_8||b_7\oplus a_7||b_6\oplus a_6||b_5\oplus a_5$ }
          else      { $L_i \leftarrow b_5\oplus a_5||b_6\oplus a_6||b_7\oplus a_7||b_8\oplus a_8$ }
          $R_i \leftarrow b_1\oplus a_1||b_2\oplus a_2||b_3\oplus a_3||b_4\oplus a_4$
          if $\mathcal{C}_r$ = 1 then      { $en_i = R_i || R_i\oplus L_i$}
          else                  { $en_i = R_i\oplus L_i || R_i$ }}
else
{if $\mathcal{M}_r$ = 1 then  { $R_i \leftarrow b_4\oplus a_4||b_3\oplus a_3||b_2\oplus a_2||b_1\oplus a_1$ }
          else      { $R_i \leftarrow b_1\oplus a_1||b_2\oplus a_2||b_3\oplus a_3||b_4\oplus a_4$ }
          $L_i \leftarrow b_5\oplus a_5||b_6\oplus a_6||b_7\oplus a_7||b_8\oplus a_8$
          if $\mathcal{C}_r$ = 1 then      { $en_i = R_i\oplus L_i || L_i$ }
          else                  { $en_i = L_i || R_i\oplus L_i$ }}
$\mathcal{c}_i \leftarrow en_i \oplus \Lambda_{\mathrm{mod}(s+i,32)}$
}
denote $\mathcal{C} = \mathcal{c}_1||\mathcal{c}_2||\mathcal{c}_3|| \dots || \mathcal{c}_n$
m ← $n^2$
denote $\mathcal{I} = f_1||f_2|| f_3 || \dots ||f_m$
 li_array ← convert($\mathcal{C}$ )
denote li_array = $el_1||el_2||el_3|| \dots ||el_m$
denote $f_j = r_j||g_j||b_j$
for i ← 1 to m { $f_i \leftarrow \beta( f_j , el_i)$ }
$\mathcal{G} \leftarrow f_1||f_2|| f_3 || \dots ||f_m$
return($\mathcal{G}$)
```

### 7.2 Notations

| | |
|---|---|
| $\mathcal{p}$ : Plaintext | $k_1$ : Alphanumeric Key |
| $k_2$ : Numeric Key | $\Pi_P$ : Permutation matrix |
| $\beta$ : Bit distribution parameter | $\mathcal{H}_d$ : Half decision |
| $\oplus$ : Xor | $\mathcal{M}_r$ : Mirror decision |
| $\mathcal{I}$ : Image | $\mathcal{C}_r$ : Cross decision |
| $\mathcal{Cs}$ : Control Signal | $S$ : Shifter value |
| $x_4$ : Present Value | $\mathbb{x}_i$ : Previous Value |
| $\underline{\sqcap_n}$ : Rotate n (byte) | $\prod_n$ : Rotate n (bit) |
| $\Lambda_n$ : Value at location n | \|\| : Divide data in group |

## 8. Protocol Requirement

This algorithm requires lots of control parameter along with two keys. These control parameters has to be pass for proper & unique decryption. To pass this parameter, system requires certain protocol which transfers the information regarding the control parameters. These control parameters are sending in such a way that it can't be utilized.

*Table 1:* Control Parameter Distribution For Protocol Along With Size

| Sr. No. | Function | Size (bit) |
|---|---|---|
| 1 | Rotating Key Function | 24 |
| 2 | Rotating Permutation | 24 |
| 3 | Modified XORing | 3 |
| 4 | Rotating Odd Shifter | 5 |
| 5 | Bit Distribution Function | 24 |

Table 1 depicts the distribution for control parameter along with size of them.

## 9. Perfect Secrecy

Perfect Secrecy is measure for any system to possess the highest amount of security & it is derived from probability distribution function of Plaintext, Key & cipher Key [3]. It states that crypto system said to possess the property of Perfect Secrecy if & only if the Ciphertext is independent from message [3].

For analyzing this property of Digital Fortress Algorithm, assumption is taken that Rotating Key Function generate randomize Key & consider only permuted XORing function.

$\mathcal{P}(\mathcal{M})$ - Probability distribution of plain text $\mathcal{M}$

$\mathcal{P}(\mathcal{C})$ - Probability distribution of cipher text $\mathcal{C}$

$\mathcal{P}(\mathcal{M}/\mathcal{C})$ - Conditional Probability distribution of Plaintext $\mathcal{M}$ over Ciphertext $\mathcal{C}$

$$\mathcal{P}(\mathcal{M}/\mathcal{C}) = \mathcal{P}(\mathcal{M}\ and\ \mathcal{C}) / \mathcal{P}(\mathcal{C}) \qquad (1)$$

The event ($\mathcal{M}$ *and* $\mathcal{C}$) is the same as the event ($\mathcal{M}$ *and* $\mathcal{p}$) where $\mathcal{p}$ is the pad which equals $\mathcal{M}\oplus C$. Since the message and the pad are independent events.

From eq.1

$$\mathcal{P}(\mathcal{M}\ and\ \mathcal{C}) = \mathrm{P}(\mathcal{M}\ and\ \mathcal{p})$$

$$= \mathcal{P}(\mathcal{M})\ \mathcal{P}(\mathcal{p}) \qquad (2)$$

The probability of $\mathcal{P}(\mathcal{C})$ is the probability that a message $\mathcal{M}$ and a pad $\mathcal{p}$ came together to form $C$.

For every message $\mathcal{M}_i$ there is exactly one pad $\mathcal{p}_i$ yielding $\mathcal{C}$, namely, $\mathcal{p}_i = \mathcal{M}_i \oplus C$

So, $\mathcal{P}(\mathcal{C}) = \sum_i \mathcal{P}(\mathcal{M}_i\ and\ \mathcal{p}_i) = \sum_i \mathcal{P}(\mathcal{M}_i)\ \mathcal{P}(\mathcal{p}_i)$

$$= (1/2^n)\ \sum_i \mathcal{P}(\mathcal{M}_i) = 1/2^n \qquad (3)$$

Also, $\mathcal{P}(\mathcal{p}_i) = \mathcal{P}(\mathcal{p}) = 1/2^n \qquad (4)$

So, from eq.3 and eq.4 : $\mathrm{P}(\mathcal{C}) = \mathcal{P}(\mathcal{p}) \qquad (5)$

Substituting eq.5 & eq.2 in to eq.1

$$\mathcal{P}(\mathcal{M}/\mathcal{C}) = \mathcal{P}(\mathcal{M})\ \mathcal{P}(\mathcal{p}) / \mathcal{P}(\mathcal{p}) = \mathrm{P}(\mathcal{M})$$

So, $P(\mathcal{M}/\mathcal{C}) = \mathcal{P}(\mathcal{M})$

It means that knowledge about message can’t be extracted from Ciphertext because dependency does not exist between them.

## 10. Cryptanalysis

This algorithm produces ciphertext in such a way that that only few frequency component is present. It is impossible to attack on this algorithm by any kind of attack because these algorithm posses the property of Perfect Secrecy & hence having the infinite unicity distance [3]. Unicity distance indicate that no. of ciphertext symbol require to decrypt it in unique manner [3]. Here along with two key certain controls are necessary for unique & meaningful decryption which makes algorithm more immune to known plaintext attack [4].

## 11. Limitation

This algorithm has certain limitations like control parameters have to be passed. If this parameter is changed then unique decryption is not possible. Another problem is requirement of image which lead to increase the size of encrypted data if Bit distribution function is utilized. This will lead to introduce redundancy which is not at all desire for any part of data but only for hiding the data. The redundancy requirement is as high as twice of original data which imposes certain limitation on system in terms of memory utilization. One of the problems is advance version of this algorithm because it requires lots of power along with high memory requirement.

## 12. Simulation Result of Algorithm

Simulation of this algorithm is done with the help of MATLAB 7.0. Simulation result is shown in graphical format in terms of figure.

As input, a file having size of 4kB is employed along with two Key & certain parameters are shown below.

*Table 2:* Simulation Parameter

| Parameter | Value |
|---|---|
| Input File | Plaintext.text |
| Output File | Ciphertext.dat |
| Alphanumeric Key | asdfgbnmv |
| Numeric Key | 231253 |
| Shifter | 23 |
| Permutation Matrix | [ 2 1 3 7 8 5 4 6 ] |
| Right / Left | 1 |
| Mirror / Simple | 1 |
| Cross / Normal | 1 |

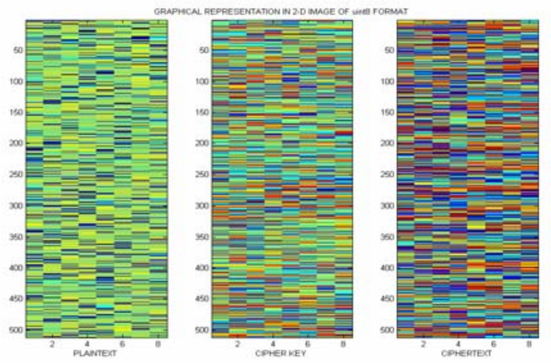

*Figure 3*: Continuous Data In 8 Byte Format of Plaintext, Key & Ciphertext

Figure 3 show the data format in 8 byte format where each color represents the certain symbol ranging from 0 to 255. This is the continuous data format in progressive scanning from top to bottom & left to right.

Figure 4 show the histogram representation of Plaintext, Key & Ciphertext. By analyzing it, it is clear from distribution that in the Ciphertext each symbol is equally probable which satisfy the condition of Perfect Secrecy. Figure 5 show the frequency components of Plaintext, Key, and Ciphertext. Figure 6 & 7 are original & manipulated image respectively. It shows that Plaintext is having the entire frequency component & according to Information Theory, it has highest amount of information but in cipher text only few frequencies are dominant so it has very low amount of information present in it. This lucid observation lead to that linear attack, differential attack & brute force attack are not possible for this algorithm.

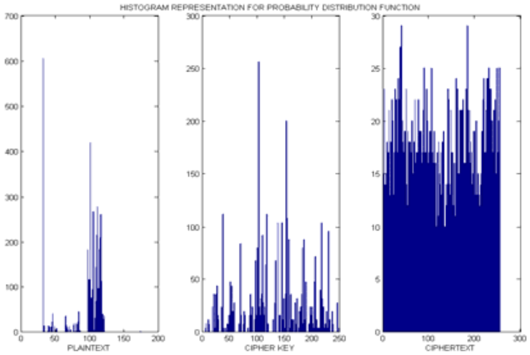

*Figure 4:* Histogram Representation of Plaintext, Key & Ciphertext

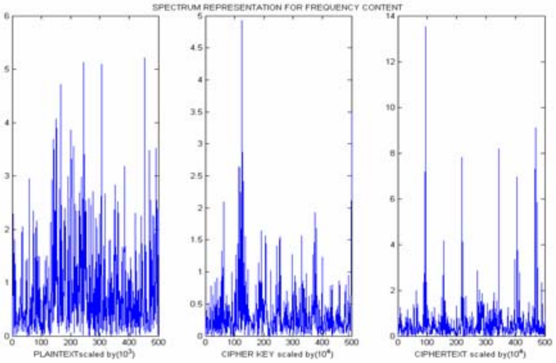

*Figure 5:* Spectrum of Plaintext, Key & Ciphertext

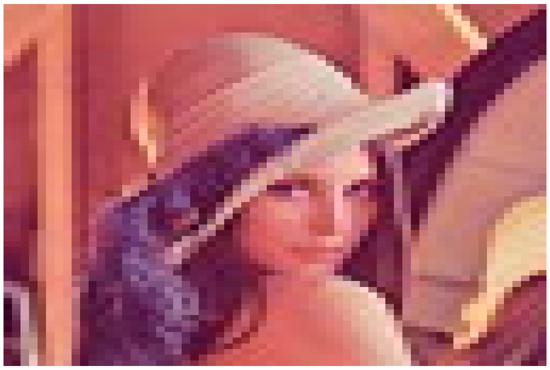

*Figure 6:* Original Image

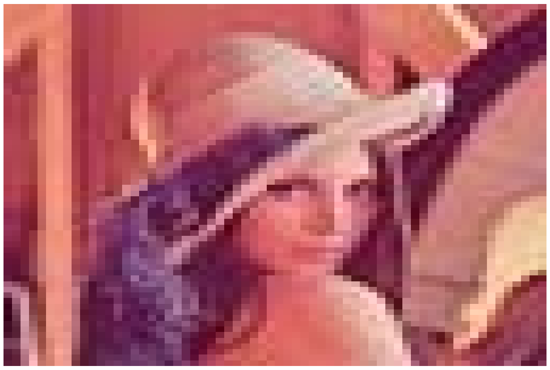

*Figure 7:* Manipulated Image

## 13. Conclusion

Digital Fortress is a proposed crypto algorithm which is enhanced & modified version of Vernam's OTP to attain a state of Perfect Secrecy which lead algorithm to the pinnacle of secrecy so all the attack fails against it. By this algorithm Perfect Secrecy is achieved by randomizing finite small key by Rotating Key Function to support Permuted XORing which utilizes the Rotating Permutation, Modified XORing & Rotating Odd Shifter. This algorithm performs all the basic function in its primitive form so computing power requirement is very low. Proposed algorithm is immune towards all kind of existing attack which is shown in simulation result & proven by mathematical formulas given by Perfect Secrecy Theory. Proposed algorithm is in its basic form so there are future scopes to enhance it by including recursivity in it as done in most of the encryption standards. Moreover, this algorithm is answer to the requirement of modern communication system like low computation power, lower time for execution & immune to attack.

## 14. Acknowledgement

Author is really grateful towards the Department of Electronics & communication, Institute of Technology, Nirma University, Ahmedabad for their generous help & support.

Author would like to thank Prof. Y. N. Trivedi & Prof. N. P. Gajjar for their helpful comments & discussions. Author would also like to thank the colleague students for their comments & reviews.

Author is really grateful towards the Mr. Dan Brown, the author of book "Digital Fortress" for giving such a wonderful idea through the book & also for such a good book. Last but not the least; author is thankful to the Almighty who blessed me with the zeal to work hard.

## 15. References

[1] *"Security Requirements for cryptographic modules "*, FIPS (Federal Information Processing Standard) Publication 140-1,US Department of Commerce / National Institute of standards & technology

[2] D. Welsh; *"Codes & Cryptography"*, Oxford Science Publication ,1998

[3] J. A. Buchmann; *"Introduction to Cryptography",* Springer-Verlag, New York, Second Edition,2001

[4] Gilbert Vernam; *"Vernam's Cipher",* Bell System Technology Journal,1918

[5] C. E. Shannon; *"Communication Theory of Secrecy System"*, Bell System Technology Journal,1949

[6] D. Stinson; *"Cryptography, Theory & Practice"*, CRC Press, Florida, Second Edition, 2002

[7] A. J. Menezes, S. A. Vanstone, D. C. Van Oorschot; *"Hand-book of Applied Cryptography"*, CRC Press, Florida, 1996

[8] *"Glossary for Computer System Security "*, FIPS (Federal Information Processing Standard) Publication 39,US Department of Commerce / National Institute of standards & technology

[9] Z. A. Kissel; *"Obfuscation of The Standard XOR Encryption Algorithm"*, Crossroads, The ACM Student Magazine

[10] *"Data Encryption Standard "*, FIPS (Federal Information Processing Standard) Publication 46-3,US Department of Commerce / National Institute of standards & technology