Encryption using XOR based Extended Key for Information Security – A Novel Approach

E. Anupriya*
School of Computing Science & Engineering
VIT University, Vellore, 632 014, India.

Amit Agnihotri
School of Computing Science & Engineering
VIT University, Vellore, 632 014, India.

Sachin Soni
School of Computing Science & Engineering
VIT University, Vellore, 632 014, India.

Sourabh Babelay
School of Computing Science & Engineering
VIT University, Vellore, 632 014, India.

Abstract - **The explosive growth of information, places a high demand for Information Security. Information Security deals with securing the information from unauthorized access or misuse of information either intentionally or accidentally. Information may be represented in many forms like text, documents, audio, video, images or maps. The standard and widely used form is documents. The objective of our work is to secure information present in these documents especially in a shared environment like peer-to- peer environment. We have proposed a novel symmetric encryption method which uses XOR based extended key to encrypt all characters which includes alphabets, numerals and special characters. Our approach differs from traditional approaches which rely on numeric key values. [6], [1]. Our intention is to strengthen the confusion which inherently makes the decryption a challenging task.**

Keywords: Information security, Encryption

I. INTRODUCTION

Huge number of documents are stored, retrieved, exchanged and used across the Web. These documents when shared are more vulnerable to information disclosure, destruction, and misuse on open access. In many cases the information is either hidden completely or presented in the encrypted form. The three primary issues of information security is to maintain confidentiality, integrity and availability of information. The confidentiality of information is to prevent the information from unauthorized access keeping intact, the correctness and availability of information.

The standard security mechanisms are authorization and authentication. Authentication is the process of validating users with a valid key. Once authenticated, the user is authorized for information access. When a user attempts to breach the basic security level, the next level of securing the information is to present the information in the encrypted form. Encryption may be symmetric or asymmetric based on the same key or different key used for encryption and decryption. [2], [4], [11]

Encryption is the process of encoding the message where as the decryption is the reverse process of encryption to get back the original information. Confusion and Diffusion are the two aspects attempted to achieve in any information security algorithms. The concept of confusion is to keep the relationship between key and cipher text as complex as possible. And the concept of diffusion is to ensure non-uniformity in the distribution of the individual letters in the plain text and redistribution in large cipher text structures, which is hard to detect. Symmetric key encryption is applied either for blocks or streams. [8] In block cipher, the plain text is divided into equal blocks and one key per block is used. Whereas in stream cipher, each character of the plain text is mapped with a key character using which the cipher text is generated. A good encryption algorithm possess the following properties: [4] , [11]

a) The encryption and decryption should be an abstraction for user, i.e. user should visualize that he is interacting with original non encrypted database, firing the required query.
b) Sensitive information, of the stored document, should be encrypted in such a way that it should not be directly usable by the adversary.
c) The length of the encrypted message should not exceed the original message.
d) The strength of the algorithm depends upon the security of the key. The security of the key is maintained using probability and substitution method. As the number of substitution increases the security of key increases.

We present a novel symmetric encryption method which uses XOR based extended key to encrypt all characters which includes alphabets, numerals and special characters. Also, we have adopted hybrid approach in which block ciphering and stream ciphering are applied. Formerly, the plain text is divided into equal blocks to the extent possible. Block ciphering is applied to this part. The remaining plain text whose length is lesser than the block size is applied with stream cipher. However, the strength of the algorithm is purely based on the key. The security of the key is maintained using probability and substitution method. We have extended the key by applying the XOR operation on the key characters to generate a new key for subsequent block. Therefore, even the repeated words will not have the same cipher text. Inherently, this increases the security of the key. Also, we have used characters from infeasible range (i.e. non keyboard printable character) for mapping the plain text into cipher text. This aspect will confuse the information intruder and remains as strength to our algorithm.

II. OUR APPROACH

Classical cryptographic schemes especially, encryption schemes depend on the user key alone. Classical Vigenere's algorithm is limited with the use of alphabets alone for substitution. (i.e. 26 X 26 alphabet matrix arranged in circular fashion in each row). Other algorithms stated in [1],[2], [4], [6],[11] uses either matrix of numerals as in θ-Vigenere's algorithm or matrix of alphabets in rows and numerals in columns for substitution. 3Kdec algorithm [1] is limited with key repetition. (i.e. every tenth key will be same and repeated)

The example shown in Table 1 illustrates the limitation of using Classical Vignere's algorithm. In Table 1, key of six characters length is used and the second occurrence of CRYPTO in plaintext has the similar cipher text as the cipher text of first occurrence of CRYPTO. In practical, documents are not restricted with numerals and alphabets alone.

Documents may contain information in combination with alphabets, numerals and special characters (i.e. any ASCII character in the range of 0-255) may appear. However, the control characters (ASCII value of 0-31) are not considered as document text and may be ignored. Our approach combats the limitations like, use of alphabets alone, use of numeric values alone and repetition of cipher text. And they are handled with our new key expansion algorithm.

Table 1: Classical Vignere's Algorithm limitation

Key:	ABCDAB CD ABCDA BCD ABCDABCDABCD
Plain text	***CRYPTO*** IS SHORT FOR **CRYPTO**GRAPHY
Cipher text	**CSASTP** KV SIQUT GQU **CSASTP**IUAQJB

Our paper proposes a novel symmetric encryption method which uses XOR based extended key to encrypt all characters which includes alphabets, numerals and special characters present in document. The primary intention is to strengthen the confusion part of the algorithm. The strengthening is achieved through:

Non Key Repetition: A document may contain not only ASCII keyboard characters (ASCII value range 32- 127) but also Extended ASCII characters (ASCII value range128-255). User specified key is not used for enciphering the whole document instead it is used as an input to first block of plain text to generate key using the XOR based key expansion algorithm. Using which, the cipher text is synthesized. Therefore, a new key is generated for every subsequent block of plain text.

Hybrid substitution: User uploads the document to be encrypted as an input to the encrypting algorithm. The key expansion algorithm generates new key for every block and substitutes in a hybrid manner which includes both block cipher substitution followed by stream cipher substitution. This approach uses extended modified Vignere's matrix. The column index ranges from 255 to 0. (Originally it is 0-255) User enterable ASCII keyboard characters range from 32 to 127. (i.e. row index range from 0-95). The final matrix size is of 96 X 256. This mapping solution confuses the usage of characters thereby strengthening our algorithm. Extended modified Vignere's matrix is shown in figure 1.

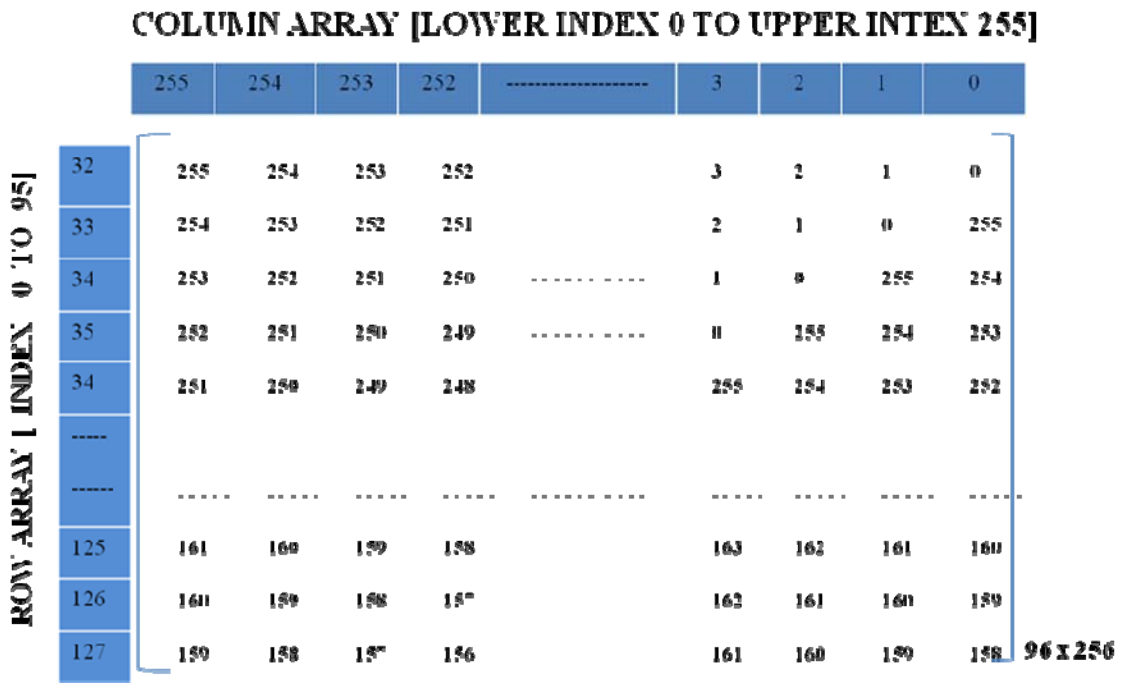

Figure 1. Extended modified Vignere's matrix

The cryptographic system is represented by a *quintuple (M, C, K, E_k, D_k)*,where M is for Plain text message ,C is for Cipher text message ,K is for set of Keys , E_k is for family of encoding function , D_k is for family of decoding function by the key. The size of the plain text block is equal to the size of the user specified key. The user specific key length should be no less than six characters to ensure key complexity. Any cryptographic mechanism should adhere to Equation 1.i.e.The decryption of encrypted information should yield the original information.

$$D_\theta(E_\theta(M)) = M \qquad \text{(Eq. 1)}$$

Where

D is the decoding function of plain text M
E is the encoding function of plain text M
θ is the size of the user specific key

Section 3 deals with key expansion and section 4 deals with encryption and decryption process. Section 5 includes implementation and discussion. Section 6 concludes our work.

III. KEY EXTENSION

The repetition of key is major disadvantage. Any information hacker can easily find out the plain text from cipher using key repetition strategy adopted. Originally, repetition of key concept is suggested by Kasiski and Kerckhoff [4]. The key repetition concept relies on the fact that specific alphabets in a language are relatively repetitive as per Index of Coincidence. (i.e. commonly referred as IC) The value of IC states the frequency of use of a particular character in a language.[11] This repetition factor opens up high possibilities to derive at the length of the key. Once the length of the key is obtained, it is easier to arrive at the original key.

Our new key expansion algorithm is based on extended modified Vignere's matrix. This extended matrix helps to introduce strong confusion which attempts to show poor relationship between user key and the cipher text. The confusion is achieved by expanding the user key by applying exclusive OR (XOR) operation on actual user key, character by character. The key algorithm starts with the understanding the range of ASCII characters and their categories shown in Table 2.

Table 2. ASCII Value Range by category

No.	Category	ASCII Value Range	Bits
1.	ASCII Control Characters	0 to 31	5
2.	ASCII keyboard character	32 to 127	6 or 7
3.	Extended ASCII Code	128 to 255	8

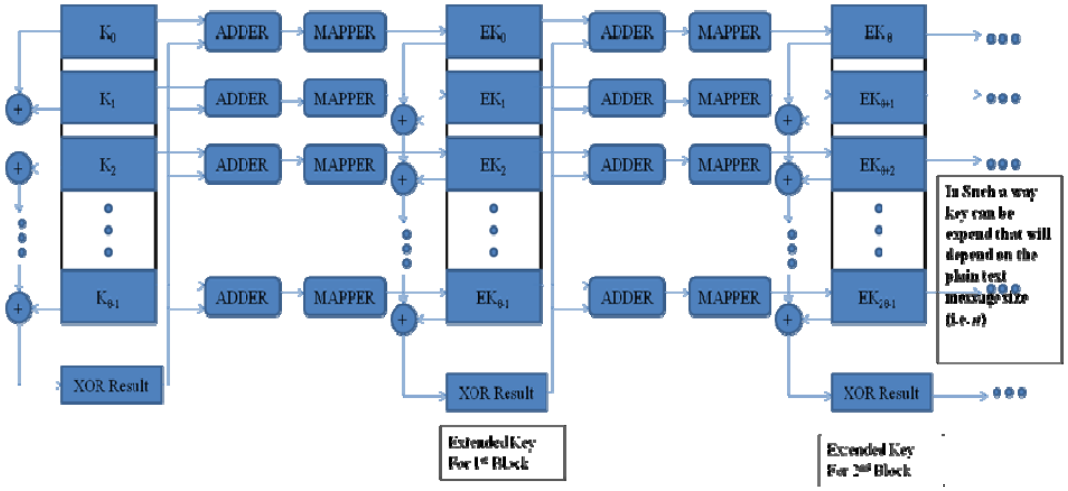

Figure 2. Key Expansion Block Diagram

User may enter any key with length not less than six characters which may include ASCII keyboard characters specified in the Table 2. Each user key character is padded with zero in the most significant bit of the character to obtain compatibility with ASCII character if the key length is less than seven. Now, the binary value equivalent of ASCII value for first character in the key is XOR ed with binary value equivalent of ASCII value for second character. This continues until the key length.

The final resultant of the XOR operation is added with each original key character to generate the first key for the first plain text block. Similarly, this first key is given to subsequent block for second key generation and the second key as input to the third block and so on. It continues until the block expires. The left out characters at the end are substituted using stream cipher.

Figure 2 sketches the key expansion process and figure 3 presents the key expansion algorithm. The ADDER at each stage may yield seven or eight bits as a result. If the resultant is in seven bits then it falls in the feasible range of ASCII values (32-127), hence no mapping is required. Otherwise, if the resultant is eight bits, then it falls in the extended ASCII range (128-255) which is special characters. Then, using range mapping scheme, the extended range is mapped onto the feasible range to get the extended key.

```
Input: "K" is an array of user key
       "θ" is length of the key
       "VAL" is counter for XORed value of keys
Output : "EK" is an array of extended key

KEYGEN(KEY[])
{
  Step 1.- VAL = 0

  Step 2.- For I = 1 :to θ repeat step 3

  Step 3 - VAL = VAL XOR K[I]

  Step 4.- For I = 1 :to θ repeat step 5

  Step 5.- EK [I] = K [I] + VAL
           IF EK [I] >127
               EK [I] = EK [I] – 32
               EK [I] = EK [I] MOD 96
               EK [I] = EK [I] + 32

           IF EK [I] < 32
               EK [I] = EK [I] + 32

  Step 6.- Return EK
```

Figure 3. Key Expansion Algorithm

Range mapping scheme is applicable, when the key obtained is in extended ASCII range. The ADDER resultant binary values may have ASCII values either in range 32-127 or 128-255 or in other cases; it may not generate ASCII values less than 32. The rationale behind is, the least value of six bit number is zero and when added with the original key character value will yield the original key character. On the other side, the maximum value of six bit number is 63, which is within the feasible region of the Vingere's Matrix.

Mapping Scheme: If the ASCII value of key is greater than 127, then value of 32 is subtracted from it. The result is applied with modulo of 96, which is the total number of rows of the matrix; as the row is meant for key and the column for plain text. It is again added with 32 to get the extended key. Say if key value is 135 then

$new\_key = 135 - 32 = 103 \bmod 96 = 7 + 32 = 39$

After all the blocks are exhausted then the remaining characters are substituted using stream cipher in the same way.

IV. ENCRYPTION AND DECRYPTION

A. Encryption

The PM_i represents plain text character, where i ranges as $0 \leq i \leq (n-1)$, and n is the total number characters present in the document. The plain text stream is divided into x blocks. For a given plain text of length n, there exists n / θ blocks, where θ is user key length and $x = n / \theta$.

PM_0	PM_1	-- --	PM_{n-1}

The x may take integer or fractional value.[9] Practically, the plain text may not be perfectly divided. When x takes fractional value, x equals to $x + (w/\theta)$, where w is the number of remaining characters after applying block cipher. Subsequently, the stream cipher is applied on w/θ *characters.* The value of w lies in between $1 \leq w \leq \theta$ because θ divides *n.*

Block is considered as *Datagram* for plain text block and *Cryptogram* for cipher text block [6], in which θ will be the length of each block. Our extended key approach with hybrid substitution relaxes the constraints on the message length and the key length. Hence, our approach may be applied on plain text with no restriction on either message length or the key length. The construction of each plain text block and the corresponding extended key block is represented below.

Plain text block:

PM_{θ} *to* $PM_{(\theta-1)}$ → $PM_{\theta[1]}$
PM_{θ} *to* $PM_{(2*\theta)-1}$ → $PM_{\theta[2]}$
……
$PM_{(x-1)*\theta}$ *to* $PM_{(x*\theta)-1}$ → $PM_{\theta[x]}$

The remaining characters are applied with stream cipher like $PM_{(x*\theta)}$ *to* $PM_{(n-1)}$.

Extended key block for corresponding plain text block:

For each plain text block ranging from 0 to x, the corresponding extended key block is generated through the key expansion algorithm. The extended key is represented as EK.

EK_{θ} *to* $EK_{\theta-1}$ → $EK_{\theta[1]}$
EK_{θ} *to* $EK_{(2*\theta)-1}$ → $EK_{\theta[2]}$
.....
$EK_{(x-1)*\theta}$ *to* $EK_{(x*\theta)-1}$ → $EK_{\theta[x]}$

The transformation of plain text blocks $PM_{\theta[x]}$ to cipher blocks $CM_{\theta[x]}$ is shown in Figure 4 and algorithm in Figure 5. The enhanced algorithm includes the equation 2 which eliminates the reference to Vignere's matrix to improve the time complexity.

$$C_i = P_i + K_i - 32 \qquad \text{(Eqn.2)}$$

$P_i \rightarrow$ Plain Text Character
$K_i \rightarrow$ Key Character
$C_i \rightarrow$ Cipher Character

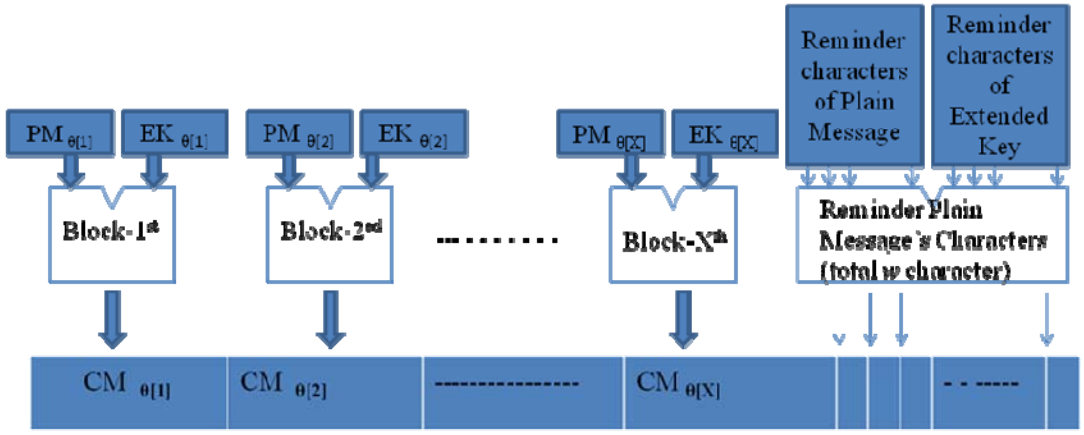

Figure 4. Encryption of plain text to cipher text

```
Input  : "PM" is an array of plain text
         "α" is the length of plain text
         "EK" is the array of extended key
         "θ" is the length of extended key
Output : "CM" is array of encrypted text
         "CNT" is counter for encrypted array

ENCRYPT(PM[])
{
  Step 1 - CM ← NULL
           CNT ← 1

  Step 2 - For I = 1 :α repeat α  step 3

  Step 3 - CM [I] = CM[I] + GETVAL(PM[I],EK[CNT])
           CNT = CNT + 1

           If CNT MOD θ = 0
              CNT = 1
              EK = KEYGEN(EK)

  Step 4 - Return CM
}

GETVAL(TXT,KEY)
{
  Step 1 - RTXT = TXT – KEY +32

  Step 2 - IF RTXT < 0
              RTXT = RTXT + 256

  Step 3 - Return RTXT
}
```

Figure 5. Encryption Algorithm

B. Decryption

Decryption is the reverse process of encryption and it is shown in figure 6. Since the proposed algorithm is a symmetric key algorithm, the same key is used for decryption process. For each cipher block and for each character in i^{th} position, the index i of the extended key character EK_{i-1} identifies the row index which will also be the row index of CM_{i-1}. The column index of the cipher character CM_{i-1} and the row index indentified already yields the plain text character PM_{i-1}. Therefore, all plain text characters of x blocks are retrieved and the same process is also repeated for remaining w characters of the stream. The plain text without reference to Vigenere's matrix may be retrieved using equation 3 and the decryption algorithm is shown in figure 7 .

$$P_i = C_i - K_i + 32 \qquad \text{(Eqn.3)}$$

$P_i \rightarrow$ Plain Text Character
$C_i \rightarrow$ Cipher Character
$K_i \rightarrow$ Key Character

V. IMPLEMENTATION AND RESULTS

The system is implemented on .Net platform with Visual studio 2005 as front end and Microsoft SQL Server 2005 as back end. User friendly front end is developed to browse the files to be encrypted or decrypted with folder navigation options. Documents of different types with file extensions like .txt, .doc, .docx, .html may be encrypted and decrypted. The whole document is converted to cipher text. In case of html documents, the html tags are ignored and the rest of the text is encrypted or decrypted.

A. Time and Space Complexity

The encryption or decryption mechanism for each character makes a reference to Vignere's matrix. Our attempt to reduce this search space resulted in equation 2 and 3 which yielded the same character as retrieved using Vignere's matrix. The complexity of overall work is divided into two parts; First, the time required to encrypt or decrypt. Second, the time required to generate extended key and character substitution in blocks. For encryption or decryption, the algorithm runs for α times, where α is the total number of characters to be encrypted or decrypted. For x blocks, the key generation algorithm consumes θ*x time, where θ is the user key length. Therefore, the total time required is

T = Encryption / Decryption time + Extended key generation and substitution time

$T = \alpha + \theta * x$
$T = \alpha + \alpha$ (as $\alpha = \theta * x$)
$T = 2\alpha$
$T = O(\alpha)$

The algorithm exhibits linear behavior on eliminating the reference to Vignere's matrix and is shown in figure 8 for 100 documents.

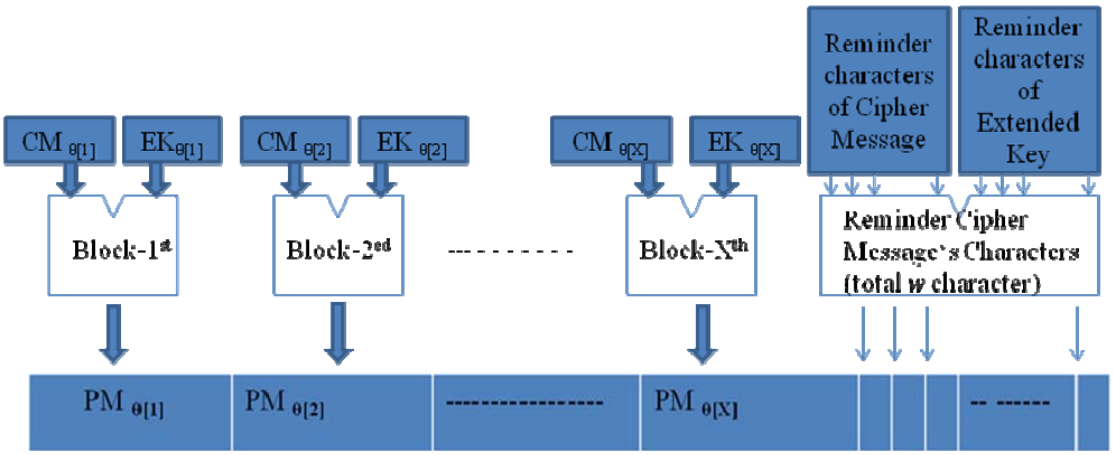

Figure 6. Decryption of cipher text to plain text

```
Input  -"CM" is an array of plain text
        "a" is the length of plain text
        "EK" is the array of extended key
        "θ" is the length of extended key
Output -"DM" is array of encrypted text
        "CNT" is counter for encrypted array

ENCRYPT(CM[])
{
  Step 1 - DM ← NULL
           CNT ← 1

  Step 2 - For I = 1 to repeat a  step 3

  Step 3 - DM[I] = DM[I] + GETVAL(CM(I),EK[CNT])
           CNT = CNT + 1

           If CNT MOD θ = 0
              CNT = 1
              EK = KEYGEN(EK)

  Step 4 - Return DM
}

GETVAL.(TXT,KEY)
{
  Step 1 - RTXT = TXT - KEY +32

  Step 2 - IF RTXT < 0
           RTXT = RTXT - 256

  Step 3 - Return RTXT
}
```

Figure 7. Decryption Algorithm

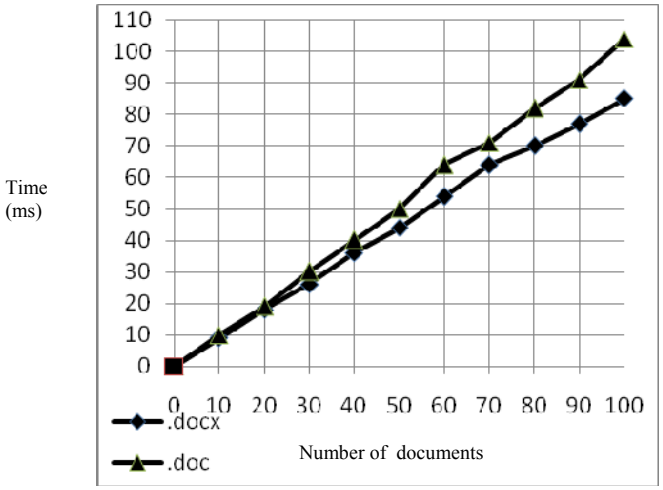

Figure 8. Document Vs Time Graph

VI. CONCLUSION

Information, mostly exist in the form of documents. When, information is shared in environment like peer-to-peer environment, the documents are highly vulnerable to risk of being hacked. Our work proposes a novel encryption algorithm using XOR operation to resolve the security issue by strengthening the confusion part and by using extended key characters for substitution, which lie mostly in the range of special characters. The observation illustrates the generation of cipher text as junk characters or special characters which does not provide information directly and confuses the hacker.

VII. REFERENCES

[1]. Kaur K. Dhindsa, K.S. Singh G, "Numeric To Numeric Encryption of Databases: Using 3Kdec Algorithm", In proceedings of IEEE International Conference on Advance Computing, IACC 2009, 2009.
[2]. Swarnendu Mukherjee, Swarnedu Bhattacharya and Amlan Chadhury, "Triple Layer Data Security" ACM Ubiquity.April 29-May 5,2008.
[3]. Lianzhong Liu Jingfen Gai, "A new lightweight database encryption scheme transparent to applications", In proceedings of sixth International Conference on Industrial Informatics INDIN 2008, 2008.
[4]. Phillip I Wilson and Mario Garcia,"A Modified Version of the Vignere Algorithm", International Journal of Computer Science and Network Security, vol.6, 2006.
[5]. Ghoualmi-Zine Nacira and Arar Abdelaziz, "Secured Net-Banking by θ-Vigenere in Syverson's Protocol", IEEE International Conference, 2005.
[6]. Nacira G, Abdelaziz A, "The θ-vigenere cipher extended to numerical data", In proceedings of International Conference on Information and Communication Technologies: From Theory to Applications, 2004.
[7]. Yuki Mitsumaya, Zaldy Andales, Takao Onoye, and Isao Shirakawa. "Burst Mode: A New Acceleration Mode for 128-bit Block Ciphers", In proceedings of IEEE International Conference on Custom Integrated Circuits, 2002.
[8]. Abdul Hamid M. Ragab, Nabil A. Ismai, Osama S. Farag Allah, "Enhancement and Implementation of RC6^{TM} Block Cipher for Data Security", In proceedings of IEEE International Conference of Data Security, 2001.
[9]. Yuke Wang, "Residue To Binary Converter Based on New Chinese Remainder Theorems", IEEE Transactions on Circuit and System-II Analog and Digital Signal Processing , vol. 47, 2000.
[10]. P.Pfleeger, Sari Lawrence Pfleeger and Deven Shah, "Security in Computing" Fourth Edition, Book
[11]. Schineier B, "Applied Cryptography", John Wiley and Sons publications, New York, 1995