ENHANCE THE DATA SECURITY BY CHANGING THE ENCRYPTION TECHNIQUE BASED ON DATA PATTERN IN BLOCK BASED PRIVATE KEY DATA ENCRYPTION

Sanjit Mazumder  
Assistant Professor (CSE),  
Modern Institute of Engineering & Technology, Bandel, West Bengal.

Neha Kumari Shaw, Bidisha Dey, and Farhin Mahmuda Laskar  
CSE 4th Year student of Modern Institute of Engineering & Technology, Bandel, West Bengal.

Abstract: Cryptography is the process of data hiding from unauthorized users. Historically, the term “Cryptography” has been associated with the problem of designing and analyzing encryption schemes (i.e., scheme that provide secret communication over insecure communication media). With the growth of internet secure data transmission is more and more essential and important. Sometime single encryption technique is not sufficient to protect the valuable data. We developed a new algorithm which can change the encryption technique with change of plane text pattern to enhance the confusion and as well as diffusion.

Keyword: Encryption, Decryption, Cipher Text, Private Key Data Encryption

1. INTRODUCTION

With the increase of internet users, the secure data transmission through internet is more and more essential and important. There are many private key data encryption algorithm are available. Some of them are time consuming and some of them are power consuming. We develop a new algorithm which can change the encryption process with change of data pattern to improve the data security. In section 2, Algorithm is defined. While section 3 shows the example of whole process, section 4 is result and section 5 is discuss the analysis and conclusion.

2. ALGORITHM

Here we use symmetric key block based encryption technique. We choose block cipher approach as it is more secure and convenient to use. In our program we use a single key(n x n matrix), as private key, to encrypt and decrypt data. In this algorithm the plain text is converted into cipher text or encrypted text using the private key and this cipher text is decrypted into plain text using the same key. The key should be shared by both sender and receiver using a secure channel. Basic in this section we discuss the key generation process in section 2.1, concept is shown in fig:1 and fig:2.

2.1 Key Generation Process

Step 1: Here we generated a randomly generated m*m matrix as a key.

Step 2: The key should contain information about the length of UB (unchanged bits) which will be append with the sub key to generate the original key.

2.2 Encryption Process[2]
Step 1: Decompose the $L$ bits plain text into $n$ no of blocks (they are $B_1, B_2, ..., B_n$) so that each blocks will contain $m \times m$ no of bits. After splitting, no of blocks will be $[n/(m\times m)]$. Let’s say it is $x$.

Step 2: After splitting the text the remaining bits will be $[n - (n/(m\times m))] = UB$. Those bits will append with the cipher text.

Step 3: then each block are transposed. After transpose the blocks are $B^T_1, B^T_2, ..., B^T_n$.

Step 4: Each block are checked as

If ($\lceil n/m\times m \rceil = 0$)

```
    Loop: 1 to x
    $C_i = B^T_i + \text{Key}$; [where $B^T_i$ represent the blocks and $i = 1$ to $x$]
```

else

```
    Loop: 1 to x
    $C_i = B^T_i - \text{Key}$; [where $B^T_i$ represent the blocks and $i = 1$ to $x$]
```

Step 5: $C_i \text{ UB}$ is appended to produce the encrypted text. [where $i = 1$ to $x$]

Step 8: Exit.

2.3. Decryption Process[2]

Step 1: We take the previous encrypted text and decompose it into $n$ no of blocks so that each block contains $m\times m$ no of bits. After splitting the no of blocks will be $\lceil n / (m\times m) \rceil$. Let’s say it is $x$.

Step 2: After splitting the remaining bits will be $[n -(n / m\times m)]$. This will be treated as unchanged blocks (UB).

Step 4: Check

If ($\lceil n / m\times m \rceil = 0$)

```
    Loop : 1 to x. [where $x$ is the no of blocks]
    $C_i = B_i - \text{Key}$ [where $B_i$ represent the blocks and $i = 1$ to $x$.]
```

else

```
    Loop : 1 to x. [where $x$ is no of blocks]
    $C_i = B_i + \text{Key}$ [where $B_i$ represent the blocks and $i = 1$ to $x$.]
```

Step 5: Transpose each blocks(say $C^T_1, C^T_2, ..., C^T_n$).

Step 6: Append $C^T_1 \text{ UB}$ to produce the decrypted text. [where $i = 1$ to $x$]

Step 7: Exit.

3. EXAMPLE

3.1 Key generation: Randomly generate an 3 x 3 matrix which is

$$
\begin{pmatrix}
3 & 3 & 3 \\
3 & 3 & 3 \\
3 & 3 & 3 \\
\end{pmatrix}
$$

3.2 Encryption[3]: Let assume the plane text is: “It is an ex of encryption”.

3.2.1 Decompose the plain text into no of blocks, where the block size is same as the key size, which are $B_i =$

```
* I t *
* i s *
* a n *
```

and

```
B_2 =
* e x *
* o f *
* e n c *
```

[where * represent the blank space]

After decomposing, the remaining bits are – encryption (unchanged bits)

3.2.2 Each blocks now transposed

```
B^T_1 =
* I t *
* i a *
* t s *
* n *
* * *
```

and

```
B^T_2 =
* e o *
* e n *
* x f *
* * *
* c *
```

3.2.3 If ($\lceil n / m\times m \rceil = 0$) then $C_1 = B^T_1 + \text{key}$

Else $C_1 = B^T_1 - \text{key}$

So, $B^T_1 - \text{key} = C_1 =$

```
* F f *
* q p *
* k *
* * *
```

and

```
B^T_2 + \text{key} = C_2 =
* h r *
* h *
* i q *
* # # f *
```

So, encrypted text will be Ff^qpkhrh{iq##f

3.3 Decryption[3]

3.3.1 Decompose the cipher text[3] into blocks
\[ B_1 = \begin{bmatrix} F & f & \wedge \\ q & p & k \\ * & * & * \end{bmatrix} \]

and

\[ B_2 = \begin{bmatrix} h & r & h \\ \{ & i & q \\ # & # & f \end{bmatrix} \]

3.3.2 Check if \( \left\lfloor \frac{n}{m*m} \right\rfloor \equiv 0 \) then \( C_1 = B_1 - \text{Key} \)

Else \( C_1 = B_1 + \text{Key} \)

So, \( B_1 + \text{Key} = C_1 = \]

\[ \begin{bmatrix} I & i & a \\ t & s & n \\ * & * & * \end{bmatrix} \]

and

\( B_2 - \text{Key} = C_2 = \]

\[ \begin{bmatrix} e & o & e \\ x & f & n \\ * & * & c \end{bmatrix} \]

3.3.3 Each block then transposed.

\[ C^T_1 = \]

\[ \begin{bmatrix} I & t & * \\ i & s & * \\ a & n & * \end{bmatrix} \]

and

\[ C^T_2 = \]

\[ \begin{bmatrix} E & x & * \\ O & f & * \\ E & n & c \end{bmatrix} \]

So, the decrypted text \([3]\) is – It is an ex of enc.

3.3.4 Unchanged bits are now appended with the encrypted text to generate the encrypted text – It is an ex of encryption.

Key structure:

<table>
<thead>
<tr>
<th>Segment</th>
<th>Decryption</th>
<th>Maximum no of bits required (size)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment-1</td>
<td>Main part of the key</td>
<td>(m*m)</td>
</tr>
<tr>
<td>Segment-2</td>
<td>Unchanged block</td>
<td>([L%(m*m)])</td>
</tr>
<tr>
<td>Segment-3</td>
<td>Data information</td>
<td>1-bit</td>
</tr>
</tbody>
</table>

Figure Details:

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Figure No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Block diagram of encryption process ([3]).</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Block diagram of decryption</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>File Size (Byte)</th>
<th>Decryption Time (RSA)</th>
<th>Decryption Time (Our Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>10.764501</td>
<td>15.43956</td>
</tr>
<tr>
<td>115</td>
<td>4.876532</td>
<td>4.065934</td>
</tr>
<tr>
<td>116</td>
<td>5.984565</td>
<td>3.956044</td>
</tr>
<tr>
<td>121</td>
<td>8.764543</td>
<td>3.956044</td>
</tr>
<tr>
<td>130</td>
<td>6.345298</td>
<td>5.769231</td>
</tr>
<tr>
<td>134</td>
<td>7.639087</td>
<td>4.230769</td>
</tr>
<tr>
<td>150</td>
<td>5.109312</td>
<td>5.846154</td>
</tr>
<tr>
<td>152</td>
<td>4.298222</td>
<td>7.802198</td>
</tr>
<tr>
<td>219</td>
<td>6.729874</td>
<td>4.56044</td>
</tr>
<tr>
<td>223</td>
<td>7.297845</td>
<td>6.483516</td>
</tr>
<tr>
<td>225</td>
<td>2.198376</td>
<td>3.956044</td>
</tr>
<tr>
<td>229</td>
<td>7.029801</td>
<td>4.010989</td>
</tr>
<tr>
<td>231</td>
<td>7.280012</td>
<td>4.010989</td>
</tr>
<tr>
<td>234</td>
<td>15.638902</td>
<td>12.142857</td>
</tr>
<tr>
<td>238</td>
<td>7.892892</td>
<td>4.340659</td>
</tr>
<tr>
<td>243</td>
<td>5.298732</td>
<td>4.010989</td>
</tr>
<tr>
<td>244</td>
<td>5.982302</td>
<td>3.956044</td>
</tr>
<tr>
<td>269</td>
<td>7.92831</td>
<td>5.10989</td>
</tr>
<tr>
<td>280</td>
<td>4.801223</td>
<td>4.835165</td>
</tr>
<tr>
<td>288</td>
<td>9.30928</td>
<td>6.703297</td>
</tr>
<tr>
<td>294</td>
<td>8.29845</td>
<td>5.43956</td>
</tr>
<tr>
<td>358</td>
<td>8.092287</td>
<td>4.450549</td>
</tr>
<tr>
<td>428</td>
<td>8.27892</td>
<td>7.857143</td>
</tr>
<tr>
<td>594</td>
<td>10.876533</td>
<td>8.406593</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>File Size (Byte)</th>
<th>Encryption Time (Our Algorithm)</th>
<th>Decryption Time (Our Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>15.43956</td>
<td>0.354333</td>
</tr>
<tr>
<td>115</td>
<td>4.065934</td>
<td>0.008965</td>
</tr>
<tr>
<td>116</td>
<td>3.956044</td>
<td>0.052784</td>
</tr>
<tr>
<td>121</td>
<td>3.956044</td>
<td>0.052267</td>
</tr>
<tr>
<td>130</td>
<td>5.769231</td>
<td>0.053345</td>
</tr>
<tr>
<td>134</td>
<td>4.230769</td>
<td>0.053696</td>
</tr>
<tr>
<td>150</td>
<td>5.846154</td>
<td>0.053699</td>
</tr>
<tr>
<td>152</td>
<td>7.802198</td>
<td>0.053742</td>
</tr>
<tr>
<td>219</td>
<td>4.56044</td>
<td>0.053781</td>
</tr>
<tr>
<td>223</td>
<td>6.483516</td>
<td>0.153856</td>
</tr>
<tr>
<td>225</td>
<td>3.956044</td>
<td>0.054231</td>
</tr>
<tr>
<td>229</td>
<td>4.010989</td>
<td>0.054256</td>
</tr>
<tr>
<td>231</td>
<td>4.010989</td>
<td>0.060533</td>
</tr>
<tr>
<td>234</td>
<td>12.142857</td>
<td>0.454689</td>
</tr>
<tr>
<td>238</td>
<td>4.340659</td>
<td>0.054767</td>
</tr>
<tr>
<td>243</td>
<td>4.010989</td>
<td>0.054863</td>
</tr>
<tr>
<td>244</td>
<td>3.956044</td>
<td>0.064879</td>
</tr>
<tr>
<td>269</td>
<td>5.10989</td>
<td>0.054902</td>
</tr>
<tr>
<td>280</td>
<td>4.835165</td>
<td>0.054943</td>
</tr>
<tr>
<td>288</td>
<td>6.703297</td>
<td>0.254863</td>
</tr>
<tr>
<td>294</td>
<td>5.43956</td>
<td>0.154945</td>
</tr>
<tr>
<td>358</td>
<td>4.450549</td>
<td>0.154745</td>
</tr>
<tr>
<td>428</td>
<td>7.857143</td>
<td>0.154945</td>
</tr>
<tr>
<td>594</td>
<td>8.406593</td>
<td>0.354645</td>
</tr>
</tbody>
</table>

5 ANALYSIS AND CONCLUSION

In our encryption process[5] the following advantage are provided: the encryption is perform on binary data. All data which is under stable by the computer is finally converted
into binary bits. So it can be implemented for any data type encryption process[5].

As the key length is not fixed in this algorithm, we can take large key length for making it more complex. If the key length is assumed as m*m then the complexity of guessing is \(2^{m^2}\). Hence the complexity of the key is increase exponentially with respect to the increase of key length. In this algorithm the length of the plane text is not restricted so it can be applicable for any large file.

6 REFERENCES


