SECURED E-MAIL SYSTEM USING BASE 128 ENCODING SCHEME

Chandra Segar Thirumalai, Senthil Kumar M

1Assistant Professor (senior), SITE school, VIT University, Vellore-632014, Tamil Nadu, India.

Email: mosenkum@gmail.com

Received on: 18.10.2016

Accepted on: 11.11.2016

Abstract

In cryptography, encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it undetectable to anyone except those possessing special knowledge, usually referred to as a secret key. The result of the process is encrypted information (referred to as cipher text). The reverse process is to make the encrypted information readable again, is referred to as decryption (i.e., to make it unencrypted). Encryption can protect the confidentiality of messages by itself, but other techniques are still needed to protect the integrity and authenticity of a message. In this paper we are going to implement BASE128 encoding technique to convert 8-bit character to 7-bit character so that they can be transferred over the internet. Base128 encoding standard can be used by almost all the email clients to encode the files which are sent as attachments by Multipurpose Internet Mail Extension (MIME).

Keywords: Cipher text, Decryption, Encryption, Plaint text, Secret key.

1. Introduction

Encryption is the most effective way to achieve data security over the network. Encryption means to convert readable data in such a manner that it becomes unreadable [1]. It is used almost everywhere where any secret data is needed to be kept or transferred so that any unauthorized person can’t access that secret data. Encryption goes side by side with decryption which means to convert unreadable encrypted data to its original meaningful form. The encrypted data is referred as the cipher text and unencrypted data as the Plaintext [2].
There are two kinds of cryptosystems:

1.1. Symmetric

1.2. Asymmetric

Symmetric cryptosystems use the same key i.e. a secret key to encrypt and decrypt a message, and asymmetric cryptosystems use only one key which is a public key to encrypt a message and a different key which is a private key to decrypt it, or vice versa. Some of the various popular cryptography algorithms are IDEA (International Data Encryption Algorithm), RC4, DES (Digital Encryption Standard), Unix Crypt, PGP, RSA, Diffie-Hellman.
This paper has two major purposes. The first purpose is to define some of the terms and concepts behind basic cryptographic methods, and to offer a way to implement encryption and decryption on integer value. The second is to provide some real examples of cryptography in today’s world. Among all the encryption algorithms, BASE64 and BASE128 technique is one of the most widely used and has gained immense popularity in very short span of time because of the high security provided by these algorithms and are easy to implement and use[3]. A BASE128 string consists of the 128 characters, 0-9, A-Z, a-z, and various special characters and can be determined by up to two ‘=’ characters for padding. Each four BASE128 string will always be an exact multiple of four characters long. In programs that decode from BASE128 will ignore characters that are not valid; but if a”=” character is found, it should indicate the end of data.

2. Related Works

Kejing He et.al [3], proposed a lossless base62 compressed encoding mechanism that only use alphanumeric characters and a simple substitution encryption is incorporated to represent the original data. Burrows-Wheeler Transform (BWT) is used for compressing data before it is transferred to base62 encoder with move to-front transform, range encoding and zero length encoding which can provide basic security over the network. Allam Mousa et.al [4], invited the proposal where various data types were analysed and the role of the data type was also emphasized. The encryption key length and data file size is directly related to speed of encryption or decryption time. Data type is also important since image data requires larger time to be processed than text (small data size) or sound data mainly due to the larger file size. Vinod Shokeen, et.al [5], proposed a fast and secure encryption algorithm using translation, substitution mapping and transposing operations which is a simple, direct mapping algorithm using matrix and arrays. The advantage of proposed symmetric encryption technique is that they are simpler, faster and security level is high. Rakesh Agrawal, et.al [6] proposed a more serious problem is to estimation exposure i.e., the expected gap between two encrypted values is proportional to the gap between the corresponding plaintext values. The plaintext distribution nature can be referred from the encrypted values. The use of standard encryption techniques for this purpose results in degradation in the performance of the database system. Tinkyuan Nie, et.al [7] proposed that Encryption algorithm plays an important role in the field of information security. The performances of two symmetric key encryption algorithms are evaluated. These are CAST and RC5 algorithm. These algorithms are commonly used for network data encryption. Vivek K Goyal, [8], proposed the advantage of transform coding is a
reduction in memory requirements for entropy coding. The possible advantage is to reduce the memory requirement further than conventional transform coding by using a single common scalar entropy code book for all components.

3. Proposed System

Base64 algorithm is considered as the weakest encoding standard although this BASE64 algorithm can easily be reverse by implementing some of the few simple steps. Base32 representation takes roughly 20% more space than BASE64 therefore BASE64 is more useful tool in compare with BASE32[9]. There is also BASE16 algorithm which is generally used for memory addresses not for encoding purpose because data is not more secure[10]. So now we are trying to modify these algorithms i.e. BASE16, BASE32 and BASE64 with extra features so that we can overcome with the drawbacks of previous algorithms. This will make data more secure and efficient over the network [10].

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length of Base64 and Base32 notations as percentage of binary data</strong></td>
</tr>
<tr>
<td><strong>Base64</strong></td>
</tr>
<tr>
<td>8-bit</td>
</tr>
<tr>
<td>7-bit</td>
</tr>
</tbody>
</table>

3.1 Solution Methodology

Implementation of BASE128 algorithm is described via example by following steps which are as follows:

<table>
<thead>
<tr>
<th>Table 2 BASE128 Alphabet Chart.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>&quot;A&quot;</td>
</tr>
<tr>
<td>&quot;L&quot;</td>
</tr>
<tr>
<td>&quot;W&quot;</td>
</tr>
<tr>
<td>&quot;h&quot;</td>
</tr>
<tr>
<td>&quot;s&quot;</td>
</tr>
<tr>
<td>&quot;d&quot;</td>
</tr>
<tr>
<td>&quot;o&quot;</td>
</tr>
<tr>
<td>&quot;z&quot;</td>
</tr>
<tr>
<td>&quot;k&quot;</td>
</tr>
<tr>
<td>&quot;v&quot;</td>
</tr>
<tr>
<td>&quot;g&quot;</td>
</tr>
<tr>
<td>&quot;r&quot;</td>
</tr>
</tbody>
</table>

IJP| Dec-2016 | Vol. 8 | Issue No.4 | 21797-21806  
Page 21800
4. Encryption Algorithm:

![Encryption Flow Diagram](image)

Figure 4 Encryption flow diagram sender side.

Step1: Original input value: 647221
we took any decimal number of finite length and pairing them in order of 2 such that the above number is 64 72 21
and now converted into their binary equivalents by using any conversion chart.
The 8-bit binary equivalents of 64 72 21 are as follows:
01000000 01001000 00010101

Step2: Now all the binary equivalents obtained are concatenated or joined to obtain a large set of binary number.
01000000100010000010101

Setp3: This large set of binary number is now divided into equal sections and each section must contain only 7-bit. In this case we have 24-bit which can be divided into 7-bit sets give us 4 such set. Hence the four equal 7-bit sets are:
0100000 0100100 0000100 0000010 0000101

Step4: These 5 equal sets of 7-bit are now converted into their decimal equivalents using any conversion chat.
0100000 =32 0010010 =18 0000010=2 0000101=5

Step 5: The decimal equivalents thus obtained are changed into the character on the basis of created chart.
32=B 18=4 2=# 5=&.
Table-3: Process of encoded data.

<table>
<thead>
<tr>
<th>UNENCODED</th>
<th>8-BIT CONVERSION</th>
<th>DECIMAL CODE</th>
<th>ENCODED</th>
<th>INTERCHANGE (FIRST AND LAST CHARACTER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>01000000</td>
<td>010000</td>
<td>B</td>
<td>&amp;</td>
</tr>
<tr>
<td>72</td>
<td>01001000</td>
<td>0010010</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>00010101</td>
<td>00001010</td>
<td>2</td>
<td>#</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000101</td>
<td>&amp;</td>
<td>B</td>
</tr>
</tbody>
</table>

The integer is converted into B4#&.

Step 6: Now in last step we are interchanging the position of first character with the last one so that encrypted data will be more secure over the network.

Finally our encrypted data is “&4#B”.

![Figure-5: Base 128 encoding mechanism.](image)

Decryption Algorithm:

![Figure 6 Decryption flow diagram receiver side.](image)
Step 1: As we have the encrypted data “&4#B”. Now interchanging the position of first and the last character. So after interchanging the position we have encrypted data as “B4#&”.

Step 2: Convert the character value into decimal on the basis of chart. So after converting we have decimal values as:

B=32  4=18  #=2  &\=5

Step 3: The decimal values thus obtained are then converted into their binary equivalents using any conversion chart.

32=00100000  18=00010010  2=00000010  5=00000101

STEP 4: The first bit are truncated from each one of these binary numbers and then the 4 sets of 7-bits are combined to form one large string of binary digits.

Now, the resulting large string will be:

01000000100100001000001000100000101

STEP 5: This large string is now split into groups of 8-bits. The resulting 3 groups are:

01000000  01001000  00010000  00000101

STEP 6: Now write the corresponding decimal value for each set.

01000000= 64  01001000=72  00010000=16  00000101=5

Now, we designed our algorithm in such a way that the always last two sets when added give the value of original number 16+5 =21 which is our required decrypted data.

So, our final decrypted data is: 64 72 21.

Table-4: Process of decoded data.

<table>
<thead>
<tr>
<th>INTERCHANGE (FIRST AND LAST CHARACTER)</th>
<th>DECIMAL CODE</th>
<th>8.BIT BINARY CODE</th>
<th>TRUNCATE FIRST BIT</th>
<th>8.BIT BINARY CODE</th>
<th>DECIMAL CODE</th>
<th>ADDING LAST TWO NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>32</td>
<td>00100000</td>
<td>010000</td>
<td>01000000</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>00010010</td>
<td>061000</td>
<td>01001000</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>#</td>
<td>2</td>
<td>00000010</td>
<td>0600010</td>
<td>00100000</td>
<td>16</td>
<td>21</td>
</tr>
<tr>
<td>&amp;</td>
<td>5</td>
<td>00000101</td>
<td>0600101</td>
<td>0000101</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Figure-7: Base 128 decoding mechanism.
4. Result

Example showing how Base128 works. For the number: "10 11 13 26 02 09 27" it is encoded as "&#…s"*4=" (7 bytes). Here a 7 bytes string is encoded as 7 bytes string (base128). Here "10" is converted into "5" (7 bytes) and then the corresponding value for 5 is taken from the table is used to encode "10" …in this case it is "&". Base64: number="10 11 13 26 02 09 27" base64 encoding= "DBN/B*+'$" (10 bytes). Here in base64 encoding a 7 bytes string is encoded as 10 bytes string (So there is a loss of 3 bytes) but in base128 encoding only 0 byte is lost.

The graphs below shows encryption and decryption time in base128 versus variable key length for a small text data:

5. Conclusion

The respective algorithm has been implemented from BASE16, BASE32 and BASE64 to BASE128. The space occupied by base128 is less as compare to base64 encoding. When data is being transmitted over the network, it is easier in case of base128 as only fewer amounts of data needs to be transmitted when compared with base64. So with this algorithm data is more secure and is compressed. This algorithm is most widely used for the binary data – printable string conversion.

References


