Abstract—Encryption is used to securely transmit data in open networks. Most of the available encryption algorithms are mainly used for textual data and may not be suitable for multimedia data such as images. In this paper, we introduce a block-based transformation algorithm based on the combination of image transformation and a well-known encryption and decryption algorithm called Blowfish. The original image was divided into blocks, which were rearranged into a transformed image using a transformation algorithm, and then the transformed image was encrypted using the Blowfish algorithm. The results showed that the correlation between image elements was significantly decreased by using the proposed technique. The results also show that increasing the number of blocks by using smaller block sizes resulted in a lower correlation and higher entropy.

Index Terms—Image correlation, Image encryption, Image entropy, Permutation.

I. INTRODUCTION
Technology has changed the world dramatically. Last few decades have witnessed some great events, especially in the field of digital communication[1]. Internet has shrunk the whole world. Now it’s very easy to access the information as well as data anywhere. To prevent the data from unauthorized access, encryption technique is widely used. Encryption method try to convert original data to another data that is hard to understand and only the intended recipient will be able to understand it. Most of the available encryption algorithms are mainly used for textual data. But due to large data size and real time constrains, algorithms that are good for textual data may not be suitable for multimedia data [2]-[4]. So, we introduce a block-based transformation algorithm which tries to reduce the correlation between the neighbor pixels value to ensure the security of digital images.

In most of the natural images, the values of the neighboring pixels are strongly correlated (i.e., the value of any given pixel can be reasonably predicted from the values of its neighbors) [5]-[7]. In order to dissipate the high correlation among pixels and increase the entropy value, we propose a transformation algorithm that divides the image into blocks and then shuffles their positions before it passes them to the Blowfish encryption algorithm. By using the correlation and entropy as a measure of security, this process results in a lower correlation and a higher entropy value when compared to using the Blowfish algorithm alone, and thus improving the security level of the encrypted images. There are two main keys to increase the entropy; the variable secret key of the transformation process and the variable secret key of the Blowfish algorithm.

II. THE PROPOSED TECHNIQUE
A. Description of the Transformation Algorithm
The transformation technique works as follows: the original image is divided into a random number of blocks that are then shuffled within the image. The generated (or transformed) image is then fed to the Blowfish encryption algorithm. The main idea that is used in this transformation algorithm is that an image can be viewed as an arrangement of blocks. The intelligible information present in an image is due to the correlation among the image elements in a given arrangement. This perceivable information can be reduced by decreasing the correlation among the image elements using certain transformation techniques.

The secret key of this approach is used to determine theseed. The seed plays a main role in
building the transformation table, which is then used to generate the transformed image with different random number of block sizes. The transformation process refers to the operation of dividing and replacing an arrangement of the original image.
The image can be decomposed into blocks; each one contains a specific number of pixels. The blocks are transformed into new locations. For better transformation the block size should be small, because fewer pixels keep their neighbors. In this case, the correlation will be decreased and thus it becomes difficult to predict the value of any given pixel from the values of its neighbors. At the receiver side, the original image can be obtained by the inverse transformation of the blocks. A general block diagram of the transformation method is shown below.

The transformation algorithm is presented below. It generates a transformation table that will be used then to build a newly transformed image.

ALGORITHM
CREATE_TRANSFORMATION_TABLE
1: Select Image (Bmp format) to be encryption from data store
2: Insert key of 256 bits
3: Calculate Image Pixels Value
   Horizontal Value of Pixel = PixelWidth/10
   Vertical Value of Pixel = PixelHeight/10
4: Select a Random Function to Calculate Final value for Horizontal and Vertical Pixels
   • HorizontalPixel = Select Random Value between Horizontal Value of Pixel and PixelWidth
   • VerticalPixel = Select Random Value between Vertical Value of Pixel and PixelHeight
5: Select a Variable No-Of-Pixel to store Multiple Value of HorizontalPixel and VerticalPixel
   No-Of-Pixel = HorizontalPixel * VerticalPixel
6: Using Hash Function (Here I am using SHA-1) I am generating a Seed Value. This SHA-1 will apply on 256 bits Selected Key
   Seed = SHA-1(Above Selected KEY)
7: Divide Seed into two Part equally Seed-1 and Seed-2
   • Seed-1 = First Half of Seed
   • Seed-2 = Second Half of Seed
8: If Seed-1 is Greater Then Seed-2 Then We Will Select another Variable SeedValue and assign any numeric value between 0 to 4 (Randomly Chooseable) Otherwise Value of SeedValue Variable vary between 5 to 9 (Randomly Chooseable).
9: If Variable SeedValue is Equal Between 0 to 4 thencalculate new seed value (Here we are working on ASCII value of seed).
   • Seed = Seed + (Seed-1 Mod 2) + 1
   Otherwise
   • Seed = Seed + (Seed-2 Mod 2) + 1
10: Repeat Process 8 to 9 till No-Of-Pixel/2
11: Final Output of Step 10 will represent END
CREATE_TRANSFORMATION_TABLE
**ALGORITHM**

PERFORM_TRANSFORMATION

1: For I = 0 to No-Of-Pixel -1
1.1: Get the new location of block I from the transformation table
1.2: Set block I in its new location
END PERFORM_TRANSFORMATION

Output: Transformed Image.

---

**A. Description of the Blowfish Algorithm**

Blowfish is a 64-bit symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times.

The Blowfish algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion consists of generating the initial contents of one array (the P array), namely, eighteen 32-bit sub keys, and four arrays (the S boxes), each of size 256 by 32 bits, from a key of at most 448 bits.

Data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

The Blowfish algorithm is presented below. It generates an encrypted image.

**Algorithm**

**Note:** Pi used in this algorithm is the subkey generated from 448-bit key. F function is basically a process of applying substitution and the permutation process.

---

1: Take the transformed image generated in the transformation process.
2: Divide the image into blocks containing 64 bits in each block.
3: For each block, do
   - Divide block into two 32-bit halves: xL, xR.
   - Then, for i = 1 to 16:
     - xEL = xR
     - xER = [F(xR) and F(Pi)] XOR xL
     - Swap xL and xR
4: Repeat the step 3 for each block and finally we get encrypted image.

---

**III. EXPERIMENTS**

Results are evaluated on the following parameters.

1. **NORMALISED CROSS CORELATION**

   Normalized Cross Correlation (NCC) has been commonly used as a metric to evaluate the degree of similarity or dissimilarity between two compared images. The main advantage of the normalized cross correlation over the cross correlation is that it is less sensitive to linear
changes in the amplitude of illumination in the two compared images.

\[
C_{ab} = \frac{\frac{1}{r \cdot c} \sum_{i=1}^{r} \sum_{j=1}^{c} (A_{i,j} - \bar{A})(B_{i,j} - \bar{B})}{\sqrt{\left(\frac{1}{r \cdot c} \sum_{i=1}^{r} \sum_{j=1}^{c} (A_{i,j} - \bar{A})^2\right) \cdot \left(\frac{1}{r \cdot c} \sum_{i=1}^{r} \sum_{j=1}^{c} (B_{i,j} - \bar{B})^2\right)}}
\]

\(A_{i,j}\) and \(B_{i,j}\) are the pixels in the \(i\)th row and \(j\)th column of \(A\) and \(B\) respectively and \(r, c\) represent the no. of rows and columns in the image.

(4) ENTROPY-
In information theory, entropy is a measure of the uncertainty in a random variable. In this context, the term usually refers to the entropy, which quantifies the expected value of the information contained in a message. Entropy is typically measured in bits, nats or bans. Shannon entropy is the average unpredictability in a random variable, which is equivalent to its information content.

\[
h = -\sum(p_i \log_2 p_i)
\]

Where \(p_i\) is the frequency of intensity level \(i\) in the image. The maximum \(h\) an 8–bit image can attain is 8.

In order to evaluate the impact of the number of blocks on the correlation and entropy, three different cases were tested. The number of blocks and the block sizes for each case are shown in Table I.

Each case produces three output images; (a) a ciphered image using the Blowfish algorithm, (b) a transformed image using the proposed algorithm, and (c) a ciphered image using the proposed algorithm followed by the Blowfish algorithm. For the rest of this paper, we use image A, image B, image C, and image D to denote the original image, the ciphered image using the Blowfish algorithm, the transformed image, and the ciphered image using the proposed algorithm followed by the Blowfish algorithm respectively.

**Case 1:** The image is decomposed into 10 pixels \(\times\) 10 pixels blocks. Fig.1. shows the resulted images.

Fig.1. Results of encryption by using 10 pixels \(\times\) 10 pixels blocks. (a) Original image. (b) Encrypted image using Blowfish. (c) Transformed image. (d) Encrypted image using transformation followed by the Blowfish algorithm.

The correlation and entropy results of this case are summarized in Table I.

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Number of blocks</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30(\times)30</td>
<td>10 pixels (\times) 10 pixels</td>
</tr>
<tr>
<td>2</td>
<td>60(\times)60</td>
<td>5 pixels (\times) 5 pixels</td>
</tr>
<tr>
<td>3</td>
<td>100(\times)100</td>
<td>3 pixels (\times) 3 pixels</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.933</td>
<td>0.035</td>
<td>0.843</td>
<td>0.034</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.936</td>
<td>0.107</td>
<td>0.844</td>
<td>0.038</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.919</td>
<td>0.032</td>
<td>0.748</td>
<td>0.013</td>
</tr>
<tr>
<td>Opposite</td>
<td>0.916</td>
<td>0.034</td>
<td>0.745</td>
<td>0.009</td>
</tr>
<tr>
<td>Average</td>
<td>0.926</td>
<td>0.052</td>
<td>0.795</td>
<td>0.023</td>
</tr>
</tbody>
</table>

Table I: Results of Correlation and Entropy values of Case 1.
Case 2: The image is decomposed into 5 pixels × 5 pixels blocks. Fig. 2 shows the resulted images.

Fig.2. Results of encryption by using 10 pixels × 10 pixels blocks. (a) Original image. (b) Encrypted image using Blowfish. (c) Transformed image. (d) Encrypted image using transformation followed by the Blowfish algorithm.

The correlation and entropy results of this case are summarized in Table II.

Table II: Results of Correlation and Entropy values of Case 2.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.9325</td>
<td>0.0346</td>
<td>0.6289</td>
<td>0.0129</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9362</td>
<td>0.1073</td>
<td>0.6265</td>
<td>0.0034</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9186</td>
<td>0.0331</td>
<td>0.4145</td>
<td>0.0014</td>
</tr>
<tr>
<td>Opposite Diagonal</td>
<td>0.9156</td>
<td>0.0337</td>
<td>0.4146</td>
<td>0.0049</td>
</tr>
<tr>
<td>Average</td>
<td>0.9257</td>
<td>0.0519</td>
<td>0.5211</td>
<td>0.0056</td>
</tr>
<tr>
<td>Entropy value</td>
<td>2.4305</td>
<td>4.799</td>
<td>2.4305</td>
<td>5.5281</td>
</tr>
</tbody>
</table>

Case 3: The image is decomposed into 3 pixels × 3 pixels blocks. Fig. 3 shows the resulted images.

Fig.3. Results of encryption by using 10 pixels × 10 pixels blocks. (a) Original image. (b) Encrypted image using Blowfish. (c) Transformed image. (d) Encrypted image using transformation followed by the Blowfish algorithm.

The correlation and entropy results of this case are summarized in Table III.

Table III: Results of Correlation and Entropy values of Case 3.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.9325</td>
<td>0.0346</td>
<td>0.6289</td>
<td>0.0129</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9362</td>
<td>0.1073</td>
<td>0.6265</td>
<td>0.0034</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9186</td>
<td>0.0331</td>
<td>0.4145</td>
<td>0.0014</td>
</tr>
<tr>
<td>Opposite Diagonal</td>
<td>0.9156</td>
<td>0.0337</td>
<td>0.4146</td>
<td>0.0049</td>
</tr>
<tr>
<td>Average</td>
<td>0.9257</td>
<td>0.0519</td>
<td>0.5211</td>
<td>0.0056</td>
</tr>
<tr>
<td>Entropy value</td>
<td>2.4305</td>
<td>4.799</td>
<td>2.4305</td>
<td>5.5281</td>
</tr>
</tbody>
</table>

IV. RESULTS AND DISCUSSION

The above cases show that using the transformation algorithm followed by the Blowfish algorithm resulted in a lower correlation and a higher entropy compared to using the Blowfish alone. Dividing the image into a larger number of blocks made the performance even better. The results showed that the correlation was reduced even further and the entropy was increased as blocks size is decreased.

Table IV summarizes the results of the different cases.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Number of blocks</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average correlation</td>
<td>30 × 30</td>
<td>0.0257</td>
<td>0.0519</td>
<td>0.7952</td>
<td>0.0234</td>
</tr>
<tr>
<td></td>
<td>60 × 60</td>
<td></td>
<td></td>
<td>0.6881</td>
<td>0.0092</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td></td>
<td></td>
<td>0.5211</td>
<td>0.0056</td>
</tr>
<tr>
<td>Entropy</td>
<td>30 × 30</td>
<td>2.4305</td>
<td>4.799</td>
<td>2.4305</td>
<td>5.2305</td>
</tr>
<tr>
<td></td>
<td>60 × 60</td>
<td></td>
<td></td>
<td>2.4305</td>
<td>4.7357</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td></td>
<td></td>
<td>2.4305</td>
<td>5.5281</td>
</tr>
</tbody>
</table>
V. CONCLUSION
In this paper a simple and strong method has been proposed for image security using a combination of block-based image transformation and encryption techniques. The cases showed that the correlation was decreased when the proposed algorithm was applied to them before the Blowfish algorithm. Experimental results of the proposed technique showed that an inverse relationship exists between number of blocks and correlation, and a direct relationship between number of blocks and entropy. When compared to many commonly used algorithms, the proposed algorithm resulted in the best performance; the lowest correlation and the highest entropy.

VI. REFERENCES