Image Encryption by Redirection & Cyclical Shift

Dr. Artyom M. Grigoryan

Bryan A. Wiatrek

Dr. Sos S. Again

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Engineering
Department of Electrical & Computer Engineering
May 2015

Agenda

- Abstract
- Redirecting Image
- Image Encryption
- Image Decryption
- Color Images
- Correlation of Adjacent Pixel in Encrypted Image
- Experimental Results
- Conclusion
- References

Abstract

- A novel method for encrypting and decrypting images, both grayscale and color, without the lost of information, and using private keys of varying lengths will be presented
  - Based on the concept of the tensor representation of an image and splitting two-dimensional (2-D) discrete Fourier transform (DFT) by one-dimensional (1-D) DFTs of signals from the tensor representation, or transform [1,2,3]
  - Iterations of redirecting an image and sub image parts followed by a cyclical shift makes for an encrypted image that is uncorrelated

Redirecting Image

- Here we consider the concept of the redirected image, by using the tensor representation of an image in the three-dimension (3-D) space [4,5,6,7,8,9]
  - Two dimensions of the space are for frequency, and one dimension is for time
-
- For simplicity of our calculations, we describe the $N \times N$ case, when $N = 2^r; r > 1$

Redirecting Image Cont.

- Let $f = \{f_{n,m}\}$ be the distance image defined on the square Cartesian lattice $X = X_{N,N} = \{(n, m); n, m = 0: (N-1)\}$. The image is considered to be extended periodically on the plane
-
    - In tensor representation, the image is described by a set of $\frac{3N}{2}$ splitting-signals of length N each,

$$\{f_{p,s,0}, f_{p,s,1}, f_{p,s,2}, \dots, f_{p,s,N-1}\},$$

which are generated by the set of frequencies (p, s); which can be taken as

$$J_{N,N} = \{(0,1), (1,1), (2,1), \dots, (N-1,1)\} \cup \{(1,0), (1,2), (1,4), \dots, (1, N-2)\}$$

Redirecting Image Cont.

- Frequency-generators (p, s) are pairs of coprime numbers of type $(p, 1)$ and $(1,2s)$, where $p = 0:(N-1)$ and $s = 0:\left(\frac{N}{2}-1\right)$. Each component $f_{p,s,t}$ of the splitting-signal is the sum of the image at points of the set

$$V_{p,s,t} = \{(n,m); (np+ms)\mathrm{mod}N = t\}$$

- Thus,

$$f_{p,s,t} = \sum\{f_{n,m}; (n,m) \in V_{p,s,t}\}, \qquad t = 0:(N-1)$$

Redirecting Image Cont.

- The set $V_{p,s,t}$ contains N points on the lattice each, and the lattice $X_{N,N}$ can be reordered in such a way, that the summation of each components of the splitting signal will be performed only along the rows (or columns). Thus, we can map uniquely the original lattice $X_{N,N}$ into another one $Y_{N,N}$ of the same size
-

$$\mathcal{M}(p,s): X_{N,N} \rightarrow Y_{N,N} = \begin{cases} S_{p,s,0} \\ S_{p,s,1} \\ \dots \\ S_{p,s,N-1} \end{cases}$$

whose rows (or columns) are the sets $V_{p,s,t},\ \ t = 0:(N-1)$

Redirecting Image Cont.

- Given the generator $(p, s) \in J_{N,N}$ and an image, f, of size $N \times N$ (where $N = 2^r; r > 1$), the redirected image, f_φ, for when $p \geq s$ can be calculated using

$$f_{\varphi_{(p,s)}}(l, k) = X\big((-ks)\mathrm{mod}N, (l + ks)\mathrm{mod}N\big), \text{ where } l, k = 0: (N-1),$$

for when $p < s$

- $$f_{\varphi_{(p,s)}}(l, k) = X\big((l - ks)\mathrm{mod}N, (kp)\mathrm{mod}N\big), \text{ where } l, k = 0: (N-1)$$

- The redirected image, f_φ, is a permutation of the original image, which is redirected cyclically along the direction defined by the angle $\varphi = \varphi(p, s) = \tan\left(\frac{s}{p}\right)$, or $\pi - \tan\left(\frac{s}{p}\right)$

Image Encryption

- Original image is size $N \times N$, where $N = 2^r, r > 4$

- Step 1: Generate secret key

$$\mathbf{Key} = \mathbf{Key}(k) = \{k, \mathbf{G}(p), \mathbf{V}(2)\}$$

  - where k is the number of iterations, which us consider to be $k = r - 3$
  - $\mathbf{G}(p)$ is the set of $\frac{(4^k-1)}{3} + 1$ random integers, numbers p from 1 to $N - 1$, which are used as generators $(p, 1)$ or $(1, p)$ for redirecting image
  - set $\mathbf{V}(2)$ contains k random vectors $v_n = (x_n, y_n)$ which will be used for cyclically shifting the image on different stages of the encryption.

$$\text{length}(\mathbf{Key}) = L(k) = 1 + \frac{(4^k - 1)}{3} + 1 + 2\text{k}$$

Image Encryption Cont.

- Step 2: Take the first number from the set $\mathbf{G}(p)$ and redirect the original image as

$$f_{n,m} \rightarrow f_{\varphi_0}(l,k), \qquad \varphi_0 = \varphi(p_0,1), \qquad l,k = 0:(N-1)$$

- Step 3: Take the first vector $v_0 = (x_0, y_0)$ from the set $\mathbf{V}(2)$ and shift cyclically the redirected image

$$f_{\varphi_0}(l,k) \rightarrow f_{\varphi_0;v_0}(l,k) = f_{\varphi_0}\big((l+x_0)\mathrm{mod}N, (k+y_0)\mathrm{mod}N\big),$$
$$l,k = 0:(N-1)$$

Image Encryption Cont.

$$[f_{\varphi_0;v_0}(l,k)] = \begin{bmatrix} [f_{n,m}^{(1)}] & f_{n,m}^{(2)} \\ f_{n,m}^{(3)} & f_{n,m}^{(4)} \end{bmatrix}, \qquad \left(n,m = 0:\left(\frac{N}{2}-1\right)\right)$$

- Stage 5: Take the next four numbers $p = p_1, p_2, p_3,$ and p_4 from the set $\mathbf{G}(p)$ and redirect the image parts as
-

$$f_{n,m}^{(1)} \rightarrow f_{\varphi_1}^{(1)}(l_1,k_1) = f_{\varphi_{(p_1),1}}^{(1)}(l_1,k_1) \qquad f_{n,m}^{(2)} \rightarrow f_{\varphi_2}^{(2)}(l_1,k_1) = f_{\varphi_{(p_2),1}}^{(2)}(l_1,k_1)$$

$$f_{n,m}^{(3)} \rightarrow f_{\varphi_3}^{(3)}(l_1,k_1) = f_{\varphi_{(p_3),1}}^{(3)}(l_1,k_1) \qquad f_{n,m}^{(4)} \rightarrow f_{\varphi_4}^{(4)}(l_1,k_1) = f_{\varphi_{(p_4),1}}^{(4)}(l_1,k_1)$$

$$l_1, k_1 = \left(\frac{N}{2}-1\right)$$

Image Encryption Cont.

- Stage 6: Take the next vector $v_1 = (x_1, y_1)$ from the set $\mathbf{V}(2)$ and shift cyclically the new image

$$[f_{\varphi;v_1}(l,k)] = \begin{bmatrix} \left[f_{\varphi_1}^{(1)}(l_1,k_1)\right] & \left[f_{\varphi_2}^{(2)}(l_1,k_1)\right] \\ \left[f_{\varphi_3}^{(3)}(l_1,k_1)\right] & \left[f_{\varphi_4}^{(4)}(l_1,k_1)\right] \end{bmatrix}, \qquad \left(l_1, k_1 = \left(\frac{N}{2} - 1\right)\right)$$

as

$$f_{\varphi_0;v_0}(l,k) \to f_{\varphi;v_1}(l,k) = f_{\varphi;v_0}\big((l + x_1)\mathrm{mod}N, (k + y_1)\mathrm{mod}N\big),$$
$$l, k = 0 : (N-1)$$

the parameter φ is the vector parameter $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$

Image Encryption Cont.

Stage 7: Divide the obtained image by 16 parts of size $\frac{N}{2} \times \frac{N}{2}$ each,

- $$[f_{\varphi_0;v_0}(l,k)] = \begin{bmatrix} \left[f_{n,m}^{(01)}\right] & \left[f_{n,m}^{(02)}\right] & \left[f_{n,m}^{(03)}\right] & \left[f_{n,m}^{(04)}\right] \\ \left[f_{n,m}^{(05)}\right] & \left[f_{n,m}^{(06)}\right] & \left[f_{n,m}^{(07)}\right] & \left[f_{n,m}^{(08)}\right] \\ \left[f_{n,m}^{(09)}\right] & \left[f_{n,m}^{(10)}\right] & \left[f_{n,m}^{(11)}\right] & \left[f_{n,m}^{(12)}\right] \\ \left[f_{n,m}^{(13)}\right] & \left[f_{n,m}^{(14)}\right] & \left[f_{n,m}^{(15)}\right] & \left[f_{n,m}^{(16)}\right] \end{bmatrix}, \qquad \left(n,m = 0\colon\left(\frac{N}{4} - 1\right)\right)$$

and redirect each part by one of the random generators $(p_k, 1), p_k \in \mathbf{G}(p)$. After that, put these parts back into the image $N \times N$ and shift cyclically the new image by the next vector $v_2 = (x_2, y_2)$ of the set $\mathbf{V}(2)$.

Image Encryption Cont.

- Step 8: Continue the process of partitioning the image and redirecting its small parts, following with cyclical shifting of the image, until the parts are of size 16×16 each.
- Step 9: Take the last number p from the set $\mathbf{G}(p)$ and redirect the obtained image, as

$$o(n,m) \to o_{\varphi}(l,k), \qquad \varphi = \varphi(p,1), \qquad l,k = 0{:}\,(N-1)$$

  - Last step added to entangle the encrypted image more, and to remove the "boundaries" of all blocks 16×16, in case such boundaries can be found, which is highly unlikely

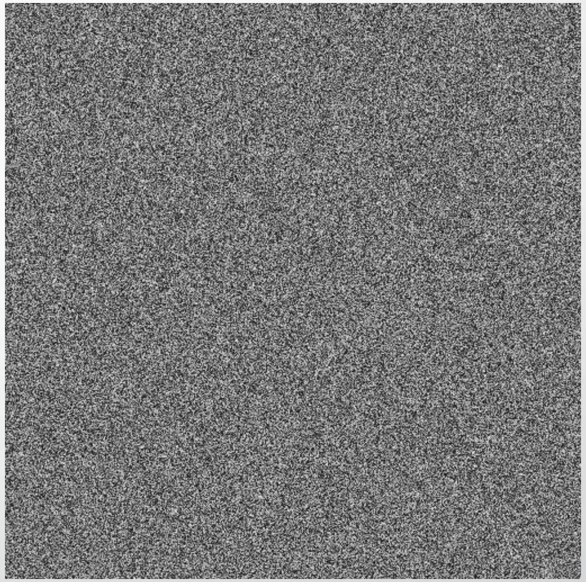
Image Encryption Cont.

Lena Image 512×512

Six standard stages: Redirect image or sub image followed by cyclical shift of image

One nonstandard stage: Redirection of image, last step of encryption

Encrypted Image: stage 6+1



During the encryption process, the partitions of the image are encrypted by parts $2^n \times 2^n$, where $n = 9,8,7,6,5,4$, and again 9 on the last stage of encryption

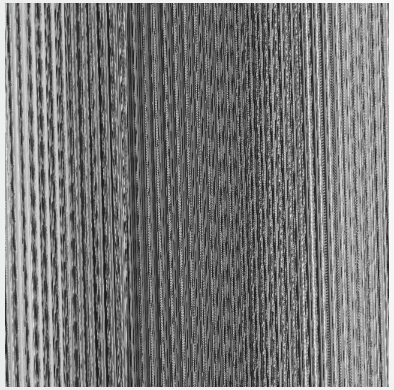
We denote this partition by the vector $P = P(512) = (9,8,7,6,5,4,9)$

The image encryption general case for $N = 2^r$, when $r > 8$, assumes the partition $P = (r, r-1, r-2, \dots, 4, r)$ is used

Image Encryption Cont.

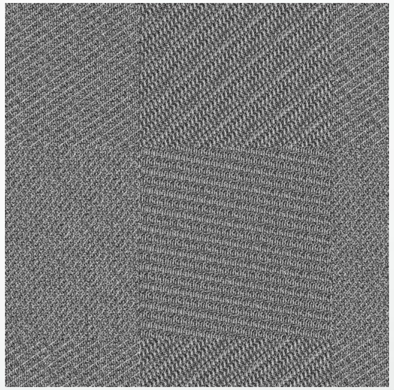
Original Image
$(512 \times 512 \times 1)$

Stage 1 Encryption
Block Size: 512



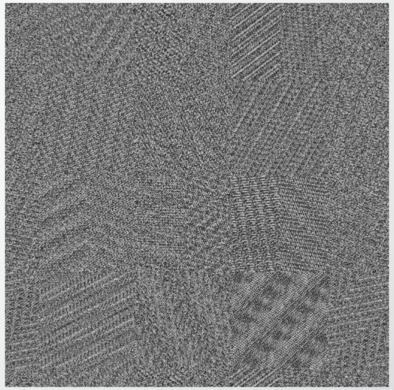
Redirection & Cyclical Shift

Stage 2 Encryption
Block Size: 256



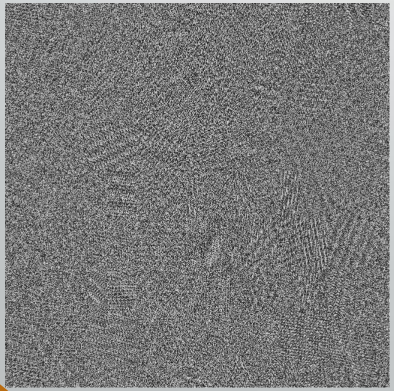
Redirection & Cyclical Shift

Stage 3 Encryption
Block Size: 128



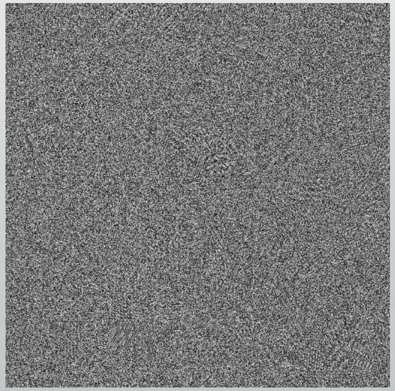
Redirection & Cyclical Shift

Stage 4 Encryption
Block Size: 64



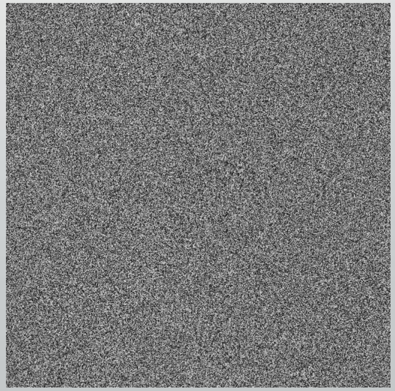
Redirection & Cyclical Shift

Stage 5 Encryption
Block Size: 32



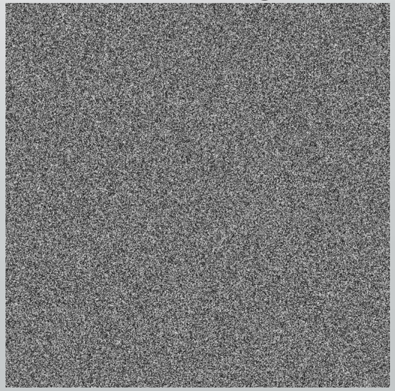
Redirection & Cyclical Shift

Stage 6 Encryption
Block Size: 16



Redirection & Cyclical Shift

Stage 7 Encryption
Block Size: 512



Redirection

Image Decryption

- Each stage in the encryption algorithm is reversible, therefore the decryption of the image is performed in the reverse order
  - User has to load the encrypted image (of size $N \times N$, where $N = 2^r, r > 4$) and use the same key that was used during encryption
-
- Step 1: Read encryption key. The values of p from the set $\mathbf{G}(p)$, as well as vectors v_n from $\mathbf{V}(2)$ will be taken consequently from these sets starting from their last points

Image Decryption Cont.

$o_\varphi(l,k)$, where $l,k = 0:(N-1)$

- Step 3: Perform image transposition,

$$o_\varphi(l,k) \rightarrow o(m,n), \qquad \varphi = \varphi(p,1), \qquad k,l = 0:(N-1)$$

-
- Step 4: Take the last number p from the set $\mathbf{G}(p)$ and redirect back the image

$$o_\varphi(k,l) \rightarrow o(m,n), \qquad \varphi = \varphi(p,1), \qquad k,l = 0:(N-1)$$

Image Decryption Cont.

$$o(m,n) \rightarrow o(n,m), \text{where } n,m = 0:(N-1)$$

Process to revert back to an original image after redirection

$$f_{n,m} \rightarrow Redirect: f_{\varphi_0}(l,k), \rightarrow Transposition: f_{\varphi_0}(k,l) = T(k,l) \rightarrow$$
$$Redirect: T_{\varphi_0}(k,l) = f_{m,n} \rightarrow Transposition:\ f_{n,m}$$

- Step 6: Take the last vector $v_{N-1} = (x_{N-1}, y_{N-1})$ from the set $\mathbf{V}(2)$ and reverse cyclically shift the transposed image,

$$o(n,m) \rightarrow [f_{v_{N-1}}(l,k)] = f\big((l - x_{N-1})\text{mod}N, (k - y_{N-1})\text{mod}N\big)\quad l,k = 0:(N-1)$$

Image Decryption Cont.

- Step 7: Divide the obtained image by parts of size 16×16 each,

$$[f(l,k)] = \begin{bmatrix} \left[f_{l_1,k_1}^{(01)}\right] & \left[f_{l_1,k_1}^{(02)}\right] & \cdots & \left[f_{l_1,k_1}^{(L_1)}\right] \\ \left[f_{l_1,k_1}^{(L_1+1)}\right] & \left[f_{l_1,k_1}^{(L_1+2)}\right] & \cdots & \left[f_{l_1,k_1}^{(2L_1)}\right] \\ \cdots & \cdots & \cdots & \cdots \\ \left[f_{l_1,k_1}^{(M_1L_1+1)}\right] & \left[f_{l_1,k_1}^{(M_1L_1+1)}\right] & \cdots & \left[f_{l_1,k_1}^{(L_1L_1)}\right] \end{bmatrix}, \qquad (l_1, k_1 = 0{:}15),$$

$$L_1 = \frac{N}{16}, \qquad M_1 = L_1 - 1$$

Image Decryption Cont.

- Step 8: Perform transposition of each part individually,

$$[f(l,k)^{[*]}] \rightarrow \begin{bmatrix} [f_{k_1,l_1}^{(01)}] & [f_{k_1,l_1}^{(02)}] & \cdots & [f_{k_1,l_1}^{(L_1)}] \\ [f_{k_1,l_1}^{(L_1+1)}] & [f_{k_1,l_1}^{(L_1+2)}] & \cdots & [f_{k_1,l_1}^{(2L_1)}] \\ \cdots & \cdots & \cdots & \cdots \\ [f_{k_1,l_1}^{(M_1L_1+1)}] & [f_{k_1,l_1}^{(M_1L_1+1)}] & \cdots & [f_{k_1,l_1}^{(L_1L_1)}] \end{bmatrix}, \qquad (l_1, k_1 = 0{:}15),$$

- $$L_1 = \frac{N}{16}, \qquad M_1 = L_1 - 1,$$

and redirect each part back by the corresponding generator $(p_k), p_k \in \mathbf{G}(p)$. After that, transpose each part individually again, place back these parts into the image $N \times N$, and reverse cyclically shift the new image by the vector $-v_k = (-x_k, -y_k)$, which is the last vector of $\mathbf{V}(2)$

- Step 9: Continue the process of partitioning the image, and having each part transposed, redirected back, transposed again, followed by the image composed from these parts that has been cyclically shifted back. The partitions are by parts $32 \times 32, 64 \times 64, \ldots,$ until the next parts would be size $\frac{N}{2} \times \frac{N}{2}$

Image Decryption Cont.

- Step 10: Transpose the obtained image,

$$f(l,k)^* = f(k,l), \qquad \big(k,l = 0{:}\,(N-1)\big)$$

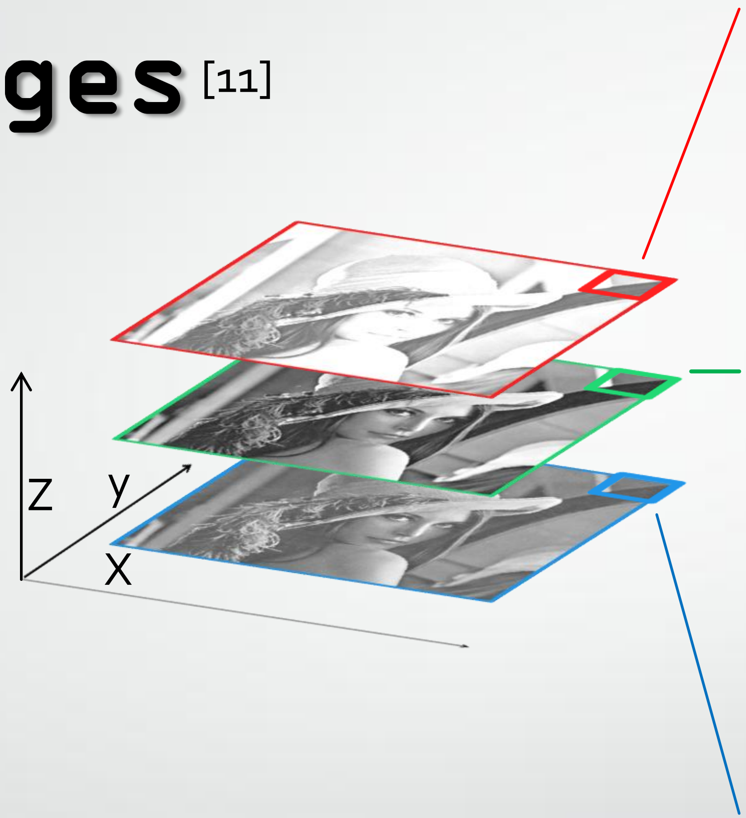
- Step 11: Redirect the transposed image using the first number from the set $\mathbf{G}(p)$

$$f(k,l) \to f_{\varphi_0}(k,l), \qquad \varphi_0 = \varphi(p_0,1), \qquad \big(k,l = 0{:}\,(N-1)\big)$$

- Step 12: Perform the final transposition of the image to return to the original image, $f_{n,m}$

$$f_{\varphi_0}(k,l)^* \to f_{\varphi_0}(l,k) = f_{n,m}, \qquad \big(k,l = 0{:}\,(N-1)\big)$$

Color Images[11]



211	224	229	231	234	230	221	200
211	224	229	231	234	230	221	200
211	224	229	231	234	230	221	200
211	224	229	231	234	230	221	200
211	224	229	231	234	230	221	200
211	219	216	212	208	198	172	135
195	197	187	171	156	141	118	101
183	165	146	122	113	96	99	90

107	139	144	147	149	148	130	99
107	139	144	147	149	148	130	99
107	139	144	147	149	148	130	99
107	139	144	147	149	148	130	99
107	139	144	147	149	148	130	99
107	113	122	120	111	93	72	47
97	99	94	85	63	48	35	35
84	62	53	44	41	24	29	21

104	131	129	126	123	122	110	90
104	131	129	126	123	122	110	90
104	131	129	126	123	122	110	90
104	131	129	126	123	122	110	90
104	131	129	126	123	122	110	90
101	97	116	103	102	87	81	81
100	92	95	97	77	75	71	68
86	80	74	71	81	66	74	64

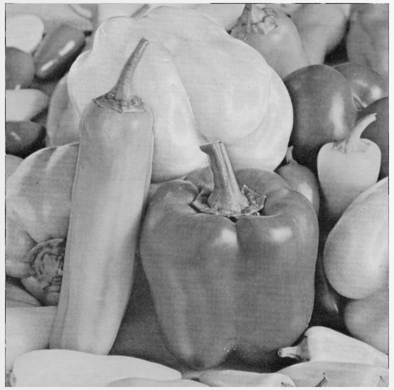
A RGB (red, green, and blue) digital image can be broken down into three grayscale digital images, thus the encryption algorithm can be applied to each image separately (combining the three grayscale images back in the same order will return the encrypted color image)

The same decryption algorithm can be applied, too, by separating the encrypted color image into its three Corresponding encrypted grayscale images, and applying the decryption algorithm to each grayscale image

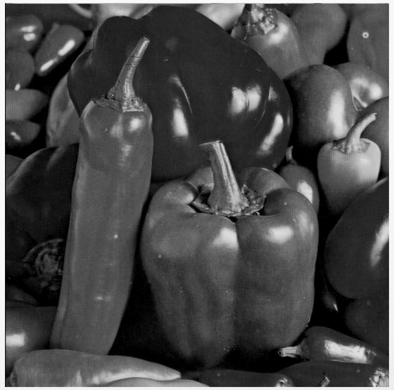
Color Images Cont.[11]

Original Color 512 × 512

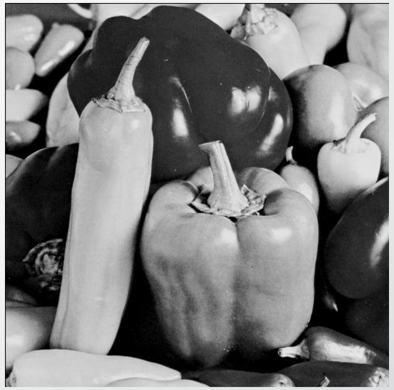
Original Red



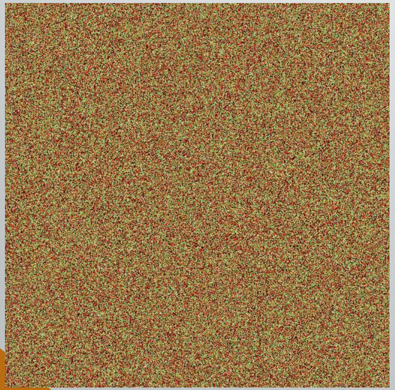
Original Green



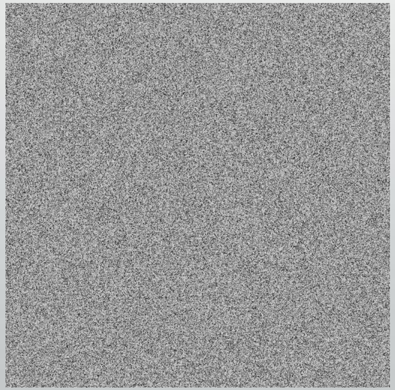
Original Blue



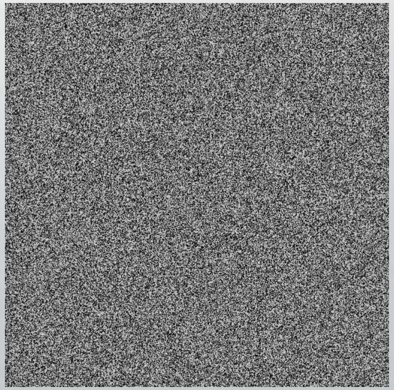
Encrypted Color



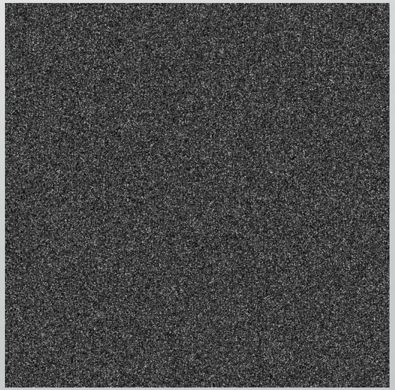
Encrypted Red



Encrypted Green



Encrypted Blue



Correlation of Adjacent Pixels in Encrypted Image

- Encryption strength was tested by calculating the correlation coefficients between pairs of horizontally, vertically, and diagonally adjacent pixels for encrypted images, and compared with the originals.
- Correlation coefficients were calculated randomly by selecting R random pairs of two adjacent pixels and using the following discrete formulas, where x and y are grayscale values of the two adjacent pixels in the image [10]

$$E(x) = \frac{1}{R}\sum_{i=1}^{R} x_i, \qquad D(x) = \frac{1}{R}\sum_{i=1}^{R} \left(x_i - E(x)\right)^2$$

- $$\operatorname{cov}(x,y) = \frac{1}{R}\sum_{i=1}^{R} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \qquad \text{where } R = 2N - 2(N-2)$$

$$r_{xy} = \frac{\operatorname{conv}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

- Correlation coefficients for color images was done by finding the correlation coefficient for each of the grayscale parts using the same R, random pairs. Then the resulting horizontal, vertical, and diagonal absolute value correlation coefficients are averaged together to produce the final horizontal, vertical, and diagonal correlation coefficient values

Experimental Results

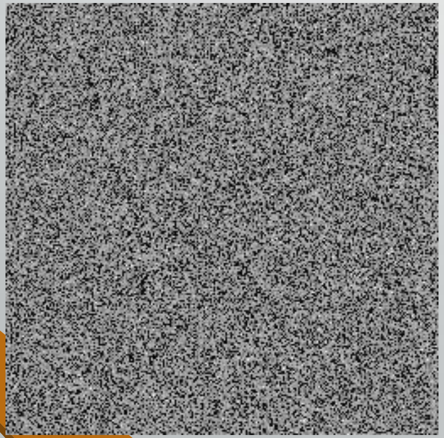
Cameraman 256×256

Lena 512×512

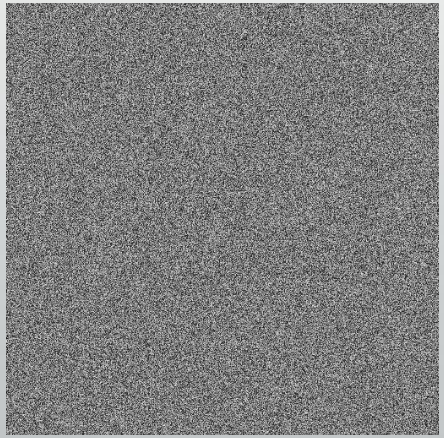
Barbara 512×512

Man 1024×1024

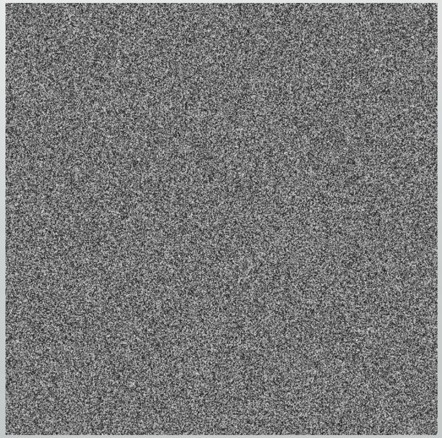
Encrypted Cameraman



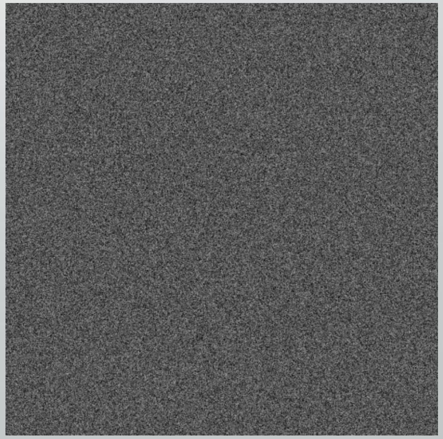
Encrypted Lena



Encrypted Barbara



Encrypted Man



Experimental Results Cont.

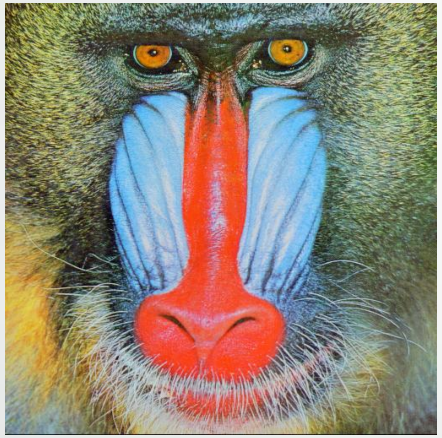
Correlation Coefficients for grayscale sample images

Image Name	Size	Original Image			Encrypted Image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Cameraman	256 x 256 x 1	0.93336	0.95933	0.90793	0.00293	-0.00017	-0.00173
Lena	512 x 512 x 1	0.97183	0.98491	0.96843	-0.00069	-0.00008	-0.00131
Barbara	512 x 512 x 1	0.89494	0.95875	0.90515	-0.00103	-0.00019	0.00064
Man	1024 x 1024 x 1	0.97744	0.98126	0.96671	-0.00006	0.00132	0.00007

Experimental Results Cont.

Tree 256 × 256

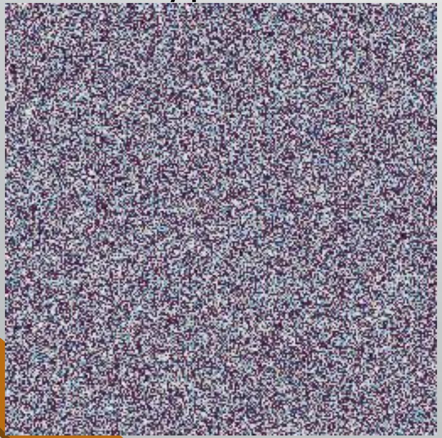
Mandrill 512 × 512



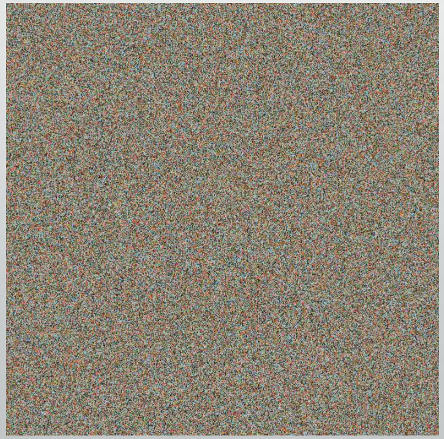
Peppers 512 × 512

Stockton 1024 × 1024

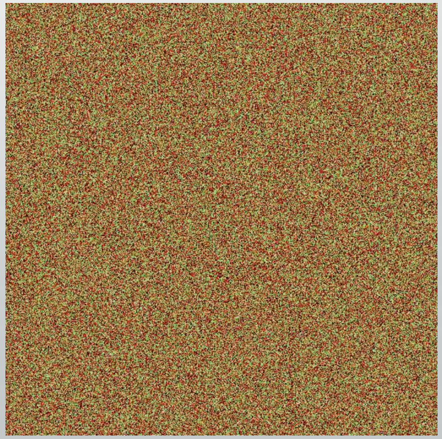
Encrypted Tree



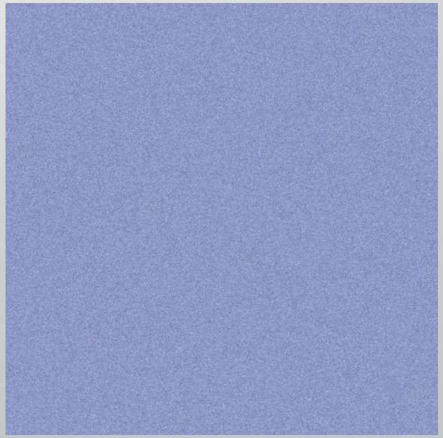
Encrypted Mandrill



Encrypted Peppers



Encrypted Stockton



Experimental Results Cont.

Correlation Coefficients for color sample images

Image Name	Size	Original Image			Encrypted Image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Tree	256 x 256 x 3	0.96365	0.94600	0.92757	0.00076	0.00326	0.00245
Mandrill	512 x 512 x 3	0.89819	0.83970	0.80960	0.00075	0.00305	0.00169
Peppers	512 x 512 x 3	0.97325	0.97138	0.95799	0.00220	0.00093	0.00130
Stockton	1024 x 1024 x 3	0.77014	0.75911	0.72619	0.00085	0.00025	0.00040

Experimental Results Cont.

Encryption & decryption timing for grayscale and color sample images

Image Name	Size	Total Encryption Time (seconds)	Total Decryption Time (seconds)	Total Time (seconds)
		Grayscale		
Cameraman	256 x 256 x 1	0.01563	0.03125	0.04688
Lena	512 x 512 x 1	0.06250	0.10938	0.17188
Barbara	512 x 512 x 1	0.07813	0.09375	0.17188
Man	1024 x 1024 x 1	0.25000	0.48438	0.73438
		Color		
Tree	256 x 256 x 3	0.03125	0.04688	0.07813
Mandrill	512 x 512 x 3	0.15625	0.25000	0.40625
Peppers	512 x 512 x 3	0.18750	0.28125	0.46875
Stockton	1024 x 1024 x 3	0.79688	1.50000	2.29688

Conclusion

- Introduced was a novel & fast secret key encryption algorithm that can be used to encrypt grayscale or color images of the size $N \times N$, where $N = 2^r; r > 4$ without the lost of information
- General case for encryption, which was stated to be for $N = 2^r$, when $r > 8$, and assumes that the partition $P = (r, r-1, r-2, \dots, 4, r)$
- Computer simulations completed in MATLAB were provided, and showed that the encryption method was very fast, while it created encrypted images with very small correlation coefficients

Questions?

References

Y. Mao, G. Chen and S. Lian, "A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps," International Journal of Bifurcation and Chaos, vol. 14, no. 10, pp. 3613-3624, 2004.

A. Grigoryan and M. M. Grigoryan, "Two-Dimensional Fourier Transform in the Tensor Presentation and New Orthogonal Functions," *Avtometria, AS USSR Siberian Section*, no. 1, pp. 21-27, 1986.

A. Grigoryan and S. Agaian, Multidimensional Discrete Unitary Transforms: Representation, Partitioning and Algorithms, New York: Marcel Dekker, 2003.

A.M. Grigoryan and B. Wiatrek, "Cell-Phone Medical Image Encryption Based on Around Spirals Method," Chapter 11 in *Mobile Imaging for Healthcare Applications*, J. Tang, A. Sos, and J. Tan, Eds., SPIE Press, Bellingham, Washington, (in press).

A. M. Grigoryan, B. A. Wiatrek and S. S. Again, "Image Encryption by Redirection and Cyclical Shift," in *SPIE Sensing Technology + Applications Conference*, Baltimore, Maryland (US), 2015

M. Grigoryan, "An Algorithm for Computing the Discrete Fourier Transform with Arbitrary Orders," *Journal Vichislitelnoi Matematiki i Matematicheskoi Fiziki, AS USSR*, vol. 30, no. 10, pp. 1576-1581, 1991.

A. M. Grigoryan, "New Algorithms for calculating discrete Fourier Transforms," *Journal Vichislitelnoi Matematiki i Matematicheskoi Fiziki, AS USSR*, vol. 26, no. 9, pp. 1407-1412, 1986.

A. M. Grigoryan and N. Du, "Principle of Superposition by Direction Image," *IEEE Trans. on Image Processing*, vol. 20, no. 9, pp. 2531-2541, September 2011.

A. M. Grigoryan, "Fourier Transform Representation by Frequency-Time Wavelets," *Signal Processing, IEEE Transactions on*, vol. 53, no. 7, pp. 2489-2497, 2005.