ON COMBINING ENCRYPTION FOR MULTIPLE DATA STREAMS

Adeel Bhutta and Hassan Foroosh

School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL, USA
{abhutta, foroosh} @ cs.ucf.edu

ABSTRACT

A novel technique is proposed to combine the encryption of multiple data streams by generating a *single* encrypted stream, from which any of the original data streams can subsequently be decrypted. By transforming the original data onto random bases and using specially selected duals, a secure method is proposed for the simultaneous encryption and restricted decryption of data of any dimension. These special duals allow the simple addition of transformation coefficients while allowing perfect decryption of each data stream, which otherwise is not possible. Experimentation in encrypting different images and multiple videos are presented, demonstrating the applicability of our approach. We show that our encryption technique is not only easy to code but is highly secure. The proposed technique can be used in any application where data protection is desired from unauthorized users or where restricted access is required.

1. Introduction

Data security has taken center stage in nearly all modern applications not limited to LANs, WANs, Pay-per-View and Video-on-Demand. With the recent proliferation of wireless and communication networks, the role of data security has become immensely critical. During last few decades researchers have invested considerable time and effort in devising ways to protect data from unauthorized users. In this paper, we propose a novel approach to data encryption with applications in areas where restricted authorization is to be provided to users. Situations where data decryption is to be restricted, i.e., where only a certain predefined portion of the data should be decrypted by a certain user, find frequent application in day to day life. Cable companies provide a wide variety of channels, but allow different users access different channels depending on their type of subscriptions. Computational software often provide options for various toolboxes depending on the purchase made by the consumer. In general, with the increasing requirements for personalized services, there is a need to develop methods for such restricted decryption.

We focus on the problem of simultaneously encrypting multiple data streams, while allowing restricted decryption depending on the authorization of the 'decryptor'. We formulate encryption as a transformation of data onto specific bases. We note and exploit an interesting property of random matrices, using them as bases to transform the data. Given the transformation matrix, the original data can then be simply decrypted by inversion and 're-transformation'. We use this paradigm to simultaneously encrypt multiple data streams (i.e., multiple images or video frames) while allowing restricted decryption. The primary contribution of this work is to allow simple addition of transformation coefficients while perfectly reconstructing encrypted data when summed coefficients are used for reconstruction. This simple addition of coefficients is not possible with any other transformation, making this work particularly unique. The resulting algorithm which works for both compressed and uncompressed data is as secure as the random number generator used.

The rest of the paper has been organized as follows: Section 2 reviews the related work in this area while Section 3 covers the problem formulation and describes the proposed solution. Section 4 discusses the proposed encryption algorithm, followed by results, discussion and conclusions in Section 5, Section 6 and Section 7 respectively.

2. Related Work

Data encryption, a way to protect data from unwanted users, has widely been studied by researchers over last few decades [6]-[13]. Although many encryption techniques exist in literature [4] but modern technological demands have raised the stakes for more reliable and secure encryption. In general, encryption techniques can roughly be classified into two categories: those that deal with the value of data [7], [10] and others that have to do with the position of data [9]. Our technique falls into the first category. These techniques can also be classified in terms of security and computational complexity. Generally, encryption algorithms such as Triple-DES [12] and RC5 [13] that are computationally

complex, are considered secure. On the other hand, chaotic encryptions that require simple computational procedures offer less secure alternatives [6], [10], [11]. Our technique lies somewhere between these two classes.

Yen and Guo [7] proposed a method that encrypted images on a per pixel basis. Grey level values of each pixel were encrypted bit-by-bit using one of the two predetermined keys, generated by a chaotic system. In contrast, our technique does not encrypt every pixel separately. Instead, we propose to encrypt the image data *collectively*. Li and Zheng [8] later pointed out that Yen and Guo's technique is not secure when any secret key is repeatedly used to encrypt more than one image. In contrast, our technique does not have this problem because in our technique any key matrix can only decrypt its own data. The idea of multiplexing several signals through superframes was first coined by Balan [2] when he proposed it to be used for time and frequency division multiplexing of signals. Kuo's work in [3] is more relevant to image multiplexing but he has used an m-sequence to obtain the encrypted image whereas the original image is recovered through correlation with the corresponding m-sequence. The idea of coefficient summation was initially proposed by Han and Larson [1] when they proved that dual frames can be constructed in continuous domain with the summable property. This paper extends their idea to discrete domain and proposes a technique that can achieve summable coefficients for data of any dimension such as signals, images and videos.

3. Problem Formulation

In this section we describe the process of simultaneously encrypting multiple data streams and showing how restricted decryption can be performed. Consider the available data $\mathbf{x}$, an $M \times N$. We can transform the data onto a M-dimensional basis defined by Φ using a linear transformation $\mathbf{c} = \Phi\mathbf{x}$, where $\mathbf{c}$ are the coefficients of the basis of transformation. The original data can then be recovered through an inverse transformation $\mathbf{x}' = \tilde{\Phi}\mathbf{c}$ where $\tilde{\Phi}$ is the dual of Φ (i.e., $\tilde{\Phi}\Phi = \mathbf{I}$). The aforementioned process can be conceptualized as an encryption-decryption process, since the original data cannot be recovered unless one has access to the basis Φ, referred to as the encryption basis. It is observed that any random matrix forms a set of linearly independent vectors and can therefore be used as a bases for transformation. This straightforward transformation onto a random basis forms the backbone of the proposed approach.

3.1 Encrypting Two Data Streams

Consider $\mathbf{x_1}$ and $\mathbf{x_2}$ are the available data, both of size $M \times N$. To achieve *single* encrypted data stream (or matrix), data is encrypted using two M-dimensional random bases, however, the coordinate vectors are of length $2M$. As a result, the column vectors of $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ span a M-dimensional subspaces each of length $2M$. Thus,

$$\mathbf{c_1} = \mathbf{\Phi_1 x_1} \tag{1}$$
$$\mathbf{c_2} = \mathbf{\Phi_2 x_2} \tag{2}$$

where $\mathbf{c_1}$ and $\mathbf{c_2}$ matrices represent the coefficients of the bases (Φ_1 and Φ_2) of transformation. To achieve a single encrypted matrix $\mathbf{C}$, we sum the coefficient matrices together,

$$\mathbf{C} = \mathbf{c_1} + \mathbf{c_2} \tag{3}$$

Ordinarily, such a relationship cannot exist if any other technique is used, because perfect reconstruction of data becomes impossible when several coefficient matrices are added together. To reconstruct the original signals from the summed coefficients,

$$\mathbf{x_1'} = \tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{C} \tag{4}$$
$$\mathbf{x_2'} = \tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{C} \tag{5}$$

Now,

$$\begin{aligned}\mathbf{x_1'} = \tilde{\mathbf{\Phi}}_\mathbf{1}(\mathbf{c_1} + \mathbf{c_2}) &= \tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{c_1} + \tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{c_2} \\ &= \tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{\Phi_1 x_1} + \tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{\Phi_2 x_2}\end{aligned}$$

Hence, if we choose $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ such that $\tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{\Phi_2} = 0$, i.e. that the subspaces spanned by $\tilde{\mathbf{\Phi}}_1$ and $\mathbf{\Phi_2}$ are orthogonal, then we can see that $\mathbf{x_1'} = \mathbf{x_1}$.

Similarly, we can derive $\mathbf{x_2'} = \mathbf{x_2}$ from

$$\begin{aligned}\mathbf{x_2'} = \tilde{\mathbf{\Phi}}_\mathbf{2}(\mathbf{c_1} + \mathbf{c_2}) &= \tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{c_1} + \tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{c_2} \\ &= \tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{\Phi_1 x_1} + \tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{\Phi_2 x_2}\end{aligned}$$

if $\tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{\Phi_1} = 0$. Thus, four constraints are proposed for the general solution of the problem,

$$\tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{\Phi_2} = 0 \tag{6}$$
$$\tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{\Phi_1} = 0 \tag{7}$$
$$\tilde{\mathbf{\Phi}}_\mathbf{1}\mathbf{\Phi_1} = \mathbf{I} \tag{8}$$
$$\tilde{\mathbf{\Phi}}_\mathbf{2}\mathbf{\Phi_2} = \mathbf{I} \tag{9}$$

Since $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ are known (the initial encryption matrices), $\tilde{\mathbf{\Phi}}_1$ and $\tilde{\mathbf{\Phi}}_2$ can be computed by finding the least squares solution of the system,

$$\mathbf{A} \cdot \mathbf{Y} = \mathbf{B} \tag{10}$$

where,

$$\mathbf{A} = \begin{pmatrix} \mathbf{\Phi_1} & 0 \\ \mathbf{\Phi_2} & 0 \\ 0 & \mathbf{\Phi_1} \\ 0 & \mathbf{\Phi_2} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} \tilde{\mathbf{\Phi}}_\mathbf{1} \\ \tilde{\mathbf{\Phi}}_\mathbf{2} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \mathbf{I} \\ 0 \\ 0 \\ \mathbf{I} \end{pmatrix}.$$

where 0 (zero matrix) and $\mathbf{I}$ (identity matrix) are the same sizes as $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$. It should be noted that the duals can be constructed if matrix $\mathbf{A}$ is non-singular. Since it is made up of several randomly generated matrices the probability of $\mathbf{A}$ being singular is extremely small [5]. It should be emphasized that the perfect reconstruction of original data from the summed coefficients is only possible if the constraints in (6)-(9) are satisfied.

3.2 Encrypting Multiple Data Streams

The extension of this formulation to multiple streams is a simple extrapolation. For data $\{\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_n}\}$, a set of random encryption matrices are required $\{\mathbf{\Phi_1}, \mathbf{\Phi_2}, \ldots \mathbf{\Phi_n}\}$, each spanning an M-dimensional subspace of an nM-space. The coefficients can be computed directly,

$$\mathbf{c_i} = \mathbf{\Phi_i x_i}, \quad \text{where} \quad i = 1, ..., n \tag{11}$$

The coefficients can then be summed,

$$\mathbf{C} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{c_i} \tag{12}$$

and the encrypted matrix $\mathbf{C}$ can be transmitted. Finally, the correct duals can be computed,

$$\mathbf{A} \cdot \mathbf{Y} = \mathbf{B} \tag{13}$$

where,

$$\mathbf{A} = \begin{pmatrix} \mathbf{\Phi_1} & 0 & \cdots & 0 \\ 0 & \mathbf{\Phi_1} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Phi_1} \\ \vdots & & & \vdots \\ \mathbf{\Phi_n} & 0 & \cdots & 0 \\ 0 & \mathbf{\Phi_n} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Phi_n} \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} \tilde{\mathbf{\Phi}}_1 \\ \tilde{\mathbf{\Phi}}_2 \\ \vdots \\ \tilde{\mathbf{\Phi}}_n \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{I} & 0 & \ldots & 0 & \mathbf{I} & 0 & \ldots & 0 & \mathbf{I} \end{pmatrix}^{\mathbf{T}}.$$

4. Encryption Algorithm

Although this technique can be used for encrypting any type of data, here we have presented its application to images and videos. We create random bases to project our data into summable coefficients. We, then, can achieve perfect reconstruction of images using specially selected duals of our encryption bases. The block diagram for the entire encryption and decryption process is shown in Fig. 1.

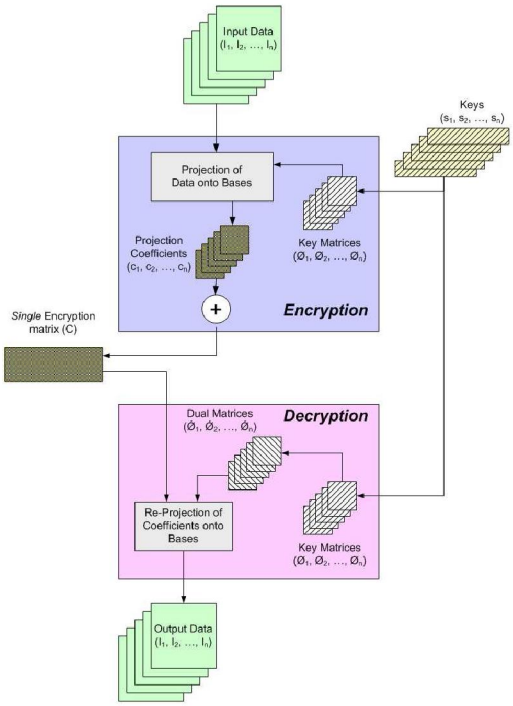

Fig. 1. Encryption and Decryption Process

Given n images $I_1, I_2, \ldots, I_n$, a set of random matrices (referred as key matrices) $\mathbf{\Phi_1}, \mathbf{\Phi_2}, \ldots, \mathbf{\Phi_n}$ can be generated from seed values $\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_n}$ to compute $\mathbf{A}$ and $\mathbf{B}$ and setup the system in (13). We can find $\mathbf{Y}$ in order to get duals of the bases (or key matrices) $\tilde{\mathbf{\Phi}}_1, \tilde{\mathbf{\Phi}}_2, \ldots, \tilde{\mathbf{\Phi}}_n$ using the least squares solution, and subsequently calculate $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_n}$ using (11). We then can generate a *single* encrypted data $\mathbf{C}$ using (12) which can be used to decrypt data using a particular key matrix.

5. Results

The proposed algorithm was implemented to encrypt three images. Input images and encryption results are shown in Fig.2(a) and (b) respectively. It is worth noting that all of these appear to be completely random but when used for decryption, produce a perfect reconstruction of images shown in Fig.2(c).

No. of Images	Size (pixels)	Total Time	LSQ Time	% of Total Time
2	100x100	0.093	0.062	66.7
3	100x100	0.59	0.49	83
5	100x100	8.89	8.36	94

Table 1. Encryption and decryption time (in seconds)

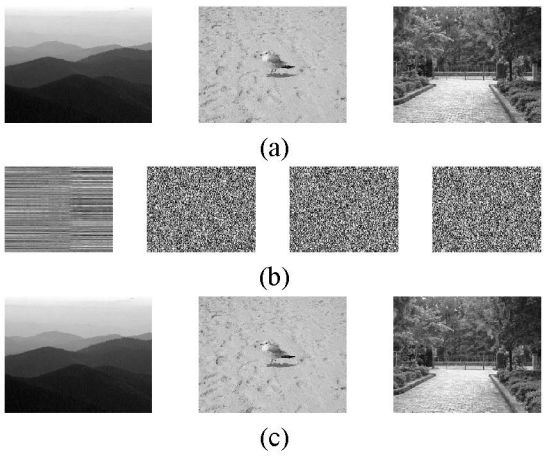

Fig. 2. Image encryption for three images: (a) Original Images, (b) (leftmost) Encrypted Matrix and the three Key Matrices. (Note that the encryption and key matrices are not shown to the scale or aspect ratio) (c) Reconstructed Images.

The experiment was repeated for a set of nine different images shown in Fig.3(a) and the encryption results are shown in Fig.3(b). The value of the mean square error (14) between the reconstructed image $\hat{I}$ and original image I is again negligible (i.e., perfect reconstruction).

The average processing times [1] for the proposed algorithm are shown in Table 1. It should be noted that the computation of the least squares solution takes most of the computation time. For two JPEG images of size 400x318, the average computation time for the complete algorithm is 3.35 seconds whereas the average computation time of the least square part only is 2.55 seconds (76%). The total processing time reported for the same image in Sobhy's work [11] is 20 seconds. Our technique reduces the processing time by 83% in this case.

The processing time for our technique can further be improved by reusing old bases (i.e., key matrices). Fig.4 shows three new images which were reconstructed using the same bases that we generated for images in Fig.2(c). The results show perfect reconstruction (i.e., MSE is [9×10^{-21}, 8×10^{-23}]). This is useful for simultaneous encryption of multiple videos. For instance, A digital cable company may want to show three movies to three different subscribers while every subscriber should have only one key for the entire movie. This can be achieved by encrypting one frame from all three movies together, generating a single encrypted matrix and three key matrices. For all subsequent frames, the same bases will be used to generate new encrypted image. We will send keys to our subscribers only once in the beginning of the transmission, while a single encrypted matrix will be transmitted for every frame. This reduces the computation requirements on the subscriber end to a simple reconstruction operation (i.e., convolution).

[1] using MATLAB on Intel desktop Pentium IV, 2.4GHz, 1GB RAM

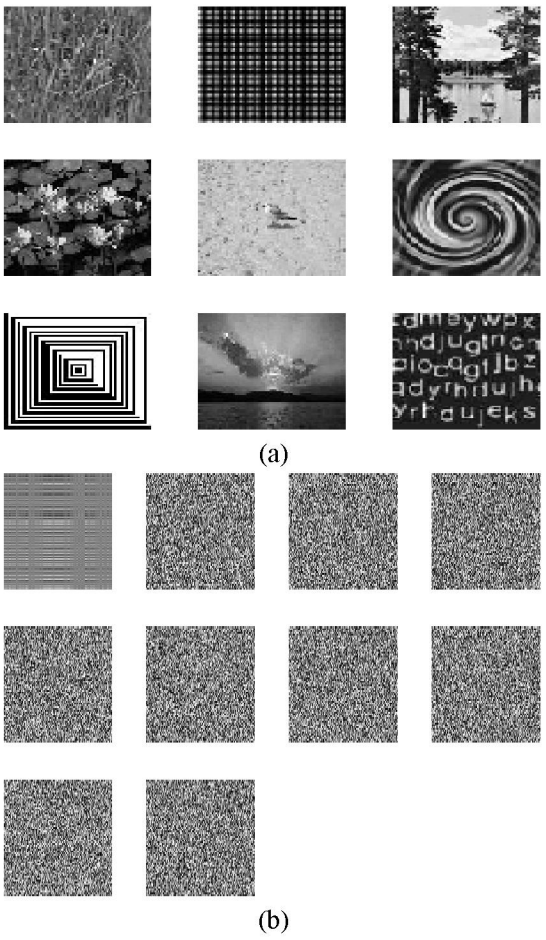

Fig. 3. Image encryption for nine images: (a) Original Images, (b) (top left) Encrypted Matrix and the nine Key matrices. (Note that the encryption and key matrices are not shown to the scale or aspect ratio).

$$MSE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - \hat{I}(i,j)]^2. \quad (14)$$

Encryption results for image sequences are shown in Fig.5. Each row corresponds to successive frames of one image sequence and the leftmost column shows the key matrices used for that sequence. The last row represents a *single* encrypted matrix of corresponding frames. It should be pointed out that each subscriber is provided a unique key which will ensure that he would only be able to access his own data. The ability of our technique to provide unique keys within an encryption process and reusability of keys across several encryption processes makes it superior to other techniques.

Fig. 4. Image encryption for three images when reusing old bases.

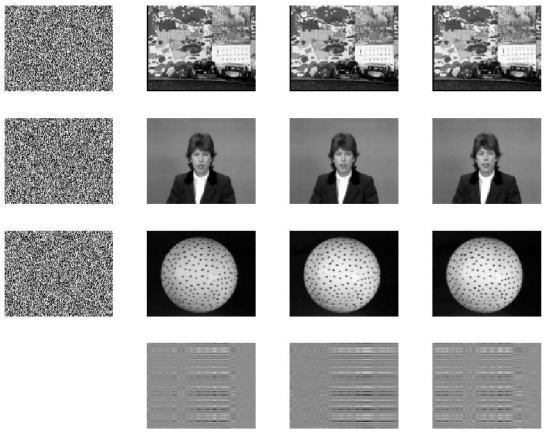

Fig. 5. Simultaneous encryption of three standard image sequences: leftmost column shows key matrices for each user and each row corresponds to successive frames of one image sequence. The last row shows a *single* encrypted matrix for corresponding frames in that column. (Note that the encryption and keys matrices are not shown to the scale or aspect ratio)

6. Discussion

In order to evaluate the security of our encryption technique, we devised a simple 'attack' mechanism. A random key matrix was generated from a seed value and valid encrypted image was used to reconstruct the image. Reconstruction error for each image was consistently high [5.06×10^5, 1.92×10^6], which shows that the encryption process is highly secure. The experiment was repeated **one million** times and results are shown in Fig.6(b). Results of the first few values have been magnified in Fig.6(a). It should be emphasized that a binary random number generator can generate $2^{M \times N}$ different key matrices for encrypting two data streams of size $M \times N$. This means that one will need approximately 2.6803×10^{254} years to find a key matrices for encrypted image of size 30×30 using brute force if one attempt takes 1 nSec.

In order to emphasize that a partial encrypted image will not give a good reconstruction, we observed the impact of changing the encrypted image on the reconstruction error. We introduced invalid values in the encrypted image one column at a time. Although the valid key matrix was used, reconstruction error was quite high [0.5×10^3, 2.7×10^4]. These results verify that perfect reconstruction is possible only with a valid encrypted image and valid key matrix.

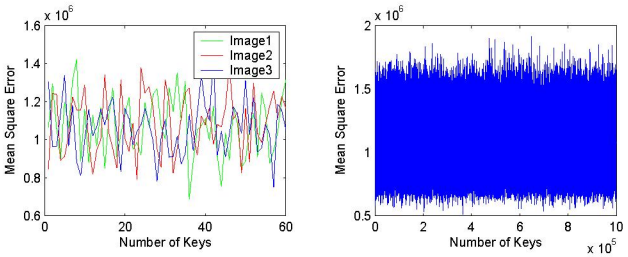

Fig. 6. MSE when random key matrices are used for image reconstruction: (a) First 60 random key matrices, (b) One million random key matrices

Although our encryption algorithm generates a key matrix for each image in the sequence, it does not require huge bandwidth as it may appear. This is due to the fact that these key matrices are generated locally using random number generators that use seed values (or state of the system) provided by the user in order to generate a key matrix. It should be mentioned that a random number generator can provide same key matrix if a seed value (or state of the system) is provided. This reduces the memory requirements for key matrices from several kilobytes to few characters when generated locally. Therefore, the knowledge of seed value along with type of random number generator used is sufficient to generate key matrices locally. We used 931316785 as seed value and [3.6193×10^9, 3.2483×10^9] as state of the system. Our use of seed values (or state of the system) makes this technique at par with any encryption technique hence needing a seed value and *single* encrypted matrix for complete encryption-decryption process.

7. Conclusions

We have proposed an algorithm to combine the encryption of multiple data streams into a *single* encrypted stream. We have demonstrated that such a stream (or matrix) is sufficient for simultaneous encryption of data streams. Although this technique can be used for encrypting any type of data, only results for multiple images and videos have been presented. The encryption and decryption process uses random bases and their specially selected duals making it highly secure. An attack study has been presented to evaluate the security of the algorithm. We have also demonstrated that the key matrices can be re-used without compromising the security of our encryption technique. It has also been shown that our encryption algorithm is not only simple, and computationally efficient but also has acceptable bandwidth requirements. On the basis of results presented, we believe that this technique outperforms other encryption techniques and can be used in a wide variety of real world applications.

8. References

[1] Han D. and Larson D.R., "Frames, Bases and Group Representation", Memoirs of the American Mathematical Society, No. 697, Sep 2000.

[2] Balan Radu, "Multiplexing of Signal using Superframes", SPIE: Wavelets Applications in Signal and Image Processing VIII, vol.4119, pp.118-130, 2000.

[3] Kuo C.J. and Harriett R., "Image Multiplexing by Code Division Technique", SPIE: Applications of DIP XII, Vol. 1153, 1989.

[4] Ming Yang, Nikolaos Bourbakis and Shujun Li, "Data-Image-Video Encryption", IEEE Potentials, Vol. 23, No. 3, pp. 28-34, Aug-Sep, 2004.

[5] Kahn J., Komls J., and Szemeredi E., "On the Probability that a Random +/-1 Matrix is Singular", Journal of American Mathematical Society 8, pp.223-240, 1995.

[6] Franco C., Lorenzo C., Ennio G., Paola P., and Maurizio R., "A New Chaotic Algorithm for Video Encryption", IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, Nov 2002.

[7] Yen C.J and Guo J.I., "A New Chaotic Image Encryption Algorithm and its VLSI architecture", Proceedings of IEEE Workshop on Signal Processing Systems, pp. 430-437,1999.

[8] Li S. and Zheng X., "On the Security of an Image Encryption Method", Proceedings of IEEE International Conference on Image Processing, Vol. 2, pp. 925-928, 2002.

[9] Kuo C.J. and Chen M. S., "A New Signal Encryption Technique and its Attack Study", IEEE International Conference on Security Technology, pp. 149-153, 1991.

[10] Fridrich J., "Image Encryption based on Chaotic Maps", IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulations, pp.1105-1110, 1997.

[11] Sobhy M.I. and Shehata A.R., "Chaotic Algorithms for Data Encryption", Proc. IEEE Acoustic, Speech and Signal Processing, 2001, pp. 997-1000.

[12] National Bureau of Standards, "Data Encryption Standard", U.S. Department of Commerce, FIPS pub. 46, Jan 1977.

[13] Ron Rivest,"The RC5 Encryption Algorithm," Dr. Dobb's Journal, vol. 20 no. 1, pp. 146-148. January 1995.